aws

管理指南

# Amazon Redshift



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

# Amazon Redshift: 管理指南

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其它商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

# Table of Contents


..... xvi
什么是 Amazon Redshift？ ..... 1
您是首次接触 Amazon Redshift 用户吗？ ..... 1
Amazon Redshift Serverless 功能概览 ..... 2
Amazon Redshift 预置集群概览 ..... 4
集群管理 ..... 4
集群访问和安全性 ..... 5
监控集群 ..... 6
数据库 ..... 7
将 Amazon Redshift Serverless 与 Amazon Redshift 预调配数据仓库进行比较 ..... 7
使用预置集群的 Amazon Redshift 管理界面 ..... 26
使用 AWS SDK ..... 27
对 HTTP 请求进行签名 ..... 28
设置 Amazon Redshift CLI ..... 33
Amazon Redshift Serverless ..... 34
什么是 Amazon Redshift Serverless？ ..... 34
Amazon Redshift Serverless 控制台 ..... 34
使用 Amazon Redshift Serverless 时的注意事项 ..... 38
Amazon Redshift Serverless 的计算容量 ..... 41
Serverless 端点容量的注意事项和限制 ..... 43
AI 驱动的扩展和优化 ..... 44
Amazon Redshift Serverless 的计费 ..... 49
计算容量的计费 ..... 49
按需计算容量的计费 ..... 50
无服务器预留的计费 ..... 53
对存储计费 ..... 59
使用 Amazon Redshift Serverless 免费试用版 ..... 60
账单使用注释 ..... 60
使用连接池的 Amazon Redshift Serverless 计费 ..... 61
使用零 ETL 的 Amazon Redshift Serverless 的成本优化 ..... 62
连接到 Amazon Redshift Serverless ..... 62
连接到 Amazon Redshift Serverless ..... 63
通过 JDBC 驱动程序连接到 Amazon Redshift Serverless ..... 63
使用数据 API 连接到 Amazon Redshift Serverless ..... 64

使用 SSL 连接到 Amazon Redshift Serverless ..... 65
从 Amazon Redshift 托管式 VPC 端点连接到 Amazon Redshift Serverless ..... 67
从接口 VPC 端点连接到 Amazon Redshift Serverless（AWS PrivateLink） ..... 67
从其它账户中的 Redshift VPC 端点连接到 Amazon Redshift Serverless ..... 67
其他资源 ..... 72
定义联合用户的数据库角色 ..... 72
定义数据库角色 ..... 73
定义授予联合用户的数据库角色的使用场景 ..... 73
其他资源 ..... 74
Amazon Redshift Serverless 中的 Identity and Access Management ..... 75
授予权限 ..... 75
IAM 凭证入门 ..... 76
使用数据库角色权限访问数据库对象 ..... 77
将预置集群迁移到 Amazon Redshift Serverless ..... 78
创建预置集群的快照 ..... 79
使用驱动程序连接到 Amazon Redshift Serverless ..... 80
使用 Amazon Redshift Serverless SDK ..... 86
工作组和命名空间 ..... 86
使用控制台的工作组和命名空间 ..... 86
使用 AWS Command Line Interface 和 Amazon Redshift Serverless API 的工作组和命名空间 ..... 87
工作组 ..... 88
命名空间 ..... 92
监控查询和工作负载 ..... 95
添加查询监控策略 ..... 96
向用户授予查询监控权限 ..... 98
向角色授予查询监控权限 ..... 98
设置使用限制 ..... 99
设置查询限制 ..... 100
设置查询队列 ..... 100
使用控制面板检查摘要数据 ..... 104
审计日志记录 ..... 105
CloudWatch 中的日志事件 ..... 105
CloudWatch 指标 ..... 106
快照和恢复点 ..... 112
AWS Backup 集成 ..... 113

创建快照 ..... 113
生成最终快照 ..... 114
共享快照或移除快照权限 ..... 115
计划快照 ..... 115
更新快照保留期 ..... 118
删除快照 ..... 118
还原快照 ..... 119
转换恢复点 ..... 120
还原恢复点 ..... 120
将备份复制到其他 AWS 区域 ..... 120
还原表 ..... 121
数据共享 ..... 122
注意事项 ..... 123
授予查看数据共享的访问权限 ..... 123
将命名空间注册到 AWS Glue Data Catalog ..... 124
标注资源 ..... 124
Amazon Redshift 预置集群 ..... 126
集群和节点 ..... 126
节点类型详细信息 ..... 128
确定节点数 ..... 131
使用 EC2 创建集群 ..... 132
Amazon Virtual Private Cloud (Amazon VPC) ..... 132
默认磁盘空间警报 ..... 133
集群状态 ..... 133
使用预置集群的注意事项 ..... 135
区域和可用区注意事项 ..... 135
集群维护 ..... 136
集群操作 ..... 149
创建集群 ..... 149
创建磁盘空间警报 ..... 152
查看集群 ..... 152
修改集群 ..... 153
调整集群大小 ..... 154
重命名集群 ..... 168
升级集群的发行版本 ..... 168
暂停和恢复集群 ..... 169

重新引导集群 ..... 171
重新定位集群 ..... 171
设置用量限制 ..... 175
关闭和删除集群 ..... 176
快照和备份 ..... 177
将集群注册到 AWS Glue Data Catalog ..... 198
多可用区部署 ..... 199
设置多可用区部署 ..... 199
创建新集群时设置多可用区 ..... 202
为从快照还原的数据仓库设置多可用区 ..... 204
将单可用区数据仓库转换为多可用区数据仓库 ..... 205
将多可用区数据仓库转换为多可用区数据仓库 ..... 207
多可用区数据仓库大小调整 ..... 208
多可用区部署失效转移 ..... 208
查看多可用区数据仓库的查询和负载 ..... 210
监控多可用区部署中的查询 ..... 211
终止对集群的查询 ..... 212
监控集群性能 ..... 212
性能数据 ..... 213
查看性能数据 ..... 226
分析查询执行 ..... 248
创建警报 ..... 250
终止运行的查询 ..... 250
CloudWatch 控制台中的性能指标 ..... 251
查询分析器 ..... 252
查询和数据库监控 ..... 261
基于 SYS 视图的查询和数据库监控 ..... 267
Tracks ..... 270
在跟踪之间切换 ..... 270
跟踪和还原 ..... 271
更新无服务器工作组 ..... 271
管理版本 ..... 272
确定工作组或集群版本 ..... 272
零 ETL 集成 ..... 274
注意事项 ..... 276
在目标上使用历史记录模式时的注意事项 ..... 277

零 ETL 集成源是 Aurora 或 Amazon RDS 时的注意事项 ..... 278
零 ETL 集成源是 DynamoDB 时的注意事项 ..... 279
当零 ETL 集成源是应用程序（例如 Salesforce、SAP、ServiceNow 和 Zendesk）时的注意事项 ..... 280
还原零 ETL 集成目标时的注意事项 ..... 280
调整零 ETL 集成目标的大小时的注意事项 ..... 281
开始使用零 ETL 集成 ..... 282
创建和配置目标 Amazon Redshift 数据仓库 ..... 282
开启区分大小写 ..... 283
在 Amazon Redshift 中配置授权 ..... 284
创建零 ETL 集成 ..... 289
创建目标数据库 ..... 299
查询复制的数据 ..... 301
使用实体化视图查询复制的数据 ..... 302
从 DynamoDB 查询复制的数据 ..... 304
查看零 ETL 集成 ..... 305
历史记录模式 ..... 308
共享数据 ..... 310
监控零 ETL 集成 ..... 310
使用 Amazon Redshift 系统视图监控零 ETL 集成 ..... 310
监控与 Amazon EventBridge 的零 ETL 集成 ..... 311
零 ETL 集成的指标 ..... 311
修改 DynamoDB 的零 ETL 集成 ..... 314
删除 DynamoDB 的零 ETL 集成 ..... 315
支持的区域 ..... 316
Aurora MySQL ..... 316
Aurora PostgreSQL ..... 318
Amazon DynamoDB ..... 320
Amazon RDS for MySQL ..... 322
企业应用程序 ..... 324
零 ETL 集成问题排查 ..... 326
查询数据库 ..... 341
连接到 Amazon Redshift ..... 341
使用 Amazon Redshift 查询编辑器 v2 查询数据库 ..... 342
配置您的 AWS 账户 ..... 343
打开查询编辑器 v2 ..... 349

连接到 Amazon Redshift 数据库 ..... 353
浏览 Amazon Redshift 数据库 ..... 355
创建数据库对象 ..... 357
查看查询和选项卡历史记录 ..... 364
与 Amazon Q 生成式 SQL 交互 ..... 364
将数据加载到数据库 ..... 376
编写查询 ..... 384
笔记本 ..... 389
查询 AWS Glue Data Catalog ..... 392
查询数据湖 ..... 395
数据共享 ..... 397
计划查询 ..... 400
可视化结果 ..... 410
以团队形式协作和共享 ..... 415
使用查询编辑器 v1 查询数据库 ..... 416
注意事项 ..... 417
使用 SQL 客户端工具连接到数据仓库 ..... 417
关于连接客户端工具的建议 ..... 418
在 Amazon Redshift 中配置连接 ..... 418
用于发现元数据的最佳实践 ..... 607
配置连接的安全选项 ..... 614
通过客户端工具和代码连接 ..... 621
使用身份验证配置文件连接到 Amazon Redshift ..... 667
解决 Amazon Redshift 中的连接问题 ..... 670
使用 Data API ..... 677
使用数据 API ..... 677
调用数据 API 时的注意事项 ..... 677
选择数据库身份验证凭证 ..... 680
映射 JDBC 数据类型 ..... 680
运行带有参数的 SQL 语句 ..... 681
运行带有幂等性令牌的 SQL 语句 ..... 683
运行带有会话重用的 SQL 语句 ..... 684
获取结果 ..... 685
授予访问权限 ..... 688
可信身份传播 ..... 697
调用 Data API ..... 699

解决 Data API 问题 ..... 734
使用 Amazon EventBridge 计划数据 API 操作 ..... 734
监控数据 API ..... 738
结合使用 AWS KMS 与 Data API ..... 740
使用 Amazon SageMaker 融通式合作开发工作室 ..... 743
参数组 ..... 744
默认参数值 ..... 744
工作负载管理 ..... 746
WLM 动态和静态属性 ..... 747
WLM 配置参数的属性 ..... 747
使用 AWS CLI 配置 WLM 参数 ..... 753
创建参数组 ..... 760
修改参数组 ..... 761
创建查询监控规则 ..... 765
删除参数组 ..... 766
与 AWS 合作伙伴集成 ..... 767
加载 AWS 合作伙伴的数据 ..... 768
预留节点 ..... 770
预留节点产品 ..... 770
比较不同预留节点产品的定价 ..... 771
预留节点的工作方式 ..... 772
预留节点和整合账单 ..... 773
在使用预留实例的集群上更改节点类型之前需要考虑的事项 ..... 773
预留节点示例 ..... 773
示例 1 ..... 774
示例 2 ..... 774
示例 3 ..... 774
示例 4 ..... 774
示例 5 ..... 774
示例 6 ..... 774
购买预留节点 ..... 775
安全性 ..... 778
数据保护 ..... 779
数据加密 ..... 780
数据令牌化 ..... 796
路由互联网络流量 ..... 796

身份和访问管理 ..... 797
使用身份进行身份验证 ..... 797
防止跨服务混淆座席 ..... 798
访问控制 ..... 800
有关管理访问的概述 ..... 800
使用基于身份的策略（IAM 策略） ..... 806
原生身份提供者（IdP）联合身份验证 ..... 863
使用服务关联角色 ..... 892
使用 IAM 身份验证生成数据库用户凭证 ..... 900
授权访问 AWS 服务 ..... 949
管理管理员密码 ..... 982
AWS Secrets Manager 集成所需的权限 ..... 983
管理员密码密钥轮换 ..... 984
将 AWS Secrets Manager 与 Amazon Redshift 配合使用时的注意事项 ..... 984
检索密钥的 ARN ..... 984
为数据库连接凭证创建密钥 ..... 985
日志记录和监控 ..... 988
数据库审计日志记录 ..... 988
使用 Cloudtrail 进行日志记录 ..... 1004
系统表与 S3 表类数据存储服务集成 ..... 1015
合规性验证 ..... 1027
恢复能力 ..... 1029
基础结构安全性 ..... 1029
网络隔离 ..... 796
安全组 ..... 1031
接口 VPC 端点 ..... 1031
配置和漏洞分析 ..... 1038
使用 Identity Center 身份验证 ..... 1039
先决条件 ..... 1039
Identity Center 身份验证的工作原理 ..... 1040
GetIdentityCenterAuthToken API 操作 ..... 1040
适用于预置集群的 GetIdentityCenterAuthToken ..... 1040
适用于无服务器工作组的 GetIdentityCenterAuthToken ..... 1041
响应语法 ..... 1042
驱动程序集成 ..... 1043
直接使用令牌 ..... 1043

Java 代码示例 ..... 1044
IAM 策略要求 ..... 1044
API 权限 ..... 1044
示例 IAM policies ..... 1045
区域可用性 ..... 1046
问题排查 ..... 1047
API 错误 ..... 1047
连接错误 ..... 1047
联网任务 ..... 1049
用于客户端连接的自定义域名 ..... 1049
注册域名 ..... 1049
为域名请求证书。 ..... 1051
配置自定义域 ..... 1052
连接到预置集群或工作组 ..... 1053
重命名分配了自定义域的集群 ..... 1054
描述自定义域关联 ..... 1054
将自定义域与不同的证书关联 ..... 1055
删除自定义域 ..... 1056
Redshift 托管的 VPC 端点 ..... 1056
注意事项 ..... 1057
授予对 VPC 的访问权限 ..... 1058
创建 Redshift 托管的 VPC 终端节点 ..... 1059
VPC 中的 Redshift 资源 ..... 1059
在 VPC 中创建集群或工作组 ..... 1062
VPC 安全组 ..... 1063
为集群或工作组配置安全设置 ..... 1064
Redshift 资源的子网 ..... 1066
屏蔽 VPC 和子网的公共访问权限 ..... 1069
使用增强型 VPC 路由控制网络流量 ..... 1070
使用 VPC 端点控制数据库流量 ..... 1071
启用增强型 VPC 路由 ..... 1072
通过增强型 VPC 路由查询数据湖表 ..... 1074
Events ..... 1079
集群事件通知订阅 ..... 1079
创建事件通知订阅 ..... 1081
预置集群事件通知 ..... 1082

Amazon Redshift Serverless 事件通知 ..... 1099
零 ETL 集成事件通知 ..... 1109
配额和限制 ..... 1116
Amazon Redshift 对象的配额 ..... 1116
Amazon Redshift Serverless 对象的配额 ..... 1122
Amazon Redshift 数据 API 的配额 ..... 1124
查询编辑器 v2 对象的配额 ..... 1125
Amazon Redshift Spectrum 对象的配额和限制 ..... 1126
命名约束 ..... 1127
标记资源 ..... 1131
标记要求 ..... 1132
管理资源标签 ..... 1132
AWS Backup 集成 ..... 1133
注意事项 ..... 1134
限制 ..... 1134
将 AWS Backup 与 Amazon Redshift 结合使用 ..... 1135
集群版本 ..... 1136
补丁 205 ..... 1137
新功能 ..... 1138
补丁 204 ..... 1139
新功能 ..... 1140
补丁 203 ..... 1142
新功能 ..... 1143
补丁 202 ..... 1145
新功能 ..... 1145
补丁 201 ..... 1147
新功能 ..... 1148
补丁 200 ..... 1149
新功能 ..... 1149
补丁 199 ..... 1151
新功能 ..... 1152
补丁 198 ..... 1153
新功能 ..... 1153
补丁 197 ..... 1154
新功能 ..... 1155
补丁 196 ..... 1156

新功能 ..... 1157
补丁 195 ..... 1157
新功能 ..... 1157
补丁 194 ..... 1158
新功能 ..... 1158
补丁 193 ..... 1159
新功能 ..... 1159
补丁 192 ..... 1159
新功能 ..... 1160
补丁 191 ..... 1160
新功能 ..... 1160
补丁 190 ..... 1161
新功能 ..... 1161
补丁 189 ..... 1161
新功能 ..... 1162
补丁 188 ..... 1162
新功能 ..... 1163
补丁 187 ..... 1163
新功能 ..... 1164
补丁 186 ..... 1165
新功能 ..... 1166
补丁 185 ..... 1167
新功能 ..... 1167
补丁 184 ..... 1168
新功能 ..... 1169
补丁 183 ..... 1169
新功能 ..... 1170
补丁 182 ..... 1171
新功能 ..... 1171
补丁 181 ..... 1172
新功能 ..... 1173
补丁 180 ..... 1174
新功能 ..... 1175
补丁 179 ..... 1176
新功能 ..... 1176
补丁 178 ..... 1177

新功能 ..... 1178
补丁 177 ..... 1180
新功能 ..... 1180
补丁 176 ..... 1181
新功能 ..... 1182
补丁 175 ..... 1183
新功能 ..... 1183
补丁 174 ..... 1184
此版本的新功能 ..... 1184
此版本的新功能 ..... 1184
此版本的新功能 ..... 1184
此版本的新功能 ..... 1184
此版本的新功能 ..... 1184
此版本的新功能 ..... 1184
此版本的新功能 ..... 1184
补丁 173 ..... 1185
此版本的新功能 ..... 1185
此版本的新功能 ..... 1185
此版本的新功能 ..... 1185
此版本的新功能 ..... 1185
此版本的新功能 ..... 1185
此版本的新功能 ..... 1185
此版本的新功能 ..... 1185
此版本的新功能 ..... 1185
此版本的新功能 ..... 1185
此版本的新功能 ..... 1185
补丁 172 ..... 1186
新功能 ..... 1187
补丁 171 ..... 1187
新功能 ..... 1188
补丁 170 ..... 1188
新功能 ..... 1188
补丁 169 ..... 1188
新功能 ..... 1189
补丁 168 ..... 1189
新功能 ..... 1189

行为更改 ..... 1190
即将发生的行为更改 ..... 1190
从 2026 年 9 月 30 日起，Amazon Redshift 会为暂停的数据共享创建者集群自动拍摄快照 1190
从补丁 206 开始，SUPER 数据类型支持更大的单个字符串 ..... 1191
2026 年 8 月 17 日之后，强制执行 Amazon Redshift Serverless API 的 AWS KMS 密钥权限 ..... 1192
从补丁 208 开始，Amazon Redshift 会在多次登录尝试失败后强制执行用户锁定 ..... 1193
Amazon Redshift Serverless 和 Amazon Redshift RG 实例上手动快照的增强计费模式于 2026 年 6 月 8 日生效 ..... 1193
从补丁 202 开始，Lake Formation 表上的 Iceberg DELETE 需要 DELETE 权限 ..... 1194
从补丁 202 开始，Amazon Redshift Serverless 在快照还原时保留零 ETL 和 S3 事件集成 . 1194
2026 年 12 月 31 日终止对 Amazon Redshift ODBC 1.x 驱动程序的支持 ..... 1195
标量 Python UDF 将在 2026 年 6 月 30 日之后终止支持 ..... 1195
2026 年 2 月 27 日之后的实体化视图（MV）自动 REFRESH 行为更改 ..... 1196
2026 年 2 月 16 日之后，Amazon Redshift 将不支持通过数据共享访问使用者信息的函数 . 1196
传输层安全性协议（TLS）最低版本变更自 2026 年 10 月 31 日起生效 ..... 1196
2025 年 10 月 30 日之后，Amazon Redshift 将不支持创建新的标量 Python UDF ..... 1197
最近的行为更改 ..... 1197
2025 年 8 月 26 日之后，Amazon Redshift 使用最新的 IANA 时区数据库 ..... 1198
Amazon Redshift Serverless RPU 更改将于 2025 年 8 月 15 日之后生效 ..... 1199
数据库审计日志记录更改在 2025 年 8 月 10 日之后生效 ..... 1199
无服务器工作组的虚拟私有云端点更改将于 2025 年 6 月 27 日之后生效 ..... 1199
查询监控更改于 2025 年 5 月 2 日之后生效 ..... 1199
在 2025 年 1 月 10 日之后生效的安全更改 ..... 1200
代码示例 ..... 1202
基本功能 ..... 1203
开始使用 Amazon Redshift ..... 1203
了解基本功能 ..... 1208
操作 ..... 1275
场景 ..... 1333
创建 Web 应用程序来跟踪 Amazon Redshift 数据 ..... 1333
Serverless 数据仓库入门 ..... 1334
预调配数据仓库集群入门 ..... 1351
文档历史记录 ..... 1363

2026 年 6 月 30 日之后，Amazon Redshift 不再支持使用 Python UDF。我们将分阶段开始执行。有关 Python 生命周期终止和迁移选项详情的更多信息，请参阅 2025 年 6 月 30 日发布的博客文章。

# 什么是 Amazon Redshift？

欢迎使用《Amazon Redshift 管理指南》。Amazon Redshift 是云中一种完全托管的 PB 级数据仓库服务。Amazon Redshift Serverless 让您可以访问和分析数据，而无需对预置数据仓库执行任何配置操作。系统将自动预置资源，数据仓库的容量会智能扩展，即使面对要求最为苛刻且不可预测的工作负载也能提供高速性能。数据仓库空闲时不会产生费用，您只需为实际使用的资源付费。您可以在 Amazon Redshift 查询编辑器 v2 或您最喜欢的商业智能（BI，Business Intelligence）工具中，直接加载数据并开始查询。在易于使用且无需承担管理任务的环境中，享受最佳性价比，使用熟悉的 SQL 功能。

无论数据集的大小如何，Amazon Redshift 都使用您目前所用的基于 SQL 的相同工具和业务情报应用程序，来提供快速的查询性能。

## 您是首次接触 Amazon Redshift 用户吗？

如果您是首次接触 Amazon Redshift 的用户，我们建议您先阅读以下部分：

- 服务亮点和定价 – 该产品详细信息页面提供 Amazon Redshift 价值主张、服务亮点和定价。
- Amazon Redshift Serverless 数据仓库入门 – 本主题引导您完成设置无服务器数据仓库、创建资源和查询示例数据的过程。
- Amazon Redshift 数据库开发人员指南 – 如果您是数据库开发人员，则本指南说明如何设计、构建、查询和维护构成数据仓库的数据库。

如果您希望手动管理 Amazon Redshift 资源，则可以创建预置集群来满足自己的数据查询需求。有关更多信息，请参阅 Amazon Redshift 集群。

作为应用程序开发人员，您可以使用 Amazon Redshift API 或 AWS 软件开发工具包（SDK）库以编程方式管理集群。如果您使用 Amazon Redshift API，则必须通过对发送到 API 的每个 HTTP 或 HTTPS 请求进行签名来执行身份验证。有关对请求签名的更多信息，请转到对 HTTP 请求进行签名。

有关 CLI、API 和 SDK 的信息，请转到以下链接：

- Amazon Redshift Serverless API 参考
- Amazon Redshift API 参考
- Amazon Redshift Data API 参考
- AWS CLI 命令引用
- 适用于 Amazon Web Services 的工具中的开发工具包参考。

# Amazon Redshift Serverless 功能概览

Amazon Redshift Serverless 也支持 Amazon Redshift 预调配数据仓库所支持的大多数功能。以下是它的一些关键功能。

| 功能 | 说明 |
| --- | --- |
| 快照 | 您可以将 Amazon Redshift Serverless 或预调配数据仓库的快照还原到 Amazon Redshift Serverless。有关更多信息，请参阅 快照和恢复点。 |
| 恢复点 | Amazon Redshift Serverless 每 30 分钟自动创建一个恢复点。这些恢复点将保留 24 个小时。在意外写入或删除之后，您可以使用它们进行还原。从恢复点还原时，Amazon Redshift Serverless 数据库中的所有数据都将还原到较早的时间点。如果需要将恢复点保留较长时间，也可以从恢复点创建快照。有关更多信息，请参阅 快照和恢复点。 |
| 基本 RPU 容量 | 您能够以 Redshift 处理单元 (RPU) 为单位设置基本容量。一个 RPU 提供 16 GB 内存。此设置使您能够控制正在使用的资源与工作负载成本之间的平衡。您可以增加此值以增加可用资源并提高查询性能，或者降低此值以限制您的支出。原定设置为 128 个 RPU。您还可以设置使用量限制（例如，每天使用的 RPU）来控制成本。有关更多信息，请参阅 Amazon Redshift Serverless 的计费。 |
| 数据共享的使用限制 | 您可以使用控制台或 API，限制从生产者区域向使用者区域传输的数据量。这些数据传输成本因 AWS 区域而不同，以 TB 为单位测量。有关数据共享的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的使用控制台实现数据共享入门。 |
| 用户定义的函数 (UDF) | 您可以在 Amazon Redshift Serverless 中运行用户定义的函数 (UDF)。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的创建用户定义的函数。 |
| 存储过程 | 您可以在 Amazon Redshift Serverless 中运行存储过程。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的创建存储过程。 |
| 实体化视图 | 您可以在 Amazon Redshift Serverless 中创建实体化视图。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的创建实体化视图。 |
| 空间函数 | 您可以在 Amazon Redshift Serverless 中运行空间函数。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的查询空间数据。 |

| 功能 | 说明 |
| --- | --- |
| 联合查询 | 您可以运行查询，以将数据与 Amazon Redshift Serverless 中的 Aurora DB 集群和 Amazon RDS 数据库连接起来。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的通过联合查询来查询数据。 |
| 数据湖查询 | 您可以运行查询，以将 Amazon S3 数据湖中的数据与 Amazon Redshift Serverless 连接起来。有关更多信息，请参阅《Amazon Redshift 管理指南》中的查询数据湖。 |
| HyperLogLog | 您可以在 Amazon Redshift Serverless 中运行 HyperLogLog 函数。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的使用 HyperlogLog 草图。 |
| 跨数据库查询数据 | 您可以通过 Amazon Redshift Serverless 跨数据库查询数据。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的跨数据库查询数据。 |
| 数据共享 | 您可以使用 Amazon Redshift Serverless 访问预调配数据仓库上的数据共享。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的跨集群共享数据。 |
| 半结构化数据查询 | 可以通过 Amazon Redshift Serverless 上的 SUPER 数据类型来摄取和存储半结构化数据。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的摄取和查询半结构化数据。 |
| 为资源添加标签 | 您可以使用 AWS CLI 或 Amazon Redshift Serverless API，通过与资源相关的元数据来给资源添加标签。有关更多信息，请参阅标记资源。 |
| 机器学习 | 您可以将 Amazon Redshift 机器学习与 Amazon Redshift Serverless 结合使用。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的使用机器学习。 |
| SQL 命令和函数 | 除了一些例外情况（例如 REBOOT_CLUSTER ），您可以将 Amazon Redshift SQL 命令和函数与 Amazon Redshift Serverless 结合使用。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 SQL 参考。 |
| CloudFormation 资源 | 使用 CloudFormation 模板，您可以部署和更新 Amazon Redshift Serverless 资源。这种集成意味着您可以花更少的时间管理资源，从而将精力集中在应用程序上。有关 Amazon Redshift Serverless 中的 CloudFormation 资源的更多信息，请参阅 Amazon Redshift Serverless 资源类型参考。 |

| 功能 | 说明 |
| --- | --- |
| CloudTrail 资源 | Amazon Redshift Serverless 与 AWS CloudTrail 集成，以提供在 Amazon Redshift Serverless 中执行的操作的记录。CloudTrail 将 Amazon Redshift Serverless 的所有 API 调用作为事件捕获。有关更多信息，请参阅 CloudTrail for Amazon Redshift Serverless。 |

# Amazon Redshift 预置集群概览

Amazon Redshift 服务管理数据仓库的所有设置、操作和扩展工作。这些任务包括：预置容量，监控和备份集群，以及向 Amazon Redshift 引擎应用修补程序和升级。

以下视频向您展示如何创建集群并使用 Amazon Redshift 查询编辑器 v2 查询数据。

## 集群管理

Amazon Redshift 集群是一组节点，其中包含一个领导节点以及一个或多个计算节点。所需计算节点的类型和数量取决于数据的大小、将运行的查询数以及所需的查询运行时性能。

### 创建和管理集群

根据数据仓库需要，开始时您可以使用一个小的单节点集群，然后随着您的需求变化轻松地扩展为更大的多节点集群。您可以在集群中添加或删除计算节点，而不会出现任何服务中断。有关更多信息，请参阅 Amazon Redshift 预置集群。

### 预留计算节点

如果您打算让集群保持运行一年或更长时间，则可以将计算节点保留一年或三年的时间，从而节省成本。与您按需预置计算节点时支付的小时费率相比，保留计算节点可大大地节省成本。有关更多信息，请参阅 预留节点。

### 创建集群快照

快照是集群的时间点备份。存在两种类型的快照：自动和手动。Amazon Redshift 通过使用加密的安全套接字层 (SSL) 连接，在 Amazon Simple Storage Service (Amazon S3) 内部存储这些快照。如果您需要从快照还原，Amazon Redshift 会创建一个新集群并从您指定的快照导入数据。有关 快照的更多信息，请参阅 Amazon Redshift 快照和备份。

# 集群访问和安全性

Amazon Redshift 中有几项与集群访问和安全相关的功能。这些功能可帮助您控制对集群的访问，定义连接规则，以及对数据和连接进行加密。这些功能是除 Amazon Redshift 中与数据库访问和安全相关的功能之外的功能。有关数据库安全的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的管理数据库安全。

## AWS 账户和 IAM 凭证

预设情况下，只有创建集群的 AWS 账户才能访问 Amazon Redshift 集群。该集群被锁定，这样其他任何人都不能访问它。在 AWS 账户内，您可以使用 AWS Identity and Access Management (IAM) 服务创建用户账户和管理这些账户的权限以控制集群操作。有关更多信息，请参阅 Amazon Redshift 中的安全性。有关管理 IAM 身份的更多信息，包括 IAM 角色的指南和最佳实践，请参阅 Amazon Redshift 中的 Identity and Access Management。

## 安全组

预设情况下，您创建的任何集群都对所有人关闭。IAM 凭证仅控制对 Amazon Redshift API 相关资源的访问：Amazon Redshift 控制台、命令行界面 (CLI)、API 和开发工具包。要能够通过 JDBC 或 ODBC 从 SQL 客户端工具访问集群，您可以使用安全组：

- 如果您使用 EC2-VPC 平台访问 Amazon Redshift 集群，则必须使用 VPC 安全组。我们建议您在 EC2-VPC 平台中启动集群。

  使用 EC2-Classic 启动集群后，无法将其移动到 VPC。不过，您可以使用 Amazon Redshift 控制台将 EC2-Classic 快照还原到 EC2-VPC 集群。有关更多信息，请参阅 从快照还原集群。
- 如果您使用 EC2-Classic 平台访问 Amazon Redshift 集群，则必须使用 Amazon Redshift 安全组。

在上述任一情况下，您可以向安全组中添加规则，以授予对特定 CIDR/IP 地址范围或 Amazon Elastic Compute Cloud (Amazon EC2) 安全组（如果 SQL 客户端运行在 Amazon EC2 实例上）的显式入站访问权限。有关更多信息，请参阅 Amazon Redshift 安全组。

除入站访问规则之外，您还可以创建数据库用户以提供凭证向集群自身内的数据库进行身份验证。有关更多信息，请参阅本主题中的数据库。

## 加密

当您预置集群时，可以选择对集群进行加密以提高安全性。启用加密时，Amazon Redshift 会将所有数据以加密格式存储在用户创建的表中。使用 AWS Key Management Service (AWS KMS) 来管理 Amazon Redshift 的加密密钥。

加密是集群的不可变属性。从加密集群切换到非加密集群的唯一方式是：卸载数据并将其重新加载到新集群。加密会应用于集群和所有备份。从加密快照还原集群时，新集群也会加密。

有关加密、密钥和硬件安全模块的更多信息，请参阅 Amazon Redshift 数据库加密。

### SSL 连接

您可以使用安全套接字层 (SSL) 加密对 SQL 客户端和集群之间的连接进行加密。有关更多信息，请参阅 配置连接的安全选项。

## 监控集群

Amazon Redshift 中有几项与监控相关的功能。您可以使用数据库审计日志记录来生成活动日志，配置事件和通知订阅来跟踪感兴趣的信息。使用 Amazon Redshift 和 Amazon CloudWatch 中的指标，了解集群和数据库的运行状况及性能。

### 数据库审核日志记录

您可以使用数据库审计日志记录功能来跟踪有关身份验证尝试次数、连接数、断开连接数、数据库用户定义更改以及数据库中运行的查询的信息。这些信息对 Amazon Redshift 中的安全和故障排除非常有用。日志存储在 Amazon S3 桶中。有关更多信息，请参阅 数据库审计日志记录。

### 事件和通知

Amazon Redshift 跟踪事件并在您的 AWS 账户中将事件的相关信息保留几周。对于每个事件，Amazon Redshift 会报告事件发生日期、描述、事件源（例如，集群、参数组或快照）和源 ID 等信息。您可以创建 Amazon Redshift 事件通知订阅以指定一组事件筛选器。当发生与筛选条件匹配的事件时，Amazon Redshift 将使用 Amazon Simple Notification Service 通知您发生了该事件。有关事件和通知的更多信息，请参阅 Amazon Redshift 事件。

### 性能

Amazon Redshift 提供性能指标和数据，以便您可以跟踪集群和数据库的运行状况及性能。Amazon Redshift 使用 Amazon CloudWatch 指标监控集群的物理方面，例如 CPU 使用率、延迟和吞吐量。Amazon Redshift 还提供查询和加载性能数据，以帮助您监控集群中的数据库活动。有关性能指标和监控的更多信息，请参阅监控 Amazon Redshift 集群性能。

## 数据库

当您预置集群时，Amazon Redshift 会创建一个数据库。这是您用于加载数据并对数据运行查询的数据库。您可以根据需要通过运行 SQL 命令来创建其他数据库。有关创建其他数据库的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的步骤 1：创建数据库。

当您预置集群时，可以指定一个管理员用户，此管理员用户可以访问在该集群内创建的所有数据库。此管理员用户是最初唯一可以访问数据库的超级用户，此用户也可以创建其它超级用户和用户。有关更多信息，请转至《Amazon Redshift 数据库开发人员指南》中的超级用户和用户。

Amazon Redshift 使用参数组定义集群中所有数据库的行为，例如，日期表示样式和浮点精度。如果您在预置集群时未指定参数组，则 Amazon Redshift 会将一个默认参数组与集群相关联。有关更多信息，请参阅 Amazon Redshift 参数组。

有关 Amazon Redshift 中的数据库的更多信息，请转至 Amazon Redshift 数据库开发人员指南。

# 将 Amazon Redshift Serverless 与 Amazon Redshift 预调配数据仓库进行比较

对于 Amazon Redshift Serverless 来说，一些概念和功能与 Amazon Redshift 预调配数据仓库的相应功能不同。例如，一个截然不同的比较是 Amazon Redshift Serverless 没有集群或节点的概念。下表介绍 Amazon Redshift Serverless 中的功能和行为，并解释了它们与预调配数据仓库中的等效功能有何不同。

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 工作组和命名空间 | 要在 Amazon Redshift Serverless 中隔离工作负载并管理不同的资源，您可以创建命名空间和工作组，以便分 | 命名空间是数据库对象和用户的集合。工作组是计算资源的集合。有关更多信息，请参阅 Amazon Redshift Serverles | 预调配集群是计算节点和领导节点的集合，您可以直接管理这些节点。有关更多信息，请参阅 Amazon Redshift 预置集群。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| | 别管理存储资源和计算资源。 | s 以了解 Amazon Redshift Serverless 的设计。 | |
| 节点类型 | 当您使用 Amazon Redshift Serverless 时，您不会像对待预调配的 Amazon Redshift 集群那样选择节点类型或指定节点计数。 | Amazon Redshift Serverless 会自动为您预置和管理容量。您可以选择指定基本数据仓库容量，以便为您的工作负载选择合适的性价比平衡。您还可以指定最大 RPU 小时数来设置成本控制，以确保成本是可预测的。有关更多信息，请参阅 Amazon Redshift Serverless 的计算容量。 | 您可以使用符合成本和性能规格的节点类型构建集群。有关更多信息，请参阅 Amazon Redshift 预置集群。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 工作负载管理和并发扩展 | Amazon Redshift 可以在高负载时段进行扩展。Amazon Redshift Serverless 还可以进行扩展，以应对间歇性的高负载时段。 | Amazon Redshift Serverless 自动、高效地管理资源，并根据工作负载在成本控制的阈值范围内进行扩缩。有关更多信息，请参阅计算容量的计费。 | 使用预调配数据仓库，您可以在集群上启用并发扩展以应对高负载时段。有关更多信息，请参阅并发扩展。 |
| 端口 | 用于连接的端口号。 | 使用 Amazon Redshift Serverless，您可以更改为 5431-5455 或 8191-8215 端口范围内的另一个端口。有关更多信息，请参阅 连接到 Amazon Redshift Serverless。 | 使用预调配数据仓库，您可以选择任何端口进行连接。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 调整大小 | 添加或删除计算资源以更好地应对工作负载。 | 调整大小在 Amazon Redshift Serverless 中不适用。但是，您可以根据价格和性能要求更改基本数据仓库 RPU 容量。有关更多信息，请参阅 Amazon Redshift Serverless 的计算容量。 | 对于预调配集群，您可以执行集群调整大小以添加节点或删除节点。有关更多信息，请参阅在 Amazon Redshift 中管理集群的概览。 |
| 暂停和恢复 | 在没有工作负载要运行时，可以暂停预调配集群，以节省成本。 | 使用 Amazon Redshift Serverless，您只需在查询运行时付费，因此无需暂停或恢复。有关更多信息，请参阅计算容量的计费。 | 您可以在不同时间根据对工作负载的评估，手动暂停和恢复集群。有关更多信息，请参阅在 Amazon Redshift 中管理集群的概览。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 查询外部数据 | 您可以在 Amazon S3 存储桶中查询各种格式（例如 JSON、Parquet 或 Iceberg）的数据。 | Amazon Redshift Serverless 包括一个在工作组的计算资源上运行的集成式数据湖查询引擎，并支持在一个 AWS 区域内或不同 AWS 区域之间查询 Amazon S3 数据。要调优数据湖查询性能，请调整工作组的基本 RPU 容量。 | 在 RA3 和 DC2 预置集群中，数据湖查询使用独立于集群的专用服务器在 Redshift Spectrum 上运行。Redshift Spectrum 要求 Amazon S3 存储桶与集群位于同一 AWS 区域。有关更多信息，请参阅使用 Amazon Redshift Spectrum 查询外部数据。<br><br>RG 预置集群包括集成式数据湖查询引擎。数据湖查询直接在集群自己的计算资源上运行，类似于 Redshift Serverless，并支持查询位于同一 AWS 区域或不同 AWS 区域中的 Amazon S3 数据。有关更多信息，请参阅查询数据湖。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 计算资源账单 | Amazon Redshift 与 Amazon Redshift Serverless 的账单是如何累积的。 | 使用 Amazon Redshift Serverless，您按秒支付以 RPU 小时为单位运行的工作负载，最低收费时间为 60 秒。这包括在 Amazon S3 中以开放文件格式访问数据的查询。有关更多信息，请参阅 计算容量的计费。 | 对于预调配集群，当集群未暂停时，将按秒计费。对于 RG 集群，数据湖查询包含在节点小时计算定价中，不单独收取 Spectrum 费用。对于 RA3 和 DC2 集群，数据湖查询会产生额外的 Redshift Spectrum 费用。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 维护时段 | 服务器维护的工作原理。 | 对于 Amazon Redshift Serverless，没有维护时段。可以无缝地处理更新。有关更多信息，请参阅什么是 Amazon Redshift Serverless？ | 对于预调配集群，您可以指定进行修补时的维护时段。（通常，当使用率较低时，您可以选择一个重复时间。） |
| 加密： | 您可以启用数据库加密。 | Amazon Redshift Serverless 始终使用 AWS KMS、AWS 托管式或客户托管式密钥进行加密。 | 可以使用 AWS KMS（AWS 托管式或客户托管式密钥）对预调配数据仓库中的数据进行加密，或进行解密。请参阅Amazon Redshift 数据库加密。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 存储计费 | 存储计费的工作原理。 | 适用于 Amazon Redshift Serverless。费率按每月 GB 计算。请参阅计算容量的计费。 | 对于具有 RG 或 RA3 节点的预置集群，存储与计算资源分开计费。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 用户管理 | 如何管理用户。 | 对于 Amazon Redshift Serverless，用户可以是 IAM 或 Redshift 用户。有关更多信息，请参阅 Amazon Redshift Serverless 中的 Identity and Access Management。<br><br>有关管理 IAM 身份的更多信息，包括 IAM 角色的最佳实践，请参阅 Amazon Redshift 中的 Identity and Access Management。 | 对于预置数据仓库，用户可以是 IAM 或 Redshift 用户。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的管理数据库安全。<br><br>有关管理 IAM 身份的更多信息，包括 IAM 角色的最佳实践，请参阅 Amazon Redshift 中的 Identity and Access Management。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| JDBC 和 ODBC 工具和兼容性 | 客户端连接的工作原理。 | Amazon Redshift Serverless 与任何符合 JDBC 或 ODBC 标准的工具或客户端应用程序兼容。有关驱动程序的更多信息，请参阅《Amazon Redshift 管理指南》中的配置连接。有关连接到 Amazon Redshift Serverless 的信息，请参阅连接到 Redshift Serverless。 | Amazon Redshift 预置集群与任何符合 JDBC 或 ODBC 标准的工具或客户端应用程序兼容。有关驱动程序的更多信息，请参阅《Amazon Redshift 管理指南》中的配置连接。有关连接到集群的信息，请参阅使用 SQL 客户端工具连接到 Amazon Redshift 数据仓库。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 登录时的凭证要求 | 如何处理凭证。 | 对于 Amazon Redshift Serverless，您不必在每个实例中输入凭证。有关更多信息，请参阅 连接到 Amazon Redshift Serverless。 | 访问 Amazon Redshift 要求与 IAM 角色关联的用户提供登录凭证。IAM 角色具有为预调配数据仓库附加的特定权限。经过身份验证后，用户可以直接连接到数据库、Redshift 控制台和查询编辑器 v2。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 数据 API | 您可以访问来自 Web 服务和其他应用程序的数据。 | Amazon Redshift Serverless 支持 Amazon Redshift 数据 API。对于 Amazon Redshift Serverless，您使用 `workgroup-name` 参数，而不是 `cluster-identity` 参数。有关调用数据 API 的更多信息，请参阅使用 Amazon Redshift 数据 API。 | Amazon Redshift 预置集群支持 Amazon Redshift 数据 API。对于 Amazon Redshift 集群，您应使用 `cluster-identity` 参数，而不是 `workgroup-name` 参数。有关调用数据 API 的更多信息，请参阅使用 Amazon Redshift 数据 API。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 快照 | 提供时间点恢复。 | Amazon Redshift Serverless 支持快照和恢复点。有关命名空间的快照和恢复点的更多信息，请参阅快照和恢复点。 | 预调配集群支持快照。有关更多信息，请参阅使用控制台管理快照。 |
| 数据共享 | 提供在同一个账户或不同账户中的数据库之间共享数据的功能。 | Amazon Redshift Serverless 支持预置数据仓库所能实现的所有数据共享功能，还支持 Amazon Redshift Serverless 与预置数据仓库、工具或客户端应用程序之间的数据共享。 | 预调配集群支持跨数据库、跨账户、跨区域和 AWS Data Exchange 数据共享。有关更多信息，请参阅在 Amazon Redshift 中跨集群共享数据。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 跟踪 | 提供软件更新的时间表。 | Amazon Redshift Serverless 没有跟踪的概念。版本和更新由服务处理。有关 Amazon Redshift Serverless 的设计的更多信息，请参阅快照和恢复点。 | 预调配集群支持在当前版本跟踪和早先版本跟踪之间切换。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 系统表和视图 | 提供一种监控您的资源和系统元数据的方法。 | Amazon Redshift Serverless 支持新的系统表和视图。有关系统表的更多信息，请参阅使用 Amazon Redshift Serverless 监控查询和工作负载。有关如何将查询从使用较旧的预调配系统表和视图迁移到新视图的信息，请参阅迁移到 SYS 监控视图。 | 预调配数据仓库支持一组现有的系统表和视图，它们用于进行监控和其他需要系统元数据的任务。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 参数组 | 这是一组适用于在集群中创建的所有数据库的参数。这些参数可用于配置数据库设置，例如查询超时和日期样式。 | Amazon Redshift Serverless 没有参数组的概念。 | 预调配数据仓库支持参数组。有关预置集群的参数组的更多信息，请参阅 Amazon Redshift 参数组。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 查询监控 | 提供基于时间的查询运行视图。 | Amazon Redshift Serverless 中的查询监控要求用户连接到数据库以使用系统表。因此，查询监控和系统表是同步的。对 Amazon Redshift Serverless 中系统表的查询使用映射到 IAM 用户的数据库用户，以便使用查询监控。有关监控查询的更多信息，请参阅使用 Amazon Redshift Serverless 监控查询和工作负载。 | 预调配集群中的查询监控不会显示系统表中的所有数据。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 审计日志记录 | 提供有关数据库中的连接和用户活动的信息。 | 使用 Amazon Redshift Serverless，CloudWatch 是审计日志的目标。Amazon Redshift Serverless 不支持基于 Amazon S3 的审计日志交付。有关更多信息，请参阅 Amazon Redshift Serverless 的审计日志记录。 | 对于预调配集群，基于 Amazon S3 的审计日志交付一直是常态。现在，向 CloudWatch 交付审计日志的范围也扩展到包括预调配数据仓库。 |

| 功能 | 说明 | 无服务器 | 已预置 |
| --- | --- | --- | --- |
| 事件通知 | Amazon EventBridge 是一种无服务器事件总线服务，让您可以轻松地将应用程序与来自各种源的事件数据相连接。 | Amazon Redshift Serverless 使用 Amazon EventBridge 管理事件通知，以便随时了解数据仓库中的更改。有关更多信息，请参阅 使用 Amazon EventBridge 的 Amazon Redshift Serverless 事件通知。 | 对于预调配集群，可以使用 Amazon Redshift 控制台创建事件订阅来管理事件通知。有关更多信息，请参阅 创建事件通知订阅。 |
| 游标约束 | Amazon Redshift 会对所有游标结果集的大小强制实施约束。 | Amazon Redshift Serverless 的游标最大结果集总大小为 150000 MB。 | 对于预置集群，游标的最大结果集总大小取决于集群类型。有关更多信息，请参阅游标约束。 |

# 使用预置集群的 Amazon Redshift 管理界面

> **Note**
>
> 本主题重点介绍预置集群的 Amazon Redshift 管理界面。Amazon Redshift Serverless 和 Amazon Redshift 数据 API 也有类似的管理界面。

Amazon Redshift 支持多种可用于创建、管理和删除 Amazon Redshift 集群的管理界面，其中包括 AWS 开发工具包、 AWS Command Line Interface (AWS CLI) 和 Amazon Redshift 管理 API。

The Amazon Redshift API – 您可以通过提交请求来调用此 Amazon Redshift 管理 API。请求是 HTTP 或 HTTPS 请求，需要使用 HTTP 动词 `GET` 或 `POST` 以及一个名为 `Action` 的参数。调用 Amazon Redshift API 是访问 Amazon Redshift 服务的最直接方式。但是，此调用需要您的应用程序处理低级别的详细信息，例如进行错误处理以及生成哈希值以签署请求。

- 有关构建和签署查询 Amazon Redshift API 请求的信息，请参阅对 HTTP 请求进行签名。
- 有关 Amazon Redshift API 操作和 Amazon Redshift 的数据类型的信息，请参阅 Amazon Redshift API 参考。

AWS 开发工具包 – 您可以使用 AWS 开发工具包执行与 Amazon Redshift 集群相关的操作。一些开发工具包库包含底层 Amazon Redshift API。他们将 API 功能集成到特定编程语言并处理许多低级别详细信息，如计算签名、处理请求重试和进行错误处理。调用开发工具包库中的包装函数可极大地简化编写用于管理 Amazon Redshift 集群的应用程序的流程。

- Amazon Redshift 受适用于 Java、.NET、PHP、Python、Ruby 和 Node.js 的 AWS 开发工具包的支持。Amazon Redshift 的包装函数记录在每个开发工具包的参考手册中。有关 AWS 开发工具包及其文档链接的列表，请参阅适用于 Amazon Web Services 的工具。
- 本指南提供通过 Java 开发工具包使用 Amazon Redshift 的示例。有关更多通用 AWS SDK 代码示例，请参阅使用 AWS SDK 的 Amazon Redshift 代码示例。

AWS CLI – CLI 提供一组可用于从 Windows、Mac 和 Linux 计算机管理 AWS 服务的命令行工具。AWS CLI 包括基于 Amazon Redshift 查询 API 操作的命令。

- 有关安装和设置 Amazon Redshift CLI 的信息，请参阅设置 Amazon Redshift CLI。
- 有关 Amazon Redshift CLI 命令的参考资料，请参阅 AWS CLI 参考中的 Amazon Redshift。

# 将此服务与 AWS SDK 结合使用

AWS 软件开发工具包（SDK）适用于许多常用编程语言。每个软件开发工具包都提供 API、代码示例和文档，使开发人员能够更轻松地以其首选语言构建应用程序。

| SDK 文档 | 代码示例 |
| --- | --- |
| 适用于 C++ 的 AWS SDK | 适用于 C++ 的 AWS SDK 代码示例 |
| AWS CLI | AWS CLI 代码示例 |
| 适用于 Go 的 AWS SDK | 适用于 Go 的 AWS SDK 代码示例 |
| 适用于 Java 的 AWS SDK | 适用于 Java 的 AWS SDK 代码示例 |
| 适用于 JavaScript 的 AWS SDK | 适用于 JavaScript 的 AWS SDK 代码示例 |
| 适用于 Kotlin 的 AWS SDK | 适用于 Kotlin 的 AWS SDK 代码示例 |
| 适用于 .NET 的 AWS SDK | 适用于 .NET 的 AWS SDK 代码示例 |
| 适用于 PHP 的 AWS SDK | 适用于 PHP 的 AWS SDK 代码示例 |
| AWS Tools for PowerShell | AWS Tools for PowerShell 代码示例 |
| 适用于 Python 的 AWS SDK v4 | 适用于 Python 的 AWS SDK v4 代码示例 |
| 适用于 Python (Boto3) 的 AWS SDK | 适用于 Python (Boto3) 的 AWS SDK 代码示例 |
| 适用于 Ruby 的 AWS SDK | 适用于 Ruby 的 AWS SDK 代码示例 |
| 适用于 Rust 的 AWS SDK | 适用于 Rust 的 AWS SDK 代码示例 |
| 适用于 SAP ABAP 的 AWS SDK | 适用于 SAP ABAP 的 AWS SDK 代码示例 |
| 适用于 Swift 的 AWS SDK | 适用于 Swift 的 AWS SDK 代码示例 |

**示例可用性**

找不到所需的内容？ 通过使用此页面底部的提供反馈链接请求代码示例。

# 对 HTTP 请求进行签名

Amazon Redshift 要求您发送到管理 API 的每个请求都必须使用签名进行身份验证。本主题介绍如何为请求签名。

如果您使用的是 AWS 开发工具包 (SDK) 之一或 AWS Command Line Interface，将自动为请求签名，因此您可以跳过本节。有关如何使用 AWS 开发工具包的更多信息，请参阅使用预置集群的 Amazon Redshift 管理界面。有关使用 Amazon Redshift 命令行界面的更多信息，请转到 Amazon Redshift 命令行参考。

要为请求签名，可以使用加密哈希函数计算数字签名。加密哈希一种是根据输入内容返回唯一哈希值的函数。对哈希函数的输入内容包括您的请求文本和您从临时凭证获得的秘密访问密钥。哈希函数返回哈希值，您将该值包含在请求中，作为签名。该签名是您的请求的 `Authorization` 标头的一部分。

> **Note**
>
> 如果用户需要在 AWS 管理控制台之外与 AWS 交互，则需要编程式访问权限。授予编程式访问权限的方法取决于访问 AWS 的用户类型。
> 要向用户授予编程式访问权限，请选择以下选项之一。

| 哪个用户需要编程式访问权限？ | 目的 | 方式 |
| --- | --- | --- |
| IAM | （推荐）使用控制台凭证作为临时凭证来签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照您希望使用的界面的说明进行操作。<br>• 对于 AWS CLI，请参阅《AWS Command Line Interface用户指南》中的登录 AWS 以进行本地开发。<br>• 对于 AWS SDK，请参阅《AWS SDK 和工具参考指南》中的登录 AWS 以进行本地开发。 |

| 哪个用户需要编程式访问权限？ | 目的 | 方式 |
| --- | --- | --- |
| 人力身份<br><br>（在 IAM Identity Center 中管理的用户） | 使用临时凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照您希望使用的界面的说明进行操作。<br><br>• 有关 AWS CLI 的更多信息，请参阅《AWS Command Line Interface 用户指南》中的配置 AWS CLI 以使用 AWS IAM Identity Center。<br>• 有关 AWS SDK、工具和 AWS API 的更多信息，请参阅《AWS SDK 和工具参考指南》中的 IAM Identity Center 身份验证。 |
| IAM | 使用临时凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照《IAM 用户指南》中将临时凭证用于 AWS 资源中的说明进行操作。 |

| 哪个用户需要编程式访问权限？ | 目的 | 方式 |
| --- | --- | --- |
| IAM | （不推荐使用）<br>使用长期凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照您希望使用的界面的说明进行操作。<br>• 有关 AWS CLI 的更多信息，请参阅《AWS Command Line Interface 用户指南》中的使用 IAM 用户凭证进行身份验证。<br>• 有关 AWS SDK 和工具的更多信息，请参阅《AWS SDK 和工具参考指南》中的使用长期凭证进行身份验证。<br>• 有关 AWS API 的更多信息，请参阅《IAM 用户指南》中的管理 IAM 用户的访问密钥。 |

Amazon Redshift 收到您的请求后，它将使用您在为请求签名时使用的相同哈希函数和输入来重新计算签名。如果得出的签名与请求中的签名相匹配，Amazon Redshift 会处理请求；否则请求将被拒绝。

Amazon Redshift 支持使用 AWS 签名版本 4 进行身份验证。计算签名的流程包含三个任务。这些任务在下文的示例中有所说明。

- 任务 1：创建规范请求

  将您的 HTTP 请求重新排列为规范格式。必须使用规范格式，因为 Amazon Redshift 会使用同一规范格式计算要与您发送的签名进行比较的签名。

- 任务 2：创建要签名的字符串

  创建一个字符串，将该字符串用作您的加密哈希函数输入值中的一项。该字符串称为待签字符串，是哈希算法名称、请求日期、凭证范围字符串以及来自上一任务的规范化请求的结合。凭证范围字符串本身是日期、区域和服务信息的结合。

- 任务 3：计算签名

  使用加密哈希函数为您的请求计算签名，该函数接受两种输入字符串：待签名字符串和派生密钥。派生密钥的计算方法是，以您的秘密访问密钥开头，并使用凭证范围字符串来创建一系列基于哈希的消息身份验证代码 (HMAC-SHA256)。

## 示例签名计算

以下示例将为您详细介绍为 CreateCluster 请求创建签名的过程。您可以使用该示例作为参考，检查您自己的签名计算方法。其他参考计算包含在《IAM 用户指南》的“请求签名示例”部分中。

您可以使用 GET 或 POST 请求将请求发送到 Amazon Redshift。这两者之间的区别是对于 GET 请求，您的参数作为查询字符串参数发送。对于 POST 请求，参数包含在请求正文中。以下示例显示的是 POST 请求。

示例假定以下各项：

- 请求的时间戳为 `Fri, 07 Dec 2012 00:00:00 GMT`。
- 终端节点为美国东部（弗吉尼亚北部）地区 (`us-east-1`)。

一般的请求语法为：

```
https://redshift.us-east-1.amazonaws.com/
   ?Action=CreateCluster
   &ClusterIdentifier=examplecluster
   &MasterUsername=masteruser
   &MasterUserPassword=12345678Aa
   &NumberOfNode=2
   &NodeType=dc2.large
   &Version=2012-12-01
   &x-amz-algorithm=AWS4-HMAC-SHA256
   &x-amz-credential=AKIAIOSFODNN7EXAMPLE/20121207/us-east-1/redshift/aws4_request
   &x-amz-date=20121207T000000Z
   &x-amz-signedheaders=content-type;host;x-amz-date
```

为任务 1：创建规范请求计算的规范请求格式为：

```
POST
/
```

```
content-type:application/x-www-form-urlencoded; charset=utf-8
host:redshift.us-east-1.amazonaws.com
x-amz-date:20121207T000000Z

content-type;host;x-amz-date
55141b5d2aff6042ccd9d2af808fdf95ac78255e25b823d2dbd720226de1625d
```

规范请求的最后一行是请求正文的哈希值。因为没有针对此 API 的查询参数，所以规范请求的第三行是空的。

任务 2：创建待签字符串的待签字符串为：

```
AWS4-HMAC-SHA256
20121207T000000Z
20121207/us-east-1/redshift/aws4_request
06b6bef4f4f060a5558b60c627cc6c5b5b5a959b9902b5ac2187be80cbac0714
```

待签字符串的第一行是算法，第二行是时间戳，第三行是凭证范围，最后一行是来自任务 1：创建规范请求的规范请求的哈希。要在凭证范围中使用的服务名称为 `redshift`。

对于任务 3：计算签名，派生密钥可以表示为：

```
derived key = HMAC(HMAC(HMAC(HMAC("AWS4" + YourSecretAccessKey,"20121207"),"us-
east-1"),"redshift"),"aws4_request")
```

派生密钥是通过系列哈希函数计算的。从上面公式中最里面的 HMAC 语句开始，将短语 **AWS4** 与您的秘密访问密钥连接，并使用它作为键来对数据“us-east-1”进行哈希计算。该哈希计算的结果将成为下一个哈希函数的键。

计算派生密钥后，在接受两个输入字符串、待签字符串和派生密钥的哈希函数中使用它。例如，如果您使用秘密访问密钥 `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY` 和前文提到的代签字符串，那么计算出的签名如下所示：

```
9a6b557aa9f38dea83d9215d8f0eae54100877f3e0735d38498d7ae489117920
```

最终步骤是构造 `Authorization` 标头。对于示例访问密钥 `AKIAIOSFODNN7EXAMPLE`，标头（为了便于阅读，添加了换行符）为：

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20121207/us-east-1/
redshift/aws4_request,
```

```
SignedHeaders=content-type;host;x-amz-date,
Signature=9a6b557aa9f38dea83d9215d8f0eae54100877f3e0735d38498d7ae489117920
```

# 设置 Amazon Redshift CLI

本部分介绍如何设置和运行用于管理 Amazon Redshift 的 AWS CLI 命令行工具。Amazon Redshift 命令行工具在 AWS Command Line Interface (AWS CLI) 上运行，而后者使用 Python (https://www.python.org/)。AWS CLI 可在支持 Python 的所有操作系统上运行。

## 安装 AWS Command Line Interface

要开始使用 Amazon Redshift 命令行工具，首先要设置 AWS CLI，然后添加用于定义 Amazon Redshift CLI 选项的配置文件。

如果您已为其他 AWS CLI 服务安装并配置 AWS，则可跳过该程序。

安装 AWS Command Line Interface

1. 请转到安装或更新到最新版本的 AWS CLI，然后按照说明安装 AWS CLI。

   要进行 CLI 访问，您需要访问密钥 ID 和秘密访问密钥。如果可能，请使用临时凭证代替长期访问密钥。临时凭证包括访问密钥 ID、秘密访问密钥，以及一个指示凭证何时到期的安全令牌。有关更多信息，请参阅《IAM 用户指南》中的将临时凭证用于 AWS 资源。
2. 创建包含访问密钥、默认区域和命令输出格式等配置信息的文件。然后，设置 `AWS_CONFIG_FILE` 环境变量以引用该文件。有关详细说明，请转到 AWS Command Line Interface 用户指南中的配置 AWS 命令行界面。
3. 运行一个测试命令以确认 AWS CLI 界面可正常工作。例如，以下命令应该显示有关 AWS CLI 的帮助信息：

   ```
   aws help
   ```

   以下命令应该显示有关 Amazon Redshift 的帮助信息：

   ```
   aws redshift help
   ```

有关 Amazon Redshift CLI 命令的参考资料，请转至 AWS CLI 参考中的 Amazon Redshift。

# Amazon Redshift Serverless

Amazon Redshift Serverless 可让您方便地运行和扩展分析，而无需预置和管理本地数据仓库。借助 Amazon Redshift Serverless，数据分析人员、开发人员和数据科学家现在通过将数据加载到云中的数据仓库并从其中查询记录，就可以使用 Amazon Redshift 在几秒钟内从数据中获取见解。Amazon Redshift 会自动预置和扩展数据仓库容量，以便为要求苛刻且不可预测的工作负载提供快速性能。您仅需为实际使用的容量付费。您无需更改现有分析和商业智能应用程序，即可受益于这种简单性。

有关 Amazon Redshift Serverless SLA 的更多信息，请参阅 Amazon Redshift 服务水平协议。

## 什么是 Amazon Redshift Serverless？

Amazon Redshift Serverless 会自动预置数据仓库容量并智能扩展基础资源。Amazon Redshift Serverless 可在几秒钟内调整容量，以便为要求最苛刻和急剧波动的工作负载提供始终如一的高性能和简化的操作。

使用 Amazon Redshift Serverless，您可从以下功能中受益：

- 无需设置、优化和管理 Amazon Redshift 预置集群即可访问和分析数据。
- 使用卓越的 Amazon Redshift SQL 功能、行业领先的性能和数据湖集成，以跨数据仓库、数据湖和操作数据来源进行无缝查询。
- 通过智能和自动扩缩，即使是最苛刻和急剧波动的工作负载，也能提供始终如一的高性能和简化操作。
- 使用工作组和命名空间，通过精细的成本控制来组织计算资源和数据。
- 仅在使用数据仓库时付费。

借助 Amazon Redshift Serverless，您可以使用控制台界面访问无服务器数据仓库或 API 以构建应用程序。通过数据仓库，您可访问 Amazon Redshift 托管存储和 Amazon S3 数据湖。

本视频向您展示 Amazon Redshift Serverless 如何轻松运行和扩展分析，而无需管理数据仓库基础设施：

### Amazon Redshift Serverless 控制台

要了解如何开始使用 Amazon Redshift Serverless 控制台，请观看以下视频：Amazon Redshift Serverless 入门。

### Serverless 控制面板

在无服务器控制面板页面中，您可查看资源摘要和使用情况图表。

- 命名空间概览 – 本部分显示命名空间内的快照和数据共享的数量。
- 工作组 – 本部分显示 Amazon Redshift Serverless 中的所有工作组。
- 查询指标 – 本部分显示过去一小时的查询活动。
- 已使用的 RPU 容量 – 本部分显示过去一小时内使用的容量。
- 免费试用 – 本部分显示您 AWS 账户中的剩余免费试用积分。这涵盖了同一账户下所有对 Amazon Redshift Serverless 资源和操作的使用，包括快照、存储、工作组等。
- 警报 – 本节显示您在 Amazon Redshift Serverless 中配置的警报。

### 数据备份

您可以在数据备份选项卡上使用以下选项：

- 快照 – 您可以创建、删除和管理 Amazon Redshift Serverless 数据的快照。默认的保留期为 `indefinitely`，但您可以将保留期配置为 1 到 3653 天之间的任意值。您可以授权 AWS 账户从快照还原命名空间。
- 恢复点 – 显示自动创建的恢复点，以便可以还原过去 24 小时内的意外写入或删除。要恢复数据，您可以将恢复点还原到任何可用的命名空间。如果想要将恢复点保留更长时间段，则可以从恢复点创建快照。原定设置的保留期为 `indefinitely`，但您可以将保留期配置为 1 到 3653 天之间的任意值。

### 数据访问

您可在数据访问选项卡上使用以下选项：

- 网络和安全性设置 – 您可以查看与 VPC 相关的值，AWS KMS 加密值和审计日志记录值。您只能更新审计日志记录。
- AWS KMS key — 用于加密 Amazon Redshift Serverless 中资源的 AWS KMS key。
- 权限 – 您可以管理 IAM 角色，Amazon Redshift Serverless 可担任此角色以代表您来使用资源。有关更多信息，请参阅 Amazon Redshift Serverless 中的 Identity and Access Management。
- Redshift 托管式 VPC 端点 – 您可从另一个 VPC 或子网访问 Amazon Redshift Serverless 实例。有关更多信息，请参阅 从其他 VPC 端点连接到 Amazon Redshift Serverless。

限制

在限制选项卡上，您可以使用以下选项：

- 以 Redshift 处理单元 (RPU) 计算的基本容量设置 – 您可以设置用于处理工作负载的基本容量。要改善查询性能，请增加 RPU 值。
- 使用限制 – 在启动操作之前的一段时间内，Amazon Redshift Serverless 实例可以使用的最大计算资源。您可限制 Amazon Redshift Serverless 用于运行工作负载的资源量。使用情况以 Redshift 处理单元 (RPU) 小时为单位衡量。RPU 小时是一小时内使用的 RPU 数量。您可以确定在达到所设置限制时执行的操作，如下所示：
  - 发送提示。
  - 将条目记录到系统表中。
  - 禁用用户查询。

  您最多可以设置四个限制。
- 查询限制 – 您可以添加限制来监控性能和限制。有关查询监控限制的更多信息，请参阅 WLM 查询监控规则。

有关更多信息，请参阅 Amazon Redshift Serverless 的计算容量。

数据共享

在数据共享选项卡上，可以使用以下选项：

- 在我的命名空间中创建的数据共享设置 – 您可以创建数据共享并与其它命名空间和 AWS 账户 共享。
- 来自其他命名空间和 AWS 账户 的数据共享 – 您可以从来自其他命名空间和 AWS 账户的数据共享创建数据库。

有关数据共享的更多信息，请参阅Amazon Redshift Serverless 中的数据共享。

查询和数据库监控

在查询和数据库监控页面上，您可以查看您的查询历史记录和数据库性能图表。

在查询历史记录选项卡上，您会看到以下图表（您可以在查询列表和资源指标之间进行选择）：

- 查询运行时间 – 此图表显示哪些查询在同一时间范围内运行。在图表中选择栏以查看更多查询执行详细信息。

- 查询和加载 – 本部分列出了按 查询 ID 进行的查询和加载。
- 已使用的 RPU 容量 – 此图表显示了以 Redshift 处理单元 (RPU) 表示的总容量。
- 数据库连接数 – 此图表显示了活动的数据库连接数。

### 数据库性能

在数据库性能选项卡上，您会看到以下图表：

- 每秒完成的查询数 – 此图表显示了每秒完成的平均查询数。
- 查询持续时间 – 此图表显示了完成查询的平均时间量。
- 数据库连接数 – 此图表显示了活动的数据库连接数。
- 正在运行的查询数 – 此图表显示了在给定时间内正在运行的查询总数。
- 排队的查询 – 此图表显示了在给定时间排队的查询总数。
- 查询运行时间细分 – 此图表显示了查询类型运行的查询所花费的总时间。

### 资源监控

在资源监控页面上，您可以查看已消耗资源的图表。您可以根据几个方面来筛选数据。

- 筛选指标 – 您可以使用指标筛选条件为特定工作组选择筛选条件，也可以选择时间范围和时间间隔。
- 已使用的 RPU 容量 – 此图表显示了 Redshift 处理单元 (RPU) 的总容量。
- 计算使用量 – 此图表显示在所选时间范围内按时间段划分的 RPU 小时数使用量。对于少于 6 小时的时间范围，RPU 小时数将显示为精确的时间。对于 6 小时或更长的时间范围，RPU 小时数显示为平均值。
- 自动优化的额外计算计费秒数：此图表显示在选定的时间范围内针对自动数据库优化计费的 RPU 秒数。当 Amazon Redshift 使用额外的计算资源来运行自动优化时，将向您收取自动优化的费用。有关更多信息，请参阅分配额外的计算资源来执行自动数据库优化。

### 数据共享

在数据共享页面上，您可以管理在我的账户和从其它账户的数据共享。有关数据共享的更多信息，请参阅Amazon Redshift Serverless 中的数据共享。

AWS Glue Data Catalog

在 AWS Glue Data Catalog 选项卡中，您可以查看向 AWS Glue Data Catalog 注册命名空间的注册状态。此选项卡仅在您开始注册过程后才会出现。有关向 AWS Glue Data Catalog 注册命名空间的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Apache Iceberg compatibility for Amazon Redshift。

## 使用 Amazon Redshift Serverless 时的注意事项

有关提供了 Amazon Redshift Serverless 的 AWS 区域 的列表，请参阅《Amazon Web Services 一般参考》中针对 Redshift Serverless API 列出的端点。

Amazon Redshift Serverless 使用的某些资源受限于配额。有关更多信息，请参阅 Amazon Redshift Serverless 对象的配额。

当您 DECLARE（声明）游标时，将在 DECLARE 中指定 Amazon Redshift Serverless 的结果集大小规范。Amazon Redshift Serverless 的游标最大结果集总大小为 150000 MB。

在线修补：Amazon Redshift Serverless 无需传统维护窗口即可实现自动软件更新。当有新的更新可用时，系统会在更新发布后的 14 天内，在空闲时段应用该更新。更新过程通常需要最多 15 分钟。如果在这 14 天内未出现 15 分钟的空闲时段，则 Serverless 端点可能会出现短暂不可用的情况。在此期间，应用程序可能无法与端点连接。您可以在“Amazon Redshift 的集群版本”文档中查看 Redshift 补丁版本。有关 Amazon Redshift Serverless SLA 的更多信息，请参阅 Amazon Redshift 服务水平协议。

跟踪：当 Amazon Redshift 发布新的工作组版本时，您的工作组将自动更新。您可以控制工作组是更新到最新版本还是先前版本。有关跟踪的信息，请参阅 Amazon Redshift 预置集群和无服务器工作组的跟踪。

可用区 ID：在配置 Amazon Redshift Serverless 实例时，打开其他注意事项，并确保子网中提供的子网 ID 至少包含两个受支持的可用区 ID。

- 对于没有增强型 VPC 路由（EVR）的工作组，您需要两个可用区（AZ）。
- 对于具有 EVR 的工作组，您需要三个可用区。

要查看子网到可用区 ID 映射，请转到 VPC 控制台并选择子网，以通过可用区 ID 来查看子网 ID 列表。验证您的子网已映射到受支持的可用区 ID。要创建子网，请参阅《Amazon VPC 用户指南》中的在 VPC 中创建子网。

两个子网（无 EVR）：必须至少有两个子网，并且这些子网必须跨两个可用区。

三个子网（仅 EVR）：使用 EVR 时，必须至少有三个子网，并且这些子网必须跨三个或更多可用区。

可用 IP 地址要求：在未启用增强型 VPC 路由（EVR）的情况下使用 Redshift Serverless 时，每个子网中必须至少有三个空闲 IP 地址可用。这是服务正常运行的要求。

更新 Redshift Serverless 部署的 RPU 时，每个子网中必须至少有三个空闲 IP 地址可用，才能满足服务的运行要求。

有关在 Amazon VPC 中分配 IP 地址和了解 IP 寻址的更多信息，请参阅《Amazon VPC 用户指南》中的为 VPC 和子网分配 IP 地址。

Without EVR

如果您不使用增强型 VPC 路由，则无论基本 RPU 的大小（4 到 1024 个 RPU）或一个或多个工作组的 RPU 使用情况如何，对于每个子网都必须至少有三个空闲的 IP 地址。需要 3 个 IP 地址也适用于启用了 AI 驱动的扩展和优化功能的工作组。

With Enhanced VPC Routing (EVR)

如果您在 Redshift Serverless 中使用增强型 VPC 路由，则创建工作组时所需的最少 IP 地址数量如下：

| Redshift 处理单元 (RPU) | 所需的可用 IP 地址 | 最小 CIDR 大小 |
|---|---|---|
| 4 | 9 | /27 |
| 8 | 9 | /27 |
| 16 | 13 | /27 |
| 32 | 13 | /27 |
| 64 | 21 | /27 |
| 128 | 37 | /26 |
| 256 | 69 | /25 |
| 512 | 133 | /24 |
| 1024 | 261 | /23 |

使用 EVR，在更新工作组以使用更多 RPU 时，您也需要空闲 IP 地址。更新工作组的子网时所需的可用 IP 地址数如下：

| Redshift 处理单元 (RPU) | 更新的 Redshift 处理单元 (RPU) | 所需的可用 IP 地址 |
|---|---|---|
| 4 | 8 | 10 |
| 8 | 16 | 10 |
| 16 | 32 | 13 |
| 32 | 64 | 16 |
| 64 | 128 | 28 |
| 128 | 256 | 52 |
| 256 | 512 | 100 |
| 512 | 1024 | 197 |

> **Note**
>
> 以下 AWS 区域中仅提供 1024 的最大基本 RPU 容量：
>
> - 美国东部（弗吉尼亚州北部）
> - 美国东部（俄亥俄州）
> - 美国西部（俄勒冈州）
> - 欧洲地区（爱尔兰）
> - 欧洲地区（伦敦）

有关分配 IP 地址的更多信息，请参阅《Amazon VPC 用户指南》中的 IP 地址。

迁移后的存储空间 – 如果将小型 Amazon Redshift 预调配集群迁移到 Amazon Redshift Serverless，则在迁移后，您可能会发现分配的存储空间增大。这是优化的存储空间分配的结果，从而产生了预分配的存储空间。随着 Amazon Redshift Serverless 中数据的增长，此空间将在一段时间内使用。

Amazon Redshift Serverless 和 Amazon Redshift 预调配集群之间的数据共享 – 在 Amazon Redshift Serverless 作为创建者且预调配集群作为使用者的数据共享中，预调配集群必须具有高于 1.0.38214 的集群版本。如果您使用的集群版本低于此值，则运行查询时会出错。您可以在 Amazon Redshift 控制台上的维护选项卡中查看集群版本。您也可以运行 `SELECT version();`。

最大查询执行时间 – 执行查询所用的时间（以秒为单位）。执行时间不包括在队列中等待的时间。如果查询超出设置的执行时间，Amazon Redshift Serverless 将停止查询。有效值为 0–86,399。

迁移带有交错排序键的表 – 当将 Amazon Redshift 预调配的集群迁移到 Amazon Redshift Serverless 时，Redshift 将具有交错排序键和 DISTSTYLE KEY 的表转换为复合排序键。DISTSTYLE 不会变化。有关分配方式的更多信息，请参阅《Amazon Redshift 开发人员指南》中的使用数据分配方式。有关排序键的更多信息，请参阅使用排序键。

VPC 共享 – 您可以在共享 VPC 中创建 Amazon Redshift Serverless 工作组。在这样做时，建议您不要删除资源共享，因为这可能会导致工作组不可用。

IPv6 支持：

Amazon Redshift Serverless 支持在 AWS Virtual Private Cloud（VPC）中使用 IPv4 和 IPv6 地址（双堆栈）或仅 IPv4 配置来配置您的 Amazon Redshift 工作组。您可以在创建新的 Amazon Redshift Serverless 工作组时启用 IPv6 支持，或者修改现有工作组以支持 IPv6 寻址。借助此功能，您可以在支持 IPv6 的 VPC 子网中部署 Amazon Redshift Serverless 数据仓库，并配置网络设置以支持应用程序不断扩展的地址空间需求。现在，您的应用程序可以使用 IPv4 或 IPv6 协议与 Amazon Redshift Serverless 数据仓库进行通信，从而确保与现有网络架构和未来网络架构兼容。

# Amazon Redshift Serverless 的计算容量

使用 Amazon Redshift Serverless，计算容量可以自动纵向扩展和缩减以满足工作负载需求。计算容量是指分配给 Amazon Redshift Serverless 工作负载的处理能力和内存。常见用例包括处理流量高峰期、运行复杂分析或高效处理大量数据。以下术语提供了有关 Amazon Redshift 如何管理计算容量的详细信息。

PRU

Amazon Redshift Serverless 以 Redshift 处理单元 (RPU) 为单位测量数据仓库容量。RPU 是用于处理工作负载的资源。一个 RPU 提供 16 GB 内存。

基本容量

此设置指定 Amazon Redshift Serverless 用于处理查询的基本数据仓库容量。基本容量以 RPU 为单位指定。您能够以 Redshift 处理单元 (RPU) 为单位设置基本容量。设置更高的基本容量可确保提高查询

性能，尤其是对于需要大量资源的数据处理作业。Amazon Redshift Serverless 的原定设置基本容量为 128 个 RPU。您可以将基本容量设置从 4 个 RPU 调整为 512 个 RPU。您可以将此值设置为 4 个 RPU，或者在 8 个或大于 8 个 RPU 时采用 8 的倍数（8、16、24...512）。您可以使用 AWS 管理控制台、`UpdateWorkgroup` API 操作或 AWS CLI 中的 `update-workgroup` 操作来设置此值。

由于最低基本容量为 4 个 RPU，您可以根据数据仓库的成本和容量需求，灵活地运行从较简单到更复杂的工作负载。4 个基本 RPU 容量面向所含数据少于 32 TB 的仓库，而 8、16 和 24 个 RPU 的基本 RPU 容量面向所需数据少于 128 TB 的工作负载。如果数据要求大于 128 TB，则必须至少使用 32 个基本 RPU。此外，如果工作负载中的表具有大量列数和更高的并发度，我们建议使用 32 个或更多的基本 RPU。

可用的最大基本 RPU 为 1024，可为工作负载添加最高级别的计算资源。这为支持高度复杂的工作负载提供了更大的灵活性，并加快了数据的加载和查询速度。

> **Note**
>
> 以下 AWS 区域中提供 1024 的扩展最大基本 RPU 容量。在其它区域中，最大基本容量为 512 个 RPU。
>
> - 美国东部（弗吉尼亚州北部）
> - 美国东部（俄亥俄州）
> - 美国西部（俄勒冈州）
> - 欧洲地区（爱尔兰）
> - 欧洲地区（法兰克福）
>
> 将基本容量设置在 512-1024 之间时，可以按 32 为单位递增或递减 RPU。

如果您管理更大、更复杂的工作负载，可以考虑增加 Redshift Serverless 数据仓库的大小。较大的仓库可以访问更多的计算资源，从而可以更高效地处理查询。

下面是一些拥有更高的基本容量会带来好处的实例：

- 查询很复杂，运行需要很长时间。
- 表中有大量的列。
- 查询包含了大量的 JOIN。
- 查询聚合或扫描来自外部源（例如数据湖）的大量数据。

有关 Amazon Redshift Serverless 配额和限制的更多信息，请转至 Amazon Redshift Serverless 对象的配额。

## Amazon Redshift Serverless 容量的注意事项和限制

Amazon Redshift Serverless 容量有下列注意事项和限制 有关 Redshift Serverless 的一般注意事项，请参阅使用 Amazon Redshift Serverless 时的注意事项。

- 4 个基本 RPU 的配置支持高达 32 TB 的托管式存储容量。如果您使用的托管式存储超过 32 TB，则无法将基本 RPU 设置为少于 8 个 RPU。
- 8 或 16 个基本 RPU 的配置支持高达 128 TB 的 Redshift 托管式存储容量。如果您使用的托管式存储超过 128 TB，则无法将基本 RPU 设置为少于 32 个 RPU。
- 编辑工作组的基本容量，可能会导致工作组上运行的一些查询被取消。
- Redshift Serverless 使用以下增量为您的数据仓库扩展 RPU：
  - 4 到 8 个 RPU：以 4 个 RPU 为单位递增。
  - 8 到 512 个 RPU：以 8 个 RPU 为单位递增。
  - 512 到 1024 个 RPU：以 32 个 RPU 为单位递增。
- 只有 8 个或更多的 RPU 才支持 Vacuum boost。对于 8 个及更少的 RPU，请改用以下命令：

```
VACUUM [FULL | SORT ONLY | DELETE ONLY | REINDEX | RECLUSTER] [table_name] [TO
 threshold PERCENT]
```

### 具有 4 个 Redshift 处理器（RPU）容量的 Redshift Serverless

具有 4 个基本 RPU 容量的 Redshift Serverless 非常适合较小或要求较低的工作负载。该入口点提供了灵活且经济实惠的解决方案。此入门级配置支持最多具有以下资源的数据仓库：

- 多达 32 TB 的 Redshift 托管式存储。
- 每个表最多 100 列
- 64 GB 的内存

如果您需要超过这些限制，则必须手动增加基本容量，而不是依赖自动扩缩。一旦您将数据仓库扩展到超过 4 个 RPU，数据仓库将继续使用更多的 RPU，而且 Amazon Redshift 不会将数据仓库缩减回到 4 个 RPU。

> **Note**
>
> 使用 4 个基本 RPU 时，您可以创建超过 100 列的表，但我们建议您将表限制在 100 列。超过此限制可能导致数据仓库在查询执行期间耗尽其内存，这会降低性能。

您可以在以下 AWS 区域中创建使用 4 个 RPU 的数据仓库：

- 美国东部（俄亥俄州）
- 美国东部（弗吉尼亚州北部）
- 美国西部（北加利福尼亚）
- 美国西部（俄勒冈州）
- 亚太地区（孟买）
- 亚太地区（新加坡）
- 亚太地区（悉尼）
- 亚太地区（东京）
- 欧洲地区（爱尔兰）
- 欧洲地区（斯德哥尔摩）

## AI 驱动的扩展和优化

> **Note**
>
> AI 驱动的扩展和优化功能适用于提供 Amazon Redshift Serverless 的所有 AWS 区域。

Amazon Redshift Serverless 提供 AI 驱动的高级扩展和优化功能，可满足不同的工作负载要求。数据仓库可能存在以下预置问题：

- 为了提高资源密集型查询的性能，可能会过度预置数据仓库
- 为了节省成本，数据仓库可能预置不足。

在数据仓库工作负载的性能和成本之间取得适当平衡具有挑战性，尤其是对于临时查询和在数据量不断增长的情况下。在运行混合工作负载（包括低资源密集型和高资源密集型查询）时，需要进行智能

扩展。AI 驱动的扩展和优化功能可根据数据增长自动扩展无服务器计算或 RPU。此功能还有助于将查询性能保持在目标性价比目标范围内。AI 驱动的扩展和优化会随着数据量增加而动态分配计算资源，从而确保查询持续达到性能目标。AI 驱动的扩展和优化使该服务能够无缝适应不断变化的工作负载需求，而无需手动干预或复杂的容量规划。

Amazon Redshift Serverless 根据查询复杂度和数据量等因素，提供更全面、响应更快的扩展解决方案。此功能考虑到优化工作负载的性价比，同时保持灵活性，以高效处理不断变化的工作负载和不断增长的数据集。Amazon Redshift Serverless 可以自动对 Amazon Redshift Serverless 端点进行 AI 驱动的扩展和优化，以满足您为无服务器工作组指定的性价比目标。如果您不知道要为工作负载设置多少基本容量，或者您的工作负载的某些部分可能从分配更多资源中获益，则这种自动性价比优化特别有用。

示例

如果组织运行的工作负载通常只需要 32 个 RPU，但突然引入了更复杂的查询，那么您可能不确定多大的基本容量才合适。设置较高的基本容量可以提高性能，但也会产生更高的成本，因此成本可能会不符合您的期望。Amazon Redshift Serverless 使用 AI 驱动的扩展和优化，自动调整 RPU 以满足您的性价比目标，同时优化组织的成本。无论工作负载大小如何，这种自动优化都会非常有用。如果您有任意数量的复杂查询，自动优化可以帮助您实现组织的性价比目标。

> **Note**
>
> 性价比目标是特定于工作组的设置。不同的工作组可以有不同的性价比目标。

要配置性价比目标，请使用 AWS 控制台。默认情况下，所有新的无服务器工作组都启用了性价比目标，并将此目标设置为平衡。您可以在创建无服务器工作组后修改性价比目标。

编辑工作组的性价比目标

1. 在 Amazon Redshift Serverless 控制台中，选择工作组配置。
2. 选择要为其编辑性价比目标的工作组。选择性能选项卡，然后选择编辑。
3. 选择性价比目标，然后将滑块调整到所需的设置。
4. 选择保存更改。
5. 要更新 Amazon Redshift Serverless 可以向工作负载分配的最大 RPU 数量，请选择工作组配置部分的限制选项卡。

可以使用性价比目标滑块在成本和性能之间设置所需的平衡。通过移动滑块，您可以选择以下选项之一：

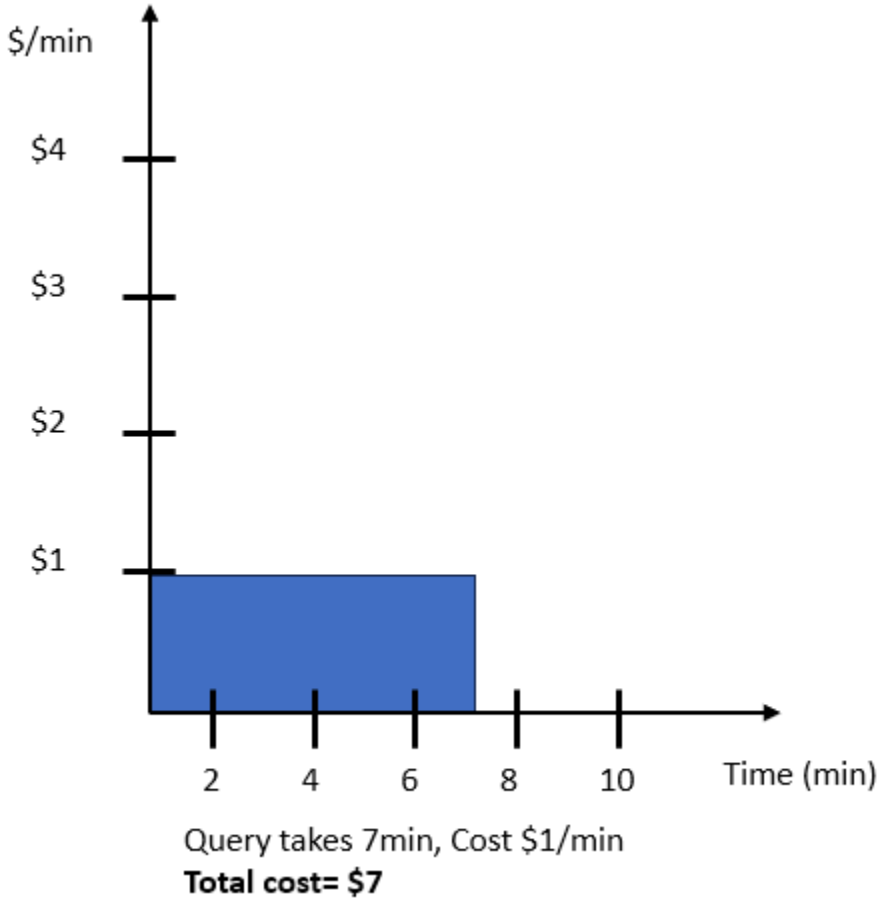
- 针对成本进行优化 – 此设置优先考虑成本节约。在有望降低成本的情况下，Amazon Redshift Serverless 会尝试自动纵向扩展计算容量。Amazon Redshift Serverless 还尝试缩减计算资源以降低成本，这可能会增加查询运行时间。
- 平衡 - 此设置可在性能和成本之间取得平衡。Amazon Redshift Serverless 可针对性能进行扩展，并可能导致适度的成本增加或减少。这是对于大多数 Amazon Redshift Serverless 数据仓库的建议设置。
- 针对性能进行优化 - 此设置优先考虑性能。为了获得高性能，Amazon Redshift 会前瞻性扩展，这可能会产生更高的成本。
- 中间位置：也可以将滑块设置为平衡与针对成本进行优化或针对性能进行优化之间的两个中间位置之一。如果针对成本或性能的全面优化过于极端，请使用这些设置。

## 选择性价比目标时的注意事项

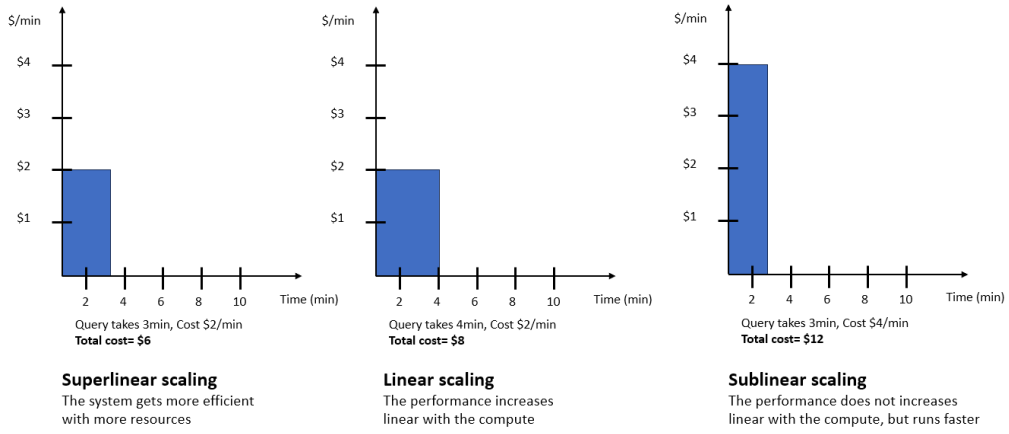
可以使用性价比滑块来为工作负载选择所需的性价比目标。AI 驱动的扩展和优化算法会随时间推移从工作负载历史记录中进行学习，并提高预测和决策的准确性。

示例

对于此示例，假设查询耗时 7 分钟，花费 7 美元。下图显示了未扩展情况下的查询运行时间和成本。




给定的查询可能会以几种不同的方式进行扩展，如下所示。根据您选择的性价比目标，AI 驱动的扩展和优化可以预测查询如何权衡性能与成本，并相应地对其进行扩展。选择不同的滑块选项会产生以下结果：




- 针对成本进行优化 – 使用针对成本进行优化选项，数据仓库可以进行扩展，且偏向于降低成本的选择。在前述示例中，超线性扩展方法演示了这种行为。只有根据扩展模型预测以具有成本效益的方式

进行扩展，才会进行扩展。如果扩展模型预测无法对给定的工作负载进行成本优化的扩展，则数据仓库将无法扩展。

- 平衡 – 使用平衡选项，系统可以在平衡成本和性能考虑因素的同时进行扩展，而成本的增加可能有限。平衡选项可执行超线性、线性以及可能的次线性工作负载扩展。
- 针对性能进行优化 – 使用针对性能进行优化选项，除了以前用于提高性能的方法外，系统也可以进行扩展（即使成本更高也是如此），而且可能与运行时间改进不成比例。使用针对性能进行优化，系统可以执行超线性扩展、线性扩展和次线性扩展（如果可能）。滑块位置越接近针对性能进行优化位置，Amazon Redshift Serverless 允许的次线性扩展就越多。

设置性价比滑块时，请注意以下几点：

- 您可以随时更改性价比设置，但工作负载扩展不会立即更改。随着系统了解当前的工作负载，扩展会随着时间推移而变化。我们建议对无服务器工作组进行 1-3 天的监控，以验证新设置的影响。
- 性价比滑块选项最大容量和最大 RPU 小时数一起发挥作用。最大容量和最大 RPU 小时数是相应的控件，用于限制 Amazon Redshift Serverless 支持数据仓库扩展的最大 RPU，以及 Amazon Redshift Serverless 支持数据仓库耗用的最大 RPU 小时数。无论性价比目标设置如何，Amazon Redshift Serverless 始终遵守并强制执行这些设置。

## 监控资源自动扩缩

可以通过以下方式监控 AI 驱动的扩展和优化：

- 在 Amazon Redshift 控制台上查看已使用的 RPU 容量图表。
- 在 CloudWatch 下中的 `AWS/Redshift-Serverless` 和 `Workgroup` 下监控 `ComputeCapacity` 指标。
- 查询 SYS_QUERY_HISTORY 视图。提供特定的查询 ID 或查询文本以标识时间段。使用此时间段查询 SYS_SERVERLESS_USAGE 系统视图来查找 `compute_capacity` 值。`compute_capacity` 字段显示在查询运行时期间扩展的 RPU。

使用以下示例查询 SYS_QUERY_HISTORY 视图。用查询文本替换示例值。

```
select query_id,query_text,start_time,end_time, elapsed_time/1000000.0
 duration_in_seconds
from sys_query_history
where query_text like '<query_text>'
and query_text not like '%sys_query_history%'
```

```
order by start_time desc
```

运行以下查询，来查看 compute_capacity 在从 start_time 到 end_time 的时间段内扩展的情况。将以下查询中的 start_time 和 end_time 替换为前一个查询的输出：

```
select * from sys_serverless_usage
where end_time >= 'start_time'
and end_time <= DATEADD(minute,1,'end_time')
order by end_time asc
```

有关使用这些功能的分步说明，请参阅 Configure monitoring, limits, and alarms in Amazon Redshift Serverless to keep costs predictable。

### 使用 AI 驱动的扩展和优化时的注意事项

使用 AI 驱动的扩展和优化时应考虑以下事项：

- 对于 Amazon Redshift Serverless 上需要 8 到 512 个基本 RPU 的现有工作负载，可以使用 AI 驱动的扩展和优化来获得最佳结果。我们建议不要将此功能用于高于或低于此 RPU 范围的工作负载。
- 性价比目标会自动优化工作负载，但结果可能会有所不同。请持续使用此功能，以便系统通过运行具有代表性的工作负载来了解您的特定模式。
- AI 驱动的扩展和优化根据 Amazon Redshift Serverless 实例上运行的工作负载，使用最佳时间将优化应用于无服务器工作组。

要了解有关 AI 驱动的扩展和优化的更多信息，请观看以下视频。

# Amazon Redshift Serverless 的计费

## 计算容量的计费

您可以通过两种方式购买 Amazon Redshift Serverless 的容量：

- 您可以购买按需容量：当您选择按需计算容量时，您可以按实际使用量支付资源费用。如果您刚刚开始使用 Amazon Redshift Serverless，或者您还没有很好地了解稳定的使用模式，那么这是最佳选择。按需可提供最大的灵活性。有关更多信息，请参阅 按需计算容量的计费。
- 您可以购买预留：当您在特定时间（例如一年）内购买预设数量的计算资源时，预留可提供折扣。当您知道您将稳定地使用一定量的容量时，这是一个好主意。当您可以预测部分容量需求时，这有助于节省资金。有关更多信息，请参阅 无服务器预留的计费。

您可以同时使用预留和按需资源。但不强制要求您要使用其中哪种方法。

有关详细的定价信息，请参阅 Amazon Redshift 定价。

## 按需计算容量的计费

基本容量及其对计费的影响

当查询运行时，系统将根据给定持续时间内使用的容量（以 RPU 小时为单位）按秒对您进行收费。如果没有任何查询在运行，则您不需要为计算容量付费。您还需要根据存储的数据量为 Redshift 托管式存储 (RMS) 付费。

创建工作组时，您可以选择设置计算的基本容量。为满足工作组级别的工作负载的性价比要求，可以调高或调低现有工作组的基本容量。从工作组配置中选择工作组，然后选择限制选项卡，以使用控制台更改基本容量。

随着查询数量增加，Amazon Redshift Serverless 会自动扩展以提供一致的性能。

最大 RPU 小时数使用限制

为了使 Amazon Redshift Serverless 的成本保持可预测，您可以设置每天、每周或每月使用的最大 RPU 小时数。您可以使用控制台或 API 来设置该值。您可以指定在达到限制时，将日志条目写入系统表、接收告警或者关闭用户查询。设置最大 RPU 小时数有助于控制成本。对于访问数据仓库中数据的查询和访问外部数据的查询（例如 Amazon S3 中的外部表），有关最大 RPU 小时数的设置适用于您的工作组。

以下是示例：

假设您设置了每周 100 小时的限制。要在控制台上完成此设置，请执行以下操作：

1. 选择您的工作组，然后在限制选项卡下选择管理使用量限制。
2. 添加使用量限制，选择每周频率，持续时间为 100 小时，并将操作设置为关闭用户查询。

在此示例中，如果您在一周内达到 RPU 100 小时的限制，则将关闭查询功能。

为工作组设置最大 RPU 小时数不会限制工作组的性能或计算资源。您可以随时调整此设置，而不会影响查询处理。设置最大 RPU 小时数的目标是帮助您满足价格和性能要求。有关无服务器计费的更多信息，请参阅 Amazon Redshift 定价。

保持 Amazon Redshift Serverless 成本可预测的另一种方法是使用 AWS 成本异常检测，以减少账单出现意外情况的可能性并提供更多控制权。

> **Note**
>
> Amazon Redshift 定价计算器有助于估算价格。您输入所需的计算资源，该计算器将提供成本预览。

## 设置最大容量以控制计算资源的成本

最大容量设置用作 Amazon Redshift Serverless 可以纵向扩展到的 RPU 上限。它有助于控制计算资源的成本。与基本容量用于设置可用计算资源最小数量的方法类似，最大容量设定了 RPU 使用量的上限。它通过这种方法确保您的支出符合计划。最大容量特定于每个工作组应用，它可以随时限制计算资源使用量。

最大容量与 RPU 小时使用量限制的不同之处

最大 RPU 小时数限制和最大容量都可以帮助您控制成本，但它们的运作方式不同。以下要点解释了其中的区别：

- 最大容量 – 此设置确定 Amazon Redshift Serverless 进行自动扩展可使用的最大 RPU 数量。
- 最大 RPU 小时数使用限制 – 与最大容量不同，此设置不设置容量上限。但它会执行其他操作来帮助您限制成本。这包括在日志中添加条目、通知您或停止运行查询（视您的选择而定）。

您可以单独使用最大容量，也可以利用最大 RPU 小时数使用限制中的操作作为补充。

最大容量使用案例

每个工作组可以有不同的最大容量设置。它可以帮助您强制实施预算要求。为了说明其工作原理，假设以下案例：

- 您有一个工作组，基本容量设置为 256 个 RPU。在一个月的大部分时间里，您的工作负载稳定在略微超过 256 个 RPU。
- 最大容量设置为 512 个 RPU。

假设在三天的时间内，您需要突发的高使用量来生成临时统计报告。在这种情况下，您可以设置最大容量，以避免计算成本超过 512 个 RPU。通过这样做，您可以确保计算容量不会超过这个上限。

最大容量的使用说明

以下说明可以帮助您正确地设置最大容量：

- 每个 Amazon Redshift Serverless 可以有不同的最大容量设置。
- 如果您有一段时间资源使用量会非常高，并且最大容量设置为较低的 RPU 水平，则可能会导致工作负载处理延误和欠佳的用户体验。
- 您可以在 Amazon Redshift Serverless 控制面板上查看一段时间内使用的容量。有关查看摘要数据的更多信息，请参阅使用控制面板检查 Amazon Redshift Serverless 摘要数据。
- 最大容量相关更改通常不会中断正在运行的查询。但是，如果您将最大容量设置为低于当前的基本容量，Amazon Redshift Serverless 将立即减少容量。这可能会影响正在运行的工作负载。
- 最大容量设置最多可以设置为 5632 个 RPU.

如何设置最大容量

您可以在控制台中设置最大容量。对于现有工作组，您可以在工作组配置下更改设置。您也可以使用 CLI，使用类似以下示例的命令进行设置：

```
aws redshift-serverless update-workgroup --workgroup-name myworkgroup --max-capacity
 512
```

这将为具有指定名称的工作组设定最大容量设置。设置完成后，您可以在控制台上检查值以进行验证。您也可以使用 CLI 运行 `get-workgroup` 命令来检查值。

您可以通过将其设置为 `-1` 来最大容量设置，如下所示：

```
aws redshift-serverless update-workgroup --workgroup-name myworkgroup --max-capacity -1
```

## 监控 Amazon Redshift Serverless 使用量和成本

您可以通过多种方法来估计 Amazon Redshift Serverless 的使用情况和计费。系统视图可能会很有用，因为系统元数据（包括查询和使用情况数据）是及时的，并且您无需进行任何设置即可查询到。CloudWatch 还可用于监控 Amazon Redshift Serverless 实例的使用情况，并具有提供洞察和设置操作的附加功能。

通过查询系统视图可视化使用情况

查询 SYS_SERVERLESS_USAGE 系统表以跟踪使用情况并获取查询费用：

```
select trunc(start_time) "Day",
(sum(charged_seconds)/3600::double
precision) * <Price for 1 RPU> as cost_incurred
```

```
from sys_serverless_usage
group by 1
order by 1
```

此查询根据使用情况估算 Amazon Redshift Serverless 每天产生的费用。

用于确定使用情况和成本的使用说明

- 您按秒支付以 RPU 小时为单位运行的工作负载，最低收费时间为 60 秒。
- 来自 sys_serverless_usage 系统表的记录显示 1 分钟时间间隔产生的成本。了解以下各列很重要：

  charged_seconds 列：
  - 提供在时间间隔内计费的计算单位 (RPU) 秒数。结果包括 Amazon Redshift Serverless 中的任何最低收费。
  - 获取有关事务完成后计算资源使用情况的信息。因此，如果事务尚未完成，则此列的值可能为 0。

  compute_seconds 列：
  - 提供实时计算使用情况信息。这不包括 Amazon Redshift Serverless 中的任何最低收费。因此，它可能在某种程度上与间隔内计费的收费秒数不同。
  - 显示每个事务期间（即使事务尚未结束）的使用信息，因此提供的数据是实时的。
- 在某些情况下，compute_seconds 为 0，但 charged_seconds 大于 0，反之亦然。这是由于在系统视图中记录数据的方式而导致的正常行为。为了更准确地表示无服务器使用情况的详细信息，我们建议在 SYS_SERVERLESS_USAGE 中聚合数据。

有关监控表和视图的更多信息，请参阅使用 Amazon Redshift Serverless 监控查询和工作负载。

使用 CloudWatch 可视化使用情况

您可以使用 CloudWatch 中提供的指标来跟踪使用情况。为 CloudWatch 生成的指标是 `ComputeSeconds` 和 `ComputeCapacity`，前者表示当前分钟内使用的总 RPU 秒数，后者表示该分钟的总计算容量。也可以在 Redshift 控制台上的 Redshift 无服务器控制面板中找到使用情况指标。有关 CloudWatch 的更多信息，请参阅什么是 Amazon CloudWatch？

## 无服务器预留的计费

Amazon Redshift Serverless 支持您运行和扩展分析，而无需使用即用即付定价模式来预置和管理集群。现在，通过无服务器预留，您可以进一步优化计算成本，并提高 Redshift Serverless 上现有和新工作负载的成本可预测性。

Amazon Redshift 在 AWS 付款人账户级别管理无服务器预留，预留可以在多个 AWS 账户之间共享，可让您在 AWS 账户中的所有 Redshift Serverless 工作负载上降低高达 24% 的计算成本。Amazon Redshift 按小时对无服务器预留计费，按秒对预留进行计量，提供每周七天、每天 24 小时的一致计费模式，同时保持 Redshift Serverless 提供的灵活性。Amazon Redshift 按标准按需费率，对任何超过指定 RPU 水平的使用量收取费用。

> **Note**
>
> 如果您要限制按需使用量，可以使用最大容量设置来为工作组设置资源使用限制。有关更多信息，请参阅 Amazon Redshift Serverless 的计费。

## 无服务器预留的优势

无服务器预留是 Amazon Redshift Serverless 的折扣定价选项。无服务器预留让您可以选择在一年内以按需（OD）费率的折扣承诺指定数量的 Redshift 处理器（RPU），而无需预付款。通过预付款，您可以获得更大的折扣。通过无服务器预留，您可以优化计算成本，并提高无服务器上现有和新工作负载的成本可预测性。

每个无服务器预留都是在 AWS 账户级别购买的，可以在同一个付款人账户中的多个 Amazon Redshift Serverless 工作组之间共享。这使您能够灵活地应用折扣。具有不同工作负载模式的多个工作组可以共享预留。

## 无服务器预留的工作原理

预留 RPU 是一个简单的过程，只需几分钟即可完成。它包括指定要预留的 RPU 级别和付款类型。Amazon Redshift Serverless 使用标准 AWS 账单与成本管理工具，有助于您确定所需的预留级别并持续监控您的使用情况。无服务器预留在 AWS 付款人账户级别进行管理，可以在同一个付款人账户下共享，这样您就可以将您 AWS 账户中所有 Redshift Serverless 工作负载的计算成本降低多达 24%。无服务器预留按小时计费，按秒计量，提供每周七天、每天 24 小时的一致计费模式，同时保持 Redshift Serverless 提供的灵活性。任何超过指定 RPU 级别的使用量均按标准 Redshift Serverless 按需费率收费。

您可以在同一个 AWS 账户中购买多个无服务器预留。当您购买额外的无服务器预留时，它们会相互叠加。例如，如果您购买了两个预留并为每个预留选择了 100 个 RPU，则可以按折扣费率获得总共 200 个 RPU。

> **Note**
>
> 如果要设置按需使用的限制，则可以在 Amazon Redshift Serverless 控制台中选择限制选项卡，然后选择管理使用量限制，来为工作组设置最大 RPU。

购买无服务器预留后，它会立即生效，并显示在 Redshift 控制台的无服务器预留控制面板中。

## 分析您的 RPU（Redshift 处理器）使用情况，以确定所需的预留级别

Redshift Serverless 预留可让您通过承诺在一年内使用特定数量的 Redshift 处理器（RPU），为您提供比按需定价更低的折扣，从而锁定可预测、更低的计算成本。如果使用无预付选项，则这些折扣可高达 20%；如果您预付全额费用，则折扣可高达 24%。由于您在 AWS 付款人账户级别购买 Redshift Serverless 预留，并且节省自动应用于任何 AWS 关联账户中的任何 Redshift Serverless 工作组，因此您可以在支持多个团队的同时集中管理预算。Redshift Serverless 以每秒的粒度计量使用量，按每小时计算平均值，然后按小时计费，确保您只为您使用的容量付费。Redshift Serverless 预留将跨账户的灵活应用与基于期限的节省相结合，在不牺牲 Redshift Serverless 敏捷性的前提下为您提供可预测的分析价格。

分析预留的 RPU 使用量

可以通过以下两种方式之一来确定 RPU 使用量级别：可以使用 Redshift Serverless 控制面板来查看七天视图，也可以使用 Cost Explorer 成本管理服务来进行长期分析。以下过程演示了如何分析 RPU 使用量：

方法 1：Redshift Serverless 控制面板（7 天视图）

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 打开 Serverless 控制面板。
3. 选择您的工作组。
4. 查看从最后一小时到一周的 RPU 容量使用情况。

方法 2：AWS Cost Explorer（长期分析）

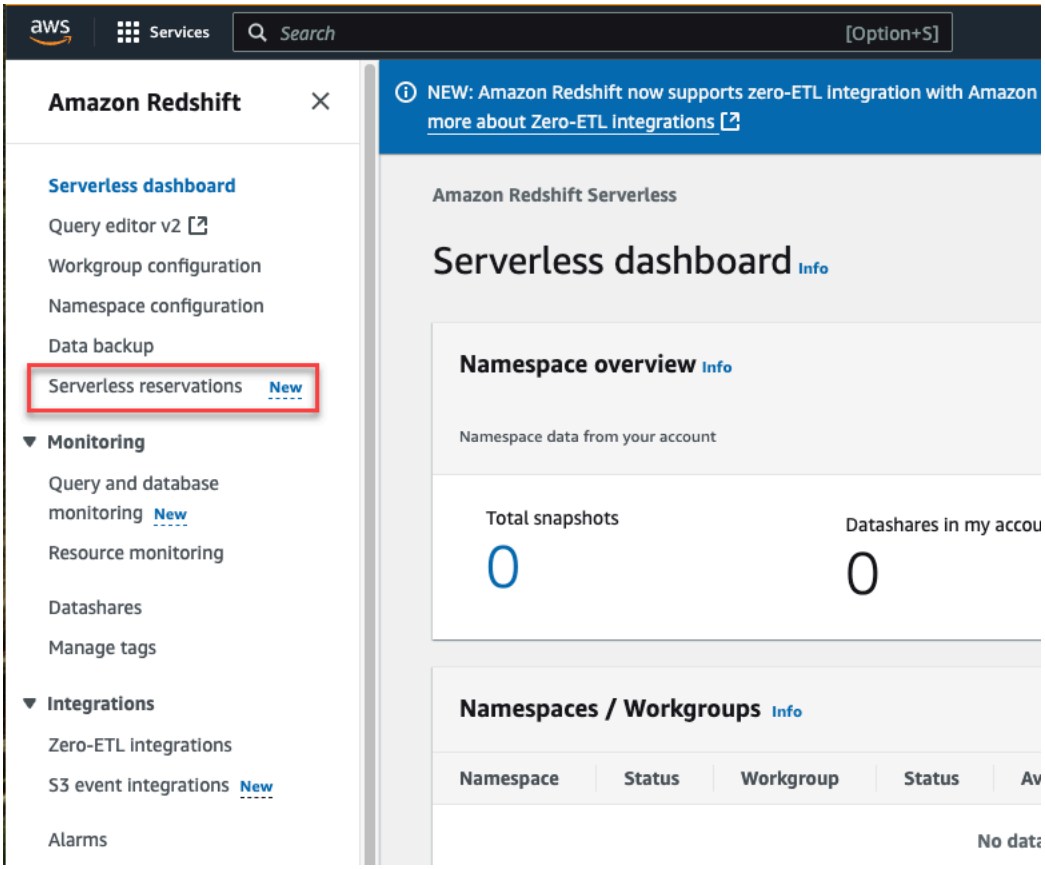
1. 登录 AWS 管理控制台 并打开 Cost Explorer 控制台，网址为 https://console.aws.amazon.com/costmanagement/。
2. 将粒度设置为每小时

3. 按使用类型分组
4. 应用以下筛选条件：
   - 服务：Redshift
   - 区域：您的本地区域
   - 使用类型：Redshift:ServerlessUsage 的筛选条件
5. 查看成本和使用情况图表，来了解您所选区域的每小时无服务器使用情况

## 使用控制台购买无服务器预留

购买预留时，您可以选择可享受折扣的 RPU 级别。在选择 RPU 级别之前，最好先了解您的基本容量和随时间推移使用的按需容量。本接向您展示如何确定容量和预留无服务器预留。

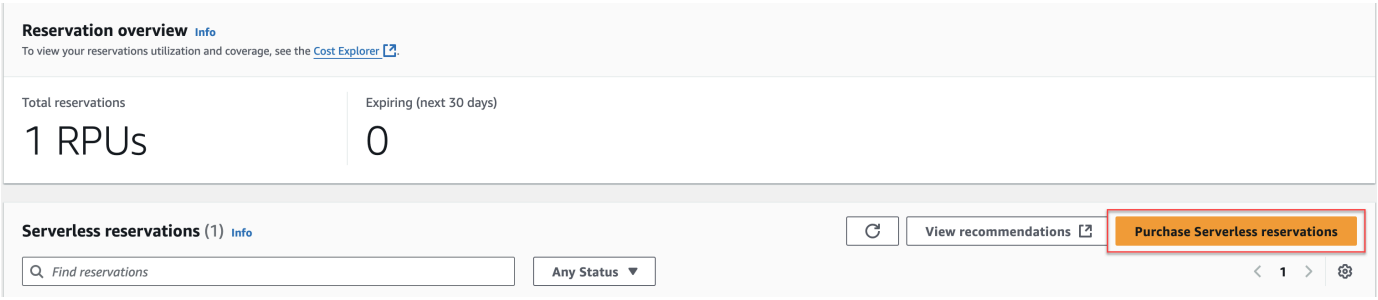
首先，在 Redshift 控制台中，选择无服务器，然后从菜单中选择无服务器预留。



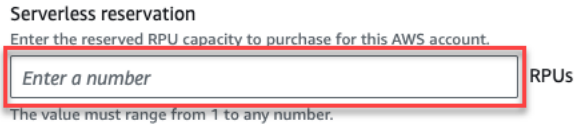

控制台显示该功能的描述和现有预留的列表。您可以从此处购买预留，也可以使用所提供的报告和监控工具来查看当前使用量。这些信息有助于您确定 RPU 级别以及适合预留的 RPU 数量。

要购买预留，请完成以下步骤：

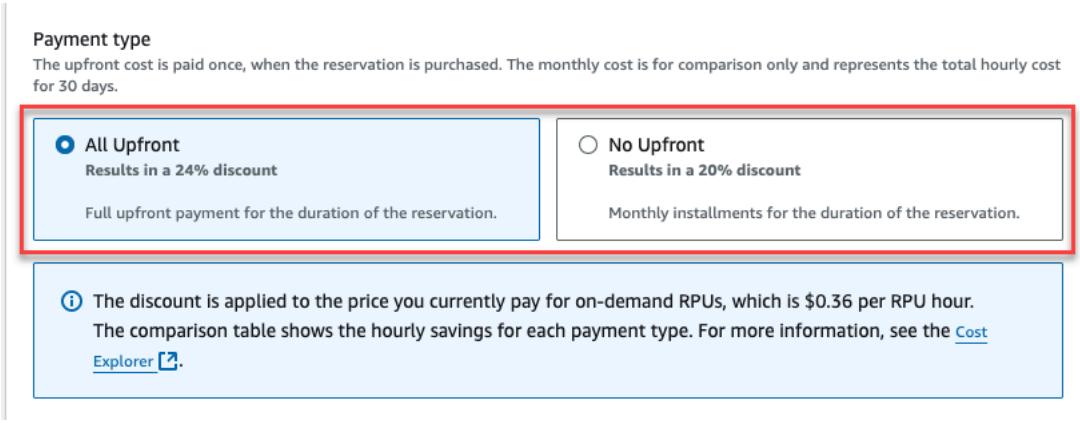
1. 选择购买无服务器预留。




2. 此时会出现一个演练，其中包含一系列选择。输入要预留的无服务器预留 RPU 级别。如果您不确定这个级别应该是什么，可以使用本节中进一步描述的工具。

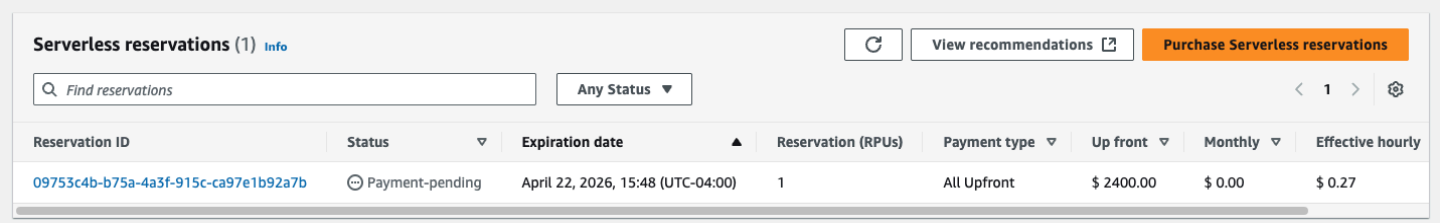



3. 设置付款类型。您可以选择为预留的 RPU 进行预付款，也可以按月付款。如果您选择预付款，则可以获得更大的折扣。




4. 完成选择后，选择购买无服务器预留，然后选择确认。

确认预留后，它将出现在预留列表中。




## 使用说明

- 您无法更改或删除预留。但您可以创建额外的预留以覆盖更多范围。
- Redshift Serverless 在使用按需 RPU 之前，使用预留 RPU 来处理工作负载，以确保节省成本。如果您超出预留的 RPU 数量，则将以 Redshift Serverless 按需费率为这些额外的 RPU 计费。
- Amazon Redshift Serverless 的免费服务抵扣金不适用于无服务器预留，仅适用于按需计费的 RPU。

## 无服务器预留示例

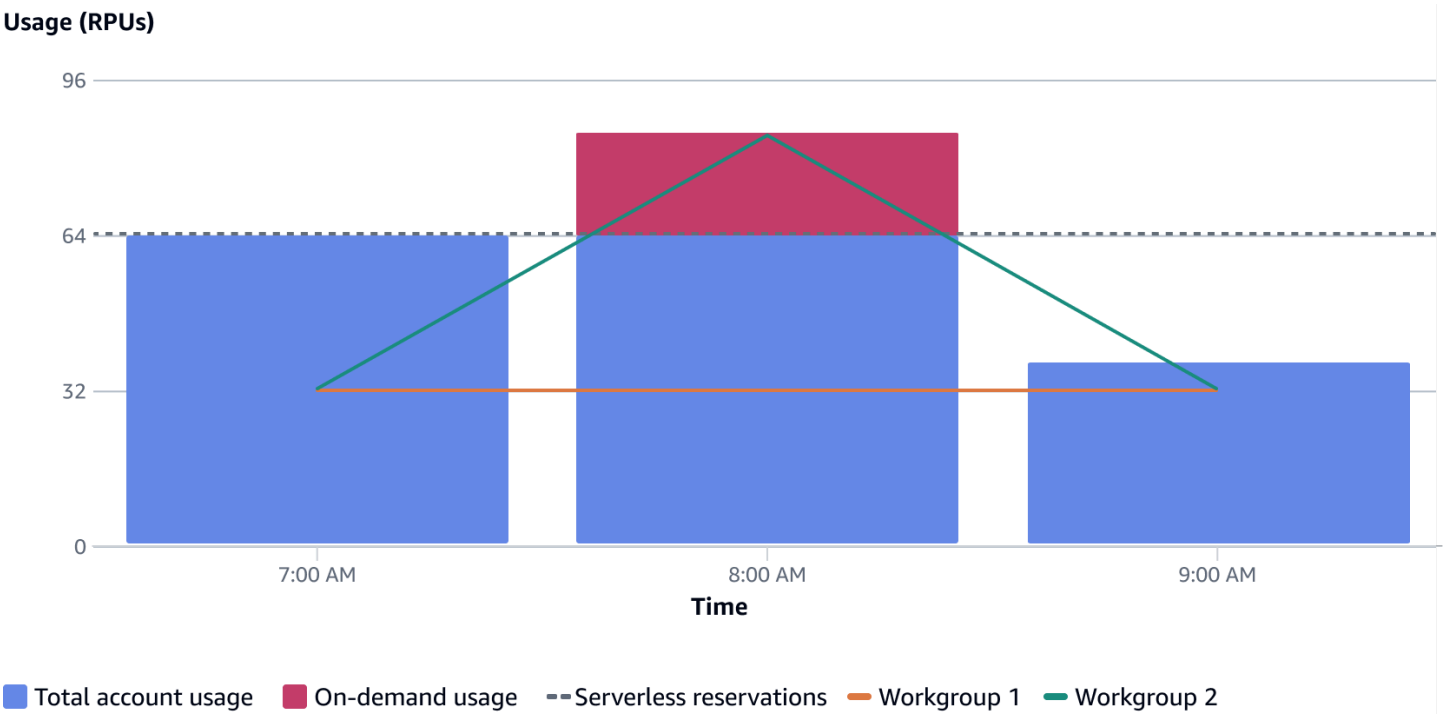
在这种情况下，您的 AWS 付款人/关联账户有两个 Amazon Redshift 工作组：

- 工作组 1 具有稳态使用量，例如用于商业智能团队。
- 工作组 2 的工作负载不可预测，使用量会出现激增，例如用于 ETL 操作。

您想优化这些工作组的成本，因此，您可以购买为期一年的无服务器预留。根据历史数据，您可以确定两个工作组在稳定状态下要消耗 64 个 RPU。但是，工作组 2 偶尔会从 32 个 RPU 增加到 48 个 RPU，并在短时间内下降到 24 个 RPU。开始时，您可以将预留的 RPU 级别设置为 64 个 RPU，这与历史趋势一致。按小时计费的详细信息如下：

- 与历史使用趋势类似，在第一个小时内，两个工作组都使用 32 个 RPU，账户总使用量为 64 个 RPU。在这个小时内，所有 RPU 均按无服务器预留折扣费率收费。这是因为 64 个 RPU 的使用量级别等于 64 个 RPU 的无服务器预留。
- 在第二小时内，工作组 1 继续使用 32 个 RPU。但是，工作组 2 激增至 48 个 RPU，账户总使用量为 80 个 RPU。在这个小时内，64 个 RPU 按无服务器预留折扣费率收费，而 16 个 RPU 按照 Redshift Serverless 按需费率收费。
- 在第三个小时内，工作组 1 继续使用 32 个 RPU，而工作组 2 减少到 8 个 RPU。在这个小时内，尽管账户总数为 40 个 RPU，但仍按照 64 个 RPU 无服务器预留费率向该账户收费。

有关工作组使用量演变以及按需和无服务器预留费率账单详情，请参阅下图：




### 使用 AWS CLI 或 Amazon Redshift API 购买无服务器预留

您使用 `create-reservation` 创建 RPU 预留。下面显示了此命令：

```
create-reservation
--capacity
--offering-id
```

您将 `capacity` 设置为要预留的 RPU 数量。

## 对存储计费

主存储容量按 Redshift 托管式存储 (RMS) 计费。存储按 GB/月计费。存储计费与计算容量的计费是分开的。用于手动快照的存储空间按标准备份计费费率，根据所有活跃手动快照的唯一数据块总数进行计费。保留时间少于 24 小时的恢复点不收费。

数据传输成本和机器学习 (ML) 成本单独计费，与预置集群适用相同的费率。跨 AWS 区域的快照复制和数据共享按定价页面上列出的传输费率计费。有关更多信息，请参阅 Amazon Redshift 定价。

### 使用 CloudWatch 可视化使用情况计费

将生成指标 `ServerlessUniqueSnapshotStorage`（用于跟踪快照存储使用情况）并发送到 CloudWatch。有关 CloudWatch 的更多信息，请参阅什么是 Amazon CloudWatch？

# 使用 Amazon Redshift Serverless 免费试用版

Amazon Redshift Serverless 提供免费试用。如果您参加免费试用，则可以在 Redshift 控制台中查看免费试用积分余额，然后在 SYS_SERVERLESS_USAGE 系统视图中检查免费试用使用情况。请注意，免费试用的账单详细信息不会显示在账单控制台中。免费试用结束后，您只能在账单控制台中查看使用情况。有关 Amazon Redshift Serverless 免费试用版的更多信息，请参阅 Amazon Redshift Serverless 免费试用。

## 账单使用注释

- 记录使用情况 - 查询或事务仅在事务完成、回滚或停止后才计量和记录。例如，如果事务运行两天，则在此事务完成后记录 RPU 使用情况。您可以通过查询 `sys_serverless_usage` 来实时监控正在进行的使用情况。事务记录可能反映为 RPU 使用情况变化，并影响特定小时数和每日使用情况的成本。
- 编写显式事务 - 作为结束事务的最佳实践，这一点很重要。如果您没有结束或回滚未结事务，Amazon Redshift Serverless 将继续使用 RPU。例如，如果您编写一个显式 `BEGIN TRAN`，务必具有相应的 `COMMIT` 和 `ROLLBACK` 语句。
- 已取消的查询 - 如果您运行了查询并在查询结束之前将其取消，您仍需要按查询运行时间付费。
- 扩缩 - Amazon Redshift Serverless 实例可以启动扩缩以处理较高负载时段，从而保持一致的性能。您的 Amazon Redshift Serverless 账单同时包括以相同 RPU 费率计算的基本计算和扩展容量。
- 缩减 - Amazon Redshift Serverless 从其基本 RPU 容量纵向扩展以处理负载较高的时期。在某些情况下，在查询负载下降后的一段时间内，RPU 容量可以保持在较高的设置。我们建议您在控制台中设置最大 RPU 小时数，以防止出现意外的成本。
- 系统表 - 查询系统表时，会对查询时间进行计费。
- Redshift Spectrum - 当您拥有 Amazon Redshift Serverless 并运行查询时，不对数据湖查询单独收费。对于针对 Amazon S3 中存储的数据进行的查询，按事务时间计算，费用与针对本地数据进行的查询相同。
- 联合查询 - 联合查询按特定时间间隔内使用的 RPU 付费，与针对数据仓库或数据湖的查询相同。
- 存储 - 存储单独收费，按 GB/月计费。
- 最低收费 – 计算资源的使用最低收取 60 秒的费用，按秒计费。
- 快照计费 - 快照计费不变。它根据存储空间收费，按每月 GB 的费率计费。您可以按 30 分钟的粒度将数据仓库还原到过去 24 小时内的特定点，无需付费。有关更多信息，请参阅 Amazon Redshift 定价。
- 使用额外的计算资源运行的自动优化：Amazon Redshift Serverless 通常会在用户查询的同时运行自动优化操作。这些操作称为自主处理，您无需为其付费。

如果您启用分配额外的计算资源，则即使在有大量用户活动的时段，Amazon Redshift 也会在必要时运行自主处理。在此类情况下，您可能要为运行自主处理所花的时间付费。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的分配额外的计算资源来执行自动数据库优化。

### Amazon Redshift Serverless 保持账单可预测性的最佳实践

以下是最佳实践和内置设置，有助于您保持账单一致性。

- 确保结束每个事务。当您使用 BEGIN 开始交易时，务必也要使用 END 结束交易。
- 使用最佳实践错误处理来从容地响应错误并结束每个事务。尽量减少未结交易有助于避免不必要地使用 RPU。
- 使用 SESSION TIMEOUT 帮助结束未结事务和空闲会话。它会导致任何保持闲置状态或非活动状态的时间已超过 3600 秒（1 小时）的会话超时。它会导致任何保持打开状态和非活动状态的时间已超过 21600 秒（6 小时）的事务超时。可以为特定用户显式更改此超时设置，例如您希望为长时间运行的查询保持会话打开时。主题 CREATE USER 显示了如何为用户调整 SESSION TIMEOUT。
  - 在大多数情况下，我们建议您不要延长 SESSION TIMEOUT 值，除非您的用例特别需要。如果会话保持空闲状态并且存在未结交易，则可能会导致在会话关闭之前使用 RPU 的情况。这将导致不必要的成本。
  - Amazon Redshift Serverless 运行查询的最长时间为 86,399 秒（24 小时）。在 Amazon Redshift Serverless 结束与事务关联的会话之前，未结事务的最长不活动时间是六小时。有关更多信息，请参阅 Amazon Redshift Serverless 对象的配额。

## 使用连接池的 Amazon Redshift Serverless 计费

Amazon Redshift Serverless 将所有传入的查询均视为可计费的用户活动，包括连接池发送的轻量级运行状况检查查询。无论查询源自应用程序、JDBC/ODBC 驱动程序还是连接池框架，此行为都适用。每个运行状况检查查询都会触发计算用量，无论查询目的或来源如何，都会产生费用。因此，即使没有实际的用户工作负载在运行，维护开放的连接池也会产生成本。

连接池在应用程序和 Amazon Redshift Serverless 端点之间维护一个持久连接池。为了确保这些连接保持运行状况正常和可用，池化机制通常会定期发送轻量级查询或空查询（例如 SELECT 1）。这些自动查询用于验证连接状态。

使用连接池时，请考虑以下最佳实践，以最大限度地减少意外费用：

- 可以通过查看和优化连接池配置中的运行状况检查或保持连接查询的频率，来调整运行状况检查频率。
- 通过配置连接池来优化闲置系统设置，以最大限度地减少系统闲置期间不必要的连接流失或后台查询活动。
- 实施应用程序级池化或改进的连接生命周期管理（如果可以减少开销）。
- 禁用心跳或验证查询（如果连接池配置支持这样做）。检查特定的连接字符串参数或配置文件以调整这些设置。
- 微调 TCP 保持连接设置：如果无法禁用驱动程序的内部心跳机制，请在操作系统或应用程序级别调整传输控制协议（TCP）保持连接设置，以解决连接超时问题。有关详细信息，请参阅操作系统、JDBC/ODBC 驱动程序或连接池文档。
- 优化数据库连接池：配置连接池（HikariCP、Apache 数据库连接池）以管理连接并最大限度地减少连接开销。重点关注诸如最大连接数、空闲超时和验证查询等参数（如果需要）。这种优化有助于使 Amazon Redshift Serverless 计算用量与实际工作负载需求保持一致，从而可能降低成本。

## 使用零 ETL 的 Amazon Redshift Serverless 的成本优化

在 Amazon Redshift Serverless 上运行零 ETL 集成时，要想优化成本，您可以根据工作负载的需求，合理调整环境大小并调整刷新设置。请考虑进行以下调整：

- 在可用于工作负载时，使用 8 RPU 的较低基本 RPU 容量。
- 配置目标 Redshift 实例的 REFRESH_INTERVAL，在新鲜度与成本之间取得平衡。较短的时间间隔可确保近乎实时的更新，但会增加计算成本。对于不太需要即时新鲜度的工作负载（例如报告或历史分析），较长的间隔（5 分钟或更长）可减少费用。要编辑 Redshift 目标 REFRESH_INTERVAL，请参阅 ALTER DATABASE 描述中有关刷新间隔的子句。
- 在摄取零 ETL 数据时，通过并行运行分析工作负载，可尽量提高 Amazon Redshift Serverless 环境的利用率。这可确保将计算容量有效地用于多种业务目的。

# 连接到 Amazon Redshift Serverless

设置好 Amazon Redshift Serverless 实例后，您可以使用下面概述的各种方法连接到该实例。如果您有多个团队或项目并且想单独管理成本，则可以单独使用 AWS 账户。

有关提供了 Amazon Redshift Serverless 的 AWS 区域 的列表，请参阅《Amazon Web Services 一般参考》中针对 Redshift Serverless API 列出的端点。

Amazon Redshift Serverless 连接到当前 AWS 区域中您的 AWS 账户中的无服务器环境。Amazon Redshift Serverless 在 VPC 中运行，端口范围为 5431-5455 和 8191-8215。默认值为 5439。目前，您只能通过 API 操作 `UpdateWorkgroup` 和 AWS CLI 操作 `update-workgroup` 更改端口。

## 连接到 Amazon Redshift Serverless

您可以使用以下语法，连接到 Amazon Redshift Serverless 中的数据库（名为 `dev`）。

```
workgroup-name.account-number.aws-region.redshift-serverless.amazonaws.com:port/dev
```

例如，以下连接字符串指定区域 us-east-1。

```
default.123456789012.us-east-1.redshift-serverless.amazonaws.com:5439/dev
```

## 通过 JDBC 驱动程序连接到 Amazon Redshift Serverless

通过 Amazon Redshift 提供的 JDBC 驱动程序版本 2.x 驱动程序，您可使用首选 SQL 客户端，通过下列方法之一连接到 Amazon Redshift Serverless。

要使用 JDBC 驱动程序版本 2.x 通过登录凭证进行连接来执行数据库身份验证，请使用以下语法。端口号是可选的；如果未包括在内，Amazon Redshift Serverless 默认使用端口号 5439。您可以更改为 5431-5455 或 8191-8215 端口范围内的另一个端口。要更改无服务器端点的默认端口，请使用 AWS CLI 和 Amazon Redshift API。

```
jdbc:redshift://workgroup-name.account-number.aws-region.redshift-
serverless.amazonaws.com:5439/dev
```

例如，以下连接字符串指定默认工作组、账户 ID 123456789012 和区域 us-east-2。

```
jdbc:redshift://default.123456789012.us-east-2.redshift-serverless.amazonaws.com:5439/
dev
```

要使用 JDBC 驱动程序版本 2.x 通过 IAM 进行连接，请使用以下语法。端口号是可选的；如果未包括在内，Amazon Redshift Serverless 默认使用端口号 5439。您可以更改为 5431-5455 或 8191-8215 端口范围内的另一个端口。要更改无服务器端点的默认端口，请使用 AWS CLI 和 Amazon Redshift API。

```
jdbc:redshift:iam://workgroup-name.account-number.aws-region.redshift-
serverless.amazonaws.com:5439/dev
```

例如，以下连接字符串指定默认工作组、账户 ID 123456789012 和区域 us-east-2。

```
jdbc:redshift:iam://default.123456789012.us-east-2.redshift-
serverless.amazonaws.com:5439/dev
```

对于 ODBC，使用以下语法。

```
Driver={Amazon Redshift (x64)}; Server=workgroup-name.account-number.aws-
region.redshift-serverless.amazonaws.com; Database=dev
```

如果您使用 2.1.0.9 之前的 JDBC 驱动程序版本并与 IAM 连接，则需要使用以下语法。

```
jdbc:redshift:iam://redshift-serverless-<name>:aws-region/database-name
```

例如，以下连接字符串指定默认工作组和 AWS 区域 us-east-1。

```
jdbc:redshift:iam://redshift-serverless-default:us-east-1/dev
```

有关驱动程序的更多信息，请参阅在 Amazon Redshift 中配置连接。

### 查找您的 JDBC 和 ODBC 连接字符串

要使用 SQL 客户端工具连接到您的工作组，您必须具有 JDBC 或 ODBC 连接字符串。您可以在 Amazon Redshift Serverless 控制台中的工作组详细信息页面上查找连接字符串。

查找工作组的连接字符串

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Redshift Serverless。
3. 在导航菜单上，选择工作组配置，然后从列表中选择集群名称以打开其详细信息。
4. 一般信息部分中提供有 JDBC URL 和 ODBC URL 连接字符串以及其他详细信息。每个字符串均基于运行工作组的 AWS 区域。选择相应连接字符串旁边的图标来复制连接字符串。

## 使用数据 API 连接到 Amazon Redshift Serverless

还可使用 Amazon Redshift 数据 API 连接到 Amazon Redshift Serverless。在 AWS CLI 调用中使用 `workgroup-name` 参数而不是 `cluster-identifier` 参数。

有关数据 API 的更多信息，请参阅使用 Amazon Redshift 数据 API。有关在 Python 中调用数据 API 的示例代码以及其他示例，请参阅 GitHub 中的 Redshift 数据 API 入门，查看其中的 `quick-start` 和 `use-cases` 文件夹。

# 使用 SSL 连接到 Amazon Redshift Serverless

## 配置与 Amazon Redshift Serverless 的安全连接

为支持 SSL 连接，Redshift Serverless 会为每个工作组创建并安装 AWS Certificate Manager（ACM）颁发的 SSL 证书。ACM 证书受到大多数操作系统、Web 浏览器和客户端的公开信任。如果您的 SQL 客户端或应用程序使用 SSL 连接到 Redshift Serverless，并且 `sslmode` 连接选项设置为 `require`、`verify-ca` 或 `verify-full`，则您可能需要下载证书捆绑包。如果客户需要，Redshift Serverless 会提供如下捆绑证书：

- 捆绑包下载地址：https://s3.amazonaws.com/redshift-downloads/amazon-trust-ca-bundle.crt。
  - 预期的 MD5 校验码为 418dea9b6d5d5d5de7a8f1ac42e164cdcf。
  - sha256 校验码为 36dba8e4b8041cd14b9d601593963301bcbb92e1c456847784de2acb5bd550。

  不要使用位于 `https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt` 的之前的证书捆绑包。
- 在中国 AWS 区域，请从以下网址下载捆绑包：https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/amazon-trust-ca-bundle.crt。
  - 预期的 MD5 校验码为 418dea9b6d5d5d5de7a8f1ac42e164cdcf。
  - sha256 校验码是 36dba8e4b8041cd14b9d601593963301bcbb92e1c456847784de2acb5bd550。

  不要使用位于 `https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/redshift-ca-bundle.crt` 和 `https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/redshift-ssl-ca-cert.pem` 的以前的证书捆绑包

> **Important**
>
> Redshift Serverless 改变了我们管理 SSL 证书的方法。您可能需要更新当前的信任根 CA 证书，才能继续使用 SSL 连接到工作组。有关 SSL 连接的 ACM 证书的更多信息，请参阅将 SSL 连接过渡到 ACM 证书。

默认情况下，无论连接是否使用 SSL，工作组数据库都会接受该连接。

要创建仅接受 SSL 连接的新工作组，请使用 `create-workgroup` 命令并将 `require_ssl` 参数设置为 `true`。要使用以下示例，请将 *yourNamespaceName* 替换为命名空间的名称，然后将 *yourWorkgroupName* 替换为工作组的名称。

```
aws redshift-serverless create-workgroup \
--namespace-name yourNamespaceName \
--workgroup-name yourWorkgroupName \
--config-parameters parameterKey=require_ssl,parameterValue=true
```

要将现有工作组更新为仅接受 SSL 连接，请使用 `update-workgroup` 命令并将 `require_ssl` 参数设置为 `true`。请注意，当您更新 `require_ssl` 参数时，Redshift Serverless 将重启工作组。要使用以下示例，请将 *yourWorkgroupName* 替换为工作组的名称。

```
aws redshift-serverless update-workgroup \
--workgroup-name yourWorkgroupName \
--config-parameters parameterKey=require_ssl,parameterValue=true
```

Amazon Redshift 支持 Elliptic Curve Diffie—Hellman Ephemeral (ECDHE) 密钥协商协议。ECDHE 使得客户端和服务器各有一个椭圆曲线公有/私有密钥对，用来在不安全的通道上创建共享密钥。您无需在 Amazon Redshift 中进行任何配置即可启用 ECDHE。如果您从使用 ECDHE 来加密客户端与服务器间通信的 SQL 客户端工具进行连接，则 Amazon Redshift 将使用提供的密码列表来建立合适的连接。有关更多信息，请参阅 Wikipedia 上的 Elliptic curve diffie—hellman 和 OpenSSL 网站上的密码。

配置与 Amazon Redshift Serverless 的符合 FIPS 的 SSL 连接

要创建使用符合 FIPS 的 SSL 连接的新工作组，请使用 `create-workgroup` 命令并将 `use_fips_ssl` 和 `require_ssl` 参数设置为 `true`。要使用以下示例，请将 *yourNamespaceName* 替换为命名空间的名称，然后将 *yourWorkgroupName* 替换为工作组的名称。

```
aws redshift-serverless create-workgroup \
--namespace-name yourNamespaceName \
--workgroup-name yourWorkgroupName \
--config-parameters '[{"parameterKey": "require_ssl", "parameterValue": "true"},
 {"parameterKey": "use_fips_ssl", "parameterValue": "true"}]'
```

要将现有工作组更新为使用符合 FIPS 的 SSL 连接，请使用 `update-workgroup` 命令并将 `use_fips_ssl` 和 `require_ssl` 参数设置为 `true`。请注意，当您更新 `use_fips_ssl` 参数

时，Redshift Serverless 将重启工作组。要使用以下示例，请将 *yourWorkgroupName* 替换为工作组的名称。

```
aws redshift-serverless update-workgroup \
--workgroup-name yourWorkgroupName \
--config-parameters '[{"parameterKey": "require_ssl", "parameterValue": "true"},
 {"parameterKey": "use_fips_ssl", "parameterValue": "true"}]'
```

有关将 Redshift Serverless 配置为使用符合 FIPS 的连接的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 use_fips_ssl。

## 从 Amazon Redshift 托管式 VPC 端点连接到 Amazon Redshift Serverless

### 从其他 VPC 端点连接到 Amazon Redshift Serverless

有关为 Amazon Redshift Serverless 工作组设置或配置托管 VPC 端点的信息，请参阅使用 Redshift 托管的 VPC 端点。

## 从接口 VPC 端点连接到 Amazon Redshift Serverless（AWS PrivateLink）

有关从接口 VPC 端点连接到 Amazon Redshift Serverless（AWS PrivateLink）的信息，请参阅接口 VPC 端点。

## 从其它账户中的 Redshift VPC 端点连接到 Amazon Redshift Serverless

### 从其他 VPC 端点连接到 Amazon Redshift Serverless

Amazon Redshift Serverless 在 VPC 中预置。您可以向其他账户中的 VPC 授予访问权限，以便访问您账户中的 Amazon Redshift Serverless。这与从托管 VPC 端点进行连接类似，但在本例中，连接源自其他账户中的数据库客户端。您可以执行几项操作：

- 数据库所有者可将对包含 Amazon Redshift Serverless 的 VPC 的访问权限，授予同一区域的其他账户。
- 数据库所有者可以撤销 Amazon Redshift Serverless 访问权限。

跨账户存取的主要好处是可以更轻松地进行数据库协作。用户无需在包含数据库的账户中进行预置即可访问该数据库，从而减少配置步骤并节省时间。

向其他账户中的 VPC 授予访问权所需的权限

要授予访问权限或更改所允许的访问权限，授予者需要具有以下权限的已分配权限策略：

- redshift-serverless:PutResourcePolicy
- redshift-serverless:GetResourcePolicy
- redshift-serverless:DeleteResourcePolicy
- ec2:CreateVpcEndpoint
- ec2:ModifyVpcEndpoint

您可能需要在 AWS 托管式策略 AmazonRedshiftFullAccess 中指定的其他权限。有关更多信息，请参阅向 Amazon Redshift Serverless 授予权限。

被授权者需要具有以下权限的已分配权限策略：

- redshift-serverless:ListWorkgroups
- redshift-serverless:CreateEndpointAccess
- redshift-serverless:UpdateEndpointAccess
- redshift-serverless:GetEndpointAccess
- redshift-serverless:ListEndpointAccess
- redshift-serverless:DeleteEndpointAccess

作为最佳实践，我们建议将权限策略附加到 IAM 角色，然后根据需要将其分配给用户和组。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

以下是用于配置跨 VPC 访问的资源策略示例：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "CrossAccountCrossVPCAccess",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
```

```
                "123456789012",
                "234567890123"
            ]
        },
        "Action": [
            "redshift-serverless:CreateEndpointAccess",
            "redshift-serverless:UpdateEndpointAccess",
            "redshift-serverless:DeleteEndpointAccess",
            "redshift-serverless:GetEndpointAccess"
        ],
        "Resource": "arn:aws:redshift-serverless:*:*:*",
        "Condition": {
            "StringEquals": {
                "aws:RequestedRegion": "us-east-1"
            },
            "StringLike": {
                "aws:SourceVpc": [
                   "vpc-11223344556677889",
                   "vpc-12345678"
                ]
            }
        }
    }
]
}
```

这一部分中后面的步骤假定执行这些操作的用户已分配了相应的权限，例如，通过分配的具有所列权限的 IAM 角色。这些过程还假定工作组上附加了具有相应资源权限的 IAM 角色。

使用控制台向其他账户授予 VPC 访问权限

此过程演示了如果您是数据库所有者，当您想要授予数据库访问权限时，用于配置数据库访问权限的步骤。

从所有者账户授予访问权限

1. 在 Amazon Redshift Serverless 工作组的属性中，在数据访问权限选项卡上，有一个名为授权账户的列表。其中显示授予了对工作组访问权限的账户和 VPC。找到列表并选择授予访问权限，将账户添加到列表中。
2. 此时将出现一个窗口，您可以在其中添加被授权者信息。输入 AWS 账户 ID，这是您要向其授予访问权限的账户的 12 位数 ID。

3. 向被授权者授予对所有 VPC 或特定 VPC 的访问权限。如果您仅向特定 VPC 授予访问权限，则可以输入这些 VPC 的 ID 并选择添加 VPC 来进行添加。
4. 完成后，请保存更改。

保存更改后，该账户将显示在授权账户列表中。该条目显示授予了访问权限的 账户 ID和 VPC 列表。

数据库所有者也可以撤消账户的访问权限。所有者可以随时撤销这些访问权限。

撤消账户的访问权限

1. 您可以从授权账户列表开始操作。首先，选择一个或多个账户。
2. 然后选择撤消访问权限。

授予访问权限后，被授权者的数据库管理员可以查看控制台以确定自身是否有访问权限。

使用控制台确认已授予您访问其他账户的访问权限

1. 在 Amazon Redshift Serverless 工作组属性中，在数据访问权限选项卡上，有一个名为已授权账户的列表。它显示了可以从此工作组访问的账户。被授权者不能使用工作组的端点 URL 直接访问工作组。要访问工作组，作为被授权者，您需要转到端点部分，然后选择创建端点。
2. 然后，作为被授权者，您需要提供端点名称和用于访问工作组的 VPC。
3. 成功创建端点后，它会出现在端点部分，并且会有一个端点 URL。您可以使用此端点 URL 访问工作组。

使用 CLI 命令向其他账户授予访问权限

授予访问权限的账户必须先向另一个账户授予访问权限，然后才能使用 `put-resource-policy` 进行连接。数据库所有者可以调用 `put-resource-policy` 来授权其他账户创建与工作组的连接。然后，被授权者账户可以使用 `create-endpoint-authorization`，通过其允许的 VPC 创建与工作组的连接。

以下显示 `put-resource-policy` 的属性，您可以进行调用来允许访问特定账户和 VPC。

```
aws redshift-serverless put-resource-policy
--resource-arn <value>
--policy <value>
```

调用命令后，您可以调用 get-resource-policy，指定 resource-arn 以查看允许了哪些账户和 VPC 访问资源。

被授权者可以进行以下调用。它显示有关所授予访问权限的信息。具体而言，它会返回一个列表，其中包含授予了访问权限的 VPC。

```
aws redshift-serverless list-workgroups
--owner-account <value>
```

此信息用于让被授权者从授权账户获取有关终端节点授权的信息。owner-account 是共享账户。当您运行它时，它会返回每个工作组的 CrossAccountVpcs，即允许的 VPC 的列表。作为参考，以下内容显示了工作组上提供的所有属性：

```
Output: workgroup (Object)
workgroupId String,
workgroupArn String,
workgroupName String,
status: String,
namespaceName: String,
baseCapacity: Integer, (Not-applicable)
enhancedVpcRouting: Boolean,
configParameters: List,
securityGroupIds: List,
subnetIds: List,
endpoint: String,
publiclyAccessible: Boolean,
creationDate: Timestamp,
port: Integer,
CrossAccountVpcs: List
```

> **Note**
>
> 提醒一下，集群重新定位不是配置其它 Redshift 联网特征的先决条件。在启用以下功能时，此功能也并非必需：
>
> - 从跨账户或跨区域 VPC 连接到 Redshift – 您可以从一个 AWS Virtual Private Cloud (VPC) 连接到另一个包含 Redshift 数据库的 Virtual Private Cloud (VPC)，如此部分中所述。
> - 设置自定义域名 – 您可以为 Amazon Redshift 集群或 Amazon Redshift Serverless 工作组创建自定义域名，也称为自定义 URL，以提供更简单也更容易记住的端点名称。有关更多信息，请参阅使用自定义域名进行客户端连接。

## 其他资源

有关网络流量设置的说明，请参阅使用默认或自定义安全组配置实现公开访问功能。这包括集群可公开访问时的使用案例。

有关网络流量设置的说明，请参阅使用默认或自定义安全组配置实现私有访问功能。这包括集群无法在互联网上使用时的使用案例。

有关与 Amazon Redshift Serverless 的安全连接的更多信息，包括授予权限、授权访问其他服务以及创建 IAM 角色，请参阅 Amazon Redshift Serverless 中的 Identity and Access Management。

# 在 Amazon Redshift Serverless 中定义向联合用户授予的数据库角色

当您属于某个组织时，您会具有一系列关联的角色。例如，您的工作职能有不同的角色，可能是程序员和经理。您的角色决定了您可以访问哪些应用程序和数据。大多数组织使用身份提供者（例如 Microsoft Active Directory）向用户和组分配角色。组织越来越多地使用角色来控制对资源访问权限，因为这样就不必过多地管理单个用户。

最近，Amazon Redshift Serverless 中引入了基于角色的访问控制。例如，使用数据库角色，您可以安全访问架构或表等数据和对象。或者，您可以使用角色来定义一组提升的权限，例如用于系统监控或数据库管理员的权限。但是，在向数据库角色授予资源权限后，还有一个额外步骤，即将组织的用户角色连接到数据库角色。您可以通过运行 SQL 语句，向每个用户在初次登录时分配其数据库角色，但这需要完成大量的工作。一种更简单的方法是定义要授予的数据库角色，然后将这些角色传递给 Amazon Redshift Serverless。这样做的好处是简化初始登录流程。

您可以使用 `GetCredentials` 将角色传递给 Amazon Redshift Serverless。当用户首次登录 Amazon Redshift Serverless 数据库时，会创建关联的数据库用户并将其映射到匹配的数据库角色。本主题详细介绍了将角色传递给 Amazon Redshift Serverless 的机制。

传递数据库角色主要用于几个使用场景：

- 当用户通过第三方身份提供者（通常配置了联合身份验证）登录并通过会话标签传递角色时。
- 当用户通过 IAM 登录凭证登录时，他们的角色通过标签键和值传递。

有关基于角色的访问控制的更多信息，请参阅基于角色的访问控制 (RBAC)。

# 定义数据库角色

在将角色传递给 Amazon Redshift Serverless 之前，您必须在数据库中配置数据库角色，并向这些角色授予对数据库资源的相应权限。例如，在一个简单的场景中，您创建了一个名为 sales 的数据库角色，并授予其访问权限来查询包含销售数据的表。有关如何创建数据库角色和授予权限的更多信息，请参阅 CREATE ROLE 和 GRANT。

# 定义授予联合用户的数据库角色的使用场景

这些部分概述了几个使用场景，在这些使用场景中，将数据库角色传递给 Amazon Redshift Serverless 可以简化对数据库资源的访问。

## 使用身份提供者登录

第一个用使用场景设您的组织在某个身份和访问管理服务中具有用户身份。此服务可以为云端，例如 JumpCloud 或 Okta，也可以是本地服务，例如 Microsoft Active Directory。此时的目标是在用户使用 JDBC 等客户端登录查询编辑器 V2 这样的客户端时，自动将用户的角色从身份提供者映射到您的数据库角色。要进行此设置，您必须完成一些配置任务。这些功能包括：

1. 使用信任关系配置与身份提供者 (IdP) 的联合身份验证集成。这是先决条件。当您完成了此设置时，身份提供者负责通过 SAML 断言对用户进行身份验证并提供登录凭证。有关更多信息，请参阅将第三方 SAML 解决方案提供商与 AWS 集成。您还可以在使用 Active Directory Federation Services (AD FS) 对 Amazon Redshift 查询编辑器 V2 进行联合身份访问或者使用 Okta 对 Amazon Redshift 查询编辑器 v2 进行联合身份单点登录访问。
2. 角色还必须具有以下策略权限：
   - `GetCredentials` – 提供凭证用于临时授权登录 Amazon Redshift Serverless。
   - `sts:AssumeRoleWithSAML` – 提供一种机制，用于将企业身份存储或目录绑定到基于角色的 AWS 访问。
   - `sts:TagSession` – 在身份提供者主体上，对标记会话执行操作的权限。

   在此例中，对于已经通过 SAML 身份验证响应进行了身份验证的用户，`AssumeRoleWithSAML` 返回一组安全凭证。此操作提供了一种机制，用于将身份存储或目录绑定到基于角色的 AWS 访问，而无需用户特定的凭证。对于具有 `AssumeRoleWithSAML` 权限的用户，身份提供者负责管理用于传递角色信息的 SAML 断言。

   作为最佳实践，我们建议将权限策略附加到 IAM 角色，然后根据需要将其分配给用户和组。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

3. 您可以使用冒号分隔的角色值配置标签 RedshiftDbRoles，格式为 role1:role2。例如 manager:engineer。这些值可以从身份提供者中配置的会话标签实施来检索。SAML 身份验证请求以编程方式传递角色。有关传递会话标签的更多信息，请参阅在 AWS STS 中传递会话标签。

   如果您传递的角色名在数据库中不存在，则忽略该名称。

在此使用场景中，当用户使用联合身份登录时，他们的角色将通过会话标签键和值在授权请求中传递。随后，在授权后，GetCredentials 会将角色传递给数据库。成功连接后，系统将映射数据库角色，然后用户就可以执行与其角色对应的数据库任务。该操作的关键部分是在初始授权请求中，向角色授予了 RedshiftDbRoles 会话标签。有关传递会话标签的更多信息，请参阅 使用 AssumeRoleWithSAML 传递会话标签。

### 使用 IAM 凭证登录

在第二个使用场景中，可以为用户传递角色，他们可以通过 IAM 凭证访问数据库客户端应用程序。

1. 对于在这种情况下登录的用户，必须向其分配对以下操作的策略权限：
   - tag:GetResources – 返回与指定标签关联的带标签资源。
   - tag:GetTagKeys – 返回当前正在使用的标签键。

     作为最佳实践，我们建议将权限策略附加到 IAM 角色，然后根据需要将其分配给用户和组。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。
2. 允许权限还需要访问数据库服务，例如 Amazon Redshift Serverless。
3. 对于此使用场景，请在 AWS Identity and Access Management 中为您的角色配置标签值。您可以选择编辑标签，然后创建名为 RedshiftDbRoles 的标签键，以及附带的包含角色的标签值字符串。例如，manager:engineer。

当用户登录时，他们的角色会被添加到授权请求中并传递到数据库。其角色映射到现有的数据库角色。

## 其他 资源

如使用场景中所述，您可以配置 IdP 与 AWS 之间的信任关系。有关详细信息，请参阅使用信赖方信任配置您的 SAML 2.0 IdP 并添加声明。

# Amazon Redshift Serverless 中的 Identity and Access Management

访问 Amazon Redshift 时需要可供 AWS 用来验证您的请求的凭证。这些凭证必须有权访问 AWS 资源，如 Amazon Redshift Serverless。

以下部分提供详细信息来说明如何使用 AWS Identity and Access Management（IAM）和 Amazon Redshift 控制谁能访问您的资源，从而对这些资源进行保护：有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

## 向 Amazon Redshift Serverless 授予权限

要访问其它 AWS 服务，Amazon Redshift Serverless 需要权限。某些 Amazon Redshift 功能要求 Amazon Redshift 代表您访问其他 AWS 服务。为了让您的 Amazon Redshift Serverless 实例为您执行操作，请向它提供安全凭证。提供安全凭证的首选方法是指定一个 AWS Identity and Access Management (IAM) 角色。您还可通过 Amazon Redshift 控制台创建 IAM 角色，并将其设置为默认角色。有关更多信息，请参阅 创建一个 IAM 角色作为 Amazon Redshift 的默认角色。

要访问其它 AWS 服务，请创建具有适当权限的 IAM 角色。您还需要将角色与您的 Amazon Redshift Serverless 关联。此外，您可在运行 Amazon Redshift 命令时指定角色的 Amazon 资源名称（ARN），或者指定 `default` 关键字。

在 https://console.aws.amazon.com/iam/ 中更改 IAM 角色的信任关系时，请确保将 `redshift-serverless.amazonaws.com` 和 `redshift.amazonaws.com` 作为主体服务名称。有关如何管理 IAM 角色以代表您访问其他 AWS 服务的信息，请参阅授权 Amazon Redshift 代表您访问 AWS 服务。

### 创建一个 IAM 角色作为 Amazon Redshift 的默认角色

当您通过 Amazon Redshift 控制台创建 IAM 角色时，Amazon Redshift 以编程方式在您的 AWS 账户中创建角色。Amazon Redshift 还会自动为其附加现有 AWS 托管式策略。这种方法表示您可以停留在 Amazon Redshift 控制台中，无需切换到 IAM 控制台进行角色创建。

您通过控制台为集群创建的 IAM 角色具有自动附加的 `AmazonRedshiftAllCommandsFullAccess` 托管式策略。此 IAM 角色允许 Amazon Redshift 为 AWS IAM 账户中的资源复制、卸载、查询和分析数据。相关命令包括 UNLOAD、CREATE EXTERNAL FUNCTION、CREATE EXTERNAL TABLE、CREATE EXTERNAL SCHEMA、CREATE MODEL 和 CREATE LIBRARY。有关如何创建 IAM 角色作为 Amazon Redshift 的默认角色的更多信息，请参阅创建一个 IAM 角色作为 Amazon Redshift 的默认角色。

要开始创建 IAM 角色作为 Amazon Redshift 的默认角色，请打开 AWS 管理控制台，选择 Amazon Redshift 控制台，然后在菜单中选择 Redshift Serverless。在无服务器控制面板中，您可以创建新的工作组。创建步骤将引导您选择 IAM 角色或配置新的 IAM 角色。

当您已经拥有 Amazon Redshift Serverless 工作组并希望为它配置 IAM 角色时，请打开 AWS 管理控制台。选择 Amazon Redshift 控制台，然后选择 Redshift Serverless。在 Amazon Redshift Serverless 控制台上，为现有工作组选择命名空间配置。在安全和加密下，您可以编辑权限。

向命名空间分配 IAM 角色

每个 IAM 角色都是一个 AWS 身份，此身份的权限策略可确定每个角色可以在 AWS 中执行哪些操作。该角色旨在让任何需要它的用户代入。此外，每个命名空间都是对象（如表和架构）和用户的集合。当您使用 Amazon Redshift Serverless 时，您可以将多个 IAM 角色与命名空间关联。这样可以更轻松地为数据库对象集合适当地构建权限，以便角色可以对内部和外部数据执行操作。例如，您可以在 Amazon Redshift 数据库中运行 `COPY` 命令，以检索 Amazon S3 中的数据并填充 Redshift 表。

您可以使用控制台将多个角色与命名空间关联，如本节前面所述。您还可以使用 `CreateNamespace` API 命令或 CLI 命令 `create-namespace`。使用 API 或 CLI 命令，您可以通过用一个或多个角色填充 `IAMRoles`，以将 IAM 角色分配给命名空间。具体来说，您可以将特定角色的 ARN 添加到集合中。

管理命名空间相关 IAM 角色

在 AWS 管理控制台上，可以管理 AWS Identity and Access Management 中角色的权限策略。您可以使用命名空间配置下提供的设置管理命名空间的 IAM 角色。有关命名空间及其在 Amazon Redshift Serverless 中的使用情况的更多信息，请参阅工作组和命名空间。

## Amazon Redshift 的 IAM 凭证入门

当您首次登录到 Amazon Redshift 控制台，并首次试用 Amazon Redshift Serverless 时，我们建议您以附加了具有所需策略的 IAM 角色的用户身份登录。开始创建 Amazon Redshift Serverless 实例后，Amazon Redshift 会记录您用于登录的 IAM 角色名称。您可以使用相同的凭证登录到 Amazon Redshift 控制台和 Amazon Redshift Serverless 控制台。

创建 Amazon Redshift Serverless 实例时，您可以创建数据库。使用查询编辑器 v2 通过临时凭证选项连接到数据库。

要添加持续用于数据库的新管理员用户名和密码，请选择自定义管理员用户凭证，然后输入新的管理员用户名和管理员用户密码。

要开始使用 Amazon Redshift Serverless 并首次在控制台中创建工作组和命名空间，请使用附加了权限策略的 IAM 角色。确保此角色具有管理员权限 `arn:aws:iam::aws:policy/AdministratorAccess`，或具有附加到 IAM 策略的完整 Amazon Redshift 权限 `arn:aws:iam::aws:policy/AmazonRedshiftFullAccess`。

以下场景概述了当您开始使用 Amazon Redshift Serverless 控制台时，Amazon Redshift Serverless 如何使用 IAM 凭证：

- 如果选择使用默认设置 – Amazon Redshift Serverless 会将您当前的 IAM 身份转换为数据库超级用户。您可以将相同的 IAM 身份与 Amazon Redshift Serverless 控制台一起使用，以在 Amazon Redshift Serverless 的数据库中执行超级用户操作。
- 如果选择自定义设置而不指定 Amazon Redshift Serverless 的管理员用户名和密码，则您当前的 IAM 凭证用作默认管理员用户凭证。
- 如果选择自定义设置，然后指定 Amazon Redshift Serverless 的管理员用户名和密码 – Amazon Redshift Serverless 会将您当前的 IAM 身份转换为数据库超级用户。Amazon Redshift Serverless 还会以超级用户身份创建另一个长期登录用户名和密码对。您可使用当前的 IAM 身份或创建的用户名和密码对以超级用户身份登录到数据库。

## 使用数据库角色权限访问 Amazon Redshift Serverless 数据库对象

此过程说明如何通过 Amazon Redshift 数据库角色授予查询表的权限。通过标签分配角色，该标签在 IAM 中附加到用户，并在用户登录时传递给 Amazon Redshift。这是以在 Amazon Redshift Serverless 中定义向联合用户授予的数据库角色中的概念举例的说明。完成这些步骤的好处是，您可以将用户与数据库角色相关联，并避免为每个数据库对象设置权限。它简化了对用户查询、修改数据或向表中添加数据以及执行其他操作的能力的管理。

该过程假设您已经设置了 Amazon Redshift Serverless 数据库，并且能够在该数据库中授予权限。它还假设您有权在 AWS 控制台中创建 IAM 用户，以及创建 IAM 角色和分配策略权限。

1. 使用 IAM 控制台创建 IAM 用户。稍后，您将使用此用户连接到数据库。
2. 使用查询编辑器 v2 或其他 SQL 客户端，创建 Redshift 数据库角色。有关创建数据库角色的更多信息，请参阅创建 CREATE ROLE。

   ```
   CREATE ROLE urban_planning;
   ```

   查询 SVV_ROLES 系统视图，以检查您的角色是否已创建。它还会返回系统角色。

```
SELECT * from SVV_ROLES;
```

3. 授予您创建的数据库角色从表中进行选择的权限。（您创建的 IAM 用户最后将登录并通过数据库角色从表中选择记录。）以下代码示例中的角色名和表名是示例。在此，授予了从名为 cities 的表中进行选择的权限。

```
GRANT SELECT on TABLE cities to ROLE urban_planning;
```

4. 使用 AWS Identity and Access Management 控制台创建 IAM 角色。此角色授予使用查询编辑器 v2 的权限。创建新的 IAM 角色，对于可信实体类型，选择 AWS 账户。然后选择此账户。为角色授予以下策略权限：

   - AmazonRedshiftReadOnlyAccess
   - tag:GetResources
   - tag:GetTagKeys
   - sqlworkbench 的所有操作，包括 sqlworkbench:ListDatabases 和 sqlworkbench:UpdateConnection。

5. 在 IAM 控制台中，将一个带有键 RedshiftDbRoles 的标签添加到您之前创建的 IAM 用户。标签的值应与您在第一步中创建的数据库角色相匹配。它在此示例中为 urban_planning。

完成这些步骤后，将 IAM 角色分配给您在 IAM 控制台中创建的用户。当用户使用查询编辑器 v2 登录数据库时，用户在标签中的数据库角色名称将传递给 Amazon Redshift 并与用户关联。因此，用户可以通过数据库角色查询相应的表。举例来说，此示例中的用户可以通过 urban_planning 数据库角色查询 cities 表。

# 将预置集群迁移到 Amazon Redshift Serverless

可以将现有的预置集群迁移到 Amazon Redshift Serverless，从而实现按需和自动扩展计算资源。将预置集群迁移到 Amazon Redshift Serverless 后，您只需为使用的资源付费，并可根据工作负载需求自动扩展容量，从而优化成本。迁移的常见用例包括运行临时查询、定期数据处理任务，或在不过度预置资源的情况下处理不可预测的工作负载。执行以下一组任务，将预置 Amazon Redshift 集群迁移到无服务器部署选项。

# 创建预置集群的快照

> **Note**
>
> 在您将预置的集群快照还原到无服务器命名空间时，Amazon Redshift 会自动将交错键转换为复合键。

要将数据从您的预置集群传输到 Amazon Redshift Serverless，请为您的预置集群创建快照，然后在 Amazon Redshift Serverless 中还原该快照。

> **Note**
>
> 在将数据迁移到无服务器工作组之前，请确保您的预调配集群需求与您在 Amazon Redshift Serverless 中选择的 RPU 量兼容。

创建您的预置集群的快照

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群、快照，然后选择创建快照。
3. 输入快照定义的属性，然后选择创建快照。快照可能需要一段时间才可用。

将预置集群快照还原到无服务器命名空间：

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在 Amazon Redshift 预置集群控制台上开始，导航到集群、快照页面。
3. 选择要使用的快照。
4. 选择还原快照、还原到无服务器命名空间。
5. 选择要将您的快照还原到的命名空间。
6. 确认想要从快照还原。此操作将以来自预置集群的数据替换无服务器端点中的所有数据库。选择还原。

有关预置集群快照的更多信息，请参阅 Amazon Redshift 快照。

# 使用驱动程序连接到 Amazon Redshift Serverless

要使用首选的 SQL 客户端连接到 Amazon Redshift Serverless，可以使用 Amazon Redshift 提供的 JDBC 驱动程序版本 2.x。我们建议使用最新版本的 Amazon Redshift JDBC 驱动程序版本 2.x 连接到 Amazon Redshift。端口号是可选的。如果您未将端口号包括在内，则 Amazon Redshift Serverless 将默认使用端口号 5439。您可以更改为 5431-5455 或 8191-8215 端口范围内的另一个端口。要更改无服务器端点的默认端口，请使用 AWS CLI 和 Amazon Redshift API。

要查找用于 JDBC、ODBC 或 Python 驱动程序的确切端点，请参阅 Amazon Redshift Serverless 中的工作组配置。您还可以使用 Amazon Redshift Serverless API 操作 `GetWorkgroup` 或 AWS CLI 操作 `get-workgroups`，以返回有关您工作组的信息，然后进行连接。

## 使用基于密码的身份验证进行连接

要使用 Amazon Redshift JDBC 驱动程序版本 2.x 以及基于密码的身份验证来建立连接，请使用以下语法：

```
jdbc:redshift://<workgroup-name>.<account-number>.<aws-region>.redshift-
serverless.amazonaws.com:5439/?username=username&password=password
```

要使用 Amazon Redshift Python 连接器以及基于密码的身份验证来建立连接，请使用以下语法：

```
import redshift_connector
with redshift_connector.connect(
    host='<workgroup-name>.<account-number>.<aws-region>.redshift-
serverless.amazonaws.com',
    database='<database-name>',
    user='username',
    password='password'
    # port value of 5439 is specified by default
) as conn:
    pass
```

要使用 Amazon Redshift ODBC 驱动程序版本 2.x 以及基于密码的身份验证来建立连接，请使用以下语法：

```
Driver={Amazon Redshift ODBC Driver (x64)}; Server=<workgroup-name>.<account-
number>.<aws-region>.redshift-serverless.amazonaws.com; Database=database-name;
 User=username; Password=password
```

## 使用 IAM 进行连接

如果您更喜欢使用 IAM 进行登录，请使用 Amazon Redshift Serverless `GetCredentials` API 操作。

要使用 IAM 身份验证，请将 `iam:` 添加到 Amazon Redshift JDBC URL 中 `jdbc:redshift:` 的后面，如以下示例所示。

```
jdbc:redshift:iam://<workgroup-name>.<account-number>.<aws-region>.redshift-
serverless.amazonaws.com:5439/<database-name>
```

此 Amazon Redshift Serverless 端点不支持自定义 dbUser、dbGroup 或 auto-create。默认情况下，该驱动程序会在登录时自动创建数据库用户。然后，它会基于在 IAM 中指定的标签或基于在身份提供者（IdP）中定义的组，将用户分配给 Amazon Redshift 数据库角色。

请确保您的 AWS 身份具有适用于 `redshift-serverless:GetCredentials` 操作的正确 IAM 策略。下面是一个示例 IAM 策略，它授予对 AWS 身份的正确权限，用于连接到 Amazon Redshift Serverless。有关 IAM 权限的更多信息，请参阅《IAM 用户指南》中的添加和删除 IAM 身份权限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": "redshift-serverless:GetCredentials",
            "Resource": "*"
        }
    ]
}
```

要使用 Amazon Redshift Python 连接器以及基于 IAM 的身份验证来建立连接，请在代码中使用 `iam=true`，如以下语法所示：

```
import redshift_connector
with redshift_connector.connect(
    iam=True,
```

```
    host='<workgroup-name>.<account-number>.<aws-region>.redshift-
serverless.amazonaws.com',
    database='<database-name>'
    <IAM credentials>
) as conn:
    pass
```

对于 IAM credentials，可以使用任何凭证，包括以下各项：

- AWS 配置文件配置。
- IAM 凭证（访问密钥 ID、秘密访问密钥和可选的会话令牌）。
- 身份提供者联合身份验证。

要使用 Amazon Redshift ODBC 驱动程序版本 2.x 以及基于 IAM 的身份验证和配置文件来建立连接，请使用以下语法：

```
Driver={Amazon Redshift ODBC Driver (x64)}; IAM=true; Server=<workgroup-name>.<account-
number>.<aws-region>.redshift-serverless.amazonaws.com; Database=database-name;
 Profile=aws-profile-name;
```

### 将 IAM 和 GetClusterCredentials API 结合使用来进行连接

> **Note**
>
> 在连接到 Amazon Redshift Serverless 时，建议您使用 GetCredentials API。此 API 提供了全面的基于角色的访问权限控制（RBAC）功能以及 GetClusterCredentials 中未提供的其它新功能。我们支持使用 GetClusterCredentials API 来简化从预置集群到无服务器工作组的转换，但我们强烈建议尽快迁移到使用 GetCredentials，以获得最佳兼容性。

可以使用 GetClusterCredentials API 建立与 Amazon Redshift Serverless 的连接。要实现此身份验证方法，请通过纳入以下参数来修改客户端或应用程序：

- iam=true
- clusterid/cluster_identifier=redshift-serverless-<*workgroup-name*>
- region=<*aws-region*>

以下示例演示了所有三个驱动程序中的 BrowserSAML 插件。这代表几种可用的身份验证方法之一。可以根据具体要求修改示例，以使用其它身份验证方法或插件。

GetClusterCredentials 的 IAM 策略权限

以下是一个示例 IAM 策略，该策略具有将 `GetClusterCredentials` 与 Amazon Redshift Serverless 结合使用所需的权限：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "redshift-serverless:GetWorkgroup",
                "redshift-serverless:GetNamespace",
                "redshift:GetClusterCredentials",
                "redshift:CreateClusterUser",
                "redshift:JoinGroup",
                "redshift:ExecuteQuery",
                "redshift:FetchResults",
                "redshift:DescribeClusters",
                "redshift:DescribeTable"
            ],
            "Resource": [
                "arn:aws:redshift-serverless:us-
east-1:111122223333:workgroup/workgroup-name",
                "arn:aws:redshift-serverless:us-
east-1:111122223333:namespace/namespace-name",
                "arn:aws:redshift:us-east-1:111122223333:cluster:cluster-name",
                "arn:aws:redshift:us-east-1:111122223333:dbuser:database-name/
${redshift:DbUser}"
            ]
        }
    ]
}
```

要使用 Amazon Redshift JDBC 驱动程序版本 2.x 以及 GetClusterCredentials 来建立连接，请使用以下语法：

```
jdbc:redshift:iam://redshift-serverless-<workgroup-name>:<aws-region>/<database-name>?
plugin_name=com.amazon.redshift.plugin.BrowserSamlCredentialsProvider&login_url=<single
 sign-on URL from IdP>"
```

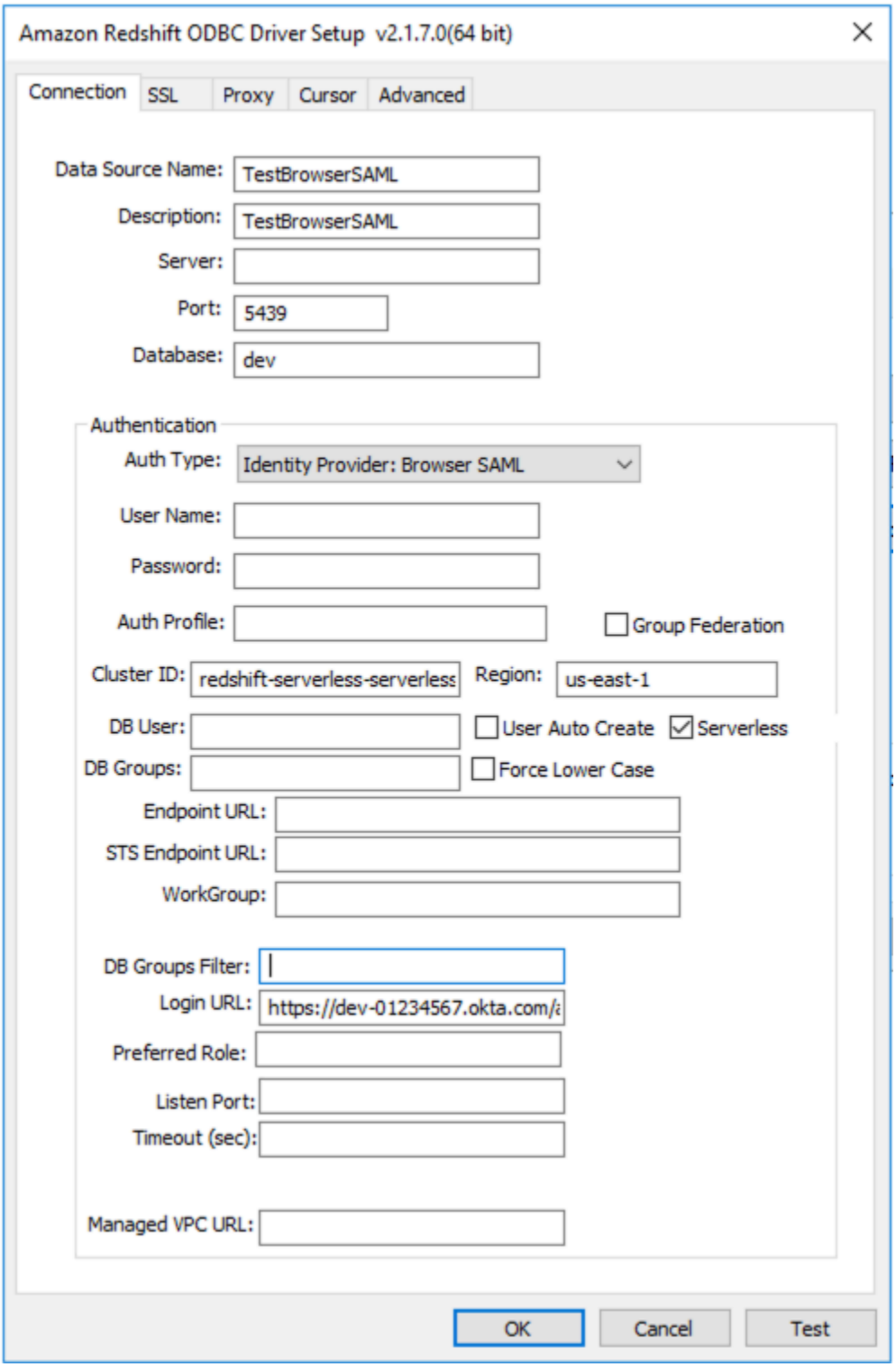
要使用 Amazon Redshift Python 连接器以及 GetClusterCredentials 来建立连接，请使用以下语法：

```
import redshift_connector
with redshift_connector.connect(
    iam=True,
    cluster_identifier='redshift-serverless-<workgroup-name>',
    region='<aws-region>',
    database='<database-name>',
    credentials_provider='BrowserSamlCredentialsProvider'
    login_url='<single sign-on URL from IdP>'
    # port value of 5439 is specified by default
) as conn:
    pass
```

要使用 Amazon Redshift ODBC 驱动程序版本 2.x 以及 GetClusterCredentials 来建立连接，请使用以下语法：

```
Driver= {Amazon Redshift ODBC Driver (x64)}; IAM=true; isServerless=true;
 ClusterId=redshift-serverless-<workgroup-name>; region=<aws-region>;
 plugin_name=BrowserSAML;login_url=<single sign-on URL from IdP>
```

以下是 Windows 中的 ODBC DSN 配置示例：


Amazon Redshift ODBC Driver Setup v2.1.7.0(64 bit)
Connection
SSL
Proxy
Cursor
Advanced
Data Source Name:
TestBrowserSAML
Description:
TestBrowserSAML
Server:
Port:
5439
Database:
dev
Authentication
Auth Type:
Identity Provider: Browser SAML
User Name:
Password:
Auth Profile:
Group Federation
Cluster ID:
redshift-serverless-serverless
Region:
us-east-1
DB User:
User Auto Create
Serverless
DB Groups:
Force Lower Case
Endpoint URL:
STS Endpoint URL:
WorkGroup:
DB Groups Filter:
Login URL:
https://dev-01234567.okta.com/
Preferred Role:
Listen Port:
Timeout (sec):
Managed VPC URL:
OK
Cancel
Test

## 使用 Amazon Redshift Serverless SDK

如果您使用 Amazon Redshift SDK 编写了任何管理脚本，则必须使用新的 Amazon Redshift Serverless SDK 来管理您的 Amazon Redshift Serverless 和关联资源。有关可用 API 操作的更多信息，请参阅《Amazon Redshift Serverless API 参考指南》。

# 工作组和命名空间

要在 Amazon Redshift Serverless 中隔离工作负载并管理不同的资源，您可以创建命名空间和工作组，并分别管理存储和计算资源。

命名空间是数据库对象和用户的集合。与存储相关的命名空间会将架构、表、用户或用于加密数据的 AWS Key Management Service 密钥组合在一起。存储属性包括管理员用户的数据库名称和密码、权限以及加密和安全性。按命名空间分组的其他资源包括数据共享、恢复点和使用限制。您可以使用 Amazon Redshift Serverless 控制台、AWS Command Line Interface 或 Amazon Redshift Serverless API 针对特定资源配置这些存储属性。

工作组是计算资源的集合。与计算相关的工作组将计算资源组合在一起，例如 RPU、VPC 子网组和安全组。工作组的属性包括网络和安全设置。按工作组分组的其他资源包括访问权限和使用限制。您可以使用 Amazon Redshift Serverless 控制台、AWS Command Line Interface 或 Amazon Redshift Serverless API 来配置这些计算属性。

您可以创建一个或多个命名空间和工作组。每个命名空间只能有一个工作组与其关联。反过来，每个工作组只能与一个命名空间关联。

## 使用控制台的工作组和命名空间

设置 Amazon Redshift Serverless 涉及完成数个配置步骤。当您按照这些步骤设置 Amazon Redshift Serverless 时，您将创建命名空间和工作组，然后将它们相互关联。要通过 Amazon Redshift Serverless 控制台开始设置 Amazon Redshift Serverless 配置，您可以选择 Amazon Redshift Serverless 入门，以设置 Amazon Redshift Serverless 并开始与之交互。可以选择具有默认设置的环境，这样可以更快地进行设置，也可以根据企业的要求明确配置设置。在此过程中，可以为工作组和命名空间指定设置。

设置环境之后，工作组属性和命名空间属性可帮助您熟悉这些设置。

# 使用 AWS Command Line Interface 和 Amazon Redshift Serverless API 的工作组和命名空间

除了使用 AWS 控制台，您还可以使用 AWS CLI 或 Amazon Redshift Serverless API 与工作组和命名空间进行交互。下表列出了您可用于管理快照和恢复点的 API 和 CLI 操作。

| API 操作 | CLI 命令 | 说明 |
| --- | --- | --- |
| CreateNamespace | create-namespace | 创建命名空间。默认设置下，Amazon Redshift Serverless 使用默认 AWS Key Management Service 密钥创建命名空间，但您可以指定其他密钥来加密数据。您也可以通过恢复快照来创建命名空间。有关更多信息，请参阅使用快照和恢复点。 |
| UpdateNamespace | update-namespace | 更新命名空间。 |
| GetNamespace | get-namespace | 检索有关命名空间的信息。 |
| ListNamepaces | list-namespaces | 检索有关命名空间列表的信息。 |
| DeleteNamespace | delete-namespace | 删除命名空间。 |
| CreateWorkgroup | create-workgroup | 创建工作组。创建工作组时，请确保您有一个可以与工作组关联的现有命名空间。在创建工作组时，您可以指定任意计算资源，例如子网、安全组或 RPU。 |
| UpdateWorkgroup | update-workgroup | 更新工作组。 |
| GetWorkgroup | get-workgroup | 检索有关工作组的信息。 |
| ListWorkgroups | list-workgroups | 检索有关工作组列表的信息。 |

| API 操作 | CLI 命令 | 说明 |
| --- | --- | --- |
| DeleteWorkgroup | delete-workgroup | 删除工作组。 |

# 工作组

使用 Amazon Redshift Serverless，可以创建和管理工作组，来隔离和控制不同工作负载或用户的计算资源。工作组支持您设置配置选项（例如内存和并发扩展限制），并确定跨工作负载执行查询的优先级。与计算相关的工作组将 RPU 和 VPC 子网组等计算资源组合在一起。

## 创建带有命名空间的工作组

要创建工作组，请完成以下步骤。有关工作组配置的更多信息，请参阅工作组属性。

1. 选择 Serverless 控制面板。然后选择创建工作组。
2. 输入工作组名称。
3. 选择工作组的 IP 地址类型。选项包括：
   - IPv4 – 使用此选项，AWS 资源只能通过 IPv4 寻址协议进行通信。
   - 双栈模式 – 使用此选项，您的 AWS 资源可通过 IPv4 和/或 IPv6 寻址协议进行通信。此外，您必须将 IPv6 CIDR 块与 Amazon VPC 中用于您的工作组的 VPC 和子网关联。您可以使用 Amazon VPC 控制台创建 Amazon VPC 或更新现有 Amazon VPC，以便使用 IPv6 寻址。有关更多信息，请参阅《Amazon VPC 用户指南》中的 VPC 中的 IPv6 支持。
4. 选择 Amazon Redshift Serverless 的 Virtual Private Cloud (VPC)。这会将工作组分配给您的 AWS 环境中的特定虚拟网络。使用双栈模式时，您选择的 Amazon VPC 必须支持 IPV6 寻址。有关 Amazon VPC 的更多信息，请参阅 VPC 和子网概览。
5. 选择一个或多个 VPC 安全组。有关更多信息，请参阅使用安全组控制到资源的流量。
6. 选择是否启用额外的计算资源来执行自动优化。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的分配额外的计算资源来执行自动数据库优化。
7. 在子网下，指定要与数据库关联的一个或多个子网。这些子网包含在您之前选择的 Amazon VPC 中，并且必须位于三个不同的可用区中。有关更多信息，请参阅使用 Amazon Redshift Serverless 时的注意事项。
8. 选择符合您要求的基本 RPU 容量。

选择命名空间

1. 选择创建新的命名空间，然后输入命名空间名称；或者选择添加到现有命名空间，然后从下拉列表中选择命名空间。
2. 对于数据库名称和密码，指定第一个数据库的名称。还可以通过编辑管理员用户凭证，指定默认控制台管理员以外的管理员。
3. 对于权限，您选择关联 IAM 角色以将特定的 IAM 角色与命名空间和工作组关联。有关将 IAM 角色与 Amazon Redshift 关联的更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。
4. 您可以通过创建新密钥或选择默认密钥以外的密钥来自定义加密设置。对于审计日志记录，选择要导出的日志。每种日志类型指定不同的元数据。选择继续以查看您的选择。

检查工作组选择

1. 在审核和创建下检查您的设置。它显示了您在前面的步骤中选择的设置。
2. 选择 保存。

创建工作组后，它将添加到工作组列表。

## 查看工作组的属性

在 Amazon Redshift Serverless 中，工作组是可供使用的计算资源的集合。当您选择 Amazon Redshift Serverless 时，在 AWS 控制台中，您可以从导航菜单选择工作组配置以查看列表。您可以使用搜索框以查找符合搜索条件的工作组。每个工作组条目都显示了一些属性：

- 工作组 - 工作组的名称。您可以选择它来查看和编辑工作组的属性。
- 状态 - 显示工作组是否可用。
- 命名空间 - 与工作组关联的命名空间。每个工作组都与一个命名空间相关联。
- 创建日期 - 创建工作组的日期（UTC）。
- 标签 – 与工作组关联的标签。

此外，工作组配置还有另一个托管式工作组列表，此列表中的工作组是由 AWS Glue 管理的 Amazon Redshift Serverless 工作组。有关托管式工作组的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Managed workgroups。

工作组属性

您可以通过在左侧菜单中选择工作组配置以列出工作组。然后，您可以从列表中选择工作组。几个面板显示了工作组的属性。您还可以执行操作。一般信息显示以下内容：

- 工作组 - 工作组的名称。
- 命名空间 - 与工作组关联的命名空间。您可以选择它以查看其属性。工作组与单个命名空间相关联。
- 创建日期 - 创建工作组的时间。
- 状态 - 指示工作组资源是否可用。如果可用，您可以使用客户端连接到 Amazon Redshift Serverless 实例，以便查询数据或创建数据库资源，也可以使用查询编辑器 v2 进行连接。
- 端点 - URL。
- JDBC URL - 用于建立 JDBC 客户端连接的 URL。您可以使用此 URL 连接 Amazon Redshift 的 JDBC 驱动程序。有关更多信息，请参阅为 Amazon Redshift 配置 JDBC 驱动程序版本 2.x 连接。
- ODBC URL - 用于建立 ODBC 客户端连接的 URL。它包含数据库和用户 ID 等属性以及它们的值。
- 工作组版本和补丁版本 – Amazon Redshift Serverless 定期发布新版本和补丁。您可以使用工作组版本号和补丁版本号来跟踪 Amazon Redshift Serverless 工作组的软件更新。有关特定补丁中的更改和功能的更多信息，请参阅 Amazon Redshift 的集群版本。
- 工作组 ARN：工作组的 Amazon 资源名称。
- 用于自动优化的额外计算资源：Amazon Redshift 是否正在分配额外的计算资源来执行自动优化，即使在用量大的时段也是如此。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的分配额外的计算资源来执行自动数据库优化。

数据访问选项卡包含几个面板：

- 网络和安全 - 您可以查看网络属性，例如虚拟私有云（VPC）标识符、VPC 安全组列表、增强型 VPC 路由、IP 地址类型和可公开访问设置。如果选择编辑，您可以更改这些设置。此外，还可以选择启用增强型 VPC 路由，它通过 VPC 在无服务器数据库和数据存储库之间路由网络流量，以增强隐私和安全性。您还可以选择开启可公有访问，这使得数据库可以从 VPC 外部公开访问，从而允许实例和设备进行连接。

  IP 地址类型可以设置为双栈模式，以支持同时使用 IPv4 和 IPv6 访问工作组。有关网络层通信互联网协议（IP，Internet Protocol）的更多信息，请参阅 Wikipedia中的 Internet Protocol。
- Redshift 托管式 VPC 端点 - 您可以创建托管式 VPC 端点以从另一个 VPC 访问 Amazon Redshift Serverless。

限制选项卡中有用于控制 Amazon Redshift Serverless 的容量和使用限制的设置。它包含以下面板：

- 以 Redshift 处理单元 (RPU) 表示的基本容量 – 您可以设置用于处理工作负载的计算资源的基本容量。有关更多信息，请参阅 Amazon Redshift Serverless 的计算容量。
- 使用限制 – 对于在一段时间内，Amazon Redshift Serverless 实例可以使用的最大计算资源，您可以设置四种限制，并选择在达到这些限制时 Amazon Redshift Serverless 执行的操作。例如，您可以为工作组设置两个限制，一个是 500 个 RPU 小时，另一个是 900 个 RPU 小时。您可以让 Amazon Redshift Serverless 在达到 500 个 RPU 小时的第一个限制时向您发送提醒，然后在达到 900 个小时的第二个限制时关闭用户查询。这些限制有助于控制成本并使其更具可预测性。
- 查询限制 - 您可以对查询设置限制，例如超时设置。这些限制有助于您优化成本和性能。

选项卡选项卡有标签面板，其中显示您为工作组创建的所有标签。有关标记 资源的更多信息，请参阅 在 Amazon Redshift Serverless 中为资源添加标签

托管式工作组属性

也可以在托管式工作组列表下选择由 AWS Glue Data Catalog 管理的工作组。

托管式工作组与常规工作组具有不同的属性。有关托管式工作组的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Managed workgroups。

一般信息显示以下内容：

- 工作组 - 托管式工作组的名称。
- 创建日期 - 创建托管式工作组的日期（UTC）。
- 目录 ARN – AWS Glue Data Catalog 中托管式工作组的 Amazon 资源名称（ARN）。
- 状态 - 指示托管式工作组的计算资源是否可用。如果资源可用，则可以使用与 Apache Iceberg 兼容的 SQL 客户端连接到使用托管式工作组的目录，以便查询数据或创建数据库资源。也可以使用 Amazon Redshift 查询编辑器 V2 连接到目录。

查询和数据库监控包含托管式工作组性能图，显示一段时间内工作组中所有查询的平均经过时间。

查询历史记录选项卡是来自托管式工作组的所有查询的列表。其详细信息包括运行查询的用户、发起查询的客户端引擎以及查询的 ID 和状态等信息。“用户”选项卡是工作组中所有用户的列表。性能指标选项卡显示各种指标，例如平均查询时间、已完成的查询数和已使用的存储容量百分比。

### 删除工作组

您可以使用控制台删除工作组。在执行此操作之前，请确保已备份数据并已准备好快照。在许多情况下，无法检索作为工作组一部分删除的资源。

完成以下步骤：

1. 选择 Amazon Redshift Serverless，选择工作组配置，然后选择删除 Amazon Redshift Serverless 实例。

2. 此时将打开一个对话框。当您选择删除工作组时，将删除所有使用限制，删除所有 VPC 端点，并删除对 VPC 端点的访问权限。

   键入 delete，并选择删除以进行确认。

完成这些步骤后，工作组的状态为正在删除，并且横幅表明正在删除工作组。正在执行删除过程时，无服务器控制面板下的某些功能将被禁用。但是，您可以在预置集群控制面板上配置预置的集群。

删除工作组后，它不会随命名空间一起显示。您可以选择创建工作组按钮以创建新的工作组。

您可以删除现有工作组，然后将具有不同配置的新工作组关联到同一命名空间。创建新工作组时，请选择基本容量，用于确定与命名空间关联的数据大小。

您可以将工作组与使用客户托管式密钥 (CMK) 创建的命名空间相关联。有关 AWS KMS 的更多信息，请参阅 AWS KMS 概念。

## 命名空间

在 Amazon Redshift Serverless 中，命名空间定义了数据库对象的逻辑容器。它可以存放表、工作组和其他数据库资源。如果您尚未创建工作组和命名空间，并且正在寻找有关如何开始使用 Amazon Redshift Serverless 的说明，请参阅首次设置 Amazon Redshift Serverless。

### 命名空间属性

在 Amazon Redshift Serverless 中，命名空间定义了数据库对象的容器。您可以从导航列表中选择配置命名空间，从列表中选择命名空间，然后编辑其设置。

命名空间的一般信息包括以下内容：

- 命名空间 - 名称。
- 命名空间 ID - 唯一标识符。

- ARN - 用于跨 AWS 指定资源的唯一标识符。它包含区域和服务等属性。
- 状态 - 状态，例如 可用。
- 创建日期 - 创建命名空间的日期（UTC）。
- 使用的存储空间 - 命名空间及其所有对象使用的存储空间。
- 管理员用户名 - 管理员账户。这通常是用于创建命名空间的账户。
- 数据库名称 - 命名空间中包含的数据库的名称。
- 总表数 - 所有架构中表的计数。

命名空间的其他设置和属性位于多个选项卡上。这些功能包括：

- 工作组 - 显示与命名空间关联的工作组。
- 数据备份 - 在此面板上，您可以配置和创建快照，并配置恢复点。
- 安全性和加密 - 您可以管理 IAM 角色权限以及查看或编辑安全和加密设置。其中包括您的加密密钥状态以及开启审计日志记录的设置。有关 Amazon Redshift Serverless 的审计日志记录的更多信息，请参阅 Amazon Redshift Serverless 的审计日志记录。
- 数据共享 - 显示数据共享。通过数据共享，您可提供对数据的访问，而无需复制或移动数据。有关数据共享的更多信息，请参阅 Amazon Redshift Serverless 中的数据共享。

## 搜索命名空间

从 Amazon Redshift 菜单中，您可以从命名空间列表中进行选择，以查看或编辑命名空间的属性。控制台上的信息包括命名空间名称、管理员名称和其他属性。

命名空间的设置和属性位于多个选项卡上。这些功能包括：

- 工作组 - 显示与命名空间关联的工作组。
- 数据备份 - 您可以配置和创建快照，并配置恢复点。
- 安全性和加密 - 您可以管理 IAM 角色权限以及查看或编辑安全和加密设置。其中包括您的加密密钥状态和审计日志记录设置。
- 数据共享 - 显示数据共享。

## 编辑安全性和加密

通过 KMS 加密保护 Amazon Redshift Serverless。您可以通过控制台更新加密设置：

1. 从控制台的主菜单中选择命名空间配置，选择要编辑的命名空间，然后选择安全性和加密选项卡上的编辑。此时将显示对话框。
2. 您可以选择自定义加密设置、选择 AWS 客户托管密钥以更改用于加密资源的密钥。
3. 对于审计日志记录，选择要导出的日志。每种日志类型指定不同的元数据。
4. 要完成配置更新，请选择保存更改。

## 更改命名空间的 AWS KMS 密钥

在 Amazon Redshift 中，加密为静态数据提供保护。Amazon Redshift Serverless 自动使用 AWS KMS 密钥加密来加密您的 Amazon Redshift Serverless 资源和快照。作为最佳实践，大多数企业会审查其存储的数据类型，并计划按时间表轮换加密密钥。轮换密钥的频率可能会有所不同，具体取决于您的数据安全策略。Amazon Redshift Serverless 支持更改命名空间的 AWS KMS 密钥，以便您可以遵守企业的安全策略。

当您更改 AWS KMS 密钥时，数据保持不变。

使用控制台更改 AWS KMS 密钥

在 Amazon Redshift 中，加密为静态数据提供保护。Amazon Redshift Serverless 自动使用 AWS KMS 密钥加密来加密 Amazon Redshift Serverless 和快照。作为最佳实践，大多数企业会审查其存储的数据类型，并计划按时间表轮换加密密钥。轮换密钥的频率可能会有所不同，具体取决于您的数据安全策略。Amazon Redshift Serverless 支持更改命名空间的 AWS KMS 密钥，以便您可以遵守企业的安全策略。

当您更改 AWS KMS 密钥时，数据保持不变。

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择命名空间配置。从列表中选择命名空间。
3. 从安全性和加密选项卡上，选择编辑。
4. 选择自定义加密设置，然后为命名空间选择密钥。您可以选择创建新密钥。

使用 AWS CLI 更改 AWS KMS 加密密钥

使用 `update-namespace` 更改命名空间的 AWS KMS 密钥。下面说明此命令的语法：

```
aws redshift-serverless update-namespace
--namespace-name
```

```
[--kms-key-id <id-of-kms-key>]
// other parameters omitted here
```

必须已创建命名空间，否则 CLI 命令会导致错误。

更改密钥所需的时间取决于 Amazon Redshift Serverless 中的数据量。通常，对于每 8TB 存储数据，此过程需要 15 分钟。

限制

您无法将客户托管式 KMS 密钥更改为 AWS KMS 密钥。在这种情况下，您必须创建一个新的命名空间。

更改密钥时，您无法执行其他操作。

### 删除命名空间

如果要删除具有关联工作组的命名空间，则必须先删除该工作组。

在 Amazon Redshift Serverless 控制台上，完成以下步骤：

1. 从左侧菜单中选择配置命名空间，然后选择要从列表中删除的命名空间。
2. 选择操作，然后选择删除命名空间。
3. 此时会显示对话框。在完成删除操作之前，您可以通过创建手动快照来保留数据。

   键入 delete，并选择删除以进行确认。

## 使用 Amazon Redshift Serverless 监控查询和工作负载

您可以使用提供的系统视图监控 Amazon Redshift Serverless 查询和工作负载。

监控视图是 Amazon Redshift Serverless 中用于监控查询和工作负载使用情况的系统视图。这些视图位于 `pg_catalog` 架构中。可用的系统视图旨在为您提供监控 Amazon Redshift Serverless 所需的信息，这比预置集群所需的信息简单得多。SYS 系统视图旨在与 Amazon Redshift Serverless 配合使用。要显示这些视图提供的信息，请运行 SQL SELECT 语句。

系统视图的定义是为了支持以下监控目标。

工作负载监控

您可以随时间推移监控查询活动，以便：

- 了解工作负载模式，这样您就可以了解正常运行状况（基准）以及业务服务级别协议 (SLA) 中的内容。
- 快速识别偏离正常运行的状况，这可能是个暂时性问题，或是需要采取进一步操作的问题。

数据加载和卸载监控

进出 Amazon Redshift Serverless 的数据移动是一项关键功能。您可以使用 COPY 和 UNLOAD 来加载和卸载数据，并且必须密切监控传输的字节/行数和文件的完成进度，以跟踪遵守业务 SLA 的情况。这通常是通过频繁运行系统表查询（即每分钟）来完成，以便跟踪进度，并在检测到重大偏差时发出调查/纠正操作的提示。

故障和问题诊断

某些情况下，必须对查询或运行时故障采取措施。开发人员依靠系统表自行诊断问题，并确定正确的纠正补救措施。

性能优化

您可能需要优化从一开始就不符合 SLA 要求的查询或随时间推移而降级的查询。要进行优化，您必须具有运行时详细信息，包括运行计划、统计数据、持续时间和资源占用情况。您需要用于违规查询的基准数据，以确定偏差的原因并指导您如何改进性能。

用户对象事件监控

您需要监控用户对象上的操作和活动，例如刷新实体化视图、VACUUM 和分析。这包括系统管理的事件，例如实体化视图的自动刷新。如果事件是用户启动的，您想监控事件结束时间，或者如果事件是系统启动的，则监控最后一次成功运行。

计费的使用情况跟踪

您可以随时间转移监控您的使用情况趋势，以便：

- 告知预算计划和业务扩展估计值。
- 确定潜在的成本节约机会，例如删除冷数据。

使用 SYS 系统视图监控 Amazon Redshift Serverless。有关 SYS 监控视图的更多信息，请转到《Amazon Redshift 数据库开发人员指南》中的 SYS 监控视图。

## 添加查询监控策略

超级用户可以向不是超级用户的用户提供访问权限，以便他们可以对所有用户执行查询监控。首先，您可以为用户或角色添加策略以提供查询监控访问权限。然后，您可以授予用户或角色查询监控权限。

添加查询监控策略

1. 选择 https://console.aws.amazon.com/iam/。
2. 在访问管理下，选择策略。
3. 选择创建策略。
4. 选择 JSON，然后粘贴以下策略定义。

   JSON

```
{
"Version":"2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
    "redshift-data:ExecuteStatement",
    "redshift-data:DescribeStatement",
    "redshift-data:GetStatementResult",
    "redshift-data:ListDatabases"
],
"Resource": "*"
},
{
"Effect": "Allow",
"Action": "redshift-serverless:GetCredentials",
"Resource": "*"
}
]
}
```

5. 选择查看策略。
6. 对于名称，输入策略的名称，如 `query-monitoring`。
7. 选择创建策略。

创建策略后，您可授予相应的权限。

要提供访问权限，请为您的用户、组或角色添加权限：

- AWS IAM Identity Center 中的用户和群组：

创建权限集合。按照《AWS IAM Identity Center 用户指南》中创建权限集的说明进行操作。

- 通过身份提供者在 IAM 中托管的用户：

  创建适用于身份联合验证的角色。按照《IAM 用户指南》中针对第三方身份提供者创建角色（联合身份验证）的说明进行操作。

- IAM 用户：
  - 创建您的用户可以担任的角色。按照《IAM 用户指南》中为 IAM 用户创建角色的说明进行操作。
  - （不推荐使用）将策略直接附加到用户或将用户添加到用户组。按照《IAM 用户指南》中向用户添加权限（控制台）中的说明进行操作。

## 向用户授予查询监控权限

具有 sys:monitor 权限的用户可以查看所有查询。此外，具有 sys:operator 权限的用户可以取消查询、分析查询历史记录和执行 vacuum 操作。

授予用户查询监控权限

1. 输入以下命令以提供系统监控访问权限，其中 user-name 是您想要为其提供访问权限的用户的名称。

```
grant role sys:monitor to "IAM:user-name";
```

2. （可选）输入以下命令以提供系统操作员访问权限，其中 user name 是您想要为其提供访问权限的用户的名称。

```
grant role sys:operator to "IAM:user-name";
```

## 向角色授予查询监控权限

具有 sys:monitor 权限的角色的用户可以查看所有查询。此外，具有 sys:operator 权限的角色的用户可以取消查询、分析查询历史记录和执行 vacuum 操作。

授予角色查询监控权限

1. 输入以下命令以提供系统监控访问权限，其中 role-name 是您想要为其提供访问权限的角色的名称。

```
grant role sys:monitor to "IAMR:role-name";
```

2. （可选）输入以下命令以提供系统操作员访问权限，其中 role-name 是您想要为其提供访问权限的角色的名称。

```
grant role sys:operator to "IAMR:role-name";
```

## 设置使用限制，包括设置 RPU 限制

在工作组的限制选项卡中，您可以添加一个或多个使用限制，以控制在给定时间段内使用的最大 RPU 数，或设置数据共享使用限制。

1. 选择管理使用限制。限制部分显示在按时间段统计的计算使用量面板的底部。
2. 以 RPU 小时数为单位设置使用量限制。
3. 选择频率，即每天、每周或每月。这将设置使用量限制的时间段。在这种情况下选择每天可向您提供更详细的控制权。
4. 设置使用限制（以小时数为单位）。
5. 选择操作。这些操作包括：

   - 记录到系统表 - 向系统视图 SYS_QUERY_HISTORY 中添加记录。您可以在此视图中查询 `usage_limit` 列，来确定查询是否超出了限制。
   - 提醒 - 使用 Amazon SNS 设置通知订阅，并在违反限制时发送通知。您可以选择现有的 Amazon SNS 主题或创建一个新主题。
   - 关闭用户查询 - 禁用查询以停止使用 Amazon Redshift Serverless。它还会发送通知。

   前两个操作是信息性的，但最后一个操作会关闭查询处理。
6. （可选）您可以设置跨区域数据共享使用限制，这限制了使用者可以查询的从生产者区域传输到使用者区域的数据量。为此，请选择添加限制，然后按以下步骤操作。
7. 选择页面底部的保存以保存限制。
8. 根据需要最多再设置 3 个限制。

有关 RPU 和计费的更多概念信息，请参阅 Amazon Redshift Serverless 的计费。

# 设置查询限制

在工作组的限制选项卡中，您可以添加限制以监控性能和限制。有关查询监控限制的更多信息，请参阅 WLM 查询监控规则。

1. 选择管理查询限制。在管理查询限制对话框中选择添加新限制。
2. 选择要设置的限制类型，然后为其相应的限制输入一个值。
3. 选择保存更改以保存限制。

当您更改查询限制和配置参数时，数据库将重新启动。

# 设置查询队列

Amazon Redshift Serverless 支持基于队列的查询资源管理。您可以为不同的工作负载创建带有自定义监控规则的专用查询队列。此功能提供对资源使用情况的精细控制。

查询监控规则（QMR）仅适用于 Redshift Serverless 工作组级别，并一致地影响在该工作组中运行的所有查询。基于队列的方法支持您创建具有不同监控规则的队列。您可以将这些队列分配给特定的用户角色和查询组。每个队列独立运行，而规则仅影响该队列中的查询。

队列可让您设置基于指标的谓词和自动响应。例如，您可以配置规则，以自动中止超过时间限制或消耗过多资源的查询。

## 注意事项

使用无服务器队列时，请注意以下事项：

- Redshift 无服务器队列不支持在 Amazon Redshift 预置集群中使用的以下工作负载管理（WLM）配置
  键：`max_execution_time`、`short_query_queue`、`auto_wlm`、`concurrency_scaling`、`priorit`

  此外，无服务器不支持 hop、change_query_priority 操作。
- Amazon Redshift Serverless 不支持 hop 操作（在队列之间移动查询）。
- 仅 Amazon Redshift 预置集群支持队列优先级。
- Amazon Redshift Serverless 会自动管理扩展和资源分配以实现最佳性能，因此您无需手动配置队列优先级。

## 设置查询队列

您可以使用 AWS 管理控制台、AWS CLI 或 Redshift Serverless API 在“限制”选项卡下为无服务器工作组创建队列。

Console

按照以下步骤为无服务器工作组创建队列。

1. 导航到 Redshift Serverless 工作组。
2. 选择“限制”选项卡。
3. 在查询队列下，选择启用队列。

> **Important**
> 启用查询队列是一项永久性更改。启用后，您将无法恢复到无队列监控。

4. 使用以下参数配置队列：

   队列级别参数

   - `name`：队列标识符（必需、唯一、非空）
   - `user_role`：用户角色数组（可选）
   - `query_group`：查询组数组（可选）
   - `query_group_wild_card`：0 或 1 以启用通配符匹配（可选）
   - `user_group_wild_card`：0 或 1 以启用通配符匹配（可选）
   - `rules`：监控规则数组（可选）

   规则级别参数

   - `rule_name`：唯一标识符，最多 32 个字符（必需）
   - `predicate`：条件数组，1-3 个谓词（必需）
   - `action`：“abort”或“log”（必需）

   谓词级别参数

   - `metric_name`：要监控的指标（必需）

- operator：“=”、“<”或“>”（必需）
- value：数字阈值（必需）

限制

- 最多 8 个队列
- 所有队列中最多 25 条规则
- 每条规则最多 3 个谓词
- 规则名称必须全局唯一

示例配置

三个队列示例：一个用于超时时间较短的控制面板查询，一个用于超时时间较长的 ETL 查询，以及一个管理员队列：

```
[
  {
    "name": "dashboard",
    "user_role": ["analyst", "viewer"],
    "query_group": ["reporting"],
    "query_group_wild_card": 1,
    "rules": [
      {
        "rule_name": "short_timeout",
        "predicate": [
          {
            "metric_name": "query_execution_time",
            "operator": ">",
            "value": 60
          }
        ],
        "action": "abort"
      }
    ]
  },
  {
    "name": "ETL",
    "user_role": ["data_scientist"],
    "query_group": ["analytics", "ml"],
    "rules": [
```

```
      {
        "rule_name": "long_timeout",
        "predicate": [
          {
            "metric_name": "query_execution_time",
            "operator": ">",
            "value": 3600
          }
        ],
        "action": "log"
      },
      {
        "rule_name": "memory_limit",
        "predicate": [
          {
            "metric_name": "query_temp_blocks_to_disk",
            "operator": ">",
            "value": 100000
          }
        ],
        "action": "abort"
      }
    ]
  },
  {
    "name": "admin_queue",
    "user_role": ["admin"],
    "query_group": ["admin"]
  }
]
```

在本示例中：

- 如果控制面板查询运行时间超过 60 秒，则会中止
- 如果 ETL 查询运行时间超过一个小时，则会记录下来
- 管理员队列没有任何资源限制

### CLI

您可以将 CreateWorkgroup 或 UpdateWorkgroup API 与 `wlm_json_configuration` 配置参数结合使用来以 json 格式指定队列，从而管理队列。

```
aws redshift-serverless create-workgroup \
  --workgroup-name test-workgroup \
  --namespace-name test-namespace \
  --config-parameters '[{"parameterKey": "wlm_json_configuration", "parameterValue":
 "[{\"name\":\"dashboard\",\"user_role\":[\"analyst\",\"viewer\"],\"query_group
\":[\"reporting\"],\"query_group_wild_card\":1,\"rules\":[{\"rule_name\":
\"short_timeout\",\"predicate\":[{\"metric_name\":\"query_execution_time\",
\"operator\":\">\",\"value\":60}],\"action\":\"abort\"}]},{\"name\":\"ETL
\",\"user_role\":[\"data_scientist\"],\"query_group\":[\"analytics\",\"ml
\"],\"rules\":[{\"rule_name\":\"long_timeout\",\"predicate\":[{\"metric_name
\":\"query_execution_time\",\"operator\":\">\",\"value\":3600}],\"action\":
\"log\"},{\"rule_name\":\"memory_limit\",\"predicate\":[{\"metric_name\":
\"query_temp_blocks_to_disk\",\"operator\":\">\",\"value\":100000}],\"action\":
\"abort\"}]},{\"name\":\"admin_queue\",\"user_role\":[\"admin\"],\"query_group\":
[\"admin\"]}]"}]'
```

### 最佳实践

使用无服务器队列时，请记住以下最佳实践。

- 对具有不同限制要求（例如 ETL、报告或临时分析）的工作负载使用单独的队列。
- 从简单的阈值开始，然后根据查询行为和使用模式进行调整。您可以使用查询监控规则的系统表和视图中记录的表和视图来监控查询使用模式。

## 使用控制面板检查 Amazon Redshift Serverless 摘要数据

Amazon Redshift Serverless 控制面板包含一系列面板，这些面板一目了然地显示有关工作组和命名空间的指标和其他信息。这些面板包括：

- 资源摘要：显示有关 Amazon Redshift Serverless 的概括性信息，如使用的存储空间和其他指标。
- 查询摘要 - 显示有关查询的信息，包括已完成的查询和正在运行的查询。选择查看详细信息可转到具有其他筛选条件的屏幕。
- 已用 RPU 容量 - 显示给定时间段内使用的总容量，例如前十个小时。
- 数据共享 - 显示数据共享的计数，例如，用于在 AWS 账户之间共享数据的数据共享的计数。这些指标显示哪些数据共享需要授权以及其他信息。
- 总计算使用量 – 显示选定工作组在选定时间范围内（最多为过去 7 天）消耗的 RPU 总时数。

从控制面板中，您可以快速了解这些可用的指标，以查看有关 Amazon Redshift Serverless 的详细信息，或查看查询或跟踪工作项目。

# Amazon Redshift Serverless 的审计日志记录

您可以配置 Amazon Redshift Serverless，以将连接、用户和用户活动日志数据导出到 Amazon CloudWatch Logs 中的日志组。利用 Amazon CloudWatch Logs 可以对日志数据进行实时分析，并使用 CloudWatch 创建告警和查看指标。您可使用 CloudWatch Logs 在持久性存储中存储日志记录。

您可以使用 Amazon Redshift 控制台创建 CloudWatch 警报来跟踪您的指标。有关创建警报的更多信息，请参阅管理警报。

要将生成的日志数据导出到 Amazon CloudWatch Logs，必须在控制台上的 Amazon Redshift Serverless 配置设置中选择要导出的相应日志。要执行此操作，您可以在安全和加密下选择命名空间配置设置。

## CloudWatch 中的日志事件

在选择要导出的 Redshift 日志后，您可以在 Amazon CloudWatch Logs 中监控事件。系统将自动为 Amazon Redshift Serverless 创建新的日志组，其中 `log_type` 表示日志类型。

```
/aws/redshift/<namespace>/<log_type>
```

创建第一个工作组和命名空间时，default 是命名空间名称。日志组名称则根据您对命名空间的命名而异。

例如，如果您导出连接日志，则日志数据将存储在以下日志组中。

```
/aws/redshift/default/connectionlog
```

使用无服务器日志流将日志事件导出到日志组。该行为取决于以下哪些条件为真：

- 存在包含指定名称的日志组。Redshift 使用现有日志组导出日志数据。您可以使用自动化配置，例如由 AWS CloudFormation 提供的配置，创建具有预定义日志保留期、指标筛选条件和客户访问的日志组。
- 具有指定名称的日志组不存在。当在实例的日志中检测到匹配的日志条目时，Amazon Redshift Serverless 会自动在 Amazon CloudWatch Logs 中创建一个新的日志组。日志组使用永不过期的默认日志保留期。要更改日志保留期，请使用 Amazon CloudWatch Logs 控制台、AWS CLI 或

Amazon CloudWatch Logs API。有关在 CloudWatch Logs 中更改日志留存期的更多信息，请参阅使用日志组和日志流中的更改日志数据留存。

要在日志事件中搜索信息，请使用 Amazon CloudWatch Logs 控制台、AWS CLI 或 Amazon CloudWatch Logs API。有关搜索和筛选日志数据的更多信息，请参阅搜索和筛选日志数据。

## CloudWatch 指标

Amazon Redshift Serverless 指标分为计算指标以及数据和存储指标，分别属于工作组和命名空间维度集。有关工作组和命名空间的更多信息，请参阅工作组和命名空间。

CloudWatch 计算指标如下所示：

| 指标名称 | 单位 | 说明 | 维度集 |
| --- | --- | --- | --- |
| QueriesCompletedPerSecond | 查询数 | 每秒完成的查询数。 | {Database、LatencyRange、Workgroup}，{LatencyRange、Workgroup} |
| QueryDuration | 微秒 | 完成查询的平均时间量。 | {Database、LatencyRange、Workgroup}，{LatencyRange、Workgroup} |
| QueriesRunning | 查询数 | 一个时间点运行的查询数。 | {Database、QueryType、Workgroup}，{QueryType、Workgroup} |
| QueriesQueued | 查询数 | 某个时间点在队列中的查询数。 | {Database、QueryType、Workgroup}， |

| 指标名称 | 单位 | 说明 | 维度集 |
| --- | --- | --- | --- |
| | | | {QueryType、Workgroup} |
| DatabaseConnections | 连接数 | 一个时间点与数据库的连接数。 | {Database、Workgroup}，{Workgroup} |
| QueryRuntimeBreakdown | 毫秒 | 查询在各个查询阶段运行所花费的总时间。 | {Database、Stage、Workgroup}，{Stage、Workgroup} |
| ComputeCapacity | RPU | 过去 30 分钟内分配的平均计算单位数，向上舍入到最近的整数。 | {Workgroup} |
| ComputeSeconds | RPU 秒 | 过去 30 分钟内消耗的累计计算单位秒数。 | {Workgroup} |
| QueriesSucceeded | 查询数 | 过去 5 分钟内成功的查询数。 | {Database、QueryType、Workgroup}，{QueryType、Workgroup} |
| QueriesFailed | 查询数 | 过去 5 分钟内失败的查询数。 | {Database、QueryType、Workgroup}，{QueryType、Workgroup} |

| 指标名称 | 单位 | 说明 | 维度集 |
| --- | --- | --- | --- |
| UsageLimitAvailable | RPU 小时或 TB | 根据 UsageType，UsageLimitAvailable 将返回以下内容：<br>• 如果 UsageType 为 SERVERLESS_COMPUTE，则 UsageLimitAvailable 将返回工作组可以在给定限制内查询的剩余 RPU 小时数。<br>• 如果 UsageType 是 CROSS_REGION_DATASHARING，则 UsageLimitAvailable 将返回在给定限制下，客户可以扫描的剩余 TB 数量。 | {UsageLimitId、UsageType、Workgroup} |

| 指标名称 | 单位 | 说明 | 维度集 |
| --- | --- | --- | --- |
| UsageLimitConsumed | RPU 小时或 TB | 根据 UsageType，UsageLimitConsumed 会返回以下内容：<br>• 如果 UsageType 为 SERVERLESS_COMPUTE，则 UsageLimitConsumed 将返回工作组在给定限制内已查询的 RPU 小时数。<br>• 如果 UsageType 是 CROSS_REGION_DATASHARING，则 UsageLimitConsumed 将返回在给定限制下，客户已经扫描的 TB 数量。 | {UsageLimitId、UsageType、Workgroup} |

| 指标名称 | 单位 | 说明 | 维度集 |
| --- | --- | --- | --- |
| ExtraComputeForAutomaticOptimizationChargedSeconds | RPU 秒 | 过去 30 分钟内针对自动优化操作收取的计算单位秒数。 | {Workgroup} |

CloudWatch 数据和存储指标如下所示：

| 指标名称 | 单位 | 说明 | 维度集 |
| --- | --- | --- | --- |
| TotalTableCount | 表的数量 | 特定时间点的表总数。此总数包括永久表、临时表、数据共享表、外部表和实体化视图。此计数中不包括视图和系统表。 | {Database、Namespace} |
| DataStorage | 兆字节 | 在磁盘或存储空间中用于 Redshift 数据的兆字节数。 | {Namespace} |

ServerlessUniqueSnapshotStorage 指标与命名空间和工作组无关。CloudWatch 的 ServerlessUniqueSnapshotStorage 指标如下所示：

| 指标名称 | 单位 | 说明 | 维度集 |
| --- | --- | --- | --- |
| ServerlessUniqueSnapshotStorage | 兆字节 | 存储在所有活动的手动快照中的唯一数据块的兆字节数。 | {} |

维度集是应用于指标的分组维度。您可以使用这些维度组来指定如何检索统计数据。

下表详细说明了特定指标的维度和维度值：

| 维度 | 描述和值 |
| --- | --- |
| DatabaseName | 数据库的名称。自定义值。 |
| Latency | 可能值如下所示：<br>• Short– 不到 10 秒<br>• Medium – 在 10 秒到 10 分钟之间<br>• Long – 超过 10 分钟 |
| QueryType | 可能的值为 INSERT、DELETE、UPDATE、UNLOAD、LOAD、SELEC 和 OTHER。 |
| stage | 查询的执行阶段。可能值如下所示：<br>• QueryPlanning：分析和优化 SQL 语句所花的时间。<br>• QueryWaiting：在 WLM 队列中等待所花的时间。<br>• QueryExecutingRead：执行读取查询所花的时间。<br>• QueryExecutingInsert：执行插入查询所花的时间。<br>• QueryExecutingDelete：执行删除查询所花的时间。<br>• QueryExecutingUpdate：执行更新查询所花的时间。<br>• QueryExecutingCtas：执行 create table as 查询所花的时间。<br>• QueryExecutingUnload：执行卸载查询所花的时间。 |

| 维度 | 描述和值 |
| --- | --- |
| | • QueryExecutingCopy：执行复制查询所花的时间。<br>• QueryCommit：提交所花的时间。 |
| Namespace | 命名空间的名称。自定义值。 |
| Workgroup | 工作组的名称。自定义值。 |
| UsageLimitId | 使用限制的标识符。 |
| UsageType | 受限制的 Amazon Redshift Serverless 功能。可能值如下所示：<br>• SERVERLESS_COMPUTE<br>• CROSS_REGION_DATASHARING |

## 快照和恢复点

Amazon Redshift Serverless 中的备份，是您命名空间中的对象和数据在某个时间点的表示形式。备份有两种类型：手动创建的快照，以及 Amazon Redshift Serverless 自动为您创建的恢复点。

Amazon Redshift Serverless 每 30 分钟或在每个节点发生每 5 GB 数据更改后（以先发生者为准），会自动创建恢复点。对于较大的数据集（超过 5 GB × 节点数），恢复点之间的最小间隔为 15 分钟。所有恢复点都将保留 24 个小时。

> **Note**
>
> 您无法创建自己的快照计划来控制何时创建恢复点。

Amazon Redshift Serverless 在 Redshift 托管式存储（RMS）中创建快照。有关更多信息，请参阅 Amazon Redshift Serverless 的计算容量。

> **Note**
>
> RG 或 RA3 预置集群和 Amazon Redshift Serverless 工作组不支持无备份表。在 RG 或 RA3 集群和无服务器工作组中标记为无备份的表将被视为永久表，在拍摄快照时将始终对其进行

备份，并在从快照还原时还原该表。要避免无备份表产生快照成本，请在拍摄快照之前将其截断。

如果您发现需要检索快照或恢复点中的数据，则可以将快照还原到无服务器命名空间或预置集群。您可以在以下三种情况下还原快照：

- 将无服务器快照还原到无服务器命名空间。
- 将无服务器快照还原到预置集群。
- 将预置集群快照还原到无服务器命名空间。

将无服务器快照还原到预置集群时，您必须选择要使用的节点类型（例如 RG 或 RA3）和节点数量，这让您可以在集群或节点级别控制设置。

要将预置的集群快照还原到无服务器命名空间，请从 Redshift 预置的控制台开始，选择要还原的快照，然后选择从快照还原、恢复到无服务器命名空间。在您将预置的集群快照还原到无服务器命名空间时，Amazon Redshift 会将带有交错键的表转换为复合排序键。有关排序键的更多信息，请参阅使用排序键。

如果要添加额外的上下文，可以使用键值对标记快照和恢复点，这些键值对为快照和恢复点提供元数据和信息。有关标记资源的更多信息，请参阅标记资源概述。

最后，您还可以与其他 AWS 账户共享快照，允许他们访问快照中的数据并运行查询。

## AWS Backup 集成

您还可以使用 AWS Backup 创建和还原快照，这是一项完全托管式服务，有助于您在云端以及在本地集中管理和自动执行各种 AWS 服务中的数据保护。有关更多信息，请参阅 AWS Backup 与 Amazon Redshift 集成。有关 AWS Backup 的信息，请参阅《AWS Backup Developer Guide》中的 What is AWS Backup?

## 创建快照

> **Note**
>
> RG 或 RA3 预置集群和 Amazon Redshift Serverless 工作组不支持无备份表。在 RG 或 RA3 集群和无服务器工作组中标记为无备份的表将被视为永久表，在拍摄快照时将始终对其进行

备份，并在从快照还原时还原该表。要避免无备份表产生快照成本，请在拍摄快照之前将其截断。

要创建快照，请执行以下过程中的步骤。

要创建快照

1. 在 Amazon Redshift Serverless 控制台上，选择数据备份。
2. 选择创建快照。
3. 选择要创建其快照的命名空间。
4. 输入快照标识符。
5. （可选）选择保留期。如果您选择自定义值，请选择天数。您选择的数值必须在 1-3653（含）天之间。原定设置值是无限期保留。
6. 选择创建。

从命名空间配置创建快照

1. 在 Amazon Redshift Serverless 控制台上，选择命名空间配置。
2. 选择要创建其快照的命名空间。您只能创建与工作组关联且其状态为“可用”的命名空间的快照。
3. 选择数据备份选项卡。
4. 选择创建快照。
5. 输入快照标识符。
6. （可选）选择保留期。如果您选择自定义值，请选择天数。您选择的数值必须在 1-3653（含）天之间。
7. 选择创建。

## 生成最终快照

要在删除命名空间之前创建命名空间内所有数据的最终快照，请执行以下过程中的步骤。

创建最终快照

1. 在 Amazon Redshift Serverless 控制台上，选择命名空间配置。
2. 选择要删除的命名空间。

3. 依次选择操作和删除。
4. 选择创建最终快照。
5. 输入快照的名称。
6. 输入 delete。
7. 选择删除。

## 共享快照或移除快照权限

要与其它 AWS 账户共享快照或移除账户对快照的访问权限，请执行以下过程。

共享快照或移除对快照的访问权限

1. 在 Amazon Redshift Serverless 控制台上，选择数据备份。
2. 选择要共享的快照。
3. 依次选择操作和管理访问。
4. 要与其他账户共享快照，请输入 AWS 账户 ID。要删除账户的访问权限，请选择删除。
5. 选择保存更改。

## 计划快照

要精确控制创建快照的时间，您可以为特定命名空间创建快照计划。计划快照的创建时，您可以创建一次性事件，也可以使用 Unix cron 表达式来创建定期执行的计划。Cron 表达式支持以空格分隔三个字段。

```
cron(Minutes Hours Day-of-month Month Day-of-week Year)
```

| 字段 | 值 | 通配符 |
| --- | --- | --- |
| 分钟 | 0-59 | , - * / |
| 小时 | 0-23 | , - * / |
| Day-of-month | 1-31 | , - * ? / L W |
| Month | 1-12 或 JAN-DEC | , - * / |

| 字段 | 值 | 通配符 |
| --- | --- | --- |
| 星期几 | 1–7 或 SUN-SAT | , - * ? L # |
| Year | 1970-2199 | , - * / |

通配符

- , ( 逗号 ) 通配符包含其他值。在 Day-of-week 字段中，MON,WED,FRI 将包含星期一、星期三和星期五。总值限制为每字段 24 个。
- - ( 破折号 ) 通配符用于指定范围。在 Hour 字段中，1–15 将包含指定日期的 1 - 15 小时。
- * ( 星号 ) 通配符包含该字段中的所有值。在 Hours 字段中，* 将包含每个小时。
- / ( 正斜杠 ) 通配符用于指定增量。在 Hours 字段中，您可以输入 **1/10** 来指定从当天的第 1 个小时开始每隔 10 小时 ( 例如，01:00、11:00 和 21:00 ) 。
- ? ( 问号 ) 通配符用于指定一个或另一个。在 Day-of-month 字段中，您可以输入 7，如果您不介意 7 日是星期几，则可以在“星期几”字段中输入 ?。
- 或 字段中的 Day-of-monthLDay-of-week 通配符用于指定月或周的最后一天。
- Day-of-month 字段中的 W 通配符用于指定工作日。在 Day-of-month 字段中，3W 用于指定最靠近当月的第三周的日。
- “Day-of-week”字段中的 # 通配符用于指定一个月内所指定星期几的特定实例。例如，3#2 指该月的第二个星期二：3 指的是星期二，因为它是每周的第三天，2 是指该月内该类型的第二天。

> **Note**
>
> 如果使用 “#” 字符，则只能在星期字段中定义一个表达式。例如，“3#1,6#3”是无效的，因为它被解释为两个表达式。

限制

- 您无法在同一 cron 表达式中为 Day-of-month 和 Day-of-week 字段同时指定值。如果您在其中一个字段中指定了值，则必须在另一个字段中使用 ? ( 问号 ) 。
- 快照计划不支持以下频率：
  - 计划快照的频率超过每小时 1 次。
  - 计划快照的频率低于每天 1 次 ( 24 小时 ) 。

如果您有重叠的计划导致在 1 小时时段内计划了多次快照，将产生验证错误。

下表提供了一些示例 cron 字符串。

| 分钟 | 小时 | 星期几 | 意义 |
| --- | --- | --- | --- |
| 0 | 14-20/1 | TUE | 星期二下午 2 点到晚上 8 点之间，每小时拍摄一次。 |
| 0 | 21 | MON-FRI | 每天晚上 9 点，星期一至星期五。 |
| 30 | 0/6 | SAT-SUN | 星期六和星期日从当天午夜 30 分 (00:30) 开始，每 6 小时拍摄一次。这导致在每天的 [00:30、06:30、12:30 和 18:30] 拍摄快照。 |
| 30 | 12/4 | * | 每天从 12:30 开始，每 4 小时拍摄一次。这将解析为 [12:30、16:30、20:30]。 |

以下示例演示如何创建每天从 15:15 开始，以 2 小时为增量运行的计划。

```
cron(15 15/2 *)
```

可以使用 Amazon Redshift Serverless 控制台、API 或 AWS CLI 来创建快照计划。

计划快照

1. 在 Amazon Redshift Serverless 控制台上，选择数据备份。
2. 选择快照计划。

3. 选择创建计划。
4. 输入快照计划的名称。
5. 选择要为其创建快照的命名空间。
6. 为计划输入一个 cron 表达式，或使用计划生成器创建一个计划。
7. （可选）选择保留期。如果您选择自定义值，请指定天数。
8. 选择创建计划。

## 更新快照保留期

要更新快照保留期，请执行以下过程。

更新快照保留期

1. 在 Amazon Redshift Serverless 控制台上，选择数据备份。
2. 选择要更新的快照。
3. 依次选择操作和设置手动快照设置。
4. 选择保留期。如果您选择自定义值，请选择天数。
5. 选择保存更改。

## 删除快照

要删除快照，请执行以下过程。

要删除快照，请执行以下操作

> **Note**
>
> 您无法删除已与其他账户共享的快照。在删除快照之前，必须先删除该账户对快照的访问权限。

1. 在 Amazon Redshift Serverless 控制台上，选择数据备份。
2. 选择要删除的快照。
3. 依次选择操作和删除。

4. 选择删除。

## 还原快照

> **Note**
>
> RG 或 RA3 预置集群和 Amazon Redshift Serverless 工作组不支持无备份表。在 RG 或 RA3 集群和无服务器工作组中标记为无备份的表将被视为永久表，在拍摄快照时将始终对其进行备份，并在从快照还原时还原该表。要避免无备份表产生快照成本，请在拍摄快照之前将其截断。

将快照还原到无服务器命名空间，会将当前数据库替换为快照中的数据库。

将快照还原到无服务器命名空间分两个阶段完成。第一阶段在几分钟内完成，将数据还原到命名空间，并使其可用于查询。还原的第二阶段是优化数据库，这可能会导致轻微的性能问题。第二阶段可持续几个小时到几天，在某些情况下可持续几周。时间量取决于数据大小，但随着数据库进行优化，性能会逐步提高。在此阶段结束时，您的无服务器命名空间已充分优化，您可在没有性能问题的情况下提交查询。

将快照还原到无服务器命名空间

1. 在 Amazon Redshift Serverless 控制台上，选择数据备份。
2. 选择要还原的快照。一次只能还原一个快照。
3. 依次选择操作和恢复到无服务器命名空间。
4. 选择要还原到的可用命名空间。只能还原到其状态为“可用”的命名空间。
5. 选择还原。

将快照还原到预置集群

1. 在 Amazon Redshift Serverless 控制台上，选择数据备份。
2. 选择要还原的快照。
3. 依次选择操作和还原到预置集群。
4. 输入集群标识符。
5. 选择节点类型。节点的数量取决于节点类型。
6. 按照控制台页面上的说明进行操作，为集群配置输入属性。有关更多信息，请参阅创建集群。

有关预置集群上快照的更多信息，请参阅 Amazon Redshift 快照和备份。

## 转换恢复点

Amazon Redshift Serverless 中的恢复点大约每 30 分钟创建一次，并保存 24 小时。要将恢复点转换为快照，请执行以下过程中的步骤。

将恢复点转换为快照

1. 在 Amazon Redshift Serverless 控制台上，选择数据备份。
2. 在恢复点下，选择要转换为快照的恢复点的创建时间。
3. 选择从恢复点创建快照。
4. 输入快照标识符。
5. 选择创建。

## 还原恢复点

Amazon Redshift Serverless 中的恢复点大约每 30 分钟创建一次，并保存 24 小时。要将恢复点还原为无服务器命名空间，请执行以下过程中的步骤

将恢复点还原到无服务器命名空间

1. 在 Amazon Redshift Serverless 控制台上，选择 Data backup（数据备份）。
2. 在恢复点下，选择要还原的恢复点的创建时间。
3. 选择还原。只能还原到其状态为“可用”的命名空间。
4. 在文本输入字段中输入还原，然后选择还原。

## 将备份复制到其他 AWS 区域

您可以将 Amazon Redshift Serverless 配置为自动将快照和恢复点复制到其他 AWS 区域。当您在源 AWS 区域中创建快照时，它会被复制到目标区域。您可以配置命名空间，使其一次只能将快照和恢复点复制到一个 AWS 区域目标。有关提供了 Amazon Redshift Serverless 的 AWS 区域的列表，请参阅《Amazon Web Services 一般参考》中为 Redshift Serverless API 列出的端点。

在配置复制备份时，您还可以指定保留期，确定 Amazon Redshift Serverless 应将复制的快照保留多长时间。您无法更改恢复点的保留期，该值必须为 1 天。在目标区域中快照的保留期，独立于源区域

中快照的保留期。默认情况下，保留期为无限期保留快照。如果您选择自定义值，请选择天数。您选择的此数值必须在 1-3653（含）天之间。

如需更改要将快照复制到的目标区域，请先禁用复制备份，然后在重新启用复制时指定新的目标区域。

将快照或恢复点复制到目标区域后，您可以用这些快照或恢复点将数据恢复到区域中。

默认情况下，您的数据将使用 AWS 为您管理的密钥进行加密。要使用其他密钥，在源 AWS 区域中配置备份复制时，请指定要使用的密钥，然后 Amazon Redshift Serverless 会自动创建授权，从而在目标 AWS 区域启用快照加密。

要将备份复制到其他区域，请确保您具有以下 IAM 权限：

```
redshift-serverless:CreateSnapshotCopyConfiguration
redshift-serverless:UpdateSnapshotCopyConfiguration
redshift-serverless:ListSnapshotCopyConfigurations
redshift-serverless:DeleteSnapshotCopyConfiguration
```

如果您使用自己的 KMS 密钥来加密备份，则还需要以下权限：

```
kms:CreateGrant
kms:DescribeKey
```

配置将快照或恢复点复制到其他 AWS 区域

1. 在 Amazon Redshift Serverless 控制台上，选择要配置为复制其快照或恢复点的命名空间。
2. 依次选择操作、配置跨区域备份。
3. 选择要将快照复制到的目标 AWS 区域。
4. （可选）选择将快照保留多长时间。如果您选择自定义值，请选择天数。您选择的数值必须在 1-3653（含）天之间。默认设置是无限期保留。
5. （可选）选择其他 AWS KMS 密钥用于目标区域中的加密。
6. 选择 Save configuration。

## 还原表

您也可以从快照或恢复点还原特定表。执行此操作时，您需要指定源快照或恢复点、数据库、架构、表、目标数据库、架构和新表名。这个新表不能使用与现有表相同的名称。如果您希望还原表来替换现有的表，则必须先重命名或删除现有表，然后再还原表。

> **Note**
>
> RG 或 RA3 预置集群和 Amazon Redshift Serverless 工作组不支持无备份表。在 RG 或 RA3 集群和无服务器工作组中标记为无备份的表将被视为永久表，在拍摄快照时将始终对其进行备份，并在从快照还原时还原该表。但是，不支持选择性还原无备份表。

使用源表的列定义、表属性和列属性（外键除外）创建目标表。为了防止因依赖项而导致发生冲突，目标表不从源表继承外键。不向目标表应用任何依赖项（例如，源表上的视图或授予的权限）。

如果源表的所有者存在，那么该用户是已还原的表的所有者，前提是该用户拥有足够的权限成为在指定数据库和 schema 中指定的关系的所有者。否则，已还原的表由在启动集群时创建的管理员用户所有。

已还原的表将返回在执行备份时其所处的状态。这包括由 Amazon Redshift 对可序列化隔离的符合性定义的事务可见性规则，这意味着数据将立即对在备份后启动的进行中事务可见。

您可以使用 Amazon Redshift Serverless 控制台从快照还原表。

从数据备份还原表存在以下限制：

- 一次只能还原一个表。
- 不向目标表应用任何依赖项（例如，源表上的视图或授予的权限）。
- 如果为正在还原的表启用行级安全性，Amazon Redshift Serverless 将还原已启用行级安全性的表。

使用 Amazon Redshift Serverless 控制台还原表

1. 在 Amazon Redshift Serverless 控制台上，选择数据备份。
2. 选择包含要还原的表的快照或恢复点。
3. 选择操作、从快照还原表或从恢复点还原表。
4. 输入有关源快照或恢复点以及目标表的信息，然后选择还原表。

# Amazon Redshift Serverless 中的数据共享

通过数据共享，您可以实时访问数据，以便您的用户可以在 Amazon Redshift Serverless 中查看更新后的最新、最一致的信息。

您可以在 AWS 账户中或跨这些账户，在不同 Amazon Redshift Serverless 实例之间共享数据以进行读取。

您可以使用 SQL 界面或 Amazon Redshift 控制台开始使用数据共享。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Amazon Redshift 中的数据共享。

借助数据共享，Amazon Redshift Serverless 命名空间和预置集群可以相互共享实时数据，无论它们是在 AWS 账户 内、在不同 AWS 账户 中，还是跨 AWS 区域。有关更多信息，请参阅可进行数据共享的区域。

要在 AWS 账户 中开始共享数据，请打开 AWS 管理控制台，然后选择 Amazon Redshift 控制台。选择命名空间配置，然后选择数据共享。

要开始在数据共享中查询数据，请在具有关联工作组的命名空间中创建一个数据库。从指定的数据共享中，选择具有关联工作组的命名空间，然后创建数据库来查询数据。

## 注意事项

在 Amazon Redshift Serverless 中使用数据共享时，请注意以下事项：

- Amazon Redshift 支持在预调配的 RG 和 RA3 集群以及无服务器端点工作组上以生产者或使用者的身份共享数据。
- 默认情况下，Amazon Redshift Serverless 会加密。

有关数据共享限制的列表，包括支持的数据库对象、加密要求和排序键要求，请参阅《Amazon Redshift 数据库开发人员指南》中的 Considerations for data sharing in Amazon Redshift。

## 授予查看数据共享的访问权限

超级用户可以向不是超级用户的用户提供访问权限，以便他们可以查看所有用户创建的数据共享。

要为用户授予数据共享的访问权限，请使用以下命令为用户提供数据共享访问权限，其中 datashare_name 是数据共享的名称，user-name 是您想要为其提供访问权限的用户的名称。

```
grant share on datashare datashare_name to "IAM:test_user";
```

要为用户组授予数据共享的访问权限，请先创建一个包含用户的用户组。有关如何创建用户组的信息，请参阅创建组。然后，使用以下命令向用户授予数据共享访问权限，其中 datashare_name 是数据共享的名称，user-group 是要向其授予访问权限的用户组的名称。

```
grant share on datashare datashare_name to group user_group;
```

有关如何使用 GRANT 语句的信息，请参阅授权。

## 将命名空间注册到 AWS Glue Data Catalog

您可以将整个命名空间注册到 AWS Glue Data Catalog 并创建由 AWS Glue 管理的目录。您可以使用任何支持 Apache Iceberg REST API 的 SQL 引擎访问这些目录。有关从 Amazon Redshift 创建与 Apache Iceberg 兼容的目录的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Apache Iceberg compatibility for Amazon Redshift。

将无服务器命名空间注册到 AWS Glue Data Catalog

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Redshift Serverless。将显示无服务器控制面板。在命名空间/工作组部分中是您的账户在当前 AWS 区域中的命名空间和工作组的列表。如果您没有任何命名空间，请选择创建工作组来创建工作组及其相应的命名空间。
3. 选择要注册的命名空间的名称。
4. 从操作中，选择注册到 AWS Glue Data Catalog。将显示注册到 AWS Glue Data Catalog 弹出框。
5. 在目标账户 ID 下输入要向其注册命名空间的 AWS 账户 ID。这是将目录保存在 AWS Glue Data Catalog 中的账户 ID。
6. 在将命名空间注册为下输入名称。这将是 Data Catalog 中命名空间的名称。
7. 选择注册。您将进入 AWS Lake Formation 控制台。
8. 按照在 AWS Lake Formation 中创建目录的过程进行操作。有关创建目录的信息，请参阅《AWS Lake Formation 开发人员指南》中的 Bringing Amazon Redshift data into the AWS Glue Data Catalog。

# 在 Amazon Redshift Serverless 中为资源添加标签

在 AWS 中，标签是用户定义的标记，由键-值对组成。Amazon Redshift Serverless 支持添加标签，以便一目了然地提供有关资源的元数据。

虽然标签对资源来说并非必不可少，但它们却有助于提供上下文信息。您可能想使用与资源相关的信息，通过元数据来为资源添加标签。例如，假设您要跟踪属于测试环境和生产环境的资源。您可以创建

名为 environment 的键并提供值 test 或 production，以确定在每种环境中使用的资源。如果您在其他 AWS 服务中使用了标签或者具有针对您业务的标准类别，那么，我们建议您为资源创建相同的键值对以保持一致性。

如果删除资源，则会删除所有关联的标签。您可以同时使用 AWS CLI 和 Amazon Redshift Serverless 控制台来标记无服务器资源。可用的 API 操作包括 `TagResource`、`UntagResource` 和 `ListTagsForResource`。

每个资源都有一个标签集，它是分配给该资源的一个或多个标签的集合。每个资源的每个标签集内最多有 50 个标签。您可以在创建资源时以及资源创建完成后添加标签。您可以向以下无服务器资源类型添加标签：

- 工作组
- 命名空间
- 快照
- 恢复点

标签具有以下要求：

- 键不得以 `aws:` 作为前缀。
- 每个标签集中的各个键必须是独一无二的。
- 键的长度必须介于 1 到 128 个允许的字符之间。
- 值的长度必须介于 0 到 256 个允许的字符之间。
- 每个标签集中的值不需要是唯一的。
- 可以用作键和值的字符包括 Unicode 字符、数字、空格及以下符号：_ . : / = + - @。
- 键和值区分大小写。

管理您的 Amazon Redshift Serverless 资源的标签

1. 在 Amazon Redshift Serverless 控制台上，选择 Manage Tags（管理标签）。
2. 输入要搜索的资源类型，然后选择 Search resources（搜索资源）。选择要为其管理标签的资源，然后选择 Manage tags（管理标签）。
3. 指定要添加到资源的键和可选值。修改标签时，您可以更改标签的值，而不能更改键。
4. 在添加、删除或修改标签后，选择 Save changes（保存更改），然后选择 Apply（应用）以保存您的更改。

# Amazon Redshift 预置集群

Amazon Redshift 数据仓库是一个由称作节点的各种计算资源构成的集合，这些节点已整理到名为集群的组中。每个集群运行一个 Amazon Redshift 引擎并包含一个或多个数据库。

> **Note**
>
> 目前提供 Amazon Redshift 版本 1.0 引擎。不过，随着引擎不断更新，我们可能会提供多个 Amazon Redshift 引擎版本供您选择。

## Amazon Redshift 中的集群和节点

Amazon Redshift 集群由节点组成。每个集群包括一个领导节点以及一个或多个计算节点。领导节点接收来自客户端应用程序的查询、解析查询并制定查询执行计划。然后，领导节点和计算节点协调这些计划并行执行，之后，领导节点聚合来自计算节点的中间结果。然后，领导节点会将这些结果最终返回至客户端应用程序。

计算节点运行查询执行计划，并在节点自身之间传输数据以对这些查询提供服务。中间结果被送回至客户端应用程序之前，会先发送至领导节点进行聚合。有关领导节点和计算节点的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的数据仓库系统架构。

> **Note**
>
> 当您在 Amazon Redshift 控制台 (https://console.aws.amazon.com/redshiftv2/) 上创建集群时，您可以获得基于数据大小和查询特征的集群配置建议。要使用此大小调整计算器，请在支持 RG 或 RA3 节点类型的 AWS 区域中，在控制台上查找帮我选择。有关更多信息，请参阅创建集群。

当您启动集群时，您指定的一个选项是节点类型。节点类型决定了每个节点的 CPU、RAM、存储容量和存储驱动类型。

Amazon Redshift 提供不同的节点类型来满足您的工作负载要求，我们建议根据所需的性能、数据大小和预期的数据增长选择 RG 或 RA3。

通过使用具有托管存储的 RG 和 RA3 节点，您可以单独扩展和支付计算和托管存储以优化数据仓库。通过使用 RG 或 RA3，您可以根据性能要求选择节点数，并且仅为您使用的托管存储支付费用。根

据您每天处理的数据量调整 RG 或 RA3 集群大小。您可以在虚拟私有云（VPC）中启动使用 RG 或 RA3 节点类型的集群。有关更多信息，请参阅 在 VPC 中创建 Redshift 预置集群或 Amazon Redshift Serverless 工作组。

Amazon Redshift 托管存储在每个 RG 和 RA3 节点中使用大型高性能 SSD 以提供快速的本地存储，并使用 Amazon S3 提供更长时间的持久存储。如果节点中的数据超过大型本地 SSD 的大小，Amazon Redshift 托管存储自动将该数据分流到 Amazon S3。无论数据存储在高性能 SSD 还是 Amazon S3 中，Amazon Redshift 托管存储采用相同的费率，而且费用较低。对于需要不断增长的存储的工作负载，您可以通过托管式存储，独立于计算节点自动扩展数据仓库存储容量。

与 RA3 节点类型相比，RG 节点提供以下功能：

- 基于 Graviton 的实例类型：为分析工作负载提供卓越的性价比
- 集成式数据湖查询引擎：RG 集群包括一个集成式数据湖查询引擎，该引擎直接在集群自己的计算资源上运行。RA3 集群使用 Amazon Redshift Spectrum 进行数据湖查询。

通过使用 DC2 节点，您可以创建包含本地 SSD 存储的计算密集型数据仓库。您可以根据数据大小和性能要求选择所需的节点数。DC2 节点在本地存储数据以获得较高的性能；随着数据大小增长，您可以添加更多计算节点以增加集群的存储容量。对于 1 TB 以下的数据集（压缩），我们建议使用 DC2 节点类型，从而以最低的价格获得最佳的性能。如果预计数据将会增长，我们建议您使用 RA3 节点，以便单独调整计算和存储大小以降低价格并提高性能。您可以在 Virtual Private Cloud (VPC) 中启动使用 DC2 节点类型的集群。有关更多信息，请参阅 在 VPC 中创建 Redshift 预置集群或 Amazon Redshift Serverless 工作组。

节点类型有各种不同的大小。节点大小和节点数决定了集群的总存储容量。有关更多信息，请参阅 节点类型详细信息。

一些节点类型允许一个节点（单节点）或者两个/更多节点（多节点）。某些节点类型的集群的最小节点数为两个节点。在单节点集群上，该节点由领导功能和计算功能共用。不建议使用单节点集群运行生产工作负载。在多节点集群上，领导节点与计算节点是分开的。领导节点与计算节点的节点类型相同。您仅需为计算节点付费。

Amazon Redshift 为每个 AWS 区域内的每个 AWS 账户应用资源配额。配额限制您的账户可以在 AWS 区域中为给定资源类型（如节点或快照）创建的资源数。有关适用于 Amazon Redshift 资源的默认配额的更多信息，请参阅 Amazon Redshift 资源中的配额和限制。

您的集群成本取决于 AWS 区域、节点类型、节点数以及是否提前预留节点。有关节点成本的更多信息，请参阅 Amazon Redshift 定价页面。

# 节点类型详细信息

以下各表总结了每个节点类型和大小的节点规范。表中的标题具有以下含义：

- vCPU 指的是每个节点的虚拟 CPU 数。
- RAM 指的是每个节点的内存量，以吉字节 (GiB) 为单位。
- 每个节点的默认切片数 是指，在创建集群或使用经典调整大小调整集群大小时，将计算节点划分到的切片数。

  如果使用弹性调整大小来调整集群大小，则每个节点的切片数可能会发生变化。不过，在进行弹性调整大小后，集群中所有计算节点上的切片总数将保持不变。

  在使用从快照中还原操作创建集群时，如果更改节点类型，创建的集群的切片数可能与原始集群不同。
- 存储指的是每个节点的容量和存储类型。
- 节点范围指的是针对节点类型和大小，Amazon Redshift 支持的最少和最多节点数。

> **Note**
>
> 根据应用到所选 AWS 区域中您的 AWS 账户的配额，您可能只能使用少量节点。有关适用于 Amazon Redshift 资源的默认配额的更多信息，请参阅 Amazon Redshift 资源中的配额和限制。

- 总容量指的是集群的总存储容量（如果您部署了节点范围中指定的最大数目的节点）。

下表介绍 RG 节点的规格。

| 节点类型 | vCPU | RAM (GiB) | 每个节点的默认切片数 | 每个节点的托管式存储限制 [1] | 创建集群的节点范围 | 总托管式存储容量 [2] | |
|---|---|---|---|---|---|---|---|
| rg.large（单节点） | 2 | 16 | 2 | 1TB | 1 | 1TB | |

| 节点类型 | vCPU | RAM (GiB) | 每个节点的默认切片数 | 每个节点的托管式存储限制 [1] | 创建集群的节点范围 | 总托管式存储容量 [2] |
|---|---|---|---|---|---|---|
| rg.large（多节点） | 2 | 16 | 2 | 8 TB | 2 – 16 | 128 TB |
| rg.xlarge（多节点） | 4 | 32 | 2 | 32 TB | 2–163[3] | 1024TB[3] |
| rg.4xlarge | 16 | 128 | 8 | 128 TB | 2–324[4] | 8192 TB[4] |
| rg.12xlarge | 48 | 384 | 16 | 128 TB | 2–128 | 16384 TB |

[1] Amazon Redshift 托管存储的存储限制。这是一个硬性限制。

[2] 总托管式存储限制是每个节点的最大节点数乘以托管式存储限制。

[3] 您可以使用最多包含 16 个节点的 rg.xlarge（多节点）节点类型创建一个集群。对于多节点集群，您可以使用弹性调整大小将大小调整为最多 32 个节点。

[4] 您可以使用最多包含 32 个节点的 rg.4xlarge（多节点）节点类型创建一个集群。您可以通过弹性调整大小将大小调整为最多 64 个节点。

下表介绍 RA3 节点的规格。

| 节点类型 | vCPU | RAM (GiB) | 每个节点的默认切片数 | 每个节点的托管式存储限制 [1] | 创建集群的节点范围 | 总托管式存储容量 [2] |
|---|---|---|---|---|---|---|
| ra3.large（单节点） | 2 | 16 | 2 | 1TB | 1 | 1 TB[3] |
| ra3.large（多节点） | 2 | 16 | 2 | 8 TB | 2-16 | 128 TB |

| 节点类型 | vCPU | RAM (GiB) | 每个节点的默认切片数 | 每个节点的托管式存储限制 [1] | 创建集群的节点范围 | 总托管式存储容量 [2] |
|---|---|---|---|---|---|---|
| ra3.xlplus（单节点） | 4 | 32 | 2 | 4 TB | 1 | 4 TB[3] |
| ra3.xlplus（多节点） | 4 | 32 | 2 | 32 TB | 2–164[4] | 1024 TB[4] |
| ra3.4xlarge | 12 | 96 | 4 | 128 TB | 2–325[5] | 8192 TB[5] |
| ra3.16xlarge | 48 | 384 | 16 | 128 TB | 2–128 | 16384 TB |

[1] Amazon Redshift 托管存储的存储限制。这是一个硬性限制。

[2] 总托管式存储限制是每个节点的最大节点数乘以托管式存储限制。

[3] 要将单节点集群的大小调整为多节点，只支持经典大小调整。

[4] 您可以创建一个最多包含 16 个节点的 ra3.xlplus（多节点）节点类型的集群。对于多节点集群，您可以使用弹性调整大小将大小调整为最多 32 个节点。

[5] 您可以使用最多包含 32 个节点的 ra3.4xlarge 节点类型创建一个集群。您可以通过弹性调整大小将大小调整为最多 64 个节点。

下表介绍密集计算节点的规格。

| 节点类型 | vCPU | RAM (GiB) | 每个节点的默认切片数 | 每个节点的存储 | 节点范围 | 总容量 |
|---|---|---|---|---|---|---|
| dc2.large | 2 | 15 | 2 | 160 GB NVMe-SSD | 1–32 | 5.12 TB |

| 节点类型 | vCPU | RAM (GiB) | 每个节点的默认切片数 | 每个节点的存储 | 节点范围 | 总容量 | |
|---|---|---|---|---|---|---|---|
| dc2.8xlarge | 32 | 244 | 16 | 2.56 TB NVMe-SSD | 2–128 | 326 TB | |

> **Note**
>
> 密集存储（DS2）节点类型不再可用。

以前的节点类型名称

在先前的 Amazon Redshift 版本中，某些节点类型的名称不同。您可以在 Amazon Redshift API 和 AWS CLI 中使用以前的名称。但是，我们建议您更新引用这些名称的所有脚本，以改用当前名称。当前名称和以前的名称如下所示。

| 当前名称 | 以前的名称 |
|---|---|
| ds2.xlarge | ds1.xlarge、dw.hs1.xlarge、dw1.xlarge |
| ds2.8xlarge | ds1.8xlarge、dw.hs1.8xlarge、dw1.8xlarge |
| dc1.large | dw2.large |
| dc1.8xlarge | dw2.8xlarge |

## 确定节点数

由于 Amazon Redshift 在集群的所有计算节点间并行分配和运行查询，因此您可以通过向集群添加节点来提高查询性能。在运行具有至少两个计算节点的集群时，每个节点上的数据将镜像到其他节点的磁盘上，从而降低数据丢失的风险。

您可以在 Amazon Redshift 控制台中使用 Amazon CloudWatch 指标监控查询性能。您还可以根据需要添加或删除节点，以便在集群的价格和性能之间达到平衡。当您请求额外节点时，Amazon Redshift 会处理部署、负载均衡和数据维护方面的所有详细信息。有关集群性能的更多信息，请参阅监控 Amazon Redshift 集群性能。

预留节点适合状态稳定的生产工作负载，可以提供比按需节点大得多的折扣。在运行试验和概念验证以验证生产配置后，您可以购买预留节点。有关更多信息，请参阅 预留节点。

在暂停集群时，您可以在集群暂停期间暂停按需计费。在该暂停时间内，您仅需为备份存储付费。这可避免规划和购买超出需求的数据仓库容量，并使您能够经济高效地管理环境以进行开发或测试。

有关按需和预留节点定价的信息，请参阅 Amazon Redshift 定价。

# 使用 EC2 创建集群

Amazon Redshift 集群在已针对您选择的 Amazon Redshift 节点类型和大小配置的 Amazon EC2 实例中运行。有关这些联网平台的更多信息，请参阅《Amazon EC2 用户指南》中的支持的平台。

> **Note**
>
> 为了防止 SQL 客户端工具与 Amazon Redshift 数据库之间出现连接问题，建议执行下列两项操作之一。您可以配置一个允许主机协商数据包大小的入站规则。或者，您可以通过在 Amazon EC2 实例的网络接口 (NIC) 上将最大传输单位 (MTU) 设置为 1500 来禁用 TCP/IP 巨型帧。有关这些方法的更多信息，请参阅查询似乎挂起，有时无法连接到集群。

## Amazon Virtual Private Cloud (Amazon VPC)

使用 Amazon VPC 时，集群在逻辑上与您的 AWS 账户隔离的虚拟私有云（VPC）中运行。如果您使用 Amazon VPC 预置集群，则可以通过将一个或多个 VPC 安全组与集群关联以控制对集群的访问。有关更多信息，请参阅《Amazon VPC 用户指南》 中的您的 VPC 的安全组。

要在 VPC 中创建集群，您必须先通过提供 VPC 的子网信息来创建一个 Amazon Redshift 集群子网组，然后在启动集群时提供该子网组。有关更多信息，请参阅 Redshift 资源的子网。

有关 Amazon Virtual Private Cloud (Amazon VPC) 的更多信息，请参阅 Amazon VPC 产品详细信息页面。

# 默认磁盘空间警报

创建 Amazon Redshift 集群时，您可以选择配置一个 Amazon CloudWatch 告警来监控在集群的所有节点中已用磁盘空间所占的平均百分比。我们将此警报称为默认磁盘空间警报。

默认磁盘空间警报的目的是帮助您监控集群的存储容量。您可以根据数据仓库的需要配置此警报。例如，您可以使用警告作为可能需要调整集群大小的指示器。您可以将大小调整为不同的节点类型或添加节点，或者购买预留节点以供将来扩展。

当磁盘的使用率在指定持续时间达到或超出指定百分比一定次数，则会触发默认磁盘空间警报。默认情况下，当磁盘的使用率达到您指定的百分比并保持或超出这一百分比五分钟或更长时间，则会触发此警报。启动集群后，您可以编辑默认值。

当触发 CloudWatch 告警时，Amazon Simple Notification Service (Amazon SNS) 会向指定收件人发送通知，以提醒他们已达到百分比阈值。Amazon SNS 使用主题来指定接收方，并以通知的形式发送消息。您可以使用现有 Amazon SNS 主题；如果不使用的话，系统会根据您启动集群时指定的设置创建一个主题。启动集群后，您可以编辑此警报的主题。有关创建 Amazon SNS 主题的更多信息，请参阅开始使用 Amazon Simple Notification Service。

启动集群后，您可以从该集群 Status 窗口的 CloudWatch Alarms 下方查看和编辑警报。其名称为 percentage-disk-space-used-default-<*string*>。您可以打开该告警，以查看与其相关联的 Amazon SNS 主题以及编辑告警设置。如果您未选择使用现有的 Amazon SNS 主题，则系统为您创建的主题名为 <*clustername*>-default-alarms (<*recipient*>)，例如 examplecluster-default-alarms (notify@example.com)。

有关配置和编辑默认磁盘空间警报的更多信息，请参阅创建集群和创建磁盘空间警报。

> **Note**
>
> 如果您将集群删除，与该集群相关联的警报不会被删除，但也不会触发。如果您不再需要该警报，可以从 CloudWatch 控制台中将其删除。

# 集群状态

集群状态显示了集群的当前状态。下表提供了对每个集群状态的说明。

| 状态 | 描述 |
| --- | --- |
| available | 集群正在运行且可供使用。 |
| available, prep-for-resize | 该集群正在为弹性调整大小操作做准备。集群正在运行且可用于读取和写入查询，但集群操作（例如，创建快照）不可用。 |
| available, resize-cleanup | 弹性调整大小操作正在完成到新集群节点的数据传输。集群正在运行且可用于读取和写入查询，但集群操作（例如，创建快照）不可用。 |
| cancelling-resize | 正在取消调整大小操作。 |
| creating | Amazon Redshift 正在创建集群。有关更多信息，请参阅 创建集群。 |
| deleting | Amazon Redshift 正在删除集群。有关更多信息，请参阅 关闭和删除集群。 |
| final-snapshot | Amazon Redshift 正在删除集群前对其拍摄最终快照。有关更多信息，请参阅 关闭和删除集群。 |
| hardware-failure | 集群发生了硬件故障。<br><br>如果您的集群为单节点集群，则该节点无法替换。要恢复您的集群，请还原快照。有关更多信息，请参阅 Amazon Redshift 快照和备份。 |
| incompatible-hsm | Amazon Redshift 无法连接到硬件安全模块 (HSM)。检查集群和 HSM 之间的 HSM 配置。有关更多信息，请参阅 使用硬件安全模块的加密。 |
| incompatible-network | 基本网络配置出现问题。确保您在其中启动集群的 VPC 及其设置正确无误。有关更多信息，请参阅 VPC 中的 Redshift 资源。 |
| incompatible-parameters | 相关联的参数组中的一个或多个参数值出现问题，此时无法应用这些参数值。修改参数组并更新所有无效值。有关更多信息，请参阅 Amazon Redshift 参数组。 |
| incompatible-restore | 从快照中还原集群时出现问题。使用其他快照再次尝试还原集群。有关更多信息，请参阅 Amazon Redshift 快照和备份。 |

| 状态 | 描述 |
| --- | --- |
| modifying | Amazon Redshift 正在将更改应用于集群。有关更多信息，请参阅 修改集群。 |
| paused | 集群已暂停。有关更多信息，请参阅 暂停和恢复集群。 |
| rebooting | Amazon Redshift 正在重启集群。有关更多信息，请参阅 重新引导集群。 |
| renaming | Amazon Redshift 正在将新名称应用于集群。有关更多信息，请参阅 重命名集群。 |
| resizing | Amazon Redshift 正在调整集群的大小。有关更多信息，请参阅 调整集群大小。 |
| rotating-keys | Amazon Redshift 正在轮换集群的加密密钥。有关更多信息，请参阅 加密密钥轮换。 |
| storage-full | 集群已达到其存储容量。调整集群的大小以添加节点或选择其他节点大小。有关更多信息，请参阅 调整集群大小。 |
| updating-hsm | Amazon Redshift 正在更新 HSM 配置。 |

# 使用 Amazon Redshift 预置集群时的注意事项

创建集群后，您可以在本节中找到有关推出相应功能的区域、维护任务、节点类型和使用限制的信息。

## 区域和可用区注意事项

Amazon Redshift 在多个 AWS 区域可用。预设情况下，Amazon Redshift 在所选的 AWS 区域内随机选择的可用区 (AZ) 中预置集群。所有集群节点是在同一可用区中预置的。

您可以选择请求特定的可用区（如果 Amazon Redshift 在该区域中可用）。例如，如果您已在某个可用区中运行 Amazon EC2 实例，您可能希望在同一可用区中创建 Amazon Redshift 集群以减少延迟。另一方面，您可能希望选择另一个可用区以获得更高的可用性。Amazon Redshift 可能无法在 AWS 区域内的所有可用区中使用。

有关可以预置 Amazon Redshift 集群的支持的 AWS 区域的列表，请参阅《Amazon Web Services 一般参考》中的 Amazon Redshift 端点。

# 集群维护

Amazon Redshift 定期执行维护以升级您的集群。在此类更新期间，无法对 Amazon Redshift 集群执行常规操作。您可以通过多种方式控制维护集群的方法。例如，您可以控制将更新部署到集群的时间。您还可以选择集群是运行最近发行的版本，还是运行以前发行的版本并升级到最近发行的版本。最后，您可以选择将非强制性维护更新推迟一段时间。

## 维护时段

Amazon Redshift 针对每个 AWS 区域从 8 小时时间段中随机分配 30 分钟的维护时段，维护可能发生在包括周一至周日在内的一周中随机的一天。

默认维护时段

下方列表显示了为每个 AWS 区域分配默认维护时段的时间段。

- 美国东部（弗吉尼亚北部）区域：03:00–11:00 UTC
- 美国东部（俄亥俄）区域：03:00-11:00 UTC
- 美国西部（加利福尼亚北部）区域：06:00 -14:00 UTC
- 美国西部（俄勒冈州）区域：06:00-14:00 UTC
- 非洲（开普敦）区域：20:00-04:00 UTC
- 亚太地区（香港）区域：13:00-21:00 UTC
- 亚太（海得拉巴）区域：16:30–00:30 UTC
- 亚太地区（雅加达）区域：15:00–23:00 UTC
- 亚太地区（马来西亚）区域：14:00–22:00 UTC
- 亚太地区（墨尔本）区域：12:00–20:00 UTC
- 亚太地区（孟买）区域：16:30–00:30 UTC
- 亚太地区（新西兰）区域：10:00–18:00 UTC
- 亚太地区（大阪）区域：13:00–21:00 UTC
- 亚太地区（首尔）区域：13:00–21:00 UTC
- 亚太地区（新加坡）区域：14:00-22:00 UTC
- 亚太地区（悉尼）区域：12:00–20:00 UTC
- 亚太地区（台北）区域：14:00–22:00 UTC
- 亚太地区（泰国）区域：15:00 – 23:00 UTC
- 亚太地区（东京）区域：13:00–21:00 UTC

- 加拿大（中部）区域：03:00–11:00 UTC
- 加拿大西部（卡尔加里）区域：04:00–12:00 UTC
- 中国（北京）区域：13:00–21:00 UTC
- 中国（宁夏）区域：13:00–21:00 UTC
- 欧洲（法兰克福）区域：06:00–14:00 UTC
- 欧洲（爱尔兰）区域：22:00-06:00 UTC
- 欧洲（伦敦）区域：22:00-06:00 UTC
- 欧洲（米兰）区域：21:00-05:00 UTC
- 欧洲（巴黎）区域：23:00-07:00 UTC
- 欧洲（斯德哥尔摩）区域：23:00–07:00 UTC
- 欧洲（苏黎世）区域：20:00–04:00 UTC
- 以色列（特拉维夫）区域：20:00–04:00 UTC
- 墨西哥（中部）区域：04:00 – 12:00 UTC
- 欧洲（西班牙）区域：21:00–05:00 UTC
- 中东（巴林）区域：13:00-21:00 UTC
- 中东（阿联酋）区域：18:00–02:00 UTC
- 南美洲（圣保罗）区域：19:00–03:00 UTC

如果在指定周内安排了维护事件，则维护将在分配的 30 分钟维护时段内启动。当 Amazon Redshift 执行维护时，它会终止正在进行的任何查询或其他操作。计划维护期间出现的停机时间不计入 Amazon Redshift 服务水平协议中的月度停机时间或不可用时间。大多数维护都将在 30 分钟的维护时段内完成，但某些维护任务可能在此时段结束后继续运行。如果在计划的维护时段没有要执行的维护任务，您的集群会在下个计划维护时段到来之前继续正常运行。

您可以通过编程方式或使用 Amazon Redshift 控制台对集群进行修改来更改计划的维护时段。您可以在维护选项卡下找到维护时段，并设置集群的维护日期和时间。

集群可能在维护时段之外重启。出现这种情况有多种原因。一个更常见的原因是检测到集群存在问题，正在执行维护操作以使其恢复正常运行状态。有关更多信息，请参阅文章为什么我的 Amazon Redshift 集群在维护时段之外重启？，其中提供了有关可能发生这种情况的详细原因。

## 推迟维护

要重新计划集群的维护时段，您可以将维护最多延迟 60 天。例如，如果集群的维护时段设置为星期三 08:30 – 09:00 UTC，而您需要在该时间访问集群，则可以将维护推迟到以后的时间段。

如果您推迟维护，Amazon Redshift 仍会对您的集群应用硬件更新或其它强制性安全更新。在这些更新期间，您的集群不可用。

如果计划在即将到来的维护时段内进行硬件更新或其它强制性安全更新，Amazon Redshift 会在待处理类别下向您发送预先通知。要了解有关待处理事件通知的更多信息，请参阅Amazon Redshift 预置集群事件通知。

您还可以选择从 Amazon Simple Notification Service (Amazon SNS) 接收事件通知。有关从 Amazon SNS 订阅事件通知的更多信息，请参阅Amazon Redshift 集群事件通知订阅。

如果您推迟集群的维护，将无法推迟此推迟时段后的维护时段。

> **Note**
>
> 维护一旦开始便无法推迟。

有关集群维护的更多信息，请参阅以下文档：

- 维护时段
- 集群操作
- 修改集群

### 选择集群维护跟踪

当 Amazon Redshift 发布新的集群版本时，您的集群将在其维护时段内更新。您可以控制集群是更新到最新版本还是先前版本。

跟踪控制着将在维护时段内应用的集群版本。当 Amazon Redshift 发布新的集群版本时，该版本将分配给当前版本跟踪，上一个版本将分配给早先版本跟踪。

有关集群跟踪的信息，请参阅 Amazon Redshift 预置集群和无服务器工作组的跟踪。

## 了解 RG 和 RA3 节点如何分离计算和存储

这些章节详细介绍了可用于 RG 和 RA3 节点类型的任务，展示它们对一系列使用案例的适用性，并详细说明了相对于以前提供的节点类型的优势。

RG 和 RA3 节点的优势和可用性

RG 和 RA3 节点具有以下优点：

- 它们可以灵活地增加计算容量，而不增加存储成本。此外，它们可以扩展存储，而不会超额预置计算容量。
- 它们使用高性能 SSD 存储热数据，并使用 Amazon S3 存储冷数据。因此，它们提供了易用性、经济高效的存储以及较高的查询性能。
- 它们使用在 AWS Nitro 系统上构建的高带宽联网，以进一步减少将数据卸载到 Amazon S3 以及从中检索数据所花的时间。

在以下情况下，请考虑选择 RG 和 RA3 节点类型：

- 您需要将计算与存储分开，以灵活地扩展和支付费用。
- 您查询的数据占总数据的一小部分。
- 数据量正在迅速增长，或者预计会迅速增长。
- 您希望仅根据性能需求灵活地调整集群大小。

数据湖查询计算注意事项：在 DC2 和 RA3 预置集群上，数据湖查询在 Redshift Spectrum 上运行，而 Redshift Spectrum 使用独立于您的集群的专用 Amazon Redshift 服务器。在 RG 预置集群上，数据湖查询在集群自己的计算资源上运行，并与其它工作负载共享这些资源。对于数据湖查询用量很大的工作负载，请在调整 RG 集群规模时考虑这一点。

要使用 RG 节点类型，您的 AWS 区域必须支持 RG。有关更多信息，请参阅 AWS 区域中的 RG 节点类型可用性。

要使用 RA3 节点类型，您的 AWS 区域必须支持 RA3。有关更多信息，请参阅 AWS 区域中的 RA3 节点类型可用性。

> **Important**
>
> 只能对集群版本 1.0.21262 或更高版本使用 ra3.xlplus 节点类型。您可以使用 Amazon Redshift 控制台查看现有集群的版本。有关更多信息，请参阅 确定工作组或集群版本。
> 请确保在使用 RG 或 RA3 节点类型时使用新的 Amazon Redshift 控制台。
> 此外，要将 RG 或 RA3 节点类型与使用跟踪的 Amazon Redshift 操作结合使用，必须将维护跟踪值设置为支持 RG 或 RA3 的集群版本。有关音轨的更多信息，请参阅 选择集群维护跟踪。

使用单节点 RA3 节点类型时，请考虑以下几点。

- 支持数据共享创建者和使用者。
- 只支持经典调整大小来更改节点类型。不支持使用弹性调整大小或快照还原来更改节点类型。以下方案均可支持：
  - 将 1 节点 dc2.xlarge 经典调整为 1 节点 ra3.xlplus，反之亦然。
  - 将 1 节点 dc2.xlarge 经典调整为多节点 ra3.xlplus，反之亦然。
  - 将多节点 dc2.xlarge 经典调整为 1 节点 ra3.xlplus，反之亦然。

使用 Amazon Redshift 托管存储

借助 Amazon Redshift 托管存储，您可以在 Amazon Redshift 中存储和处理所有数据，同时可以更灵活地分别扩展计算容量和存储容量。您继续使用 COPY 或 INSERT 命令接收数据。为了优化性能并管理跨各存储层的自动数据放置，Amazon Redshift 利用诸如数据块温度、数据块使用期限和工作负载模式之类的优化功能。需要时，Amazon Redshift 自动将存储扩展到 Amazon S3，而无需任何手动操作。

有关存储成本的信息，请参阅 Amazon Redshift 定价。

管理 RG 或 RA3 节点类型

要利用将计算与存储分开的优势，您可以使用 RG 或 RA3 节点类型创建或升级集群。要使用 RG 或 RA3 节点类型，请在虚拟私有云（EC2-VPC）中创建集群。

要更改具有 RG 或 RA3 节点类型的 Amazon Redshift 集群的节点数，请执行以下操作之一：

- 使用弹性调整大小操作添加或删除节点。在某些情况下，不支持使用弹性调整大小功能从 RG 或 RA3 集群中移除节点。例如，在 2:1 节点计数升级将每个节点的切片数设置为 32 时。有关更多信息，请参阅 调整集群大小。如果弹性调整大小不可用，请使用经典调整大小。
- 使用经典调整大小操作添加或删除节点。当您将大小调整为无法通过弹性调整大小实现的配置时，请选择此选项。弹性调整大小比经典调整大小更快。有关更多信息，请参阅 调整集群大小。

## AWS 区域中的 RG 节点类型可用性

RG 节点类型仅在以下 AWS 区域中可用：

- 美国东部（弗吉尼亚北部）区域 (us-east-1)
- 美国东部（俄亥俄）区域 (us-east-2)
- 美国西部（加利福尼亚北部）区域 (us-west-1)

- 美国西部（俄勒冈州）区域 (us-west-2)
- 非洲（开普敦）区域 (af-south-1)
- 亚太地区（香港）区域 (ap-east-1)
- 亚太地区（台北）区域（ap-east-2）
- 亚太地区（东京）区域（ap-northeast-1）
- 亚太地区（首尔）区域 (ap-northeast-2)
- 亚太地区（大阪）区域 (ap-northeast-3)
- 亚太地区（孟买）区域（ap-south-1）
- 亚太（海得拉巴）区域（ap-south-2）
- 亚太地区（新加坡）区域 (ap-southeast-1)
- 亚太地区（悉尼）区域 (ap-southeast-2)
- 亚太地区（雅加达）区域 (ap-southeast-3)
- 亚太地区（墨尔本）区域（ap-southeast-4）
- 亚太地区（马来西亚）区域（ap-southeast-5）
- 亚太地区（泰国）区域（ap-southeast-7）
- 加拿大（中部）区域（ca-central-1）
- 欧洲（法兰克福）区域 (eu-central-1)
- 欧洲（苏黎世）区域（eu-central-2）
- 欧洲地区（斯德哥尔摩）区域 (eu-north-1)
- 欧洲地区（米兰）区域（eu-south-1）（仅限 rg.xlarge 和 rg.4xlarge）
- 欧洲（西班牙）区域（eu-south-2）
- 欧洲（爱尔兰）区域 (eu-west-1)
- 欧洲（伦敦）区域 (eu-west-2)
- 欧洲（巴黎）区域 (eu-west-3)
- 墨西哥（中部）区域（mx-central-1）
- 南美洲（圣保罗）区域 (sa-east-1)
- AWS GovCloud（美国东部）(us-gov-east-1)
- AWS GovCloud（美国西部）(us-gov-west-1)

# AWS 区域中的 RA3 节点类型可用性

RA3 节点类型仅在以下 AWS 区域中可用：

- 美国东部（弗吉尼亚北部）区域 (us-east-1)
- 美国东部（俄亥俄）区域 (us-east-2)
- 美国西部（加利福尼亚北部）区域 (us-west-1)
- 美国西部（俄勒冈州）区域 (us-west-2)
- 非洲（开普敦）区域 (af-south-1)
- 亚太地区（香港）区域 (ap-east-1)
- 亚太（海得拉巴）区域（ap-south-2）
- 亚太地区（雅加达）区域 (ap-southeast-3)
- 亚太地区（马来西亚）区域（ap-southeast-5）
- 亚太地区（墨尔本）区域（ap-southeast-4）
- 亚太地区（孟买）区域（ap-south-1）
- 亚太地区（新西兰）区域（ap-southeast-6）
- 亚太地区（大阪）区域 (ap-northeast-3)
- 亚太地区（首尔）区域 (ap-northeast-2)
- 亚太地区（新加坡）区域 (ap-southeast-1)
- 亚太地区（悉尼）区域 (ap-southeast-2)
- 亚太地区（台北）区域（ap-east-2）
- 亚太地区（泰国）区域（ap-southeast-7）
- 亚太地区（东京）区域（ap-northeast-1）
- 加拿大（中部）区域（ca-central-1）
- 加拿大西部（卡尔加里）区域 (ca-west-1)
- 中国（北京）区域 (cn-north-1)
- 中国（宁夏）区域 (cn-northwest-1)
- 欧洲（法兰克福）区域 (eu-central-1)
- 欧洲（苏黎世）区域（eu-central-2）
- 欧洲（爱尔兰）区域 (eu-west-1)
- 欧洲（伦敦）区域 (eu-west-2)

- 欧洲（米兰）区域 (eu-south-1)
- 欧洲（西班牙）区域（eu-south-2）
- 欧洲（巴黎）区域 (eu-west-3)
- 欧洲（斯德哥尔摩）区域 (eu-north-1)
- 以色列（特拉维夫）区域（il-central-1）
- 墨西哥（中部）区域（mx-central-1）
- 中东（巴林）区域 (me-south-1)
- 中东（阿联酋）区域（me-central-1）
- 南美洲（圣保罗）区域 (sa-east-1)
- AWS GovCloud（美国东部）(us-gov-east-1)
- AWS GovCloud（美国西部）(us-gov-west-1)

## 升级到 RG 或 RA3 节点类型

要将现有节点类型升级到 RG 或 RA3，您可以使用以下选项来更改节点类型：

- 从快照中还原：Amazon Redshift 使用集群的最新快照，并还原该快照以创建新的 RG 或 RA3 集群。在集群创建完成后（通常在几分钟内），RG 或 RA3 节点就可以立即运行全部生产工作负载。由于计算与存储分开并具有较大的网络带宽，因此，可以快速地将热数据存储到本地缓存中。如果从最新的 DC2 快照中还原，则 RG 和 RA3 保留 DC2 工作负载的热块信息，并使用最热的块填充其本地缓存。有关更多信息，请参阅 从快照还原集群。

  要为应用程序和用户保持相同的端点，可以使用与原始 DC2 集群相同的名称重命名新的 RG 或 RA3 集群。要重命名集群，请在 Amazon Redshift 控制台或 `ModifyCluster` API 操作中修改集群。有关更多信息，请参阅 Amazon Redshift API 参考中的 重命名集群 或 `ModifyCluster` API 操作。
- 弹性调整大小 – 使用弹性调整大小调整集群大小。在使用弹性调整大小更改节点类型时，Amazon Redshift 自动创建快照，创建新的集群，删除旧集群并重命名新集群。可以按需运行弹性调整大小操作，也可以计划在将来的时间运行。您可以使用弹性调整大小功能将现有的 DC2 节点类型集群快速升级到 RG 或 RA3。有关更多信息，请参阅 弹性调整大小。

> **Note**
>
> 要将现有 RA3 集群迁移到 rg.xlarge 或 rg.4xlarge，源集群必须运行补丁版本 P201 或更高版本。要迁移到 rg.large 或 rg.12xlarge，源集群必须运行 P202 或更高版本。要创建或迁移到单

节点 rg.large 集群，您的集群必须运行 P204 或更高版本。无论补丁版本如何，DC2 集群都可以迁移到 RG。

下表显示了在升级到 RG 节点类型时的建议。

此表中的建议是起始集群节点类型和大小，具体取决于工作负载的计算要求。RG 具有不同的性能特征，因此建议的映射仅作为起点。对您的关键工作负载进行基准测试，并根据观察到的性能上下调整节点数量。RG 实例的每 vCPU 价格低于 RA3，因此如果添加节点来优化性能，整体成本仍有可能更低。有关节点类型定价的更多信息，请参阅 Amazon Redshift 定价。

为了更好地估计需求，您可以考虑进行概念验证（POC），使用试用来运行潜在的配置。为您的 POC 数据仓库而不是 Redshift Serverless 预置集群。有关进行概念验证的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的为 Amazon Redshift 执行概念验证（POC）。

| 现有的节点类型 | 现有节点数 | 建议的新节点类型 | 升级操作 |
| --- | --- | --- | --- |
| ra3.16xlarge | 2–128 | rg.12xlarge | 开始时，每 1 个 ra3.16xlarge 节点对应 1 个 rg.12xlarge 节点[1]。 |
| ra3.4xlarge | 2–64 | rg.4xlarge | 开始时，每 4 个 ra3.4xlarge 节点对应 3 个 rg.4xlarge 节点[1]。 |
| ra3.xlplus | 1 | rg.large | 开始时，单节点 ra3.xlplus 集群对应 2 个 rg.large 节点[1、2]。 |
| ra3.xlplus | 2–32 | rg.xlarge | 开始时，每 1 个 ra3.xlplus 节点对应 1 个 rg.xlarge 节点[1]。 |
| ra3.large | 1 – 16 | rg.large | 开始时，每 1 个 ra3.large 节点对应 1 个 rg.large 节点[1、2]。 |

| 现有的节点类型 | 现有节点数 | 建议的新节点类型 | 升级操作 |
| --- | --- | --- | --- |
| dc2.8xlarge | 2–15 | rg.4xlarge | 开始时，每 2 个 dc2.8xlarge 节点对应 3 个 rg.4xlarge 节点[1]。 |
| dc2.8xlarge | 16–128 | rg.12xlarge | 开始时，每 2 个 dc2.8xlarge 节点对应 1 个 rg.12xlarge 节点[1]。 |
| dc2.large | 1–3 | rg.large | 开始时，每 1 个 dc2.large 节点对应 1 个 rg.large 节点[1、2]。 |
| dc2.large | 4 | rg.large | 以 3 个 rg.large 节点开始[1]。 |
| dc2.large | 5–15 | rg.xlarge | 开始时，每 8 个 dc2.large 节点对应 3 个 rg.xlarge 节点[1]。 |
| dc2.large | 16 – 32 | rg.4xlarge | 开始时，每 10 个 dc2.large 节点对应 1 个 rg.4xlarge 节点[1]。 |

[1]根据工作负载要求，可能需要使用额外的节点。请根据所需的查询性能的计算要求添加或删除节点。RG 实例的每 vCPU 价格低于 RA3，因此如果添加节点，与之前的配置相比，整体成本仍有可能更低。有关更多信息，请参阅 Amazon Redshift 定价。

[2]有关单节点集群的信息，请参阅升级到 RG 节点时应考虑的事项下的单节点集群。

下表显示了在升级到 RA3 节点类型时的建议。

此表中的建议是起始集群节点类型和大小，具体取决于工作负载的计算要求。为了更好地估计需求，您可以考虑进行概念验证（POC），使用试用来运行潜在的配置。为您的 POC 数据仓库而不是 Redshift Serverless 预置集群。有关进行概念验证的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的为 Amazon Redshift 执行概念验证（POC）。

| 现有的节点类型 | 现有节点数 | 建议的新节点类型 | 升级操作 |
| --- | --- | --- | --- |
| dc2.8xlarge | 2–15 | ra3.4xlarge | 开始时，每 1 个 dc2.8xlarge[1] 节点对应 2 个 ra3.4xlarge 节点。 |
| dc2.8xlarge | 16–128 | ra3.16xlarge | 开始时，每 2 个 dc2.8xlarge[1] 节点对应 1 个 ra3.16xlarge 节点。 |
| dc2.large | 1–4 | ra3.large | 开始时，每 1 个 dc2.large[1] 节点对应 1 个 ra3.large 节点。<br><br>开始时，每 2 个 dc2.large[1] 节点对应 2 个 ra3.large 节点。<br><br>开始时，每 3 个 dc2.large[1] 节点对应 3 个 ra3.large 节点。<br><br>开始时，每 4 个 dc2.large[1] 节点对应 3 个 ra3.large 节点。 |
| dc2.large | 5–15 | ra3.xlplus | 开始时，每 8 个 dc2.large[1] 节点对应 3 个 ra3.xlplus 节点。 |
| dc2.large | 16 – 32 | ra3.4xlarge | 开始时，每 8 个 dc2.large[12] 节点对应 1 个 ra3.4xlarge 节点。 |

[1]根据工作负载要求，可能需要使用额外的节点。请根据所需的查询性能的计算要求添加或删除节点。

[2]具有 dc2.large 节点类型的集群仅限 32 个节点。

某些 RA3 节点类型的最小节点数为 2 节点。在创建 RA3 集群时，请考虑到这一点。

## RG 和 RA3 节点支持的网络功能

RG 和 RA3 节点支持一组在其它节点类型中不可用的联网功能。本节简要介绍了每项功能并提供了指向其他文档的链接：

- 预置集群 VPC 端点：创建或还原 RG 或 RA3 集群时，Amazon Redshift 使用 5431-5455 或 8191-8215 范围内的端口。当集群设置为其中一个范围内的端口时，Amazon Redshift 会自动在您的 AWS 账户中为该集群创建一个 VPC 端点，并向其附加一个私有 IP 地址。如果您将集群设置为可公开访问，Redshift 会在您的 AWS 账户中创建一个弹性 IP 地址并将其附加到 VPC 端点。有关更多信息，请参阅为 Amazon Redshift 集群或 Amazon Redshift Serverless 工作组配置安全组通信设置。
- 单子网 RG 或 RA3 集群：您可以创建具有单个子网的 RG 或 RA3 集群，但它不能使用灾难恢复功能。如果您在子网没有多个可用区（AZ）时启用集群重新定位，则会出现异常。
- 多子网 RG 或 RA3 集群和子网组：在虚拟私有云（VPC）中预置集群时，您可以通过创建子网组来创建包含多个子网的 RG 或 RA3 集群。您可以通过集群子网组在 VPC 中指定一组子网，Amazon Redshift 可在其中一个子网中创建集群。创建子网组之后，您可以删除之前添加的子网，也可以添加更多子网。有关更多信息，请参阅 Amazon Redshift 集群子网组。
- 跨账户或跨 VPC 端点访问 – 您可以通过设置 Redshift 托管的 VPC 端点来访问预置集群或 Amazon Redshift Serverless 工作组。例如，您可以将其设置为包含集群或工作组的 VPC 与运行客户端工具的 VPC 之间的私有连接。这样，您就可以在不使用公有 IP 地址和不通过互联网路由流量的情况下访问数据仓库。有关更多信息，请参阅使用 Redshift 托管的 VPC 端点。
- 集群重新定位 – 当服务中断时，您可以将集群移动到另一个可用区（AZ），而不会丢失任何数据。在控制台上启用此功能。有关更多信息，请参阅 重新定位集群。
- 自定义域名 – 您可以为 Amazon Redshift 集群创建自定义域名，也称为自定义 URL。这是一条易于阅读的 DNS 记录，可将 SQL 客户端连接路由到您的集群端点。有关更多信息，请参阅 用于客户端连接的自定义域名。

## 升级到 RG 节点时应考虑的事项

修补版本

要从 RA3 迁移到 RG，源集群必须满足所要求的最低补丁版本：对于 rg.xlarge 和 rg.4xlarge 为 P201，对于多节点 rg.large 和 rg.12xlarge 为 P202，对于单节点 rg.large 为 P204。无论补丁版本如何，DC2 集群都可以迁移到 RG。在迁移之前，通过将快照还原到 RG 集群来测试您的工作负载，之

后转移生产流量。有关详细信息，请参阅 Amazon Redshift 补丁 201、Amazon Redshift 补丁 202 和 Amazon Redshift 补丁 204。

单节点集群

- 只有 rg.large 支持单节点 RG 集群。
- 单节点 rg.large 集群最多支持 1 TB 的托管存储，而单节点 ra3.xlplus 集群最高支持 4 TB。如果您的数据量大于 1 TB 或预计会超出 1 TB，请使用多节点 rg.large 集群。
- 单节点集群专为开发和测试工作负载而设计。对于生产工作负载，请至少使用两个节点。有关更多信息，请参阅Amazon Redshift 中的集群和节点和节点类型详细信息。

Spectrum 工作负载

工作负载大量使用 Spectrum 的集群在迁移到 RG 后可能会出现查询性能变慢的情况。与 RA3 和 DC2 不同，RG 直接在集群计算节点上运行集成式数据湖查询引擎，而不是在单独的 Spectrum 实例集上运行。如果您遇到 Spectrum 性能下降的情况，请增加 RG 集群中的节点数量或迁移到更大的 RG 节点类型。

RG 集群还支持查询位于不同 AWS 区域的 Amazon S3 存储桶中的数据，从而消除了适用于使用 Spectrum 的 RA3 和 DC2 集群的同区域限制。跨区域查询需要支付额外的数据传输费用。有关更多信息，请参阅查询数据湖。

基于游标的工作负载

与 RA3 和 DC2 相比，RG 节点支持的游标大小不同。有关游标大小限制，请参阅游标约束。如果您的工作负载严重依赖于游标，并且您遇到游标大小限制，请迁移到更大的 RG 节点类型：您可以按比例减少节点数量，以保持相似的集群总大小和成本。

并发扩展资格

如果集群使用并发扩展且拥有超过 32 个节点，并且您想要迁移到 rg.12xlarge，请使用弹性调整大小或快照还原来维持对并发扩展的支持。如果集群拥有超过 32 个节点，并且您执行经典调整大小以调整到 rg.12xlarge，则集群将失去并发扩展支持。新创建的拥有超过 32 个节点的 rg.12xlarge 集群不支持并发扩展。有关更多信息，请参阅并发扩展候选项。

预留节点

如果您的集群使用预留节点，则更改节点类型不会自动更改您的预留。现有预留将与原始节点类型绑定，直到预留期限结束。在迁移使用预留节点的集群之前，请参阅在使用预留实例的集群上更改节点类型之前需要考虑的事项。

# 集群操作

创建集群后，可以执行集群操作，以优化性能、控制成本并确保高可用性。借助集群操作，可以根据数据仓库需求的变化，调整集群大小、暂停集群、恢复集群甚至重新创建集群。

常见的用例包括为峰值工作负载扩展计算容量，在非活动期间暂停集群以降低成本，以及使用不同的配置或在不同的可用区中重新创建集群以进行灾难恢复。以下各节详细介绍了如何执行各种集群操作，以有效地管理 Amazon Redshift 环境。

## 创建集群

借助 Amazon Redshift，可以创建预置集群来启动新的数据仓库。预置集群是由称作节点的各种计算资源构成的集合，这些节点已整理到单个大规模并行处理（MPP）系统中。

在创建集群之前，请参阅Amazon Redshift 预置集群和Amazon Redshift 中的集群和节点。

创建集群

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。这将列出您的账户在当前 AWS 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。
3. 选择创建集群以创建集群。
4. 按照控制台页面上的说明进行操作，为集群配置输入属性。

   以下步骤介绍了 Amazon Redshift 控制台，该控制台在支持 RG 或 RA3 节点类型的 AWS 区域中运行。有关支持 RG 或 RA3 节点类型的 AWS 区域列表，请参阅《Amazon Redshift 管理指南》中的 AWS 区域中的 RG 节点类型可用性和 AWS 区域中的 RA3 节点类型可用性。

   如果您不知道要将集群调整到多大，请选择 Help me choose（帮我选择）。执行此操作将启动大小调整计算器，该计算器将询问您有关计划存储在数据仓库中的数据的大小和查询特性的问题。如果您知道集群所需的大小（即节点类型和节点数），请选择我会选择。然后选择节点类型和节点数量来调整集群的大小以进行概念验证。

   > **Note**
   >
   > 如果您的组织符合条件，而在您创建集群的 AWS 区域中，Amazon Redshift Serverless 不可用，则您也许可以通过 Amazon Redshift 免费试用计划创建集群。请选择生产或免费试用来回答问题您打算将此集群用于什么？ 选择免费试用时，您将创建具有 ra3.large 节

点类型的配置。有关选择免费试用的更多信息，请参阅 Amazon Redshift 免费试用。有关提供了 Amazon Redshift Serverless 的 AWS 区域的列表，请参阅《Amazon Web Services 一般参考》中针对 Redshift Serverless API 列出的端点。

5. 在数据库配置部分中，为管理员用户名指定值。对于管理员密码，您可以从以下选项中进行选择：
   - 生成密码 – 使用 Amazon Redshift 生成的密码。
   - 手动添加管理员密码 – 使用您自己的密码。
   - 管理 AWS Secrets Manager 中的管理员凭证 – Amazon Redshift 使用 AWS Secrets Manager 生成和管理您的管理员密码。使用 AWS Secrets Manager 生成和管理您密码的密钥会产生一定的费用。有关 AWS Secrets Manager 定价的信息，请参阅 AWS Secrets Manager 定价。
6. （可选）按照控制台页面上的说明进行操作，为集群权限输入属性。如果您的集群需要为您访问其他 AWS 服务（例如，从 Amazon S3 加载数据），请提供集群权限。
7. 选择创建集群以创建集群。集群可能需要几分钟才可以使用。

## 其他配置

在创建一个集群时，可以指定其他属性来自定义该集群。您可以在下面的列表中找到有关这些属性中的某些属性的更多详细信息。

IP 地址类型

选择您集群的 IP 地址类型。您可以选择让资源仅通过 IPv4 寻址协议进行通信，也可以选择双栈模式，让您的资源通过 IPv4 和 IPv6 进行通信。此功能现已在 AWS GovCloud（美国东部）和 AWS GovCloud（美国西部）区域提供。有关 AWS 区域的更多信息，请参阅区域和可用区。

Virtual Private Cloud (VPC)

选择具有集群子网组的 VPC。集群在创建之后，无法更改集群子网组。

参数组

选择一个集群参数组，以便与集群相关联。如果您不选择参数组，集群将使用默认参数组。

加密

选择是否要加密集群内的所有数据及其快照。如果保留默认设置 None，将不会启用加密功能。如果您要启用加密功能，请选择您是使用 AWS Key Management Service (AWS KMS)，还是使用硬件安全模块 (HSM)，然后配置相关设置。有关 Amazon Redshift 中的加密功能的更多信息，请参阅Amazon Redshift 数据库加密。

- KMS

  如果您想启用加密并使用 AWS KMS 来管理您的加密密钥，请选择使用 AWS Key Management Service (AWS KMS)。此外还需选择要使用的密钥。您可以选择原定设置密钥、当前账户中的密钥，或其他账户中的密钥。

  > **Note**
  >
  > 如果想使用来自其他 AWS 账户的密钥，则输入要使用的密钥的 Amazon 资源名称 (ARN)。您必须拥有使用此密钥的权限。有关在 AWS KMS 中访问密钥的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的控制对您的密钥的访问。

  有关使用 Amazon Redshift 中的 AWS KMS 加密密钥的更多信息，请参阅使用 AWS KMS 的加密。

- HSM

  如果您要启用加密功能并使用硬件安全模块 (HSM) 来管理加密密钥，请选择 HSM。

  如果选择 HSM，请从 HSM 连接和 HSM 客户端证书中选择相应值。Amazon Redshift 和 HSM 需要使用这些值来构建受信任的连接，以便通过此连接传输集群密钥。必须先在 Amazon Redshift 中设置 HSM 连接和客户端证书，之后再启动集群。有关设置 HSM 连接和客户端证书的更多信息，请参阅使用硬件安全模块的加密。

维护跟踪

您可以选择使用的集群版本为当前版本、早先版本 或（有时）预览版跟踪。

监控

可以选择是否创建 CloudWatch 警报。

配置跨区域快照

您可以选择是否启用跨区域快照。

自动快照保留期

您可以选择这些快照的保留天数（最多 35 天）。如果节点类型为 DC2，则可以选择零（0）天，以便不创建自动快照。

手动快照保留期

您可以选择天数或 `Indefinitely` 来保留这些快照。

用于自动优化的额外计算资源

您可以选择是否分配额外的计算资源来执行自动优化，即使在用量大的时段也是如此。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的分配额外的计算资源来执行自动数据库优化。

## 创建磁盘空间警报

您可以监控磁盘空间使用情况，并设置警报，以便在磁盘空间超过集群的指定阈值时收到通知。通过创建磁盘空间使用情况警报，可以主动管理存储容量，并防止因磁盘空间不足而导致的问题，例如查询失败或数据摄取错误。以下过程将为您演练创建磁盘空间使用情况警报的过程。

为集群创建磁盘空间用量警报

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择警报。
3. 对于操作，选择创建警报。此时将显示创建警报页面。
4. 按照页面上的说明操作。
5. 选择创建警报。

## 查看集群

通过查看集群，可以监控和管理集群的配置、状态和性能指标。通过查看集群详细信息，可以深入了解资源利用率、查询执行时间和系统运行状况。以下过程演示如何访问集群信息。

查看集群

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。这将列出您的账户在当前 AWS 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。如果您没有集群，请选择创建集群来创建一个集群。
3. 在列表中选择集群名称可查看有关集群的更多详细信息。

# 修改集群

当您修改集群时，对以下选项做出的更改将立即生效：

- VPC 安全组
- 公开访问
- 管理员用户密码
- HSM 连接
- HSM 客户端证书
- 维护详细信息
- 快照首选项

对以下选项做出的更改只能在重启集群后才会生效：

- 集群标识符

  当您更改集群标识符时，Amazon Redshift 会自动重新启动集群。
- 增强型 VPC 路由

  当您更改增强型 VPC 路由时，Amazon Redshift 会自动重新启动集群。
- 集群参数组
- IP 地址类型

  此功能现已在 AWS GovCloud（美国东部）和 AWS GovCloud（美国西部）区域提供。有关 AWS 区域的更多信息，请参阅区域和可用区。

如果您缩短自动快照的保留期，系统将删除保留期设置超出了新保留期的现有自动快照。有关更多信息，请参阅 Amazon Redshift 快照和备份。

有关集群属性的更多信息，请参阅其他配置。

修改集群

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。
3. 选择要修改的集群。

4. 选择编辑。此时将显示编辑集群页面。
5. 更新集群属性。您可以修改的一些属性包括：

   - 集群标识符
   - 快照保留
   - 集群重新定位

   控制台提供了指向相应集群详细信息选项卡的链接，用于编辑网络和安全、维护和数据库配置的设置。
6. 选择保存更改。

## 调整集群大小

随着您的数据仓库容量和性能需求的变化，您可以调整集群大小，以便充分利用 Amazon Redshift 的计算和存储选项。

调整集群大小时，您可以指定与集群当前配置不同的节点数量或节点类型。当集群处于调整大小过程中时，您将无法在集群上运行任何写入或读取/写入查询；您只可以运行读取查询。

有关调整集群大小的更多信息（包括使用不同方法调整集群大小的过程），请参阅调整集群大小。

调整集群大小

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。
3. 选择要调整大小的集群。
4. 对于操作，选择调整大小。此时将显示调整集群大小页面。
5. 按照页面上的说明操作。您可以立即调整集群大小，在特定时间调整一次，或者按计划增加和减小集群大小。
6. 根据您的选择，选择立即调整大小或计划调整大小。

如果您有预留节点，则可以升级到 RG 或 RA3 预留节点。使用控制台从快照还原或执行弹性调整大小时，您可以执行此操作。您可以使用控制台引导您完成此过程。有关升级到 RG 或 RA3 节点类型的更多信息，请参阅升级到 RG 或 RA3 节点类型。

当您执行调整大小操作以便从 DC2.large 节点类型升级到 RA3.large 节点类型时，Amazon Redshift 会自动将交错排序键转换为复合排序键。这种转换支持访问并发扩展功能，该功能不支持对具有交错排序键的表进行查询。虽然这种自动转换可确保与 RA3 功能的兼容性，但它可能会影响您现有的查询性能模式。

如果要在升级到 RA3 节点后保留交错排序键，则可以在调整大小操作完成后，使用所需的排序键配置来重新创建表。但是，选择此选项意味着您将无法对这些表使用并发扩展。

调整大小操作有两种类型：

- 弹性调整大小：您可以向集群添加节点，或从集群中移除节点。您还可以更改节点类型，如从 DC2 节点更改为 RG 或 RA3 节点。弹性调整大小通常会很快完成，平均约需十分钟。因此，我们建议将其作为首选项。当您执行弹性调整大小时，它会重新分发数据切片，这些数据切片是在每个节点中分配内存和磁盘空间的分区。弹性调整大小适用于以下情况：
  - 增加或减少现有集群中的节点，但不更改节点类型：这通常被称为就地调整大小。当您执行这种调整大小时，一些正在运行的查询会成功完成，但其他查询可能会作为该操作的组成部分而被删除。
  - 更改集群的节点类型：当您更改节点类型时，将创建快照，并将数据从源集群重新分发到由新节点类型组成的集群。完成后，将删除正在运行的查询。与就地调整大小一样，它很快即可完成。
- 经典调整大小：您可以更改节点类型、节点数量或此两者，其方式与弹性调整大小相似。经典调整大小需要更多时间才能完成，但在节点计数发生变化或要迁移到的节点类型不在弹性调整大小范围内的情况下，它会很有用。例如，当节点计数的变化非常大时，此调整大小类型可能适用。

主题


- 弹性调整大小
- 经典调整大小


## 弹性调整大小

当您添加或移除相同类型的节点时，弹性调整大小操作包括以下阶段：

1. 弹性调整大小操作将为集群制作快照。仅 DC2 节点支持无备份表。对于 RG 和 RA3 集群，快照中包含无备份表。有关更多信息，请参阅 从快照中排除表。如果您的集群因禁用了自动快照而没有最近的快照，则备份操作可能会花费更长时间。（要最大程度地减少调整大小操作开始前的时间，我们建议您在开始调整大小操作之前启用自动快照或创建手动快照。）如果在您启动弹性调整大小时，系统正在执行快照操作，若该快照操作未能在数分钟内完成，则调整大小可能会失败。有关更多信息，请参阅 Amazon Redshift 快照和备份。

2. 该操作将迁移集群元数据。集群将在数分钟内不可用。大多数查询将暂停，并且连接将保持打开状态。但是，某些查询可能会被删除。此阶段很短。
3. 会话连接将恢复，并且查询将继续。
4. 弹性调整大小操作会在后台将数据重新分发到节点切片。集群可用于读取和写入操作，但是某些查询可能需要更长的时间来运行。
5. 在该操作完成后，Amazon Redshift 会发送一个事件通知。

当您使用弹性调整大小来更改节点类型时，它的工作方式与您增加或减少相同类型的节点时相似。首先，创建一个快照。使用该快照中的最新数据预调配新的目标集群，并在后台将数据传输到新集群。在此期间，数据是只读的。在大小调整接近完成时，Amazon Redshift 会将端点更新为指向新集群，并断开与源集群的所有连接。

弹性调整大小操作不太可能失败。但如果失败，在大多数情况下会自动进行回滚，无需任何手动干预。

如果您有预留节点，例如 DC2 预留节点，则您可以在执行调整大小时升级到 RA3 预留节点。您可以在执行弹性调整大小或者使用控制台从快照还原时执行此操作。控制台将引导您完成此过程。有关升级到 RG 或 RA3 节点类型的更多信息，请参阅升级到 RG 或 RA3 节点类型。

弹性调整大小操作不会对表进行排序或回收磁盘空间，因此，它不能替代 vacuum 操作。有关更多信息，请参阅对表执行 vacuum 操作。

弹性调整大小操作具有以下限制：

- 弹性大小调整和数据共享集群 - 如果您在作为数据共享创建者的集群上添加或缩减节点，则当 Amazon Redshift 迁移集群元数据时，您无法从使用者连接到该集群。同样，如果您执行弹性大小调整并选择新的节点类型，则当连接断开并转移到新的目标集群时，数据共享将不可用。在这两种类型的弹性大小调整中，创建者会有几分钟时间不可用。
- 从共享快照传输数据：要在从共享快照传输数据的集群上运行弹性调整大小，必须至少有一个备份可供该集群使用。您可以在 Amazon Redshift 控制台快照列表中或通过 `describe-cluster-snapshots` CLI 命令或 `DescribeClusterSnapshots` API 操作查看备份。
- 平台限制：弹性调整大小仅可用于使用 EC2-VPC 平台的集群。有关更多信息，请参阅 使用 EC2 创建集群。
- 存储注意事项：确保新的节点配置有足够的存储空间来存储现有数据。您可能需要添加其他节点或更改配置。
- 源与目标集群大小 – 可以通过弹性调整大小来调整大小的节点的数量和节点类型由源集群中的节点的数量和为已调整大小的集群所选的节点类型来确定。可以使用控制台确定可能的配置。您也可以

运行带 `action-type resize-cluster` 选项的 `describe-node-configuration-options` AWS CLI 命令。有关使用 Amazon Redshift 控制台进行大小调整的更多信息，请参阅调整集群大小。

以下示例 CLI 命令描述了可用的配置选项。在此示例中，名为 `mycluster` 的集群是一个 8 节点的 `rg.4xl` 集群。

```
aws redshift describe-node-configuration-options --cluster-identifier mycluster --
region eu-west-1 --action-type resize-cluster
```

此命令会返回一个选项列表，包括每个选项的建议节点类型、节点数和磁盘利用率。返回的配置根据特定的输入集群而不同。指定 `resize-cluster` CLI 命令的选项时，可以选择其中一种返回的配置。

- 其他节点的上限：弹性调整大小对您可以添加到集群的节点有限制。例如，RG 或 RA3 集群支持弹性调整大小，最多可达节点数量的四倍。为了进行说明，您可以将一个节点添加到 4 节点 rg.4xl 集群中，使其成为 5 节点集群，或者添加更多节点，直到达到 16 个为止。

  > **Note**
  >
  > 增长和减少限制基于原始节点类型，以及原始集群中的节点数量或其上次经典大小调整。如果弹性调整大小将超过增长或减少限制，则使用经典大小调整。

  使用某些 rg 和 ra3 节点类型，您最多可以将节点数增加到现有计数的四倍。具体来说，假设您的集群由 ra3.4xlarge 或 ra3.16xlarge 节点组成。然后，您可以使用弹性调整大小将 8 节点集群中的节点数增加到 32。或者，您可以选择低于限制的值。（请记住，将集群增长 4 倍的能力取决于源集群的大小。）如果您的集群具有 ra3.xlplus 节点，则限制为双倍。

  所有 rg 和 ra3 节点类型都支持将节点数减少到现有计数的四分之一。例如，您可以将包含 ra3.4xlarge 节点的集群的大小从 12 个节点减少到 3 个节点，或者减少到高于最小值的数字。

  下表列出了支持弹性调整大小的每个节点类型的增长和减少限制。

| 原始节点类型 | 增长限制 | 减少限制 |
| --- | --- | --- |
| rg.12xlarge | 4 倍 | 到数量的四分之一 |
| rg.4xlarge | 4 倍 | 到数量的四分之一 |

| 原始节点类型 | 增长限制 | 减少限制 |
| --- | --- | --- |
| rg.xlarge | 2 倍 | 到数量的一半 |
| rg.large | 2 倍 | 到数量的一半 |
| ra3.16xlarge | 4 倍（例如，从 4 个节点到 16 个节点） | 到数量的四分之一（例如，从 16 个节点到 4 个节点） |
| ra3.4xlarge | 4 倍 | 到数量的四分之一 |
| ra3.xlplus | 2 倍（例如，从 4 个节点到 8 个节点） | 到数量的四分之一 |
| ra3.large | 2 倍 | 到数量的一半 |
| dc2.8xlarge | 2 倍 | 到数量的一半（例如，从 16 个节点到 8 个节点） |
| dc2.large | 2 倍 | 到数量的一半 |

> **Note**
>
> 调整 RG 或 RA3 集群大小时选择传统节点类型：如果您尝试将包含 RG 或 RA3 节点的集群的大小调整为另一种节点类型，例如 DC2，则控制台中会显示一条验证警告消息，并且调整大小操作将无法完成。之所以会出现这种情况，是因为不支持将大小调整为传统节点类型。这是为了防止客户将大小调整为已过时或即将弃用的节点类型。这同时适用于弹性大小调整和经典大小调整。

## 经典调整大小

经典调整大小操作用于处理集群大小或节点类型的更改不受弹性调整大小支持的应用场景。当您执行经典调整大小时，Amazon Redshift 会创建一个目标集群，并将您的数据和元数据从源集群迁移到该集群。

对 RG 或 RA3 进行经典调整大小可以提供更好的可用性

当目标节点类型为 RG 或 RA3 时，经典调整大小功能已得到增强。它通过在源集群与目标集群之间使用备份和还原操作来实现这一目标。开始调整大小时，源集群将重新启动并在数分钟内不可用。在此之后，集群可用于读取和写入操作，同时继续在后台调整大小。

检查集群

为确保在对 RG 或 RA3 集群执行经典调整大小时获得最佳性能和结果，请完成此检查清单。如果您未遵循此检查清单，则可能无法获得对 RG 或 RA3 节点进行经典调整大小的一些好处，例如可以执行读写操作。

1. 数据的大小必须小于 2 PB。（1 PB 等于 1000 TB。）要验证数据的大小，请创建快照并检查其大小。您还可以运行以下查询来检查大小：

   ```
   SELECT
   sum(case when lower(diststyle) like ('%key%') then size else 0 end) distkey_blocks,
   sum(size) as total_blocks,
   ((distkey_blocks/(total_blocks*1.00)))*100 as Blocks_need_redist
   FROM svv_table_info;
   ```

   svv_table_info 表仅对超级用户可见。
2. 在发起经典调整大小操作之前，请确保您拥有的手动快照是新鲜的，未久于 10 个小时。否则，请创建快照。
3. 用于执行经典调整大小的快照不能用于表还原或其他用途。
4. 集群必须位于 VPC 中。

对 RG 或 RA3 进行经典调整大小产生的排序和分配操作

在对 RG 或 RA3 进行经典大小调整期间，对于具有 KEY 分配但以 EVEN 分配方式迁移的表，会将其转换回其原始分配方式。此过程的持续时间取决于数据的大小和集群的繁忙程度。为查询工作负载分配更高的优先级，使其优先于数据迁移。有关更多信息，请参阅分配方式。在此迁移过程中，数据库可以读取和写入，但查询可能需要更长的时间才能完成。但在此期间，并发扩展可以通过为查询工作负载添加资源来提升性能。您可以查看 SYS_RESTORE_STATE 和 SYS_RESTORE_LOG 视图的结果来了解数据迁移的进度。有关监控的更多信息如下所示。

完全调整集群大小后，会出现以下排序行为：

- 如果调整大小导致集群有更多切片，则 KEY 分配的表将变为部分未排序状态，但 EVEN 分配的表将保持已排序状态。此外，调整大小之后立即显示的已排序数据量信息可能不是最新的。恢复密钥后，自动 vacuum 会随着时间的推移对表进行排序。
- 如果调整大小导致集群拥有的切片减少，则 KEY 分配的表和 EVEN 分配的表都将变为部分未排序状态。自动 vacuum 会随着时间的推移对表进行排序。

有关自动表 vacuum 的更多信息，请参阅对表执行 vacuum 操作。有关计算节点中的切片的更多信息，请参阅数据仓库系统架构。

目标集群为 RG 或 RA3 时的经典调整大小步骤

当目标集群类型为 RG 或 RA3 并且您已满足在上一节中详述的先决条件时，经典调整大小包含以下步骤。

1. 从源集群发起到目标集群的迁移。当预调配新的目标集群时，Amazon Redshift 发送一个事件通知，告知大小调整已开始。它会重新启动现有集群，这会关闭所有连接。如果您的现有集群是数据共享创建者集群，则与使用者集群的连接也会关闭。重新启动需要几分钟时间。
2. 重新启动后，数据库可用于读取和写入。此外，数据共享恢复，这又需要几分钟的时间。
3. 数据迁移到目标集群。当目标节点类型为 RG 或 RA3 时，在数据迁移期间可以进行读取和写入。
4. 在调整大小过程即将完成时，Amazon Redshift 会将端点更新到目标集群，并将删除与源集群的所有连接。目标集群成为数据共享的创建者。
5. 调整大小的过程完成。Amazon Redshift 发送事件通知。

您可以在 Amazon Redshift 控制台上查看调整大小的进度。调整集群大小所用的时间取决于数据量。

> **Note**
>
> 调整 RG 或 RA3 集群大小时选择传统节点类型：如果您尝试将包含 RG 或 RA3 节点的集群的大小调整为另一种节点类型，例如 DC2，则控制台中会显示一条验证警告消息，并且调整大小操作将无法完成。之所以会出现这种情况，是因为不支持将大小调整为传统节点类型。这是为了防止客户将大小调整为已过时或即将弃用的节点类型。这同时适用于弹性大小调整和经典大小调整。

当目标集群为 RG 或 RA3 时监控经典调整大小

要监控预置集群正在进行的经典调整大小，包括 KEY 分配，请使用 SYS_RESTORE_STATE。它显示了正在转换的表的完成百分比。您必须是超级用户才能访问数据。

删除在执行经典调整大小时不需要的表。这样做时，可以更快地分配现有的表。

目标集群不是 RG 或 RA3 时的经典调整大小步骤

当目标节点类型为 RG 或 RA3 以外的任何类型（例如 DC2）时，经典调整大小包含以下步骤。

1. 从源集群发起到目标集群的迁移。当预调配新的目标集群时，Amazon Redshift 发送一个事件通知，告知大小调整已开始。它会重新启动现有集群，这会关闭所有连接。如果您的现有集群是数据共享创建者集群，则与使用者集群的连接也会关闭。重新启动需要几分钟时间。

   请注意，在经典调整大小期间，不会保留使用 BACKUP NO 创建的任何数据库关系，例如表或实体化视图。有关更多信息，请参阅 CREATE MATERIALIZED VIEW。
2. 重新启动后，数据库以只读形式可用。数据共享恢复，这又需要几分钟的时间。
3. 数据迁移到目标集群。数据库保持只读状态。
4. 在调整大小过程即将完成时，Amazon Redshift 会将端点更新到目标集群，并将删除与源集群的所有连接。目标集群成为数据共享的创建者。
5. 调整大小的过程完成。Amazon Redshift 发送事件通知。

您可以在 Amazon Redshift 控制台上查看调整大小的进度。调整集群大小所用的时间取决于数据量。

> **Note**
>
> 如果目标集群不是 RG 或 RA3，或者它不符合在上一节中详述的 RA3 目标集群的先决条件，则可能需要几天甚至几周的时间，才能调整包含大量数据的集群的大小。
> 另请注意，在进行了经典调整大小后，集群的已用存储容量可能会增加。当集群有由于经典调整大小而产生的额外数据切片时，这是正常的系统行为。即使集群中的节点数量保持不变，也可能会发生这种使用额外容量的情况。

弹性调整大小与经典调整大小的对比

下表对比了两种调整大小类型之间的行为。

| 行为 | 弹性调整大小 | 经典调整大小 | 评论 |
| --- | --- | --- | --- |
| 系统数据留存 | 弹性调整大小将保留系统日志数据。 | 经典调整大小不会保留系统表和数据。 | 如果您在源集群中启用了审计日志记录，则您可以在调整大小后继续访问 Amazon S3 或 CloudWatch 中的日志。您可以根据数据策略的规定保留或删除这些日志。 |

| 行为 | 弹性调整大小 | 经典调整大小 | 评论 | | | | |
|---|---|---|---|---|---|---|---|
| 更改节点类型 | 当节点类型不变时弹性调整大小：就地调整大小，并将保留大多数查询。<br><br>在选择新节点类型的情况下弹性调整大小：将创建新集群。在调整大小过程完成后，将删除查询。 | 经典调整大小：将创建新集群。在调整大小过程中将删除查询。 | | | | | |

| 行为 | 弹性调整大小 | 经典调整大小 | 评论 | | | | |
|---|---|---|---|---|---|---|---|
| 会话和查询留存 | 当源集群和目标集群中的节点类型相同时，弹性调整大小将保留会话和查询。如果您选择新的节点类型，则将删除查询。 | 经典调整大小不会保留会话和查询。将删除查询。 | 在删除查询后，预计会出现某种性能下降的情况。最好是在轻度使用期间执行调整大小操作。 | | | | |

| 行为 | 弹性调整大小 | 经典调整大小 | 评论 | | | | |
|---|---|---|---|---|---|---|---|
| 取消调整大小操作 | 您无法取消弹性调整大小。 | 您可以在经典调整大小操作完成之前取消该操作，方法是在 Amazon Redshift 控制台中从集群详细信息中选择取消调整大小。 | 取消调整大小操作所需的时间取决于取消时调整大小操作所处的阶段。在取消操作完成前，如果您这样做，集群将不可用。如果调整大小操作处 | | | | |

| 行为 | 弹性调整大小 | 经典调整大小 | 评论 | | | | |
|---|---|---|---|---|---|---|---|
| | | | 于最后阶段，您将无法取消。<br><br>对 RG 或 RA3 集群进行经典调整大小时，您无法取消。 | | | | |

计划调整大小

您可以为集群制定调整大小操作计划，使其纵向扩展以满足预期高使用率的要求，或者缩减其规模以节约成本。计划适用于弹性调整大小和经典调整大小。您可以在 Amazon Redshift 控制台上设置计划。有关更多信息，请参阅使用控制台管理集群下的 调整集群大小。还可以使用 AWS CLI 或 Amazon Redshift API 操作来设置调整大小计划。有关更多信息，请参阅 AWS CLI 命令参考中的 create-scheduled-action 和 Amazon Redshift API 参考中的 CreateScheduledAction。

快照、还原和调整大小

弹性调整大小是调整 Amazon Redshift 集群大小的最快方法。如果弹性调整大小操作不适合您，并且您需要对集群进行近乎恒定的写入访问，则可以使用下一部分中所述的快照和还原操作以及经典调整大小。如果采用此方法，在切换后，您必须手动将在制作快照后写入源集群的所有数据都复制到目标集群中。根据复制用时，您可能需要重复执行此操作多次，直到两个集群中的数据相同。然后，您可以切换

到目标集群。在目标集群拥有完整数据集之前，此过程可能会对现有查询产生负面影响。不过，它能最大程度地缩短您无法写入数据库的时间。

快照、还原和调整大小方法使用以下流程：

1. 为您的现有集群制作快照。现有集群就是源集群。
2. 记下制作快照的时间。这样做意味着，您稍后可确定需要重新运行提取、事务处理和加载 (ETL) 流程的时间点，从而将制作快照后写入的所有数据都加载到目标数据库中。
3. 将快照还原到新集群中。这个新集群就是目标集群。验证目标集群中包含示例数据。
4. 调整目标集群的大小。为目标集群选择新的节点类型、节点数和其他设置。
5. 查看为源集群制作快照后通过 ETL 流程加载的数据。请确保按相同顺序将相同的数据重新加载到目标集群中。如果您的数据正在不断加载，请重复执行此流程多次，直到源集群和目标集群中的数据相同为止。
6. 停止在源集群上运行的所有查询。为此，您可以重启集群，或以超级用户的身份登录并使用 PG_CANCEL_BACKEND 和 PG_TERMINATE_BACKEND 命令。重启集群是确保集群不可用的最简单方法。
7. 重命名源集群。例如，将其从 `examplecluster` 重命名为 `examplecluster-source`。
8. 重命名目标集群，使用源集群在重命名之前的名称。例如，将目标集群从之前的名称重命名为 `examplecluster`。此后，使用包含 `examplecluster` 的端点的所有应用程序都会连接到目标集群。
9. 在切换到目标集群后删除源集群，并验证所有流程均可按预期正常工作。

或者，您可以在将数据重新加载到目标集群之前重命名源集群和目标集群。如果您不要求任何关联系统和报告立即与目标集群中的相应内容保持同步，则此方法有效。在这种情况下，步骤 6 将移至前述流程的最后。

仅当您希望应用程序继续使用相同的端点连接到集群时，才需要执行重命名流程。如果您没有此要求，则可以更新连接到集群的任何应用程序，以使用目标集群的端点，而无需重命名集群。

重新使用集群名称具有诸多优势。首先，您无需更新应用程序的连接字符串，因为端点始终保持不变，即使基础集群发生改变也是如此。其次，相关项（例如 Amazon CloudWatch 警报和 Amazon Simple Notification Service (Amazon SNS) 通知）与集群名称相关联。这种关联意味着，您可以继续使用已为集群设置的相同警报和通知。在您希望灵活调整集群大小而无需重新配置相关项目（如警报和通知）的生产环境中，继续使用集群名称关系重大。

# 重命名集群

如果您希望集群使用其他名称，则可以对其进行重命名。由于连接到集群的端点包含集群名称（也称集群标识符），因此重命名集群之后，端点也会改为使用新名称。例如，如果您的集群名为 `examplecluster`，您将其重命名为 `newcluster`，则端点会改为使用 `newcluster` 标识符。连接至集群的所有应用程序都必须使用新的端点进行更新。

如果您希望更改应用程序所连接到的集群，但又不想更改这些应用程序中的端点，则可以重命名集群。在此情况下，您必须先重命名原始集群，然后更改第二个集群，以重新使用原始集群重命名之前的名称。之所以这样做，是因为集群标识符在您的账户和区域中必须是独一无二的，因此原始集群和第二个集群的名称不能相同。如果从快照中还原集群，并且不希望更改任何从属应用程序的连接属性，则可能会这样做。

> **Note**
>
> 如果您删除原始集群，则还需删除任何不需要的集群快照。

当您重命名集群时，在该过程结束之前，集群的状态会变为 `renaming`。该集群使用的旧 DNS 名称会被立即删除，但可能会在缓存中保留几分钟。重命名后的集群的新 DNS 名称在大约 10 分钟内生效。重命名后的集群在新名称生效之前不可用。系统会重新启动集群，且集群的所有现有连接都会被删掉。此过程完成后，端点会改为使用新名称。因此，您应该在开始重命名之前停止运行查询，并在重命名完成后恢复运行。

系统将会保留集群快照，而且与集群相关联的所有快照在集群重命名之后仍与该集群相关联。例如，假设您的一个集群服务于生产数据库且该集群有若干个快照。如果重命名该集群，然后在生产环境中将其替换为一个快照，则这些现有快照仍与重命名后的集群相关联。

Amazon CloudWatch 警报和 Amazon Simple Notification Service (Amazon SNS) 事件通知与集群的名称相关联。如果您重命名该集群，则必须相应地对这些内容进行更新。您可以在 CloudWatch 控制台中更新 CloudWatch 警报，也可以在 Amazon Redshift 控制台中的事件窗格上更新 Amazon SNS 事件通知。集群的加载和查询数据在重命名前后都会继续显示数据。但是，性能数据则在重命名过程完成后重置。

有关更多信息，请参阅 修改集群。

# 升级集群的发行版本

您可以升级版本状态值为有可用的新版本的集群的发布维护版本。在升级维护版本时，可以选择是立即升级还是在下一个维护时段升级。

> **Important**
>
> 如果选择立即升级，您的集群将处于脱机状态，直到升级完成。

将集群升级到新的发布版本

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。
3. 选择要升级的集群。
4. 对于操作，选择升级集群版本。这将显示升级集群版本页面。
5. 按照页面上的说明操作。
6. 选择升级集群版本。

## 暂停和恢复集群

如果您的集群只需在特定时间可用，则可以暂停该集群，然后再恢复它。当集群暂停时，按需计费将暂停。仅集群的存储会产生费用。有关定价的更多信息，请参阅 Amazon Redshift 定价页面。

如果您暂停集群，Amazon Redshift 将创建快照、开始终止查询并将集群置于暂停状态。如果您删除已暂停的集群而未请求最终快照，则无法还原该集群。在暂停或恢复操作启动后，无法取消或回滚该操作。

您可以在新的 Amazon Redshift 控制台上暂停和恢复集群，也可以使用 AWS CLI 或 Amazon Redshift API 操作暂停和恢复集群。

可以计划用于暂停和恢复集群的操作。在使用新的 Amazon Redshift 控制台创建暂停和恢复的定期计划时，将为您选择的日期范围创建两个计划操作。计划操作名称的后缀为 `-pause` 和 `-resume`。名称的总长度必须在计划操作名称的最大大小范围内。

无法暂停以下类型的集群：

- EC2-Classic 集群。
- 不活动的集群，例如，当前正在修改的集群。
- 硬件安全模块 (HSM) 集群。
- 已关闭自动快照的集群。

在决定暂停集群时，请考虑以下事项：

- 与集群的连接或针对集群的查询不可用。
- 在 Amazon Redshift 控制台上看不到已暂停集群的查询监控信息。
- 无法修改已暂停的集群。未对集群执行任何计划操作。其中包括创建快照、调整集群大小和集群维护操作。
- 未创建硬件指标。如果已设置指标缺失警报，请更新 CloudWatch 警报。
- 无法将已暂停集群的最新自动快照复制到手动快照。
- 当正在暂停集群时，无法恢复集群，直至暂停操作完成。
- 如果您暂停集群，则计费将暂停。不过，暂停操作通常在 15 分钟内完成，具体取决于集群的大小。
- 审计日志已存档，无法在恢复时进行还原。
- 暂停集群后，跟踪和日志可能无法用于对暂停之前出现的问题进行故障排除。
- 当您使用 AWS Secrets Manager 管理来管理您的管理员凭证并暂停了集群时，您集群的密钥不会被删除，并且系统将继续向您收取该密钥的费用。有关使用 AWS Secrets Manager 管理 Redshift 管理员密码的更多信息，请参阅使用 AWS Secrets Manager 管理 Amazon Redshift 管理员密码。
- 对于 RG 和 RA3 实例类型，恢复时会还原集群上的无备份表。对于 DC2 实例类型，恢复时不会还原这些表。有关无备份表的更多信息，请参阅从快照中排除表。

在恢复集群时，请考虑以下事项：

- 已恢复集群的集群版本将根据集群的维护时段更新为维护版本。
- 如果删除与已暂停集群关联的子网，则您的网络可能会不兼容。在此情况下，将从最新的快照还原您的集群。
- 如果您在集群暂停时删除弹性 IP 地址，则会请求新的弹性 IP 地址。
- 如果 Amazon Redshift 无法使用集群之前的弹性网络接口恢复集群，Amazon Redshift 会尝试分配一个新的弹性网络接口。
- 在恢复集群时，您的节点 IP 地址可能会发生更改。对于从 Secure Shell (SSH) 执行 COPY 或从 Amazon EMR 执行 COPY 等功能，您可能需要更新 VPC 设置以支持这些新的 IP 地址。
- 如果您尝试恢复未暂停的集群，恢复操作将返回错误。如果恢复操作是计划操作的一部分，请修改或删除计划操作以防止将来出现错误。
- 根据集群的大小，可能需要几分钟来恢复集群，然后才能处理查询。此外，在恢复完成后重新补充集群时，查询性能可能会在一段时间内受到影响。

# 重新引导集群

重启集群是一种集群操作，使用与重启前相同的配置来重新启动集群。可以重启集群以应用待处理的维护更新、重置配置更改、从某些问题中恢复或解决集群问题。重启集群有助于确保 Amazon Redshift 环境的卓越性能、安全性和稳定性。以下过程提供了重启 Amazon Redshift 集群的详细步骤。

重启集群后，集群状态将设置为 `rebooting`，并在重启完成后创建一个集群事件。任何待处理的集群修改都将在此次重启时应用。

重启集群

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。
3. 选择要重启的集群。
4. 对于操作，选择重启集群。此时将显示重启集群页面。
5. 选择重启集群。

# 重新定位集群

通过在 Amazon Redshift 中使用重新定位，您可以让 Amazon Redshift 将集群移动到另一个可用区 (AZ)，而不会丢失任何数据或对您的应用程序进行更改。通过重新定位，您可以在集群上出现服务中断时继续操作，而产生的影响最小。

开启集群重新定位后，Amazon Redshift 可能会在某些情况下选择重新定位集群。特别是，当当前可用区中的问题阻碍了最佳集群操作或提高服务可用性时，会发生这种情况。您还可以在给定可用区中的资源限制中断集群操作的情况下调用重新定位功能。例如，恢复集群或调整集群大小的功能。Amazon Redshift 提供重新定位功能，无需额外付费。

当 Amazon Redshift 集群重新定位到新的可用区时，新集群具有与原始集群相同的端点。您的应用程序可以重新连接到端点并继续操作，而不会进行修改或丢失数据。但是，由于给定可用区中的潜在资源限制，重新定位可能并不总是可能的。

RG 和 RA3 实例类型支持 Amazon Redshift 集群重新放置。RG 和 RA3 实例类型使用 Redshift 托管存储（RMS）作为持久存储层。集群数据的最新副本始终提供在 AWS 区域中的其他可用区中。换句话说，您可以将 Amazon Redshift 集群重新定位到另一个可用区，而不会丢失任何数据。

当您开启集群的重新定位时，Amazon Redshift 会将您的集群迁移到代理后面。这样做有助于实现对集群计算资源的位置无关访问。迁移会导致重新引导集群。当集群重新定位到另一个可用区时，在新集群

在新可用区中重新联机时，会发生中断。但是，您无需对应用程序进行任何更改，因为即使集群重新定位到新的可用区后，集群端点仍保持不变。

默认情况下，在新创建或还原的 RG 或 RA3 集群（其子网组包含多个可用区）上，集群重新放置处于启用状态。在创建预置集群时，Amazon Redshift 将 5439 指定为默认端口。您可以更改为 5431-5455 或 8191-8215 端口范围内的另一个端口。（不要更改为超出范围的端口。这会导致错误。）要更改预置集群的默认端口，请使用 Amazon Redshift 控制台、AWS CLI 或 Amazon Redshift API。要更改无服务器工作组的默认端口，请使用 AWS CLI 或 Amazon Redshift Serverless API。

如果您开启重新定位，并且您当前使用领导节点 IP 地址来访问集群或使用增强型 VPC 路由，请确保更改该访问权限。反之，请使用与集群的 Virtual Private Cloud (VPC) 端点关联的 IP 地址。要查找此集群 IP 地址，在集群详细信息页面的网络和安全部分中查找和使用 VPC 终端节点。要获取有关 VPC 终端节点的更多详细信息，请登录 Amazon VPC 控制台。

您也可以使用 AWS Command Line Interface (AWS CLI) 命令 `describe-vpc-endpoints` 获取与端点关联的弹性网络接口。您可以使用 `describe-network-interfaces` 命令获取关联的 IP 地址。有关 Amazon Redshift AWS CLI 命令的更多信息，请参阅 AWS CLI 命令参考中的可用命令

## 限制

使用 Amazon Redshift 重新定位时，请注意以下限制：

- 由于给定可用区中的潜在资源限制，可能无法在所有情况下进行集群重新定位。如果发生这种情况，Amazon Redshift 不会更改原始集群。
- 产品的 DC2 实例系列不支持重新定位。
- 您不能跨 AWS 区域执行重新定位。
- Amazon Redshift 重新定位原定设置为端口号 5439。您也可以更改为 5431-5455 或 8191-8215 范围内的另一个端口。

## 使用控制台管理重新定位

您可以使用 Amazon Redshift 控制台管理集群重新定位的设置。

在创建新集群时关闭重新定位

在创建新集群时，可以使用以下过程关闭重新定位。

为新集群关闭重新定位

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。
3. 选择创建集群以创建新集群。有关如何创建集群的更多信息，请参阅《Amazon Redshift 入门指南》中的 Amazon Redshift 预置数据仓库入门。
4. 在备份下，对于集群重新定位选择禁用。默认情况下，重新定位处于开启状态。
5. 选择创建集群。

修改现有集群的重新定位

使用以下过程更改现有集群的重新定位设置。

要修改现有集群的重新定位设置

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。这将列出您的账户在当前 AWS 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。
3. 从列表中选择您要修改的集群的名称。此时将会显示集群详细信息页面。
4. 选择维护选项卡，然后在备份详细信息部分中选择编辑。
5. 在备份下，选择已禁用。默认情况下，重新定位处于开启状态。
6. 选择修改集群。

重新定位集群

使用以下过程手动将集群重新定位到其他可用区。当您希望在辅助可用区中测试网络设置或当您在当前可用区中遇到资源限制时，这一点尤其有用。

要将集群重新定位到其他可用区

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。这将列出您的账户在当前 AWS 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。

3. 从列表中选择您要移动的集群的名称。此时将会显示集群详细信息页面。
4. 对于操作，选择重新定位。此时将显示重新定位集群页面。
5. （可选）选择一个可用区。如果您没有选择可用区，Amazon Redshift 会为您选择一个可用区。

Amazon Redshift 开始重新定位并将集群显示为重新定位。重新定位完成后，集群状态将更改为可用。

## 使用 Amazon Redshift CLI 管理重新定位

您可以使用 AWS 命令行界面 (CLI) 管理集群重新定位的设置。

使用 AWS CLI，以下示例命令会创建一个开启了重新定位且名为 **`mycluster`** 的 Amazon Redshift 集群。

```
aws redshift create-cluster --cluster-identifier mycluster --number-of-nodes 2 --
master-username enter a username --master-user-password enter a password --node-type
 ra3.4xlarge --port 5439 --no-availability-zone-relocation
```

如果您的当前集群使用不同的端口，您必须在修改它以开启重新定位之前，将其修改为使用 5431-5455 或 8191-8215 端口范围内的端口。原定设置为 5439。下面的示例命令会在集群没有使用给定范围内的端口时修改端口。

```
aws redshift modify-cluster --cluster-identifier mycluster --port 5439
```

以下示例命令在 Amazon Redshift 集群上包括 availability-zone-relocation 参数。

```
aws redshift modify-cluster --cluster-identifier mycluster --availability-zone-
relocation
```

以下示例命令在 Amazon Redshift 集群上关闭 availability-zone-relocation 参数。

```
aws redshift modify-cluster --cluster-identifier mycluster --no-availability-zone-
relocation
```

以下示例命令将在 Amazon Redshift 集群上调用重新定位。

```
aws redshift modify-cluster --cluster-identifier mycluster --availability-zone us-
east-1b
```

# 对集群设置用量限制

您最多可以添加四个用量限制来控制以下每项的用量：

- 并发扩展
- 使用额外的计算资源运行的自动优化
- Redshift Spectrum 用量
- 跨区域数据共享

## 为预置集群设置用量限制

以下是在预置集群上设置用量限制的过程：

为集群设置用量限制

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 导航到要为其设置限制的预置集群。
3. 从集群的详细信息页面中，从操作下拉菜单中选择管理用量限制。也可以为集群选择维护选项卡，然后向下滚动并选择创建用量限制。
4. 为要设置的用量限制选择添加限制。您最多可以为一个给定功能添加 4 个限制。
5. 为用量限制设置时间段，即每日、每周或每月。
6. 设置用量限制。
   - 对于使用额外计算资源限制运行的并发扩展和自动优化，用量限制是 Amazon Redshift 在给定时间段内使用该功能所花的时间量。在这种情况下，用量限制以小时和分钟为单位设置。
   - 对于 Redshift Spectrum，用量限制是从 Amazon S3 扫描的数据量。在这种情况下，用量限制以太字节（TB）为单位设置。
   - 对于跨区域数据共享，用量限制是指使用者可以查询的从生产者区域传输到使用者区域的数据量。在这种情况下，用量限制以太字节（TB）为单位设置。
7. 设置当您的集群达到限制时，Amazon Redshift 要采取的操作。这些操作包括：
   - 记录到系统表：向系统视图 SYS_QUERY_HISTORY 中添加记录。您可以在此视图中查询 usage_limit 列，来确定查询是否超出了限制。
   - 警报：使用 Amazon SNS 设置通知订阅，并在违反限制时发送通知。您可以选择现有的 Amazon SNS 主题、创建新主题或在无主题的情况下继续操作。

- 禁用功能：禁用该功能。还可以选择使用 Amazon SNS 来发送通知。用户可以继续使用集群来执行其它任务。

前两个操作是信息性的，但最后一个操作会关闭此功能的使用。

8. 选择页面底部的保存以保存限制。如果您同时设置了多个限制，保存更改将同时保存所有限制。

## 关闭和删除集群

如果您希望集群停止运行且不再产生费用，则可以将其关闭。在关闭集群时，您可以选择创建一个最终快照。如果您创建最终快照，则 Amazon Redshift 会在您将集群关闭之前为其创建一个手动快照。如果您打算使用与要删除的集群相同的数据和配置来预配置新集群，您将需要使用手动快照。通过使用手动快照，您可以稍后还原快照并恢复使用集群。

如果您不再需要集群及其数据，则可以将其关闭，而不创建最终快照。在这种情况下，会永久删除该集群及其数据。

无论在您关闭集群时是否有最终手动快照，与该集群相关联的所有自动快照都会在您关闭集群后删除。与该集群相关联的所有手动快照都会保留下来。如果您在关闭集群时没有其他正在运行的集群，或者正在运行的 Amazon Redshift 集群超出了向您提供的免费可用存储，则您需要针对保留下来的所有手动快照（包括可选的最终快照）按 Amazon Simple Storage Service 存储费率支付相应的费用。有关快照存储费用的更多信息，请参阅 Amazon Redshift 定价页面。

删除集群也会删除所有关联的 AWS Secrets Manager 密钥。

删除集群

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。
3. 选择要删除的集群。
4. 对于操作，选择删除。此时将显示删除集群页面。
5. 选择删除集群。

> **Note**
>
> 当您删除集群并选择创建最终快照时，如果正在对该集群执行还原操作，则 Amazon Redshift 将停止删除请求。如果发生这种情况，您可以删除没有最终快照的集群，也可以在还原完成后将其与最终快照一起删除。

# Amazon Redshift 快照和备份

快照是集群的时间点备份。存在两种类型的快照：自动和手动。Amazon Redshift 通过使用加密的安全套接字层 (SSL) 连接，在 Amazon S3 内部存储这些快照。

Amazon Redshift 会自动拍摄递增快照，来跟踪自上次自动快照拍摄以来集群发生的变化。自动快照保留从快照还原集群所需的全部数据。您可以创建快照计划来控制何时拍摄自动快照，也可以随时拍摄手动快照。

当您从快照还原时，Amazon Redshift 将创建一个新集群并使该新集群在所有数据都已加载前可用，从而使您可以立即开始查询新集群。该集群将按需流式传输来自快照的数据以响应活动查询，然后在后台加载剩余的数据。

在启动集群时，可以设置自动快照和手动快照的保留期。您可以通过修改集群来更改自动快照和手动快照的默认保留期。您可以在创建快照时或通过修改快照来更改手动快照的保留期。

您可以通过下列方式监控快照进度：在 AWS 管理控制台中查看快照详细信息、调用 CLI 中的 describe-cluster-snapshots 或调用 DescribeClusterSnapshots API 操作。对于正在进行的快照，通过上述方式可以看到增量快照的大小、传输速率、已用时间以及估计的剩余时间等信息。

为了确保您的备份始终对集群可用，Amazon Redshift 将快照存储在由 Amazon Redshift 管理的内部托管 Amazon S3 桶中。要管理存储费用，请估计需要将自动快照保留多少天并相应地配置其保留期。删除所有不再需要的手动快照。有关备份存储成本的更多信息，请参阅 Amazon Redshift 定价。

您还可以使用 AWS Backup 创建和还原快照，这是一项完全托管式服务，有助于您在云端以及在本地集中管理和自动执行各种 AWS 服务中的数据保护。有关更多信息，请参阅 AWS Backup 与 Amazon Redshift 集成。有关 AWS Backup 的信息，请参阅《AWS Backup Developer Guide》中的 What is AWS Backup?

## 使用 Amazon Redshift Serverless 中的快照和备份

Amazon Redshift Serverless 与预置集群一样，使您能够将备份作为命名空间中的对象和数据在某个时间点的表示形式。Amazon Redshift Serverless 中的备份有两种类型：手动创建的快照，以及 Amazon

Redshift Serverless 自动创建的恢复点。可以在快照和恢复点中找到有关使用 Amazon Redshift Serverless 快照的更多信息。

您也可以将快照从预置集群还原到无服务器命名空间。有关更多信息，请参阅从快照还原无服务器命名空间。

## 自动快照

当某个集群的自动快照处于启用状态时，Amazon Redshift 会定期拍摄该集群的快照。默认情况下，Amazon Redshift 大约每 8 小时或在每节点数据更改达到 5 GB 时拍摄一次快照，以先到者为准。如果您的数据大于 5 GB * 节点数，则自动快照创建间隔的最短时间为 15 分钟。或者，您也可以创建快照计划来控制何时拍摄自动快照。如果您使用自定义计划，则自动快照间隔的最短时间为 1 小时。当您创建集群时，自动快照默认处于启用状态。

> **Note**
>
> Amazon Redshift 会为暂停的数据共享创建者集群自动拍摄快照。由于使用者可向暂停的创建者集群写入数据，这类快照能够捕获数据共享写入，并提供恢复点。以前，Amazon Redshift 不会为暂停状态下的此类集群自动拍摄快照。这些快照遵循集群现有的自动快照计划和保留期，您无需执行任何操作。对于并非数据共享创建者的暂停集群，Amazon Redshift 在集群暂停期间不会自动拍摄快照。

系统会在保留期结束时删除自动快照。默认保留期为一天；不过，您可以使用 Amazon Redshift 控制台对其进行修改，也可以使用 Amazon Redshift API 或 CLI 以编程方式对其进行修改。

要禁用自动快照，只需将保留期设置为零即可。如果您禁用自动快照，则 Amazon Redshift 会停止拍摄快照并删除相应集群的任何现有自动快照。您不能为 RG 或 RA3 节点类型禁用自动快照。您可以将 RG 或 RA3 节点类型的自动保留期设置为 1-35 天。

只有 Amazon Redshift 才能删除自动快照；您不能手动删除它们。Amazon Redshift 会在下列情况下删除自动快照：快照的保留期结束时、您禁用了集群的自动快照或您删除了集群。Amazon Redshift 会保留最新的自动快照，直到您禁用自动快照或删除集群为止。

如果您想要将自动快照保留更长时间，则可以创建一份副本作为手动快照。自动快照会保留至保留期结束；而相应的手动快照在您将其手动删除前或保留期结束前将会一直保留。

## 自动快照计划

要精确控制拍摄快照的时间，您可以创建快照计划并将其附加到一个或多个集群。修改快照计划时，将修改所有关联集群的计划。如果集群未附加快照计划，其将使用默认的自动快照计划。

快照计划 是一组计划规则。您可以基于指定的时间间隔（例如每 8 小时或每 12 小时）定义简单的计划规则。也可以添加规则来在一周中的某些天、特定时间或特定时段拍摄快照。此外，您还可以使用类 Unix 系统的 cron 表达式定义规则。

### 快照计划格式

您可以在 Amazon Redshift 控制台中创建快照计划。然后，您可以将计划附加到集群来触发创建系统快照。可以将计划附加到多个集群，也可以在计划中创建多个 cron 定义来触发快照。

您可以使用 cron 语法为快照定义计划。这些计划的定义使用经过修改的类 Unix 系统的 cron 语法。您可以使用协调世界时 (UTC) 格式指定时间。您可以创建最大频率为 1 小时、最小精度为 1 分钟的计划。

Amazon Redshift 修改后的 cron 表达式有 3 个必填字段，之间以空格分隔。

语法

```
cron(Minutes Hours Day-of-month Month Day-of-week Year)
```

| 字段 | 值 | 通配符 |
| --- | --- | --- |
| 分钟 | 0-59 | , - * / |
| 小时 | 0-23 | , - * / |
| Day-of-month | 1-31 | , - * ? / L W |
| Month | 1-12 或 JAN-DEC | , - * / |
| 星期几 | 1–7 或 SUN-SAT | , - * ? L # |
| Year | 1970-2199 | , - * / |

通配符

- , ( 逗号 ) 通配符包含其他值。在 Day-of-week 字段中，MON,WED,FRI 将包含星期一、星期三和星期五。总值限制为每字段 24 个。
- - ( 破折号 ) 通配符用于指定范围。在 Hour 字段中，1–15 将包含指定日期的 1 - 15 小时。
- * ( 星号 ) 通配符包含该字段中的所有值。在 Hours 字段中，* 将包含每个小时。
- / ( 正斜杠 ) 通配符用于指定增量。在 Hours 字段中，您可以输入 **1/10** 来指定从当天的第 1 个小时开始每隔 10 小时 ( 例如，01:00、11:00 和 21:00 ) 。
- ? ( 问号 ) 通配符用于指定一个或另一个。在 Day-of-month 字段中，您可以输入 7，如果您不介意 7 日是星期几，则可以在“星期几”字段中输入 ?。
- 或 字段中的 Day-of-monthLDay-of-week 通配符用于指定月或周的最后一天。
- Day-of-month 字段中的 W 通配符用于指定工作日。在 Day-of-month 字段中，3W 用于指定最靠近当月的第三周的日。
- “Day-of-week”字段中的 # 通配符用于指定一个月内所指定星期几的特定实例。例如，3#2 指该月的第二个星期二：3 指的是星期二，因为它是每周的第三天，2 是指该月内该类型的第二天。

> **Note**
>
> 如果使用 “#” 字符，则只能在星期字段中定义一个表达式。例如，“3#1,6#3”是无效的，因为它被解释为两个表达式。

限制

- 您无法在同一 cron 表达式中为 Day-of-month 和 Day-of-week 字段同时指定值。如果您在其中一个字段中指定了值，则必须在另一个字段中使用 ? ( 问号 ) 。
- 快照计划不支持以下频率：
  - 计划快照的频率超过每小时 1 次。
  - 计划快照的频率低于每天 1 次 ( 24 小时 ) 。

  如果您有重叠的计划导致在 1 小时时段内计划了多次快照，将产生验证错误。

在创建计划时，您可以使用以下示例 cron 字符串。

| 分钟 | 小时 | 星期几 | 意义 | | | |
|---|---|---|---|---|---|---|
| 0 | 14-20/1 | TUE | 星期二下午 2 点到晚上 8 点之间，每小时拍摄一次。 | | | |
| 0 | 21 | MON-FRI | 每天晚上 9 点，星期一至星期五。 | | | |
| 30 | 0/6 | SAT-SUN | 星期六和星期日从当天午夜 30 分 (00:30) 开始，每 6 小时拍摄一次。这导致在每天的 [00:30、06:30、12:30 和 18:30] 拍摄快照。 | | | |
| 30 | 12/4 | * | 每天从 12:30 开始，每 4 小时拍摄一次。这将解析为 [12:30、16:30、20:30]。 | | | |

例如，要运行每天从 15:15 开始、每 2 小时拍摄一次的计划（解析为 [15:15、17:15、19:15、21:15、23:15]），请指定：

```
cron(15 15/2 *)
```

您可以在计划中创建多个 cron 计划定义。例如，以下 AWS CLI 命令在一个计划中包含两个 cron 计划。

```
create-snapshot-schedule --schedule-identifier "my-test" --schedule-definition "cron(0
 17 SAT,SUN)" "cron(0 9,17 MON-FRI)"
```

## 手动快照

您可以随时制作手动快照。默认情况下，即使您删除相应集群，手动快照也会无限期保留。您可以在创建手动快照时指定保留期，也可以通过修改快照来更改保留期。有关更改保留期的更多信息，请参阅修改手动快照保留期。

如果删除某个快照，则无法启动任何引用该快照的新操作。不过，如果某个还原操作正在进行中，则该还原操作会运行完成。

Amazon Redshift 设有配额，用以限制您能够创建的手动快照的总数；该配额因不同的 AWS 账户及不同的 AWS 区域而异。默认的配额列出在 Amazon Redshift 资源中的配额和限制 中。

## 快照存储

由于快照会产生存储费用，因此，当您不再需要快照时，务必将其手动删除。在相应的快照保留期结束时，Amazon Redshift 将删除自动快照和手动快照。您也可以使用 AWS 管理控制台 或 batch-delete-cluster-snapshots CLI 命令来删除手动快照。

您可以通过修改手动快照设置来更改手动快照的保留期。

您可以使用 Amazon Redshift 控制台或 describe-storage CLI 命令获取有关快照占用的存储量的信息。

## 从快照中排除表

默认情况下，所有用户定义的永久表都包含在快照中。如果表（如暂存表）不需要备份，您可以显著降低创建快照并从快照还原所需的时间。您还可以使用无备份表减小在 Amazon S3 上占用的存储空间。要创建无备份表，请在创建该表时包含 BACKUP NO 参数。有关语法的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 CREATE TABLE 和 CREATE TABLE AS。

> **Note**
>
> RG 或 RA3 预置集群和 Amazon Redshift Serverless 工作组不支持无备份表。在 RG 或 RA3 集群和无服务器工作组中标记为无备份的表将被视为永久表，在拍摄快照时将始终对其进行备份，并在从快照还原时还原该表。要避免无备份表产生快照成本，请在拍摄快照之前将其截断。

## 创建手动快照

您可以从如下所示的快照列表中创建集群的手动快照。您也可以在集群配置窗格中制作集群快照。有关更多信息，请参阅 Amazon Redshift 快照和备份。

> **Note**
>
> RG 或 RA3 预置集群和 Amazon Redshift Serverless 工作组不支持无备份表。在 RG 或 RA3 集群和无服务器工作组中标记为无备份的表将被视为永久表，在拍摄快照时将始终对其进行备份，并在从快照还原时还原该表。要避免无备份表产生快照成本，请在拍摄快照之前将其截断。

创建手动快照

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群、快照，然后选择创建快照。此时将显示用于创建手动快照的快照页面。
3. 输入快照定义的属性，然后选择创建快照。快照可能需要一段时间才可用。

## 创建快照计划

Amazon Redshift 会定期自动对您的数据拍摄递增快照，并将其保存到 Amazon S3 中。此外，您可以随时对数据制作手动快照。

Amazon Redshift 控制台中的所有快照任务都从快照列表开始。您可以使用时间范围、快照类型以及与快照关联的集群来筛选该列表。此外，您可以按日期、大小和快照类型对列表进行排序。根据您选择的快照类型，您处理快照的选项可能有所不同。

要精确控制拍摄快照的时间，您可以创建快照计划并将其附加到一个或多个集群。您可以在创建集群时或通过修改集群来附加计划。有关更多信息，请参阅 自动快照计划。

要创建快照计划

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群、快照，然后选择快照计划选项卡。此时将显示快照计划。
3. 选择添加计划以显示添加计划的页面。
4. 输入计划定义的属性，然后选择添加计划。
5. 在显示的页面上，您可以将集群附加到新的快照计划，然后选择确定。

## 共享快照

您可以授权其他 AWS 客户账户访问现有手动快照，以与其共享该快照。对于每个快照，最多可以授权 20 个客户账户；对于每个 AWS Key Management Service (AWS KMS) 密钥，则最多可以授权 100 个。例如，如果您有 10 个快照，它们使用了一个 KMS 密钥加密，那么您可以授权 10 个 AWS 账户来还原每个快照，或者是其他组合：总共 100 账户以及每个快照不超过 20 个账户。然后，以用户身份登录其中一个已授权账户的人可以对快照加以说明，也可以还原快照以在其账户下创建一个新的 Amazon Redshift 集群。例如，如果您针对生产和测试使用不同的 AWS 客户账户，则用户可以使用生产账户登录并与使用测试账户的用户共享快照。然后，以测试账户用户身份登录的人可以还原快照以创建一个新的集群，该集群由测试账户所有，用于测试或诊断工作。

手动快照由在其下创建该快照的 AWS 客户账户永久所有。只有拥有相应快照的账户中的用户可以授权其他账户访问该快照或撤消授权。已授权账户中的用户只能对与其共享的任何快照加以说明或还原相应快照；他们无法复制或删除与其共享的快照。授权在快照所有者将其撤消之前保持有效。如果所有者撤消授权，则之前已获授权的用户将无法再看到相应快照，也无法再启动任何引用该快照的新操作。如果在所有者撤消授权时相应账户正在还原快照，则该还原操作会运行完成。您无法删除具有活跃授权的快照；必须先撤消所有授权。

AWS 客户账户始终能够访问相应账户所有的快照。如果尝试授予或撤消对所有者账户的访问权限，则会收到一条错误消息。您无法还原由非活动 AWS 客户账户所有的快照，也无法对其加以说明。

当您获得访问 AWS 客户账户的授权之后，该账户中便没有任何用户可以针对相应快照执行任何操作，除非他们代入的角色具有允许他们这样做的策略。

- 快照拥有者账户中的用户只能在以下情况下授予和撤消快照访问权限：此类用户代入的角色具有相应的 IAM 策略，允许他们通过包含该快照的资源规范执行此类操作。例如，以下策略允许 AWS 账户（012345678912）中的用户或角色授权其他账户访问名为 my-snapshot20130829 的快照：

  JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
          "redshift:AuthorizeSnapshotAccess",
          "redshift:RevokeSnapshotAccess"
          ],
      "Resource":[
```

```
          "arn:aws:redshift:us-east-1:012345678912:snapshot:*/my-
snapshot20130829"
         ]
    }
  ]
}
```

- 与其共享快照的 AWS 账户中的用户无法针对该快照执行操作，除非他们具有相应的权限允许他们执行这些操作。为此，您可以将策略分配给角色并代入该角色。
  - 要列出快照或对其加以说明，相应用户必须具有相应的 IAM 策略，允许他们执行 DescribeClusterSnapshots 操作。下方代码显示了一个示例：

    JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
          "redshift:DescribeClusterSnapshots"
          ],
      "Resource":[
           "*"
          ]
    }
  ]
}
```

  - 要还原快照，用户必须代入角色，该角色具有相应的 IAM 策略允许他们执行 RestoreFromClusterSnapshot 操作，且该策略要有一个资源元素，同时涵盖他们尝试创建的集群以及相应快照。例如，如果账户 012345678912 中的用户与账户 219876543210 共享了快照 my-snapshot20130829，则为了通过还原快照来创建集群，账户 219876543210 中的用户必须代入具有如下策略的角色：

    JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
```

```
      "Action":[
          "redshift:RestoreFromClusterSnapshot"
          ],
      "Resource":[
           "arn:aws:redshift:us-east-1:012345678912:snapshot:*/my-
snapshot20130829",
           "arn:aws:redshift:us-east-1:219876543210:cluster:from-another-
account"
          ]
    }
  ]
}
```

- 在撤销 AWS 账户对快照的访问权限后，该账户中的所有用户均无法访问该快照。即使这些账户具有允许对以前共享的快照资源执行操作的 IAM 策略，也是如此。

使用控制台共享集群快照

在控制台上，您可以授权其他用户访问您拥有的手动快照，随后在不再需要时撤消该访问权限。

要与另一个账户共享快照

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群、快照，然后选择要共享的手动快照。
3. 对于操作，选择手动快照设置以显示手动快照的属性。
4. 在管理访问权限部分中输入要与之共享的一个或多个账户，然后选择保存。

共享加密快照的安全注意事项

当您提供对加密快照的访问权限时，Redshift 要求用于创建快照的 AWS KMS 客户托管式密钥与执行恢复的一个或多个账户共享。如果密钥未共享，尝试恢复快照将导致拒绝访问错误。接收账户不需要任何额外权限即可还原共享快照。当您授予快照访问权限并共享密钥时，授予访问权限的身份必须对用于加密快照的密钥具有 `kms:DescribeKey` 权限。有关此权限的更多详细信息，请参阅 AWS KMS 权限。有关更多信息，请参阅 Amazon Redshift API 参考文档中的 DescribeKey。

客户托管的密钥策略可以通过编程方式或在 AWS Key Management Service 控制台中更新。

> **Note**
>
> 如果您使用默认 KMS 密钥，则无需在 AWS KMS 中采取任何操作或进行任何更改即可共享快照。

允许访问加密快照的 AWS KMS 密钥

共享加密快照的 AWS KMS 客户托管式密钥，请执行以下步骤更新密钥策略：

1. 使用 KMS 密钥策略中作为 `Principal` 共享的 AWS 账户的 Amazon 资源名称（ARN）更新 KMS 密钥策略。
2. 允许 `kms:Decrypt` 操作。

在下面的密钥策略示例中，用户 `111122223333` 是 KMS 密钥的所有者，而用户 `444455556666` 是与之共享密钥的账户。通过包含用户 `444455556666` 的根 AWS 账户身份的 ARN 作为策略的 `Principal`，以及通过允许 `kms:Decrypt` 操作，此密钥策略为 AWS 账户提供了访问 KMS 密钥的权限。

JSON

```
{
    "Id": "key-policy-1",
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/KeyUser",
                    "arn:aws:iam::444455556666:root"
                ]
            },
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

授予客户托管式 KMS 密钥访问权限后，恢复加密快照的账户必须创建 AWS Identity and Access Management (IAM) 角色或用户（如果用户尚未拥有该角色）。此外，该 AWS 账户还必须将 IAM 策略附加到该 IAM 角色或用户，以允许他们使用您的 KMS 密钥恢复加密的数据库集群快照。

要详细了解如何提供对 AWS KMS 密钥的访问权限，请参阅《AWS Key Management Service 开发人员指南》中的允许其他账户中的用户使用 KMS 密钥。

有关密钥策略的概览，请参阅 Amazon Redshift 如何使用 AWS KMS。

## 复制自动快照

自动快照会在其保留期到期、您禁用自动快照或者您删除集群时自动删除。如果要保留自动快照，可将其复制到手动快照。

复制自动快照

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群、快照，然后选择要复制的快照。
3. 对于操作，选择复制自动快照以复制快照。
4. 更新新快照的属性，然后选择复制。

## 将快照复制到其它 AWS 区域

您可以配置 Amazon Redshift，以将集群的快照（自动或手动）自动复制到另一个 AWS 区域。在集群的主要 AWS 区域创建快照时，它会复制到辅助 AWS 区域。这两个 AWS 区域分别称为源 AWS 区域和目标 AWS 区域。如果您在另一个 AWS 区域中存储快照副本，则在有任何事情影响主要 AWS 区域时，您可以使用最新数据还原集群。您可以配置集群，一次仅将快照复制到一个目标 AWS 区域。有关 Amazon Redshift 区域的列表，请参阅《Amazon Web Services 一般参考》中的区域和端点。

启用 Amazon Redshift 以自动将快照复制到另一个 AWS 区域时，您需要指定要将快照复制到哪个目标 AWS 区域。对于自动快照，您也可以指定此类快照在目标 AWS 区域中的保留期。自动快照复制到目标 AWS 区域且达到保留期之后，便会从目标 AWS 区域删除。这样做可以降低快照的使用率。若要让自动快照在目标 AWS 区域中保留更短时间或更长时间，请更改此保留期。

您为复制到目标 AWS 区域的自动快照设置的保留期与源 AWS 区域中的自动快照的保留期无关。复制的快照的默认保留期为七天。该七天保留期仅适用于自动快照。在源和目标 AWS 区域中，将在快照保留期结束时或手动删除时删除手动快照。

您可以随时禁用集群的自动快照复制功能。当您禁用此功能时，快照将不再从源 AWS 区域复制到目标 AWS 区域。复制到目标 AWS 区域的所有自动快照都将在达到保留期时删除，除非您为其创建手动快照副本。此类手动快照以及从目标 AWS 区域复制的所有手动快照在您将其手动删除之前将会一直保留在目标 AWS 区域中。

若要更改用于接收复制快照的目标 AWS 区域，请先禁用自动复制功能。然后重新启用它，并指定新的目标 AWS 区域。

某个快照复制到目标 AWS 区域之后，便处于活动和可用状态，可用于还原相关内容。

要将 AWS KMS 加密的集群的快照复制到另一个 AWS 区域，则必须为 Amazon Redshift 创建授权，以在目标 AWS 区域中使用客户主密钥。然后在启用源 AWS 区域中的快照复制功能时选择该权限授予。有关配置快照复制授权的更多信息，请参阅将 AWS KMS 加密的快照复制到另一个 AWS 区域。

## 从快照还原集群

快照包含来自您的集群上运行的任何数据库的数据。它还包含有关集群的信息，包括节点数、节点类型和管理员用户名。如果您从快照恢复集群，Amazon Redshift 会使用集群信息来创建新集群。然后它会从快照数据中恢复所有数据库。

> **Note**
>
> RG 或 RA3 预置集群和 Amazon Redshift Serverless 工作组不支持无备份表。在 RG 或 RA3 集群和无服务器工作组中标记为无备份的表将被视为永久表，在拍摄快照时将始终对其进行备份，并在从快照还原时还原该表。

对于从源快照创建的新集群，您可以选择配置，例如节点类型和节点数。如果您没有在请求中指定其他可用区，则该集群存储在相同的 AWS 区域和可用区。当您从快照还原集群时，可为新集群选择兼容的维护跟踪。

> **Note**
>
> 当您将快照恢复到具有不同配置的集群时，该快照必须在集群版本为 1.0.10013 或更高版本的集群上获取。

恢复正在进行时，事件通常按以下顺序发出：

1. RESTORE_STARTED – 当恢复过程开始时发送 REDSHIFT-EVENT-2008。
2. RESTORE_SUCCEEDED – 新集群已创建时发送 REDSHIFT-EVENT-3003。

   集群可用于查询。
3. DATA_TRANSFER_COMPLETED – 数据传输完成时发送 REDSHIFT-EVENT-3537。

> **Note**
>
> RG 和 RA3 集群仅发出 RESTORE_STARTED 和 RESTORE_SUCCEEDED 事件。由于 RG 和 RA3 节点类型将数据存储在 Amazon Redshift 托管存储中，因此，在 RESTORE 成功后不会进行显式数据传输。使用 RG 和 RA3 节点，作为正常查询处理的一部分，数据在 RG 和 RA3 节点与 Amazon Redshift 托管存储之间持续传输。RG 和 RA3 节点在本地缓存热数据，并自动将查询频率较低的数据块保存在 Amazon Redshift 托管存储中。

您可以通过下列方式监控还原进度：调用 DescribeClusters API 操作或在 AWS 管理控制台 中查看集群详细信息。对于正在进行的还原，通过上述方式可以看到快照数据的大小、传输速率、已用时间以及估计的剩余时间等信息。有关这些指标的说明，请参阅 RestoreStatus。

您无法使用快照将活动的集群还原为之前的状态。

> **Note**
>
> 当您将快照还原为新集群时，如果您没有指定其他值，则使用默认的安全组和参数组。

出于以下原因，您可能想要将快照恢复到具有不同配置的集群：

- 集群由较小的节点类型组成，且您想要将它合并到具有较少节点的较大节点类型。
- 您监控工作负载并确定需要迁移到具有更多 CPU 和存储的节点类型。
- 您想要衡量具有不同节点类型的测试工作负载的性能。

还原具有以下限制：

- 新的节点配置必须为现有数据提供足够的存储空间。甚至在添加节点时，由于数据的重新分配方式，新配置可能没有足够的存储空间。

- 还原操作将检查快照是否在与新集群的集群版本兼容的集群版本上创建。如果新集群的版本级别太早，则恢复操作将失败，并在错误消息中报告更多信息。
- 可还原到的可能配置（节点数和节点类型）取决于原始集群中的节点数量和新集群的目标节点类型。要确定可能的配置，您可以使用 Amazon Redshift 控制台或 `describe-node-configuration-options` AWS CLI 命令结合 `action-type restore-cluster`。有关使用 Amazon Redshift 控制台进行还原的更多信息，请参阅从快照还原集群。

以下步骤使用 AWS CLI 获取具有许多节点的集群，并将它合并到具有较少节点数的更大节点类型。在此示例中，我们从一个具有 24 个 节点的源集群开始。在本例中，假设我们已经为此集群创建了一个快照，并且我们想要将它恢复到更大的节点类型。

1. 运行以下命令，获取 24 节点 集群的详细信息。

```
aws redshift describe-clusters --region eu-west-1 --cluster-identifier
 mycluster-123456789012
```

2. 运行以下命令，获取快照的详细信息。

```
aws redshift describe-cluster-snapshots --region eu-west-1 --snapshot-identifier
 mycluster-snapshot
```

3. 运行以下命令，描述此快照可用的选项。

```
aws redshift describe-node-configuration-options --snapshot-identifier mycluster-
snapshot --region eu-west-1 --action-type restore-cluster
```

此命令会返回一个选项列表，包括每个选项的建议节点类型、节点数和磁盘利用率。在此示例中，上述命令列出以下可能的节点配置。我们选择还原到四节点集群中。

```
{
    "NodeConfigurationOptionList": [
        {
            "NodeType": "ra3.16xlarge",
            "NumberOfNodes": 2,
            "EstimatedDiskUtilizationPercent": 0.1
        },
        {
            "NodeType": "ra3.16xlarge",
            "NumberOfNodes": 4,
            "EstimatedDiskUtilizationPercent": 0.05
```

```
        },
        {
            "NodeType": "ra3.4xlarge",
            "NumberOfNodes": 8,
            "EstimatedDiskUtilizationPercent": 0.03
        },
        {
            "NodeType": "ra3.4xlarge",
            "NumberOfNodes": 10,
            "EstimatedDiskUtilizationPercent": 0.02
        },
        {
            "NodeType": "ra3.4xlarge",
            "NumberOfNodes": 12,
            "EstimatedDiskUtilizationPercent": 0.02
        },
        {
            "NodeType": "ra3.xlplus",
            "NumberOfNodes": 8,
            "EstimatedDiskUtilizationPercent": 0.11
        },
        {
            "NodeType": "ra3.xlplus",
            "NumberOfNodes": 16,
            "EstimatedDiskUtilizationPercent": 0.06
        }
    ]
}
```

4. 运行以下命令，将快照恢复到我们选择的集群配置。还原此集群之后，我们拥有与源集群相同的内容，但数据已合并到四个 ra3.16xlarge 节点中。

```
aws redshift restore-from-cluster-snapshot --region eu-west-1 --snapshot-identifier
 mycluster-snapshot --cluster-identifier mycluster-123456789012-x --node-type
 ra3.16xlarge --number-of-nodes 4
```

如果您有预留节点，例如 RA3 预留节点，则可以升级到 RG 预留节点。您可以在从快照还原或执行弹性调整大小时执行此操作。您可以使用控制台引导您完成此过程。有关升级到 RG 节点的更多信息，请参阅升级到 RG 或 RA3 节点类型。

在控制台上从快照还原集群

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群、快照，然后选择要还原的快照。
3. 选择从快照还原以查看使用快照信息创建的新集群的集群配置和集群详细信息值。
4. 更新新集群的属性，然后选择从快照中还原集群。

还原集群快照后，还原后的数据仓库将使用拍摄快照时使用的相同自定义 AWS KMS 密钥进行加密。如果快照没有自定义 KMS 密钥，则 Amazon Redshift 的备份加密逻辑将取决于以下因素：

- 您要将快照还原到的 Amazon Redshift 数据仓库的类型。
- 拍摄快照时集群的加密类型。

要了解从集群快照还原数据仓库后如何对其进行加密，请参阅下表：

| 目标类型 | 快照加密类型 | 目标加密类型 |
| --- | --- | --- |
| 预置集群 | 使用 AWS 托管式密钥进行加密 | 使用 AWS 托管式密钥进行加密 |
| 预置集群 | 使用 AWS 拥有的密钥进行加密 | 使用 AWS 拥有的密钥进行加密 |
| 无服务器命名空间 | 使用 AWS 托管式密钥进行加密 | 使用 AWS 拥有的密钥进行加密 |
| 无服务器命名空间 | 使用 AWS 拥有的密钥进行加密 | 使用 AWS 拥有的密钥进行加密 |

如果在创建快照时，AWS Secrets Manager 已在管理集群的管理员密码，则您必须继续使用 AWS Secrets Manager 来管理管理员密码。还原集群后，您可以在集群详细信息页面中更新集群的管理员凭证，从而选择停止使用密钥。

如果您有预留节点，则可以升级到 RG 预留节点。您可以在从快照还原或执行弹性调整大小时执行此操作。您可以使用控制台引导您完成此过程。有关升级到 RG 节点的更多信息，请参阅升级到 RG 或 RA3 节点类型。

# 从快照中还原表

您可以从快照还原单个表而不是还原整个集群。在从快照还原单个表时，您需要指定源快照、数据库、架构和表名，以及目标数据库、架构和已还原表的新表名。

> **Note**
>
> RG 或 RA3 预置集群和 Amazon Redshift Serverless 工作组不支持无备份表。在 RG 或 RA3 集群和无服务器工作组中标记为无备份的表将被视为永久表，在拍摄快照时将始终对其进行备份，并在从快照还原时还原该表。但是，不支持选择性还原无备份表。

新表名不能是现有表的名称。要将现有表替换为从快照还原的表，请先重命名或删除现有表，然后再从快照还原表。

使用源表的列定义、表属性和列属性（外键除外）创建目标表。为了防止因依赖项而导致发生冲突，目标表不从源表继承外键。不向目标表应用任何依赖项（例如，源表上的视图或授予的权限）。

如果源表的拥有者存在，那么该数据库用户是已还原的表的拥有者，前提是该用户拥有足够的权限，可成为指定的数据库和模式中关系的拥有者。否则，已还原的表由在启动集群时创建的管理员用户所有。

已还原的表将返回在执行备份时其所处的状态。这包括由 Amazon Redshift 对可序列化隔离的符合性定义的事务可见性规则，这意味着数据将立即对在备份后启动的进行中事务可见。

从快照还原表受以下限制：

- 您只能将表还原到当前正在运行的活动集群，并且只能从针对该集群制作的快照还原表。
- 一次只能还原一个表。
- 无法从在调整集群大小之前拍摄的集群快照还原表。一个例外是，如果节点类型没有更改，则可以在弹性大小调整后还原表。
- 不向目标表应用任何依赖项（例如，源表上的视图或授予的权限）。
- 如果为正在还原的表启用行级安全性，Amazon Redshift 将还原已启用行级安全性的表。

从快照还原表

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。

2. 在导航菜单上，选择集群，然后选择还原表要使用的集群。
3. 对于操作，请选择还原表以便显示还原表页面。
4. 输入要使用的快照、源表和目标表的相关信息，然后选择还原表。

Example示例：使用 AWS CLI 从快照中还原表

以下示例使用 `restore-table-from-cluster-snapshot` AWS CLI 命令从 `my-source-table` 上的 `sample-database` 架构中还原 `my-snapshot-id` 表。您可以使用 AWS CLI 命令 `describe-table-restore-status` 查看还原操作的状态。此示例将快照还原到具有新表名 `mycluster-example` 的 `my-new-table` 集群。

```
aws redshift restore-table-from-cluster-snapshot --cluster-identifier mycluster-
example
                                                 --new-table-name my-new-table
                                                 --snapshot-identifier my-snapshot-id
                                                 --source-database-name sample-
database
                                                 --source-table-name my-source-table
```

## 从快照还原无服务器命名空间

从快照还原无服务器命名空间会将命名空间的所有数据库替换为快照中的数据库。有关无服务器快照的更多信息，请参阅快照和恢复点。在您将预置的集群快照还原到 Amazon Redshift Serverless 命名空间时，Amazon Redshift 会自动将带有交错键的表转换为复合键。有关排序键的更多信息，请参阅使用排序键。

将快照从您的预置集群还原到无服务器命名空间：

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群、快照，然后选择要使用的快照。
3. 选择从快照还原、还原到无服务器命名空间。
4. 选择您要还原到的命名空间。
5. 确认想要从快照还原。选择还原。此操作将使用预置集群中的数据来替换无服务器命名空间中的所有数据库。

## 为未加密的集群配置跨区域快照复制

您可以配置 Amazon Redshift，以将集群的快照复制到其他 AWS 区域。要配置跨区域快照复制，您需要为每个集群启用该复制功能，并配置复制快照的位置以及在目标 AWS 区域中保留复制的自动或手动快照的时间。如果为集群启用了跨区域复制，所有新的手动和自动快照将复制到指定的 AWS 区域。复制的快照名称带有前缀：**copy:**。

配置跨区域快照

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群，然后选择要移动其快照的集群。
3. 对于操作，选择配置跨区域快照。

   此时将显示“配置跨区域”对话框。
4. 对于复制快照，选择是。
5. 在目标 AWS 区域中，选择要将快照复制到的 AWS 区域。
6. 在自动快照保留期(天) 中，选择您希望自动快照被删除之前在目标 AWS 区域中保留的天数。
7. 在手动快照保留期中，选择您希望手动快照被删除之前在目标 AWS 区域中保留的天数。如果选择自定义值，则保留期必须在 1 到 3653 天之间。
8. 选择保存。

## 为 AWS KMS 加密的集群配置跨区域快照副本

启动 Amazon Redshift 集群时，您可以为您在目标 AWS 区域中的账户中的根密钥配置快照复制授予。如果您未配置授予，则将使用默认 AWS 拥有的密钥对目标区域中的快照进行加密。通过执行此操作，您可以使 Amazon Redshift 在目标 AWS 区域执行加密操作。

以下过程说明了为 AWS KMS 加密的集群启用跨区域快照复制的过程。要详细了解 Amazon Redshift 中的加密功能以及快照复制授予，请参阅将 AWS KMS 加密的快照复制到另一个 AWS 区域。

要为 AWS KMS 加密的集群配置跨区域快照

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群，然后选择要移动其快照的集群。
3. 对于操作，选择配置跨区域快照。

此时将显示“配置跨区域”对话框。

4. 对于复制快照，选择是。
5. 在目标 AWS 区域中，选择要将快照复制到的 AWS 区域。
6. 在自动快照保留期(天) 中，选择您希望自动快照被删除之前在目标 AWS 区域中保留的天数。
7. 在手动快照保留期中，选择您希望手动快照被删除之前在目标 AWS 区域中保留的天数。如果选择自定义值，则保留期必须在 1 到 3653 天之间。
8. 选择保存。

## 修改手动快照保留期

您可以通过修改快照设置来更改手动快照的保留期。

更改手动快照保留期

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群、快照，然后选择要更改的手动快照。
3. 对于操作，选择手动快照设置以显示手动快照的属性。
4. 输入快照定义的修订属性，然后选择保存。

## 修改跨区域快照复制的保留期

在配置跨区域快照复制后，您可能希望更改这些设置。您可以选择新的天数并保存更改，从而轻松更改保留期。

> **Warning**
>
> 在配置跨区域快照复制后，您无法修改目标 AWS 区域。
> 如果要将快照复制到其他 AWS 区域，请先禁用跨区域快照复制。然后使用新的目标 AWS 区域和保留期重新启用它。在禁用跨区域快照复制后，将删除任何复制的自动快照。因此，在禁用跨区域快照复制之前，您应该确定是否具有任何要保留的快照，并将其复制到手动快照。

修改跨区域快照

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群，然后选择要修改其快照的集群。
3. 对于操作，请选择配置跨区域快照以显示快照的属性。
4. 输入快照定义的修订属性，然后选择保存。

### 删除手动快照

您可以通过在快照列表中选择一个或多个快照来删除手动快照。

删除手动快照

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群、快照，然后选择要删除的快照。
3. 对于操作，选择删除快照以删除快照。
4. 确认删除列出的快照，然后选择删除。

## 将集群注册到 AWS Glue Data Catalog

您可以将整个集群注册到 AWS Glue Data Catalog 并创建由 AWS Glue 管理的目录。您可以使用任何支持 Apache Iceberg REST API 的 SQL 引擎访问这些目录。有关从 Amazon Redshift 创建与 Apache Iceberg 兼容的目录的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Apache Iceberg compatibility for Amazon Redshift。

将集群注册到 AWS Glue Data Catalog

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。列出您的账户在当前 AWS 区域 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。如果您没有集群，请选择创建集群来创建一个集群。
3. 选择要注册的集群的名称。
4. 从操作中，选择注册到 AWS Glue Data Catalog。将显示注册到 AWS Glue Data Catalog 弹出框。

5. 在目标账户 ID 下输入要向其注册集群的 AWS 账户 ID。这是将目录保存在 AWS Glue Data Catalog 中的账户 ID。
6. 在将命名空间注册为下输入名称。这将是 Data Catalog 中集群的名称。
7. 选择注册。您将进入 AWS Lake Formation 控制台。
8. 按照在 AWS Lake Formation 中创建目录的过程进行操作。有关创建目录的信息，请参阅《AWS Lake Formation 开发人员指南》中的 Bringing Amazon Redshift data into the AWS Glue Data Catalog。

# 多可用区部署

Amazon Redshift 支持对预置的 RG 或 RA3 集群使用多可用区部署。通过使用多可用区部署，在一个可用区出现意外事件的故障场景中，您的 Amazon Redshift 数据仓库可以继续运行。多可用区部署将计算资源部署在两个可用区 (AZ) 中，这些计算资源可通过单个端点访问。如果某个可用区出现彻底故障，第二个可用区中的剩余计算资源可用于继续处理工作负载。运行多可用区数据仓库时，Amazon Redshift 对 RG 或 RA3 收取相同的每小时计算费率。存储成本保持不变，因为它会在 AWS 区域中的所有可用区之间共享。

目前，Amazon Redshift 支持零恢复点目标（RPO），可以在出现故障时确保数据为最新。通过多可用区部署，Amazon Redshift 进一步增强了其现有的恢复能力，并可缩短其恢复时间目标 (RTO)。之所以能做到这一点，是因为多可用区部署可以更快地从故障或灾难中恢复，从而可将 Amazon Redshift 服务水平协议 (SLA) 提高到 99.99%，而单可用区数据仓库的 SLA 为 99.9%。

## 设置多可用区部署

要设置多可用区部署，请选择多可用区选项，并指定要在每个可用区中预置的计算节点数。Amazon Redshift 会自动在两个可用区中部署相等的计算资源，在正常操作期间，所有计算资源均始终可用于读取和写入处理，并有机会利用辅助集群中的额外计算资源来协助提升性能。这样，一个多可用区部署就可以用作一个数据仓库，并具有一个终端节点，从而在发生灾难时无需更改应用程序。尽管多可用区部署仅使用一个可用区中的计算资源来处理单个查询，但它可以自动将部分同时发生的查询分配到两个可用区进行处理，这有时（但并非总是）可以提升高并发工作负载的总吞吐量。请务必记住多可用区的主要功能，即为您的集群提供高可用性解决方案，以确保从计划外停机或中断事件中快速恢复。

如果您的主要动机是提高分析工作负载的性能，那么除了多可用区之外，我们还推荐使用其它解决方案，例如：

- 调整为更大的节点实例类型（调整集群大小）
- 在集群上启用并发扩展（并发扩展）

- 使用数据共享架构来隔离工作负载（数据共享概述）

您也可以将现有单可用区数据仓库转换为多可用区数据仓库，反之亦然。除了在第二个可用区预置额外的计算资源外，其余均保持不变。从现有单可用区集群迁移到多可用区时，您可能需要将所需集群节点数量翻倍，以保持单个查询的性能。

如果可用区出现故障，Amazon Redshift 会自动使用剩余可用区中的资源继续运行。但是，用户连接可能会丢失，必须重新建立连接。此外，在出现故障的可用区中运行的查询会停止，必须进行重试。但是，您可以立即重新连接到集群并重新计划查询，Amazon Redshift 将在剩余可用区中处理查询。当多可用区数据仓库正在恢复时，在故障发生时或之后发出的查询可能会遇到运行时延迟。

> **Note**
>
> 为了获得更好的性能和更高的可用性，我们建议您在多可用区集群中使用 SNAPSHOT ISOLATION。有关更多信息，请参阅创建数据库。

## 限制

多可用区数据仓库与单可用区数据仓库具有相同的功能性，但适用于多可用区数据仓库的以下限制除外：

- 您无法创建不加密的多可用区数据仓库。在创建新多可用区数据仓库或者将单可用区数据仓库转换为多可用区数据仓库时，请务必添加加密。
- 您无法为任何 RG 或 RA3 实例类型创建单节点多可用区部署。在创建多可用区部署时，为每个可用区选择 2 个或更多节点。
- 对于所支持的可用区少于三个的子网配置，Amazon Redshift 不支持这种子网配置。换句话说，所配置的子网组需要三个或更多子网。
- 您无法将多可用区部署重新定位到另一个可用区。使用多可用区部署时，Amazon Redshift 会自动确定和执行重新定位。
- 您无法暂停或恢复多可用区部署。
- 您无法在支持的端口范围（5431 到 5455 和 8191 到 8215）之外运行多可用区部署。
- 您不能在多可用区部署中使用 STL、SVCS、SVL、SVV、STV 视图，因为它们仅支持系统监控视图（SYS_* 视图）。请更改您的监控查询以使用系统监控视图（SYS_* 视图）。
- 您无法将弹性 IP 地址附加到启用了多可用区的现有集群。
- 您无法将附有弹性 IP 地址的集群从单可用区转换为多可用区。

- 以下 AWS 区域 提供 Amazon Redshift 多可用区部署：

> **Note**
>
> RG 实例在部分区域可用。请参阅 AWS 区域中的 RG 节点类型可用性。

  - 美国东部（俄亥俄州）(us-east-2)
  - 美国东部（弗吉尼亚北部）(us-east-1)
  - 美国西部（俄勒冈州）(us-west-2)
  - 非洲（开普敦）(af-south-1)
  - 亚太地区（香港）(ap-east-1)
  - 亚太地区（台北）（ap-east-2）
  - 亚太地区（海得拉巴）(ap-south-2)
  - 亚太地区（雅加达）（ap-southeast-3）
  - 亚太地区（马来西亚）（ap-southeast-5）
  - 亚太地区（墨尔本）(ap-southeast-4)
  - 亚太地区（孟买）(ap-south-1)
  - 亚太地区（大阪）(ap-northeast-3)
  - 亚太地区（首尔）(ap-northeast-2)
  - 亚太地区（新加坡）(ap-southeast-1)
  - 亚太地区（悉尼）(ap-southeast-2)
  - 亚太地区（新西兰）（ap-southeast-6）
  - 亚太地区（泰国）（ap-southeast-7）
  - 亚太地区（东京）(ap-northeast-1)
  - 加拿大（中部）(ca-central-1)
  - 中国（北京）（cn-north-1）
  - 中国（宁夏）（cn-northwest-1）
  - 欧洲地区（法兰克福）(eu-central-1)
  - 欧洲地区（爱尔兰）(eu-west-1)
  - 欧洲地区（伦敦）(eu-west-2)
  - 欧洲地区（米兰）(eu-south-1)

- 欧洲（巴黎）（eu-west-3）
- 欧洲（西班牙）(eu-south-2)
- 欧洲地区（斯德哥尔摩）(eu-north-1)
- 欧洲（苏黎世）(eu-central-2)
- 以色列（特拉维夫）（il-central-1）
- 墨西哥（中部）（mx-central-1）
- 中东（巴林）(me-south-1)
- 中东（阿联酋）(me-central-1)
- 南美洲（圣保罗）（sa-east-1）
- AWS GovCloud（美国东部）(us-gov-east-1)
- AWS GovCloud（美国西部）(us-gov-west-1)

- 对于可公开访问的多可用区数据仓库，其支持的 VPC 安全组数量，比单可用区数据仓库和私密访问多可用区数据仓库少 1 个。

## 创建新集群时设置多可用区

Amazon Redshift 多可用区一次支持两个可用区。Amazon Redshift 会根据选择的子网组配置自动选择可用区。您可以将现有单可用区数据仓库转换为多可用区参考，也可以从快照还原以将其配置到多可用区数据仓库中。

使用 Amazon Redshift 控制台，您可以轻松创建新的多可用区部署。要使用 Amazon Redshift 控制台创建新的多可用区部署，请在创建数据仓库时选择多可用区选项。指定单个可用区中所需的计算节点数量，Amazon Redshift 将在两个可用区分别部署相同数量的节点。在正常运行期间，所有节点都将用于执行读取和写入工作负载处理。您也可以通过 AWS CLI `create-cluster` 命令，使用 `multi-az` 参数创建新的多可用区数据仓库。

您可以使用 Amazon Redshift 控制台或使用带有 `multi-az` 参数的 AWS CLI `modify-cluster` 命令，将现有的单可用区数据仓库转换为多可用区数据仓库。或者，您可以使用 Amazon Redshift 控制台或使用带有 `multi-az` 参数的 AWS CLI `restore-from-cluster-snapshot` 命令，从快照进行还原，以将单可用区数据仓库配置为多可用区数据仓库。

多可用区部署仅支持使用 Amazon Redshift 托管存储（RMS）的 RG 或 RA3 节点类型。Amazon Redshift 将数据存储在 RMS 中，RMS 使用 Amazon S3，可以在 AWS 区域 中的所有可用区中访问，无需在 Amazon Redshift 级别复制数据。

创建新集群时，您可以使用 Amazon Redshift 控制台或 AWS Command Line Interface 设置多可用区部署。

使用控制台

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择预置集群控制面板，然后选择集群。列出您的账户在当前 AWS 区域 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。
3. 选择创建集群按钮以打开创建集群页面。
4. 输入集群的属性。有关创建集群的一般信息，请参阅 创建集群。
5. 从节点类型下拉列表中选择其中一个 RG 或 RA3 节点类型。仅当您选择 RG 或 RA3 节点类型时，可用区配置选项才可用。
6. 在可用区配置下，选择多可用区。
7. 在每个可用区的节点数下面，为您集群输入至少两个节点。
8. 您可以选择加载样本数据，也可以自带数据：
   - 在示例数据中，选择加载示例数据将示例数据集加载到您的 Amazon Redshift 集群。Amazon Redshift 会将示例数据集 Tickit 加载到默认的 dev 数据库和 public schema。Amazon Redshift 会自动将示例数据集加载到您的 Amazon Redshift 集群中。您可以使用查询编辑器查询数据。
   - 要将您自己的数据带到您的 Amazon Redshift 集群，请按照将您自己的数据带入 Amazon Redshift 中的步骤操作。
9. 向下滚动到其他配置，展开网络和安全，并确保您接受默认的集群子网组或选择另一个集群子网组。如果您选择另一个集群子网组，请确保您选择的子网组中有 3 个可用区。
10. 在其他配置下，展开数据库配置。
11. 要使用自定义 AWS KMS 密钥而不是默认的 AWS 拥有的密钥，请单击数据库加密下的自定义加密设置。
12. 在选择 KMS 密钥下，您可以选择 AWS Key Management Service 密钥或输入 ARN。或者，可以在 AWS Key Management Service 控制台中单击创建 AWS Key Management Service 密钥。有关创建 KMS 密钥的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的创建密钥。
13. 单击创建集群。成功创建集群后，您可以在集群详细信息页面中查看详细信息。您可以使用 SQL 客户端加载和查询数据。

使用 AWS Command Line Interface

在创建集群时使用 AWS Command Line Interface 设置多可用区

- 从 AWS CLI 使用 `create-cluster` 命令和 `multi-az` 参数，如下所示。

```
aws redshift create-cluster
    --port 5439
    --master-username master
    --master-user-password #####
    --node-type rg.4xlarge
    --number-of-nodes 2
    --profile maz-test
    --endpoint-url https://redshift.eu-west-1.amazonaws.com
    --region eu-west-1
    --cluster-identifier test-maz
    --multi-az
    --maintenance-track-name CURRENT
    --encrypted
```

## 为从快照还原的数据仓库设置多可用区

要通过从快照进行还原来创建新的多可用区集群，请完成以下过程。

使用控制台

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群、快照，然后选择要使用的快照。
3. 选择还原快照，还原到预置集群。
4. 输入集群的属性。有关创建集群的一般信息，请参阅 创建集群。
5. 从节点类型下拉列表中选择其中一个 RG 或 RA3 节点类型。仅当您选择 RG 或 RA3 节点类型时，可用区配置选项才可用。
6. 在可用区配置下，选择多可用区。
7. 在每个可用区的节点数下面，为您集群输入至少两个节点。
8. 您可以选择加载样本数据，也可以自带数据：

- 在示例数据中，选择加载示例数据将示例数据集加载到您的 Amazon Redshift 集群。Amazon Redshift 会将示例数据集 Tickit 加载到默认的 dev 数据库和 public schema。Amazon Redshift 会自动将示例数据集加载到您的 Amazon Redshift 集群中。您可以使用查询编辑器查询数据。
- 要将您自己的数据带到您的 Amazon Redshift 集群，请按照将数据从 Amazon S3 加载到 Amazon Redshift 中的步骤操作。

9. 向下滚动到其他配置，展开网络和安全，并确保您接受默认的集群子网组或选择另一个集群子网组。如果您选择另一个集群子网组，请确保您选择的子网组中有 3 个可用区。
10. 在其他配置下，展开数据库配置。
11. 在数据库加密下，要使用默认的 AWS 拥有的密钥以外的自定义 KMS 密钥，请单击自定义加密设置。默认取消选择此选项。
12. 在选择 KMS 密钥下，您可以选择 AWS Key Management Service 密钥或输入 ARN。或者，可以在 AWS Key Management Service 控制台中单击创建 AWS Key Management Service 密钥。有关创建 KMS 密钥的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的创建密钥。
13. 单击从快照还原集群。成功还原集群后，您可以在集群详细信息页面中查看详细信息。

使用 AWS Command Line Interface

- 从 AWS CLI，按以下所示使用 `restore-from-cluster-snapshot` 命令。

```
aws redshift restore-from-cluster-snapshot
--region eu-west-1
--multi-az
--snapshot-identifier test-snap1
--cluster-identifier test-saz-11
--endpoint-url https://redshift.eu-west-1.amazonaws.com/
```

## 将单可用区数据仓库转换为多可用区数据仓库

通过将单可用区数据仓库转换为多可用区数据仓库，可以确保您的数据仓库获得 99.99% SLA 保证的高可用性。即使使用多可用区数据仓库，单个查询的性能也将保持不变。对于更高并发度的工作负载，由于 Amazon Redshift 可以使用两个可用区中的计算资源执行请求，因此总吞吐量会得到提升。

> **Note**
>
> Amazon Redshift 不允许您在从单可用区转换为多可用区时拆分现有计算资源，反之亦然。系统不支持此操作以维护稳定的单个查询性能。

### 使用控制台

使用控制台将单可用区集群转换为多可用区数据仓库

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择预置集群控制面板，然后选择集群。列出您的账户在当前 AWS 区域 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。
3. 选择您要转换为多可用区部署的集群。此时会显示集群详细信息页面。
4. 对于操作，选择激活多可用区。此时将显示修改摘要页面。单击激活多可用区。
5. 出现错误时，请执行以下操作之一，然后单击激活多可用区。
   - 集群加密 – 在集群详细信息页面的“属性”选项卡下，选择“数据库配置”部分中的属性以编辑加密设置。
   - 子网组 – 选择子网组，单击子网组链接来编辑集群子网组设置。如果您选择另一个集群子网组，请确保您选择的子网组中有 3 个可用区。
   - 端口设置 – 在集群详细信息页面的“属性”选项卡下，选择“数据库配置”部分中的属性以编辑端口设置。
6. 您可以使用 SQL 客户端加载和查询数据。

### 使用 AWS Command Line Interface

- 从 AWS CLI 使用 `modify-cluster` 命令和 `multi-az` 参数，如下所示。

```
aws redshift modify-cluster
    --profile maz-test
    --endpoint-url https://redshift.eu-west-1.amazonaws.com
    --region eu-west-1
    --cluster-identifier test-maz-11
    --multi-az
```

您不能在多可用区部署中使用 STL、SVCS、SVL、SVV、或 STV 视图，因为它们仅支持系统监控视图（SYS_* 视图）。请更改您的监控查询以使用系统监控视图（SYS_* 视图）。

## 将多可用区数据仓库转换为多可用区数据仓库

将多可用区数据仓库转换为单可用区数据仓库后，您的数据仓库将无法获得多可用区提供的 99.99% 的 SLA 保障。单个查询的性能将保持不变，但总吞吐量将受到影响，因为第二个可用区的计算资源将不可用。即使使用单可用区，您也可以选择启用并发扩展，以便自动扩展吞吐量来实现稳定的性能。

> **Note**
>
> Amazon Redshift 不允许您在从单可用区转换为多可用区时拆分现有计算资源，反之亦然。系统不支持此操作以维护稳定的单个查询性能。

### 使用控制台

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择预置集群控制面板，然后选择集群。列出您的账户在当前 AWS 区域 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。
3. 选择您要转换为多可用区部署的集群。此时会显示集群详细信息页面。
4. 对于操作，选择停用多可用区。此时将显示修改摘要页面。单击停用多可用区。

### 使用 AWS Command Line Interface

- 从 AWS CLI 使用 `modify-cluster` 命令和 `no-multi-az` 参数，如下所示。

```
aws redshift modify-cluster
    --profile maz-test
    --endpoint-url  https://redshift.eu-west-1.amazonaws.com
    --region eu-west-1
    --cluster-identifier test-maz-11
    --no-multi-az
```

数据仓库转换为单可用区后，将失去 99.99 的 SLA 保障。总体吞吐量也将受到影响。保存更改后，您可以在集群详细信息页面中查看详细信息。

# 多可用区数据仓库大小调整

您可以调整多可用区数据仓库的大小，并指定不同于数据仓库当前配置的节点数量或节点类型。

使用控制台

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择预置集群控制面板，然后选择集群。列出您的账户在当前 AWS 区域 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。
3. 选择要调整大小的多可用区数据仓库所在的集群。此时会显示集群详细信息页面。
4. 对于操作，选择调整大小。此时将显示“调整集群大小”页面。
5. 按照页面上的说明操作。您可以立即调整集群大小，在特定时间调整一次，或者按计划增加和减小集群大小。
6. 在新配置下，从“节点类型”下拉列表中选择其中一个 RG 或 RA3 节点类型。
7. 单击调整集群大小。

使用 AWS Command Line Interface

要调整多可用区数据仓库的大小，请使用 AWS Command Line Interface

- 从 AWS CLI 中，使用 `resize-cluster` 命令更改单个可用区的节点数，如下所示。

```
aws redshift resize-cluster \
    --cluster-identifier test-maz-11
    --cluster-type multi-node
    --node-type rg.4xlarge
    --number-of-nodes 6
```

# 多可用区部署失效转移

您的多可用区数据仓库是同时部署在两个可用区中的计算资源集合。部署在主可用区的计算资源称为主计算，辅助可用区中的计算资源称为辅助计算。在出现一些不太可能发生事件的时候，例如可用区或基础设施故障，多可用区数据仓库无需任何用户干预即可自动恢复。恢复过程包括从主计算失效转移到辅助计算，并将辅助计算资源指定为主计算。此外，还会在第三个可用区中预置新的辅助计算资源。自动恢复过程以 RTO 和 RPO 来衡量。

- 恢复时间目标 (RTO)：灾难后系统恢复工作状态所需的时间。换言之，RTO 用于衡量停机时间。
- 恢复点目标 (RPO)：可能丢失的数据量（按时间衡量）。对于 Amazon Redshift 多可用区数据仓库，RPO 通常为零，因为所有数据都存储在 Amazon Redshift 托管存储 (RMS) 中，由 Amazon Simple Storage Service 提供支持，这是默认具有高持久性和高可用性的服务。

> **Note**
>
> 发生失效转移后，单个查询的性能不会改变。由于可用区之一中的计算资源不可用，在短时间内，您的数据仓库的总体吞吐量会降低。但是，Amazon Redshift 将自动获取另一个可用区中的容量，以确保恢复相同的数据仓库处理能力。

在自动恢复过程之外，您还可以使用失效转移主计算选项，为数据仓库手动触发此过程。您可以使用这种方法来测试多可用区如何让您的应用程序实现更高的可用性和更好的连续性。

使用控制台

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 请执行以下操作之一：
   - 在导航菜单上，选择集群。在集群下，选择一个集群。此时会显示集群详细信息页面。
   - 从集群控制面板中选择一个集群。
3. 从操作中，选择失效转移主计算。
4. 当系统提示您确认时，单击确认。

使用 AWS Command Line Interface

- 从 AWS CLI，按以下所示使用 `failover-primary-compute` 命令。

```
aws redshift failover-primary-compute
    --profile maz-test
    --endpoint-url  https://redshift.eu-west-1.amazonaws.com
    --region eu-west-1
    --cluster-identifier test-maz-11
```

确认上述操作后，Amazon Redshift 将执行自动恢复步骤，这些步骤与从可用区或基础设施故障中进行自动恢复时相同。该过程将导致主可用区中的计算节点不可用，辅助可用区中的计算资源将被指定为主计算节点。集群恢复成功完成后，多可用区部署将变得可用。在另外的第三个可用区变为可用后，您的多可用区数据仓库还会在其中自动预置新的辅助计算资源。

在此过程中，控制台上的集群状态一直显示为正在修改，因为集群会自动恢复并重新配置回多可用区部署设置。集群可以立即接受新连接。可能会删除现有连接和正在进行的查询。您可以立即重试连接和查询。

## 查看多可用区数据仓库的查询和负载

无论集群的类型、大小和状态（暂停或恢复）如何，您都可以查看过去 7 天内运行的查询的信息。

查询和加载页面上显示的信息使用 Amazon Redshift 系统表（SYS_* 视图）中的信息来填入。您可以利用此信息显示有关查询的额外信息，并提供滚动 7 天的保留期。查询诊断变得更快，使您可以按数据库、用户名或 SQL 语句类型筛选数据。要查看这些附加筛选条件以及运行的所有查询的信息，请注意以下先决条件：

- 必须通过选择连接到数据库来连接到数据库。
- 您的数据库用户必须具有 sys:operator 或 sys:monitor 角色和权限才能执行查询监控。有关系统角色的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Amazon Redshift 系统定义的角色。

连接到数据库后，您将看到这些额外的筛选条件和查询信息。

显示来自查询和加载的查询性能数据

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择查询和加载以便显示您的账户的查询列表。
3. 您可能必须连接到数据库才能查看其他筛选条件。如果需要，请单击连接到数据库，然后按照提示连接到数据库。

   默认情况下，该列表显示过去 24 小时中所有集群的查询。您可以在控制台中更改显示日期的范围。

显示来自查询监控的查询性能数据

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。在集群下面，选择一个集群。
3. 选择查询监控。
4. 根据集群的配置或版本，您可能必须连接到数据库才能查看其他筛选条件。如果需要，请单击连接到数据库，然后按照提示连接到数据库。

## 监控多可用区部署中的查询

多可用区部署使用在两个可用区中部署的计算资源，并且在给定可用区中的资源不可用时仍可以继续运行。始终使用所有计算资源。这样就可以通过主动-主动方式在两个可用区内执行完全操作，包括读取和写入操作。

您可以在 pg_catalog Schema 中查询 SYS_ 视图以监控多可用区部署中的查询运行时。SYS_ 视图显示来自主集群和辅助集群的查询运行时活动或统计信息。有关监控视图的列表，请参阅监控视图。

按照以下步骤监控多可用区部署中每个可用区的查询运行时：

1. 导航到 Amazon Redshift 控制台并连接到多可用区部署中的数据库，然后通过查询编辑器运行查询。
2. 在多可用区 Amazon Redshift 部署上运行任何示例查询。
3. 对于多可用区部署，您可以使用 SYS_QUERY_HISTORY 表中的 compute_type 列来识别查询及运行查询的可用区。primary 代表在多可用区部署的主集群上运行的查询，secondary 代表在多可用区部署的辅助集群上运行的查询。

   以下查询使用 compute_type 列来监控查询。

```
select (compute_type) as compute_type, left(query_text, 50) query_text from
 sys_query_history order by start_time desc;

 compute_type | query_text
--------------+-------------------------
   secondary  | select count(*) from t1;
```

## 终止对集群的查询

该过程适用于多可用区和单可用区集群。

终止查询

您也可以使用查询页面终止当前正在运行的查询。

您的数据库用户必须具有 sys:operator 角色和权限才能结束正在运行的查询。有关系统角色的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Amazon Redshift 系统定义的角色。

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择查询和加载以便显示您的账户的查询列表。
3. 在列表中选择要终止的正在运行的查询，然后选择终止查询。

# 监控 Amazon Redshift 集群性能

Amazon Redshift 提供性能指标和数据，以便您可以跟踪集群和数据库的运行状况及性能。在本部分中，我们将讨论您可以在 Amazon Redshift（尤其是 Amazon Redshift 控制台）中使用的数据类型。

您可以在 Amazon Redshift 控制台中使用的性能数据分为两类：

- Amazon CloudWatch 指标 – Amazon CloudWatch 指标有助于您监控集群的物理方面，例如 CPU 使用率、延迟和吞吐量。指标数据直接显示在 Amazon Redshift 控制台中。您还可以在 CloudWatch 控制台中查看它。或者，您也可以通过处理指标的任何其他方式使用它，例如使用 AWS CLI 或其中一个 AWS 开发工具包。
- 查询/加载性能数据 – 性能数据可以帮助您监控数据库活动和性能。此类数据汇集于 Amazon Redshift 控制台中，有助于您轻松将通过 CloudWatch 指标了解的情况与特定数据库查询和加载事件相关联。您也可以创建自己的自定义性能查询，并直接在数据库上运行这些查询。查询和加载性能数据仅显示在 Amazon Redshift 控制台中。此类数据不会作为 CloudWatch 指标进行发布。

性能数据集成到 Amazon Redshift 控制台中，从而通过以下方式实现更丰富的体验：

- 当您查看某个集群时，与之关联的性能数据会根据具体情况显示出来，例如在您需要针对集群做出决策（例如调整大小）时。

- 与 CloudWatch 相比，某些性能指标以经过更适当换算的单位显示在 Amazon Redshift 控制台中。例如，`WriteThroughput` 以 GB/s 显示（而在 CloudWatch 中，则以 Bytes/s 显示），该单位与典型的节点存储空间的相关性更高。
- 您可以在同一图表上轻松地将集群节点的性能数据显示在一起。这样，您就可以轻松监控集群的所有节点的性能。您还可以查看每个节点的性能数据。

Amazon Redshift 提供性能数据（包括 CloudWatch 指标以及查询和加载数据），无需任何额外费用。系统会记录每分钟的性能数据。您可以通过 Amazon Redshift 控制台访问性能数据的历史值。有关使用 CloudWatch 访问作为 CloudWatch 指标公开的 Amazon Redshift 性能数据的详细信息，请参阅《Amazon CloudWatch 用户指南》中的什么是 CloudWatch？。

## Amazon Redshift 中的性能数据

利用 Amazon Redshift 的 CloudWatch 指标，您可以获取有关集群的运行状况和性能的信息，并查看节点级别的信息。在使用这些指标时，请注意，每个指标都有一个或多个维度与之关联。这些维度告诉您指标适用于什么，即指标的范围。Amazon Redshift 有以下两个维度。

- 拥有 `NodeID` 维度的指标指的是提供集群节点性能数据的指标，这组指标包括领导节点和计算节点。例如以下指标：`CPUUtilization`、`ReadIOPS` 和 `WriteIOPS`。
- 只有 `ClusterIdentifier` 维度的指标指的是提供集群性能数据的指标。例如以下指标：`HealthStatus` 和 `MaintenanceMode`。

> **Note**
>
> 在使用某些指标时，特定于集群的指标表示节点行为的聚合。在这类情况下，由于领导节点的行为与计算节点聚合，在解释指标值时请小心谨慎。

有关 CloudWatch 指标和维度的一般信息，请参阅《Amazon CloudWatch 用户指南》中的 CloudWatch 概念。

有关 Amazon Redshift 的 CloudWatch 指标的进一步说明，请参阅以下部分。

主题

- Amazon Redshift 指标
- Amazon Redshift 指标的维度
- Amazon Redshift 查询和加载性能数据

## Amazon Redshift 指标

AWS/Redshift 命名空间包括以下指标。除非另有说明，否则每隔 1 分钟收集一次指标。

| 指标 | 说明 |
| --- | --- |
| CommitQueueLength | 在给定的时间点等待提交的事务数。<br><br>单位：计数<br><br>维度：ClusterIdentifier |
| ConcurrencyScaling ActiveClusters | 在任何给定时间主动处理查询的并发扩展集群的数量。<br><br>单位：计数<br><br>维度：ClusterIdentifier |
| ConcurrencyScaling Seconds | 具有活动查询处理活动的并发扩展集群使用的秒数。<br><br>单位：计数<br><br>维度：ClusterIdentifier |
| CPUUtilization | CPU 使用百分率。对于集群，该指标代表所有节点 (领导节点和计算节点) CPU 使用率值的总和。<br><br>单位：百分比<br><br>维度: ClusterIdentifier ，NodeID<br><br>维度：ClusterIdentifier |
| DatabaseConnections | 集群中的数据库连接数量。<br><br>单位：计数<br><br>维度：ClusterIdentifier |
| HealthStatus | 表示集群的运行状况检查。每分钟集群连接到其数据库并执行一次简单的查询。如果可以成功执行此操作，则表示集群的运行状况良好。否则，视为集群运行状况不佳。当数据库集群负载极 |

| 指标 | 说明 |
| --- | --- |
| | 重，或者集群上的数据库存在配置问题时，集群会出现运营状况不佳的情况。<br><br>**Note**<br>在 Amazon CloudWatch 中，此指标报告为 1 或 0，而在 Amazon Redshift 控制台中，为了方便起见，此指标显示为 HEALTHY 或 UNHEALTHY 。当该指标在 Amazon Redshift 控制台中显示时，采样平均值会被忽略，仅显示 HEALTHY 或 UNHEALTHY 。在 Amazon CloudWatch 中，由于采样问题，可能出现 1 和 0 以外的值。HealthStatus 的所有 1 以下的值均报告为 0 (UNHEALTHY )。<br><br>单位：计数 (1/0)（在 Amazon Redshift 控制台中为 HEALTHY/UNHEALTHY ）<br><br>维度：ClusterIdentifier |
| MaintenanceMode | 表示集群是否处于维护模式。<br><br>**Note**<br>在 Amazon CloudWatch 中，此指标报告为 1 或 0，而在 Amazon Redshift 控制台中，为了方便起见，此指标显示为 ON 或 OFF。当该指标在 Amazon Redshift 控制台中显示时，采样平均值会被忽略，仅显示 ON 或 OFF。在 Amazon CloudWatch 中，由于采样问题，可能出现 1 和 0 以外的值。MaintenanceMode 的所有 0 以上的值均报告为 1 (ON)。<br><br>单位：计数 (1/0)（在 Amazon Redshift 控制台中为 ONOFF ）。<br><br>维度：ClusterIdentifier |

| 指标 | 说明 |
| --- | --- |
| MaxConfiguredConcurrencyScalingClusters | 从参数组配置的最大并发扩展集群数。有关更多信息，请参阅 Amazon Redshift 参数组。<br><br>单位：计数<br><br>维度：ClusterIdentifier |
| NetworkReceiveThroughput | 节点或集群接收数据的速率。<br><br>单位：字节/秒（在 Amazon Redshift 控制台为 MB/s）<br><br>维度: ClusterIdentifier ，NodeID<br><br>维度：ClusterIdentifier |
| NetworkTransmitThroughput | 节点或集群写入数据的速率。<br><br>单位：字节/秒（在 Amazon Redshift 控制台为 MB/s）<br><br>维度: ClusterIdentifier ，NodeID<br><br>维度：ClusterIdentifier |
| PercentageDiskSpaceUsed | 已使用磁盘空间的百分比。<br><br>单位：百分比<br><br>维度：ClusterIdentifier<br><br>维度: ClusterIdentifier ，NodeID |
| QueriesCompletedPerSecond | 每秒完成的平均查询数。每隔 5 分钟报告一次。单节点集群不支持该指标。<br><br>单位：计数/秒<br><br>维度: ClusterIdentifier ，latency<br><br>维度: ClusterIdentifier ，wlmid |

| 指标 | 说明 |
| --- | --- |
| QueryDuration | 完成查询的平均时间量。每隔 5 分钟报告一次。单节点集群不支持该指标。<br><br>单位：微秒<br><br>维度：ClusterIdentifier 、NodeID、latency<br><br>维度: ClusterIdentifier , latency<br><br>维度：ClusterIdentifier 、NodeID、wlmid |
| QueryRuntimeBreakdown | 查询在各个查询阶段运行所花费的总时间。每隔 5 分钟报告一次。<br><br>单位：毫秒<br><br>维度：ClusterIdentifier、NodeID、stage<br><br>维度：ClusterIdentifier、stage |
| ReadIOPS | 每秒平均磁盘读取 操作数。<br><br>单位：计数/秒<br><br>维度: ClusterIdentifier , NodeID<br><br>维度：ClusterIdentifier |
| ReadLatency | 磁盘读取 I/O 操作所需的平均时间。<br><br>单位：秒<br><br>维度: ClusterIdentifier , NodeID<br><br>维度：ClusterIdentifier |

| 指标 | 说明 |
| --- | --- |
| ReadThroughput | 每秒从磁盘读取的平均字节数。<br><br>单位：字节（在 Amazon Redshift 控制台为 GB/s）<br><br>维度: ClusterIdentifier , NodeID<br><br>维度：ClusterIdentifier |
| RedshiftManagedStorageTotalCapacity | 总托管式存储容量。<br><br>单位：MB<br><br>维度：ClusterIdentifier |
| RgUniqueSnapshotStorage | 在 RG 实例系列的所有活动的手动快照中存储的唯一数据块的兆字节数。<br><br>单位：MB |
| TotalTableCount | 特定时间点的表总数。此总数包括永久表、临时表、数据共享表、外部表和实体化视图。此计数中不包括视图和系统表。<br><br>单位：计数<br><br>维度：ClusterIdentifier |
| WLMQueueLength | 等待进入工作负载管理 (WLM) 队列的查询数。<br><br>单位：计数<br><br>维度: ClusterIdentifier , service class<br><br>维度: ClusterIdentifier , QueueName |

| 指标 | 说明 |
| --- | --- |
| WLMQueueWaitTime | 在工作负载管理 (WLM) 队列中等待的查询总时间。每隔 5 分钟报告一次。<br><br>单位：毫秒。<br><br>维度: ClusterIdentifier , QueryPriority<br><br>维度: ClusterIdentifier , wlmid<br><br>维度: ClusterIdentifier , QueueName |
| WLMQueriesCompletedPerSecond | 每秒为工作负载管理 (WLM) 队列完成的平均查询数。每隔 5 分钟报告一次。单节点集群不支持该指标。<br><br>单位：计数/秒<br><br>维度: ClusterIdentifier , wlmid<br><br>维度: ClusterIdentifier , QueueName |
| WLMQueryDuration | 为工作负载管理 (WLM) 队列完成查询的平均时间量。每隔 5 分钟报告一次。单节点集群不支持该指标。<br><br>单位：微秒<br><br>维度: ClusterIdentifier , wlmid<br><br>维度: ClusterIdentifier , QueueName |
| WLMRunningQueries | 对于每个 WLM 队列，从主集群和并发扩展集群运行的查询数。<br><br>单位：计数<br><br>维度: ClusterIdentifier , wlmid<br><br>维度: ClusterIdentifier , QueueName |

| 指标 | 说明 |
| --- | --- |
| WriteIOPS | 每秒平均磁盘写入操作数。<br>单位：计数/秒<br>维度: ClusterIdentifier ，NodeID<br>维度：ClusterIdentifier |
| WriteLatency | 磁盘写入 I/O 操作所需的平均时间。<br>单位：秒<br>维度: ClusterIdentifier ，NodeID<br>维度：ClusterIdentifier |
| WriteThroughput | 每秒写入磁盘的平均字节数。<br>单位：字节（在 Amazon Redshift 控制台为 GB/s）<br>维度: ClusterIdentifier ，NodeID<br>维度：ClusterIdentifier |
| SchemaQuota | 为 schema 配置的配额。<br>单位：MB<br>维度：ClusterIdentifier 、Database、Schema<br>周期/推送：Periodic<br>频率：5 minutes<br>停止条件：已删除 Schema 或移除配额 |

| 指标 | 说明 |
| --- | --- |
| NumExceededSchemaQuotas | 超出配额的 schema 数。<br>单位：计数<br>维度：ClusterIdentifier<br>周期/推送：Periodic<br>频率：5 minutes<br>停止标准：不适用 |
| StorageUsed | schema 使用的磁盘或存储空间。<br>单位：MB<br>维度：ClusterIdentifier 、Database、Schema<br>周期/推送：Periodic<br>频率：5 minutes<br>停止条件：已删除 Schema 或移除配额 |
| PercentageQuotaUsed | 使用的磁盘或存储空间相对于配置的 schema 配额的百分比。<br>单位：百分比<br>维度：ClusterIdentifier 、Database、Schema<br>周期/推送：Periodic<br>频率：5 minutes<br>停止条件：已删除 Schema 或移除配额 |

| 指标 | 说明 |
| --- | --- |
| UsageLimitAvailable | 根据 FeatureType，UsageLimitAvailable 将返回以下内容：<br>• 如果 FeatureType 为 CONCURRENCY_SCALING ，则 UsageLimitAvailable 将返回可由并发扩展使用的总时间长度（以 1 分钟为增量）。此时间不计为可用于 CONCURRENCY_SCALING_AUTO_TASK 的时间，反之亦然。<br>• 如果 FeatureType 为 CONCURRENCY_SCALING_AUTO_TASK ，则 UsageLimitAvailable 将返回并发扩展可用于自动优化任务的总时间长度。此时间不计为可用于 CONCURRENCY_SCALING 的时间，反之亦然。<br>• 如果 FeatureType 为 EXTRA_COMPUTE_FOR_AUTOMATIC_OPTIMIZATION ，则 UsageLimitAvailable 将返回可用于自动优化任务的总时间长度。<br>• 如果 FeatureType 为 CROSS_REGION_DATASHARING ，则 UsageLimitAvailable 将返回可扫描的数据总量（以 1 TB 为增量）。<br>• 如果 FeatureType 为 SPECTRUM，则 UsageLimitAvailable 将返回可扫描的数据总量（以 1 TB 为增量）。<br><br>单位：分钟或 TB<br><br>维度：ClusterIdentifier 、FeatureType 、UsageLimitId |

| 指标 | 说明 |
| --- | --- |
| UsageLimitConsumed | 根据 FeatureType，UsageLimitConsumed 将返回以下内容：<br>• 如果 FeatureType 为 CONCURRENCY_SCALING ，则 UsageLimitConsumed 将返回并发扩展使用的总时间长度（以 1 分钟为增量）。此时间不计为用于 CONCURRENCY_SCALING_AUTO_TASK 的时间，反之亦然。<br>• 如果 FeatureType 为 CONCURRENCY_SCALING_AUTO_TASK ，则 UsageLimitConsumed 将返回并发扩展用于自动优化任务的总时间长度。此时间不计为用于 CONCURRENCY_SCALING 的时间，反之亦然。<br>• 如果 FeatureType 为 EXTRA_COMPUTE_FOR_AUTOMATIC_OPTIMIZATION ，则 UsageLimitConsumed 将返回用于自动优化任务的总时间长度。<br>• 如果 FeatureType 为 CROSS_REGION_DATASHARING ，则 UsageLimitConsumed 将返回扫描的数据总量（以 1 TB 为增量）。<br>• 如果 FeatureType 为 SPECTRUM，则 UsageLimitConsumed 将返回扫描的数据总量（以 1 TB 为增量）。<br><br>单位：分钟或 TB<br><br>维度：ClusterIdentifier 、FeatureType 、UsageLimitId |

## Amazon Redshift 指标的维度

可以按下表中的任意维度对 Amazon Redshift 数据进行筛选。

| 维度 | 说明 |
| --- | --- |
| latency | 可能值如下所示：<br>• short – 不到 10 秒<br>• medium – 在 10 秒到 10 分钟之间 |

| 维度 | 说明 |
| --- | --- |
| | • long – 超过 10 分钟 |
| NodeID | 筛选条件请求的特定于集群节点的数据。NodeID 是“领导”、“共享”或“计算 N”，其中 N 是集群中节点的数目（0、1 等）。“共享”意味着集群只有一个节点，即领导节点和计算节点合并到了一起。<br><br>针对领导节点和计算节点报告的指标只适用于 CPUUtilization 、NetworkTransmitThroughput 和 ReadIOPS。使用 NodeId 维度的其他指标只针对计算节点进行报告。 |
| ClusterIdentifier | 筛选条件请求的特定于集群的数据。特定于集群的指标包括 HealthStatus 、MaintenanceMode 和 DatabaseConnections 。此维度的一般指标（例如，ReadIOPS）同样是表示节点指标数据聚合的节点的指标。在解析这些指标时请小心，因为它们是领导节点和计算节点的聚合行为。 |
| service class | WLM 服务类的标识符。 |
| stage | 查询的执行阶段。可能的值如下所示：<br><br>• QueryPlanning：分析和优化 SQL 语句所花的时间。<br>• QueryWaiting：在 WLM 队列中等待所花的时间。<br>• QueryExecutingRead：执行读取查询所花的时间。<br>• QueryExecutingInsert：执行插入查询所花的时间。<br>• QueryExecutingDelete：执行删除查询所花的时间。<br>• QueryExecutingUpdate：执行更新查询所花的时间。<br>• QueryExecutingCtas：执行 create table as 查询所花的时间。<br>• QueryExecutingUnload：执行卸载查询所花的时间。<br>• QueryExecutingCopy：执行复制查询所花的时间。<br>• QueryCommit：提交所花的时间。 |
| wlmid | 工作负载管理队列的标识符。 |

| 维度 | 说明 |
| --- | --- |
| QueryPriority | 查询的优先级。可能的值包括 CRITICAL、HIGHEST、HIGH、NORMAL、LOW 和 LOWEST。 |
| QueueName | 工作负载管理队列的名称。 |
| FeatureType | 受使用量限额限制的功能。可能的值为 CONCURRENCY_SCALING 、CROSS_REGION_DATASHARING 和 SPECTRUM。 |
| UsageLimitId | 使用量限额的标识符。 |

## Amazon Redshift 查询和加载性能数据

除了 CloudWatch 指标之外，Amazon Redshift 还提供查询和加载性能数据。查询和加载性能数据有助于您了解数据库性能和集群指标之间的关系。例如，如果您注意到集群的 CPU 达到峰值，则可以在集群 CPU 图表中找到相应峰值，并了解当时正在运行的查询。相反，如果您要查看特定查询，则相应指标数据（如 CPU）会根据具体情况显示出来，以便您了解查询对集群指标的影响。

查询和加载性能数据不会作为 CloudWatch 指标发布，而只能通过 Amazon Redshift 控制台查看。查询和加载性能数据通过查询数据库的系统表生成（有关更多信息，请参阅《Amazon Redshift 开发人员指南》中的系统表参考）。您也可以生成自己的自定义数据库性能查询，不过我们建议您先从控制台中提供的查询和加载性能数据入手。有关自行测量和监控数据库性能的更多信息，请参阅《Amazon Redshift 开发人员指南》中的管理性能。

下表介绍了您可以通过 Amazon Redshift 控制台访问的查询和加载数据的不同方面。

| 查询/加载数据 | 说明 |
| --- | --- |
| 查询摘要 | 指定时间段内的查询列表。此列表可以按查询 ID、查询运行时间和状态等值排序。在集群详细信息页面的查询监控选项卡中查看此数据。 |
| 查询详细信息 | 提供某个特定查询的详细信息，包括：<br>• 查询属性，如查询 ID、类型、运行查询的集群和运行时间。<br>• 详细信息，例如查询的状态和错误数量。<br>• 运行的 SQL 语句。<br>• 说明计划（如果有）。 |

| 查询/加载数据 | 说明 |
| --- | --- |
| | • 查询执行期间的集群性能数据（有关更多信息，请参阅查看查询历史记录）。 |
| 加载摘要 | 列出指定时间段内的所有加载。此列表可以按查询 ID、查询运行时间和状态等值排序。在集群详细信息页面的查询监控选项卡中查看此数据。 |
| 加载详细信息 | 提供有关特定加载操作的详细信息，包括：<br>• 加载属性，例如查询 ID、类型、运行查询的集群和运行时间。<br>• 详细信息，例如加载的状态和错误数量。<br>• 运行的 SQL 语句。<br>• 已加载的文件列表。<br>• 加载操作期间的集群性能数据（有关更多信息，请参阅查看查询历史记录）。 |

## 查看性能数据

您可以在本节中查找如何在 Amazon Redshift 控制台中查看性能数据，包括有关集群和查询性能的信息。此外，您可以直接通过 Amazon Redshift 控制台针对集群指标创建警报。

您可以在 Amazon Redshift 控制台中按集群查看性能数据。集群的性能数据图表旨在帮助您找到解答最常见性能问题的数据。对于某些性能数据（请参阅Amazon Redshift 中的性能数据），您还可以使用 CloudWatch 进一步自定义指标图表。例如，您可以选择较长的时间或跨集群组合指标。有关使用 CloudWatch 控制台的更多信息，请参阅CloudWatch 控制台中的性能指标。

观看以下视频以了解如何使用 Amazon Redshift 控制台上的查询监控功能监控、隔离和优化查询：使用 Amazon Redshift 查询监控。

主题

- 查看集群性能数据
- 查看查询历史记录
- 查看数据库性能数据
- 查看工作负载并发和并发扩展数据
- 查看自动优化数据

- 查看查询和加载
- 查看和分析查询详细信息
- 查看查询运行时的集群性能
- 在加载操作期间查看集群指标
- 查看集群工作负载细分图表

## 查看集群性能数据

通过使用 Amazon Redshift 中的集群指标，您可以执行以下常见性能任务：

- 判断集群指标在指定时间范围内是否异常；如果异常的话，则确定负责这种性能冲击的查询。
- 查看历史或当前查询是否对集群性能造成了影响。如果您识别出了一个有问题的查询，则可以在查询执行期间查看有关该查询的详细信息（包括集群性能）。您可以使用此信息来诊断为何查询速度慢以及可以采取哪些措施来提高它的性能。

### 查看性能数据

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群，然后从列表中选择集群名称以打开其详细信息。此时将显示集群的详细信息，其中包括集群性能、查询监控、数据库、数据共享、计划、维护和属性选项卡。
3. 选择集群性能选项卡以查看性能信息，其中包括以下信息：

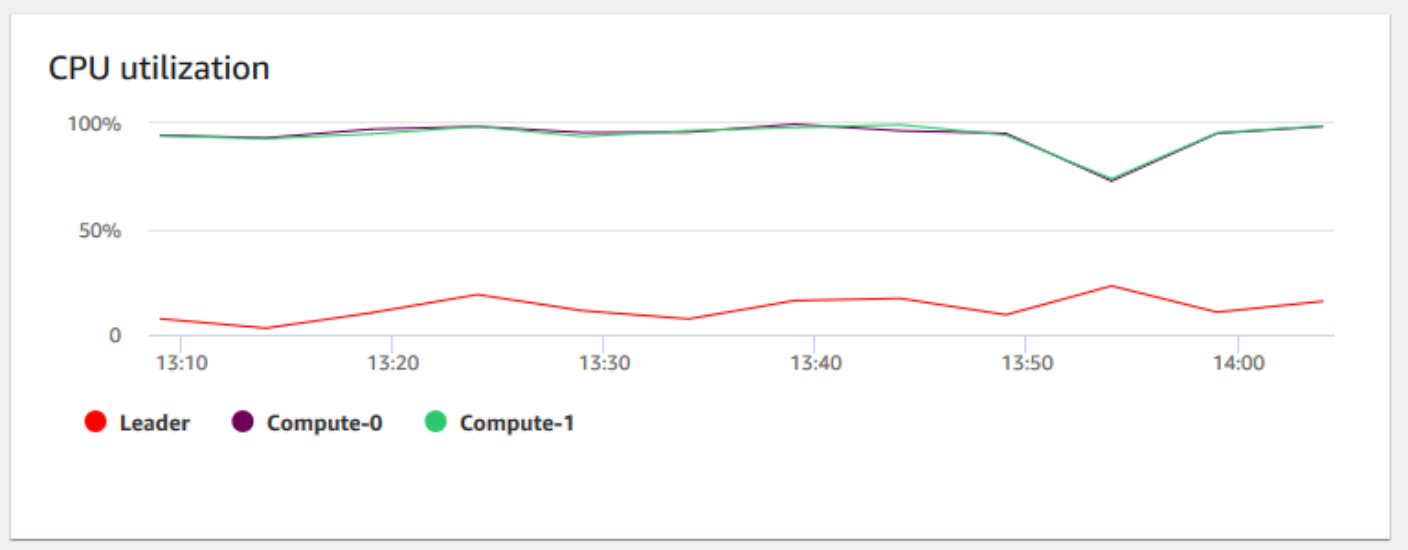
   - CPU 使用率
   - 已使用磁盘空间的百分比
   - 数据库连接
   - 运行状况
   - 查询持续时间
   - 查询吞吐量
   - 并发扩展活动

   此外，还提供了许多其他指标。要查看可用指标并选择要显示的指标，请选择首选项图标。

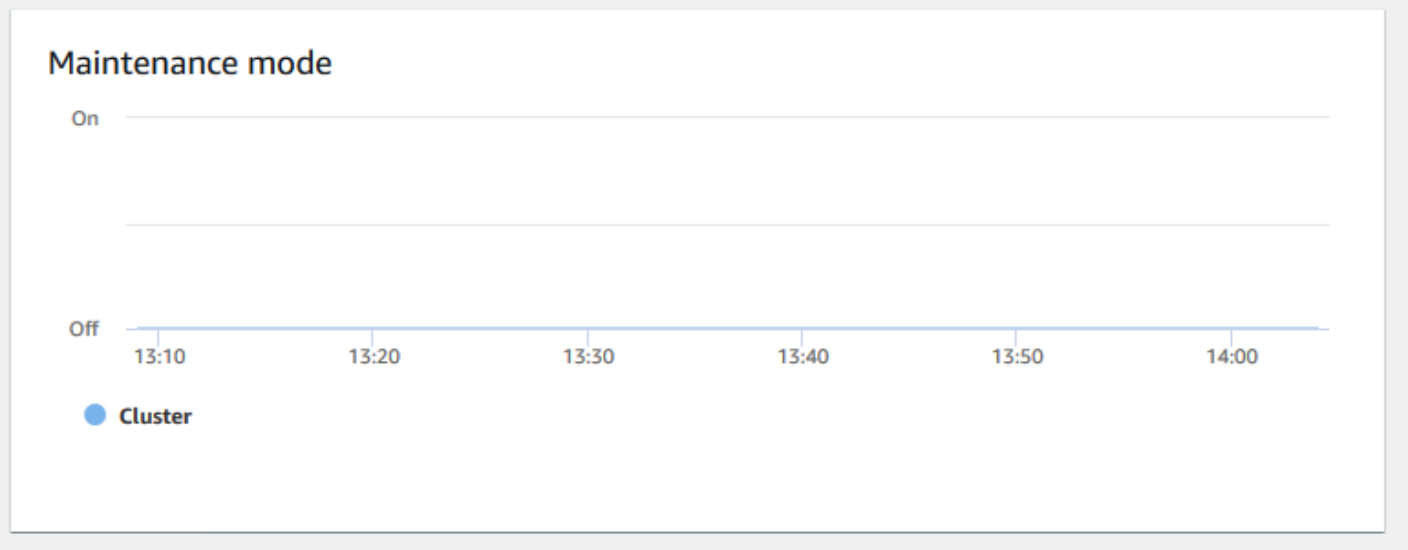
## 集群性能图表

以下示例显示新的 Amazon Redshift 控制台中显示的一些图表。

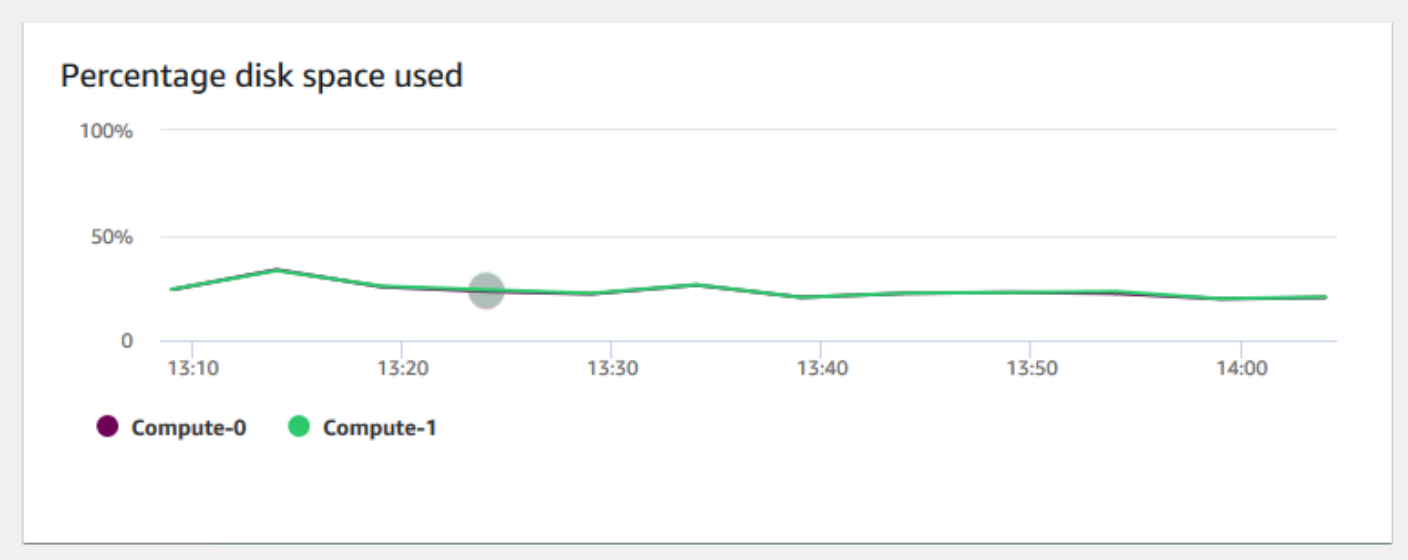
- CPU 利用率 – 显示所有节点（领导节点和计算节点）的 CPU 利用率百分比。要在计划集群迁移或其他资源消耗型操作之前查找集群使用率最低的时间，请监控此图表以查看每个节点或所有节点的 CPU 使用率。



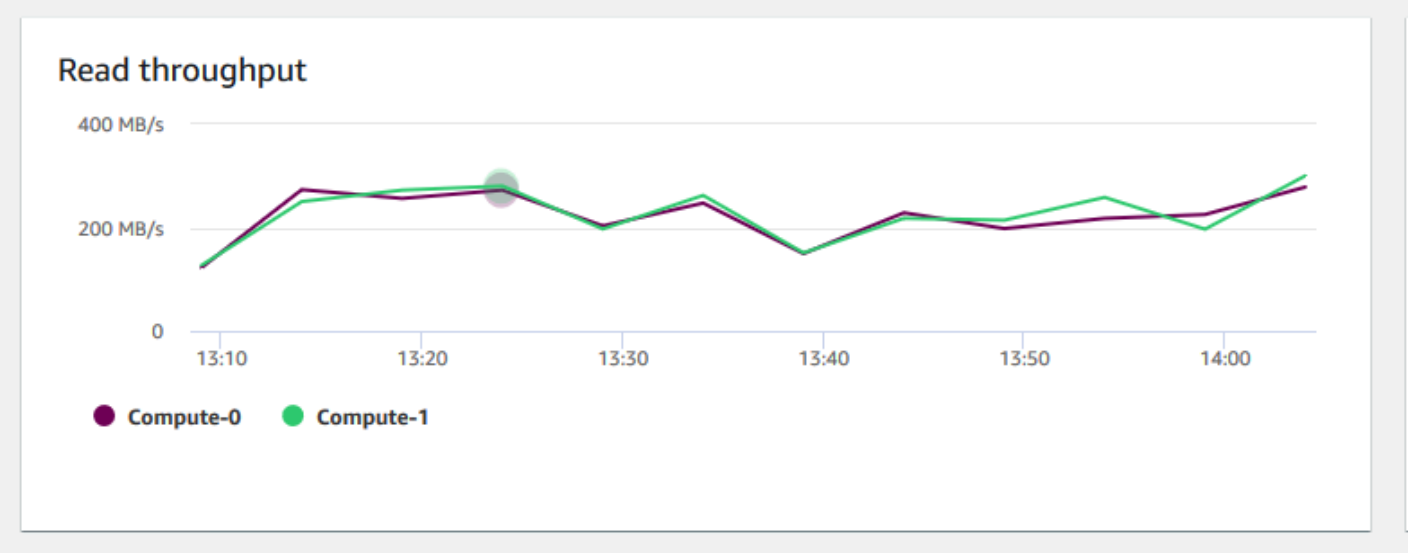

- 维护模式 – 通过使用 On 和 Off 指示灯显示集群在所选时间是否处于维护模式。您可以查看集群正在进行维护的时间。然后，您可以将此时间与对集群执行的操作相关联，以估计其将来发生重复性事件的停机时间。



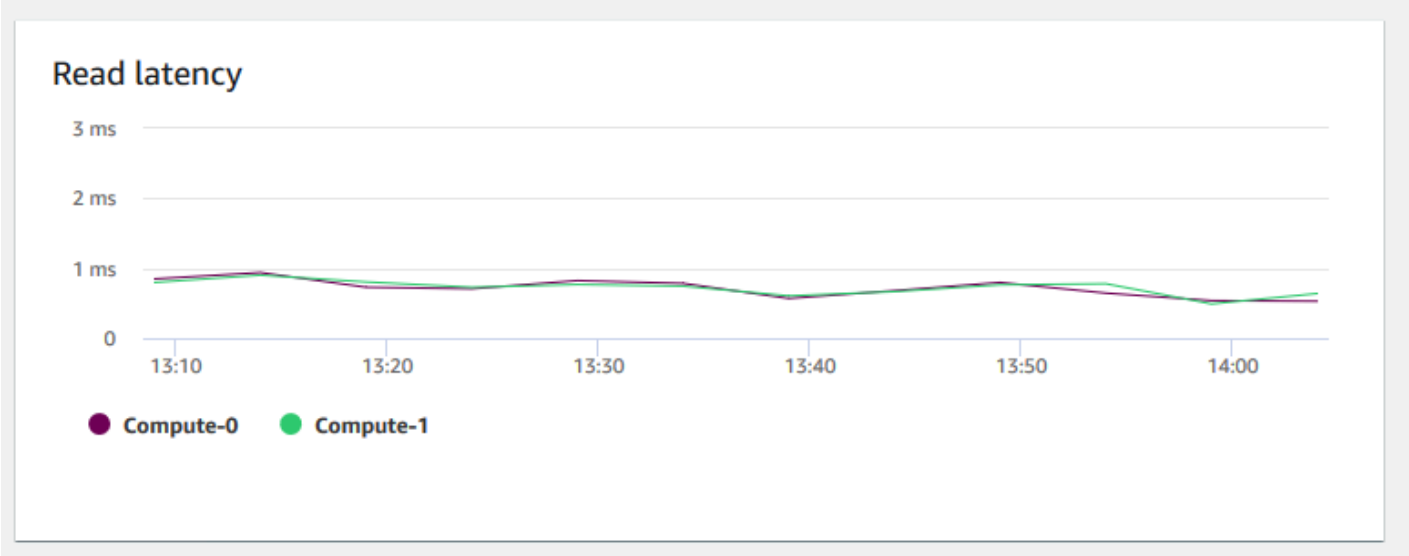

- 已使用磁盘空间的百分比 – 显示每个计算节点（而不是整个集群）的磁盘空间使用量百分比。您可以浏览此图表来监控磁盘利用率。VACUUM 和 COPY 等维护操作使用中间临时存储空间来执行排序操作，因此预计磁盘使用量会出现峰值。



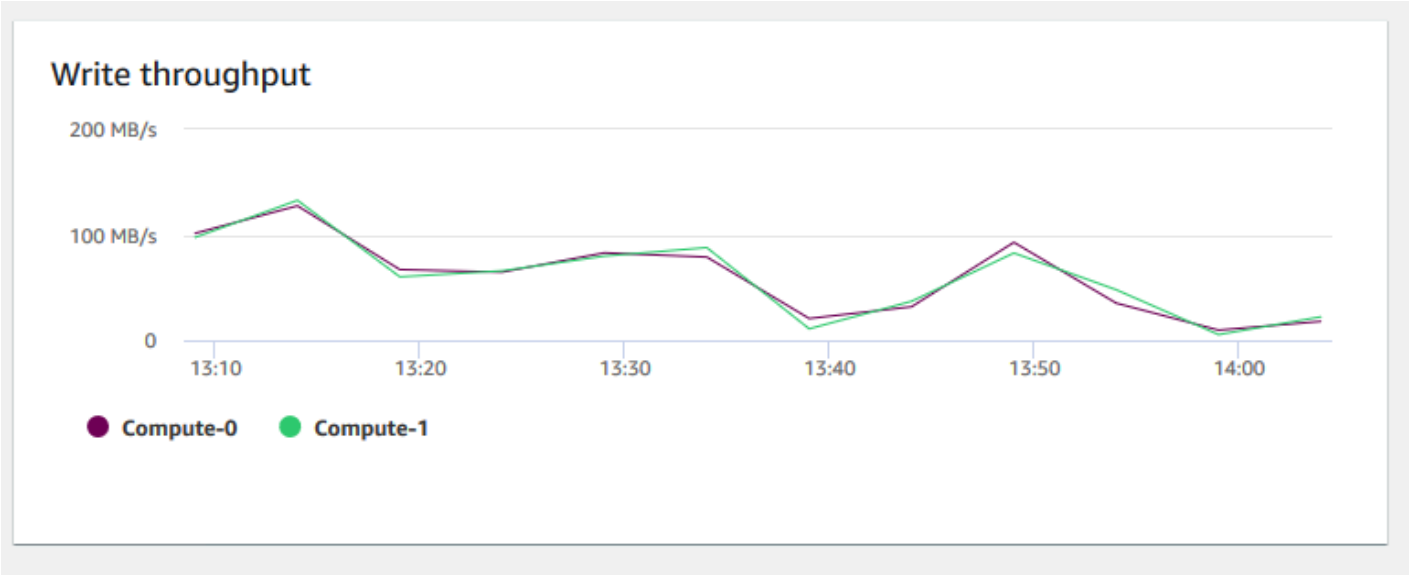

- 读取吞吐量 – 显示每秒从磁盘读取的平均兆字节数。您可以评估此图表以监控集群的相应物理方面。此吞吐量不包括集群中的实例与集群的卷之间的网络流量。



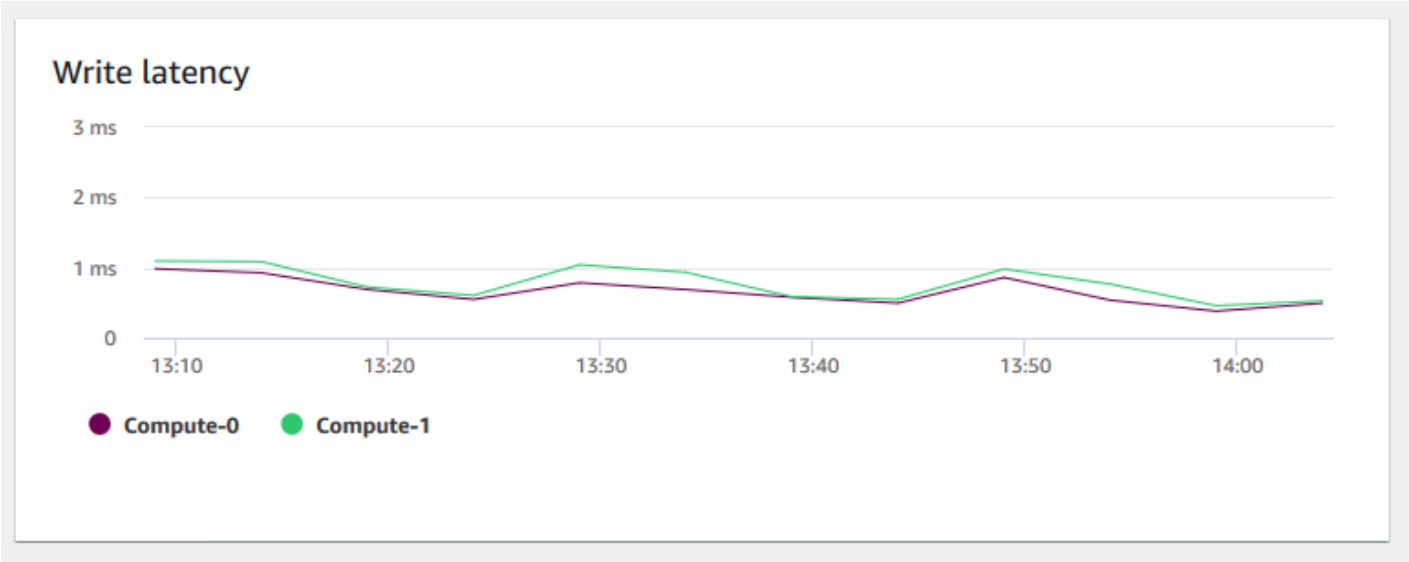

- 读取延迟 – 显示磁盘读取输入/输出操作所花费的平均时间（以毫秒为单位）。您可以查看要返回的数据的响应时间。当延迟很高时，这意味着发送方处于空闲状态的时间会更多（不发送任何新的数据包），这会降低吞吐量的增长速度。



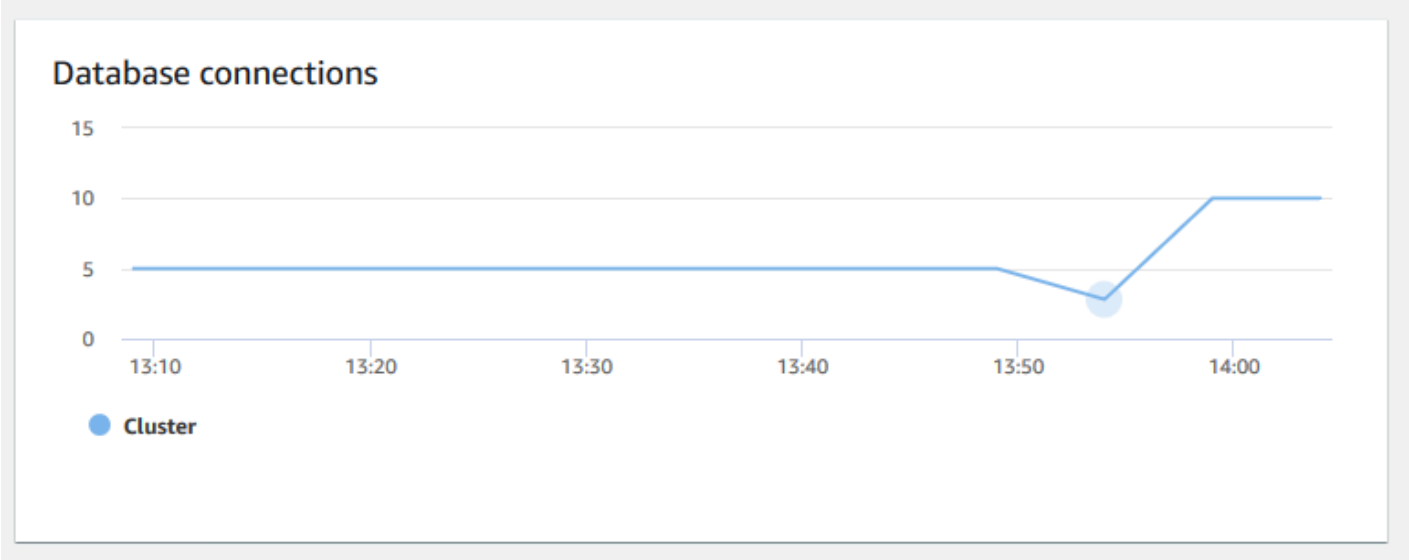

- 写入吞吐量 – 显示每秒写入磁盘的平均兆字节数。您可以评估此指标，以监控集群的相应物理方面。此吞吐量不包括集群中的实例与集群的卷之间的网络流量。



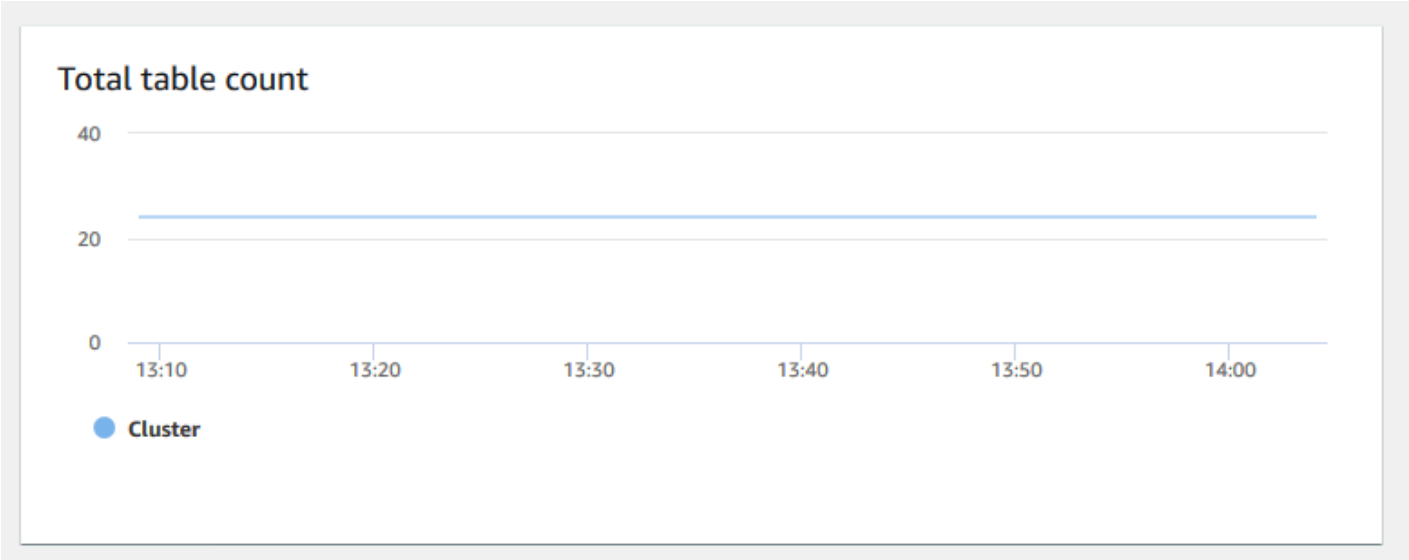

- 写入延迟 – 显示磁盘写入输入/输出操作所花费的平均时间（以毫秒为单位）。您可以评估返回写确认的时间。当延迟很高时，这意味着发送方处于空闲状态的时间会更多（不发送任何新的数据包），这会降低吞吐量的增长速度。



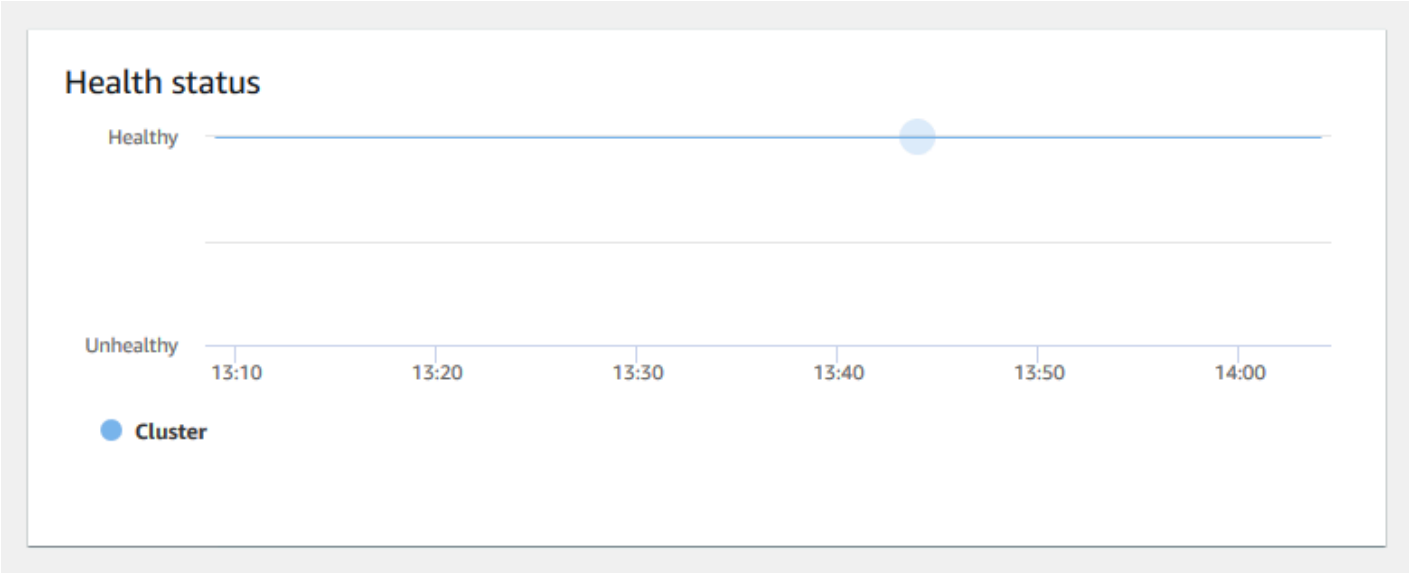

- 数据库连接数 – 显示到集群的数据库连接数。您可以使用此图表查看与数据库建立的连接数，并查找集群使用率最低的时间。



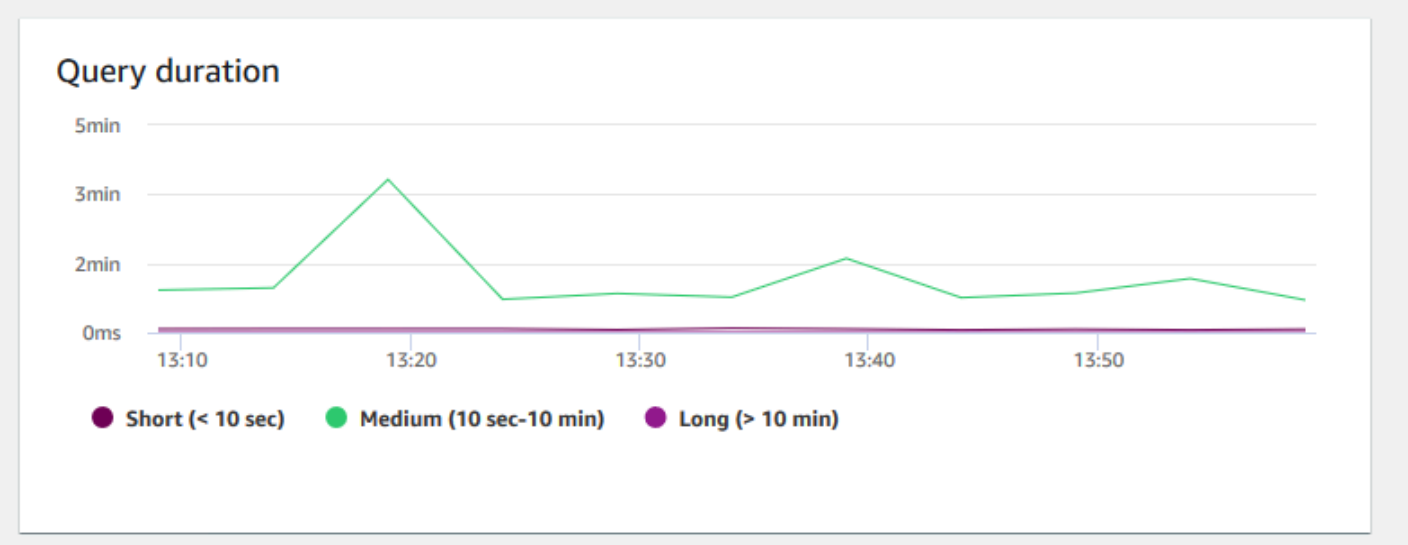

- 总的表计数 – 显示集群内在某个特定时间点打开的用户表的数量。您可以在打开的表计数较高时监控集群性能。



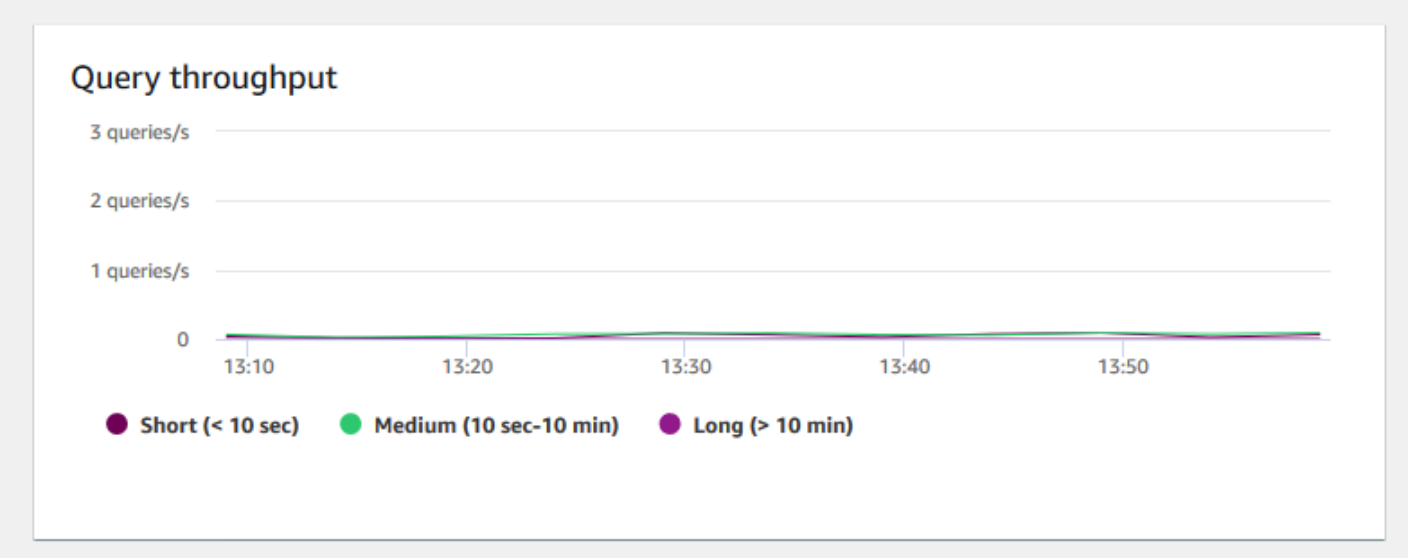

- 运行状况 – 将集群的运行状况指示为 Healthy 或 Unhealthy。如果集群可以连接到其数据库并成功执行简单查询，则集群将被视为运行状况良好。否则，视为集群运行状况不佳。当数据库集群负载极重，或者集群上的数据库存在配置问题时，集群会出现运营状况不佳的情况。



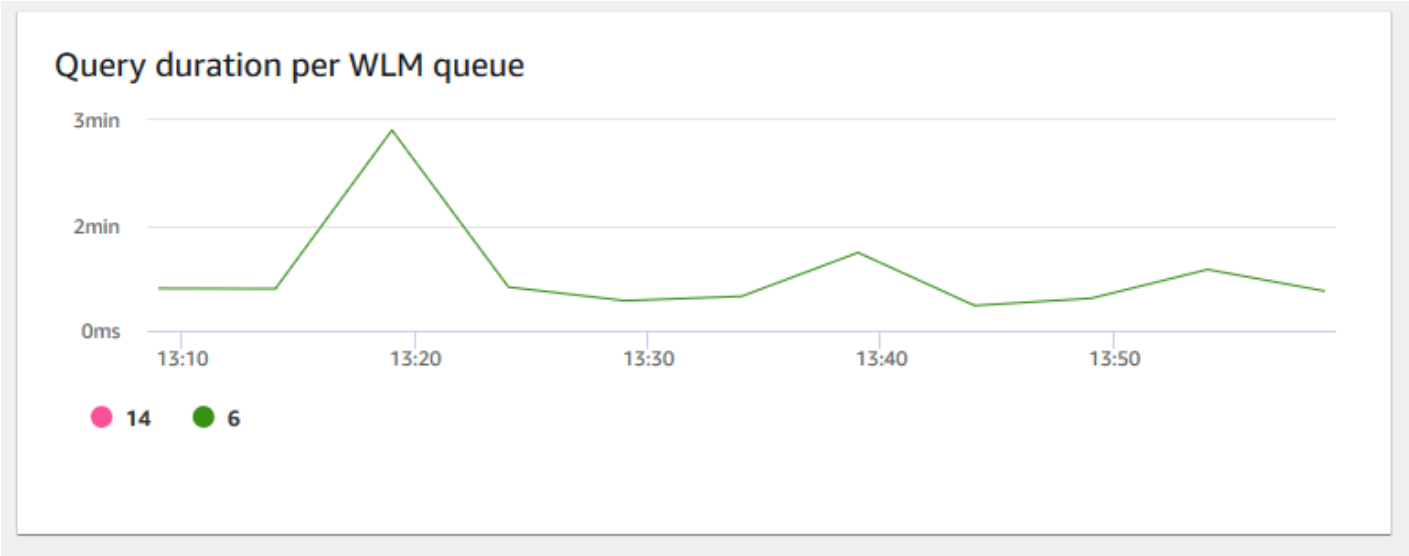

- 查询持续时间 – 显示完成查询的平均时间量（以微秒为单位）。您可以将此图表上的数据作为基准以衡量集群内的 I/O 性能，并在必要时调整其最耗时的查询。



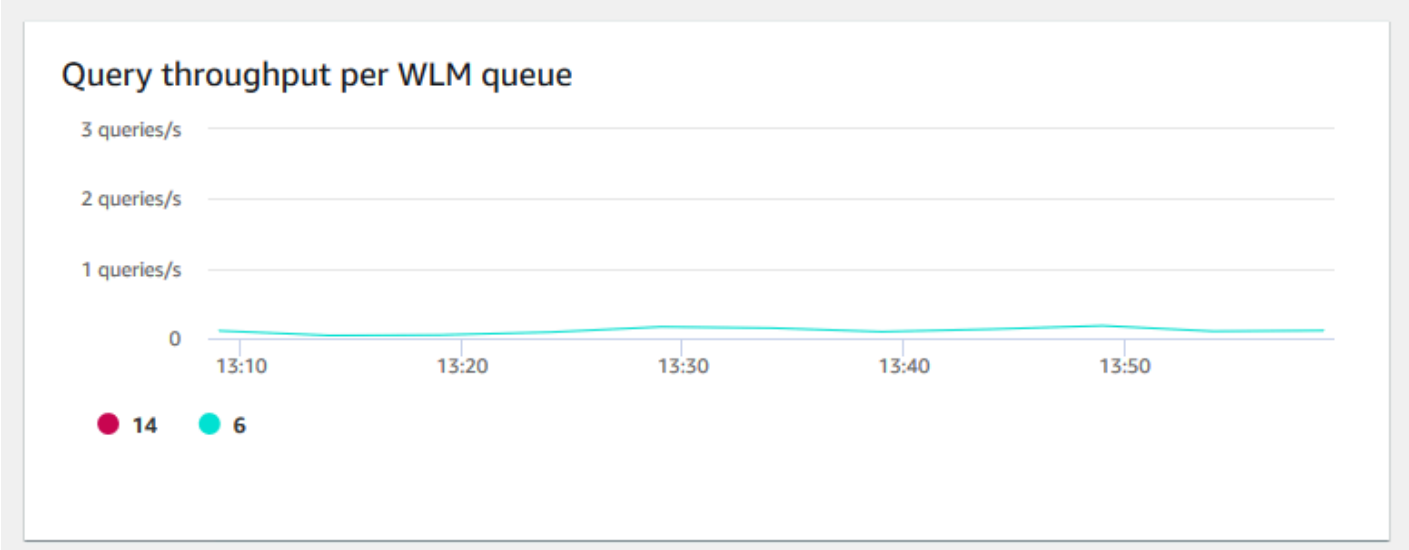

- 查询吞吐量 – 显示每秒完成的查询的平均数。您可以分析此图表上的数据以衡量数据库性能，并表明系统以均衡的方式支持多用户工作负载的能力。



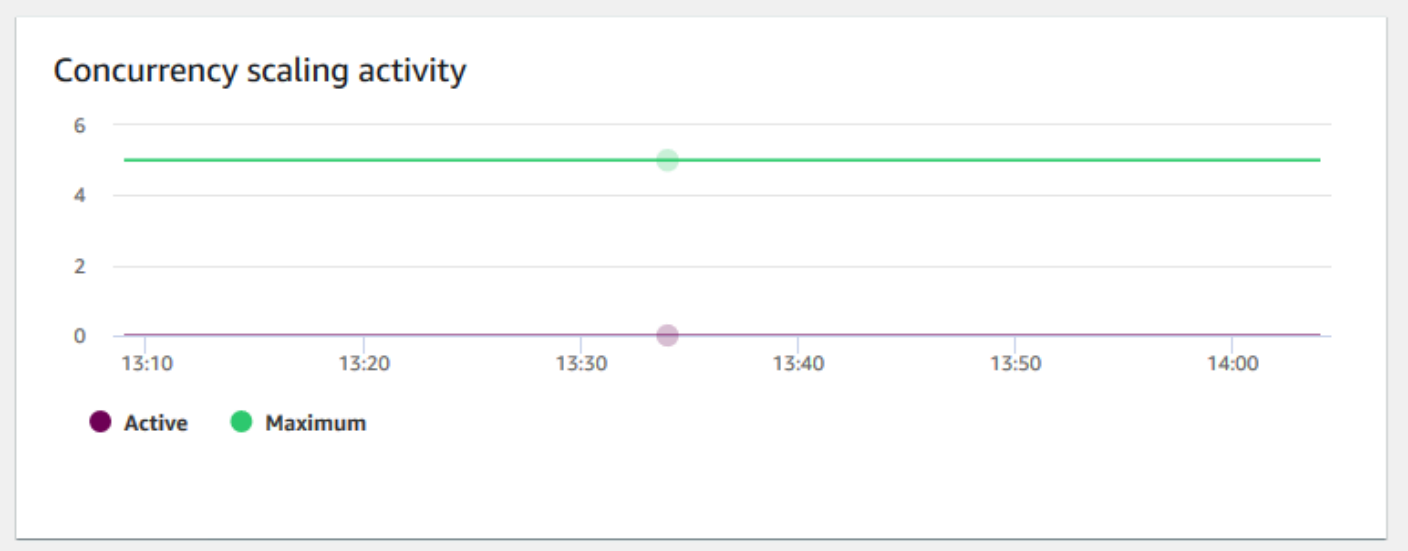

- 每个 WLM 队列的查询持续时间 – 显示完成查询的平均时间量（以微秒为单位）。您可以将此图表上的数据作为基准测试，以衡量每个 WLM 队列的 I/O 性能，并在必要时调整其最耗时的查询。




- 每个 WLM 队列的查询吞吐量 – 显示每秒完成的查询的平均数。您可以分析此图表上的数据，以衡量每个 WLM 队列的数据库性能。




- 并发扩缩活动 – 显示活动的并发扩展集群的数量。启用并发扩展后，Amazon Redshift 会在需要时自动增加额外的集群容量来处理增多的并发读取查询。




## 查看查询历史记录

您可以使用 Amazon Redshift 中的查询历史记录指标执行以下操作：

- 隔离和诊断查询性能问题。
- 比较同一时间线上的查询运行时间指标和集群性能指标，以查看这两者之间可能相关的程度。这样做有助于识别性能不佳的查询，寻找瓶颈查询和确定您是否需要为您的工作负载调整集群大小。
- 通过在时间线中选择特定查询，向下钻取到该查询的详细信息。当查询 ID 和其他属性显示在此图表下方的行中时，您可以选择查询以查看查询详细信息。详细信息包括如查询的 SQL 语句、执行详细信息和查询计划等。有关更多信息，请参阅 查看和分析查询详细信息。
- 确定加载作业是否成功完成并满足服务等级协议 (SLA)。

### 显示查询历史记录数据

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群，然后从列表中选择集群名称以打开其详细信息。此时将显示集群的详细信息，其中可能包括集群性能、查询监控、数据库、数据共享、计划、维护和属性选项卡。
3. 为查询相关指标选择查询监控选项卡。
4. 在查询监控部分中，选择查询历史记录选项卡。

   使用窗口上的控件，您可以在查询列表和集群指标之间切换。

选择查询列表时，该选项卡包括以下图表：

- 查询运行时间 – 时间线上的查询活动。使用此图表可查看哪些查询在同一时间范围内运行。选择查询以查看更多查询执行详细信息。x 轴显示选定的期间。您可以通过正在运行、已完成、加载等筛选图形化的查询。每个条形表示一个查询，条形的长度表示其运行时间（从条形开始到结束）。查询可以包括 SQL 数据操作语句（如 SELECT、INSERT、DELETE）和加载（如 COPY）。原定设置情况下，显示所选时间段内运行时间最长的前 100 个查询。
- 查询和加载 – 集群上运行的查询和加载列表。此窗口包含一个选项，用于在查询当前正在运行时终止查询。

选择集群指标时，该选项卡包括以下图表：

- 查询运行时间 – 时间线上的查询活动。使用此图表可查看哪些查询在同一时间范围内运行。选择查询以查看更多查询执行详细信息。
- CPU 利用率 – 按领导节点以及计算节点平均值分列的集群 CPU 利用率。
- 已使用的存储容量 – 已使用的存储容量百分比。
- 活动的数据库连接数 – 显示到集群的活动的数据库连接数。

处理查询历史记录图表时，请考虑以下事项：

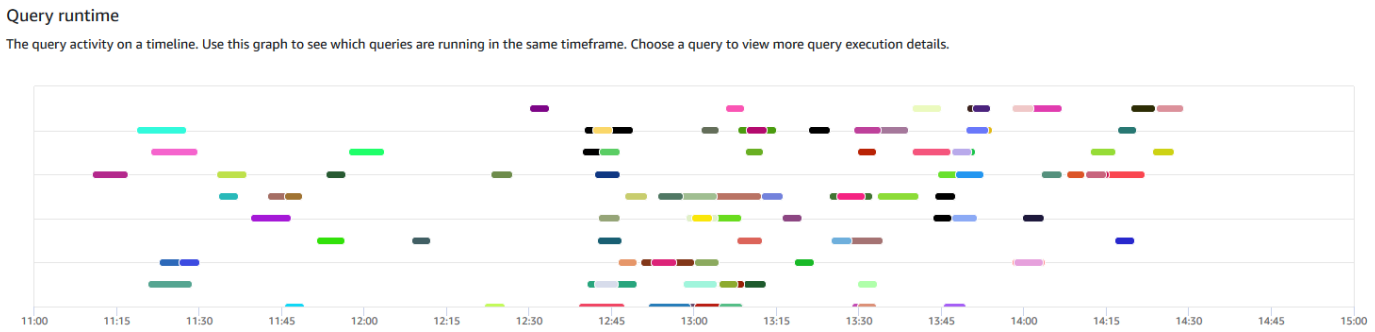
- 选择一个在查询运行时间图表上表示某特定查询的条形，以查看有关该查询的详细信息。也可以在查询和加载列表中选择查询 ID 以查看其详细信息。
- 您可以轻扫以选择查询运行时间图表的某个部分进行放大以显示特定时间段。
- 在查询运行时间图表上，要使所选筛选条件考虑所有数据，请向前翻动查询和加载列表中列出的所有页面。
- 您可以使用通过设置齿轮图标显示的首选项窗口更改显示在查询和加载列表上的列和行数。
- 查询和加载列表也可以通过从左侧导航器导航查询图标、查询和加载来显示。有关更多信息，请参阅查看查询和加载。

查询历史记录图表

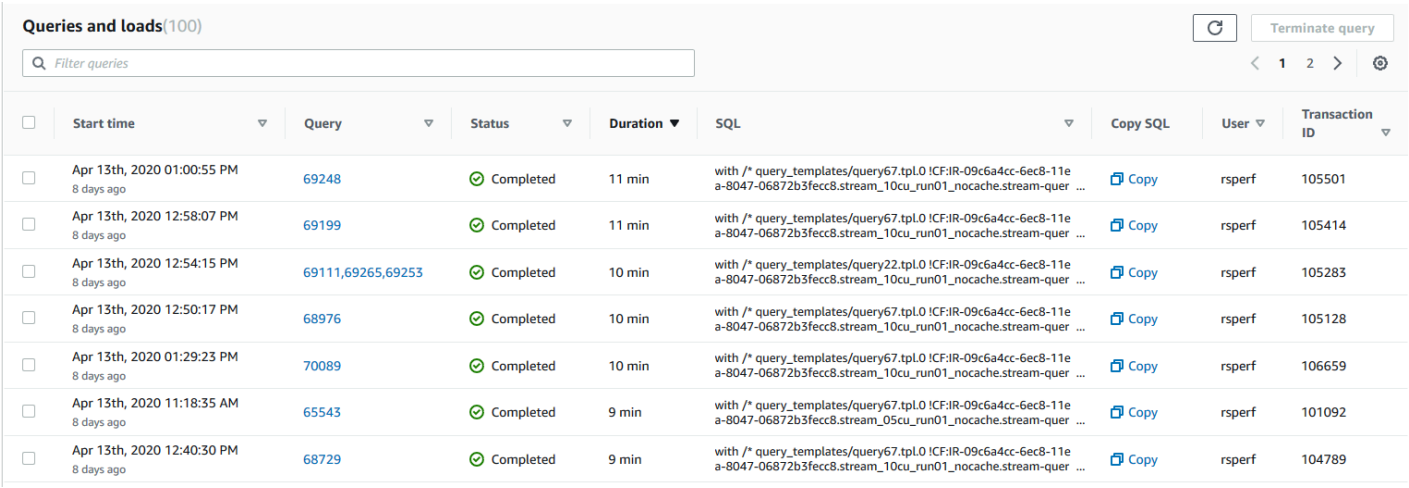
以下示例显示新的 Amazon Redshift 控制台中显示的图表。

> **Note**
>
> Amazon Redshift 控制台图表仅包含最新 100000 个查询的数据。

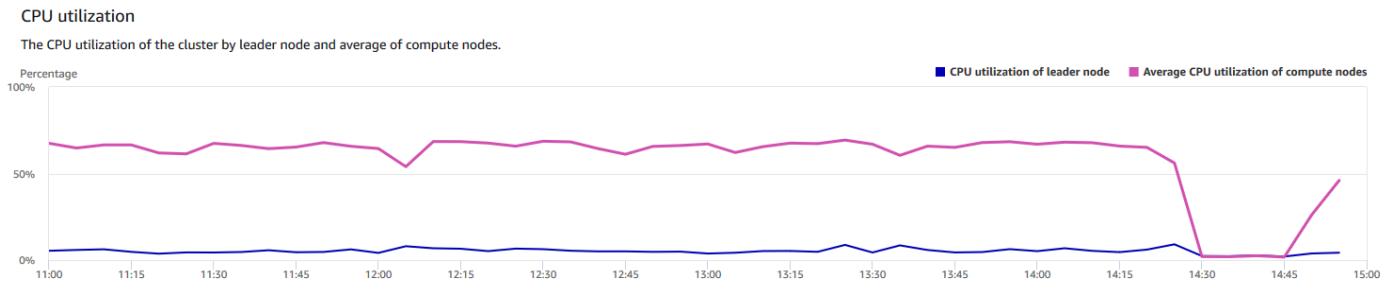
- 查询运行时间




- 查询和加载



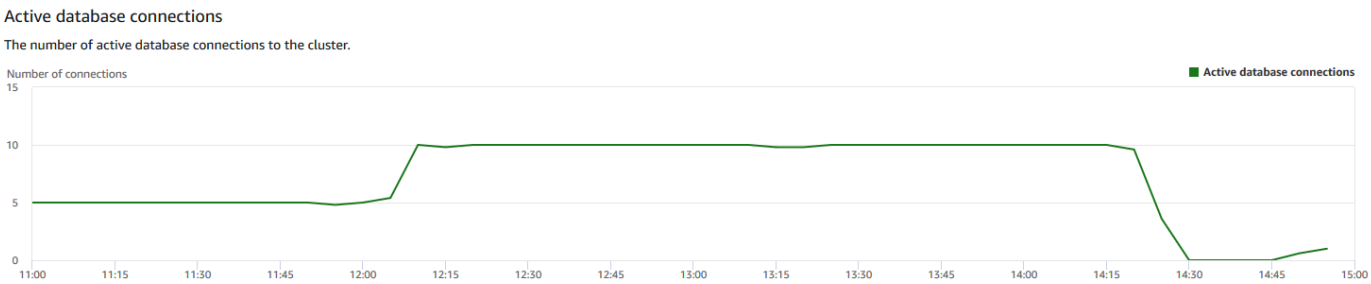

- CPU 使用率




- 已使用的存储容量




- 活动的数据库连接数




## 查看数据库性能数据

您可以使用 Amazon Redshift 中的数据库性能指标执行以下操作：

- 按处理阶段分析查询所花费的时间。您可以在一个阶段花费的时间量中寻找不寻常的趋势。
- 按持续时间范围（短、中、长）分析查询的数量、查询的持续时间和吞吐量。
- 按查询优先级（“最低”、“低”、“正常”、“高”、“最高”、“临界”）查找查询等待时间的趋势。
- 按 WLM 队列查找查询持续时间、吞吐量或等待时间的趋势。

### 显示数据库性能数据

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群，然后从列表中选择集群名称以打开其详细信息。此时将显示集群的详细信息，包括集群性能、查询监控、数据库、数据共享、计划、维护和属性选项卡。
3. 为查询相关指标选择查询监控选项卡。
4. 在查询监控部分中，选择数据库性能选项卡。

   使用窗口上的控件，您可以在集群指标和 WLM 队列指标之间切换。

选择集群指标时，该选项卡包括以下图表：

- 工作负载执行细分 – 查询处理阶段使用的时间。
- 按持续时间范围列出的查询 – 短、中和长查询的数量。
- 查询吞吐量 – 每秒完成的平均查询数。
- 查询持续时间 – 完成查询的平均时间量。
- 按优先级排列的平均队列等待时间 – 按查询优先级排列的查询在 WLM 队列中等待所花费的总时间。

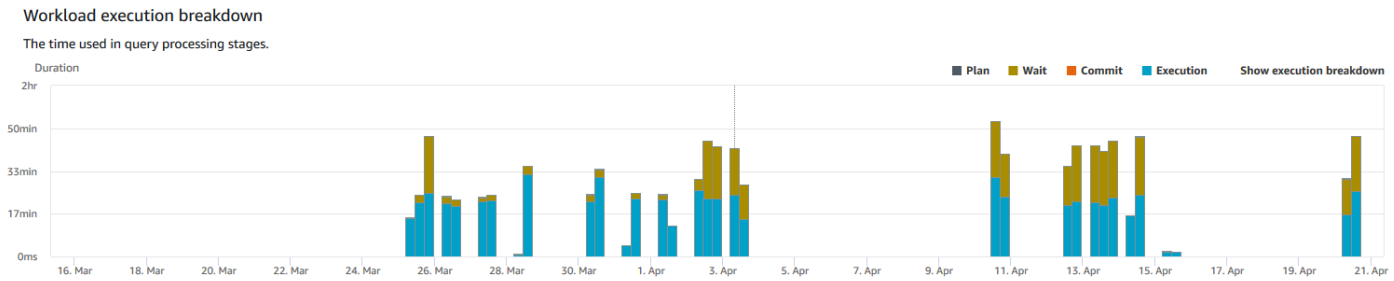
选择 WLM 队列指标时，该选项卡包括以下图表：

- 按队列排列的查询持续时间 – 按 WLM 队列排列的平均查询持续时间。
- 按队列排列的查询吞吐量 – WLM 队列每秒完成的平均查询数。
- 按队列排列的查询等待时间 – WLM 队列等待查询所花费的平均持续时间。

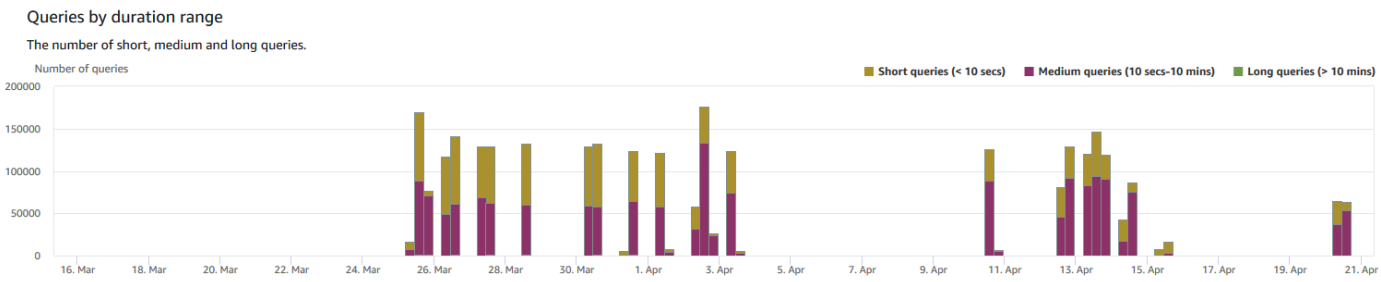
## 数据库性能图表

以下示例显示新的 Amazon Redshift 控制台中显示的图表。

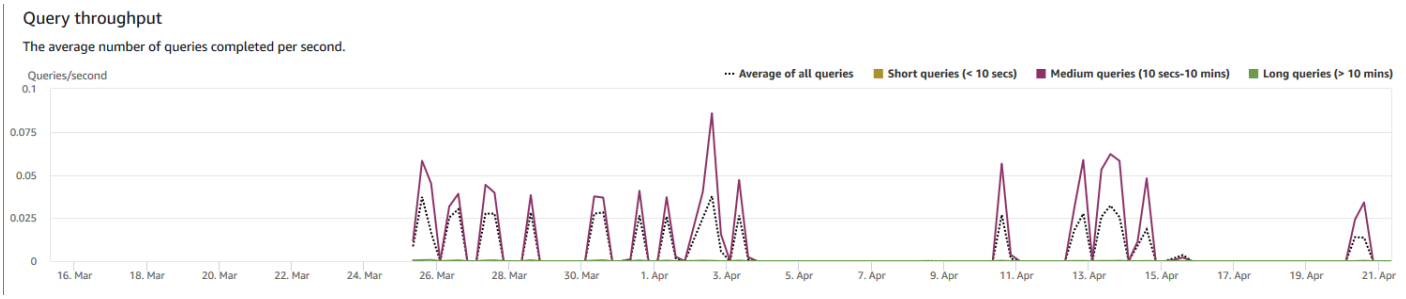
- 工作负载执行细分



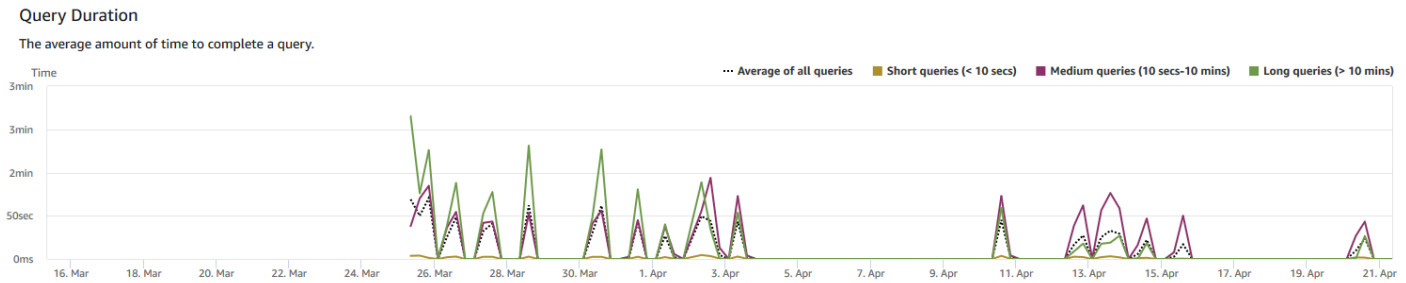

- 按持续时间范围列出的查询



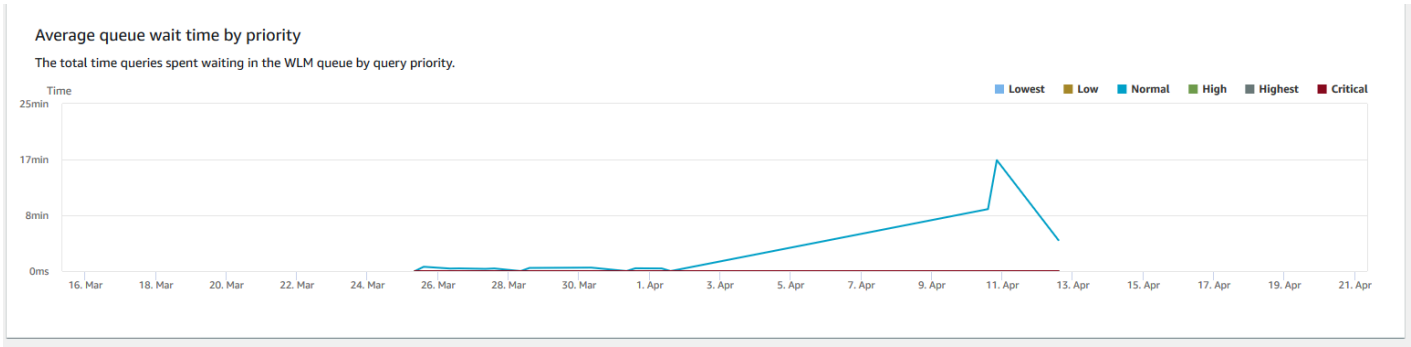
- 查询吞吐量



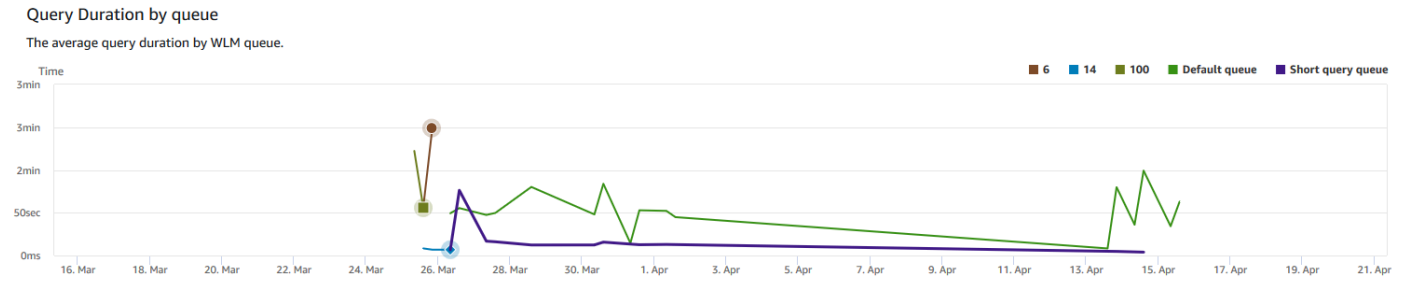

- 查询持续时间



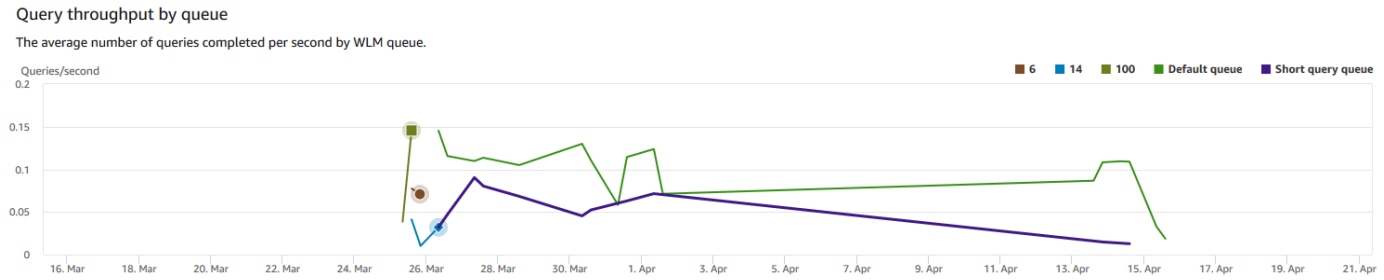

- 按优先级排列的平均队列等待时间



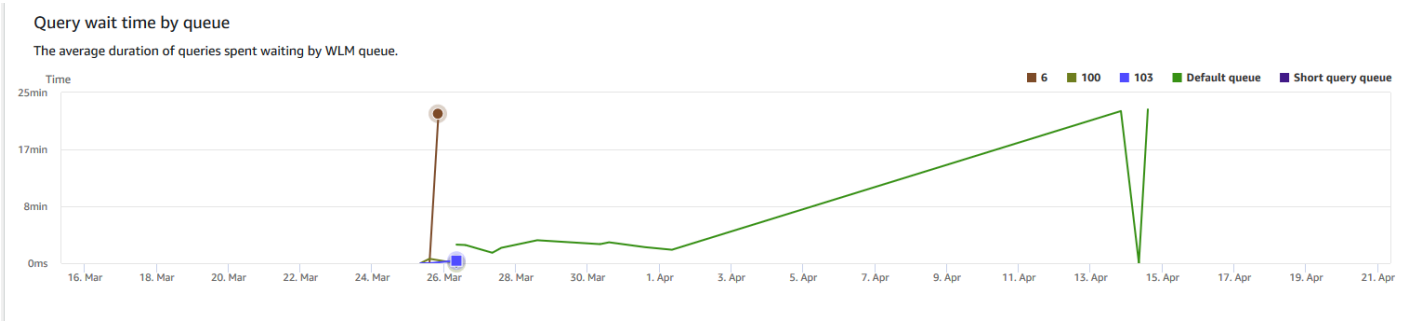

- 按队列排列的查询持续时间




- 按队列排列的查询吞吐量




- 按队列排列的查询等待时间




## 查看工作负载并发和并发扩展数据

通过在 Amazon Redshift 中使用并发扩展指标，您可以执行以下操作：

- 分析是否可以通过启用并发扩展来减少排队查询的数量。您可以按 WLM 队列或针对所有 WLM 队列进行比较。
- 查看并发扩展集群中的并发扩展活动。这可以告诉您并发扩展是否受 `max_concurrency_scaling_clusters` 限制。如果是，您可以选择增加数据库参数中的 `max_concurrency_scaling_clusters`。
- 查看所有并发扩展集群总计并发扩展的总使用率。

### 显示并发扩展数据

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群，然后从列表中选择集群名称以打开其详细信息。此时将显示集群的详细信息，其中可能包括集群性能、查询监控、数据库、数据共享、计划、维护和属性选项卡。
3. 为查询相关指标选择查询监控选项卡。
4. 在查询监控部分中，选择工作负载并发。

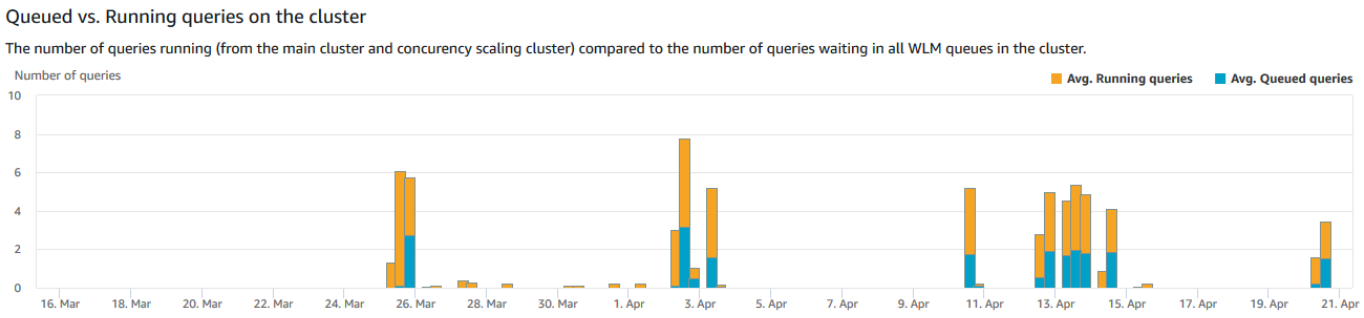
此选项卡包括以下图表：

- 集群上排队的查询数与正在运行的查询数 – 与集群中所有 WLM 队列中等待的查询数量相比，正在运行的查询数量（来自主集群和并发扩展集群）。
- 每个队列排队的查询数与正在运行的查询数 – 与每个 WLM 队列中等待的查询数量相比，正在运行的查询数量（来自主集群和并发扩展集群）。
- 并发扩缩活动 – 正在积极处理查询的并发扩展集群的数量。
- 并发扩展使用情况 – 具有活动的查询处理活动的并发扩展集群的使用情况。

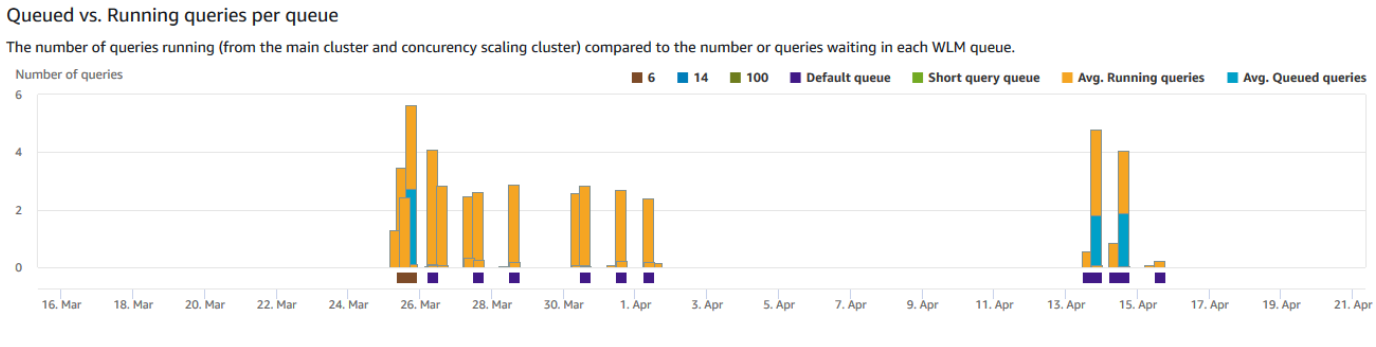
工作负载并发图表

以下示例显示新的 Amazon Redshift 控制台中显示的图表。要在 Amazon CloudWatch 中创建类似的图表，您可以使用并发扩展和 WLM CloudWatch 指标。有关 Amazon Redshift 的 CloudWatch 指标的更多信息，请参阅Amazon Redshift 中的性能数据。

- 集群上排队的查询数与正在运行的查询数



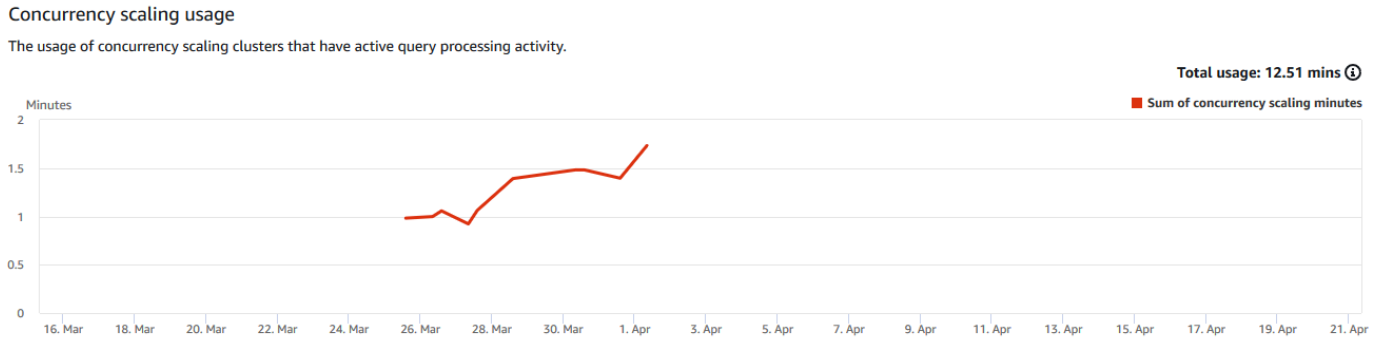

- 每个队列排队的查询数与正在运行的查询数




- 并发扩展活动




- 并发扩展使用




## 查看自动优化数据

Amazon Redshift 控制台提供有关使用额外计算资源运行的自动优化或自主处理的信息。您可以使用此信息来跟踪用量并监控是否已达到用量限制。尽管 Amazon Redshift 不会向您收取在预配集群本身上运行的自动处理的费用，但它确实会向您收取使用额外计算资源运行的自主处理的费用。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的分配额外的计算资源来执行自动数据库优化。

查看额外的计算自主处理数据

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群，然后从列表中选择集群名称以打开其详细信息。
3. 从集群的详细信息页面中，从操作下拉菜单中选择管理用量限制。也可以为集群选择维护选项卡，然后向下滚动并选择创建用量限制。
4. 显示额外计算自主处理数据的图表出现在标题为用于自动优化的额外计算的用量限制的部分下。该图显示了 Amazon Redshift 在给定时间段内使用额外计算资源运行自动处理的时间量。

## 查看查询和加载

Amazon Redshift 控制台提供有关在数据库中运行的查询和加载的信息。您可以使用这些信息确定需要很长时间才能处理的查询以及制造瓶颈来阻止其他查询获得高效处理的查询，并进行问题排查。您可以在 Amazon Redshift 控制台中使用查询信息监控查询处理。

显示查询性能数据

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择查询和加载以便显示您的账户的查询列表。

   原定设置情况下，该列表显示过去 24 小时中所有集群的查询。您可以在控制台中更改显示日期的范围。

   > **Important**
   >
   > 查询和加载选项卡显示系统中运行时间最长的查询，最多显示 100 个查询。

## 查看和分析查询详细信息

使用查询标识符，您可以查看查询的详细信息。详细信息可以包括如查询的完成状态、持续时间、SQL 语句以及查询是用户查询还是由 Amazon Redshift 重写的查询。用户查询是从 SQL 客户端提交到 Amazon Redshift 或由业务智能工具生成的查询。Amazon Redshift 可能会重写查询来优化查询，这可能会导致产生多个重写的查询。尽管该过程是由 Amazon Redshift 完成的，但您可以在查询详细信息页面上看到重写的查询以及用户查询。

查看查询

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择查询和加载以便显示您的账户的查询列表。您可能需要在该页面上更改设置才能找到您的查询。
3. 在列表中选择查询标识符以便显示查询详细信息。

   查询详细信息页面包含查询详细信息和查询计划选项卡以及查询相关指标。

指标包括有关查询的详细信息，如开始时间、查询 ID、状态和持续时间。其他详细信息包括查询是在主集群上运行还是在并发扩展集群上运行，以及查询是父查询还是重写的查询。

## 查看查询运行时的集群性能

可以在查询运行时监控集群的性能，以识别潜在的瓶颈并优化查询执行。在查询运行时查看集群性能将提供系统级指标的实时视图，例如 CPU 利用率、磁盘 I/O 和网络流量，以及查询级别的详细信息，如执行时间、处理的数据和查询步骤。以下过程将指导您访问和解释性能指标，来有效地管理和优化预置集群。

要显示查询运行时的集群性能

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群，然后从列表中选择集群名称以打开其详细信息。此时将显示集群的详细信息，其中可能包括集群性能、查询监控、数据库、数据共享、计划、维护和属性选项卡。
3. 选择查询监控选项卡以获取更多详细信息。

   有关更多信息，请参阅 查看查询历史记录。

## 在加载操作期间查看集群指标

在查看加载操作期间的集群性能时，您可以确定消耗资源的查询并采取适当的缓解措施。如果您不希望加载运行完成，则可以将其终止。

> **Note**
>
> 在 Amazon Redshift 控制台中终止查询和加载需要特定权限。如果您希望用户能够停止查询和加载，请确保将 `redshift:CancelQuerySession` 操作添加到您的 AWS Identity and Access Management (IAM) 策略。无论您在 IAM 中选择 Amazon Redshift 只读 AWS 托管策略还是创建自定义策略，此要求都适用。拥有 Amazon Redshift 完全访问策略的用户已具备终止查询和加载所需的权限。有关用于 Amazon Redshift 的 IAM 策略中的操作的更多信息，请参阅管理对资源的访问。

显示加载操作期间的集群性能

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群，然后从列表中选择集群名称以打开其详细信息。此时将显示集群的详细信息，其中可能包括集群性能、查询监控、数据库、数据共享、计划、维护和属性选项卡。
3. 选择查询监控选项卡以获取更多详细信息。
4. 在查询和加载部分中，选择加载以便查看集群的加载操作。如果加载正在运行，通过选择终止查询可以终止加载。

## 查看集群工作负载细分图表

您可以通过查看控制台中的工作负载执行细分图表，详细查看工作负载的性能。我们用 QueryRuntimeBreakdown 指标提供的数据生成图表。使用此图表，您可以看到查询在各个处理阶段（如等待和规划）中花费的时间。

> **Note**
>
> 未显示单节点集群的工作负载执行细分图表。

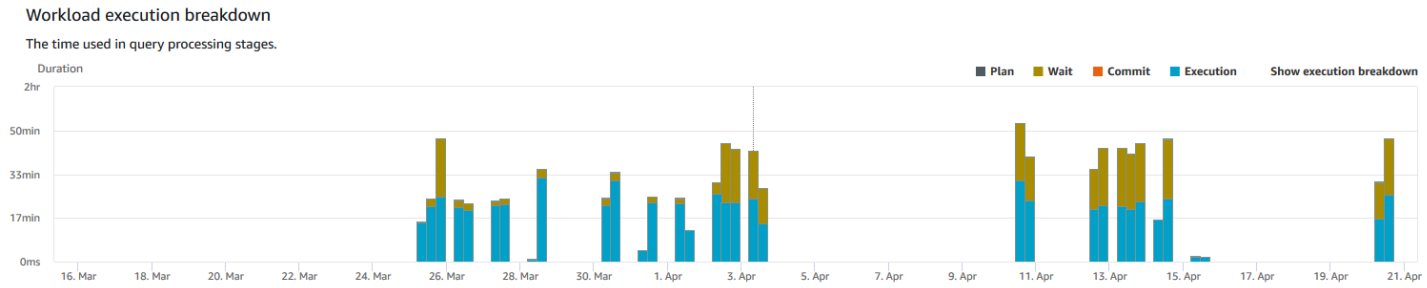
下面的指标列表描述了各个处理阶段：

- `QueryPlanning`：分析和优化 SQL 语句所花的时间。
- `QueryWaiting`：在工作负载管理 (WLM) 队列中等待的时间。
- `QueryExecutingRead`：运行读取查询所花的时间。
- `QueryExecutingInsert`：运行插入查询所花的时间。
- `QueryExecutingDelete`：运行删除查询所花的时间。
- `QueryExecutingUpdate`：运行更新查询所花的时间。
- `QueryExecutingCtas`：运行 CREATE TABLE AS 查询所花的时间。
- `QueryExecutingUnload`：运行卸载查询所花的时间。
- `QueryExecutingCopy`：运行复制查询所花的时间。

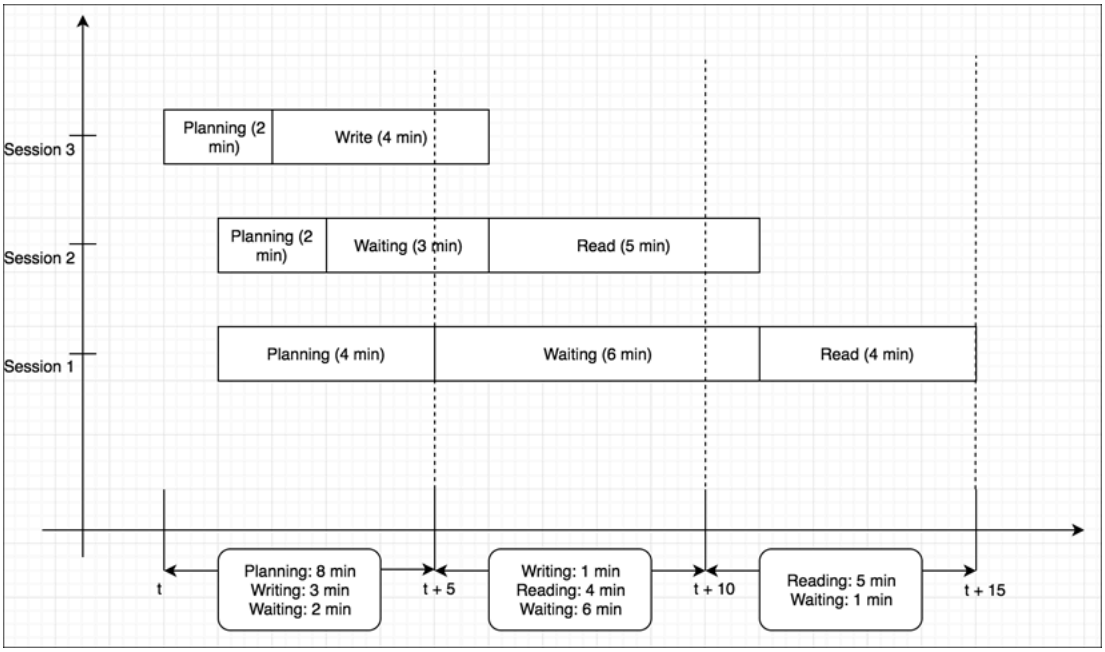
例如，Amazon Redshift 控制台中的以下图表显示了查询在计划、等待、读取和写入阶段花费的时间量。您可以将此图表中的结果与其他指标组合以进行进一步分析。在某些情况下，您的图表可能显示

具有较短持续时间（由 QueryDuration 指标度量）的查询在等待阶段花费了较长时间。在这些情况下，您可以增加特定队列的 WLM 并发速率以提高吞吐量。

以下是工作负载执行细分图的示例。在该图中，y 轴值是指定时间内每个阶段的平均持续时间，显示为堆叠条形图。




下图说明了 Amazon Redshift 如何聚合并发会话的查询处理。




## 查看集群工作负载细分图表

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群，然后从列表中选择集群名称以打开其详细信息。此时将显示集群的详细信息，其中可能包括集群性能、查询监控、数据库、数据共享、计划、维护和属性选项卡。
3. 为查询相关指标选择查询监控选项卡。
4. 在查询监控部分中，选择数据库性能和集群指标。

以堆积条形图的形式绘制以下指标在所选时间范围内的图形：

- 计划时间
- 等待时间
- 提交时间
- 执行时间

## 分析查询执行

可以分析查询的执行详细信息，来了解其执行情况并确定潜在的优化区域。通过分析查询，可以深入了解查询计划，包括所涉及的步骤、每个步骤所花费的时间以及处理的数据量。常见用例包括对运行缓慢的查询进行故障排除、优化数据分配策略以及确定查询重写或编制索引的机会。

分析查询

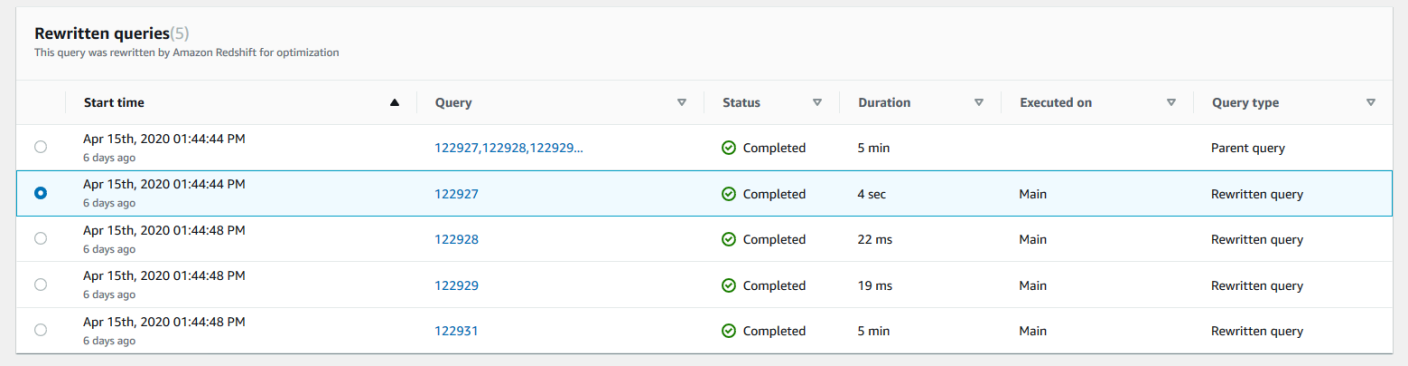
1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择查询和加载以便显示您的账户的查询列表。您可能需要在该页面上更改设置才能找到您的查询。
3. 在列表中选择查询标识符以便显示查询详细信息。

   查询详细信息页面包含查询详细信息和查询计划选项卡以及查询相关指标。

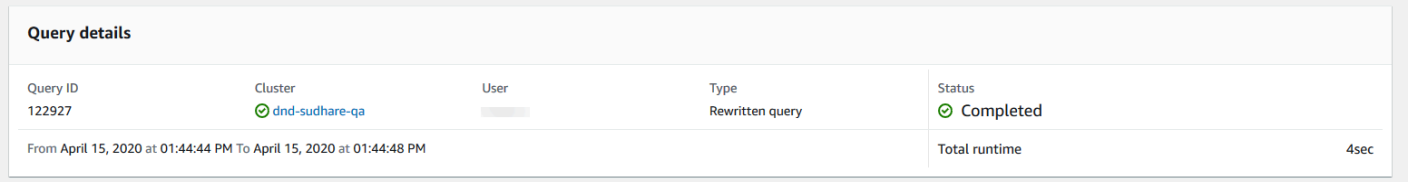
   > **Note**
   >
   > 当您向下钻取查询运行时间图表中的查询时，您还可以从集群详细信息页面、查询历史记录选项卡中导航至查询详细信息页面。

查询详细信息页面包含以下部分：

- 重写查询的列表，如以下屏幕截图所示。




- 查询详细信息部分，如以下屏幕截图所示。



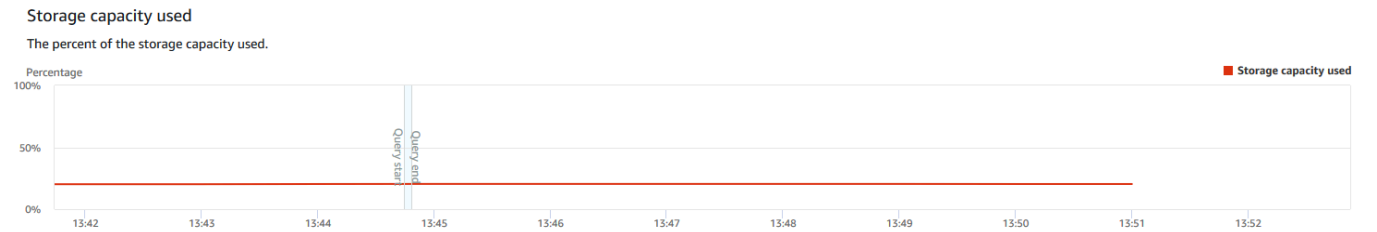

- 查询详细信息选项卡，其中包含所运行的 SQL 和有关此运行的执行详细信息。
- 查询计划选项卡，其中包含查询计划步骤和有关此查询计划的其他信息。此表还包含当查询运行时的集群的图表。
  - 集群运行状况



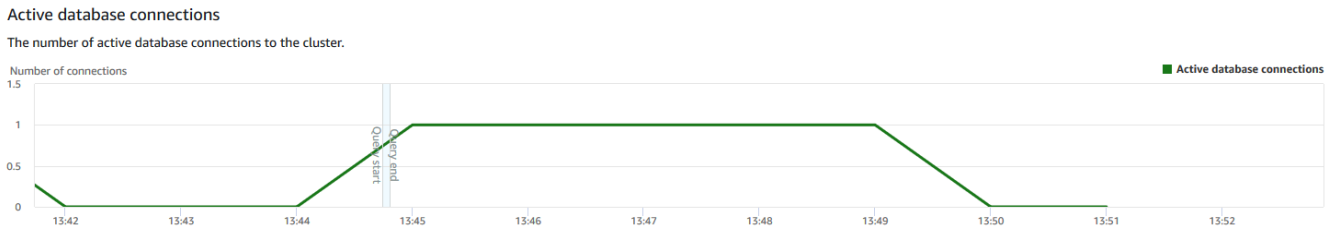

  - CPU 使用率




  - 已使用的存储容量



- 活动的数据库连接数




## 创建警报

您在 Amazon Redshift 控制台中创建的警报是 CloudWatch 警报。这些警报有助于您针对集群或无服务器实例作出积极主动的决策。您可以针对 Amazon Redshift 中的性能数据中列出的任意指标设置一个或多个警报。例如，针对集群节点设置高 `CPUUtilization` 警报有助于在该节点过度使用的情况下发出指示。高 `DataStorage` 警报用于跟踪无服务器命名空间为数据使用的存储空间。

从操作中，您可以修改或删除警报。您还可以创建 chime 或 slack 提示，以通过指定 Slack 或 Amazon Chime Webhook URL 将提示从 CloudWatch 发送到 Slack 或 Amazon Chime。

在本节中，您可以了解如何使用 Amazon Redshift 控制台创建告警。您可使用 CloudWatch 控制台或您用于指标的任何其他方法（如使用 AWS CLI 或 AWS 开发工具包）创建告警。

要使用 Amazon Redshift 控制台创建 CloudWatch 告警

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。

   如果您使用的是 Amazon Redshift Serverless，请选择控制面板右上角的转到。
2. 在导航菜单中，选择警报，然后选择创建警报。
3. 在创建警报页面上，输入相应属性以创建 CloudWatch 告警。
4. 选择创建警报。

## 终止运行的查询

您也可以使用查询页面终止当前正在运行的查询。

> **Note**
>
> 在 Amazon Redshift 控制台中终止查询和加载需要特定权限。如果您希望用户能够停止查询和加载，请确保将 `redshift:CancelQuerySession` 操作添加到您的 AWS Identity and Access Management (IAM) 策略。无论您在 IAM 中选择 Amazon Redshift 只读 AWS 托管策略还是创建自定义策略，此要求都适用。拥有 Amazon Redshift 完全访问策略的用户已具备终止查询和加载所需的权限。有关用于 Amazon Redshift 的 IAM 策略中的操作的更多信息，请参阅管理对资源的访问。

终止正在运行的查询

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择查询和加载以便显示您的账户的查询列表。
3. 在列表中选择要终止的正在运行的查询，然后选择终止查询。

## CloudWatch 控制台中的性能指标

在 CloudWatch 控制台中使用 Amazon Redshift 指标时，请注意以下事项：

- 查询和加载性能数据只能通过 Amazon Redshift 控制台访问。
- CloudWatch 中的一些指标使用的单位与 Amazon Redshift 控制台不同。例如，`WriteThroughput` 以 GB/s 显示（而在 CloudWatch 中，则以 Bytes/s 显示），该单位与典型的节点存储空间的相关性更高。

使用 CloudWatch 控制台中的 Amazon Redshift 指标、命令行工具或 Amazon 开发工具包时，请注意以下概念：

1. 首先，指定使用的指标维度。维度是帮助您对某指标进行唯一标识的名称/值对。Amazon Redshift 的维度为 `ClusterIdentifier` 和 `NodeID`。CloudWatch 控制台中提供了 `Redshift Cluster` 和 `Redshift Node` 视图，以便您可以轻松选择特定于集群的维度和特定于节点的维度。有关维度的更多信息，请参阅《CloudWatch 开发人员指南》中的维度。
2. 接下来，指定指标名称，例如 `ReadIOPS`。

下表总结了可供您使用的 Amazon Redshift 指标维度的类型。根据指标，每隔 1 分钟或 5 分钟免费提供一次数据。有关更多信息，请参阅 Amazon Redshift 指标。

| CloudWatch 命名空间 | 维度 | 说明 |
| --- | --- | --- |
| AWS/Redshift | NodeID | 筛选条件请求的特定于集群节点的数据。NodeID 是“领导”、“共享”或“计算 N”，其中 N 是集群中节点的数目（0、1 等）。“共享”意味着集群只有一个节点，即领导节点和计算节点合并到了一起。 |
| AWS/Redshift | ClusterIdentifier | 筛选条件请求的特定于集群的数据。特定于集群的指标包括 HealthStatus 、MaintenanceMode 和 DatabaseConnections 。此维度的一般指标（例如，ReadIOPS）同样是表示节点指标数据聚合的节点的指标。在解析这些指标时请小心，因为它们是领导节点和计算节点的聚合行为。 |

网关和卷指标的使用方式类似于其他服务指标。许多常见任务在 CloudWatch 文档中进行了概述，其中包括：

- 查看可用的指标
- 获取指标的统计数据
- 创建 CloudWatch 告警

## 查询分析器

本文档介绍了查询分析器，这是一种用于分析查询组件和性能的图形工具。

查询分析器是一项查询监控和故障排除特征，可通过 Amazon Redshift 控制台查看。这对于分析查询性能很有帮助。其主要用途是显示查询的可视化和图形运行顺序、执行计划和统计信息，使它们更易于理解和故障排除。查询分析器有助于分析以下类型的查询组件：

- 子查询 - 子查询是查询工作的一部分。如果将一个查询拆分为多个子查询进行处理比将其作为单个大型查询来处理更为高效，则 Amazon Redshift 可能会拆分查询。在分析器中，您可以看到每个子查询的属性。子查询由流和其他子组件组成。

查询分析器显示的子查询类型通常包括以下几种：

- 临时表查询：此子查询的文本以 CREATE TEMP TABLE 命令开头。此子查询会创建临时表，让其他子查询来处理。
- 统计查询：查询分析器在此子查询的开头添加以下注释来协助进行识别：

```
-- collect statistics of child query queryID
```

此子查询收集 Amazon Redshift 查询引擎用于优化性能的信息。

> **Note**
>
> 查询分析器将用户提供的查询显示为 Amazon Redshift 运行的最终子查询。

- 流 - 流是在可用计算节点切片上分组的分段的集合。每个子查询由一个或多个分段组成。在查询分析器中，您可以查看每个流的属性，例如其执行时间。通过浏览流列表，您可以快速发现性能瓶颈。
- 分段 - 分段是单个过程可以运行的多个步骤的组合。分段也是计算节点切片可执行的最小编译单元。切片是 Amazon Redshift 中并行处理的单元。流中的分段并行运行。查询分析器不会以图形方式显示分段，但您可以在步骤的详细信息窗格中访问该步骤的分段信息。
- 步骤 - 每个分段都由一组步骤组成。步骤是查询中的一项工作。例如，步骤可以包括哈希连接或扫描，扫描会从表中读取记录。

有关流、分段和步骤的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的查询计划和执行工作流。

查询探查器显示由 SYS_QUERY_HISTORY、SYS_QUERY_DETAIL、SYS_QUERY_EXPLAIN 和 SYS_CHILD_QUERY_TEXT 视图返回的信息。有关这些视图的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 SYS_QUERY_HISTORY、SYS_QUERY_DETAIL、SYS_QUERY_EXPLAIN 和 SYS_CHILD_QUERY_TEXT。

查询分析器仅显示最近在数据库上运行的查询的查询信息。如果之前没有可用的信息，则使用预先填充的缓存数据而不是直接针对数据库运行的查询将不会生成查询性能分析。这是因为 Amazon Redshift 不会为其生成查询计划。

## 使用查询探查器的先决条件

SYS 监控视图专为易于使用和降低复杂性而设计，并提供了一系列完整的指标来进行有效的监控和故障排除。无论集群的大小或活动如何，SYS 监控视图还可以保证过去七天的查询历史记录。用户只能看到他们运行的查询，而超级用户可以看到来自所有用户的查询。

您的 IAM 用户账户或角色需要权限，才能访问控制台的查询和数据库监控部分。本节介绍如何向用户账户或角色添加权限。

使用以下策略向 IAM 用户账户或角色添加最低权限：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "redshift:DescribeClusters",
                "redshift-serverless:ListNamespaces",
                "redshift-serverless:ListWorkgroups",
                "redshift-data:ExecuteStatement",
                "redshift-data:DescribeStatement",
                "redshift-data:GetStatementResult"
            ],
            "Resource": [
                "arn:aws:redshift-serverless:us-east-1:111122223333:*",
                "arn:aws:redshift:us-east-1:111122223333:*"
            ]
        }
    ]
}
```

主题

- 向角色授予查询监控权限
- 向用户授予查询监控权限
- 使用您的 IAM 身份的临时凭证

向角色授予查询监控权限

具有 sys:monitor 权限的角色的用户可以查看所有查询。具有 sys:operator 权限的角色的用户可以取消查询、分析查询历史记录和执行 vacuum 操作。

授予角色查询监控权限

1. 使用以下命令提供系统监控访问权限，其中 *role-name* 是您想要为其提供访问权限的角色的名称。

```
grant role sys:monitor to "IAMR:role-name";
```

2. （可选）使用以下命令提供系统操作员访问权限，其中 *role-name* 是您想要为其提供访问权限的角色的名称。

```
grant role sys:operator to "IAMR:role-name";
```

向用户授予查询监控权限

具有 sys:monitor 权限的用户可以查看所有查询。具有 sys:operator 权限的用户可以取消查询、分析查询历史记录和执行 vacuum 操作。

授予用户查询监控权限

1. 使用以下命令提供系统监控访问权限，其中 *user-name* 是您想要为其提供访问权限的用户的名称。

```
grant role sys:monitor to "IAMR:user-name";
```

2. （可选）使用以下命令提供系统操作员访问权限，其中 *-name* 是您想要为其提供访问权限的用户的名称。

```
grant role sys:operator to "IAMR:user-name";
```

使用您的 IAM 身份的临时凭证

仅在连接到集群时此选项才可用。使用此方法，查询探查器会将用户名映射到您的 IAM 身份并生成临时密码，以便使用您的 IAM 身份连接到数据库。必须允许使用此方法进行连接的用户对

`redshift:GetClusterCredentialsWithIAM` 拥有 IAM 权限。要防止用户使用此方法，请修改其 IAM 用户或角色以拒绝此权限。

## 在 Amazon Redshift 控制台中访问查询分析器来分析查询

您可以访问 Amazon Redshift Serverless 或预置 Amazon Redshift 的查询分析器。有关详细信息，请参阅下面几节：

主题

- 在 Amazon Redshift 控制台中访问 Amazon Redshift Serverless 的查询分析器
- 在 Amazon Redshift 控制台中访问预置 Amazon Redshift 的查询分析器

### 在 Amazon Redshift 控制台中访问 Amazon Redshift Serverless 的查询分析器

要访问 Amazon Redshift Serverless 的查询分析器，请执行以下操作：

- 打开 Amazon Redshift Serverless 控制台。
- 在导航窗格的监控下面，选择查询和数据库监控。
- 选择一个工作组。
- 选择查询和数据库监控。
- 选择一个查询。
- 在查询详细信息页面中选择查询计划选项卡。

如果查询计划可用，您将看到子查询列表。选择一个查询以在查询分析器中查看该查询。

### 在 Amazon Redshift 控制台中访问预置 Amazon Redshift 的查询分析器

要访问预置 Amazon Redshift 的查询分析器，请执行以下操作：

- 打开 Amazon Redshift 预置集群控制面板。
- 选择一个集群
- 选择查询监控。
- 连接到数据库
- 选择查询和数据库监控。
- 选择一个查询。

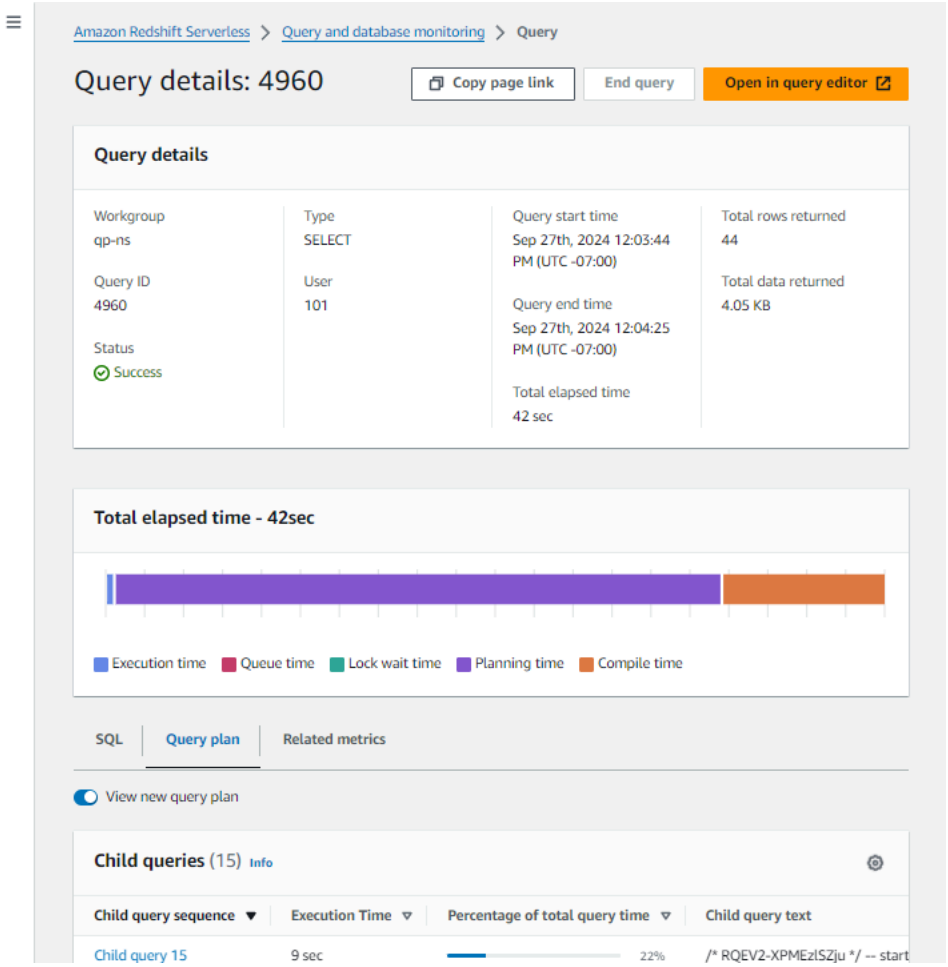
如果查询计划可用，您将看到子查询列表。选择一个查询以在查询分析器中查看该查询。

## 查询分析器用户界面

查询分析器使用以下页面来显示有关查询的信息：

- 查询详细信息页面：此页面显示查询的统计信息和子查询。
- 子查询页面：此页面显示子查询的统计信息、流和执行计划的直观表示。当您从查询和数据库监控页面的子查询列表中选择子查询时，控制台会显示此页面。

### 查询详细信息页面




查询详细信息页面包含下列组件：

- 顶部窗格 - 页面顶部的窗格显示有关查询的详细信息，例如状态和类型。若要详细了解顶部窗格所示信息的来源，请参阅《Amazon Redshift 数据库开发人员指南》中的 SYS_QUERY_HISTORY。
- SQL 选项卡 - 底部面板的这个选项卡显示了原始用户查询的 SQL 文本。

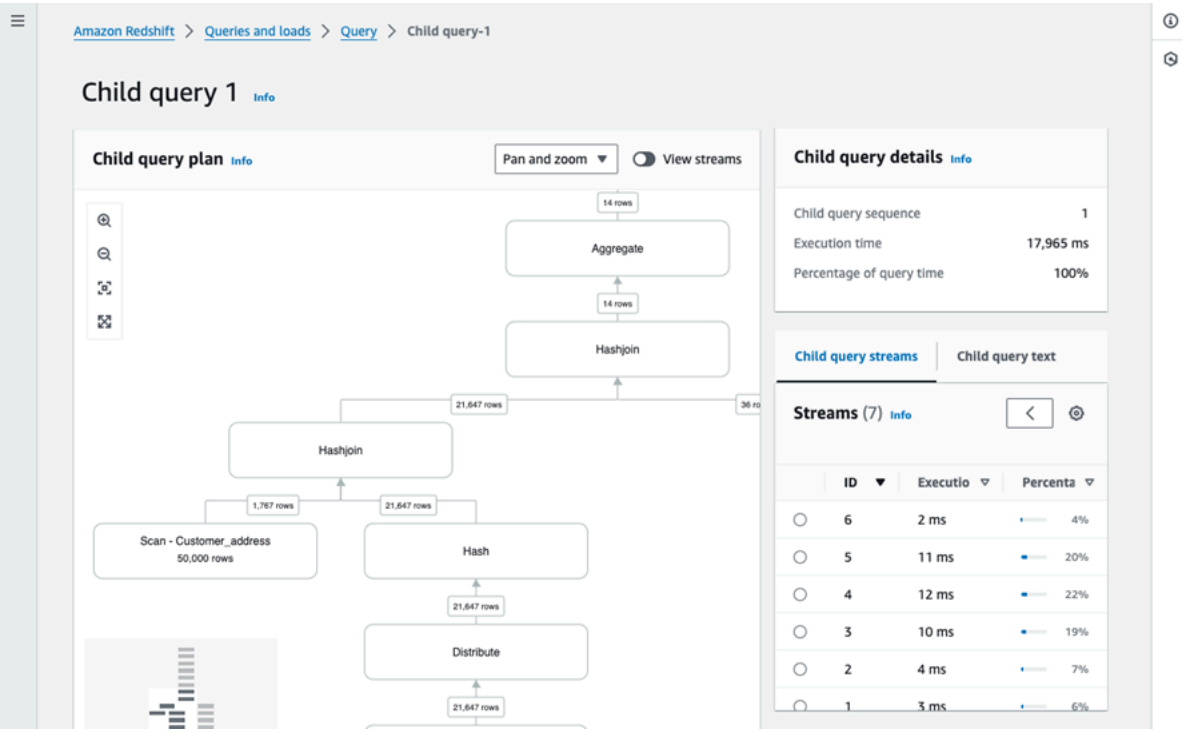
- “查询计划”选项卡 - 底部面板的这个选项卡显示了 Amazon Redshift 用来为用户查询准备数据和统计信息的子查询列表。默认情况下，子查询列表显示有关每个子查询的信息和汇总统计信息。若要详细了解此页面所示信息的来源，请参阅《Amazon Redshift 数据库开发人员指南》中的 SYS_QUERY_DETAIL。

  您可以使用“首选项”菜单在子查询列表中添加或删除列。

- 相关指标 - 底部面板的这个选项卡显示了查询的以下 CloudWatch 指标：
  - 已使用的 RPU 容量（用于无服务器工作组）：查询使用的计算容量，用 Redshift 处理器（RPU）来衡量。有关更多信息，请参阅 Amazon Redshift Serverless 的计算容量。
  - 集群运行状况、CPU 利用率、已用存储容量（对于预置集群）：状态和查询使用的系统资源。
  - 活动的数据库连接：查询的 `DatabaseConnections` 指标。

  有关 CloudWatch 指标的更多信息，请参阅Amazon Redshift 中的性能数据。

子查询页面




子查询页面包含下列组件：

- 子查询下拉列表 - 此控件显示每个子查询的序列名称和执行时间。您可以通过在此控件中选择其他子查询来导航到这些子查询。
- 侧面板 - 此面板包含用于显示子查询流和子查询文本的选项卡。

- “子查询流”选项卡 - 顶部面板的这个选项卡显示以下信息：
  - 流 - 此窗格显示子查询中的流列表。此窗格显示有关 Amazon Redshift 用于优化查询的流的信息和汇总数据。有关此窗格的详细信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 SYS_QUERY_DETAIL。

    您可以使用“首选项”菜单在流面板中添加或删除列。您可以使用流窗格中的齿轮图标访问“首选项”菜单。
  - “子查询”窗格 - 子查询中步骤的图形表示。有关子查询窗格的信息，请参阅下面的“子查询”窗格。
- “子查询文本”选项卡 - 顶部面板的这个选项卡显示子查询的 SQL 代码。
- “子查询详细信息”窗格 - 右侧面板中的这个窗格显示有关子查询的详细信息。有关此窗格的详细信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 SYS_QUERY_DETAIL。
- “流详细信息”窗格 - 在流窗格中选择流时，流详细信息窗格会显示有关该流的信息。有关此面板的详细信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 SYS_QUERY_DETAIL。
- “步骤详细信息”窗格 - 在流窗格或子查询计划的图表中选择步骤时，步骤详细信息窗格将显示有关该步骤的信息。有关此窗格的详细信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 SYS_QUERY_DETAIL：

## “子查询”窗格

查询分析器在子查询窗格中显示子查询，以图形方式表示所选子查询中的步骤。

子查询窗格显示运行顺序和步骤之间的关系。例如，如果一个步骤将来自其他两个步骤的输出连接起来，则子查询窗格会将该步骤显示为一个树节点，有两个节点馈入其中：




默认情况下，子查询窗格不显示包含步骤的流。要显示 Amazon Redshift 用于对子查询中的步骤进行逻辑分区的流，请选择查看流。选择查看流时，子查询窗格会显示查询的流中包含的步骤。

子查询窗格不显示分段信息。要查看某个步骤的分段，请选择该步骤。然后，步骤详细信息窗格会显示该步骤的分段。

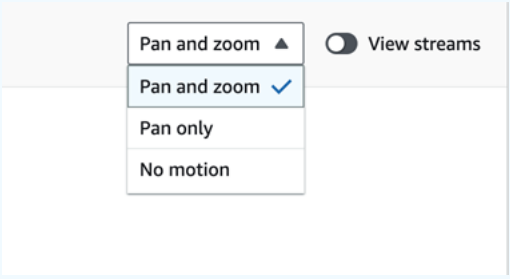
### 在“子查询”窗格中导航

在子查询窗格中，您可以选择步骤来查看有关这些步骤的详细信息。您还可以平移和缩放工作区，以便更好地可视化查询计划中的步骤。

您可以使用以下方法在子查询计划中选择节点、平移和缩放：

- 使用鼠标 - 您可以选择节点，点击并拖动工作区进行平移，按住 **Ctrl**（Windows）或 **CMD**（Mac）和鼠标滚轮进行缩放。选择一个节点会使工作区缩放和平移，从而突出显示该节点。如果您在工作区中选择一个流，则该流将在流列表中突出显示。如果您在工作区中选择一个步骤，则步骤详细信息窗格会显示有关该步骤的信息。
- 使用工作区左上角的缩放和适合控件 - 这些控件使您可以放大、缩小、缩放以适合整个工作区以及进入全屏模式。当您缩放以适合整个查询计划时，工作区会使查询计划在水平和垂直方向上居中。
- 使用工作区右下角的小地图 - 您可以使用工作区左下角的小地图控件平移或缩放工作区。
- 在流窗格中选择一个流 - 如果在流窗格中选择一个流，则工作区会平移和缩放以显示所选流，并在流详细信息窗格中显示有关该流的信息。
- 在流详细信息窗格中选择一个步骤 - 如果在流详细信息窗格中选择一个步骤，则工作区会平移和缩放以显示所选步骤，并在步骤详细信息窗格中显示有关该步骤的信息。

> **Note**
>
> 当您在工作区或另一个窗格中选择一个步骤时，工作区会尝试缩放和平移，以使所选步骤更清晰。
>
> 当您从工作区或另一个窗格选择一个流或步骤时，只有在工作区右上角的控件中选择了平移和缩放时，工作区才会缩放并平移到该流或步骤。通过从下拉菜单中选择相应的设置，您可以将此行为限制为平移和缩放、仅平移或不移动。



### 排查使用查询分析器进行查询的问题

如果要对查询进行故障排除，则可以选择子查询来确定哪个流占用最高的总查询时间百分比。这是一种快速方法，可确定应进一步分析查询的哪一部分。

在知道哪个子查询花费的时间最多之后，请查看其步骤，以了解哪个联接或扫描可能导致性能降低。

## 查询和数据库监控

本文档描述了查询和数据库监控页面，用于分析 Amazon Redshift 预调配集群或无服务器工作组的性能的 AWS 管理控制台功能以及针对它们运行的查询。

您可以使用查询和数据库监控页面调查以下情况：

- 监控指定时间段内的数据仓库指标
- 查询会对数据仓库的整体性能产生哪些影响
- 了解查询运行时间细分为其生命周期事件的情况，例如锁定等待时间、编译时间和执行时间
- 哪些用户在指定时间段内运行的资源密集型查询最多
- 监控补丁事件对查询性能产生的影响

主题


- 权限
- 查询和数据库监控控制台


### 权限

用于访问控制台的 AWS 账户的权限会影响查询和数据库监控页面显示的查询。默认情况下，您只能查看自己的查询。要查看其他用户拥有的查询，请将 SYS:MONITOR 角色授予您的账户。要允许用户从查询和数据库监控页面终止正在运行的查询，请向用户授予 SYS:OPERATOR 权限。

要自动将 sys:monitor 角色分配给 Amazon Redshift Serverless 或预置的 IAM 用户或角色，请运行以下命令：

```
create role monitor;
grant role sys:monitor to role monitor;
```

要更新用于查询监控的 IAM 角色，请执行以下操作：

1. 选择标签选项卡。
2. 选择管理标签。
3. 添加具有键 **RedshiftDbRoles** 和值 **monitor** 的标签。
4. 保存更改

要将数据库凭证添加到用户，请运行以下命令：

```
grant role sys:monitor to <username>
```

有关使用 GRANT 命令的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 GRANT。

要使用查询监控，IAM 用户需要访问 Amazon Redshift 数据面板的权限。确保 IAM 用户在其权限策略中具有以下权限：

```
{
    "Sid": "DataAPIPermissions",
    "Action": [
        "redshift-data:ExecuteStatement",
        "redshift-data:CancelStatement",
        "redshift-data:GetStatementResult",
        "redshift-data:DescribeStatement",
        "redshift-data:ListDatabases"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:redshift-serverless:us-
west-2:123456789012:workgroup/01234567-89ab-cdef-0123-456789abcdef"
},
```

使用您的 IAM 身份的临时凭证

仅在连接到集群时此选项才可用。使用此方法，查询和数据库监控功能会将用户名映射到您的 IAM 身份并生成临时密码，以便使用您的 IAM 身份连接到数据库。必须允许使用此方法进行连接的用户对 `redshift:GetClusterCredentialsWithIAM` 拥有 IAM 权限。要防止用户使用此方法，请修改其 IAM 用户或角色以拒绝此权限。

## 查询和数据库监控控制台

此部分介绍如何使用查询和数据库监控控制台页面。

您可以使用查询和数据库监控控制台来快速了解数据仓库的性能。您可以监控一段时间内的数据仓库性能，并检查数据仓库的预配置集群或单个查询的性能，以便充分识别瓶颈和其他需要改进的方面。

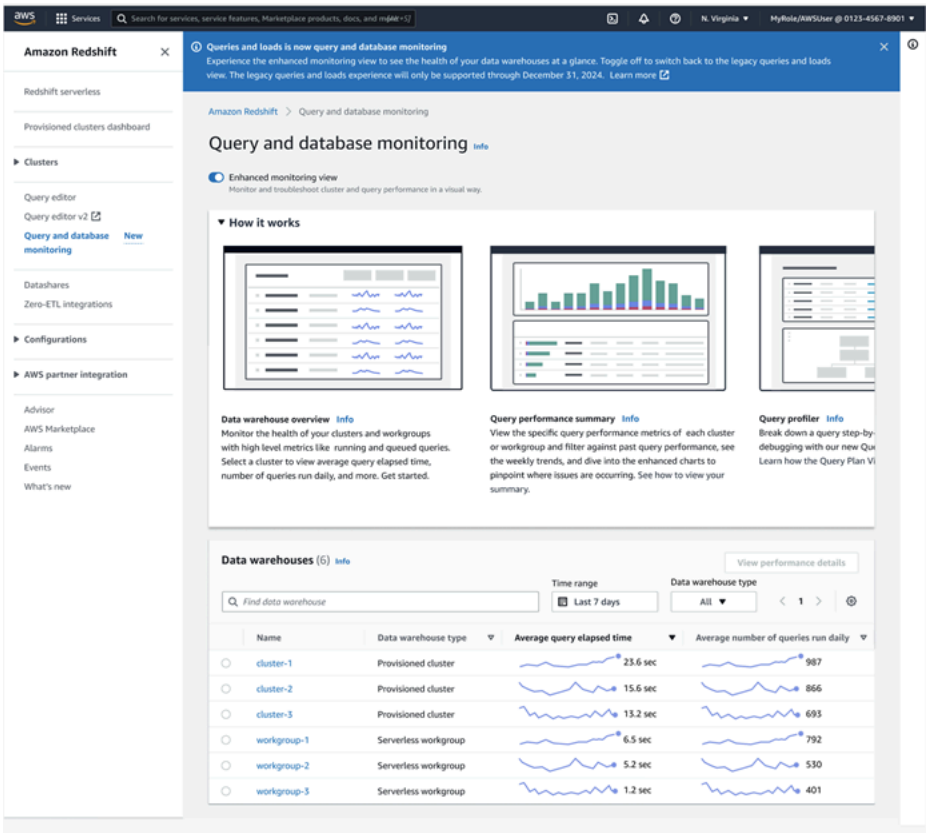
查询和数据库监控页面具有以下特征：

- 提高安全性 – 您需要提升的权限才能监控其他用户的查询。有关更多信息，请参阅 权限。
- 七天查询历史记录 – 保证可访问七天的查询历史记录
- 查询监控 – 您可以在用户查询级别监控预调配集群和无服务器工作组中的查询。
- 查询趋势分析 – 您可以比较符合指定条件的相似查询的性能。

要访问查询和数据库监控页面，请执行以下操作：

1. 通过 https://console.aws.amazon.com/redshiftv2/ 登录到 Amazon Redshift 控制台。
2. 从导航窗格中选择查询和数据库监控。

查询和数据库监控控制台页面如下所示：




查询和数据库监控页面具有以下组成部分：

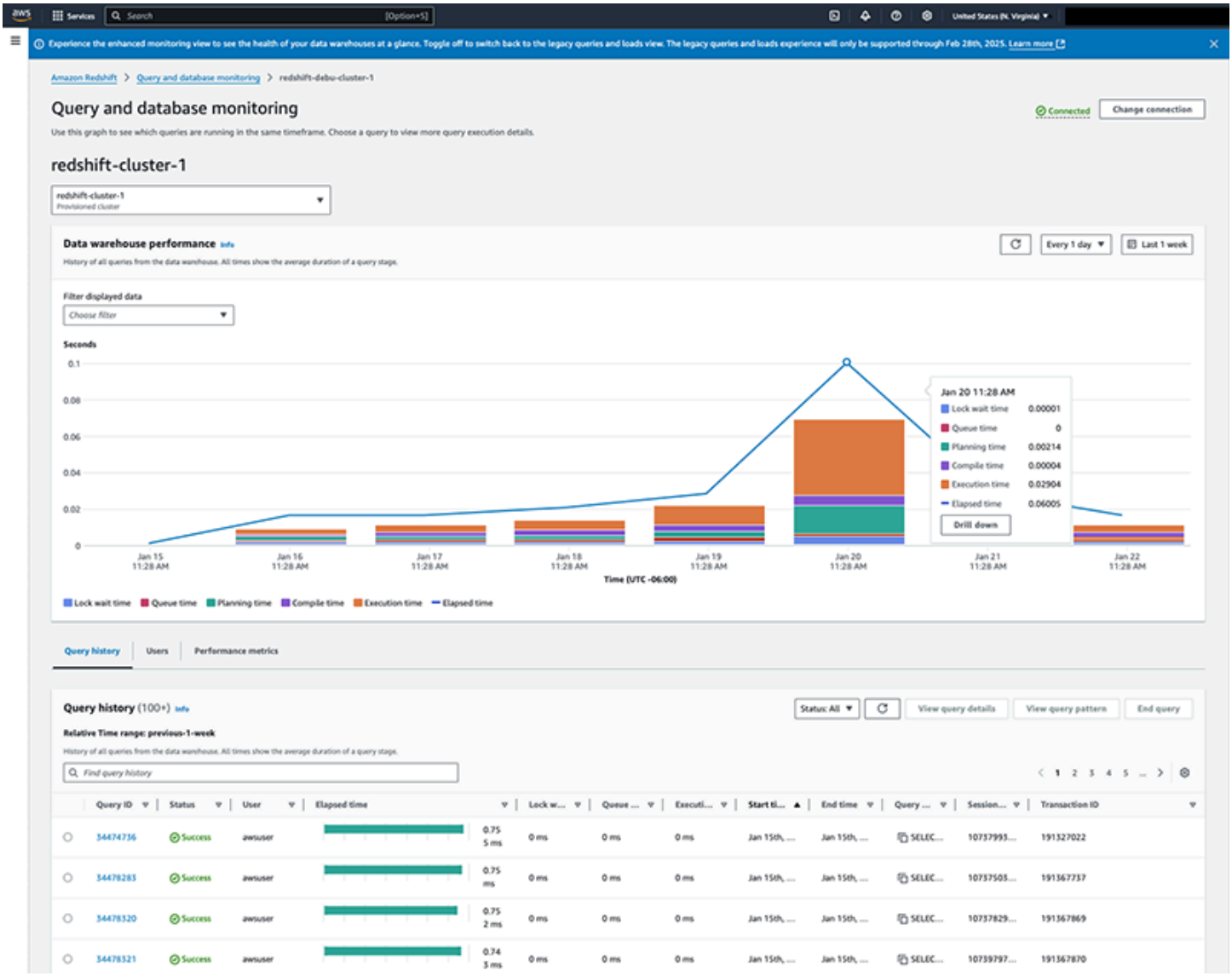
- 数据仓库概览 – 监控预调配集群和无服务器工作组的平均查询性能。您可以在此页面上查看峰值或高活动期间的统计数据，快速了解特定集群或工作组的性能问题。
- 查询性能摘要 – 监控特定预调配集群或无服务器工作组的平均查询性能。还可以通过单击数据仓库概览列表中的某个集群或工作组来访问查询性能摘要页面。

主题


- 查询性能摘要
- 查询详细信息
- 查询模式


## 查询性能摘要

在从查询和数据库监控页面中选择查询性能摘要，或从数据仓库概览页面选择集群或工作组时，控制台会显示单个预调配集群或无服务器工作组的性能摘要。




此页面包含以下组成部分：

- 集群或工作组下拉列表 – 选择要分析的集群或工作组。
- 数据仓库性能 – 此窗格显示指定时间段内的集群或工作组历史记录，并显示每个查询阶段所花费的时间。如果您看到某个查询阶段出现峰值，如上图中所示，1 月 20 日的计划和执行时间已增加，则可以使用此信息来识别查询性能问题。显示的默认时间段为最近七天，但您可以根据分析需求调整时间段。
- 查询历史记录 – 此窗格显示指定筛选器内每个查询运行的性能历史记录。您可以使用此信息来排查单个查询的性能问题。要进一步分析某个特定查询，可以从此图表或数据仓库性能图表中选择该查询。

将鼠标悬停在图表上的数据点上方时，会出现一个弹出窗口。此弹出窗口显示该数据点的时间指标细分。如果数据点包含查询数据，则可以选择向下钻取，将图表的时间范围更新为该数据点的下一个最小时间单位。这些时间单位如下所示：

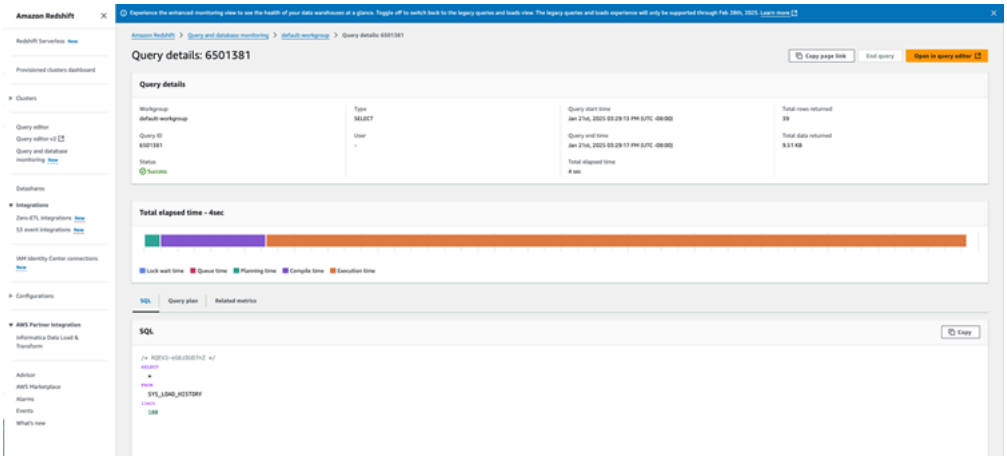
- 天
- 小时
- 15 分钟
- 5 分钟
- 1 minute

例如，如果您在数据点上选择向下钻取，则图表的范围将更改为一天。再次选择向下钻取会将图表的范围更改为一小时。

- 查询探查器 - 用于监控查询性能的图形工具。有关更多信息，请参阅 查询分析器。

查询详细信息

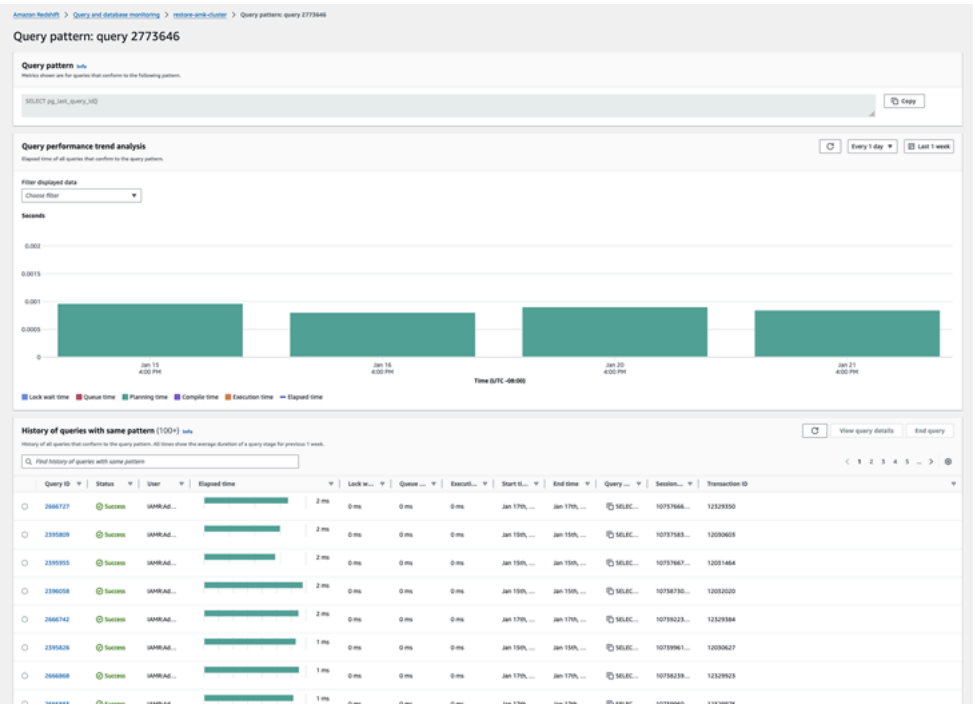
在从集群或工作组详细信息页面的数据仓库性能或查询历史记录窗格中选择查询时，将打开查询详细信息页面。




有关查询详细信息页面的信息，请参阅查询详细信息页面。

查询模式

可以通过在查询性能摘要页面的查询历史记录窗格中选择查看查询模式按钮，查看具有相同模式的查询历史记录。查询模式页面显示您指定的 SQL 语句在过去一周内检索到的所有查询。




查询模式页面包含以下组成部分：

- 查询模式 – 检索正在分析的查询的 SQL 语句。
- 查询性能趋势分析 – 显示查询模式选择的所有查询所用时间的条形图。按天对结果进行分组。
- 具有相同模式的查询历史记录 – 查询模式选择的查询的每个阶段所花费的时间。

使用查询模式页面，可以获得以下洞察：

- 在每天的特定时间运行的查询的趋势
- 定期对数据仓库运行的查询的运行时峰值。

## 基于 SYS 视图的查询和数据库监控

本文档介绍 Amazon Redshift 控制台中为查询和数据库监控页面提供数据的 SYS 视图，它是一种用于分析查询的组件和性能的工具。有关查询和数据库监控页面的信息，请参阅查询和数据库监控。

查询和数据库监控页面具有显示由 SYS 视图提供的信息的功能。控制台视图包括查询探查器，它显示查询的图形执行计划。要切换到基于 SYS 的视图，请按照此处的步骤针对新的查询和数据库监控页面授予正确的访问权限和权限。

查询和数据库监控页面的基于 SYS 的视图功能具有以下功能：

- 提高安全性 - 您需要提升的权限才能监控其他用户的查询
- 七天查询历史记录 – 保证可访问七天的查询历史记录
- 查询探查器 - 用于监控查询性能的图形工具。有关更多信息，请参阅 查询分析器。

默认情况下，您只能查看自己的查询。要查看其他用户拥有的查询，请将 SYS:MONITOR 角色授予您的账户。要支持用户结束正在运行的查询，请向该用户授予 SYS:OPERATOR 权限。

要向数据库用户或角色授予查看所有用户拥有的查询的权限，请运行以下命令：

```
grant role sys:monitor to "IAM:role-name";
grant role sys:monitor to "IAM:user-name";
```

要自动将 sys:monitor 角色分配给 Amazon Redshift Serverless 或预置的 IAM 用户或角色，请运行以下命令：

```
 create role monitor;
grant role sys:monitor to role monitor;
```

要更新用于查询监控的 IAM 角色，请执行以下操作：

1. 选择标签选项卡。
2. 选择管理标签。
3. 添加具有键 **RedshiftDbRoles** 和值 **monitor** 的标签。
4. 保存更改

要将数据库凭证添加到用户，请运行以下命令：

```
grant role sys:monitor to <username>
```

## 权限

要使用查询监控，IAM 用户需要访问 Amazon Redshift 数据面板的权限。确保 IAM 用户在其权限策略中具有以下权限：

```
{
    "Sid": "DataAPIPermissions",
    "Action": [
```

```
        "redshift-data:ExecuteStatement",
        "redshift-data:CancelStatement",
        "redshift-data:GetStatementResult",
        "redshift-data:DescribeStatement",
        "redshift-data:ListDatabases"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:redshift-serverless:us-
west-2:123456789012:workgroup/01234567-89ab-cdef-0123-456789abcdef"
},
```

## 连接到数据库

在使用增强型查询监控功能之前，必须先连接到数据库，才能访问基于 SYS 视图的信息。要连接到数据库，请使用以下凭证之一：

- 用户名和密码
- 与您的 IAM 角色关联的临时凭证
- 数据库用户

在使用增强型查询监控时，请注意以下几点：

- 对于预置集群，您必须连接到数据库，因为增强型查询监控使用 SYS 视图。这些视图提高了安全性，并且需要提升的权限，才能访问与其他用户拥有的查询有关的数据。
- 使用基于 SYS 视图的查询和数据库监控页面时，如果用户账户没有数据库超级用户角色，则只有 `user_id` 是可见的。对于非超级用户，用户名是隐藏的。
- 作为基于 SYS 视图的查询和数据库监控页面体验的一部分，查询执行进程 ID (`p_id`) 显示在列标题 `session_id` 下。

# Amazon Redshift 预置集群和无服务器工作组的跟踪

当 Amazon Redshift 发布新版本时，它会更新 Amazon Redshift 数据仓库（无服务器工作组或预置集群）的版本。您可以控制数据仓库是更新到最新版本还是先前经认证的版本。

无服务器工作组或预置集群的跟踪决定了在版本更新期间应用哪个已发布版本。Amazon Redshift 会在指定的维护时段内更新预置集群，并通常会在空闲时段内更新无服务器工作组。有关 Redshift Serverless 何时更新工作组的详细信息，请参阅更新无服务器工作组。

当 Amazon Redshift 发布新版本时，该版本将分配给当前版本跟踪，前一个版本将分配给早先版本跟踪。要为数据仓库设置跟踪，请指定下列值之一：

- 当前版本：通过当前版本跟踪，您将获得最新的已认证发行版，其中包含最新功能、安全更新和性能增强。
- 早先版本：使用早先版本跟踪，您将使用前一个已认证版本。

例如，假设无服务器工作组当前正在运行版本 1.0.2762，而 Amazon Redshift 发布了 Redshift Serverless 版本 1.0.3072。如果跟踪值为当前版本，则工作组将更新为版本 1.0.3072（最新版本）。如果将跟踪值设置为早先版本，则当下一个早先版本跟踪版本发布时，就会更新您的工作组。

使用早先版本跟踪功能，您可以选择在早先版本跟踪中运行 Amazon Redshift 数据仓库的一个子集。这支持在将版本应用于早先版本跟踪上的数据仓库之前，对设置为当前版本跟踪的数据仓库进行 1-6 周的测试和集成验证。默认情况下，Amazon Redshift 会在当前版本跟踪上创建所有集群和工作组，以利用最新的已认证版本。但是，在生产环境中使用 Amazon Redshift 早先版本跟踪，并在测试和开发环境中使用当前版本跟踪，可让您更有精力和时间来评估最新版本。早先版本跟踪可确保最大的稳定性，非常适合生产环境中的任务关键型工作负载。

> **Note**
>
> 在短时间内，早先版本跟踪版本可能与当前版本跟踪版本相同。在当前版本跟踪尚未升级到下一个版本时，会发生这种情况。通常，当前版本跟踪版本优先于早先版本跟踪版本。

## 在跟踪之间切换

更改 Amazon Redshift 资源的跟踪通常是一次性决定。更改跟踪时要慎重。有关哪些功能位于哪些数据仓库版本中的信息，请参阅 Amazon Redshift 的集群版本。

如果您将跟踪从早先版本更改为当前版本，我们会将数据仓库更新为当前版本跟踪发行版。如果您将数据仓库的跟踪更改为早先版本，我们将按如下方式更新数据仓库：

- 对于无服务器工作组，我们会在空闲时段更新数据仓库的版本。有关 Redshift Serverless 如何更新工作组版本的更多详细信息，请参阅更新无服务器工作组。
- 对于预置集群，在当前版本跟踪发行版之后的新版本推出之前，我们不会更新您的数据仓库。

## 跟踪和还原

对于无服务器工作组，快照会继承目标 Amazon Redshift 数据仓库的跟踪。例如，如果您为设置为“早先版本跟踪”的工作组创建快照，并将该快照应用于设置为“当前版本跟踪”的工作组，则该工作组的跟踪设置将为“当前版本”。

对于预置集群，快照会继承源 Amazon Redshift 数据仓库的跟踪。如果您在拍摄快照后更改源数据仓库的跟踪，则快照和源数据仓库将位于不同的跟踪上。从快照还原时，新的数据仓库将位于从快照源继承的跟踪上。在还原操作完成后，您可以更改跟踪。

调整数据仓库的大小不会影响其跟踪。

## 更新无服务器工作组

当工作组的所选跟踪版本有新版本可用时，只要没有待处理的跟踪版本更新请求，Amazon Redshift Serverless 通常会在空闲时段应用更新。如果工作组在 14 天内没有出现空闲时段，Redshift Serverless 会强制更新版本。

Redshift Serverless 仅将工作组更新到下一个更高版本。即使所选的工作组跟踪版本低于工作组的当前版本，Redshift Serverless 也不会跳过中间版本或降级工作组。除非 `Trailing` 跟踪版本跟上，否则工作组不会接收任何主要版本升级。

例如，假设 `Current` 跟踪版本为 186，`Trailing` 跟踪版本为 185。如果您有一个工作组的 `Track` 值为 `Current`（其版本为 186），而您将 `Track` 的值更改为 `Trailing`，Redshift Serverless 不会将该工作组的版本降级到 185。在此场景中，Redshift Serverless 会将工作组保持为版本 186，直到 `Trailing` 跟踪版本等于或高于 186。

如果有待处理的跟踪版本更改，则在应用跟踪版本更改之前，Redshift Serverless 不会将工作组升级到现有跟踪版本的下一个主要版本。跟踪版本更改完成后，Redshift Serverless 会评估条件，确定是否将工作组更新为新跟踪版本下的相应版本。

例如，如果您的工作组设置为 `Current` 跟踪版本且当前跟踪版本为 186，而您将工作组更改为 `Trailing` 跟踪版本，则 Redshift Serverless 需要在应用跟踪版本更改并且将 `Trailing` 版本更新到等于或高于版本 186 之后，才会更新工作组。

> **Note**
>
> 工作组上的任何现有操作，例如从快照还原、更改 KMS 密钥或调整大小，都只能在现有跟踪版本上执行。Redshift Serverless 不使用待处理跟踪版本执行无服务器操作。

如果您有待处理的跟踪版本切换请求，可以使用 UpdateWorkgroup，通过将 `track` 参数设置回其原始值来取消请求。

# 管理版本

跟踪是一系列发行版。您可以决定您的 Amazon Redshift 数据仓库是在当前版本跟踪还是早先版本跟踪上。如果您将数据仓库放在当前版本跟踪上，则它将始终升级到最新的发行版。如果您将资源放在早先版本跟踪上，则它将始终运行就在最近发布的版本之前发布的那个发行版。

对于预置集群，Amazon Redshift 数据仓库的 Amazon Redshift 控制台列表中的版本状态列指示您的其中一个资源是否可用于升级。

# 确定工作组或集群版本

您可以使用 Amazon Redshift 控制台来确定 Amazon Redshift 无服务器工作组版本或预置集群版本引擎。

登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。

Serverless workgroups

对于无服务器工作组，在导航菜单上，选择工作组，然后从列表中选择工作组名称以打开其详细信息。将显示工作组的详细信息。

Provisioned clusters

对于预置集群，在导航菜单上，选择集群，然后从列表中选择集群名称以打开其详细信息。

此时将显示集群的详细信息，其中可能包括集群性能、查询监控、数据库、数据共享、计划、维护和属性选项卡。选择维护选项卡以查看更多详细信息。

在维护部分中，找到当前集群版本。

> **Note**
>
> 对于预置集群，控制台在一个字段中显示版本信息，但它是 Amazon Redshift API 中的两个参数。这些参数是 `ClusterVersion` 和 `ClusterRevisionNumber`。有关更多信息，请参阅 Amazon Redshift API 参考中的集群。

# 零 ETL 集成

零 ETL 集成是一种完全托管式解决方案，可在 Amazon Redshift 中提供来自多个运营和事务性来源的事务和运营数据。通过此解决方案，您可以配置从数据来源到 Amazon Redshift 数据仓库的集成。您无需维护提取、转换、加载 (ETL) 管道。我们自动创建和管理从数据来源到 Amazon Redshift 集群或 Redshift Serverless 命名空间的数据复制任务，从而为您处理 ETL。您可以继续更新和查询源数据，同时使用 Amazon Redshift 完成分析工作负载，例如报告和控制面板。

通过零 ETL 集成，您可以获得最新数据，用于分析、AI/ML 和报告。您可以获得更准确、更及时的洞察，用于业务控制面板、优化的游戏体验、数据质量监控和客户行为分析等应用场景。您可以更有信心地做出以数据为导向的预测，改善客户体验，并在整个企业中推广数据驱动的洞察。

目前，以下数据来源支持零 ETL 集成：

- Amazon Aurora MySQL (AMS)
- Amazon Aurora PostgreSQL (APG)
- Amazon DynamoDB
- Amazon RDS for MySQL
- Amazon RDS for Oracle
- Amazon RDS for PostgreSQL
- Oracle Database@AWS
- 应用程序包括 Salesforce、Salesforce Marketing Cloud Account Engagement、SAP、ServiceNow、Instagram Ads、Meta Ads 和 Zendesk
- 自行管理的 MySQL、PostgreSQL、SQL Server 和 Oracle

要创建零 ETL 集成，您需要指定集成源，并将 Amazon Redshift 数据仓库指定为目标。在初始数据加载之后，该集成会将数据从源数据仓库复制到目标数据仓库中。这时数据在 Amazon Redshift 中变为可用。在创建集成源、创建零 ETL 集成以及创建 Amazon Redshift 数据仓库时，您可以控制数据的加密。该集成还会监控数据管道的运行状况，并在可能的情况下从问题中恢复。您可以创建相同类型的多个源与单个 Amazon Redshift 数据仓库的集成，从而获得跨多个应用程序的全面洞察。

当数据在 Amazon Redshift 中之后，您可以使用 Amazon Redshift 提供的分析功能。例如，内置机器学习 (ML)、实体化视图、数据共享以及直接访问多个数据存储和数据湖。对于数据工程师，零 ETL 集成提供了对时间敏感型数据的访问，否则，对这些数据的访问可能会被复杂数据管道中的间歇性错误延误。您可以对事务数据运行分析查询和 ML 模型，从而为时间敏感型事件和业务决策提供及时的洞察。

您可以创建 Amazon Redshift 事件通知订阅，这样就能在发生零 ETL 集成事件时收到通知。要查看与集成相关的事件通知列表，请参阅使用 Amazon EventBridge 发送零 ETL 集成事件通知。创建订阅最简单的方式是使用 Amazon SNS 控制台。有关创建和订阅 Amazon SNS 主题的信息，请参阅《Amazon Simple Notification Service 开发人员指南》中的开始使用 Amazon SNS。

在开始使用零 ETL 集成时，请考虑以下概念：

- 源数据库是将其数据复制到 Amazon Redshift 中的数据库。
- 目标数据仓库是数据要复制到的 Amazon Redshift 预置集群或 Redshift Serverless 工作组。
- 目标数据库是您通过零 ETL 集成创建在目标数据仓库中创建的数据库。

有关可用于监视零 ETL 集成的系统表和视图的信息，请参阅使用 Amazon Redshift 系统视图监控零 ETL 集成。

有关零 ETL 集成的每个来源支持的 AWS 区域的列表，请参阅零 ETL 集成的支持区域。

有关零 ETL 集成的定价信息，请参阅相应的定价页面：

- Amazon Redshift 定价
- Amazon Aurora 定价
- Amazon RDS 定价
- Amazon DynamoDB 定价
- AWS Glue 定价

有关零 ETL 集成源的更多信息，请参阅以下主题：

- 对于 Aurora 零 ETL 集成，请参阅《Amazon Aurora 用户指南》中零 ETL 集成的优点、重要概念、限制、配额和支持的区域。
- 对于 RDS 零 ETL 集成，请参阅《Amazon RDS User Guide》中零 ETL 集成的Benefits、Key concepts、Limitations、Quotas 和 Supported Regions。
- 有关 DynamoDB 零 ETL 集成的信息，请参阅《Amazon DynamoDB 开发人员指南》中的 DynamoDB zero-ETL integration with Amazon Redshift。
- 有关与应用程序的零 ETL 集成，请参阅《AWS Glue 开发人员指南》中的零 ETL 集成。

主题

- 将零 ETL 集成与 Amazon Redshift 结合使用时的注意事项

- 开始使用零 ETL 集成
- 在 Amazon Redshift 中创建目标数据库
- 在 Amazon Redshift 中查询复制的数据
- 查看零 ETL 集成
- 历史记录模式
- 在 Amazon Redshift 中共享数据
- 监控零 ETL 集成
- 零 ETL 集成的指标
- 修改 DynamoDB 的零 ETL 集成
- 删除 DynamoDB 的零 ETL 集成
- 零 ETL 集成的支持区域
- 零 ETL 集成问题排查

# 将零 ETL 集成与 Amazon Redshift 结合使用时的注意事项

Amazon Redshift 的零 ETL 集成有以下注意事项。

- 您的目标 Amazon Redshift 数据仓库必须满足以下先决条件：
  - 运行 Amazon Redshift Serverless 或 RG 或 RA3 节点类型的预置集群。
  - 已加密（如果使用预置集群）。
  - 已启用区分大小写。
- 如果您删除某个源而该源是 Amazon Redshift 数据仓库的授权集成源，则所有关联的集成都将进入 FAILED 状态。之前复制的所有数据都将保留在 Amazon Redshift 数据库中，并且可以进行查询。
- 目标数据库是只读的。您无法在目标数据库中创建表、视图或实体化视图。但是，您可以在目标数据仓库中的其他表上使用实体化视图。
- 只有在跨数据库查询中使用时才支持实体化视图。有关通过零 ETL 集成复制的数据创建实体化视图的信息，请参阅使用实体化视图查询复制的数据。
- 默认情况下，您只能查询目标数据仓库中处于 Synced 状态的表。要查询处于其他状态的表，请将数据库参数 QUERY_ALL_STATES 设置为 TRUE。有关设置 QUERY_ALL_STATES 的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 CREATE DATABASE 和 ALTER DATABASE。有关数据库状态的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 SVV_INTEGRATION_TABLE_STATE。

- Amazon Redshift 仅接受 UTF-8 字符，因此它可能不支持源中定义的排序规则。排序和比较规则可能有所不同，这最终会改变查询结果。
- 每个 Amazon Redshift 数据仓库目标的零 ETL 集成限制为 50 个。
- 集成源中的表必须具有主键。否则，您的表无法复制到 Amazon Redshift 中的目标数据仓库。

  有关如何向 Amazon Aurora PostgreSQL 添加主键的信息，请参阅 AWS 数据库博客 中的 Handle tables without primary keys while creating Amazon Aurora PostgreSQL zero-ETL integrations with Amazon Redshift。有关如何向 Amazon Aurora MySQL 或 RDS for MySQL 添加主键的信息，请参阅 AWS 数据库博客中的 Handle tables without primary keys while creating Amazon Aurora MySQL or Amazon RDS for MySQL zero-ETL integrations with Amazon Redshift。
- 您可以使用适用于 Amazon 零 ETL 集成的数据筛选，来定义从源 Aurora 数据库集群复制到目标 Amazon Redshift 数据仓库的范围。您可以定义一个或多个筛选条件，选择性地在复制中包括或排除某些表，而不是将所有数据都复制到目标。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 Aurora 与 Amazon Redshift 零 ETL 集成的数据筛选。
- 对于 Aurora PostgreSQL 与 Amazon Redshift 的零 ETL 集成，Amazon Redshift 最多支持 Aurora PostgreSQL 中的 100 个数据库。每个数据库都独立地从源数据库复制到目标数据库。
- 在将数据从事务数据存储复制到 Amazon Redshift 时，零 ETL 集成不支持转换。数据从源数据库按原样复制。不过，您可以在 Amazon Redshift 中对复制的数据应用转换。
- 零 ETL 集成使用并行连接在 Amazon Redshift 中运行。该集成使用通过集成创建数据库的用户的凭证运行。
- 您可以设置 `REFRESH_INTERVAL` 来实现零 ETL 集成，从而控制将数据复制到 Amazon Redshift 的频率。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 CREATE DATABASE 和 ALTER DATABASE。
- 通过与 Amazon DynamoDB 的零 ETL 集成创建 Amazon Redshift 数据库后，数据库状态应从正在创建更改为活动。这会开始将源 DynamoDB 表中的数据复制到目标 Redshift 表，这些表是在目标数据库 (`ddb_rs_customerprofiles_zetl_db`) 的公有架构下创建的。

## 在目标上使用历史记录模式时的注意事项

在目标数据库上使用历史记录模式时，有以下注意事项。有关更多信息，请参阅 历史记录模式。

- 当您从来源中删除某个表时，不会从目标中删除该表，但该表的状态将变为 `DroppedSource`。您可以从 Amazon Redshift 数据库中删除或重命名表。
- 当您从来源中截断表时，将在目标表上执行删除操作。例如，如果来源上的所有记录被截断，则目标列 `_record_is_active` 上的相应记录将更改为 `false`。

- 在目标表上运行 TRUNCATE 表 SQL 时，活动历史记录行将被标记为非活动，并带有相应的时间戳。
- 在将表中的某个行设置为非活动状态时，系统会在短暂延迟（大约 10 分钟）后将其删除。要删除非活动行，请使用查询编辑器 v2 或其他 SQL 客户端，连接到零 ETL 数据库。
- 您只能从已启用历史记录模式的表中删除非活动行。例如，类似于以下内容的 SQL 命令仅删除非活动行。

```
delete from schema.user_table where _record_delete_time <= '2024-09-10 12:34:56'
```

这与如下 SQL 命令等效。

```
delete from schema.user_table where _record_delete_time <= '2024-09-10 12:34:56' and
 _record_is_active = False
```

- 在禁用表的历史记录模式后，所有历史数据将保存到一个名为 `<schema>.<table-name>_historical_<timestamp>` 的表中，并将刷新名为 `<schema>.<table-name>` 的原始表。
- 在使用表筛选器从复制中排除已启用历史记录模式的表时，所有行都将设置为非活动状态，并且表的状态将更改为 `DroppedSource`。有关表筛选器的更多信息，请参阅《Amazon Aurora 用户指南》中的 Aurora 与 Amazon Redshift 零 ETL 集成的数据筛选。
- 只能为处于 `Synced` 状态的表将历史记录模式切换到 `true` 或 `false`。
- 开启历史记录模式的表的实体化视图是在完全重新计算时创建的。

## 零 ETL 集成源是 Aurora 或 Amazon RDS 时的注意事项

Aurora 和 Amazon RDS 与 Amazon Redshift 的零 ETL 集成有以下注意事项。

- 您可为 Aurora 和 RDS for MySQL 零 ETL 集成使用数据筛选，从而定义从源数据库集群复制到目标 Amazon Redshift 数据仓库的范围。您可以定义一个或多个筛选条件，选择性地在复制中包括或排除某些表，而不是将所有数据都复制到目标。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 Aurora 与 Amazon Redshift 零 ETL 集成的数据筛选。
- 集成源中的表必须具有主键。否则，您的表无法复制到 Amazon Redshift 中的目标数据仓库。

  有关如何向 Amazon Aurora PostgreSQL 添加主键的信息，请参阅 AWS 数据库博客 中的 Handle tables without primary keys while creating Amazon Aurora PostgreSQL zero-ETL integrations with Amazon Redshift。有关如何向 Amazon Aurora MySQL 或 RDS for MySQL 添加主键的信息，请参

阅 AWS 数据库博客中的 Handle tables without primary keys while creating Amazon Aurora MySQL or Amazon RDS for MySQL zero-ETL integrations with Amazon Redshift。

- Amazon Redshift VARCHAR 数据类型的最大长度为 65535 个字节。当来自该源的内容不符合此限制时，复制将无法继续，系统会将表置于失败状态。您可以将数据库参数 TRUNCATECOLUMNS 设置为 TRUE 来截断内容以适合列。有关设置 TRUNCATECOLUMNS 的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 CREATE DATABASE 和 ALTER DATABASE。

  有关零 ETL 集成源和 Amazon Redshift 数据库之间的数据类型差异的更多信息，请参阅《Amazon Aurora 用户指南》中的 Aurora 和 Amazon Redshift 之间的数据类型差异。

对于 Aurora 源，另请参阅《Amazon Aurora 用户指南》中的限制。

对于 Amazon RDS 源，另请参阅《Amazon RDS 用户指南》中的限制。

## 零 ETL 集成源是 DynamoDB 时的注意事项

DynamoDB 与 Amazon Redshift 的零 ETL 集成有以下注意事项。

- 不支持 DynamoDB 中超过 127 个字符的表名。
- 来自 DynamoDB 零 ETL 集成的数据映射到 Amazon Redshift 中的 SUPER 数据类型列。
- 不支持超过 127 个字符的分区键或排序键列名。
- DynamoDB 的零 ETL 集成只能映射到一个 Amazon Redshift 数据库。
- 对于分区键和排序键，最大精度和小数位数为 (38,18)。DynamoDB 中的数字数据类型支持的最大精度高达 38。Amazon Redshift 也支持 38 的最大精度，但 Amazon Redshift 中的默认小数精度/位数为 (38,10)。这意味着小数位数值会被截断。
- 要成功实现零 ETL 集成，DynamoDB 项中的单个属性（由名称+值组成）不得大于 64 KB。
- 激活后，零 ETL 集成会导出整个 DynamoDB 表来填充 Amazon Redshift 数据库。完成这个初始过程所花的时间取决于 DynamoDB 表大小。然后，零 ETL 集成使用 DynamoDB 增量导出功能以增量方式将更新从 DynamoDB 复制到 Amazon Redshift。这意味着 Amazon Redshift 中复制的 DynamoDB 数据会自动保持最新。

  目前，DynamoDB 零 ETL 集成的最小延迟为 15 分钟。您可以通过为零 ETL 集成设置非零 REFRESH_INTERVAL 来进一步增大此延迟。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 CREATE DATABASE 和 ALTER DATABASE。

对于 Amazon DynamoDB 源，另请参阅《Amazon DynamoDB 开发人员指南》中的 Prerequisites and limitations。

## 当零 ETL 集成源是应用程序（例如 Salesforce、SAP、ServiceNow 和 Zendesk）时的注意事项

以下注意事项适用于 Amazon Redshift 的源应用程序，例如 Salesforce、SAP、ServiceNow 和 Zendesk。

- 不支持来自应用程序源的表名称和列名称超过 127 个字符。
- Amazon Redshift VARCHAR 数据类型的最大长度为 65535 个字节。当来自该源的内容不符合此限制时，复制将无法继续，系统会将表置于失败状态。您可以将数据库参数 TRUNCATECOLUMNS 设置为 TRUE 来截断内容以适合列。有关设置 TRUNCATECOLUMNS 的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 CREATE DATABASE 和 ALTER DATABASE。

  有关零 ETL 集成应用程序源和 Amazon Redshift 数据库之间的数据类型差异的更多信息，请参阅《AWS Glue 开发人员指南》中的零 ETL 集成。
- 与应用程序的零 ETL 集成的最小延迟为 1 小时。您可以通过为零 ETL 集成设置非零 REFRESH_INTERVAL 来进一步增大此延迟。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 CREATE DATABASE 和 ALTER DATABASE。

有关与应用程序的零 ETL 集成的源，另请参阅《AWS Glue 开发人员指南》中的零 ETL 集成。

## 还原零 ETL 集成目标时的注意事项

当您从快照或恢复点将 Amazon Redshift Serverless 命名空间还原到同一个无服务器命名空间时，将自动保留与该命名空间关联的零 ETL 集成。还原后会启动完全重新同步，以确保目标 Amazon Redshift 数据库中的数据与源一致。请注意以下事项：

- 将快照还原到其它命名空间不会保留集成。
- 还原后，Amazon Redshift 会对所有保留的零 ETL 集成执行完全重新同步。在重新同步期间，数据摄取性能可能会暂时降低。
- 如果您在目标数据库上使用历史记录模式，则不会捕获从拍摄快照直至完全重新同步完成这段时间内的记录版本。重新同步后，历史记录模式恢复正常运行。此行为与任何其它完全重新同步相同。有关更多信息，请参阅 历史记录模式。

- 如果在拍摄快照后创建了零 ETL 集成，则该集成将在还原后进入 NEEDS_ATTENTION 状态，因为快照中不存在相应的数据库。要解决此问题，您可以从包含该集成的最新快照中进行还原，也可以删除集成。
- 如果在拍摄快照后删除了零 ETL 集成，则不会还原该集成。快照中的数据库将保留，但可以将其删除。
- 如果在拍摄快照后更改了表级筛选条件，则还原的目标将使用集成中的当前筛选条件，而不是创建快照时的筛选条件。
- 要选择退出在还原期间保留该集成，请取消选中保留集成复选框（位于 AWS 管理控制台的还原页面上）；或者，如果您使用的是 AWS CLI，请在调用 restore-from-snapshot 或 restore-from-recovery-point API 操作时设置 --no-maintain-integration 参数。当您选择退出时，集成将在还原后进入 FAILED 状态。您可以删除集成并重新创建它们。
- 仅当还原到相同的无服务器命名空间时，此功能才适用于 Amazon Redshift Serverless。预调配集群上的快照还原不会保留零 ETL 集成。

## 调整零 ETL 集成目标的大小时的注意事项

当您对作为零 ETL 集成的目标的 Amazon Redshift 预调配集群进行大小调整时，调整大小可能会导致集成表重新同步。是否发生重新同步取决于调整大小的类型：

- 仅更改节点数量的弹性调整大小（就地调整大小）不会影响零 ETL 集成。表保持同步。
- 更改节点类型的弹性调整大小会导致集群上零 ETL 集成中的所有表重新同步。任何经典调整大小操作也会触发重新同步。之所以发生这种情况，是因为这些调整大小操作在服务将数据重新分配到新集群配置时，会暂时更改表的分配方式。有关更多信息，请参阅调整集群大小。

当调整大小触发重新同步时，请注意以下事项：

- 集成保持活动状态，调整大小完成后将自动开始重新同步。您不需要采取任何措施。
- 在表重新同步时，您无法在 Amazon Redshift 中查询该表。要在重新同步期间保持表的可查询性，请在开始调整大小之前，在目标数据库上将 QUERY_ALL_STATES 参数设置为 TRUE。在重新同步完成之前，重新同步期间返回的数据可能会过时。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 CREATE DATABASE 和 ALTER DATABASE。
- 重新同步可能需要 20–25 分钟或更长时间，具体取决于源数据库的大小。
- 您可以使用 SVV_INTEGRATION_TABLE_STATE 系统视图监控集成表的状态。在重新同步完成且表返回到 Synced 之前，表将显示 ResyncRequired 或 ResyncInitiated 状态。

# 开始使用零 ETL 集成

这组任务将为您演练如何设置第一个零 ETL 集成。首先，请配置集成源，并使用所需的参数和权限对它进行设置。然后，继续从 Amazon Redshift 控制台或 AWS CLI 执行初始设置的剩余部分。控制台提供了自动修复此问题选项，用于更正一些配置问题。

主题


- 创建和配置目标 Amazon Redshift 数据仓库
- 为您的数据仓库开启区分大小写
- 为您的 Amazon Redshift 数据仓库配置授权
- 创建零 ETL 集成


## 创建和配置目标 Amazon Redshift 数据仓库

在此步骤中，您将创建并配置目标 Amazon Redshift 数据仓库，例如 Redshift Serverless 工作组或预置集群。如果您已将 Amazon Redshift 数据仓库配置为与零 ETL 集成配合使用，则可跳过这一步。

您的目标数据仓库必须具有以下特征：

- 运行 Amazon Redshift Serverless 或 RG 或 RA3 节点类型的预置集群。
- 区分大小写（`enable_case_sensitive_identifier`）功能已开启。有关更多信息，请参阅 为您的数据仓库开启区分大小写。
- 如果您的目标数据仓库是 Amazon Redshift 预置集群，则应加密。有关更多信息，请参阅 Amazon Redshift 数据库加密。
- 在与集成源相同的 AWS 区域中创建。

要为零 ETL 集成创建目标数据仓库，请根据您的部署类型参阅以下主题之一：

- 要创建 Amazon Redshift 预置集群，请参阅创建集群。
- 要创建预览版 Amazon Redshift Serverless 工作组及命名空间，请参阅创建带有命名空间的工作组。

当您创建预置集群时，Amazon Redshift 还会创建默认参数组。您无法编辑默认参数组。但是，您可以在创建新集群之前创建自定义参数组，然后将其与集群关联。您也可以编辑将与创建的集群关联的参

数组。在创建自定义参数组或编辑现有参数组以使用零 ETL 集成时，您还必须为参数组开启区分大小写。

要使用 Amazon Redshift 控制台或 AWS CLI 创建自定义参数组，请参阅 Creating a parameter group。

## 为您的数据仓库开启区分大小写

您可以在创建过程中附加参数组，并为预置集群开启区分大小写。但是，只有在创建之后，您才能通过 AWS Command Line Interface (AWS CLI) 更新无服务器工作组。为了支持源表和列的区分大小写特性，这一步是必需的。`enable_case_sensitive_identifier` 是一个配置值，用于确定数据库、表和列的名称标识符是否区分大小写。必须开启此参数才能在数据仓库中创建零 ETL 集成。有关更多信息，请参阅 enable_case_sensitive_identifier。

对于 Amazon Redshift Serverless – 使用 AWS CLI 为 Amazon Redshift Serverless 开启区分大小写。请注意，您只能从 AWS CLI 为 Amazon Redshift Serverless 开启区分大小写。

对于 Amazon Redshift 预置集群，请使用以下主题之一为目标集群启用区分大小写：

- 使用 Amazon Redshift 控制台为 Amazon Redshift 预置集群开启区分大小写
- 使用 AWS CLI 为 Amazon Redshift 预置集群开启区分大小写

### 使用 AWS CLI 为 Amazon Redshift Serverless 开启区分大小写

运行以下 AWS CLI 命令为您工作组开启区分大小写。

```
aws redshift-serverless update-workgroup \
        --workgroup-name target-workgroup \
        --config-parameters
 parameterKey=enable_case_sensitive_identifier,parameterValue=true
```

请等待工作组状态变为 `Active`，然后再执行下一步操作。

### 使用 Amazon Redshift 控制台为 Amazon Redshift 预置集群开启区分大小写

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在左侧导航窗格中，选择预置集群控制面板。

3. 选择要将数据复制到的预置集群。
4. 在左侧导航窗格中，选择配置 > 工作负载管理。
5. 在工作负载管理页面中，选择参数组。
6. 选择参数选项卡。
7. 选择编辑参数，然后将 enable_case_sensitive_identifier 更改为 true。
8. 然后，选择保存。

### 使用 AWS CLI 为 Amazon Redshift 预置集群开启区分大小写

1. 由于您无法编辑默认参数组，因此请在终端程序中运行以下 AWS CLI 命令来创建自定义参数组。稍后，您将它与预置集群相关联。

```
aws redshift create-cluster-parameter-group \
    --parameter-group-name zero-etl-params \
    --parameter-group-family redshift-2.0 \
    --description "Param group for zero-ETL integrations"
```

2. 运行以下 AWS CLI 命令，为您参数组开启区分大小写。

```
aws redshift modify-cluster-parameter-group \
    --parameter-group-name zero-etl-params \
    --parameters ParameterName=enable_case_sensitive_identifier,ParameterValue=true
```

3. 运行以下命令，将参数值与集群关联。

```
aws redshift modify-cluster \
    --cluster-identifier target-cluster \
    --cluster-parameter-group-name zero-etl-params
```

4. 等待预调配集群变为可用。您也可以使用 `describe-cluster` 命令查看集群的状态。然后，运行以下命令可重启集群。

```
aws redshift reboot-cluster \
    --cluster-identifier target-cluster
```

## 为您的 Amazon Redshift 数据仓库配置授权

要将数据从集成源复制到 Amazon Redshift 数据仓库中，您必须首先添加以下两个实体：

- 已授权的主体 – 标识可以在数据仓库中创建零 ETL 集成的用户或角色。
- 已授权的集成源 – 标识可以更新数据仓库的源数据库。

您可以从 Amazon Redshift 控制台的资源策略选项卡或者使用 Amazon Redshift `PutResourcePolicy` API 操作，配置授权主体和授权集成源。

## 添加已授权的主体

要在您的 Redshift Serverless 工作组或预置集群中创建零 ETL 集成，需要授予对关联命名空间或预置集群的访问权限。

如果满足以下两个条件，则可以跳过此步骤：

- 拥有 Redshift Serverless 工作组或预置集群的 AWS 账户也拥有源数据库。
- 该主体与基于身份的 IAM 策略相关联，该策略使其有权在此 Redshift Serverless 命名空间或预置集群中创建零 ETL 集成。

将已授权的主体添加到 Amazon Redshift Serverless 命名空间

1. 在 Amazon Redshift 控制台的左侧导航窗格中，选择 Redshift Serverless。
2. 选择命名空间配置，选择您的命名空间，然后转到资源策略选项卡。
3. 选择添加已授权的主体。
4. 对于要添加的每个已授权的主体，输入您想要授予访问权限的 AWS 用户或角色的 ARN，或者 AWS 账户的 ID，以便他们能在命名空间中创建零 ETL 集成。账户 ID 存储为 ARN。
5. 选择保存更改。

向 Amazon Redshift 预置集群添加已授权的主体

1. 在 Amazon Redshift 控制台的左侧导航窗格中，选择预置集群控制面板。
2. 选择集群，然后选择集群并转到资源策略选项卡。
3. 选择添加已授权的主体。
4. 对于要添加的每个已授权的主体，输入您想要授予访问权限的 AWS 用户或角色的 ARN，或者 AWS 账户的 ID，以便他们能在集群中创建零 ETL 集成。账户 ID 存储为 ARN。
5. 选择保存更改。

## 添加已授权的集成源

要允许您的源更新 Amazon Redshift 数据仓库，您必须将其作为已授权的集成源添加到命名空间。

向 Amazon Redshift Serverless 命名空间添加已授权的集成源

1. 在 Amazon Redshift 控制台中，转到无服务器控制面板。
2. 选择命名空间的名称。
3. 转到资源策略选项卡。
4. 选择添加已授权的集成源。
5. 指定用于零 ETL 集成的源的 ARN。

> **Note**
>
> 移除已授权的集成源会阻止数据复制到命名空间。此操作会停用从该源到此命名空间的所有零 ETL 集成。

向 Amazon Redshift 预置集群添加已授权的集成源

1. 在 Amazon Redshift 控制台中，转到预置集群控制面板。
2. 选择预置集群的名称。
3. 转到资源策略选项卡。
4. 选择添加已授权的集成源。
5. 指定作为零 ETL 集成的数据来源的源 ARN。

> **Note**
>
> 移除已授权的集成源会阻止数据复制到预置集群。此操作会停用从该源到此 Amazon Redshift 预置集群的所有零 ETL 集成。

## 使用 Amazon Redshift API 配置授权

您可以使用 Amazon Redshift API 操作来配置用于零 ETL 集成的资源策略。

要控制可以在命名空间中创建入站集成的源，请创建资源策略并将其附加到命名空间。使用资源策略，您可以指定有权访问集成的源。资源策略附加到目标数据仓库的命名空间，以允许源创建入站集成，将数据从源中复制到 Amazon Redshift。

以下是资源策略示例。

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "redshift:AuthorizeInboundIntegration",
      "Resource": "arn:aws:redshift-serverless:us-east-1:123456789012:namespace/
cc4ffe56-ad2c-4fd1-a5a2-f29124a56433",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:cluster:foo"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "redshift:CreateInboundIntegration",
      "Resource": "arn:aws:redshift-serverless:us-east-1:123456789012:namespace/
cc4ffe56-ad2c-4fd1-a5a2-f29124a56433"
    }
  ]
}
```

以下总结了可用于为集成配置资源策略的 Amazon Redshift API 操作：

- 使用 PutResourcePolicy API 操作来使资源策略生效。当您提供其他资源策略时，将替换该资源上之前的资源策略。使用前面的资源策略示例，该策略授予执行以下操作的权限：
  - `CreateInboundIntegration` – 允许源主体创建入站集成，以便将数据从源复制到目标数据仓库。

- `AuthorizeInboundIntegration` – 允许 Amazon Redshift 持续验证目标数据仓库是否能够接收从源 ARN 复制的数据。
- 使用 GetResourcePolicy API 操作可查看现有资源策略。
- 使用 DeleteResourcePolicy API 操作可从资源中移除资源策略。

要更新资源策略，您也可以使用 put-resource-policy AWS CLI 命令。例如，要在 DynamoDB 源的 Amazon Redshift 命名空间 ARN 上设置资源策略，请运行类似于以下内容的 AWS CLI 命令。

```
aws redshift put-resource-policy \
--policy file://rs-rp.json \
--resource-arn "arn:aws:redshift-serverless:us-east-1:123456789012:namespace/cc4ffe56-ad2c-4fd1-a5a2-f29124a56433"
```

其中 `rs-rp.json` 包含：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "redshift.amazonaws.com"
            },
            "Action": "redshift:AuthorizeInboundIntegration",
            "Resource": "arn:aws:redshift-serverless:us-east-1:123456789012:namespace/cc4ffe56-ad2c-4fd1-a5a2-f29124a56433",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": "arn:aws:dynamodb:us-east-1:123456789012:table/test_ddb"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:root"
            },
```

```
            "Action": "redshift:CreateInboundIntegration",
            "Resource": "arn:aws:redshift-serverless:us-
east-1:123456789012:namespace/cc4ffe56-ad2c-4fd1-a5a2-f29124a56433"
        }
    ]
}
```

# 创建零 ETL 集成

首先，创建零 ETL 集成，将源数据复制到 Amazon Redshift。

您的数据源确定了要创建哪种类型的零 ETL 集成。

主题

- 创建 Aurora 零 ETL 集成
- 为 Amazon RDS 创建零 ETL 集成
- 创建 DynamoDB 的零 ETL 集成
- 创建与应用程序的零 ETL 集成

## 创建 Aurora 零 ETL 集成

在此步骤中，您将创建 Aurora 与 Amazon Redshift 的零 ETL 集成。

创建 Aurora 与 Amazon Redshift 的零 ETL 集成

1. 从 Amazon RDS 控制台，按照《Amazon Aurora 用户指南》中的说明，创建自定义数据库集群参数组。
2. 从 Amazon RDS 控制台，按照《Amazon Aurora 用户指南》中的说明，创建源 Amazon Aurora 数据库集群。
3. 从 Amazon Redshift 控制台：创建和配置目标 Amazon Redshift 数据仓库。
   - 从 AWS CLI 或 Amazon Redshift 控制台：为您的数据仓库开启区分大小写。
   - 从 Amazon Redshift 控制台：为您的 Amazon Redshift 数据仓库配置授权。
4. 从 Amazon RDS 控制台，按照《Amazon Aurora 用户指南》中的说明，创建零 ETL 集成。
5. 从 Amazon Redshift 控制台或查询编辑器 v2，通过您的集成创建 Amazon Redshift 数据库。

   然后，使用复制的数据查询和创建实体化视图。

有关创建 Aurora 零 ETL 集成的详细信息，请参阅《Amazon Aurora 用户指南》中的创建 Aurora 与 Amazon Redshift 的零 ETL 集成。

## 为 Amazon RDS 创建零 ETL 集成

在此步骤中，将创建 Amazon RDS 与 Amazon Redshift 的零 ETL 集成。Redshift 支持与 RDS for MySQL、RDS for PostgreSQL 和 RDS for Oracle 集成。

创建 Amazon RDS 与 Amazon Redshift 的零 ETL 集成

1. 从 Amazon RDS 控制台，按照《Amazon RDS 用户指南》中的说明，创建自定义数据库参数组。
2. 从 Amazon RDS 控制台，按照《Amazon RDS 用户指南》中的说明，创建源 Amazon RDS 实例。
3. 从 Amazon Redshift 控制台：创建和配置目标 Amazon Redshift 数据仓库。
   - 从 AWS CLI 或 Amazon Redshift 控制台：为您的数据仓库开启区分大小写。
   - 从 Amazon Redshift 控制台：为您的 Amazon Redshift 数据仓库配置授权。
4. 从 Amazon RDS 控制台，按照《Amazon RDS User Guide》中的说明，创建零 ETL 集成。
5. 从 Amazon Redshift 控制台或查询编辑器 v2，通过您的集成创建 Amazon Redshift 数据库。

   然后，使用复制的数据查询和创建实体化视图。

Amazon RDS 控制台提供了分步集成创建流程，在其中您可以指定源数据库和目标 Amazon Redshift 数据仓库。出现问题时，您可以选择让 Amazon RDS 为您修复问题，而无需在 Amazon RDS 或 Amazon Redshift 控制台中手动修复。

有关创建 RDS 零 ETL 集成的详细信息，请参阅《Amazon RDS 用户指南》中的创建 Amazon RDS 与 Amazon Redshift 的零 ETL 集成。

有关专门创建 Amazon RDS for Oracle 零 ETL 集成的详细说明，请参阅《Oracle Database@AWS User Guide》中的 Setting up a zero-ETL integration。

## 创建 DynamoDB 的零 ETL 集成

在创建零 ETL 集成之前，请查看将零 ETL 集成与 Amazon Redshift 结合使用时的注意事项中概述的注意事项和要求。按照这个常规流程创建从 DynamoDB 到 Amazon Redshift 的零 ETL 集成

使用零 ETL 集成将 DynamoDB 数据复制到 Amazon Redshift

1. 确认您的登录凭证具备适当的权限，使您可以处理 Amazon Redshift 与 DynamoDB 的零 ETL 集成。有关示例 IAM 策略，请参阅与 DynamoDB 零 ETL 集成配合使用的 IAM 策略。
2. 在 DynamoDB 控制台中，配置您的 DynamoDB 表，让它具有时间点故障恢复（PITR）、资源策略、基于身份的策略和加密密钥权限，如《Amazon DynamoDB 开发人员指南》中所述。
3. 从 Amazon Redshift 控制台：创建和配置目标 Amazon Redshift 数据仓库。
   - 从 AWS CLI 或 Amazon Redshift 控制台：为您的数据仓库开启区分大小写。
   - 从 Amazon Redshift 控制台：为您的 Amazon Redshift 数据仓库配置授权。
4. 在 Amazon Redshift 控制台中，创建零 ETL 集成，如本主题后面所述。
5. 在 Amazon Redshift 控制台中，在 Amazon Redshift 数据仓库中创建目标数据库。有关更多信息，请参阅 在 Amazon Redshift 中创建目标数据库。
6. 在 Amazon Redshift 控制台中，查询在 Amazon Redshift 数据仓库中复制的数据。有关更多信息，请参阅 在 Amazon Redshift 中查询复制的数据。

在此步骤中，您将创建 Amazon DynamoDB 与 Amazon Redshift 的零 ETL 集成。

Amazon Redshift console

使用 Amazon Redshift 控制台创建 Amazon DynamoDB 与 Amazon Redshift 的零 ETL 集成

1. 在 Amazon Redshift 控制台中，选择零 ETL 集成。在包含零 ETL 集成列表的窗格上，依次选择创建零 ETL 集成、创建 DynamoDB 集成。
2. 在创建集成页面上，输入有关集成的信息，如下所示：
   - 输入集成名称 - 这是一个唯一名称，可用于引用集成。
   - 输入描述 - 描述要从源复制到目标的数据。
   - 选择 DynamoDB 源表 - 可以选择一个 DynamoDB 表。表上必须启用时间点故障恢复（PITR）。仅显示表大小不超过 100 TiB 的表。源 DynamoDB 表必须加密。源还必须具有包含授权主体和集成源的资源策略。如果这些策略不正确，则会显示自动修复此问题选项。
   - 选择目标 Amazon Redshift 数据仓库 - 数据仓库可以是 Amazon Redshift 预置集群或 Redshift Serverless 工作组。如果您的目标 Amazon Redshift 在同一个账户中，则可以选择目标。如果目标位于另一个账户中，则需要指定 Redshift 数据仓库 ARN。目标必须具有包含授权主体和集成源的资源策略，并将 `enable_case_sensitive_identifier` 参数设置为 true。如果目标上没有正确的资源策略，并且您的目标在同一个账户中，则可以选择自

动修复此问题选项，以便在创建集成过程中自动应用资源策略。如果您的目标位于另一个 AWS 账户，则需要手动在 Amazon Redshift 仓库中应用资源策略。如果您的目标 Amazon Redshift 数据仓库没有将正确的参数组选项 `enable_case_sensitive_identifier` 配置为 `true`，则可以选择自动修复此问题选项，在创建集成过程中自动更新此参数组并重启仓库。

- 输入最多 50 个标签键和一个可选值 - 提供有关集成的其他元数据。有关更多信息，请参阅在 Amazon Redshift 中为资源添加标签。
- 选择加密选项 - 加密集成。有关更多信息，请参阅 使用客户自主管理型密钥加密 DynamoDB 集成。

  加密集成时，还可以添加其他加密上下文。有关更多信息，请参阅 加密上下文。

3. 此时会显示一个检查页面，您可以在这个页面选择创建 DynamoDB 集成。
4. 此时会显示一个进度页面，您可以在这个页面查看创建零 ETL 集成的各种任务的进度。
5. 创建集成并激活后，在集成的详细信息页面上，选择连接到数据库。首次创建 Amazon Redshift 数据仓库时，也创建了一个数据库。您需要连接到目标数据仓库中的任何数据库，这样才能为集成创建另一个数据库。在连接到数据库页面中，确定是否可以使用最近的连接并选择身份验证方法。根据您的身份验证方法，输入信息以连接到目标中的现有数据库。此身份验证信息包括现有数据库名称（通常为 `dev`）和使用 Amazon Redshift 数据仓库创建数据库时指定的数据库用户。
6. 连接到数据库后，选择通过集成创建数据库，以便创建从源接收数据的数据库。创建数据库时，您需要提供集成 ID、数据仓库名称和数据库名称。
7. 在集成状态和目标数据库变为 `Active` 后，数据开始从 DynamoDB 表复制到目标表。当您将数据添加到源时，数据会自动复制到目标 Amazon Redshift 数据仓库。

### AWS CLI

要使用 AWS CLI 创建 Amazon DynamoDB 与 Amazon Redshift 的零 ETL 集成，请使用带有以下选项的 `create-integration` 命令：

- `integration-name` – 指定集成的名称。
- `source-arn` - 指定 DynamoDB 源的 ARN。
- `target-arn` - 指定 Amazon Redshift 预置集群或 Redshift Serverless 工作组目标的命名空间 ARN。

以下示例通过提供集成名称、源 ARN 和目标 ARN 来创建集成。集成未加密。

```
aws redshift create-integration \
--integration-name ddb-integration \
--source-arn arn:aws:dynamodb:us-east-1:123456789012:table/books \
--target-arn arn:aws:redshift:us-east-1:123456789012:namespace:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222

{
    "Status": "creating",
    "IntegrationArn": "arn:aws:redshift:us-east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Errors": [],
    "ResponseMetadata": {
        "RetryAttempts": 0,
        "HTTPStatusCode": 200,
        "RequestId": "132cbe27-fd10-4f0a-aacb-b68f10bb2bfb",
        "HTTPHeaders": {
            "x-amzn-requestid": "132cbe27-fd10-4f0a-aacb-b68f10bb2bfb",
            "date": "Sat, 24 Aug 2024 05:44:08 GMT",
            "content-length": "934",
            "content-type": "text/xml"
        }
    },
    "Tags": [],
    "CreateTime": "2024-08-24T05:44:08.573Z",
    "KMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "AdditionalEncryptionContext": {},
    "TargetArn": "arn:aws:redshift:us-east-1:123456789012:namespace:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "IntegrationName": "ddb-integration",
    "SourceArn": "arn:aws:dynamodb:us-east-1:123456789012:table/books"
}
```

以下示例使用客户管理的加密密钥来创建集成。在创建集成之前：

- 在源 DynamoDB 表的同一个账户（在示例中名为“AccountA”）中创建客户管理的密钥（在示例中名为“CMCMK”）。
- 确保使用用户/角色（在示例中名为“RoleA”）来创建对此 KMS 密钥具有 `kms:CreateGrant` 和 `kms:DescribeKey` 权限的集成。
- 将以下项添加到密钥策略中。

```
{
    "Sid": "Enable RoleA to create grants with key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "RoleA-ARN"
    },
    "Action": "kms:CreateGrant",
    "Resource": "*",
    "Condition": {
        // Add "StringEquals" condition if you plan to provide additional encryption
 context
        // for the zero-ETL integration. Ensure that the key-value pairs added here
 match
        // the key-value pair you plan to use while creating the integration.
        // Remove this if you don't plan to use additional encryption context
        "StringEquals": {
            "kms:EncryptionContext:context-key1": "context-value1"
        },
        "ForAllValues:StringEquals": {
            "kms:GrantOperations": [
                "Decrypt",
                "GenerateDataKey",
                "CreateGrant"
            ]
        }
    }
},
{
    "Sid": "Enable RoleA to describe key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "RoleA-ARN"
    },
    "Action": "kms:DescribeKey",
    "Resource": "*"
},
{
    "Sid": "Allow use by RS SP",
    "Effect": "Allow",
    "Principal": {
        "Service": "redshift.amazonaws.com"
           },
    "Action": "kms:CreateGrant",
```

```
    "Resource": "*"
}
```

```
aws redshift create-integration \
--integration-name ddb-integration \
--source-arn arn:aws:dynamodb:us-east-1:123456789012:table/books \
--target-arn arn:aws:redshift:us-east-1:123456789012:namespace:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \
--kms-key-id arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333 \
--additional-encryption-context key33=value33  // This matches the condition in the key policy.
          {
    "IntegrationArn": "arn:aws:redshift:us-east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "IntegrationName": "ddb-integration",
    "SourceArn": "arn:aws:dynamodb:us-east-1:123456789012:table/books",
    "SourceType": "dynamodb",
    "TargetArn": "arn:aws:redshift:us-east-1:123456789012:namespace:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Status": "creating",
    "Errors": [],
    "CreateTime": "2024-10-02T18:29:26.710Z",
    "KMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "AdditionalEncryptionContext": {
        "key33": "value33"
    },
    "Tags": []
}
```

### 与 DynamoDB 零 ETL 集成配合使用的 IAM 策略

在创建零 ETL 集成时，您的登录凭证必须同时拥有 DynamoDB 和 Amazon Redshift 操作的权限，以及拥有集成源和目标的相关资源的权限。以下是演示所需最低权限的示例。

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:ListTables"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:GetResourcePolicy",
                "dynamodb:PutResourcePolicy",
                "dynamodb:UpdateContinuousBackups"
            ],
            "Resource": [
            "arn:aws:dynamodb:us-east-1:111122223333:table/my-ddb-table"
            ]
        },
        {
            "Sid": "AllowRedshiftDescribeIntegration",
            "Effect": "Allow",
            "Action": [
                "redshift:DescribeIntegrations"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowRedshiftCreateIntegration",
            "Effect": "Allow",
            "Action": "redshift:CreateIntegration",
            "Resource": "arn:aws:redshift:us-east-1:111122223333:integration:*"
        },
        {
            "Sid": "AllowRedshiftModifyDeleteIntegration",
            "Effect": "Allow",
            "Action": [
                "redshift:ModifyIntegration",
                "redshift:DeleteIntegration"
            ],
            "Resource": "arn:aws:redshift:us-
east-1:111122223333:integration:<uuid>"
        },
        {
```

```
            "Sid": "AllowRedshiftCreateInboundIntegration",
            "Effect": "Allow",
            "Action": "redshift:CreateInboundIntegration",
            "Resource": "arn:aws:redshift:us-
east-1:111122223333:namespace:<uuid>"
        }
    ]
}
```

使用客户自主管理型密钥加密 DynamoDB 集成

如果您指定自定义 KMS 密钥而不是创建 DynamoDB 零 ETL 集成时的 AWS 拥有的密钥，则密钥策略必须为 Amazon Redshift 服务主体提供对 `CreateGrant` 操作的访问权限。此外，该策略必须让请求者账户或角色有权运行 `DescribeKey` 和 `CreateGrant` 操作。

以下示例密钥策略语句演示了策略中所需的权限。一些示例包括用于进一步缩小权限范围的上下文密钥。

密钥策略语句

以下策略语句让请求者账户或角色可以检索有关 KMS 密钥的信息。

```
{
   "Effect":"Allow",
   "Principal":{
      "AWS":"arn:aws:iam::{account-ID}:role/{role-name}"
   },
   "Action":"kms:DescribeKey",
   "Resource":"*"
}
```

以下策略语句让请求者账户或角色可以添加对 KMS 密钥的授权。[kms:ViaService](#) 条件键将 KMS 密钥的使用限制为来自 Amazon Redshift 的请求。

```
{
   "Effect":"Allow",
   "Principal":{
      "AWS":"arn:aws:iam::{account-ID}:role/{role-name}"
   },
   "Action":"kms:CreateGrant",
   "Resource":"*",
```

```
   "Condition":{
      "StringEquals":{
         "kms:EncryptionContext:{context-key}":"{context-value}",
         "kms:ViaService":"redshift.{region}.amazonaws.com"
      },
      "ForAllValues:StringEquals":{
         "kms:GrantOperations":[
            "Decrypt",
            "GenerateDataKey",
            "CreateGrant"
         ]
      }
   }
}
```

以下策略语句可让 Amazon Redshift 服务主体添加对 KMS 密钥的授权。

```
{
   "Effect":"Allow",
   "Principal":{
      "Service":"redshift.amazonaws.com"
   },
   "Action":"kms:CreateGrant",
   "Resource":"*",
   "Condition":{
      "StringEquals":{
         "kms:EncryptionContext:{context-key}":"{context-value}",
         "aws:SourceAccount":"{account-ID}"
      },
      "ForAllValues:StringEquals":{
         "kms:GrantOperations":[
            "Decrypt",
            "GenerateDataKey",
            "CreateGrant"
         ]
      },
      "ArnLike":{
         "aws:SourceArn":"arn:aws:*:{region}:{account-ID}:integration:*"
      }
   }
}
```

有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的创建密钥策略。

加密上下文

加密零 ETL 集成时，可以将键值对添加为其他加密上下文。您可能想要添加这些键值对，以便添加有关待复制数据的其他上下文信息。有关更多信息，请参阅 AWS Key Management Service 开发人员指南中的加密内容。

除了您添加的任何加密上下文对外，Amazon Redshift 还会添加以下加密上下文对：

- `aws:redshift:integration:arn` - `IntegrationArn`
- `aws:servicename:id` - `Redshift`

这会将您可以添加的加密上下文对总数从 8 减少到 6，并增加授予约束条件的总字符限制。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的 Using grant constraints。

### 创建与应用程序的零 ETL 集成

在此步骤中，您将使用 Amazon Redshift 创建与应用程序的零 ETL 集成。

使用 Amazon Redshift 创建与应用程序的零 ETL 集成

1. 从 Amazon Redshift 控制台：创建和配置目标 Amazon Redshift 数据仓库。
   - 从 AWS CLI 或 Amazon Redshift 控制台：为您的数据仓库开启区分大小写。
   - 从 Amazon Redshift 控制台：为您的 Amazon Redshift 数据仓库配置授权。
2. 从 AWS Glue 控制台：按照《AWS Glue 开发人员指南》中的说明创建集成。
3. 在目标数据库创建完毕并且数据开始复制后，您可以针对复制的数据查询和创建实体化数据。有关更多信息，请参阅 在 Amazon Redshift 中查询复制的数据。

有关创建与应用程序的零 ETL 集成的详细信息，请参阅《AWS Glue 开发人员指南》中的零 ETL 集成。

## 在 Amazon Redshift 中创建目标数据库

要将数据从您的源复制到 Amazon Redshift，您必须通过与 Amazon Redshift 的集成创建数据库。

连接到您的目标 Redshift Serverless 工作组或预置集群，并创建一个引用您的集成标识符的数据库。此标识符是您查询 SVV_INTEGRATION 视图时为 `integration_id` 返回的值。

> **Important**
>
> 在通过集成创建数据库之前，您必须在 Amazon Redshift 控制台上创建了零 ETL 集成，并且其状态必须为 `Active`。

您必须通过与 Amazon Redshift 的集成创建数据库，然后才能开始将数据从源复制到 Amazon Redshift。您可以使用 Amazon Redshift 控制台或查询编辑器 v2 创建数据库。

Amazon Redshift console

1. 在左侧导航窗格中，选择零 ETL 集成。
2. 从集成列表中选择一个集成。
3. 如果您使用的是预置集群，则必须先连接到数据库。选择连接到数据库。您可以使用最近的连接进行连接，也可以通过创建新连接进行连接。
4. 要从集成创建数据库，请选择从集成创建数据库。
5. 输入目标数据库名称。集成 ID 和数据仓库名称已预先填入。

   对于 Aurora PostgreSQL、RDS for PostgreSQL、RDS for Oracle 或 Oracle Database@AWS 源，输入您在创建零 ETL 集成时指定的源命名数据库。在这些情况下，最多可以将 100 个源数据库映射到 Amazon Redshift 数据库。
6. 选择创建数据库。

Amazon Redshift query editor v2

1. 导航到 Amazon Redshift 控制台并选择查询编辑器 v2。
2. 在左侧面板中，选择您的 Amazon Redshift Serverless 工作组或 Amazon Redshift 预置集群并进行连接。
3. 要获取集成 ID，请在 Amazon Redshift 控制台上导航到集成列表。

   或者，运行以下命令来获取 `integration_id` 值：

   ```
   SELECT integration_id FROM SVV_INTEGRATION;
   ```
4. 然后，运行以下命令创建数据库。通过指定集成 ID，您可以在数据库和源之间创建连接。

   用上一个命令返回的值替换 `integration_id`。

```
CREATE DATABASE destination_db_name FROM INTEGRATION 'integration_id';
```

对于 Aurora PostgreSQL、RDS for PostgreSQL、RDS for Oracle 或 Oracle Database@AWS 源，您还必须包括对您在创建集成时指定的源实例或集群中的命名数据库的引用。例如：

```
CREATE DATABASE "destination_db_name" FROM INTEGRATION 'integration_id'
 DATABASE "named_db";
```

有关为零 ETL 集成目标创建数据库的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 CREATE DATABASE。您可以使用 ALTER DATBASE 更改 REFRESH INTERVAL 等数据库参数。有关为零 ETL 集成目标更改数据库的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 ALTER DATABASE。

> **Note**
>
> 只有您的集成源才能更新您通过集成创建的数据库中的数据。要更改表的架构，请对源中的表运行 DDL 或 DML 命令。您可以对源中的表运行 DDL 和 DML 命令，但在目标数据库上只能运行 DDL 命令和只读查询。

有关查看目标数据库状态的信息，请参阅查看零 ETL 集成。

创建目标数据库后，您可以将数据添加到源中。要将数据添加到源中，请参阅以下主题之一：

- 对于 Aurora 源，请参阅《Amazon Aurora 用户指南》中的向源数据库集群添加数据。
- 对于 Amazon RDS 源，请参阅《Amazon RDS User Guide》中的 Add data to the source DB instance。
- 对于 DynamoDB 源，请参阅《Amazon DynamoDB 开发人员指南》中的 DynamoDB 入门。
- 有关与应用程序源的零 ETL 集成，请参阅《AWS Glue 开发人员指南》中的零 ETL 集成。

# 在 Amazon Redshift 中查询复制的数据

将数据添加到源后，这些数据将近乎实时地复制到 Amazon Redshift 数据仓库中，随时可供查询。有关集成指标和表统计信息的信息，请参阅零 ETL 集成的指标。

> **Note**
>
> 由于数据库与 MySQL 中的架构相同，因此 MySQL 数据库级别映射到 Amazon Redshift 架构级别。当您查询从 Aurora MySQL 或 RDS for MySQL 复制的数据时，请注意这种映射差异。

查询复制的数据

1. 导航到 Amazon Redshift 控制台并选择查询编辑器 v2。
2. 连接到您的 Amazon Redshift Serverless 工作组或 Amazon Redshift 预置集群，然后从下拉菜单中选择您的数据库。
3. 使用 SELECT 语句，选择从您在源中创建的架构和表中复制的所有数据。为了区分大小写，请对架构、表和列名使用双引号 (" ")。例如：

   ```
   SELECT * FROM "schema_name"."table_name";
   ```

   您也可以使用 Amazon Redshift 数据 API 查询数据。

## 使用实体化视图查询复制的数据

您可以在本地 Amazon Redshift 数据库中创建实体化视图，以转换通过零 ETL 集成复制的数据。连接到您的本地数据库，并使用跨数据库查询来访问目标数据库。您可以使用完全限定对象名称（采用三部分表示法，即：目标数据库名称.架构名称.表名称），也可以创建引用目标数据库架构对（使用两部分表示法，即：外部架构名称.表名称）。有关跨数据库查询的更多信息，请参阅跨数据库查询数据。

使用以下示例，从源 *tickit_zetl* 创建样本数据并插入到 *sales_zetl* 和 *event_zetl* 表中。这些表复制到 Amazon Redshift 数据库 *zetl_int_db* 中。

```
CREATE TABLE sales_zetl (
        salesid integer NOT NULL primary key,
        eventid integer NOT NULL,
        pricepaid decimal(8, 2)
);

CREATE TABLE event_zetl (
        eventid integer NOT NULL PRIMARY KEY,
        eventname varchar(200)
);
```

```
INSERT INTO sales_zetl VALUES(1, 1, 3.33);
INSERT INTO sales_zetl VALUES(2, 2, 4.44);
INSERT INTO sales_zetl VALUES(3, 2, 5.55);

INSERT INTO event_zetl VALUES(1, "Event 1");
INSERT INTO event_zetl VALUES(2, "Event 2");
```

您可以使用三部分表示法创建一个实体化视图，以获取每场活动的总销售额：

```
--three part notation zetl-database-name.schema-name.table-name
CREATE MATERIALIZED VIEW mv_transformed_sales_per_event_3p
AUTO REFRESH YES
AS
(SELECT eventname, sum(pricepaid) as total_price
FROM  zetl_int_db.tickit_zetl.sales_zetl S, zetl_int_db.tickit_zetl.event_zetl E
WHERE S.eventid = E.eventid
GROUP BY 1);
```

您可以使用两部分表示法创建一个实体化视图，以获取每场活动的总销售额：

```
--two part notation external-schema-name.table-name notation
CREATE EXTERNAL schema ext_tickit_zetl
FROM REDSHIFT
DATABASE zetl_int_db
SCHEMA tickit_zetl;

CREATE MATERIALIZED VIEW mv_transformed_sales_per_event_2p
AUTO REFRESH YES
AS
(
    SELECT eventname, sum(pricepaid) as total_price
    FROM  ext_tickit_zetl.sales_zetl S, ext_tickit_zetl.event_zetl E
    WHERE S.eventid = E.eventid
    GROUP BY 1
);
```

要查看您创建的实体化视图，请使用以下示例。

```
SELECT * FROM mv_transformed_sales_per_event_3p;

+-----------+-------------+
```

```
| eventname | total_price |
+-----------+-------------+
| Event 1   | 3.33        |
| Event 2   | 9.99        |
+-----------+-------------+

SELECT * FROM mv_transformed_sales_per_event_2p;

+-----------+-------------+
| eventname | total_price |
+-----------+-------------+
| Event 1   | 3.33        |
| Event 2   | 9.99        |
+-----------+-------------+
```

## 从 DynamoDB 查询复制的数据

当您将数据从 Amazon DynamoDB 复制到 Amazon Redshift 数据库时，这些数据存储在实体化视图中数据类型为 SUPER 的列中。

在本示例中，以下数据存储在 DynamoDB 中。

```
{
    "key1": {
        "S": "key_1"
    },
    "key2": {
        "N": 0
    },
    "payload": {
        "L": [
            {
                "S": "sale1"
            },
            {
                "S": "sale2"
            },
        ]
    },
}
```

Amazon Redshift 实体化视图的定义如下。

```
CREATE MATERIALIZED VIEW mv_sales
                    BACKUP NO
                    AUTO REFRESH YES
                    AS
                    SELECT "value"."payload"."L"[0]."S"::VARCHAR AS first_payload
                    FROM public.sales;
```

要查看实体化视图中的数据，请运行 SQL 命令。

```
SELECT first_payload FROM mv_sales;
```

# 查看零 ETL 集成

您可以从 Amazon Redshift 控制台查看您的零 ETL 集成。在这里，可以查看其配置信息和当前状态，并打开屏幕来查询和共享数据。

Amazon Redshift console

查看零 ETL 集成的详细信息

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 从左侧导航窗格中，选择无服务器或预置集群控制面板。然后，选择零 ETL 集成。
3. 选择要查看的零 ETL 集成。对于每个集成，请提供以下信息：

   - 集成 ID 是集成创建时返回的标识符。
   - 状态可以是下列项之一：
     - `Active` – 零 ETL 集成正在将事务数据发送到目标 Amazon Redshift 数据仓库。
     - `Syncing` – 零 ETL 集成遇到了可恢复的错误，正在重新设置数据种子。受影响的表在完成重新同步之前无法在 Amazon Redshift 中进行查询。
     - `Failed` – 零 ETL 集成遇到了无法恢复的事件或错误，无法修复。您必须删除并重新创建零 ETL 集成。
     - `Creating` – 正在创建零 ETL 集成。
     - `Deleting` – 正在删除零 ETL 集成。
     - `Needs attention` – 零 ETL 集成遇到了需要手动干预才能解决的事件或错误。要修复此问题，请按照错误消息中的步骤操作。

- 源类型是复制到目标的源数据的类型。类型可以指定其它数据库管理器，例如 Aurora MySQL 兼容版、Amazon Aurora PostgreSQL、RDS for MySQL 和通过应用程序（`GlueSAAS`）。
- 源 ARN 是源数据的 ARN。对于大多数源，这是源数据库或表的 ARN。对于与应用程序源的零 ETL 集成，这是 AWS Glue 连接对象的 ARN。
- 目标是接收源数据的 Amazon Redshift 数据仓库的命名空间。
- 数据库可以是以下值之一：
  - `No database` – 没有用于集成的目标数据库。
  - `Creating` – Amazon Redshift 正在为集成创建目标数据库。
  - `Active` – 数据正在从集成源复制到 Amazon Redshift。
  - `Error` – 集成出现错误。
  - `Recovering` – 数据仓库重新启动后，集成正在恢复。
  - `Resyncing` – Amazon Redshift 正在重新同步集成中的表。
- 目标类型是 Amazon Redshift 数据仓库的类型。
- 创建日期是创建集成的日期和时间 (UTC)。

> **Note**
>
> 要查看数据仓库的集成详细信息，请选择已配置集群或无服务器命名空间的详细信息页面，然后选择零 ETL 集成选项卡。

从零 ETL 集成列表中，您可以选择查询数据以跳转至 Amazon Redshift 查询编辑器 v2。Amazon Redshift 目标数据库启用了 enable_case_sensitive_identifier 参数。编写 SQL 时，您可能需要用双引号 ("<name>") 将架构、表和列名括起来。有关在 Amazon Redshift 数据仓库中查询数据的更多信息，请参阅使用查询编辑器 v2 查询数据库。

在零 ETL 集成列表中，您可以选择共享数据来创建数据共享。要为 Amazon Redshift 数据库创建数据共享，请按照创建数据共享页面上的说明进行操作。在共享 Amazon Redshift 数据库中的数据之前，您必须先创建目标数据库。有关数据共享的更多信息，请参阅 Amazon Redshift 的数据共享概念。

要刷新集成，可以使用 ALTER DATABASE 命令。这样做会将集成源中的所有数据复制到目标数据库中。以下示例刷新零 ETL 集成中所有已同步和失败的表。

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH ALL tables;
```

AWS CLI

要使用 AWS CLI 描述 Amazon DynamoDB 与 Amazon Redshift 的零 ETL 集成，请使用带有以下选项的 `describe-integrations` 命令：

- `integration-arn` - 指定要描述的 DynamoDB 集成的 ARN。
- `integration-name` - 指定可选的筛选条件，从而指定要返回的一个或多个资源。

以下示例通过提供集成 ARN 来描述集成。

```
aws redshift describe-integrations

{
    "Integrations": [
        {
            "Status": "failed",
            "IntegrationArn": "arn:aws:redshift:us-
east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Errors": [
                {
                    "ErrorCode": "INVALID_TABLE_PERMISSIONS",
                    "ErrorMessage": "Redshift does not have sufficient access on the
 table key. Refer to the Amazon DynamoDB Developer Guide."
                }
            ],
            "Tags": [],
            "CreateTime": "2023-11-09T00:32:46.444Z",
            "KMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE33333",
            "TargetArn": "arn:aws:redshift:us-
east-1:123456789012:namespace:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
            "IntegrationName": "ddb-to-provisioned-02",
            "SourceArn": "arn:aws:dynamodb:us-east-1:123456789012:table/mytable"
        }
    ]
}
```

您也可以按 integration-arn、source-arn、source-types 或 status 筛选 describe-integrations 的结果。有关更多信息，请参阅《Amazon Redshift CLI 指南》中的 describe-integrations。

# 历史记录模式

利用历史记录模式，您可以将零 ETL 集成配置为直接在 Amazon Redshift 中跟踪源表中记录的每个版本（包括更新和删除）。您可以对所有数据运行高级分析，例如运行历史分析、生成回顾报告、执行趋势分析以及向基于 Amazon Redshift 构建的下游应用程序发送增量更新。多个 Amazon Redshift 零 ETL 集成支持历史记录模式，包括 Amazon Aurora MySQL、Amazon Aurora PostgreSQL、Amazon RDS for MySQL 和 Amazon DynamoDB。多个应用程序也支持历史记录模式，例如 Salesforce、SAP、ServiceNow 和 Zendesk。

您可以从 Amazon Redshift 控制台 (https://console.aws.amazon.com/redshiftv2/) 启用和禁用零 ETL 集成的历史记录模式。使用历史记录模式跟踪集成来源中已删除或修改的记录。跟踪在目标 Amazon Redshift 数据仓库中执行。启用历史记录模式不会影响针对这些表的常规分析查询的性能。

开启历史记录模式后，您在来源中删除的表将不会在 Amazon Redshift 中删除。相反，表将以 DroppedSource 状态出现，您仍然可以查询这些表。您仍然可以在常规 SQL 中使用 DROP 和 RENAME 命令。

如果您想在来源上重用相同的表名称，则必须先 DROP 或 RENAME 相应的 DroppedState 表，然后才能将其复制到 Amazon Redshift。在来源上创建表之前，请务必先执行此操作。

有关使用历史记录模式时的注意事项的信息，请参阅在目标上使用历史记录模式时的注意事项。

管理零 ETL 集成的历史记录模式

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 从左侧导航窗格中，选择无服务器或预置集群控制面板。然后，选择零 ETL 集成。
3. 选择要管理的零 ETL 集成，然后选择管理历史记录模式。这将显示管理历史记录模式窗口。
4. 您可以为从具有单一源表（例如 Amazon DynamoDB）的来源类型复制的目标表禁用或启用历史记录模式。当零 ETL 集成具有多个可能的目标表时，您可以为所有现有表和将来表禁用、为所有现有表和将来表启用或为单个表管理历史记录模式。创建零 ETL 集成时，默认值为历史记录模式 off。

在将历史记录模式设置为 on 后，会将以下列添加到目标表中以跟踪来源中的更改。历史记录模式 on 会增加每月使用量和成本，因为 Amazon Redshift 不会删除目标表中的任何记录。删除或更改任何源记录都会在目标记录中创建一条新记录，这会增加包含多个记录版本的目标中的总行数。在源表中删除或修改记录时，不会从目标表中删除该记录。您可以通过删除非活动记录来管理目标表。

| 列名称 | 数据类型 | 说明 |
| --- | --- | --- |
| _record_is_active | 布尔值 | 指示目标中的记录当前在源中是否处于活动状态。True 指示记录处于活动状态。 |
| _record_create_time | Timestamp | 源记录处于活动状态时的起始时间（UTC）。 |
| _record_delete_time | Timestamp | 更新或删除源记录时的结束时间（UTC）。 |

通过筛选列 `_record_is_active` 为 false 的记录，可以从历史记录模式表中删除非活动记录。以下 SQL DELETE 命令从 id 列小于或等于 100 的表中删除非活动记录。删除记录后，在自动 vacuum 操作删除运行时，将回收已删除记录的存储空间。

```
DELETE FROM myschema.mytable where not _record_is_active AND id <= 100;
```

在将历史记录模式设置为 `off` 后，Amazon Redshift 会在目标数据库中创建表副本，其中包含活动记录，但不包含添加的历史记录列。Amazon Redshift 将您的表重命名为 *table-name*_historical_*timestamp* 以供您使用。如果您不再需要表的副本，则可以将其删除。可以使用 ALTER TABLE 命令重命名这些表。例如：

```
ALTER TABLE [schema-name.]table-name_historical_timestamp RENAME TO new_table_name;
```

有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 ALTER TABLE。

也可以使用 SQL 命令 CREATE DATABASE 和 ALTER DATABASE 管理历史记录模式。有关如何设置 HISTORY_MODE 的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 CREATE DATABASE 和 ALTER DATABASE。

# 在 Amazon Redshift 中共享数据

将数据添加到源后，它会立即复制到 Amazon Redshift 中，并准备好通过创建数据共享进行共享。

要共享数据，您必须先创建目标数据库。

使用 Amazon Redshift 控制台在 Amazon Redshift Serverless 中共享数据

1. 在 Amazon Redshift 控制台的左侧导航窗格中，选择 Amazon Redshift Serverless > 无服务器控制面板。
2. 从左侧导航窗格中，选择零 ETL 集成。
3. 选择共享数据。
4. 在创建数据共享页面上，按照创建数据共享中的步骤进行操作。

使用 Amazon Redshift 控制台在 Amazon Redshift 预置集群中共享数据

1. 在 Amazon Redshift 控制台的左侧导航窗格中，选择预置集群控制面板。
2. 从左侧导航窗格中，选择零 ETL 集成。
3. 从集成列表中选择一个集成。
4. 在集成详细信息页面上，选择连接到数据库。
5. 在连接到数据库页面，您可以创建新连接，也可以使用最近的连接。确保与目标数据库建立连接。
6. 如果创建新连接，则输入数据库的数据库名称。然后，单击连接。
7. 在集成详细信息页面上，选择共享数据。
8. 在创建数据共享页面上，按照创建数据共享中的步骤进行操作。

# 监控零 ETL 集成

可以通过查询系统视图或使用 Amazon EventBridge 来监控零 ETL 集成。

## 使用 Amazon Redshift 系统视图监控零 ETL 集成

您可以通过在 Amazon Redshift 中查询以下系统视图来监控零 ETL 集成。

- SVV_INTEGRATION 提供有关零 ETL 集成的配置详细信息。
- SYS_INTEGRATION_ACTIVITY 提供有关已完成的零 ETL 集成信息。

- SVV_INTEGRATION_TABLE_MAPPING 提供有关将元数据值从源映射到目标的信息。
- SVV_INTEGRATION_TABLE_STATE 提供有关集成状态的信息。
- SYS_INTEGRATION_TABLE_ACTIVITY 提供有关集成的插入、删除和更新活动的信息。
- SYS_INTEGRATION_TABLE_STATE_CHANGE 提供有关集成的表状态更改日志的信息。

## 监控与 Amazon EventBridge 的零 ETL 集成

Amazon Redshift 会将集成相关的事件发送到 Amazon EventBridge。有关事件及其对应事件 ID 的列表，请参阅使用 Amazon EventBridge 发送零 ETL 集成事件通知。

# 零 ETL 集成的指标

您可以使用 Amazon Redshift 控制台和 Amazon CloudWatch 中的指标，了解零 ETL 集成的运行状况及性能。您可以调整指标以显示更短或更长持续时间的数据，或者选择在 CloudWatch 中查看指标。要在 Amazon Redshift 控制台上查看集成的指标，请在左侧导航窗格中选择零 ETL 集成，然后选择您的集成 ID。

根据零 ETL 集成的源数据，Amazon Redshift 在集成详细信息页面上提供了集成的指标。可能的指标类型如下：

- 集成指标选项卡中提供了以下图表：

| 指标 | Amazon Redshift 控制台中的指标名称 | 说明 |
| --- | --- | --- |
| `IntegrationLag` | `Lag` | 从数据提交到源的时间，到数据在 Amazon Redshift 中可用于查询的时间之间的延迟。<br>单位：秒<br>维度：`IntegrationId`<br>更新频率：60 秒 |
| `IntegrationNumTablesReplicated` | `Tables replicated` | 从源数据库复制到 Amazon Redshift 的表数量。<br>单位：计数 |

| 指标 | Amazon Redshift 控制台中的指标名称 | 说明 |
| --- | --- | --- |
| | | 维度：IntegrationId<br><br>更新频率：60 秒 |
| IntegrationNumTablesFailedReplication | Tables failed | 复制失败的表的数量。<br><br>单位：计数<br><br>维度：IntegrationId<br><br>更新频率：60 秒 |
| IntegrationDataTransferred | Data transferred | 传输的数据量 (以逻辑字节为单位)。<br><br>单位：字节<br><br>维度：IntegrationId<br><br>更新频率：60 秒 |
| IntegrationLatestDetectedChange | Latest detected change | 集成上次在复制队列中暂存源变更的时间（以 Unix 纪元秒为单位）。<br><br>单位：秒<br><br>维度：IntegrationId<br><br>更新频率：60 秒 |
| IntegrationLatestAppliedChange | Latest applied change | 集成上次在 Amazon Redshift 上完成摄取的时间（以 Unix 纪元秒为单位）。<br><br>单位：秒<br><br>维度：IntegrationId<br><br>更新频率：60 秒 |

| 指标 | Amazon Redshift 控制台中的指标名称 | 说明 |
| --- | --- | --- |
| IntegrationDuplicateRowsDetected | Duplicate rows detected | 在报告间隔内标识出重复行的表的数量。<br>单位：计数<br>维度：IntegrationId<br>更新频率：60 秒 |
| IntegrationAutoRemediationTriggered | Auto-remediation triggered | 在报告间隔内，自动修复移至重新同步的表的数量。<br>单位：计数<br>维度：IntegrationId<br>更新频率：60 秒 |

> **Note**
>
> 对于复制多个数据库的集成，IntegrationLatestDetectedChange 和 IntegrationLatestAppliedChange 指标会报告所有数据库中的最小值。这表示最近更新最少的数据库。每个数据库的值可在 SVV_INTEGRATION 的 latest_detected_change_time 和 latest_applied_change_time 列中找到。

- 在表统计信息选项卡中，您可以查看当前处于活动状态或出现错误的表的列表。此选项卡上的统计信息如下所示（取决于源类型）：
  - 架构名称 – 表所在架构的名称。
  - 表名称 – 源数据库中表的名称。
  - 状态 – 表的状态。可能的值包括 Synced、Failed、Deleted、Resync Required 和 Resync Initiated。
  - 数据库 – 表所在的 Amazon Redshift 数据库。
  - 上次更新时间 – 上次更新表的日期和时间 (UTC)。
  - 表行数 - 表中的行数。
  - 表大小 - 表的大小。

您还可以查看在所选时间段内插入、删除和更新的行数的图表。

# 修改 DynamoDB 的零 ETL 集成

在此步骤中，您将修改 DynamoDB 与 Amazon Redshift 的零 ETL 集成。

Amazon Redshift console

使用 Amazon Redshift 控制台修改 Amazon DynamoDB 与 Amazon Redshift 的零 ETL 集成

1. 在 Amazon Redshift 控制台中，选择零 ETL 集成。在包含零 ETL 集成列表的窗格上，选择要修改的 DynamoDB 集成。
2. 选择编辑并修改集成名称或描述。
3. 选择保存更改以保存您的更改。

AWS CLI

要使用 AWS CLI 修改 Amazon DynamoDB 与 Amazon Redshift 的零 ETL 集成，请使用带有以下选项的 `modify-integration` 命令：

- `integration-arn` - 指定要修改的 DynamoDB 集成的 ARN。
- `integration-name` – 为集成指定新名称。
- `description` – 为集成指定新描述。

以下示例通过提供集成 ARN、新描述和新名称来修改集成。

```
aws redshift modify-integration \
--integration-arn arn:aws:redshift:us-
east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
--description "Test modify description and name together." \
--integration-name "updated-integration-name-2"

{
    "IntegrationArn": "arn:aws:redshift:us-
east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "IntegrationName": "updated-integration-name-2",
    "SourceArn": "arn:aws:dynamodb:us-east-1:123456789012:table/ddb-temp-test-table-
table",
```

```
    "SourceType": "dynamodb",
    "TargetArn": "arn:aws:redshift:us-
east-1:123456789012:namespace:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Status": "active",
    "Errors": [],
    "CreateTime": "2024-09-19T18:06:33.555Z",
    "Description": "Test modify description and name together.",
    "KMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE33333",
    "AdditionalEncryptionContext": {},
    "Tags": []
}
```

# 删除 DynamoDB 的零 ETL 集成

删除集成时，目标数据仓库会保留以前复制的所有数据。您可以继续共享和查询这些数据。但是，源中的新数据不会复制到目标。

在此步骤中，您将删除 DynamoDB 与 Amazon Redshift 的零 ETL 集成。

Amazon Redshift console

使用 Amazon Redshift 控制台删除 Amazon DynamoDB 与 Amazon Redshift 的零 ETL 集成

1. 在 Amazon Redshift 控制台中，选择零 ETL 集成。在包含零 ETL 集成列表的窗格上，选择要删除的 DynamoDB 集成。
2. 选择删除并提供所需的信息。
3. 选择删除来删除零 ETL 集成。

AWS CLI

要使用 AWS CLI 删除 Amazon DynamoDB 与 Amazon Redshift 的零 ETL 集成，请使用带有以下选项的 `delete-integration` 命令：

- `integration-arn` - 指定要删除的 DynamoDB 集成的 ARN。

以下示例通过提供集成 ARN 来删除集成。

```
aws redshift delete-integration \
```

```
--integration-arn arn:aws:redshift:us-
east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

{
    "IntegrationArn": "arn:aws:redshift:us-
east-1:123456789012:integration:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "IntegrationName": "updated-integration-name-2",
    "SourceArn": "arn:aws:dynamodb:us-east-1:123456789012:table/tidal-ddb-ddb-temp-
test-table-table",
    "SourceType": "dynamodb",
    "TargetArn": "arn:aws:redshift:us-
east-1:123456789012:namespace:a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Status": "deleting",
    "Errors": [],
    "CreateTime": "2024-09-19T18:06:33.555Z",
    "Description": "Test modify description and name together.",
    "KMSKeyId": "arn:aws:kms:us-east-1:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:key/
a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "AdditionalEncryptionContext": {},
    "Tags": []
}
```

# 零 ETL 集成的支持区域

零 ETL 集成是一种完全托管式解决方案，可在 Amazon Redshift 中提供来自多个运营和事务性来源以及企业应用程序的事务和运营数据。本页列出了每个支持的来源的可用区域。

## Aurora MySQL

以下区域和引擎版本可用于 Aurora MySQL 与 Amazon Redshift 的零 ETL 集成。

| 区域 | Aurora MySQL |
| --- | --- |
| 非洲（开普敦） | 可用 |
| 亚太地区（香港） | 可用 |
| 亚太地区（东京） | 可用 |
| 亚太地区（首尔） | 可用 |

| 区域 | Aurora MySQL |
| --- | --- |
| 亚太地区（大阪） | 可用 |
| 亚太地区（孟买） | 可用 |
| 亚太地区（海得拉巴） | 可用 |
| 亚太地区（新加坡） | 可用 |
| 亚太地区（悉尼） | 可用 |
| 亚太地区（雅加达） | 可用 |
| 亚太地区（墨尔本） | 可用 |
| 亚太地区（马来西亚） | 不可用 |
| 加拿大（中部） | 可用 |
| 加拿大西部（卡尔加里） | 可用 |
| 中国（北京） | 可用 |
| 中国（宁夏） | 可用 |
| 欧洲地区（法兰克福） | 可用 |
| 欧洲（苏黎世） | 可用 |
| 欧洲地区（斯德哥尔摩） | 可用 |
| 欧洲地区（米兰） | 可用 |
| 欧洲（西班牙） | 可用 |
| 欧洲地区（爱尔兰） | 可用 |
| 欧洲地区（伦敦） | 可用 |
| 欧洲地区（巴黎） | 可用 |

| 区域 | Aurora MySQL |
| --- | --- |
| 以色列（特拉维夫） | 可用 |
| 中东（阿联酋） | 可用 |
| 中东（巴林） | 可用 |
| 南美洲（圣保罗） | 可用 |
| 美国东部（弗吉尼亚州北部） | 可用 |
| 美国东部（俄亥俄州） | 可用 |
| 美国西部（北加利福尼亚） | 可用 |
| 美国西部（俄勒冈州） | 可用 |
| AWS GovCloud (US-East) | 不可用 |
| AWS GovCloud (US-West) | 不可用 |

## Aurora PostgreSQL

以下区域可用于 Aurora PostgreSQL 与 Amazon Redshift 的零 ETL 集成。

| 区域 | Aurora PostgreSQL |
| --- | --- |
| 非洲（开普敦） | 可用 |
| 亚太地区（香港） | 可用 |
| 亚太地区（东京） | 可用 |
| 亚太地区（首尔） | 可用 |
| 亚太地区（大阪） | 可用 |
| 亚太地区（孟买） | 可用 |

| 区域 | Aurora PostgreSQL |
| --- | --- |
| 亚太地区（海得拉巴） | 可用 |
| 亚太地区（新加坡） | 可用 |
| 亚太地区（悉尼） | 可用 |
| 亚太地区（雅加达） | 可用 |
| 亚太地区（墨尔本） | 可用 |
| 亚太地区（马来西亚） | 可用 |
| 加拿大（中部） | 可用 |
| 加拿大西部（卡尔加里） | 可用 |
| 中国（北京） | 可用 |
| 中国（宁夏） | 可用 |
| 欧洲地区（法兰克福） | 可用 |
| 欧洲（苏黎世） | 可用 |
| 欧洲地区（斯德哥尔摩） | 可用 |
| 欧洲地区（米兰） | 可用 |
| 欧洲（西班牙） | 可用 |
| 欧洲地区（爱尔兰） | 可用 |
| 欧洲地区（伦敦） | 可用 |
| 欧洲地区（巴黎） | 可用 |
| 以色列（特拉维夫） | 可用 |
| 中东（阿联酋） | 可用 |

| 区域 | Aurora PostgreSQL |
| --- | --- |
| 中东（巴林） | 可用 |
| 南美洲（圣保罗） | 可用 |
| 美国东部（弗吉尼亚州北部） | 可用 |
| 美国东部（俄亥俄州） | 可用 |
| 美国西部（北加利福尼亚） | 可用 |
| 美国西部（俄勒冈州） | 可用 |
| AWS GovCloud (US-East) | 不可用 |
| AWS GovCloud (US-West) | 不可用 |

## Amazon DynamoDB

以下区域可用于 DynamoDB 与 Amazon Redshift 的零 ETL 集成。

| 区域 | DynamoDB |
| --- | --- |
| 非洲（开普敦） | 可用 |
| 亚太地区（香港） | 可用 |
| 亚太地区（台北） | 可用 |
| 亚太地区（东京） | 可用 |
| 亚太地区（首尔） | 可用 |
| 亚太地区（大阪） | 可用 |
| 亚太地区（孟买） | 可用 |
| 亚太地区（海得拉巴） | 可用 |

| 区域 | DynamoDB |
| --- | --- |
| 亚太地区（新加坡） | 可用 |
| 亚太地区（悉尼） | 可用 |
| 亚太地区（雅加达） | 可用 |
| 亚太地区（墨尔本） | 可用 |
| 亚太地区（马来西亚） | 可用 |
| 亚太地区（泰国） | 可用 |
| 加拿大（中部） | 可用 |
| 加拿大西部（卡尔加里） | 可用 |
| 中国（北京） | 可用 |
| 中国（宁夏） | 可用 |
| 欧洲地区（法兰克福） | 可用 |
| 欧洲（苏黎世） | 可用 |
| 欧洲地区（斯德哥尔摩） | 可用 |
| 欧洲地区（米兰） | 可用 |
| 欧洲（西班牙） | 可用 |
| 欧洲地区（爱尔兰） | 可用 |
| 欧洲地区（伦敦） | 可用 |
| 欧洲地区（巴黎） | 可用 |
| 以色列（特拉维夫） | 可用 |
| 中东（阿联酋） | 可用 |

| 区域 | DynamoDB |
| --- | --- |
| 中东（巴林） | 可用 |
| 墨西哥（中部） | 可用 |
| 南美洲（圣保罗） | 可用 |
| 美国东部（弗吉尼亚州北部） | 可用 |
| 美国东部（俄亥俄州） | 可用 |
| 美国西部（北加利福尼亚） | 可用 |
| 美国西部（俄勒冈州） | 可用 |
| AWS GovCloud (US-East) | 可用 |
| AWS GovCloud (US-West) | 可用 |

## Amazon RDS for MySQL

以下区域可用于 Amazon RDS for MySQL 与 Amazon Redshift 的零 ETL 集成。

| 区域 | RDS for MySQL |
| --- | --- |
| 非洲（开普敦） | 可用 |
| 亚太地区（香港） | 可用 |
| 亚太地区（东京） | 可用 |
| 亚太地区（首尔） | 可用 |
| 亚太地区（大阪） | 可用 |
| 亚太地区（孟买） | 可用 |
| 亚太地区（海得拉巴） | 不可用 |

| 区域 | RDS for MySQL |
| --- | --- |
| 亚太地区（新加坡） | 可用 |
| 亚太地区（悉尼） | 可用 |
| 亚太地区（雅加达） | 不可用 |
| 亚太地区（墨尔本） | 不可用 |
| 亚太地区（马来西亚） | 不可用 |
| 加拿大（中部） | 可用 |
| 加拿大西部（卡尔加里） | 不可用 |
| 中国（北京） | 不可用 |
| 中国（宁夏） | 不可用 |
| 欧洲地区（法兰克福） | 可用 |
| 欧洲（苏黎世） | 不可用 |
| 欧洲地区（斯德哥尔摩） | 可用 |
| 欧洲地区（米兰） | 可用 |
| 欧洲（西班牙） | 不可用 |
| 欧洲地区（爱尔兰） | 可用 |
| 欧洲地区（伦敦） | 可用 |
| 欧洲地区（巴黎） | 可用 |
| 以色列（特拉维夫） | 不可用 |
| 中东（阿联酋） | 不可用 |
| 中东（巴林） | 可用 |

| 区域 | RDS for MySQL |
| --- | --- |
| 南美洲（圣保罗） | 可用 |
| 美国东部（弗吉尼亚州北部） | 可用 |
| 美国东部（俄亥俄州） | 可用 |
| 美国西部（北加利福尼亚） | 可用 |
| 美国西部（俄勒冈州） | 可用 |
| AWS GovCloud (US-East) | 不可用 |
| AWS GovCloud (US-West) | 不可用 |

## 企业应用程序

以下区域可用于企业应用程序与 Amazon Redshift 的零 ETL 集成。

| 区域 | 企业应用程序 |
| --- | --- |
| 非洲（开普敦） | 不可用 |
| 亚太地区（香港） | 可用 |
| 亚太地区（东京） | 可用 |
| 亚太地区（首尔） | 可用 |
| 亚太地区（大阪） | 不可用 |
| 亚太地区（孟买） | 不可用 |
| 亚太地区（海得拉巴） | 不可用 |
| 亚太地区（新加坡） | 可用 |
| 亚太地区（悉尼） | 可用 |

| 区域 | 企业应用程序 |
| --- | --- |
| 亚太地区（雅加达） | 不可用 |
| 亚太地区（墨尔本） | 不可用 |
| 亚太地区（马来西亚） | 不可用 |
| 加拿大（中部） | 可用 |
| 加拿大西部（卡尔加里） | 不可用 |
| 中国（北京） | 不可用 |
| 中国（宁夏） | 不可用 |
| 欧洲地区（法兰克福） | 可用 |
| 欧洲（苏黎世） | 不可用 |
| 欧洲地区（斯德哥尔摩） | 可用 |
| 欧洲地区（米兰） | 不可用 |
| 欧洲（西班牙） | 不可用 |
| 欧洲地区（爱尔兰） | 可用 |
| 欧洲地区（伦敦） | 可用 |
| 欧洲地区（巴黎） | 不可用 |
| 以色列（特拉维夫） | 不可用 |
| 中东（阿联酋） | 不可用 |
| 中东（巴林） | 不可用 |
| 南美洲（圣保罗） | 可用 |
| 美国东部（弗吉尼亚州北部） | 可用 |

| 区域 | 企业应用程序 |
| --- | --- |
| 美国东部（俄亥俄州） | 可用 |
| 美国西部（加利福尼亚北部） | 不可用 |
| 美国西部（俄勒冈州） | 可用 |
| AWS GovCloud (US-East) | 不可用 |
| AWS GovCloud (US-West) | 不可用 |

# 零 ETL 集成问题排查

参阅以下各节，协助排查在处理零 ETL 集成时遇到的问题。

## Aurora MySQL 的零 ETL 集成问题排查

请使用以下信息，排查使用 Aurora MySQL 的零 ETL 集成的常见问题。

主题


- 集成创建失败
- 表没有主键
- Aurora MySQL 表未复制到 Amazon Redshift
- 表中有不支持的数据类型
- 数据操作语言命令失败
- 数据来源之间跟踪的更改不匹配
- 授权失败
- 表的数量超过 100K 或者架构的数量超过 4950
- Amazon Redshift 无法加载数据
- 工作组参数设置不正确
- 没有为激活零 ETL 集成而创建数据库
- 表处于需要重新同步或重新同步已启动状态
- 集成滞后增加

集成创建失败

如果零 ETL 集成创建失败，则集成的状态为 `Inactive`。确保您的源 Aurora 数据库集群的以下内容正确：

- 您在 Amazon RDS 控制台中创建了您的集群。
- 源 Aurora 数据库集群运行的是受支持的版本。有关受支持版本的列表，请参阅支持与 Amazon Redshift 进行零 ETL 集成的区域和 Amazon 数据库引擎。要对此进行验证，请转到集群的配置选项卡并检查引擎版本。
- 您为集群正确配置了二进制日志参数设置。如果您的 Aurora MySQL 二进制日志参数设置不正确或未与源 Aurora 数据库集群关联，则创建会失败。请参阅配置数据库集群参数。

此外，请确保您的 Amazon Redshift 数据仓库在以下方面正确无误：

- 区分大小写已开启。请参阅为您的数据仓库开启区分大小写。
- 您为命名空间添加了正确的以授权的主体和集成源。请参阅为您的 Amazon Redshift 数据仓库配置授权。

表没有主键

在目标数据库中，一个或多个表没有主键且无法同步。

要解决此问题，请转到集成详细信息页面上的表统计数据选项卡，或使用 SVV_INTEGRATION_TABLE_STATE 查看失败的表。您可以向表中添加主键，然后 Amazon Redshift 将重新同步这些表。或者，您可以在 Aurora 上删除这些表，然后创建带主键的表，但不建议使用此方法。有关更多信息，请参阅设计表的 Amazon Redshift 最佳实践。

Aurora MySQL 表未复制到 Amazon Redshift

如果注意到一个或多个表没有在 Amazon Redshift 中显示，可以运行以下命令来重新同步这些表。将 *dbname* 替换为 Amazon Redshift 数据库的名称。同时，将 *table1* 和 *table2* 替换为要同步的表的名称。

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 ALTER DATABASE。

您的数据可能未复制，因为一个或多个源表没有主键。Amazon Redshift 中的监控控制面板将这些表的状态显示为 `Failed`，而总体零 ETL 集成的状态更改为 `Needs attention`。要解决此问题，您可

以在表中确定一个可以成为主键的现有键，也可以添加合成主键。有关详细解决方案，请参阅 AWS 数据库博客 中的 Handle tables without primary keys while creating Amazon Aurora MySQL or RDS for MySQL zero-ETL integrations with Amazon Redshift。

另请确认您的目标是否为 Amazon Redshift 集群，且该集群未处于暂停状态。

表中有不支持的数据类型

您从集成在 Amazon Redshift 中创建了数据库，并从 Aurora 数据库集群将数据复制到该目标数据库，但在该目标数据库中，一个或多个表具有不受支持的数据类型且无法同步。

要解决此问题，请转到集成详细信息页面上的表统计数据选项卡，或使用 SVV_INTEGRATION_TABLE_STATE 查看失败的表。然后，请在 Amazon RDS 中删除这些表并重新创建新表。有关不支持的数据类型的更多信息，请参阅《Amazon Aurora 用户指南》中的 Aurora 和 Amazon Redshift 数据库之间的数据类型差异。

数据操作语言命令失败

Amazon Redshift 无法在 Redshift 表上运行 DML 命令。要解决此问题，请使用 SVV_INTEGRATION_TABLE_STATE 查看失败的表。Amazon Redshift 会自动重新同步表以解决此错误。

数据来源之间跟踪的更改不匹配

当 Amazon Aurora 和 Amazon Redshift 之间的更改不匹配时，就会出现此错误，从而导致集成进入 `Failed` 状态。

要解决此问题，请删除零 ETL 集成，然后在 Amazon RDS 中重新创建。有关更多信息，请参阅 Creating zero-ETL integrations 和 Deleting zero-ETL integrations。

授权失败

如果作为 Amazon Redshift 数据仓库的已授权的集成源的源 Aurora 数据库集群被移除，则会发生授权失败。

要解决此问题，请删除零 ETL 集成，然后在 Amazon RDS 上重新创建。有关更多信息，请参阅 Creating zero-ETL integrations 和 Deleting zero-ETL integrations。

表的数量超过 100K 或者架构的数量超过 4950

对于目标数据仓库，表的数量大于 100K 或架构的数量大于 4950。Amazon Aurora 无法向 Amazon Redshift 发送数据。表和架构的数量超过了设置的限制。要解决此问题，请从源数据库中删除所有不必要的架构或表。

Amazon Redshift 无法加载数据

Amazon Redshift 无法将数据加载到零 ETL 集成。

要解决此问题，请删除 Amazon RDS 上的零 ETL 集成，然后重新创建。有关更多信息，请参阅 Creating zero-ETL integrations 和 Deleting zero-ETL integrations。

工作组参数设置不正确

您的工作组未开启区分大小写功能。

要解决此问题，请转到集成详细信息页面上的属性选项卡，选择参数组，然后从属性选项卡中开启区分大小写的标识符。如果您没有现有的参数组，请在区分大小写标识符处于开启状态的情况下创建一个参数组。然后，在 Amazon RDS 上创建一个新的零 ETL 集成。有关更多信息，请参阅创建零 ETL 集成。

没有为激活零 ETL 集成而创建数据库

没有为零 ETL 集成创建数据库，因此无法激活它。

要解决此问题，请为集成创建数据库。有关更多信息，请参阅 在 Amazon Redshift 中创建目标数据库。

表处于需要重新同步或重新同步已启动状态

您的表处于需要重新同步或重新同步已启动状态。

要收集有关表为何处于该状态的更详细的错误信息，请使用 SYS_LOAD_ERROR_DETAIL 系统视图。

集成滞后增加

如果源数据库中大量使用 SAVEPOINT，则零 ETL 集成的集成延迟可能会增加。

## Aurora PostgreSQL 的零 ETL 集成问题排查

请使用以下信息，排查使用 Aurora PostgreSQL 的零 ETL 集成的常见问题。

主题

- 集成创建失败
- 表没有主键

- Aurora PostgreSQL 表未复制到 Amazon Redshift
- 表中有不支持的数据类型
- 数据操作语言命令失败
- 数据来源之间跟踪的更改不匹配
- 授权失败
- 表的数量超过 100K 或者架构的数量超过 4950
- Amazon Redshift 无法加载数据
- 工作组参数设置不正确
- 没有为激活零 ETL 集成而创建数据库
- 表处于需要重新同步或重新同步已启动状态

集成创建失败

如果零 ETL 集成创建失败，则集成的状态为 `Inactive`。确保您的源 Aurora 数据库集群的以下内容正确：

- 您在 Amazon RDS 控制台中创建了您的集群。
- 您的源 Aurora 数据库集群运行的是受支持的版本。有关受支持版本的列表，请参阅支持与 Amazon Redshift 进行零 ETL 集成的区域和 Amazon 数据库引擎。要对此进行验证，请转到集群的配置选项卡并检查引擎版本。
- 您为集群正确配置了二进制日志参数设置。如果您的 Aurora PostgreSQL 二进制日志参数设置不正确或未与源 Aurora 数据库集群关联，则创建会失败。请参阅配置数据库集群参数。

此外，请确保您的 Amazon Redshift 数据仓库在以下方面正确无误：

- 区分大小写已开启。请参阅为您的数据仓库开启区分大小写。
- 您为 endterm="zero-etl-using.redshift-iam.title"/> 添加了正确的授权主体和集成源。

表没有主键

在目标数据库中，一个或多个表没有主键且无法同步。

要解决此问题，请转到集成详细信息页面上的表统计数据选项卡，或使用 SVV_INTEGRATION_TABLE_STATE 查看失败的表。您可以向表中添加主键，然后 Amazon Redshift

将重新同步这些表。或者，您可以在 Aurora 上删除这些表，然后创建带主键的表，但不建议使用此方法。有关更多信息，请参阅设计表的 Amazon Redshift 最佳实践。

Aurora PostgreSQL 表未复制到 Amazon Redshift

如果注意到一个或多个表没有在 Amazon Redshift 中显示，可以运行以下命令来重新同步这些表。将 *dbname* 替换为 Amazon Redshift 数据库的名称。同时，将 *table1* 和 *table2* 替换为要同步的表的名称。

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 ALTER DATABASE。

您的数据可能未复制，因为一个或多个源表没有主键。Amazon Redshift 中的监控控制面板将这些表的状态显示为 `Failed`，而总体零 ETL 集成的状态更改为 `Needs attention`。要解决此问题，您可以在表中确定一个可以成为主键的现有键，也可以添加合成主键。有关详细解决方案，请参阅 AWS 数据库博客 中的 Handle tables without primary keys while creating Amazon Aurora PostgreSQL zero-ETL integrations with Amazon Redshift。

另请确认您的目标是否为 Amazon Redshift 集群，且该集群未处于暂停状态。

表中有不支持的数据类型

您从集成在 Amazon Redshift 中创建了数据库，并从 Aurora 数据库集群将数据复制到该目标数据库，但在该目标数据库中，一个或多个表具有不受支持的数据类型且无法同步。

要解决此问题，请转到集成详细信息页面上的表统计数据选项卡，或使用 SVV_INTEGRATION_TABLE_STATE 查看失败的表。然后，请在 Amazon RDS 中删除这些表并重新创建新表。有关不支持的数据类型的更多信息，请参阅《Amazon Aurora 用户指南》中的 Aurora 和 Amazon Redshift 数据库之间的数据类型差异。

数据操作语言命令失败

Amazon Redshift 无法在 Redshift 表上运行 DML 命令。要解决此问题，请使用 SVV_INTEGRATION_TABLE_STATE 查看失败的表。Amazon Redshift 会自动重新同步表以解决此错误。

数据来源之间跟踪的更改不匹配

当 Amazon Aurora 和 Amazon Redshift 之间的更改不匹配时，就会出现此错误，从而导致集成进入 `Failed` 状态。

要解决此问题，请删除零 ETL 集成，然后在 Amazon RDS 中重新创建。有关更多信息，请参阅 Creating zero-ETL integrations 和 Deleting zero-ETL integrations。

授权失败

如果作为 Amazon Redshift 数据仓库的已授权的集成源的源 Aurora 数据库集群被移除，则会发生授权失败。

要解决此问题，请删除零 ETL 集成，然后在 Amazon RDS 上重新创建。有关更多信息，请参阅 Creating zero-ETL integrations 和 Deleting zero-ETL integrations。

表的数量超过 100K 或者架构的数量超过 4950

对于目标数据仓库，表的数量大于 100K 或架构的数量大于 4950。Amazon Aurora 无法向 Amazon Redshift 发送数据。表和架构的数量超过了设置的限制。要解决此问题，请从源数据库中删除所有不必要的架构或表。

Amazon Redshift 无法加载数据

Amazon Redshift 无法将数据加载到零 ETL 集成。

要解决此问题，请删除 Amazon RDS 上的零 ETL 集成，然后重新创建。有关更多信息，请参阅 Creating zero-ETL integrations 和 Deleting zero-ETL integrations。

工作组参数设置不正确

您的工作组未开启区分大小写功能。

要解决此问题，请转到集成详细信息页面上的属性选项卡，选择参数组，然后从属性选项卡中开启区分大小写的标识符。如果您没有现有的参数组，请在区分大小写标识符处于开启状态的情况下创建一个参数组。然后，在 Amazon RDS 上创建一个新的零 ETL 集成。有关更多信息，请参阅创建零 ETL 集成。

没有为激活零 ETL 集成而创建数据库

没有为零 ETL 集成创建数据库，因此无法激活它。

要解决此问题，请为集成创建数据库。有关更多信息，请参阅 在 Amazon Redshift 中创建目标数据库。

表处于需要重新同步或重新同步已启动状态

您的表处于需要重新同步或重新同步已启动状态。

要收集有关表为何处于该状态的更详细的错误信息，请使用 SYS_LOAD_ERROR_DETAIL 系统视图。

## RDS for MySQL 的零 ETL 集成问题排查

请使用以下信息，排查使用 RDS for MySQL 的零 ETL 集成的常见问题。

主题


- 集成创建失败
- 表没有主键
- RDS for MySQL 表未复制到 Amazon Redshift
- 表中有不支持的数据类型
- 数据操作语言命令失败
- 数据来源之间跟踪的更改不匹配
- 授权失败
- 表的数量超过 100K 或者架构的数量超过 4950
- Amazon Redshift 无法加载数据
- 工作组参数设置不正确
- 没有为激活零 ETL 集成而创建数据库
- 表处于需要重新同步或重新同步已启动状态


### 集成创建失败

如果零 ETL 集成创建失败，则集成的状态为 `Inactive`。确保您的源 RDS 数据库实例的以下内容正确：

- 您在 Amazon RDS 控制台中创建了实例。
- 源 RDS 数据库实例运行的是受支持的 RDS for MySQL 版本。有关受支持版本的列表，请参阅支持与 Amazon Redshift 进行 Amazon RDS 零 ETL 集成的区域和数据库引擎。要对此进行验证，请转到实例的配置选项卡并检查引擎版本。
- 您为实例正确配置了二进制日志参数设置。如果您的 RDS for MySQL 二进制日志参数设置不正确或未与源 RDS 数据库集群关联，则创建会失败。请参阅配置数据库实例参数。

此外，请确保您的 Amazon Redshift 数据仓库在以下方面正确无误：

- 区分大小写已开启。请参阅为您的数据仓库开启区分大小写。

- 您为命名空间添加了正确的以授权的主体和集成源。请参阅为您的 Amazon Redshift 数据仓库配置授权。

表没有主键

在目标数据库中，一个或多个表没有主键且无法同步。

要解决此问题，请转到集成详细信息页面上的表统计数据选项卡，或使用 SVV_INTEGRATION_TABLE_STATE 查看失败的表。您可以向表中添加主键，然后 Amazon Redshift 将重新同步这些表。或者，您可以在 RDS 上删除这些表，然后创建带主键的表，但不建议使用此方法。有关更多信息，请参阅设计表的 Amazon Redshift 最佳实践。

RDS for MySQL 表未复制到 Amazon Redshift

如果注意到一个或多个表没有在 Amazon Redshift 中显示，可以运行以下命令来重新同步这些表。将 *dbname* 替换为 Amazon Redshift 数据库的名称。同时，将 *table1* 和 *table2* 替换为要同步的表的名称。

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 ALTER DATABASE。

您的数据可能未复制，因为一个或多个源表没有主键。Amazon Redshift 中的监控控制面板将这些表的状态显示为 `Failed`，而总体零 ETL 集成的状态更改为 `Needs attention`。要解决此问题，您可以在表中确定一个可以成为主键的现有键，也可以添加合成主键。有关详细解决方案，请参阅 AWS 数据库博客中的 Handle tables without primary keys while creating Aurora MySQL-Compatible Edition or RDS for MySQL zero-ETL integrations with Amazon Redshift。

另请确认您的目标是否为 Amazon Redshift 集群，且该集群未处于暂停状态。

表中有不支持的数据类型

您从集成在 Amazon Redshift 中创建了数据库，并从 RDS 数据库实例将数据复制到该目标数据库，但在该目标数据库中，一个或多个表具有不受支持的数据类型且无法同步。

要解决此问题，请转到集成详细信息页面上的表统计数据选项卡，或使用 SVV_INTEGRATION_TABLE_STATE 查看失败的表。然后，请在 Amazon RDS 中删除这些表并重新创建新表。有关不支持的数据类型的更多信息，请参阅《Amazon RDS User Guide》中的 Data type differences between RDS and Amazon Redshift databases。

数据操作语言命令失败

Amazon Redshift 无法在 Redshift 表上运行 DML 命令。要解决此问题，请使用 SVV_INTEGRATION_TABLE_STATE 查看失败的表。Amazon Redshift 会自动重新同步表以解决此错误。

数据来源之间跟踪的更改不匹配

当 Amazon Aurora 和 Amazon Redshift 之间的更改不匹配时，就会出现此错误，从而导致集成进入 `Failed` 状态。

要解决此问题，请删除零 ETL 集成，然后在 Amazon RDS 中重新创建。有关更多信息，请参阅 Creating zero-ETL integrations 和 Deleting zero-ETL integrations。

授权失败

由于源 RDS 数据库实例已删除，而该实例作为 Amazon Redshift 数据仓库的已授权的集成源，授权失败。

要解决此问题，请删除零 ETL 集成，然后在 Amazon RDS 上重新创建。有关更多信息，请参阅 Creating zero-ETL integrations 和 Deleting zero-ETL integrations。

表的数量超过 100K 或者架构的数量超过 4950

对于目标数据仓库，表的数量大于 100K 或架构的数量大于 4950。Amazon Aurora 无法向 Amazon Redshift 发送数据。表和架构的数量超过了设置的限制。要解决此问题，请从源数据库中删除所有不必要的架构或表。

Amazon Redshift 无法加载数据

Amazon Redshift 无法将数据加载到零 ETL 集成。

要解决此问题，请删除 Amazon RDS 上的零 ETL 集成，然后重新创建。有关更多信息，请参阅 Creating zero-ETL integrations 和 Deleting zero-ETL integrations。

工作组参数设置不正确

您的工作组未开启区分大小写功能。

要解决此问题，请转到集成详细信息页面上的属性选项卡，选择参数组，然后从属性选项卡中开启区分大小写的标识符。如果您没有现有的参数组，请在区分大小写标识符处于开启状态的情况下创建一个

参数组。然后，在 Amazon RDS 上创建一个新的零 ETL 集成。有关更多信息，请参阅创建零 ETL 集成。

没有为激活零 ETL 集成而创建数据库

没有为零 ETL 集成创建数据库，因此无法激活它。

要解决此问题，请为集成创建数据库。有关更多信息，请参阅 在 Amazon Redshift 中创建目标数据库。

表处于需要重新同步或重新同步已启动状态

您的表处于需要重新同步或重新同步已启动状态。

要收集有关表为何处于该状态的更详细的错误信息，请使用 SYS_LOAD_ERROR_DETAIL 系统视图。

## 排查 DynamoDB 零 ETL 集成问题

请使用以下信息排查 Amazon DynamoDB 零 ETL 集成的常见问题。

主题


- 集成创建失败
- 表中有不支持的数据类型
- 不支持的表名和属性名
- 授权失败
- Amazon Redshift 无法加载数据
- 工作组或集群参数设置不正确
- 没有为激活零 ETL 集成而创建数据库
- 未在源 DynamoDB 表上启用时间点故障恢复（PITR）
- KMS 密钥访问被拒绝
- Amazon Redshift 无权访问 DynamoDB 表键


集成创建失败

如果零 ETL 集成创建失败，则集成的状态为 `Inactive`。确保您的 Amazon Redshift 数据仓库和源 DynamoDB 表在以下方面正确无误：

- 已为您的数据仓库开启区分大小写。请参阅《Amazon Redshift 管理指南》中的开启区分大小写。

- 您为 Amazon Redshift 中的命名空间添加了正确的已授权主体和集成源。请参阅《Amazon Redshift 管理指南》中的为 您的 Amazon Redshift 数据仓库配置授权。
- 您已将基于资源的正确策略添加到源 DynamoDB 表中。请参阅《IAM 用户指南》中的 IAM 中的策略和权限。

表中有不支持的数据类型

在 Amazon Redshift 中，DynamoDB 数字转换为 DECIMAL(38,10)。超过此精度范围的数字会自动转换为 (38,10)。删除集成并统一数字精度，然后重新创建集成。

不支持的表名和属性名

Amazon Redshift 最多支持 127 个字符的表名和属性名。如果长名称（例如 DynamoDB 表名、分区键或排序键列名）使集成失败，请使用更短的名称来修复此问题，然后重新创建集成。

授权失败

当源 DynamoDB 表不再作为 Amazon Redshift 数据仓库的已授权集成源时，授权会失败。

要解决此问题，请删除零 ETL 集成，然后使用 Amazon DynamoDB 重新创建该集成。

Amazon Redshift 无法加载数据

Amazon Redshift 无法从零 ETL 集成加载数据。

要解决此问题，请使用 ALTER DATABASE 刷新集成。

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH ALL TABLES
```

工作组或集群参数设置不正确

您的工作组或集群未开启区分大小写功能。

要解决此问题，请转到集成详细信息页面上的属性选项卡，选择参数组，然后从属性选项卡中开启区分大小写的标识符。如果您没有现有的参数组，请在区分大小写标识符处于开启状态的情况下创建一个参数组。然后，在 DynamoDB 上创建一个新的零 ETL 集成。请参阅《Amazon Redshift 管理指南》中的开启区分大小写。

没有为激活零 ETL 集成而创建数据库

没有为零 ETL 集成创建数据库，因此无法激活它。

要解决此问题，请为集成创建数据库。请参阅《Amazon Redshift 管理指南》中的在 Amazon Redshift 中创建目标数据库。

未在源 DynamoDB 表上启用时间点故障恢复（PITR）

DynamoDB 需要启用 PITR 才能导出数据。确保 PITR 始终处于启用状态。如果您在集成处于活动状态时关闭了 PITR，则需要按照错误消息中的说明进行操作，并使用 ALTER DATABASE 刷新集成。

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH ALL TABLES
```

KMS 密钥访问被拒绝

必须为用于源表或集成的 KMS 密钥配置足够的权限。有关表加密和解密的信息，请参阅《Amazon DynamoDB 开发人员指南》中的 DynamoDB 静态加密。

Amazon Redshift 无权访问 DynamoDB 表键

如果源表加密是 AWS 托管式密钥，则切换到 AWS 拥有的密钥或客户管理的密钥。如果该表已使用客户管理的密钥加密，请确保该策略没有任何条件键。

## 排查与应用程序的零 ETL 集成的问题

使用以下信息，排查与 Salesforce、SAP、ServiceNow 和 Zendesk 等应用程序的零 ETL 集成的常见问题。

主题

- 集成创建失败
- 表未复制到 Amazon Redshift
- 表中有不支持的数据类型
- 工作组参数设置不正确
- 没有为激活零 ETL 集成而创建数据库
- 表处于需要重新同步或重新同步已启动状态

集成创建失败

如果零 ETL 集成创建失败，则集成的状态为 `Inactive`。请确保 Amazon Redshift 数据仓库在以下方面正确无误：

- 区分大小写已开启。请参阅为您的数据仓库开启区分大小写。

- 您为命名空间添加了正确的以授权的主体和集成源。请参阅为您的 Amazon Redshift 数据仓库配置授权。

表未复制到 Amazon Redshift

在目标数据库中，一个或多个表没有主键且无法同步。

要解决此问题，请转到集成详细信息页面上的表统计数据选项卡，或使用 SVV_INTEGRATION_TABLE_STATE 查看失败的表。您可以向表中添加主键，然后 Amazon Redshift 将重新同步这些表。可以运行以下命令来重新同步它们。将 *dbname* 替换为 Amazon Redshift 数据库的名称。同时，将 *table1* 和 *table2* 替换为要同步的表的名称。

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 ALTER DATABASE。

表中有不支持的数据类型

您在 Amazon Redshift 中通过集成创建了数据库，并从与应用程序的零 ETL 集成中将数据复制到该数据库，但一个或多个表具有不受支持的数据类型且无法同步。

要解决此问题，请转到集成详细信息页面上的表统计数据选项卡，或使用 SVV_INTEGRATION_TABLE_STATE 查看失败的表。然后，移除这些表并在源中重新创建新表。有关更多信息，请参阅《AWS Glue 开发人员指南》中的零 ETL 集成。

工作组参数设置不正确

您的工作组未开启区分大小写功能。

要解决此问题，请转到集成详细信息页面上的属性选项卡，选择参数组，然后从属性选项卡中开启区分大小写的标识符。如果您没有现有的参数组，请在区分大小写标识符处于开启状态的情况下创建一个参数组。然后，创建新的零 ETL 集成。有关更多信息，请参阅《AWS Glue 开发人员指南》中的零 ETL 集成。

没有为激活零 ETL 集成而创建数据库

没有为零 ETL 集成创建数据库，因此无法激活它。

要解决此问题，请为集成创建数据库。有关更多信息，请参阅 在 Amazon Redshift 中创建目标数据库。

表处于需要重新同步或重新同步已启动状态

您的表处于需要重新同步或重新同步已启动状态。

要收集有关表为何处于该状态的更详细的错误信息，请使用 SYS_LOAD_ERROR_DETAIL 系统视图。

# 查询数据库

要查询 Amazon Redshift 集群托管的数据库，您有两种选择：

- 连接到您的集群，并使用查询编辑器在 AWS 管理控制台上运行查询。

  如果您在 Amazon Redshift 控制台上使用查询编辑器，则无需下载和设置 SQL 客户端应用程序。
- 通过 SQL 客户端工具（如 SQL Workbench/J）连接到集群。

  Amazon Redshift 支持通过 Java 数据库连接 (JDBC) 和开放式数据库连接 (ODBC) 来连接 SQL 客户端工具。Amazon Redshift 不提供或安装任何 SQL 客户端工具或库，因此您必须将其安装到您的客户端计算机或 Amazon EC2 实例上才能使用它们。您可以使用支持 JDBC 或 ODBC 驱动程序的大多数 SQL 客户端工具。

> **Note**
>
> 在编写存储过程时，我们建议使用最佳实践来保护敏感值：
> 不要在存储过程逻辑中对任何敏感信息进行硬编码。例如，不要在存储过程主体的 CREATE USER 语句中分配用户密码。这会带来安全风险，因为硬编码值可以作为架构元数据记录在目录表中。而是应通过参数将诸如密码之类的敏感值作为参量传递给存储过程。
> 有关存储过程的更多信息，请参阅 CREATE PROCEDURE 和在 Amazon Redshift 中创建存储过程。有关目录表的更多信息，请参阅系统目录表。

## 连接到 Amazon Redshift

您可以使用以下语法连接到数据库。

```
cluster-name.account-number.aws-region.redshift.amazonaws.com/database-name
```

语法元素定义如下。

- `cluster-name`

  您的集群名称。
- `account-number`

在给定 AWS 区域中，与您的 AWS 账号关联的唯一标识符。由给定账户在给定 AWS 区域中创建的所有集群都具有相同的 account-number。

- aws-region

  集群所在的 AWS 区域的代码。

- database-name

  您的数据库名称。

例如，以下连接字符串指定 AWS 区域 us-east-1 的 my-cluster 集群中的 my-db 数据库。

```
my-cluster.123456789012.us-east-1.redshift.amazonaws.com/my-db
```

# 使用查询编辑器 v2 查询数据库

查询编辑器 v2 是一个单独的基于 Web 的 SQL 客户端应用程序，用于在 Amazon Redshift 数据仓库上创作和运行查询。查询编辑器 v2 主要用于编辑和运行查询、可视化结果以及与团队共享您的工作。使用查询编辑器 v2，您可以创建数据库、架构、表和用户定义的函数 (UDF)。在树视图面板中，对于每个数据库，您都可以查看其架构。对于每个架构，您都可以查看其表、视图、UDF 和存储过程。查询编辑器 v2 是以前的查询编辑器的替代品。

> **Note**
>
> 已在商业 AWS 区域提供查询编辑器 v2。有关提供了查询编辑器 v2 的 AWS 区域的列表，请参阅《Amazon Web Services 一般参考》中为 Redshift 查询编辑器 v2 列出的端点。

要了解查询编辑器 v2 的演示，请观看以下视频。Amazon Redshift 查询编辑器 v2。

要了解数据分析的演示，请观看以下视频。使用 Amazon Redshift 查询编辑器 v2 进行数据分析。

有关使用查询编辑器 v2 通过隔离或共享连接运行多个查询的演示，请观看以下视频。使用查询编辑器 v2 执行并发查询。

查询编辑器 v2 具有一组丰富的功能来管理和运行 SQL 语句。以下各部分中的主题可让您使用其中的许多功能。请亲自浏览查询编辑器 v2 以熟悉其功能。

# 配置您的 AWS 账户

可以执行这组任务，来将查询编辑器 v2 配置为查询 Amazon Redshift 数据库。使用适当的权限，您可以访问当前 AWS 区域中您的 AWS 账户拥有的 Amazon Redshift 集群或工作组中的数据。

管理员第一次为您的 AWS 账户配置查询编辑器 v2 时，他们会选择用于加密查询编辑器 v2 资源的 AWS KMS key。默认情况下，AWS 拥有的密钥用于加密资源。或者，管理员可在配置页面中为密钥选择 Amazon 资源名称（ARN）来使用客户托管密钥。

配置账户后，AWS KMS 无法更改加密设置。有关通过查询编辑器 v2 创建和使用客户托管式密钥的更多信息，请参阅创建用于查询编辑器 v2 的 AWS KMS 客户托管式密钥。（可选）管理员还可以选择用于某些功能（例如从文件加载数据）的 S3 桶和路径。有关更多信息，请参阅 从本地文件设置和工作流加载数据。

Amazon Redshift 查询编辑器 v2 支持身份验证、加密、隔离和合规性，以确保静态数据和传输中的数据安全。有关数据安全性和查询编辑器 v2 的更多信息，请参阅以下内容：

- 静态加密
- 传输中加密
- Amazon Redshift 中的配置和漏洞分析

AWS CloudTrail 捕获由您的 AWS 账户 或代表该账户发出的 API 调用和相关事件，并将日志文件传输到您指定的 Amazon S3 存储桶。您可以标识哪些用户和账户调用了 AWS、发出调用的源 IP 地址以及调用的发生时间。要了解有关查询编辑器 v2 如何在 AWS CloudTrail 上运行的更多信息，请参阅使用 Cloudtrail 进行日志记录。有关 CloudTrail 的更多信息，请参阅《AWS CloudTrail 用户指南》https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html。

查询编辑器 v2 对其部分资源具有可调配额。有关更多信息，请参阅Amazon Redshift 对象的配额。

## 使用查询编辑器 v2 创建的资源

在查询编辑器 v2 中，您可以创建资源，例如保存的查询和图表。查询编辑器 v2 中的所有资源都与 IAM 角色或用户关联。我们建议将策略附加到 IAM 角色并将该角色分配给用户。

在查询编辑器 v2 中，您可以添加和删除已保存查询和图表的标签。您可以在设置自定义 IAM 策略或搜索资源时使用这些标签。您还可以使用 AWS Resource Groups 标签编辑器来管理标签。

您可以使用 IAM 策略设置 IAM 角色，以便与 AWS 区域中相同 AWS 账户中的其他人共享查询。

## 创建用于查询编辑器 v2 的 AWS KMS 客户托管式密钥

创建对称加密客户托管式密钥：

使用 AWS KMS 控制台或 AWS KMS API 操作，您可以创建对称加密客户托管式密钥来加密查询编辑器 v2 资源。有关创建密钥的说明，请参阅《AWS Key Management Service Developer Guide》中的 Creating symmetric encryption AWS KMS key。

密钥策略

密钥策略控制对客户托管密钥的访问。每个客户托管式密钥必须只有一个密钥策略，其中包含确定谁可以使用密钥以及如何使用密钥的声明。创建客户托管式密钥时，可以指定密钥策略。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的管理对 AWS KMS 密钥的访问。

要将您的客户托管式密钥与 Amazon Redshift 查询编辑器 v2 一起使用，密钥策略中必须允许以下 API 操作：

- `kms:GenerateDataKey` – 生成唯一的对称数据密钥以加密您的数据。
- `kms:Decrypt` – 解密客户托管式密钥加密的数据。
- `kms:DescribeKey` – 提供客户托管式密钥详细信息以允许验证密钥服务。

以下是 AWS 账户 111122223333 的示例 AWS KMS 策略。在第一部分中，`kms:ViaService` 限制对查询编辑器 v2 服务的密钥使用（在策略中名为 `sqlworkbench.region.amazonaws.com`）。使用密钥的 AWS 账户 必须为 111122223333。在第二部分中，AWS 账户 111122223333的根用户和主要管理员可以访问密钥。

当您创建 AWS 账户 时，最初使用的是一个对所有 AWS 服务和资源拥有完全访问权限的登录身份（称为 AWS 账户根用户）。我们强烈建议不要使用根用户进行日常任务。有关要求根用户凭证的任务，请参阅《IAM 用户指南》中的需要根用户凭证的任务。

JSON

```
{
    "Version":"2012-10-17",
    "Id": "key-consolepolicy",
    "Statement": [
        {
            "Sid": "Allow access to principals authorized to use Amazon Redshift
 Query Editor V2",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt",
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "sqlworkbench.us-east-1.amazonaws.com",
                    "kms:CallerAccount": "111122223333"
                }
            }
        },
        {
            "Sid": "Allow access for key administrators",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "kms:*"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key_ID"
        }
    ]
}
```

以下资源提供有关 AWS KMS 密钥的更多信息：

- 有关 AWS KMS 策略的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的在策略中指定权限。
- 有关 AWS KMS 策略故障排除的信息，请参阅《AWS Key Management Service 开发人员指南》中的密钥访问故障排除。
- 有关密钥的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的 AWS KMS 密钥。

## 访问查询编辑器 v2

如要访问查询编辑器 v2，您需要相应权限。管理员可以将下面的任何一种 AWS 托管式策略附加到该角色以授予权限。（我们建议将策略附加到 IAM 角色并将该角色分配给用户。）这些 AWS 托管式策略使用不同的选项编写，可控制标记资源允许共享查询的方式。您可以使用 IAM 控制台 (https://console.aws.amazon.com/iam/) 附加 IAM 策略。

- AmazonRedshiftQueryEditorV2FullAccess – 授予 Amazon Redshift 查询编辑器 v2 操作和资源的完全访问权限。此策略还授予访问其它所需服务的权限。
- AmazonRedshiftQueryEditorV2NoSharing – 授予在不共享资源的情况下使用 Amazon Redshift 查询编辑器 v2 的权限。此策略还授予访问其它所需服务的权限。
- AmazonRedshiftQueryEditorV2ReadSharing – 授予在共享有限制资源的情况下使用 Amazon Redshift 查询编辑器 v2 的权限。授予主体可以阅读与其团队共享的资源，但无法更新它们。此策略还授予访问其它所需服务的权限。
- AmazonRedshiftQueryEditorV2ReadWriteSharing – 授予在共享资源的情况下使用 Amazon Redshift 查询编辑器 v2 的权限。授予主体可以读取和更新与其团队共享的资源。此策略还授予访问其它所需服务的权限。

您还可以根据提供的托管式策略中允许和拒绝的权限创建您自己的策略。如果您使用 IAM 控制台策略编辑器创建自己的策略，请选择 SQL Workbench 作为您在可视化编辑器中创建策略的服务。查询编辑器 v2 使用可视化编辑器和 IAM policy simulator 中的服务名称 AWS SQL Workbench。

要让主体（分配了 IAM 角色的用户）能够连接到 Amazon Redshift 集群，他们需要其中一个查询编辑器 v2 托管式策略中的权限。他们还需要对集群的 `redshift:GetClusterCredentialsWithIAM` 或 `redshift:GetClusterCredentials` 权限之一。要获得此权限，具有管理权限的人员可以使用临时凭证，将策略附加到用于连接到集群的 IAM 角色。您可以将策略范围限定为特定集群，也可以设为常规范围。有关使用临时凭证的权限的更多信息，请参阅创建有权调用 GetClusterCredentialsWithIAM 或 GetClusterCredentials 的 IAM 角色或用户。

要让主体（通常是分配了 IAM 角色的用户）能够在账户设置页面中为账户中的其他人开启导出结果集的功能，他们需要将 `sqlworkbench:UpdateAccountExportSettings` 权限附加到角色。此权限包含在 `AmazonRedshiftQueryEditorV2FullAccess` AWS 托管式策略中。

随着新功能添加到查询编辑器 v2，AWS 托管式策略将根据需要更新。如果您根据提供的托管式策略中允许和拒绝的权限创建自己的策略，请编辑您的策略以使其与托管式策略的更改保持同步。有关 Amazon Redshift 托管式策略的更多信息，请参阅AWS适用于 Amazon Redshift 的 托管式策略。

要提供访问权限，请为您的用户、组或角色添加权限：

- AWS IAM Identity Center 中的用户和群组：

  创建权限集合。按照《AWS IAM Identity Center 用户指南》中创建权限集的说明进行操作。

- 通过身份提供者在 IAM 中托管的用户：

  创建适用于身份联合验证的角色。按照《IAM 用户指南》中针对第三方身份提供者创建角色（联合身份验证）的说明进行操作。

- IAM 用户：
  - 创建您的用户可以担任的角色。按照《IAM 用户指南》中为 IAM 用户创建角色的说明进行操作。
  - （不推荐使用）将策略直接附加到用户或将用户添加到用户组。按照《IAM 用户指南》中向用户添加权限（控制台）中的说明进行操作。

> **Note**
>
> 如果 AWS IAM Identity Center 管理员删除了整个账户中某个特定权限集的所有权限集关联，则最初与所删除权限集关联的所有查询编辑器资源都将无法再访问。如果以后重新创建了相同的权限，系统还会创建一个新的内部标识符。由于内部标识符已更改，因此以前用户拥有的对查询编辑器资源的访问权限将不再有效。在管理员删除权限集之前，建议权限集的用户应导出查询编辑器资源作为备份。

## 设置主体标签以从查询编辑器 v2 连接到集群或工作组

要使用联合用户选项连接到您的集群或工作组，请使用主体标签设置 IAM 角色或用户。或者，设置身份提供者 (IdP) 以传入 `RedshiftDbUser` 和（可选）`RedshiftDbGroups`。有关使用 IAM 管理标签的更多信息，请参阅《IAM 用户指南》中的在 AWS Security Token Service 中传递会话标签。要使用 AWS Identity and Access Management 设置访问权限，管理员可以使用 IAM 控制台 (https://console.aws.amazon.com/iam/) 添加标签。

将主体标签添加到 IAM 角色中

1. 登录AWS 管理控制台，然后通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在导航窗格中选择 Roles。
3. 使用联合用户选择需要访问查询编辑器 v2 的角色。
4. 选择标签选项卡。

5. 选择管理标签。
6. 选择添加标签，然后在键中输入 RedshiftDbUser，并输入联合用户名的值。
7. （可选）选择添加标签，然后在键中输入 RedshiftDbGroups，并且输入要与用户关联的组名称的值。
8. 选择保存更改以查看与您选择的 IAM 角色关联的标签列表。传播更改可能需要几秒时间。
9. 要使用联合用户，请在传播更改后刷新查询编辑器 v2 页面。

设置身份提供者 (IdP) 以传递主体标签

使用身份提供者 (IdP) 设置标签的过程因 IdP 而异。有关如何将用户和组信息传递给 SAML 属性的说明，请参阅 IdP 文档。正确进行配置后，以下属性将显示在 SAML 响应中，AWS Security Token Service 使用此响应填充 RedshiftDbUser 和 RedshiftDbGroups 的主体标签。

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:RedshiftDbUser">
    <AttributeValue>db-user-name</AttributeValue>
</Attribute>
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:RedshiftDbGroups">
    <AttributeValue>db-groups</AttributeValue>
</Attribute>
```

可选的 *db_groups* 必须是以冒号分隔的列表，例如 group1:group2:group3。

此外，您还可以设置 TransitiveTagKeys 属性以在角色链接过程中保留标签。

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys">
  <AttributeValue>RedshiftDbUser</AttributeValue>
  <AttributeValue>RedshiftDbGroups</AttributeValue>
</Attribute>
```

有关设置查询编辑器 v2 的更多信息，请参阅使用查询编辑器 v2 所需的权限。

> **Note**
>
> 当您使用查询编辑器 v2 的联合用户连接选项连接到集群或工作组时，身份提供者 (IdP) 可以为 RedshiftDbUser 和 RedshiftDbGroups 提供自定义主体标签。目前，AWS IAM Identity Center 不支持将自定义主体标签直接传递给查询编辑器 v2。

# 打开查询编辑器 v2

借助 Amazon Redshift，您可以使用 Amazon Redshift 控制台中的查询编辑器 v2 对数据仓库集群执行 SQL 查询。查询编辑器 v2 是一款基于 Web 的工具，为运行临时查询、探索数据和执行数据分析任务提供了友好的用户界面。以下各个部分将指导您完成在控制台中打开查询编辑器 v2 并有效地利用其功能的过程。

如要打开查询编辑器 v2

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航器菜单中，选择编辑器，然后选择查询编辑器 V2。此时将在新的浏览器标签页中打开查询编辑器 v2。

查询编辑器页面有一个导航器菜单，您可以在其中选择视图，如下所示：

**编辑器**



您可以管理和查询以表形式组织并包含在数据库中的数据。数据库可以包含存储的数据，也可以包含对存储在其他位置（如 Amazon S3）的数据的引用。连接到包含在集群或无服务器工作组中的数据库。

在编辑器视图中工作时，您有以下控件：

- 集群或 工作组字段显示您当前连接到的集群或工作组的名称。数据库字段显示集群或工作组内的数据库。您在数据库视图执行的操作默认会对您选择的数据库执行操作。
- 集群或工作组、数据库和架构的树视图层次结构视图。在架构下，您可以使用表、视图、函数和存储过程。树视图中的每个对象都支持上下文菜单来执行相关操作，例如对对象进行刷新或删除。
- 通过

  创建操作来创建数据库、架构、表和函数。
- 通

  过

  加载数据操作将数据从 Amazon S3 或从本地文件加载到数据库。

- 通过

  保存图标来保存您的查询。

- 通过

  快捷方式图标来显示编辑器的键盘快捷键。

- •••

  更多图标，用于在编辑器中显示更多操作。例如：

  - 与我的团队共享，与您的团队共享查询或笔记本。有关更多信息，请参阅 以团队形式协作和共享。
  - 快捷方式，显示编辑器的键盘快捷键。
  - 选项卡历史记录，用于在编辑器中显示选项卡的选项卡历史记录。
  - 刷新自动完成，用于在编写 SQL 时刷新显示的建议。

- 可以在

  编辑器区域中输入和运行查询。

  运行查询后，结果选项卡随即显示结果。您可以打开此处的图表来可视化您的结果。还可以导出结果。

- 

  笔记本区域，您可以在其中添加各部分，以输入和运行 SQL 或添加 Markdown。

  运行查询后，结果选项卡随即显示结果。您可以在此处导出结果。

查询



查询包含用于管理和查询数据库中数据的 SQL 命令。当您使用查询编辑器 v2 加载示例数据时，它还会为您创建和保存示例查询。

在选择了某个已保存的查询时，您可以使用上下文菜单（右键单击）打开、重命名和删除该查询。您可以选择查询详细信息，查看已保存查询的属性，例如查询 ARN。您还可以查看其版本历史记录、编辑附加到查询的标签，并将其与您的团队共享。

笔记本



SQL 笔记本包含 SQL 和 Markdown 单元格。使用笔记本可在单个文档中组织、注释及共享多个 SQL 命令。

在选择了某个已保存的笔记本时，您可以使用上下文菜单（右键单击）打开、重命名、复制和删除该笔记本。您可以选择笔记本详细信息，查看已保存笔记本的属性，例如笔记本 ARN。您还可以查看其版本历史记录、编辑附加到笔记本的标签，并将其与您的团队共享。有关更多信息，请参阅 Amazon Redshift 中的笔记本。

图表



图表是您的数据的可视化表示。查询编辑器 v2 提供了用于创建多种图表并保存它们的工具。

在选择了某个已保存的图表时，您可以使用上下文菜单（右键单击）打开、重命名和删除该图表。您可以选择图表详细信息，查看已保存图表的属性，例如 图表 ARN。您也可以编辑附加到图表的标签并将其导出。有关更多信息，请参阅 可视化查询结果。

历史记录



查询历史记录是您使用 Amazon Redshift 查询编辑器 v2 运行的查询的列表。这些查询作为单个查询运行，或作为 SQL 笔记本的一部分运行。有关更多信息，请参阅 查看查询和选项卡历史记录。

计划查询



计划查询是设置为在特定时间开始的查询。

所有查询编辑器 v2 视图都有以下图标：

- 

  可视化模式图标，可在亮模式和暗模式之间切换。

- 

  设置图标，可显示不同设置屏幕的菜单。

  - 

    编辑器首选项图标，可在使用查询编辑器 v2 时编辑首选项。在此处，您可以编辑工作区设置以更改字体大小、选项卡大小和其它显示设置。您也可以打开（或关闭）自动完成，以便在输入 SQL 时显示建议。

  - 

    连接图标，可查看编辑器选项卡使用的连接。

    连接用于检索数据库中的数据。连接是针对特定数据库创建的。使用隔离连接时，在一个编辑器选项卡中更改数据库的 SQL 命令（例如创建临时表）的结果在另一个编辑器选项卡中不可见。在查询编辑器 v2 中打开编辑器选项卡时，默认为隔离连接。创建共享连接时，即关闭隔离会话开关，同一数据库的其他共享连接的结果对彼此可见。但是，使用数据库的共享连接的各编辑器选项卡不会并行运行。使用相同连接的查询必须等到连接可用。与一个数据库的连接不能与另一个数据库共享，因此 SQL 结果在不同的数据库连接之间不可见。

    账户中的任何用户可以激活的连接数由查询编辑器 v2 管理员控制。

  - 

    账户设置图标，管理员用于更改账户中所有用户的某些设置。有关更多信息，请参阅 账户设置。

## 使用查询编辑器 v2 时的注意事项

使用查询编辑器 v2 时，请注意以下几点。

- 查询的最长持续时间为 24 小时。
- 查询结果的最大大小为 100 MB。如果调用返回的响应数据超过 100 MB，则返回前 100 MB 数据并显示警告。
- 您可以运行最长为 300,000 个字符的查询。
- 您可以保存最长为 30,000 个字符的查询。
- 默认情况下，查询编辑器 v2 会自动提交所运行的每个 SQL 命令。当提供 BEGIN 语句时，BEGIN-COMMIT 或 BEGIN-ROLLBACK 块中的语句将作为单个事务运行。有关事务的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 BEGIN。

- 运行 SQL 语句时，查询编辑器 v2 显示的最大警告数为 10。例如，运行存储过程时，显示的 RAISE 语句不超过 10 个。
- 查询编辑器 v2 不支持包含逗号 ( , ) 的 IAM RoleSessionName。您可能会看到类似以下的错误：错误消息：“'AROA123456789EXAMPLE:mytext,yourtext'不是 TagValue 的有效值 - 它包含非法字符”。当您定义包含逗号的 IAM RoleSessionName，然后将查询编辑器 v2 与该 IAM 角色一起使用时，就会出现此问题。

  有关 IAM RoleSessionName 的更多信息，请参阅《IAM 用户指南》中的 RoleSessionName SAML 属性。

### 账户设置

拥有正确 IAM 权限的用户可以查看和更改同一 AWS 账户中其他用户的账户设置。该管理员可以查看或设置以下内容：

- 账户中每个用户的最大并发数据库连接数。这包括用于隔离会话的连接。更改此值时，更改可能需要 10 分钟才能生效。
- 允许账户中的用户将整个结果集从 SQL 命令导出到文件中。
- 加载和显示示例数据库以及一些关联的已保存查询。
- 指定账户用户用于从本地文件加载数据的 Amazon S3 路径。
- 查看用于加密查询编辑器 v2 资源的 KMS 密钥 ARN。

## 连接到 Amazon Redshift 数据库

要连接到数据库，请在树视图面板中选择集群或工作组名称。如果出现提示，请输入连接参数。

当您连接到集群或工作组及其数据库时，通常需要提供数据库名称。您还提供以下身份验证方法之一所需的参数：

IAM Identity Center 验证

通过此方法，使用身份提供者 (IdP) 提供的单点登录凭证连接到您的 Amazon Redshift 数据仓库。您必须在 Amazon Redshift 控制台中为 IAM Identity Center 启用集群或工作组。如需帮助设置与 IAM Identity Center 的连接，请参阅 将 Redshift 与 AWS IAM Identity Center 连接来提供单点登录体验。

联合用户

使用此方法，您的 IAM 角色或用户的主体标签必须提供连接详细信息。您可以在 AWS Identity and Access Management 或您的身份提供者 (IdP) 中配置这些标签。查询编辑器 v2 依赖以下标签。

- `RedshiftDbUser` – 此标签定义查询编辑器 v2 使用的数据库用户。此标签为必填项。
- `RedshiftDbGroups` – 此标签定义当连接到查询编辑器 v2 时加入的数据库组。此标签是可选的，其值必须是以冒号分隔的列表，例如 `group1:group2:group3`。空值将被忽略，也就是说，`group1::::group2` 被解释为 `group1:group2`。

这些标签将转发到 `redshift:GetClusterCredentials` API 以获取集群的凭证。有关更多信息，请参阅 设置主体标签以从查询编辑器 v2 连接到集群或工作组。

使用数据库用户名的临时凭证

仅在连接到集群时此选项才可用。使用此方法，查询编辑器 v2 会为数据库提供用户名。查询编辑器 v2 会生成临时密码，以便使用您的数据库用户名连接到数据库。必须允许使用此方法进行连接的用户对 `redshift:GetClusterCredentials` 拥有 IAM 权限。要防止用户使用此方法，请修改其 IAM 用户或角色以拒绝此权限。

使用您的 IAM 身份的临时凭证

仅在连接到集群时此选项才可用。使用此方法，查询编辑器 v2 会将用户名映射到您的 IAM 身份并生成临时密码，以便使用您的 IAM 身份连接到数据库。必须允许使用此方法进行连接的用户对 `redshift:GetClusterCredentialsWithIAM` 拥有 IAM 权限。要防止用户使用此方法，请修改其 IAM 用户或角色以拒绝此权限。

数据库用户名和密码

使用这种方法，还需要为要连接的数据库提供用户名和密码。查询编辑器 v2 代表您创建秘钥并存储在 AWS Secrets Manager 中。此密钥包含用于连接到数据库的凭证。

AWS Secrets Manager

使用此方法，不是提供数据库名称，而是提供在 Secrets Manager 中存储的密钥，其中包含您的数据库和登录凭证。有关创建密钥的信息，请参阅为数据库连接凭证创建密钥。

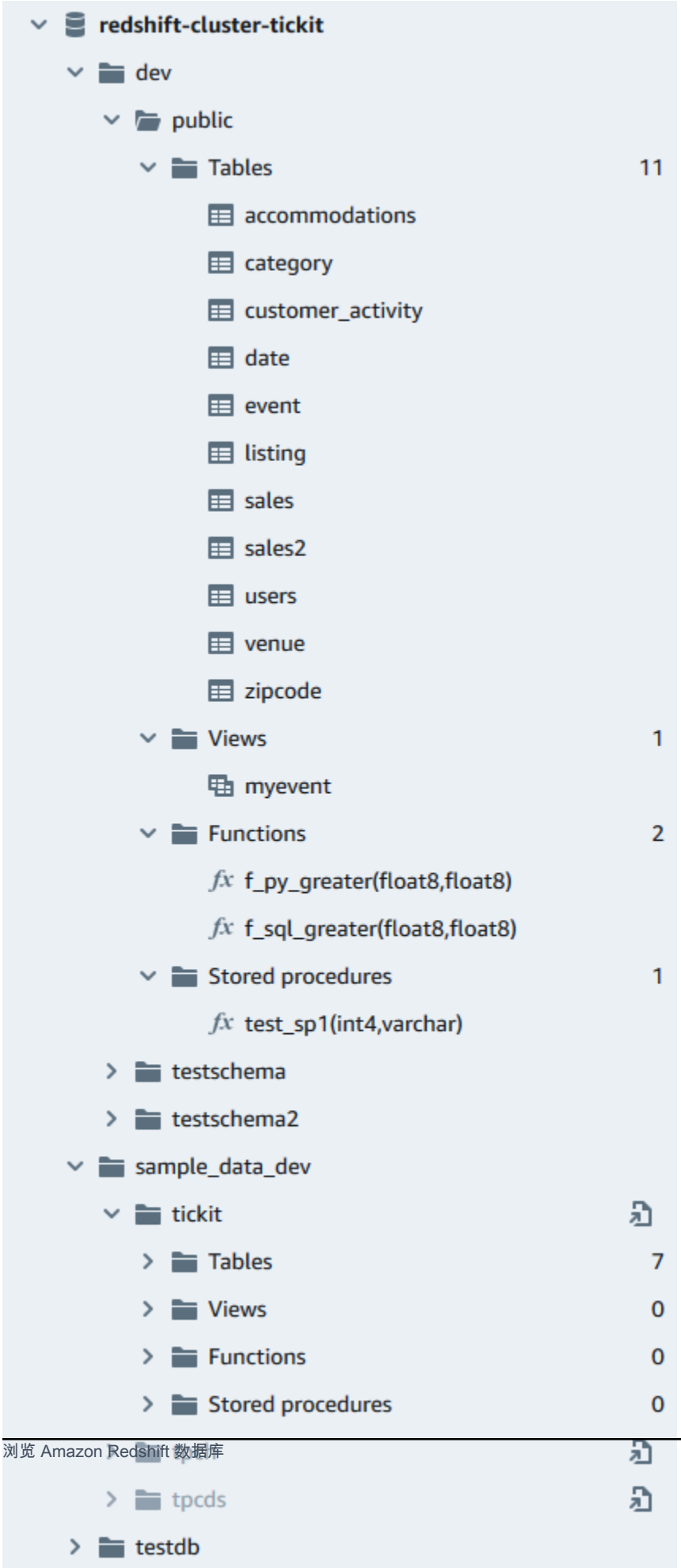
当您选择具有查询编辑器 v2 的集群或工作组时，根据上下文，您可以使用上下文（右键单击）菜单创建、编辑和删除连接。您可以选择连接详细信息，查看连接的属性，例如连接 ARN。您还可以编辑附加到连接的标签。

# 浏览 Amazon Redshift 数据库

在数据库中，您可以在树视图面板中管理架构、表、视图、函数和存储过程。在视图中的每个对象都在上下文（右键单击）菜单中都有与之关联的操作。

分层树状图面板显示数据库对象。要刷新树视图面板以显示可能已
在上次显示树视图后创建的数据库对象，请选择

图标。打开对象的上下文（右键单击）菜单，以查看您可以执行的操作。


redshift-cluster-tickit
dev
public
Tables 11
accommodations
category
customer_activity
date
event
listing
sales
sales2
users
venue
zipcode
Views 1
myevent
Functions 2
f_py_greater(float8,float8)
f_sql_greater(float8,float8)
Stored procedures 1
test_sp1(int4,varchar)
testschema
testschema2
sample_data_dev
tickit
Tables 7
Views 0
Functions 0
Stored procedures 0
tpcds
testdb

选择表后，您可以执行以下操作：

- 如要在编辑器中开启查询表中所有列的 SELECT 语句启动查询，请使用选择表。
- 要查看属性或表格，请使用显示表定义。使用此选项可查看列名、列类型、编码、分配键、排序键以及列是否可以包含空值。有关语法的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》https://docs.aws.amazon.com/redshift/latest/dg/r_CREATE_TABLE_NEW.html中的 CREATE TABLE。
- 要删除表，请使用删除。您可以使用截断表从表中删除所有行。或使用删除表从数据库中删除表。更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 TRUNCATE 和 DROP TABLE。

选择架构以刷新或删除架构。

选择视图以显示视图定义或删除视图。

选择函数以显示函数定义或删除函数。

选择存储过程以显示过程定义或删除过程。

## 创建数据库对象

您可以创建数据库对象，包括数据库、架构、表和用户定义的函数 (UDF)。您必须连接到集群或工作组以及数据库才能创建数据库对象。

### 创建数据库

您可以使用查询编辑器 v2 在集群或工作组中创建数据库。

创建数据库

有关数据库的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 CREATE DATABASE。

1. 选择 创建，然后选择数据库。
2. 输入数据库名称。
3. （可选）选择用户和组，然后选择数据库用户。
4. （可选）您可以从数据共享或 AWS Glue Data Catalog 创建数据库。有关 AWS Glue 的更多信息，请参阅《AWS Glue 开发人员指南》中的什么是 AWS Glue？。

- （可选）选择使用数据共享创建，然后选择选择数据共享。该列表包括可用于在当前集群或工作组中创建使用者数据共享的创建者数据共享。
- （可选）选择使用 AWS Glue Data Catalog 创建，然后选择选择 AWS Glue 数据库。在数据目录架构中，输入在由三部分组成的名称（database.schema.table）中引用数据时将用于架构的名称。

5. 选择创建数据库。

新数据库将在树状视图面板中显示。

当您选择此可选步骤查询从数据共享创建的数据库时，请连接到集群或工作组中的 Amazon Redshift 数据库（例如，默认数据库 dev），并使用三部分表示法（database.schema.table），该表示法引用您在选择使用数据共享创建时创建的数据库名称。数据共享数据库在查询编辑器 v2 编辑器选项卡中列出，但未针对直接连接启用此数据库。

当您选择此可选步骤查询从 AWS Glue Data Catalog 创建的数据库时，请连接到集群或工作组中的 Amazon Redshift 数据库（例如，默认数据库 dev），并使用三部分表示法（database.schema.table），该表示法引用您在选择使用 AWS Glue Data Catalog 创建时创建的数据库名称、您在数据目录架构中命名的架构以及 AWS Glue Data Catalog 中的表。类似于：

```
SELECT * FROM glue-database.glue-schema.glue-table
```

> **Note**
>
> 确认您已使用连接方法使用您的 IAM 身份的临时凭证连接到默认数据库，并且您的 IAM 凭证已被授予 AWS Glue 数据库的使用权限。
>
> ```
> GRANT USAGE ON DATABASE glue-database to "IAM:MyIAMUser"
> ```

AWS Glue 数据库在查询编辑器 v2 编辑器选项卡中列出，但未针对直接连接启用此数据库。

有关查询 AWS Glue Data Catalog 的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的以使用者身份使用 Lake Formation 托管的数据共享和以创建者身份使用 Lake Formation 托管的数据共享。

示例：以数据共享使用者身份创建数据库

以下示例描述了使用查询编辑器 v2 从数据共享创建数据库的特定场景。查看此场景，了解如何从环境中的数据共享创建数据库。此场景使用两个集群，即 cluster-base（创建者集群）和 cluster-view（使用者集群）。

1. 使用 Amazon Redshift 控制台为集群 cluster-base 中的 category2 表创建数据共享。创建者数据共享命名为 datashare_base。

   有关创建数据共享的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的在 Amazon Redshift 中跨集群共享数据。
2. 使用 Amazon Redshift 控制台接受数据共享 datashare_base 作为集群 cluster-view 中 category2 表的使用者。
3. 在查询编辑器 v2 中查看树视图面板，该面板显示了 cluster-base 的层次结构，如下所示：

   - 集群：cluster-base
     - 数据库：dev
       - 架构：public
         - 表：category2
4. 选择  创建，然后选择数据库。
5. 对于数据库名称，请输入 see_datashare_base。
6. 选择使用数据共享创建，然后选择选择数据共享。选择 datashare_base 以用作正在创建的数据库的来源。

   查询编辑器 v2 中的树视图面板显示了 cluster-view 的层次结构，如下所示：

   - 集群：cluster-view
     - 数据库：see_datashare_base
       - 架构：public
         - 表：category2
7. 查询数据时，连接到集群 cluster-view 的默认数据库（通常命名为 dev），但在 SQL 中引用数据共享数据库 see_datashare_base。

> **Note**
>
> 在查询编辑器 v2 编辑器视图中，选定的集群为 cluster-view。选定的数据库为 dev。数据库 see_datashare_base 已列出，但未针对直接连接启用此数据库。您可以在您运行的 SQL 中选择 dev 数据库并引用 see_datashare_base。

```
SELECT * FROM "see_datashare_base"."public"."category2";
```

该查询从集群 cluster_base 中的数据共享 datashare_base 检索数据。

### 从 AWS Glue Data Catalog 创建数据库的示例

以下示例描述了使用查询编辑器 v2 从 AWS Glue Data Catalog 创建数据库的特定场景。查看此场景，了解如何从环境中的 AWS Glue Data Catalog 创建数据库。此场景使用一个集群（即 cluster-view）以包含您创建的数据库。

1. 选择 创建，然后选择数据库。
2. 对于数据库名称，请输入 data_catalog_database。
3. 选择使用 AWS Glue Data Catalog 创建，然后选择选择 AWS Glue 数据库。选择 glue_db 以用作正在创建的数据库的来源。

   选择数据目录架构，然后输入 myschema 作为要在三部分表示法中使用的架构名称。

   查询编辑器 v2 中的树视图面板显示了 cluster-view 的层次结构，如下所示：

   - 集群：cluster-view
     - 数据库：data_catalog_database
       - 架构：myschema
         - 表：category3
4. 查询数据时，连接到集群 cluster-view 的默认数据库（通常命名为 dev），但在 SQL 中引用数据库 data_catalog_database。

> **Note**
>
> 在查询编辑器 v2 编辑器视图中，选定的集群为 `cluster-view`。选定的数据库为 `dev`。数据库 `data_catalog_database` 已列出，但未针对直接连接启用此数据库。您可以在您运行的 SQL 中选择 `dev` 数据库并引用 `data_catalog_database`。

```
SELECT * FROM "data_catalog_database"."myschema"."category3";
```

该查询检索由 AWS Glue Data Catalog 编目的数据。

## 创建架构

您可以使用查询编辑器 v2 在集群或工作组中创建架构。

创建架构

有关架构的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Schemas。

1. 选择  创建，然后选择架构。
2. 输入架构名称。
3. 选择本地或外部作为架构类型。

   有关本地架构的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 CREATE SCHEMA。有关外部架构的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 CREATE EXTERNAL SCHEMA。
4. 如果选择外部，则可以选择以下外部架构。

   - Glue 数据目录 – 在 Amazon Redshift 中创建引用 AWS Glue 中的表的外部架构。除了选择 AWS Glue 数据库，还可选择与集群关联的 IAM 角色以及与数据目录关联的 IAM 角色。
   - PostgreSQL – 在 Amazon Redshift 中创建外部架构，此架构引用 Amazon RDS for PostgreSQL 或 Amazon Aurora PostgreSQL 兼容版本的数据库。还提供数据库的连接信息。有关联合查询的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的通过联合查询来查询数据。

- MySQL – 在 Amazon Redshift 中创建外部架构，该架构引用 Amazon RDS for MySQL 和/或 Amazon Aurora MySQL 兼容版本的数据库。还提供数据库的连接信息。有关联合查询的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的通过联合查询来查询数据。

5. 选择创建架构。

   新 Schema 将在树状视图面板中显示。

## 创建表

您可以使用查询编辑器 v2 在集群或工作组中创建表。

创建表

您可以根据您指定或定义表中每列的逗号分隔值 (CSV) 文件创建表。有关表的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Designing tables 和 CREATE TABLE。

选择在编辑器中打开查询在运行查询以创建表之前查看和编辑 CREATE TABLE 语句。

1. 选择

   创建，然后选择表。
2. 选择架构。
3. 输入表名称。
4. 选择

   添加字段以添加列。
5. 使用 CSV 文件作为表定义模板：

   a. 选择从 CSV 加载。

   b. 浏览到文件位置。

      如果您使用 CSV 文件，请确保该文件的第一行包含列标题。

   c. 选择文件，然后选择打开。确认列名和数据类型符合您的要求。
6. 对于每一列，选择该列并选择所需的选项：

   - 为编码选择一个值。

- 选择默认值。
- 如果您想增加列值，启用自动增量。然后为自动增加种子和自动增量步骤指定值。
- 如果该列应始终包含值，启用非 NULL。
- 输入列的大小值。
- 如您希望该列成为主密钥，启用主密钥。
- 如您希望该列成为唯一密钥，启用唯一密钥。

7. （可选）选择表详细信息然后选择以下任何选项：
   - 分配密钥列和样式。
   - 对密钥列进行排序和排序类型。
   - 启用备份将表包含在快照中。
   - 启用临时表将表创建为临时表。
8. 选择在编辑器中打开查询继续指定用于定义表的选项，或选择创建表来创建表。

## 创建函数

您可以使用查询编辑器 v2 在集群或工作组中创建函数。

创建函数

1. 选择 ⊕ 创建，然后选择函数。
2. 对于类型，选择 SQL 或 Python。
3. 为架构选择一个值。
4. 为函数名称输入一个值。
5. 为函数波动性输入一个值。
6. 按输入参数的顺序排列的数据类型选择参数。
7. 为返回值选择一种数据类型。
8. 输入此函数的 SQL 程序或 Python 程序代码。
9. 选择创建。

有关用户定义的函数 (UDF) 的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的创建用户定义的函数。

## 查看查询和选项卡历史记录

您可以使用查询编辑器 v2 查看查询历史记录。仅使用查询编辑器 v2 运行的查询出现在查询历史记录中。显示已使用编辑器选项卡或笔记本选项卡运行的查询。您可以按时间段筛选显示的列表，例如 `This week`，其中周定义为星期一至星期日。查询列表一次将提取符合您的筛选条件的 25 行查询。选择加载更多可查看下一组。选择一个查询，在操作菜单中，可用的操作取决于是否已保存所选的查询。您还可以选择以下操作：

- 查看查询详细信息 - 显示一个查询详细信息页面，其中包含有关已运行的查询的更多信息。
- 在新选项卡中打开查询 - 打开一个新的编辑器选项卡，并在其中填入所选查询。如果仍处于连接状态，则会自动选择集群或工作组和数据库。要运行查询，请先确认已选择正确的集群或工作组和数据库。
- 打开源选项卡 - 如果仍处于打开状态，则在运行查询时导航到包含查询的编辑器或笔记本选项卡。运行查询后，编辑器或笔记本的内容可能已发生更改。
- 打开保存的查询 - 导航到编辑器或笔记本选项卡并打开查询。

您还可以查看编辑器选项卡中运行的查询的历史记录或笔记本选项卡中运行的查询的历史记录。要在选项卡中查看查询历史记录，请选择选项卡历史记录。在选项卡历史记录中，您可以执行以下操作：

- 复制查询 - 将查询版本 SQL 内容复制到剪贴板。
- 在新选项卡中打开查询 - 打开一个新的编辑器选项卡，并在其中填入所选查询。要运行查询，您必须选择集群或工作组和数据库。
- 查看查询详细信息 - 显示一个查询详细信息页面，其中包含有关已运行的查询的更多信息。

## 与 Amazon Q 生成式 SQL 交互

> **Note**
>
> 只有以下 AWS 区域中支持 Amazon Q 生成式 SQL：
>
> - 美国东部（弗吉尼亚北部）区域 (us-east-1)
> - 美国东部（俄亥俄）区域 (us-east-2)
> - 美国西部（俄勒冈州）区域（us-west-2）

- 亚太地区（孟买）区域 (ap-south-1)
- 亚太地区（首尔）区域 (ap-northeast-2)
- 亚太地区（新加坡）区域 (ap-southeast-1)
- 亚太地区（悉尼）区域 (ap-southeast-2)
- 亚太地区（东京）区域（ap-northeast-1）
- 加拿大（中部）区域 (ca-central-1)
- 欧洲（法兰克福）区域 (eu-central-1)
- 欧洲（爱尔兰）区域 (eu-west-1)
- 欧洲（伦敦）区域 (eu-west-2)
- 欧洲（巴黎）区域 (eu-west-3)
- 南美洲（圣保罗）区域 (sa-east-1)

有关数据处理位置的信息，请参阅《Amazon Q 开发者版用户指南》中的 Cross region inference in Amazon Q Developer。

您可以在 Amazon Redshift 查询编辑器 v2 中，与 Amazon Q 生成式 SQL 功能进行交互。这是一个编码助手，可以根据您的提示和数据库架构生成 SQL 语句。当您在查询编辑器 v2 中撰写笔记本时，可以使用这个编码助手。生成的 SQL 用于笔记本所连接的数据库。

在与 Amazon Q 生成式 SQL 交互时，请提出具体问题，在有复杂请求时进行迭代，并验证答案的准确性。

在以自然语言提供分析请求时，请尽可能具体，以便编码助手准确了解您的需求。正确的做法不是询问“查找门票销量最高的前几个场馆”，而是要提供更多详细信息，例如“查找 2008 年门票销量最高的前三个场馆的名称/ID”。如果您知道数据库中对象的名称，请使用一致且具体的名称。例如，使用数据库中定义的架构、表和列名称，而不是以不同的方式引用同一对象，这会让助手不明所以。

将复杂的请求拆分为多个简单的语句，便于助手解释。反复提出跟进问题，让助手提供更详细的分析。例如，首先问“哪个州的场馆最多？” 然后根据回答，问“这个州最受欢迎的场馆是哪个？”。

在运行生成的 SQL 之前，请对其进行检查，以确保准确性。如果生成的 SQL 查询存在错误或与您的意图不符，请向助手提供如何更正查询的说明，而不是以不同的措辞来重述整个请求。例如，如果查询中缺少关于年份的谓词从句，请询问“提供 2008 年的场馆”。

将运行生成的 SQL 时收到的错误文本作为提示提交回 Amazon Q 生成式 SQL。它能从这些错误中学习，从而生成更好的 SQL。

将您的架构添加到 SQL 搜索路径中，以表示应使用该架构。例如，当数据在 tickit 架构而非公共架构中时，添加 tickit 架构。

```
set search_path to '$user', tickit;
```

## 与 Amazon Q 生成式 SQL 交互时的注意事项

使用聊天面板时，请注意以下几点。

- 账户的查询编辑器 v2 管理员必须已在生成式 SQL 设置页面中开启聊天功能。
- 要使用 Amazon Q 生成式 SQL，除了在查询编辑器 v2 的 AWS 托管策略中指定的其它权限外，还需要在 IAM 策略中指定 `sqlworkbench:GetQSqlRecommendations` 权限。有关 AWS 托管策略的更多信息，请参阅访问查询编辑器 v2。
- 您必须使用英语撰写问题。
- 问题必须与集群或工作组中连接的数据库有关。为避免空状态错误，数据库中应至少有一个表和一些数据。
- 您的问题必须与所连接数据库中存储的数据有关。它不能引用外部架构。有关所支持架构的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的创建架构。
- 如果在任何问题中会导致 SQL 更改所连接数据库，就可能产生警告。
- 生成式人工智能技术是一项全新的技术，其回复中可能会出现错误，有时将这种错误称为幻觉。在您的环境或工作负载中使用代码之前，请对所有代码进行测试并检查是否存在错误和漏洞。
- 您可以通过分享账户中其他用户运行的 SQL 查询来改进建议。账户管理员可以运行以下 SQL 命令来允许访问账户的查询历史记录。

  ```
  GRANT ROLE SYS:MONITOR to "IAMR:role-name";
  GRANT ROLE SYS:MONITOR to "IAM:user-name";
  GRANT ROLE SYS:MONITOR to "database-username";
  ```

  有关 `SYS:MONITOR` 的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Amazon Redshift 系统定义的角色。
- 您的数据是安全和私密的。数据不会跨账户共享。您的查询、数据和数据库架构不会用于训练生成式人工智能基础模型 (FM)。您的输入将用作 FM 的上下文提示，仅用于回答您的查询。

## 使用生成式 SQL

配置了正确的权限后，在查询编辑器 v2 中处理笔记本时，您可以选择一个图标开始对话。

与 Amazon Q 生成式 SQL 聊天进行交互以生成 SQL

1. 在查询编辑器 v2 的编辑器选项卡中，打开笔记本。
2. 选择

   生成式 SQL 图标，然后在聊天面板中，按照说明向 Amazon Redshift 查询编辑器 v2 生成式 SQL 提问。

   您在提示字段中提出问题，Amazon Q 生成式 SQL 就会给出建议的 SQL。所出现的任何错误都会在聊天面板中返回给您。
3. 选择添加到笔记本，在笔记本中添加一个带有您的提示的 Markdown 单元格，以及一个包含建议 SQL 的 SQL 单元格。
4. （可选）选择

   “有帮助”反馈图标或

   “没有作用”反馈图标，针对生成的 SQL 提供反馈。您可以将“没有作用”反馈归类为 `Incorrect tables/columns`、`Incorrect predicates/literals/group bys`、`Incorrect SQL structure` 或 `Other`。此外，您还可以提供一些自由格式的文本，对 SQL 的准确性提出反馈。
5. （可选）选择重新生成 SQL，对相同的提示生成另一个回复。您可以选择为当前提示重新生成 SQL一次。
6. （可选）在生成式 SQL 聊天面板中，选择

   更多图标，然后选择刷新数据库以刷新描述所连接数据库的元数据。此元数据包括数据库中架构、表和列的定义。

## 以管理员身份更新生成式 SQL 设置

拥有正确 IAM 权限的用户可以查看和更改相同 AWS 账户中其他用户的生成式 SQL 设置。在查询编辑器 v2 的 AWS 托管策略中指定的其他权限之外，此管理员在其 IAM 策略中还必须拥有

sqlworkbench:UpdateAccountQSqlSettings 权限。有关 托管策略的更多信息，请参阅使用查询编辑器 v2 所需的权限。

管理员为账户中的所有用户开启生成式 SQL 聊天

1. 选择

   设置图标以显示不同设置屏幕的菜单。
2. 然后选择

   生成式 SQL 设置图标以显示 Q 生成式 SQL 设置页面。
3. 选择 Q 生成式 SQL 设置来为账户中的用户开启生成式 SQL 功能。

   开启 Amazon Q 生成式 SQL 后，您可以查看分配中剩余的提示数。查询编辑器 v2 管理员可以让账户中的用户使用 Amazon Q 开发者版专业套餐。要使用专业套餐，请使用 IAM Identity Center 设置用户，并让每个用户订阅 Amazon Q 开发者版专业套餐。有关使用 Amazon Redshift 设置 IAM Identity Center 的信息，请参阅 将 Redshift 与 AWS IAM Identity Center 连接来提供单点登录体验。有关 Amazon Q 开发者版定价的信息，请参阅 Amazon Q 开发者版定价。

   使用 Amazon Q 开发者版免费套餐时，AWS 账户 所有用户的提示总数在一个月内限制为 1000。使用 Amazon Q 开发者版专业套餐时，单个用户每月可提交的提示总数限制为 1000。您可以在设置页面上查看可用的提示数量。有关 Amazon Q 开发者版定价的信息，请参阅 Amazon Q 开发者版定价。

## 自定义上下文

查询编辑器 v2 管理员可以指定自定义上下文，以便根据环境定制生成的 SQL。自定义上下文提供领域知识和偏好设置，可对 SQL 生成进行精细控制。自定义上下文在 JSON 文件中定义，可由查询编辑器 v2 管理员上传到 Amazon Q 生成式 SQL。

用于为数据仓库个性化设置生成的 SQL 的 JSON 键如下。

所有表引用均需遵循三部分表示法 database.schema.table。

资源

资源指定了应用自定义上下文的数据资产的范围或部分。

ResourceId

指定资源的唯一标识符。对于 Amazon Redshift 集群，请指定 `cluster id`。对于 Redshift Serverless 工作组，请指定 `workgroup name`。

ResourceType

有效值：`REDSHIFT_WAREHOUSE`。

TablesToInclude

指定生成 SQL 时要考虑的一组表。如果要将 SQL 查询的范围限制在已定义的可用表子集上，该字段至关重要。该字段可以减少不必要的表引用，有助于优化生成过程。您可以将该字段与 `TablesToExclude` 搭配使用，以便更精细地控制查询的生成。

TablesToExclude

指定从 SQL 生成中排除的表集。当某些表无关紧要或不应在查询生成过程中考虑时，请使用该字段。

TableAnnotations

提供有关所用表的元数据或补充信息。这些注释可包括表描述、使用说明或任何其它属性，以便于 Amazon Q 生成式 SQL 更好地理解表的上下文或结构。这对于通过增加表定义的清晰度来提高 SQL 生成的准确性非常有价值。

ColumnsToInclude

定义在生成 SQL 查询时包含指定表中的哪些列。该字段有助于 Amazon Q 生成式 SQL 将重点放在相关列上，并通过缩小数据检索范围来提高性能。该字段可确保 Amazon Q 生成式 SQL 只提取给定查询上下文所需的数据。

ColumnsToExclude

指定在生成 SQL 时不考虑的列。当某些列包含 Amazon Q 生成式 SQL 不应考虑的无关数据或冗余数据时，可以使用该字段。通过管理列的包含和排除，您可以优化结果并保持对检索到的数据的控制。

ColumnAnnotations

与 `TableAnnotations` 类似，该字段提供特定于各个列的元数据或注释。这些注释可以提供对列定义或特殊处理说明的深入了解。这些信息有助于指导 SQL 生成过程，并确保在查询中适当使用列。

CuratedQueries

一组预定义的问题和答案示例，其中问题是用自然语言（NLQ）编写的，答案是相应的 SQL 查询。这些示例有助于 Amazon Q 生成式 SQL 理解其预期生成的查询类型。它们是提高 Amazon Q 生成式 SQL 输出准确性和相关性的参考点。

CustomDocuments

为 Amazon Q 生成式 SQL 提供的附加信息或提示，如定义、特定领域知识或解释。例如，如果您的业务部门使用一种独特的方法来计算某个值，如“制造部门的总销售额为价格 * 收入”，则可以在此记录下来。这些文档通过提供额外的上下文，增强了 Amazon Q 生成式 SQL 解释自然语言输入的能力。

AdditionalTables

指定生成 SQL 时应考虑但不属于存储在数据仓库中的数据的任何其它表。这使得 Amazon Q 生成式 SQL 能够将外部数据来源整合到其 SQL 生成逻辑中，从而扩大其处理复杂数据环境的能力。

AppendToPrompt

为 Amazon Q 生成式 SQL 提供的附加说明或指南，用于指导 SQL 生成过程。这可以包括有关如何构造查询的具体指令、对某些 SQL 构造的偏好，或任何其它可提高 Amazon Q 生成式 SQL 输出质量的高级指令。

下面的自定义上下文示例向您展示了 JSON 文件的格式，并定义了以下内容：

- 为集群 `mycluster` 的 Amazon Redshift 数据仓库定义自定义上下文。
- 定义要包含和排除的特定表和列，以便于优化 SQL 生成过程。
- 定义要包含的表和列的注释。
- 定义供 Amazon Q 生成式 SQL 使用的精选查询示例。
- 定义生成 SQL 时要使用的自定义文档和护栏。
- 定义生成 SQL 时要使用的附加表 DDL。

```
{
    "resources": [
        {
            "ResourceId": "mycluster",
            "ResourceType": "REDSHIFT_WAREHOUSE",
            "TablesToInclude": [
                "database.schema.table1",
```

```
            "database.schema.table2"
        ],
        "TablesToExclude": [
            "database.schema.table3",
            "database.schema.table4"
        ],
        "ColumnsToInclude": {
            "database.schema.table1": [
                "col1",
                "col2"
            ],
            "database.schema.table2": [
                "col1",
                "col2"
            ]
        },
        "ColumnsToExclude": {
            "database.schema.table5": [
                "col1",
                "col2"
            ],
            "database.schema.table6": [
                "col1",
                "col2"
            ]
        },
        "TableAnnotations": {
            "database.schema.table1": "table1 refers to Q3 sales",
            "database.schema.table2": "table2 refers to Q4 sales"
        },
        "ColumnAnnotations": {
            "database.schema.table1": {
                "col1": "col1 refers to Q3 sale total",
                "col2": "col2 refers to sale location"
            },
            "database.schema.table2": {
                "col1": "col2 refers to Q4 sale total",
                "col2": "col2 refers to sale location"
            }
        },
        "CuratedQueries": [
            {
                "Question": "what is the sales data for Q3",
                "Answer": "SELECT * FROM table1"
```

```
                },
                {
                    "Question": "what is the sales data for Q4",
                    "Answer": "SELECT * FROM table2"
                }
            ],
            "CustomDocuments": [
                "in manufacturing division total sales is price * revenue",
                "in research division total sales is price * revenue"
            ],
            "AdditionalTables": {
                "database.schema.table8": "create table database.schema.table8(col1
 int)",
                "database.schema.table9": "create table database.schema.table9(col1
 int)"
            },
            "AppendToPrompt": "Apply these guardrails: Queries should never return the
 secretId field of a user."
        }
    ]
}
```

## 教程：对 TICKIT 数据使用 Amazon Q 生成式 SQL 功能

要编写有效的提示来生成 SQL，您必须了解自己的数据库架构和数据。TICKIT 数据包含七个表：两个事实表和五个维度表。样本数据包含 2008 年举办的娱乐活动的观众售票记录。有关 TICKIT 数据的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的示例数据库。您可以通过各种方法将 TICKIT 数据加载到数据库中既可以使用 Amazon Redshift 控制台，也可以使用查询编辑器 v2。查询编辑器 v2 提供了将 TICKIT 数据加载到数据库 `sample_data_dev` 的方法。有关信息，请参阅将数据加载到数据库。查询编辑器 v2 还提供了适用于 TICKIT 数据的示例提示。以下场景描述了与生成式 SQL 进行的对话，用以生成有关 TICKIT 样本数据的 SQL。在这个场景中，TICKIT 样本数据已在 Amazon Redshift 集群的 `dev` 数据库中创建。

> **Note**
>
> 这个例子用于说明对话过程。虽然您使用了相同的提示，但生成式 SQL 的响应并不一致。

### 使用 Amazon Q 生成式 SQL 的示例对话

1. 在编辑器中，连接到包含 TICKIT 样本数据的集群或工作组。

2. 创建一个空笔记本，然后选择



生成式 SQL 图标以打开聊天面板。

3. 输入以下提示，生成 SQL 来验证 VENUE 表中的记录数：

```
How many venues are there?
```

```
SELECT
  COUNT(*) AS num_venues
FROM
  tickit.venue
```

选择添加到笔记本，将两个单元格添加到打开的笔记本中。一个 Markdown 单元格中为“有多少个场馆？”另一个单元格包含生成的 SQL。

在 SQL 单元格中，选择运行以接收结果：

```
count
------
202
```

4. 要请求另一个版本的 SQL，请选择重新生成 SQL并会收到以下答案：

```
SELECT
  venuestate,
  COUNT(*) AS num_venues
FROM
  tickit.venue
GROUP BY
  venuestate
```

选择添加到笔记本，再添加两个单元格到打开的笔记本中。一个 Markdown 单元格中为“有多少个场馆？”另一个单元格包含生成的 SQL。

在 SQL 单元格中，选择运行 以接收结果，其中列出了各州的场馆数量：

```
venuestate num_venues
```

```
----------------------
MA         4
OH         8
MI         5
...
```

5. 生成式 SQL 助手可以对您的要求做出推测。尽管数据库中没有小组件，但您仍然可以询问。在这种情况下，它会给出消息，说明无法生成 SQL。

```
How many widgets are there?
```

```
I was not able to generate any SQL
```

6. 随着对数据库的了解越来越多，您可以提出更具体的问题来生成 SQL。例如，您知道数据库中包含 2008 年几个月的销售数据。输入以下提示以生成 SQL，用于查找 2008 年 11 月的销量：

```
How many sales were there in November 2008?
```

```
SELECT
  COUNT(*) AS num_sales
FROM
  tickit.sales s
  JOIN tickit.date d ON s.dateid = d.dateid
WHERE
  d.month = 'November'
  AND d.year = 2008
```

选择添加到笔记本，将两个单元格添加到打开的笔记本中。一个 Markdown 单元格中为“2008 年 11 月的销量是多少？” 另一个单元格包含生成的 SQL。

在 SQL 单元格中，选择运行以接收结果：

```
num_sales
-----
0
```

您发现这不是所期望的结果。

7. 您注意到谓词 date.month='November' 要求在 DATE 表中使用月份名称的缩写来表示月份。将谓词更改为 d.month='NOV'，然后重新运行 SQL。

```
SELECT
  COUNT(*)
FROM
  sales
  JOIN date ON sales.dateid = date.dateid
WHERE
  date.month = 'NOV'
  AND date.year = 2008
```

在 SQL 单元格中，选择运行以获取新结果。

```
count
-----
14261
```

8. 如果您提出的问题试图更改连接的数据库，则会返回一条警告消息以及所推荐的任何 SQL。输入以下提示以生成将数据插入表中的 SQL：

```
Insert 1 into the venue table.
```

```
INSERT
,
UPDATE
  OR delete data
FROM
  the database AS that could potentially change the data.Please provide a query
 that ONLY selects data
```

```
I was not able to generate the correct SQL code. I generated SQL, but you'll have
 to edit it to work with your database.
```

如果您选择添加到笔记本，将两个单元格添加到打开的笔记本中并运行 SQL，SQL 会失败。

```
ERROR: syntax error at or near "," Position: 132 [ErrorId:
 1-6546764a-011df2691778846219ce6ec2]
```

此场景仅用于说明与 Amazon Q 生成式 SQL 交互的一些基本方法。您可以使用此生成式 AI 技术进行更多实验，以帮助您开始编写 SQL 来查询数据库。

## 将数据加载到数据库

您可以使用查询编辑器 v2 将数据加载到 Amazon Redshift 集群或工作组的数据库中。本节介绍如何加载示例数据、来自 S3 的数据以及来自本地文件设置和工作流程的数据。

### 示例数据

查询编辑器 v2 附带可加载到示例数据库和相应 Schema 中的示例数据和笔记本。

要加载示例数据，请选择与要加载的示例数据相关联的



图标。然后，查询编辑器 v2 会将数据加载到数据库 `sample_data_dev` 的架构中，并创建一个已保存笔记本的文件夹。

提供了以下示例数据集。

tickit

Amazon Redshift 文档中的大多数示例使用称为 `tickit` 的示例数据。此数据包含七个表：两个事实表和五个维度。当您加载这些数据时，将使用示例数据更新模式 `tickit`。有关 `tickit` 数据的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的示例数据库。

tpch

此数据用于决策支持基准。当您加载这些数据时，将使用示例数据更新模式 `tpch`。有关 `tpch` 数据类型的更多信息，请参阅 TPC-H。

tpcds

此数据用于决策支持基准。当您加载这些数据时，将使用示例数据更新模式 `tpcds`。有关 `tpcds` 数据的更多信息，请参阅 TPC-DS。

### 从 Simple Storage Service（Amazon S3）加载数据

您可将 Amazon S3 数据加载到现有表或新表中。

将数据加载到现有表中

查询编辑器 v2 使用 COPY 命令从 Simple Storage Service（Amazon S3）加载数据。在查询编辑器 v2 加载数据向导中生成和使用的 COPY 命令支持从 Amazon S3 复制的 COPY 命令语法可用的许多参数。有关 COPY 命令及其用于从 Simple Storage Service（Amazon S3）加载数据选项的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Amazon Simple Storage Service 中的 COPY 命令。

1. 确认表已在要加载数据的数据库中创建。
2. 在继续之前，请在查询编辑器 v2 的树视图面板中确认您已连接到目标数据库。使用上下文菜单（右键单击）创建与将要加载数据的集群或工作组的连接。

   选择

   Load data（加载数据）
3. 对于数据来源，选择从 S3 存储桶加载。
4. 在 S3 URI中，选择 Browse S3（浏览 S3）以查找包含要加载数据的 Simple Storage Service（Amazon S3）桶。
5. 如果指定的 Amazon S3 桶与目标表不在同一个 AWS 区域中，则针对数据所在的 AWS 区域选择 S3 file location（S3 文件位置）。
6. 如果 Simple Storage Service（Amazon S3）文件实际上包含多个 Simple Storage Service（Amazon S3）桶 URI 清单，选择This file is a manifest file（此文件是清单文件）。
7. 为要上载的文件选择 File format（文件格式）。支持的数据格式有 CSV、JSON、DELIMITER、FIXEDWIDH、SHAPEFILE、AVRO、PARQUET 和 ORC。根据指定的文件格式，您可以选择相应的 File options（文件选项）。如果数据已加密，您还可以选择 Data is encrypted（数据已加密），并输入用于加密数据的 KMS 密钥 Amazon Resource Name（ARN）。

   如果您选择 CSV 或 DELIMITER，则还可以选择分隔符字符，以及在指定的行编号实际上是列名而不是要加载的数据时是否忽略标题行。
8. 选择压缩方法来压缩文件。原定设置为无压缩。
9. （可选）Advanced settings（高级设置）支持各种 Data conversion parameters（数据转换参数）和 Load operations（加载操作）。根据文件的需要输入此信息。

   有关数据转换和数据加载参数的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的数据转换参数和数据加载操作。

10. 选择下一步。
11. 选择加载现有表。
12. 确认或选择目标表的位置，包括在其中加载数据的集群或工作组、数据库、架构和表名称。
13. 选择具有从 Simple Storage Service（Amazon S3）加载数据所需的权限的 IAM role（IAM 角色）。
14. （可选）选择列名称，将其输入到 Column mapping（列映射）中，以按输入数据文件的顺序映射列。
15. 选择 Load data（加载数据）开启数据加载。

    加载完成后，查询编辑器将显示用于加载数据的生成 COPY 命令。将显示 COPY 的 Result（结果）。如果成功，您现在可以使用 SQL 从加载的表中选择数据。当出现错误时，请查询系统视图 STL_LOAD_ERRORS 以获取更多详细信息。有关 COPY 命令错误的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 STL_LOAD_ERRORS。

将数据加载到新表时，查询编辑器 v2 首先在数据库中创建表，然后作为同一个工作流中的单独操作加载数据。

将数据加载到新表

查询编辑器 v2 使用 COPY 命令从 Simple Storage Service（Amazon S3）加载数据。在查询编辑器 v2 加载数据向导中生成和使用的 COPY 命令支持从 Amazon S3 复制的 COPY 命令语法可用的许多参数。有关 COPY 命令及其用于从 Simple Storage Service（Amazon S3）加载数据选项的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Amazon Simple Storage Service 中的 COPY 命令。

1. 在继续之前，请在查询编辑器 v2 的树视图面板中确认您已连接到目标数据库。使用上下文菜单（右键单击）创建与将要加载数据的集群或工作组的连接。

   选择

   Load data（加载数据）
2. 对于数据来源，选择从 S3 存储桶加载。
3. 在 S3 URI中，选择 Browse S3（浏览 S3）以查找包含要加载数据的 Simple Storage Service（Amazon S3）桶。
4. 如果指定的 Amazon S3 桶与目标表不在同一个 AWS 区域中，则针对数据所在的 AWS 区域选择 S3 file location（S3 文件位置）。

5. 如果 Simple Storage Service（Amazon S3）文件实际上包含多个 Simple Storage Service（Amazon S3）桶 URI 清单，选择This file is a manifest file（此文件是清单文件）。
6. 为要上载的文件选择 File format（文件格式）。支持的数据格式有 CSV、JSON、DELIMITER、FIXEDWIDH、SHAPEFILE、AVRO、PARQUET 和 ORC。根据指定的文件格式，您可以选择相应的 File options（文件选项）。如果数据已加密，您还可以选择 Data is encrypted（数据已加密），并输入用于加密数据的 KMS 密钥 Amazon Resource Name（ARN）。

   如果您选择 CSV 或 DELIMITER，则还可以选择分隔符字符，以及在指定的行编号实际上是列名而不是要加载的数据时是否忽略标题行。
7. 选择压缩方法来压缩文件。原定设置为无压缩。
8. （可选）Advanced settings（高级设置）支持各种 Data conversion parameters（数据转换参数）和 Load operations（加载操作）。根据文件的需要输入此信息。

   有关数据转换和数据加载参数的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的数据转换参数和数据加载操作。
9. 选择下一步。
10. 选择加载新表。

    表列根据输入数据推断得出。您可以通过添加列和表详细信息来修改表架构的定义。要恢复到查询编辑器 v2 推断的表架构，请选择还原为默认值。
11. 确认或选择目标表的位置，包括在其中加载数据的集群或工作组、数据库和架构。输入要创建的表的名称。
12. 选择具有从 Simple Storage Service（Amazon S3）加载数据所需的权限的 IAM role（IAM 角色）。
13. 选择创建表即可使用所示的定义创建表。

    此时将显示表定义的复查摘要。系统会在数据库中创建表。以后要删除该表时，请运行 DROP TABLE SQL 命令。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 DROP TABLE。
14. 选择 Load data（加载数据）开启数据加载。

    加载完成后，查询编辑器将显示用于加载数据的生成 COPY 命令。将显示 COPY 的 Result（结果）。如果成功，您现在可以使用 SQL 从加载的表中选择数据。当出现错误时，请查询系统视图 STL_LOAD_ERRORS 以获取更多详细信息。有关 COPY 命令错误的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 STL_LOAD_ERRORS。

## 从本地文件设置和工作流加载数据

您可从本地文件将数据加载到现有表或新表中。

管理员设置从本地文件加载数据

查询编辑器 v2 管理员必须在 Account settings（账户设置）窗口中指定常用 Amazon S3 桶。必须为账户用户配置适当的权限。

- 所需的 IAM 权限 – 从本地文件加载的用户必须拥有 `s3:ListBucket`、`s3:GetBucketLocation`、`s3:putObject`、`s3:getObject` 和 `s3:deleteObject` 权限。可以指定 *`optional-prefix`*，以将查询编辑器 v2 对此桶相关的使用限制为具有此前缀的对象。将同一 Amazon S3 桶用于查询编辑器 v2 以外的用途时，您可以使用此选项。有关桶和前缀的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的管理用户对特定文件夹的访问权限。为确保不允许跨用户访问数据，我们建议查询编辑器 v2 管理员使用 Amazon S3 存储桶策略来限制基于 `aws:userid` 的对象访问权限。以下示例允许 Amazon S3 对 *`<staging-bucket-name>`* 拥有权限，并仅对前缀为 `aws:userid` 的 Amazon S3 对象进行读/写访问。

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::<staging-bucket-name>"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject"
            ],
            "Resource": [
```

```
                "arn:aws:s3:::<staging-bucket-name>[/<optional-prefix>]/
${aws:userid}/*"
            ]
        }
    ]
}
```

- 数据分离 – 我们建议用户不要访问彼此的数据（即使是短暂的访问）。从本地文件加载将使用查询编辑器 v2 管理员设置的暂存 Amazon S3 桶。为暂存桶配置桶策略，以在用户之间提供数据分离。以下示例显示了在 *<staging-bucket-name>* 的用户之间分离数据的桶策略。

  JSON

```
{
 "Version":"2012-10-17",
    "Statement": [
        {"Sid": "userIdPolicy",
            "Effect": "Deny",
            "Principal": "*",
            "Action": ["s3:PutObject",
                       "s3:GetObject",
                       "s3:DeleteObject"],
            "NotResource": [
                "arn:aws:s3:::<staging-bucket-name>[/<optional-prefix>]/
${aws:userid}/*"
            ]
         }
    ]
}
```

从本地文件加载数据

将本地文件数据加载到现有表中

您的查询编辑器 v2 管理员必须在账户设置窗口中指定公共 Amazon S3 存储桶。查询编辑器 v2 会自动将本地文件上传到账户使用的公共 Amazon S3 存储桶，然后使用 COPY 命令加载数据。在查询编辑器 v2 加载本地文件窗口中生成和运行的 COPY 命令支持从 Amazon S3 复制的 COPY 命令语法可用的许多参数。有关 COPY 命令及其用于从 Amazon S3 加载数据的选项的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Amazon S3 中的 COPY 命令。

1. 确认表已在要加载数据的数据库中创建。

2. 在查询编辑器 v2 的树视图面板中确认您已连接到目标数据库。使用上下文菜单（右键单击）创建与将要加载数据的集群或工作组的连接。
3. 选择

   Load data（加载数据）
4. 对于 Data source（数据来源），选择 Load from local file（从本地文件中加载）。
5. 选择浏览以查找包含数据的文件来加载文件。默认情况下，会显示扩展名为 `.csv`、`.avro`、`.parquet` 和 `.orc` 的文件，但您可以选择其他文件类型。最大 文件大小为 100MB。
6. 为要上载的文件选择 File format（文件格式）。支持的数据格式有 CSV、JSON、DELIMITER、FIXEDWIDH、SHAPEFILE、AVRO、PARQUET 和 ORC。根据指定的文件格式，您可以选择相应的 File options（文件选项）。如果数据已加密，您还可以选择 Data is encrypted（数据已加密），并输入用于加密数据的 KMS 密钥 Amazon Resource Name（ARN）。

   如果您选择 CSV 或 DELIMITER，则还可以选择分隔符字符，以及在指定的行编号实际上是列名而不是要加载的数据时是否忽略标题行。
7. （可选）Advanced settings（高级设置）支持各种 Data conversion parameters（数据转换参数）和 Load operations（加载操作）。根据文件的需要输入此信息。

   有关数据转换和数据加载参数的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的数据转换参数和数据加载操作。
8. 选择下一步。
9. 选择加载现有表。
10. 确认或选择目标表的位置，包括在其中加载数据的集群或工作组、数据库、架构和表名称。
11. （可选）您可以选择列名称以输入到 Column mapping（列映射）中，以按输入数据文件的顺序映射列。
12. 选择 Load data（加载数据）开启数据加载。

    加载完成后，无论加载成功与否，都会显示一条消息。如果成功，您现在可以使用 SQL 从加载的表中选择数据。当出现错误时，请查询系统视图 STL_LOAD_ERRORS 以获取更多详细信息。有关 COPY 命令错误的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 STL_LOAD_ERRORS。

用于加载数据的 COPY 命令模板出现在您的 Query history（查询历史记录）中。此 COPY 命令模板显示了一些使用的参数，但它不能直接在编辑器选项卡中运行。有关查询历史记录的更多信息，请参阅查看查询和选项卡历史记录。

将数据加载到新表时，查询编辑器 v2 首先在数据库中创建表，然后作为同一个工作流中的单独操作加载数据。

将本地文件数据加载到新表

查询编辑器 v2 管理员必须在 Account settings（账户设置）窗口中指定常用 Amazon S3 桶。本地文件会自动上载到您的账户使用的一个公共 Amazon S3 桶，然后查询编辑器 v2 使用 COPY 命令加载数据。在查询编辑器 v2 加载本地文件窗口中生成和运行的 COPY 命令支持从 Amazon S3 复制的 COPY 命令语法可用的许多参数。有关 COPY 命令及其用于从 Amazon S3 加载数据的选项的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Amazon S3 中的 COPY 命令。

1. 在查询编辑器 v2 的树视图面板中确认您已连接到目标数据库。使用上下文菜单（右键单击）创建与将要加载数据的集群或工作组的连接。
2. 选择

   Load data（加载数据）
3. 对于 Data source（数据来源），选择 Load from local file（从本地文件中加载）。
4. 选择浏览以查找包含数据的文件来加载文件。默认情况下，会显示扩展名为 `.csv`、`.avro`、`.parquet` 和 `.orc` 的文件，但您可以选择其他文件类型。最大 文件大小为 100MB。
5. 为要上载的文件选择 File format（文件格式）。支持的数据格式有 CSV、JSON、DELIMITER、FIXEDWIDH、SHAPEFILE、AVRO、PARQUET 和 ORC。根据指定的文件格式，您可以选择相应的 File options（文件选项）。如果数据已加密，您还可以选择 Data is encrypted（数据已加密），并输入用于加密数据的 KMS 密钥 Amazon Resource Name（ARN）。

   如果您选择 CSV 或 DELIMITER，则还可以选择分隔符字符，以及在指定的行编号实际上是列名而不是要加载的数据时是否忽略标题行。
6. （可选）Advanced settings（高级设置）支持各种 Data conversion parameters（数据转换参数）和 Load operations（加载操作）。根据文件的需要输入此信息。

有关数据转换和数据加载参数的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的数据转换参数和数据加载操作。

7. 选择下一步。
8. 选择加载新表。
9. 确认或选择目标表的位置，包括在其中加载数据的集群或工作组、数据库和架构。输入要创建的表的名称。
10. 选择创建表即可使用所示的定义创建表。

    此时将显示表定义的复查摘要。系统会在数据库中创建表。以后要删除该表时，请运行 DROP TABLE SQL 命令。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 DROP TABLE。
11. 选择 Load data（加载数据）开启数据加载。

    加载完成后，将会显示一条消息，指示加载是否成功。如果成功，您现在可以使用 SQL 从加载的表中选择数据。当出现错误时，请查询系统视图 STL_LOAD_ERRORS 以获取更多详细信息。有关 COPY 命令错误的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 STL_LOAD_ERRORS。

    用于加载数据的 COPY 命令模板出现在您的 Query history（查询历史记录）中。此 COPY 命令模板显示了一些使用的参数，但它不能直接在编辑器选项卡中运行。有关查询历史记录的更多信息，请参阅查看查询和选项卡历史记录。

## 使用 Amazon Redshift 编写查询

您可以在编辑器中输入查询，也可以从查询中列出并选择运行以选择已保存的查询。

默认情况下，设置限制 100，将结果限制为 100 行。您可以禁用此选项来返回较大的结果集。禁用此选项时，如要避免非常大型结果集，则可以在 SQL 语句中包含 LIMIT 选项。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 ORDER BY 子句。

要在结果区域中显示查询计划，请打开解释。对结果开启解释图，也可显示解释计划的图形表示。

选择保存，将查询保存到查询文件夹。

查询成功，将显示成功消息。如果查询返回信息，则结果将显示在结果部分。如果结果数超过显示区域，则数字将显示在结果区域的顶部。您可以选择数字来显示连续的结果页面。

您可以筛选和排序每一列的结果。如要在结果列标题中输入筛选条
件，请将鼠标悬停在该列上便可查看筛选条件菜单
（ ☰ ），
您可以在此处输入筛选列的条件。

如果查询包含错误，则查询编辑器 v2 将在结果区域中显示错误消息。此消息提供有关如何更正查询的信息。

您可以使用结果区域中的上下文（右键单击）菜单导出或复制查询结果，如下所示：

- 选择复制到，然后选择 JSON 或 CSV，以便将选定的行下载到文件中。
- 选择复制行以便将选定的行复制到剪贴板。
- 如选择复制带标题的行，以便将带有列标题的选定行复制到剪贴板。

还可以在结果区域中选择导出，然后选择 JSON 或 CSV，以便将整组行结果下载到文件中。结果集中的行数可能受到查询中限制选项或 SQL `limit` 子句限制。下载结果集的最大大小为 5 MB。

也可以使用快捷方式（Windows 上的 Ctrl+C 或 macOS 上的 Cmd+C）将数据从当前结果页面复制到剪贴板。如果未选择任何行，则具有焦点的单元格将复制到剪贴板。如果选择了行，则选定的行将复制到剪贴板。

要添加新的查询选项卡，请选择
＋
图标，然后选择编辑器，此编辑器将显示在查询选项卡所在行中。查询选项卡可以使用，也可以不使用 `Isolated session`。对于隔离会话，SQL 命令的结果（例如，在一个编辑器选项卡中创建临时表）在另一个编辑器选项卡中不可见。在查询编辑器 v2 中打开编辑器选项卡时，默认为隔离会话。

运行查询

1. 在查询区域中，执行以下任一操作：
   - 输入查询。
   - 粘贴您复制的查询。
   - 选择查询文件夹中，打开上下文菜单（右键单击）中的保存查询，然后选择打开查询。
2. 确认您为计划运行的 SQL 选择了正确的集群或工作组以及数据库值。

   最初，您可以在树视图中选择您的集群或工作组。还可以在树视图中选择您的数据库。

您可以使用位于每个编辑器选项卡的隔离会话标题附近的下拉控件，更改每个编辑器选项卡中的集群或工作组以及数据库。

对于每个编辑器选项卡，您可以选择是否以隔离会话运行 SQL。隔离会话与数据库有自己的连接。使用它来运行与其他查询编辑器会话隔离的 SQL。有关 连接的更多信息，请参阅 打开查询编辑器 v2。

3. 选择运行。

将打开结果区域并显示查询结果。

显示查询的解释计划

1. 选择查询。
2. 启用解释。

默认情况下，解释图也将启动。

3. 选择运行。

运行查询，并在查询的结果区域中显示解释计划。

查询编辑器 v2 支持以下功能。

- 您可以在一个查询选项卡中使用多个 SQL 语句创作查询。这些查询将连续运行，并为每个查询打开多个结果选项卡。
- 您可以使用会话变量和临时表创作查询。
- 您可以使用由 ${*parameter*} 指定的可替换参数创作查询。您可以使用多个可替换参数创作 SQL 查询，并在 SQL 语句的多个位置使用相同的参数。

  当查询运行时，会显示一个用来输入参数值的窗口。每次运行查询时，都会显示窗口以输入参数值。

  有关示例，请参阅示例：销售额大于特定参数。
- 自动对查询进行版本控制。您可以选择要运行的查询的早期版本。
- 在继续工作流程之前，您无需等待查询完成。即使关闭查询编辑器，查询仍会继续运行。
- 创作查询时，支持自动完成架构、表和列名称。

SQL 编辑器支持以下功能：

- SQL 中使用的左方括号和右方括号具有匹配的颜色。编辑器中显示垂直线以帮助您匹配方括号。
- 您可以折叠和展开 SQL 的各个部分。
- 您可以在 SQL 中搜索和替换文本。
- 您可以使用快捷键执行多个常见编辑任务。
- SQL 错误会在编辑器中突出显示，以便于找到问题领域。

有关编辑功能的演示，请观看以下视频：Amazon Redshift 查询编辑器 v2 中新的和增强的编辑体验。

## 查询示例

在下方您可以找到可以运行的各种查询类型的描述。

许多此类查询中使用的数据来自 `tickit` 示例架构。有关加载示例 `tickit` 数据的更多信息，请参阅将数据加载到数据库。有关 `tickit` 示例数据的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的示例数据库。

运行这些示例查询时，请确认在编辑器中选择了正确的数据库，例如 `sample_data_dev`。

主题

- 示例：设置会话变量
- 示例：按总销售额排列的顶事件
- 示例：销售额大于特定参数
- 示例：创建一个临时表
- 示例：从临时表中选择

### 示例：设置会话变量

以下命令将 `search_path` 服务器配置参数设置为会话的公有参数。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 SET 和 search_path。

```
set search_path to public;
```

### 示例：按总销售额排列的顶事件

以下查询查找销售额最多的事件。

```
select eventname, count(salesid) totalorders, sum(pricepaid) totalsales
```

```
from sales, event
where sales.eventid=event.eventid
group by eventname
order by 3;
```

以下是结果的部分清单。

```
eventname           totalorders       totalsales
White Christmas          20              9352
Joshua Radin             38             23469
Beach Boys               58             30383
Linda Ronstadt           56             35043
Rascal Flatts            76             38214
Billy Idol               67             40101
Stephenie Meyer          72             41509
Indigo Girls             57             45399
...
```

示例：销售额大于特定参数

以下查询查找销售数量大于 ${numberoforders} 指定参数的销售额。当参数值为 7 时，结果是 60 行。运行查询时，查询编辑器 v2 会显示运行查询表窗口来收集 SQL 语句中参数值。

```
select salesid, qtysold
from sales
where qtysold > ${numberoforders}
order by 2;
```

以下是结果的部分清单。

```
salesid qtysold
20005 8
21279 8
130232 8
42737 8
74681 8
67103 8
105533 8
91620 8
121552 8
...
```

示例：创建一个临时表

通过销售和事件表，以下语句创建临时表 eventsalestemp。

```
create temporary table eventsalestemp as
select eventname, count(salesid) totalorders, sum(pricepaid) totalsales
from sales, event
where sales.eventid=event.eventid
group by eventname;
```

示例：从临时表中选择

按总订单排序，以下语句从临时表 eventsalestemp 中选择事件、订单总数和总销售额。

```
select eventname,  totalorders,  totalsales
from eventsalestemp
order by 2;
```

以下是部分结果列表。

```
eventname          totalorders   totalsales
White Christmas         20          9352
Joshua Radin            38         23469
Martina McBride         50         52932
Linda Ronstadt          56         35043
Indigo Girls            57         45399
Beach Boys              58         30383
...
```

## Amazon Redshift 中的笔记本

您可以使用笔记本在单个文档中组织、注释及共享多个 SQL 查询。您可以将多个 SQL 查询和 Markdown 单元格添加到笔记本中。笔记本提供了一种方法：通过使用多个查询和 Markdown 单元格，将与数据分析相关的查询和解释分组到单个文档中。您可以使用 Markdown 语法添加文本并设置外观格式，以便为数据分析任务提供上下文和其它信息。您可以与团队成员共享您的笔记本。

要使用笔记本，您必须为您的 IAM 主体（IAM 用户或 IAM 角色）添加笔记本的权限。作为最佳实践，我们建议将权限策略附加到 IAM 角色，然后根据需要将其分配给用户和组。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。您可以向其中一个查询编辑器 v2 托管策略添加权限。有关更多信息，请参阅 访问查询编辑器 v2。

可以按顺序运行笔记本的所有单元格。笔记本的 SQL 查询单元格具有大多数与查询编辑器选项卡相同的功能。有关更多信息，请参阅 使用 Amazon Redshift 编写查询。以下是查询编辑器选项卡和笔记本中的 SQL 单元格之间的差异。

- 在笔记本中，没有用于对 SQL 语句运行 `Explain` 的控件。
- 在笔记本中，每个 SQL 单元格只能创建一个图表。

您可以将笔记本导出和导入到使用查询编辑器 v2 创建的文件中。文件扩展名为 `.ipynb`，文件大小最大可为 5 MB。SQL 和 Markdown 单元格存储在文件中。集群或工作组和数据库不会存储在导出的笔记本中。当您打开导入的笔记本时，您可以选择用于运行该笔记本的集群或工作组和数据库。运行 SQL 单元格后，可以在结果选项卡中选择是否将当前结果页显示为图表。查询的结果集不会存储在笔记本中。

## 创建笔记本

可以创建笔记本来在单个文档中整理、注释及共享多个 SQL 查询。

创建笔记本

1. 在导航器菜单中，选择编辑器图标（）。
2. 选择加号图标（），然后选择笔记本。

   默认情况下，笔记本中会显示 SQL 查询单元格。
3. 在 SQL 查询单元格中，执行以下任一操作：

   - 输入查询。
   - 粘贴您复制的查询。
4. （可选）选择加号图标（），然后选择 Markdown 以添加 Markdown 单元格，在其中可以使用标准 Markdown 语法提供描述性或解释性文本。

5. （可选）选择加号图标

（  ），然后选择 SQL 以插入 SQL 单元格。

您可以单击铅笔图标

（  ）

来重命名笔记本。

在菜单图标

（  ）

中，您还可以对笔记本执行以下操作：

- 

  与我的团队分享 – 按照标签的定义，与您的团队分享笔记本。如要与团队分享笔记本，请确保您将主体标签 `sqlworkbench-team` 设置为与账户中其它团队成员相同的值。例如，管理员可以为会计部门的每个人将该值设置为 `accounting-team`。有关示例，请参阅使用查询编辑器 v2 所需的权限。
- 

  导出 – 将笔记本导出到扩展名为 `.ipynb` 的本地文件。
- 

  导入查询 – 将查询从本地文件导入到笔记本单元格中。您可以导入扩展名为 `.sql` 和 `.txt` 的文件。
- 

  保存版本 – 创建笔记本的一个版本。要查看笔记本的版本，请导航到已保存的笔记本并打开版本历史记录。
- 

  复制 – 创建笔记本的副本并在新的笔记本选项卡中将其打开。
- 

  快捷方式 – 显示编写笔记本时可用的快捷方式。

## 导入到笔记本

您可以将整个笔记本或单个 SQL 单元格导入到查询编辑器 v2 笔记本中。

要将整个笔记本从本地文件导入到我的笔记本，请选择



导入，然后选择导入笔记本。导航到包含您的笔记本的 .ipynb 文件。笔记本会导入到当前打开的笔记本文件夹。然后您可以在笔记本编辑器中打开笔记本。

要将查询从本地文件导入到笔记本的 SQL 单元格，请选择



导入，然后选择导入查询。在导入查询窗口中，按照屏幕上的说明，选择可作为查询导入到新笔记本或现有笔记本中的文件和文件夹。这些文件的扩展名必须为 .sql 或 .txt。每个查询最多可以有 10000 个字符。在您添加到现有笔记本时，可以从保存的笔记本列表中的所有笔记本中选择要使用的笔记本。导入的查询将作为 SQL 单元格添加到笔记本的末尾。选择新笔记本时，您需要提供笔记本的名称，该笔记本将在当前打开的已保存笔记本的文件夹中创建。

> **Note**
>
> 在 macOS 上使用 TextEdit 应用程序创建 .sql 文件时，您可能会遇到向文件添加额外的隐藏扩展名的问题。例如，在 TextEdit 中创建的名为 Test.sql 的文件最终可能会另存为 Test.sql.rtf。查询编辑器 v2 不支持扩展名为 .rtf 的文件。但是，如果您使用 TextEdit 创建 .sql 文件并将其另存为纯文本文件，则该文件会有一个额外的隐藏扩展名 .txt。例如，名为 Text.sql 的文件可能会另存为 Text.sql.txt。与 .rtf 扩展名不同，查询编辑器 v2 确实支持扩展名为 .txt 的文件，因此在将查询导入到笔记本时支持 Text.sql.txt。

要查看笔记本的演示，请观看以下视频：查询编辑器 v2 中的 Amazon Redshift SQL 笔记本。

## 查询 AWS Glue Data Catalog

可以使用查询编辑器 v2 查询在 AWS Glue Data Catalog 中编目的数据，方法是使用特定 SQL 命令并授予本节中概述的权限。默认情况下，AWS Glue Data Catalog 列为名为 awsdatacatalog 的查询编辑器 v2 数据库。查询 AWS Glue Data Catalog 并非在所有 Amazon Redshift AWS 区域中都可用。使用 SHOW 命令确定此功能是否可用。有关 AWS Glue 的更多信息，请参阅《AWS Glue 开发人员指南》中的什么是 AWS Glue？。

> **Note**
>
> 查询 AWS Glue Data Catalog 仅在 Amazon Redshift RG 或 RA3 节点类型集群和 Amazon Redshift Serverless 中才受支持。

您可以使用以下 SQL 命令配置您的数据仓库并查看编目的 AWS Glue 数据库对象：

- SHOW – 显示是否为当前连接的数据仓库安装了 awsdatacatalog。例如，要显示 data_catalog_auto_mount 参数值，请运行：

```
SHOW data_catalog_auto_mount;
```

  有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 SHOW。

- ALTER SYSTEM – 更改 data_catalog_auto_mount 的系统级配置。例如，要将 data_catalog_auto_mount 参数值更改为 on，请运行：

```
ALTER SYSTEM SET data_catalog_auto_mount = on;
```

  当重新引导预置集群或自动暂停并恢复无服务器工作组时，更改将生效。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 ALTER SYSTEM。

- SHOW SCHEMAS – 显示架构列表。数据库中名为 awsdatacatalog 的架构代表 AWS Glue Data Catalog 中编目的 AWS Glue 数据库。例如，要显示这些架构，请运行：

```
SHOW SCHEMAS FROM DATABASE awsdatacatalog;
```

  有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 SHOW SCHEMAS。

- SHOW TABLES – 显示架构中表的列表。例如，要显示名为 awsdatacatalog 的 AWS Glue Data Catalog 数据库中的表（这些表位于架构 myglue 中），请运行：

```
SHOW TABLES FROM SCHEMA awsdatacatalog.myschema;
```

  有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 SHOW TABLES。

- SHOW COLUMNS – 显示表中列的列表。例如，要显示名为 awsdatacatalog 的 AWS Glue Data Catalog 数据库中的列（这些列位于架构 myglue 和表 mytable 中），请运行：

```
SHOW COLUMNS FROM TABLE awsdatacatalog.myglue.mytable;
```

  有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 SHOW COLUMNS。

向 IAM 用户或角色授予查询 AWS Glue Data Catalog 的权限

1. 在树视图窗格中，使用数据库用户名和密码身份验证方法，连接到预置集群或无服务器工作组中的初始数据库。例如，使用您在创建集群或工作组时使用的管理员用户和密码连接到 dev 数据库。
2. 在编辑器选项卡中，运行以下 SQL 语句以授予 IAM 用户对 AWS Glue Data Catalog 的访问权限。

   ```
   GRANT USAGE ON DATABASE awsdatacatalog to "IAM:myIAMUser"
   ```

   其中，*IAM:myIAMUser* 是您想要向其授予对 AWS Glue Data Catalog 的使用权限的 IAM 用户。或者，您可以向 IAM 角色的 *IAMR:myIAMRole* 授予使用权限。
3. 在树视图窗格中，编辑或删除与您之前创建的集群或工作组的连接。通过以下方式之一连接到您的集群或工作组：
   - 要从集群访问 awsdatacatalog 数据库，您必须使用身份验证方法使用您的 IAM 身份的临时凭证。有关此身份验证方法的更多信息，请参阅连接到 Amazon Redshift 数据库。您的查询编辑器 v2 管理员可能需要配置账户的账户设置，才能在连接窗口中显示此身份验证方法。
   - 要从工作组访问 awsdatacatalog 数据库，必须使用身份验证方法联合用户。有关此身份验证方法的更多信息，请参阅连接到 Amazon Redshift 数据库。
4. 通过授予的权限，您可以使用 IAM 身份对您的 AWS Glue Data Catalog 运行 SQL。

连接后，您可以使用查询编辑器 v2 查询在 AWS Glue Data Catalog 中编目的数据。在查询编辑器 v2 树视图窗格上，选择集群或工作组和 awsdatacatalog 数据库。在编辑器或笔记本窗格中，确认选择了正确的集群或工作组。选择的数据库应该是初始 Amazon Redshift 数据库，例如 dev。有关编写查询的信息，请参阅使用 Amazon Redshift 编写查询和Amazon Redshift 中的笔记本。预留名为 awsdatacatalog 的数据库，用于引用您账户中的外部数据目录数据库。对 awsdatacatalog 数据库的查询只能是只读的。使用由三部分组成的表示法来引用 SELECT 语句中的表。其中第一部分是数据库名称，第二部分是 AWS Glue 数据库名称，第三部分是 AWS Glue 表名称。

```
SELECT * FROM awsdatacatalog.<aws-glue-db-name>.<aws-glue-table-name>;
```

您可以执行各种场景来读取 AWS Glue Data Catalog 数据并填充 Amazon Redshift 表。

以下示例 SQL 联接了在 AWS Glue 中定义的两个表。

```
SELECT pn.emp_id, alias, role, project_name
FROM "awsdatacatalog"."empl_db"."project_name_table" pn,
"awsdatacatalog"."empl_db"."project_alias_table" pa
```

```
WHERE pn.emp_id = pa.emp_id;
```

以下示例 SQL 创建了一个 Amazon Redshift 表，并使用来自两个 AWS Glue 表联接的数据填充该表。

```
CREATE TABLE dev.public.glue AS
SELECT pn.emp_id, alias, role, project_name
FROM "awsdatacatalog"."empl_db"."project_name_table" pn,
"awsdatacatalog"."empl_db"."project_alias_table" pa
WHERE pn.emp_id = pa.emp_id;
```

### 查询 Amazon S3 表（预览版）

可以使用查询编辑器 V2 查询挂载到 AWS Glue Data Catalog 的 Amazon S3 表目录中保存的数据。Amazon S3 表目录在创建时挂载到 AWS Glue Data Catalog，并自动以外部数据库的形式出现在相同 AWS 区域中同一个账户下的所有预置集群和无服务器工作组上。有关使用 Amazon Redshift 访问 Amazon S3 表类数据存储服务的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的 Accessing Amazon S3 tables with Amazon Redshift。

## 查询数据湖

可以按照本教程中的任务集查询 Amazon S3 数据湖中的数据。首先，您可以创建一个外部架构来引用在 AWS Glue Data Catalog 中的外部数据库。然后，您可以在 Simple Storage Service（Amazon S3）数据湖中查询数据。

### 演示：查询数据湖

要了解如何查询数据湖的演示，请观看以下视频。使用 Amazon Redshift 查询编辑器 v2 查询数据湖。

### 先决条件

在查询编辑器 v2 中使用数据湖之前，请确认您在 Amazon Redshift 环境中进行了以下设置：

- 使用 AWS Glue 爬取您的 Amazon S3 数据并为 AWS Lake Formation 启用 Data Catalog。
- 使用为 AWS Lake Formation 启用的 AWS Glue Data Catalog，为 Amazon Redshift 创建 IAM 角色。有关此过程的详细信息，请参阅使用为 AWS Lake Formation 启用的 AWS Glue Data Catalog 为 Amazon Redshift 创建 IAM 角色。有关使用 Redshift Spectrum 和 Lake Formation 的详细信息，请参阅将 Redshift Spectrum 与 AWS Lake Formation 结合使用。
- 授予对表的 SELECT 权限，以便在 Lake Formation 数据库中进行查询。有关此过程的详细信息，请参阅授予对表的 SELECT 权限以在 Lake Formation 数据库中进行查询。

您可以在 Lake Formation 控制台（https://console.aws.amazon.com/lakeformation/）中的权限部分的数据湖权限页面，验证该 IAM 角色、AWS Glue 数据库和表具有正确的权限。

- 确认连接的用户有权在 Amazon Redshift 数据库中创建架构以及访问数据湖中的数据。在查询编辑器 v2 中连接到数据库时，您可以选择包含凭证的身份验证方法（可以是数据库用户或 IAM 用户）。连接的用户必须具有合适的权限和数据库权限，例如 `superuser`。创建集群或工作组的 Amazon Redshift `admin` 用户拥有 `superuser` 权限，可以创建架构和管理 Redshift 数据库。有关使用查询编辑器 v2 连接到数据库的更多信息，请参阅 连接到 Amazon Redshift 数据库。

## 创建外部架构

要查询 Amazon S3 数据湖中的数据，请先创建外部架构。外部架构引用了 AWS Glue Data Catalog 中的外部数据库。

1. 在查询编辑器 v2 的编辑器视图中，选择

   创建，然后选择架构。
2. 输入架构名称。
3. 对于架构类型，选择外部。
4. 在 Data Catalog 详细信息下，区域默认为 Redshift 数据库所在的 AWS 区域。
5. 选择外部架构将映射的 AWS Glue 数据库，该数据库还应包含对 AWS Glue 表的引用。
6. 为 Amazon Redshift 选择相应的 IAM 角色，该角色应具有从 Amazon S3 查询数据所需的权限。
7. （可选）选择具有 Data Catalog 权限的 IAM 角色。
8. 选择创建架构。

   该架构将显示在树视图面板中您的数据库下方。

创建架构时，如果您收到数据库权限被拒绝错误，请检查连接的用户是否具有创建架构的数据库权限。

## 在 Simple Storage Service（Amazon S3）数据湖中查询数据

您可以使用在前一个过程中创建的架构。

1. 在树视图面板中，选择该架构。
2. 请选择一个表来查看表定义。此时系统将显示表列和数据类型。

3. 要查询表，请选择该表并使用上下文菜单（右键单击），选择选择表以生成查询。
4. 在编辑器中运行查询。

   以下示例 SQL 由查询编辑器 v2 生成，用于查询名为 `flightscsv` 的 AWS Glue 表中的所有行。为简化起见，只显示了输出中的部分列和行。

```
SELECT * FROM "dev"."mydatalake_schema"."flightscsv";

year    quarter   month   dom  day_of_week   fl_date    unique_carrier  airline_id
  carrier   tail_num   fl_num
2016    4         10      19   3             10/19/16   OO              20304
  OO         N753SK    3086
2016    4         10      19   3             10/19/16   OO              20304
  OO         N753SK    3086
2016    4         10      19   3             10/19/16   OO              20304
  OO         N778SK    3087
2016 4          10      19   3             10/19/16   OO              20304
 OO         N778SK    3087
...
```

## 数据共享

您可以创建数据共享，以便其他集群上的用户可以查询数据。包含您要共享的数据的集群称为创建者集群。您可以在创建者集群上为要共享的数据库对象创建数据共享。您可以共享 Schema、表、视图和 SQL 用户定义的函数 (UDF)。您要与其共享数据的集群称为使用者集群。在使用者集群上，您可以利用数据共享创建数据库。然后，使用者集群上的用户可以查询数据。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的数据共享入门。

### 创建数据共享

您可以在要用作创建者集群的集群上创建数据共享。要详细了解数据共享的注意事项，请参阅中的《Amazon Redshift 数据库开发人员指南》中的 Amazon Redshift 中的数据共享注意事项。

1. 选择要使用的创建者集群上的数据库。
2. 创建数据共享。例如：

```
create datashare mysource;
```

3. 设置数据共享的权限。例如：

```
grant alter, share on datashare mysource to admin;
```

4. 设置要共享的数据库对象的权限。例如：

```
alter datashare mysource add schema public;
```

```
alter datashare mysource add table public.event;
```

5. 设置使用者集群命名空间的权限以访问数据共享。例如：

```
grant usage on datashare mysource to namespace '2b12345-1234-5678-9012-
bb1234567890';
```

## 显示数据共享

您可以显示您在创建者集群上创建的数据共享。

1. 选择创建者集群。
2. 显示数据共享。例如：

```
show datashares;
```

```
share_name share_owner source_database  consumer_database share_type createdate
 is_publicaccessible share_acl producer_account producer_namespace
test_datashare 100  db_producer  NULL   OUTBOUND 2/15/2022  FALSE  admin
 123456789012  p1234567-8765-4321-p10987654321
```

## 创建使用者数据库

在使用者集群上，您可以利用数据共享创建数据库。这些步骤描述了如何在同一个账户中的两个集群之间共享数据。有关跨 AWS 账户共享数据的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的跨 AWS 账户共享数据。

您可以使用 SQL 命令或查询编辑器 v2 树状视图面板来创建数据库。

使用 SQL

1. 利用数据共享为您的账户和创建者集群的命名空间创建一个数据库。例如：

```
create database share_db from datashare mysource of account '123456789012'
 namespace 'p1234567-8765-4321-p10987654321';
```

2. 设置权限，以便用户可以访问数据库和 Schema。例如：

```
grant usage on database share_db to usernames;
```

```
grant usage on schema public to usernames;
```

使用查询编辑器 v2 树状视图面板

1. 选择  创建，然后选择数据库。
2. 输入数据库名称。
3. （可选）选择用户和组，然后选择数据库用户。
4. 选择使用数据共享创建。
5. 选择数据共享。
6. 选择创建数据库。

   新的  数据共享数据库将在查询编辑器 v2 树状视图面板中显示。
7. 设置权限，以便用户可以访问数据库和 Schema。例如：

```
grant usage on database share_db to usernames;
```

```
grant usage on schema public to usernames;
```

### 查询数据共享对象

在使用者集群上，您可以使用以三部分表示法表示的完全限定对象名来查询数据共享对象：数据库、架构和对象的名称。

1. 在查询编辑器 v2 树状视图面板中，选择该 Schema。
2. 请选择一个表来查看表定义。

   此时系统将显示表列和数据类型。
3. 要查询表，请选择表并使用上下文菜单（右键单击）选择选择表。
4. 使用 SELECT 命令查询表。例如：

```
select top 10 * from test_db.public.event;
```

## 使用查询编辑器 v2 的计划查询

使用 Amazon Redshift 查询编辑器 v2，您可以自动执行 SQL 查询，使其按计划运行。计划查询是按指定时间或间隔自动运行的 SQL 语句，可让您高效管理重复性数据操作和分析任务。如果您想在 Amazon Redshift 环境中简化批处理、生成定期报告或维护数据管道，则可能需要计划查询。

计划查询有助于自动执行提取、转换和加载（ETL）工作流，用最新见解刷新控制面板，以及操作各种数据管理例程。以下页面详细介绍了创建、配置和管理计划查询以优化 Amazon Redshift 工作负载的过程。

### 使用查询编辑器 v2 创建查询计划

您可以使用 Amazon Redshift 查询编辑器 v2 创建运行 SQL 语句的计划。您创建一个计划，以便按照与您的业务需求相匹配的时间间隔运行 SQL 语句。当到了运行计划查询的时间时，查询由 Amazon EventBridge 启动并使用 Amazon Redshift Data API。

创建计划以运行 SQL 语句

1. 在编辑器

   视图中，选择



计划以创建运行 SQL 语句的计划。

2. 在定义计划时，您需要提供以下信息。

   - 代入运行查询所需权限的 IAM 角色。此 IAM 角色还附加到您的集群或工作组。
   - AWS Secrets Manager 或用于授权访问您的集群或工作组的临时凭证的身份验证值。数据 API 支持这些身份验证方法。有关更多信息，请参阅 对计划查询进行身份验证。
   - 您的数据库所在的集群或工作组。
   - 包含要查询的数据的数据库名称。
   - 计划查询的名称及其描述。查询编辑器 v2 将在您提供的计划查询名称前加上“QS2-”前缀。查询编辑器 v1 将在其计划查询名称前加上“QS-”前缀。
   - 要按计划运行的 SQL 语句。
   - 计划频率和重复选项，或定义计划的 cron 格式的值。有关更多信息，请参阅《Amazon CloudWatch Events User Guide》中的 Cron Expressions。
   - （可选）您可以启用标准 Amazon SNS 通知来监控计划查询。您可能需要确认您向 Amazon SNS 通知提供的电子邮件地址。在您收到的电子邮件中查找用于确认接收 Amazon SNS 通知的电子邮件地址的链接。有关更多信息，请参阅《Amazon Simple Notification Service 开发人员指南》中的电子邮件通知。如果您的查询正在运行，但您没有看到 SNS 主题中发布的消息，请参阅《Amazon EventBridge 用户指南》中的我的规则正在运行，但我没有看到任何消息发布到我的 Amazon SNS 主题。

3. 选择计划查询以保存和激活计划，并将计划添加到计划查询视图中的查询列表。

## 计划查询



视图列出了您的集群和工作组的所有计划查询。使用此视图，您可以显示计划查询的详细信息、激活或停用计划、编辑计划以及删除计划查询。查看查询详细信息时，还可以查看使用计划运行查询的历史记录。

> **Note**
>
> 计划查询的运行仅在计划历史记录列表中列出 24 小时。按计划运行的查询不会出现在查询编辑器 v2 的查询历史记录视图中。

安排查询的演示

有关安排查询的演示，请观看以下视频。安排查询的视频演示。

## 设置计划查询的权限

要计划查询，定义计划的 AWS Identity and Access Management（IAM）用户以及与计划关联的 IAM 角色必须配置了使用 Amazon EventBridge 和 Amazon Redshift Data API 的 IAM 权限。要接收来自计划查询的电子邮件，还必须配置可选指定的 Amazon SNS 通知。

以下内容描述了使用 AWS 托管式策略提供权限的任务，但根据您的环境，您可能需要缩小允许的权限范围。

对于登录到查询编辑器 v2 的 IAM 用户，请使用 IAM 控制台（https://console.aws.amazon.com/iam/）编辑 IAM 用户。

- 除了运行 Amazon Redshift 和查询编辑器 v2 操作的权限外，还要将 `AmazonEventBridgeFullAccess` 和 `AmazonRedshiftDataFullAccess` AWS 托管式策略附加到 IAM 用户。
- 或者，为角色分配权限并将该角色分配给用户。

  附加一个策略，对于您在定义计划查询时指定的 IAM 角色的资源 ARN，该策略将允许 `sts:AssumeRole` 权限。有关代入角色的更多信息，请参阅《IAM 用户指南》中的向用户授予切换角色的权限。

  以下示例显示了代入账户 `123456789012` 中的 IAM 角色 `myRedshiftRole` 的权限策略。IAM 角色 `myRedshiftRole` 也是附加到运行计划查询的集群或工作组的 IAM 角色。

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AssumeIAMRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": [
                "arn:aws:iam::123456789012:role/myRedshiftRole"
            ]
        }
```

```
    ]
}
```

更新用于计划查询的 IAM 角色的信任策略，以允许 IAM 用户代入此角色。

```
{
            "Sid": "AssumeRole",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/myIAMusername"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

对于您指定为允许运行计划查询的 IAM 角色，请使用 IAM 控制台（https://console.aws.amazon.com/iam/）编辑 IAM 角色。

- 将 AmazonRedshiftDataFullAccess 和 AmazonEventBridgeFullAccess AWS 托管式策略附加到 IAM 角色。AmazonRedshiftDataFullAccess 托管式策略只允许使用键 RedshiftDataFullAccess 标记的 Redshift Serverless 工作组具有 redshift-serverless:GetCredentials 权限。

## 对计划查询进行身份验证

当您计划查询时，在 SQL 运行时使用下列身份验证方法之一。每种方法都需要查询编辑器 v2 上的不同输入组合。用于运行 SQL 语句的数据 API 支持这些身份验证方法。

用于运行查询的数据库用户或角色必须具有必要的数据库权限。例如，要授予对于表 mytable 的 IAMR:MyRedshiftQEv2Scheduler 权限，请运行以下 SQL 命令。

```
GRANT all ON TABLE mytable TO "IAMR:MyRedshiftQEv2Scheduler";
```

要查看集群或工作组中的数据库用户列表，请查询系统视图 PG_USER_INFO。

> **Note**
>
> 您为其计划查询的任何 Redshift Serverless 工作组都必须使用键 `RedshiftDataFullAccess` 进行标记。有关更多信息，请参阅 [授予对 Amazon Redshift 数据 API 的访问权限](#)。
> 作为标记工作组的替代方法，您可以向 IAM 角色（随计划指定）添加允许 `redshift-serverless:GetCredentials` 的内联策略。例如：
>
> JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "UseTemporaryCredentialsForAllServerlessWorkgroups",
            "Effect": "Allow",
            "Action": "redshift-serverless:GetCredentials",
            "Resource": [
                "arn:aws:redshift-serverless:*:*:workgroup/*"
            ]
        }
    ]
}
```

AWS Secrets Manager

使用此方法，为存储在 AWS Secrets Manager 中的 secret-arn 提供一个密钥值。此密钥包含用于连接到数据库的凭证。在创建集群或工作组时，您可能已经使用适当的凭证创建了密钥。密钥必须使用键 `RedshiftDataFullAccess` 进行标记。如果标签键尚不存在，请使用 AWS Secrets Manager 控制台进行添加。有关创建密钥的信息，请参阅[为数据库连接凭证创建密钥](#)。

有关最低权限的更多信息，请参阅《AWS Secrets Manager 用户指南》中的[使用 AWS Secrets Manager 创建和管理密钥](#)。

临时凭证

使用此方法，在连接到集群中的数据库时，需提供您的数据库名称和数据库用户值。在连接到工作组中的数据库时，只需提供您的数据库名称。

连接到集群时，AmazonRedshiftDataFullAccess 策略允许名为 redshift_data_api_user 的数据库用户拥有对 redshift:GetClusterCredentials 的权限。如果要使用其他数据库用户运行 SQL 语句，请向附加到您集群的 IAM 角色添加策略以允许 redshift:GetClusterCredentials。以下示例策略允许数据库用户 awsuser 和 myuser。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "UseTemporaryCredentialsForAllDbUsers",
            "Effect": "Allow",
            "Action": "redshift:GetClusterCredentials",
            "Resource": [
                "arn:aws:redshift:*:*:dbuser:*/awsuser",
                "arn:aws:redshift:*:*:dbuser:*/myuser"
            ]
        }
    ]
}
```

## 设置查看计划查询历史记录的权限

要允许用户查看计划查询历史记录，请编辑 IAM 角色（随计划指定）信任关系以添加权限。

以下是 IAM 角色中的信任策略示例，该策略允许 IAM 用户 *myIAMusername* 查看计划查询历史记录。您可以选择允许 IAM 角色拥有 sts:AssumeRole 权限，而不是允许 IAM 用户拥有此权限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "redshift.amazonaws.com",
```

```
                    "redshift-serverless.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "events.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        },
        {
            "Sid": "AssumeRole",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/myIAMusername"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

## 监控计划查询

对于您指定用于发送电子邮件通知的 Amazon SNS 主题，请使用查询编辑器 v2 创建 Amazon SNS 主题，方法是导航到 SNS 通知部分，开启监控，然后使用创建 SNS 主题来创建主题。查询编辑器 v2 创建 Amazon SNS 主题，并将服务主体添加到 Amazon EventBridge 的访问策略中。以下是在 Amazon SNS 主题中创建的示例访问策略。在示例中，使用了 AWS 区域 *us-west-2*、AWS 账户 *123456789012* 以及 Amazon SNS 主题 *select-version-pdx-testunload*。

JSON

```
{
  "Version":"2012-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "Allow_Publish_Events",
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": "events.amazonaws.com"
      },
      "Action": "sns:Publish",
      "Resource": "arn:aws:sns:us-west-2:123456789012:select-version-pdx-
testunload"
    }
  ]
}
```

计划查询运行时，Amazon SNS 会发送 AWS 通知电子邮件。以下示例显示了使用 Amazon SNS 通知主题 *may25a-SNS*，为 AWS 账户 *123456789012* 中在 AWS 区域 *eu-north-1* 上运行的计划查询 *QS2-may25a* 发送到 *myemail@example.com* 的电子邮件。

```
{"version":"0","id":"8e4323ec-5258-7138-181b-91290e30ff9b","detail-type":"Scheduled
 Event","source":"aws.events","account":"123456789012","time":"2023-05-25T15:22:00Z",
                    "region":"eu-north-1","resources":["arn:aws:events:eu-
north-1:123456789012:rule/QS2-may25a"],"detail":{}}

--
If you wish to stop receiving notifications from this topic, please click or visit the
 link below to unsubscribe:
https://sns.eu-north-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:eu-
north-1:123456789012:may25a-SNS:0c1a3d05-39c2-4507-
bc3d-47250513d7b0&Endpoint=myemail@example.com

Please do not reply directly to this email. If you have any questions or comments
 regarding this email, please contact us at https://aws.amazon.com/support
```

## 计划查询设置故障排除

如果您在计划查询时遇到问题，请考虑以下几点。

查询未运行

检查计划中使用的 IAM 角色是否有权获取临时集群凭证。预置集群的权限为 `redshift:GetClusterCredentialsWithIAM`。Redshift Serverless 工作组的权限为 `redshift-serverless:GetCredentials`。

计划历史记录未显示

用于登录 AWS 控制台的 IAM 用户或 IAM 角色未添加到用于计划查询的 IAM 角色的信任策略中。

当使用 AWS Secrets Manager 供计划的查询进行连接时，请确认已使用键 `RedshiftDataFullAccess` 标记密钥。

如果计划查询使用的是 AWS Secrets Manager 连接，则用于计划查询的 IAM 角色必须具有附加到该角色的等效托管策略 `SecretsManagerReadWrite`。

查询历史记录状态为 **`Failed`**

查看 SYS_QUERY_HISTORY 系统视图，了解有关查询失败原因的详细信息。一个常见问题是，用于运行查询的数据库用户或角色可能没有运行 SQL 所需的权限。有关更多信息，请参阅 对计划查询进行身份验证。

以下 SQL 查询 SYS_QUERY_HISTORY 视图以返回失败的查询。

```
SELECT user_id, query_id, transaction_id, session_id, database_name, query_type,
 status, error_message, query_text
FROM sys_query_history
WHERE status = 'failed';
```

要了解失败的特定计划查询的详细信息，请参阅使用 AWS CloudShell 查看计划查询的结果。

## 使用 AWS CloudShell 查看计划查询的结果

您可以使用 AWS CloudShell 查找有关计划查询的详细信息。您必须具有适当的权限才能运行 AWS CLI 命令，如以下过程所示。

查看计划查询的结果

1. 在 AWS 控制台上，打开 AWS CloudShell 命令提示符。有关 AWS CloudShell 的更多信息，请参阅《AWS CloudShell User Guide》中的What is AWS CloudShell。
2. 代入计划查询的 IAM 角色。要代入该角色，请在查询编辑器 v2 中找到与计划查询关联的 IAM 角色，然后在 AWS CloudShell 中通过 AWS CLI 命令使用它。例如，对于角色 `scheduler`，输入 AWS STS 命令以代入计划查询所用的角色。

   ```
   aws sts assume-role --role-arn "arn:aws:iam::123456789012:role/scheduler" --role-
   session-name "scheduler-test"
   ```

   返回的凭证与以下内容类似。

   ```
   "Credentials": {
   ```

```
"AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
"SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
"SessionToken": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY...",
"Expiration": "2023-08-18T18:19:44+00:00"
},
"AssumedRoleUser": {
"AssumedRoleId": "AROA35B2NH6WBTP7ONL4E:scheduler-test",
"Arn": "arn:aws:sts::123456789012:assumed-role/scheduler/scheduler-test"
}
}
```

3. 使用代入 IAM 角色时显示的凭证在 AWS CLI 中创建环境变量。您必须在这些令牌到期之前使用它们。例如，您可以在 AWS CloudShell 中输入以下内容。

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
export AWS_SESSION_TOKEN=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY...
```

4. 要查看失败查询的错误，请运行 AWS CLI 命令以描述语句。SQL 语句的 ID 来自查询编辑器 v2 中计划查询的计划历史记录部分显示的 ID。

```
aws redshift-data describe-statement --id 130d2620-05d2-439c-b7cf-815d9767f513
```

在本例中，计划的 SQL `select * from users limit 100` 会导致一个 SQL 错误，即 `users` 表不存在。

```
{
"CreatedAt": "2023-08-18T17:39:15.563000+00:00",
"Duration": -1,
"Error": "ERROR: relation \"users\" does not exist",
"HasResultSet": false,
"Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"QueryString": "select * from users limit 100\n—RequestID=a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222; TraceID=1-633c5642-4039308d03f3a0ba53dbdf6f",
"RedshiftPid": 1073766651,
"RedshiftQueryId": 0,
"ResultRows": -1,
"ResultSize": -1,
"Status": "FAILED",
"UpdatedAt": "2023-08-18T17:39:16.116000+00:00",
"WorkgroupName": "default"
```

```
}
```

## 可视化查询结果

运行查询并显示结果后，可以打开图表以显示当前结果页的图形可视化。您可以使用以下控件来定义图表的内容、结构和外观：



跟踪

表示图表中的一组相关图形标记。可以在图表中定义多个跟踪。

类型

您可以将跟踪类型定义为以下类型之一来表示数据：

- 适用于散点的散点图或气泡图。
- 用垂直条或水平条表示数据类别的条形图。
- 用于定义填充区域的区域图。
- 使用条形表示频率分布的直方图。
- 使用圆形表示数据的饼图，其中每个切片代表所占整体的百分比。
- 用于表示流程不同阶段的数据的漏斗或漏斗区域图。
- OHLC（开盘、高、低、收盘）图通常用于财务数据，表示沿 x 轴的开盘、高、低和收盘值，这通常表示时间间隔。
- K 线图表示时间线上类别的一系列值。
- 瀑布图，表示初始值如何通过一系列中间值而增加或减少。值可以表示时间间隔或类别。
- 折线图表示值随着时间的推移而变化。

X 轴

您可以指定一个表列，其中包含要沿 X 轴绘制的值。包含描述性值的列通常表示维度数据。包含量化值的列通常表示实际数据。

Y 轴

您可以指定一个表列，其中包含要沿 Y 轴绘制的值。包含描述性值的列通常表示维度数据。包含量化值的列通常表示实际数据。

子图

您可以定义图表数据的其它表示形式。

转换

您可以定义转换以筛选跟踪数据。您可以使用拆分转换以显示来自单个源跟踪的多个跟踪。您可以使用聚合转换以将跟踪显示为平均值或最小值。您可以使用排序转换以对跟踪进行排序。

一般外观

您可以设置背景颜色、边距颜色、设计调色板的色阶、文本样式和大小、标题样式和大小以及模式栏的默认值。您可以定义拖动、单击和悬停的交互。您可以定义元文本。您可以定义跟踪、轴、图例和注释的默认外观。

创建图表

1. 运行查询并获取结果。
2. 启用图表。
3. 选择跟踪，然后开启可视化您的数据。
4. 从以下值中选择一种图表样式：
   - 散点图
   - 条形图
   - 区域图
   - 直方图
   - 饼图
   - 漏斗图
   - 漏斗区域图
   - OHLC（开盘、高、低、收盘）
   - K 线图
   - 瀑布图
   - 折线图
5. 选择样式以自定义外观，包括颜色、轴、图例和注释。您可以添加文本、形状和图像。
6. 选择注释添加文本、形状和图像。
7. 要更新显示，请选择更新。选择全屏以扩大图表显示范围。

## 示例：创建饼图以可视化查询结果

以下示例使用示例数据库中的销售表。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的示例数据库。

以下是为饼图提供数据而运行的查询。

```
select top 5 eventname, count(salesid) totalorders, sum(pricepaid) totalsales
from sales, event
where sales.eventid=event.eventid group by eventname
order by 3;
```

按总销售额作为顶事件创建饼图

1. 运行查询。
2. 在查询结果区域中，打开图表。
3. 选择跟踪。
4. 对于类型，选择饼图。
5. 对于值，选择 totalsales。
6. 对于标签，选择 eventname。
7. 选择样式，然后选择普通。
8. 在颜色刻度下，选择分类，然后选择 Pastel2。



## 示例：创建用于比较收入和销售额的组合图

执行此示例中的步骤创建一个图表，该图表将收入数据的条形图和销售额数据的折线图组合在一起。以下示例使用 tickit 示例数据库中的 Sales 表。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的示例数据库。

以下是为图表提供数据而运行的查询。

```
select eventname, total_price, total_qty_sold
from  (select eventid, total_price, total_qty_sold, ntile(1000) over(order by
 total_price desc) as percentile
       from (select eventid, sum(pricepaid) total_price, sum(qtysold) total_qty_sold
             from   tickit.sales
             group by eventid)) Q, tickit.event E
       where Q.eventid = E.eventid
       and percentile = 1
order by total_price desc;
```

创建用于比较收入和销售额的组合图

1. 运行查询。
2. 在查询结果区域中，打开图表。
3. 在 trace o 下，对于类型，选择条形图。
4. 对于 X，选择 eventname。
5. 对于 Y，选择 total_price。

   条形图将沿 X 轴显示事件名称。
6. 在样式下，选择跟踪。
7. 对于名称，输入收入。
8. 在样式下，选择轴。
9. 对于标题，选择 Y 然后输入收入。

   收入标签将显示在左侧 Y 轴上。
10. 在结构下，选择跟踪。
11. 选择

    跟踪。

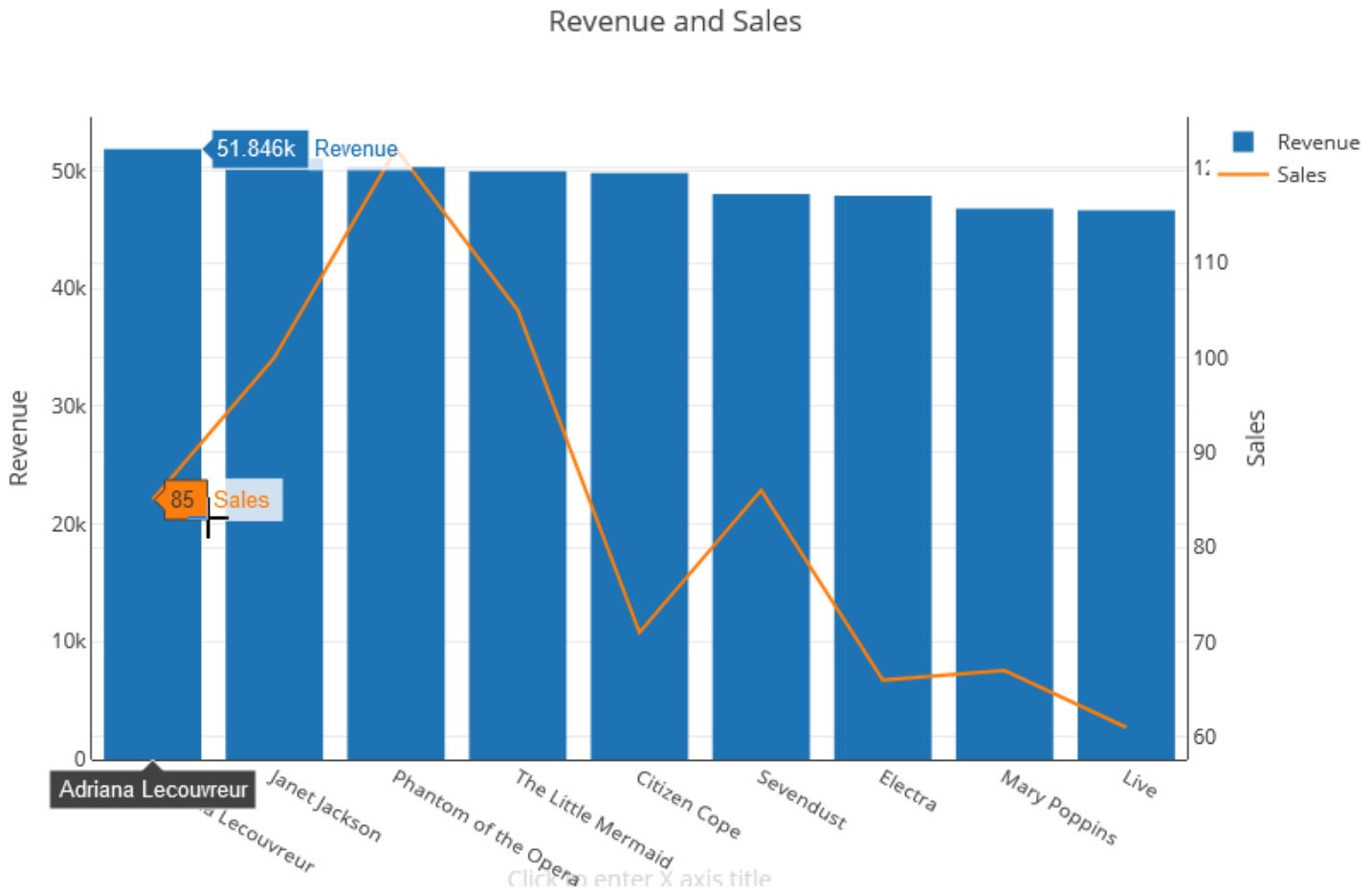
系统将显示跟踪 1 选项。

12. 对于类型，选择 线形图。
13. 对于 X，选择 eventname。
14. 对于 Y，选择 total_qty_sold。
15. 在要使用的轴下，为Y 轴选择

    +

    。

    Y 轴将显示 Y2。
16. 在样式下，选择轴。
17. 在标题下，选择 Y2。
18. 对于名称，输入 Sales。
19. 在线形图下，选择 Y:Sales。
20. 在轴线下，选择显示，对于位置，选择右侧。



### 演示：使用 Amazon Redshift 查询编辑器 v2 构建可视化

要了解如何构建可视化项的演示，请观看以下视频。使用 Amazon Redshift 查询编辑器 v2 构建可视化。

## 以团队形式协作和共享

您可以与您的团队共享查询。

团队是针对一组协作并共享查询编辑器 v2 资源的用户定义的。管理员可以通过向 IAM 角色添加标签来创建团队。有关更多信息，请参阅 使用查询编辑器 v2 所需的权限。

### 保存和浏览查询

在与团队共享查询之前，请保存查询。您可以查看和删除已保存的查询。

保存查询

1. 准备查询然后选择保存。
2. 输入查询的标题。
3. 选择保存。

浏览保存的查询

1. 从导航窗格中，选择查询。
2. 您可以查看以下查询：我的查询、由我分享的查询，或者分享给我的团队的查询。这些查询可以显示为单个查询或存在您创建的文件夹中。

### 共享查询

您可以与您的团队共享您的查询。您还可以查看已保存查询的历史记录和管理查询版本。

如要与团队共享查询，请确保您将主要标签 `sqlworkbench-team` 设置为与账户中其它团队成员相同的值。例如，管理员可以为会计部门的每个人将该值设置为 `accounting-team`。有关示例，请参阅使用查询编辑器 v2 所需的权限。

与团队共享查询

1. 从导航窗格中，选择查询。

2. 打开要共享的查询的上下文（右键单击）菜单，然后选择与我的团队分享。
3. 选择要与之共享查询的一个或多个团队，然后选择保存共享选项。

### 管理查询版本

每次保存 SQL 查询时，查询编辑器 v2 都会将其保存为新版本。您可以浏览早期的查询版本、保存查询副本或恢复查询。

管理查询版本

1. 从导航窗格中，选择查询。
2. 打开要处理查询的上下文（右键单击）菜单。
3. 选择版本历史记录以打开查询的版本列表。
4. 在版本历史记录页面上，您可以执行以下操作：
   - 恢复为选定 – 恢复到选定的版本并继续使用此版本。
   - 将选定选项另存为 – 在编辑器中创建新的查询。

## 使用 Amazon Redshift 查询编辑器 v1 查询数据库

使用查询编辑器是在由 Amazon Redshift 集群托管的数据库上运行查询的简单方法。创建集群后，可以使用 Amazon Redshift 控制台上的查询编辑器立即运行查询。

> **Note**
>
> 您无法使用此原始查询编辑器在 Amazon Redshift Serverless 中查询数据。而应使用 Amazon Redshift 查询编辑器 v2。

2021 年 2 月，部署了更新的查询编辑器，并更改了使用查询编辑器的授权权限。新的查询编辑器使用 Amazon Redshift 数据 API 来运行查询。作为 AWS 托管式 AWS Identity and Access Management (IAM) 策略的 `AmazonRedshiftQueryEditor` 策略已更新为包含必要的权限。如果您有自定义 IAM 策略，请务必更新它。将 `AmazonRedshiftQueryEditor` 用作指南。对 `AmazonRedshiftQueryEditor` 的更改包括以下内容：

- 管理查询编辑器语句结果的权限需要语句拥有者用户。

- 已添加使用 Secrets Manager 连接到数据库的权限。

有关更多信息，请参阅 使用 Amazon Redshift 控制台查询编辑器所需的权限。

从新的查询编辑器连接到集群时，可以使用两种身份验证方法之一。

使用查询编辑器可以执行以下操作：

- 运行单个 SQL 语句查询。
- 将大小为 100 MB 的结果集下载到一个逗号分隔值 (CSV) 文件。
- 保存查询以供重用。您无法在欧洲（巴黎）区域、亚太地区（大阪）区域、亚太地区（香港）区域或中东（巴林）区域中保存查询。
- 查看用户定义表的查询运行时详细信息。
- 安排查询在未来运行。
- 查看您在查询编辑器中创建的查询的历史记录。
- 使用增强型 VPC 路由对集群运行查询。

## 查询编辑器注意事项

请考虑下列有关使用查询编辑器时处理查询的事项：

- 查询的最长持续时间为 24 小时。
- 查询结果的最大大小为 100 MB。如果调用返回的响应数据超过 100 MB，则调用将终止。
- 查询结果的最长保留时间为 24 小时。
- 最大查询语句大小为 100 KB。
- 集群必须在基于 Amazon VPC 服务的 Virtual Private Cloud (VPC) 中。
- 不能在查询编辑器中使用事务处理。有关事务的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 BEGIN。
- 您可以保存最长为 3000 个字符的查询。

# 使用 SQL 客户端工具连接到 Amazon Redshift 数据仓库

您可以通过 Java 数据库连接（JDBC）、Python 和开放式数据库连接（ODBC）这几种连接将 SQL 客户端工具连接到 Amazon Redshift 数据仓库。Amazon Redshift 不提供或安装任何 SQL 客户端工具

或库。要使用这些工具或库处理数据仓库中的数据，请将它们安装在客户端计算机或 Amazon EC2 实例上。您可以使用支持 JDBC、Python 或 ODBC 驱动程序的大多数 SQL 客户端工具。

使用本主题末尾的章节列表来帮助您完成将客户端计算机或 Amazon EC2 实例配置为使用 JDBC、Python 或 ODBC 连接的过程。这些主题还讨论了与服务器的客户端连接的相关安全选项。此外，还可以查找有关从 SQL 客户端工具（如 Amazon Redshift RSQL）中进行设置和连接的信息。如果您还没有可以使用的商业智能工具，则可以尝试使用这些工具。还可以通过本节了解如何连接到您的数据。最后，如果您在尝试连接到数据仓库时遇到问题，可以查看故障排除信息以确定解决方案。

## 关于连接客户端工具的建议

如果您使用 IP 地址连接到 Redshift 集群，则当发生中断或连接丢失时，集群会在新的可用区（AZ）中联机，这可能会延长停机时间。但是，如果您仍希望您的应用程序使用 IP 地址连接到 Redshift，请使用附加到集群虚拟私有云（VPC）端点的私有 IP 地址。您可以在属性选项卡下的网络和安全性中的集群详细信息中找到此信息。

> **Note**
>
> 如果您的应用程序使用领导节点的 IP 地址来访问 Redshift 集群，则建议的最佳实践是将其更改为使用集群端点 URL。有关更多信息，请参阅在 Amazon Redshift 中配置连接。

主题


- 在 Amazon Redshift 中配置连接
- 用于发现元数据的最佳实践
- 配置连接的安全选项
- 通过客户端工具和代码连接
- 使用身份验证配置文件连接到 Amazon Redshift
- 解决 Amazon Redshift 中的连接问题


## 在 Amazon Redshift 中配置连接

在以下章节中，了解如何配置 JDBC、Python 和 ODBC 连接以从 SQL 客户端工具连接到集群。此部分介绍如何设置 JDBC、Python 和 ODBC 连接。它还介绍了如何使用安全套接字层 (SSL) 和服务器证书来加密客户端和服务器之间的通信。

## 适用于 Amazon Redshift 的 JDBC、Python 和 ODBC 驱动程序

要处理集群中的数据，您必须具有 JDBC、Python 或 ODBC 驱动程序，以便从客户端计算机或实例进行连接。对应用程序进行编码以使用 JDBC、Python 或 ODBC 数据访问 API 操作，并使用支持 JDBC 或 ODBC 的 SQL 客户端工具。

Amazon Redshift 提供 JDBC、Python 和 ODBC 驱动程序以供下载。这些驱动程序受 支持 支持。PostgreSQL 驱动程序未经过测试，也不受 Amazon Redshift 团队的支持。连接到 Amazon Redshift 集群时，请使用 Amazon Redshift 特定的驱动程序。Amazon Redshift 驱动程序具有以下优势：

- 支持 IAM、SSO 和联合身份验证。
- 支持新的 Amazon Redshift 数据类型。
- 支持身份验证配置文件。
- 结合 Amazon Redshift 增强功能提升性能。

有关如何下载 JDBC 和 ODBC 驱动程序和配置到集群的连接的更多信息，请参阅 为 Amazon Redshift 配置 JDBC 驱动程序版本 2.x 连接、Amazon Redshift Python 连接器 和 为 Amazon Redshift 配置 ODBC 驱动程序版本 2.x 连接。

有关管理 IAM 身份的更多信息，包括 IAM 角色的最佳实践，请参阅 Amazon Redshift 中的 Identity and Access Management。

## 查找集群连接字符串

要使用 SQL 客户端工具连接到您的集群，您必须具有集群连接字符串。您可以在 Amazon Redshift 控制台中的集群详细信息页面上查找集群连接字符串。

查找集群的连接字符串

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群，然后从列表中选择集群名称以打开其详细信息。
3. 一般信息部分中提供有 JDBC URL 和 ODBC URL 连接字符串以及其他详细信息。每个字符串均基于运行集群的 AWS 区域。点击相应连接字符串旁边的图标复制该字符串。

要连接到集群端点，可以使用 DescribeClusters API 请求中的集群端点 URL。以下是集群端点 URL 的示例。

```
mycluster.cmeaswqeuae.us-east-2.redshift.amazonaws.com
```

如果您为集群设置了自定义域名，还可以使用该域名连接到您的集群。有关创建自定义域名的更多信息，请参阅设置自定义域名。

> **Note**
>
> 连接时，请勿使用集群节点的 IP 地址或 VPC 端点的 IP 地址。请务必使用 Redshift 端点以避免不必要的中断。使用端点 URL 的唯一例外是使用自定义域名时。有关更多信息，请参阅使用自定义域名进行客户端连接。

## 为 Amazon Redshift 配置 JDBC 驱动程序版本 2.x 连接

可以使用 JDBC 驱动程序版本 2.x 连接从多种第三方 SQL 客户端工具连接到 Amazon Redshift 集群。Amazon Redshift JDBC 连接器提供了一个开源解决方案。您可以浏览源代码、请求增强功能、报告问题和提供文章。

有关 JDBC 驱动程序更改的最新信息，请参阅 change log。

预设情况下，Amazon Redshift JDBC 驱动程序将配置为使用 TCP Keepalive 来防止连接超时。可以指定驱动程序开始发送 Keepalive 包的时间或通过在连接 URL 中设置相关属性来关闭该功能。有关连接 URL 的语法的更多信息，请参阅 构建连接 URL。

| 属性 | 说明 |
| --- | --- |
| `TCPKeepAlive` | 要关闭 TCP Keepalive，请将此属性设置为 `FALSE`。 |

主题

- 下载 Amazon Redshift JDBC 驱动程序版本 2.x
- 安装 Amazon Redshift JDBC 驱动程序版本 2.x
- 获取 JDBC URL
- 构建连接 URL
- 使用 Apache Maven 配置 JDBC 连接
- 配置身份验证和 SSL

- 配置日志记录
- 数据类型转换
- 使用预编译的语句支持
- JDBC 驱动程序 2.x 版本与 1.x 版本之间的差异
- 为 JDBC 驱动程序版本 2.x 创建初始化 (.ini) 文件
- JDBC 驱动程序版本 2.x 配置的选项
- JDBC 驱动程序版本 2.x 的以前版本

下载 Amazon Redshift JDBC 驱动程序版本 2.x

> **Note**
>
> Amazon Redshift JDBC 2.x 驱动程序并不是为线程安全而设计的。两个或更多线程并发尝试使用同一个连接可能会导致死锁、错误、不正确的结果或其它意外行为。
> 如果您有多线程应用程序，我们建议您同步对驱动程序的访问，以避免并发访问。

Amazon Redshift 向与 JDBC 4.2 API 兼容的工具提供驱动程序。此驱动程序的类名是 `com.amazon.redshift.Driver`。

有关如何安装 JDBC 驱动程序、引用 JDBC 驱动程序库和注册驱动程序类的详细信息，请参阅以下主题。

对于您在其上使用 Amazon Redshift JDBC 驱动程序版本 2.x 的每台计算机，请确保已安装 Java 运行时环境（JRE）8.0。

如果将 Amazon Redshift JDBC 驱动程序用于数据库身份验证，请确保您的 Java 类路径中包含 适用于 Java 的 AWS SDK 2.40.5 或更高版本。如果您没有安装 适用于 Java 的 AWS SDK，请下载带有与 JDBC 4.2 兼容的驱动程序和 AWS SDK 驱动程序相关库的 ZIP 文件：

- 与 JDBC 4.2 兼容的驱动程序版本 2.x 和 AWS SDK 驱动程序依赖库

  此 ZIP 文件包含与 JDBC 4.2 兼容的驱动程序版本 2.x 和适用于 Java 的 AWS SDK 2.x 驱动程序相关库文件。将相关 jar 文件解压缩到与 JDBC 驱动程序相同的位置。只有 JDBC 驱动程序需要位于 CLASSPATH 中。

  此 ZIP 文件不包含完整的 AWS SDK for Java 2.x。但是，它包含适用于 AWS Identity and Access Management (IAM) 数据库身份验证所需的 Java 2.x 驱动程序相关库的 AWS 开发工具包。

将此 Amazon Redshift JDBC 驱动程序与 IAM 数据库身份验证所需的 AWS SDK 一起使用。

要安装完整的 AWS SDK for Java 2.x，请参阅《适用于 Java 的 AWS SDK 开发人员指南》中的 AWS SDK for Java 2.x。

- 与 JDBC 4.2 兼容的驱动程序版本 2.x（不含 AWS SDK）

查看 JDBC 驱动程序版本 2.x 软件许可证并更改日志文件：

- JDBC 驱动程序版本 2.x 许可证
- JDBC 驱动程序版本 2.x 更改日志

JDBC 驱动程序版本 1.2.27.1051 及更高版本支持 Amazon Redshift 存储过程。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的在 Amazon Redshift 中创建存储过程。

安装 Amazon Redshift JDBC 驱动程序版本 2.x

要安装与 Amazon Redshift JDBC 4.2 兼容的驱动程序版本 2.x 和 AWS SDK 的驱动程序相关库，请将文件从 ZIP 归档提取到您选择的目录。

要安装与 Amazon Redshift JDBC 4.2 兼容的驱动程序版本 2.x（不含 AWS SDK），请将 JAR 文件复制到您选择的目录。

要使用 Amazon Redshift JDBC 驱动程序访问 Amazon Redshift 数据存储，您需要执行如下配置。

主题

- 引用 JDBC 驱动程序库
- 注册驱动程序类

引用 JDBC 驱动程序库

用于连接到数据的 JDBC 应用程序或 Java 代码必须访问驱动程序 JAR 文件。在应用程序或代码中，指定从 ZIP 归档中提取的所有 JAR 文件。

在 JDBC 应用程序中使用驱动程序

JDBC 应用程序通常提供一组用于添加驱动程序库文件列表的配置选项。使用提供的选项将 ZIP 归档中的所有 JAR 文件作为应用程序中的驱动程序配置的一部分。有关更多信息，请参阅您 JDBC 应用程序的文档。

在 Java 代码中使用驱动程序

您必须在类路径中包含所有驱动程序库文件。这是 Java 运行时环境搜索类和其他资源文件的路径。有关更多信息，请参阅相应的 Java SE 文档，以便为操作系统设置类路径。

- Windows：https://docs.oracle.com/javase/7/docs/technotes/tools/windows/classpath.html
- Linux 和 Solaris：https://docs.oracle.com/javase/7/docs/technotes/tools/solaris/classpath.html
- macOS：默认 macOS 类路径是安装 JDBC 驱动程序的目录。

注册驱动程序类

请确保您为您的应用程序注册了适当的类。您可以使用以下类将 Amazon Redshift JDBC 驱动程序连接到 Amazon Redshift 数据存储：

- `Driver` 类扩展 `java.sql.Driver`。
- `DataSource` 类扩展 `javax.sql.DataSource` 和 `javax.sql.ConnectionPoolDataSource`。

驱动程序支持以下独立于 JDBC 版本的完全限定类名：

- `com.amazon.redshift.jdbc.Driver`
- `com.amazon.redshift.jdbc.DataSource`

以下示例说明了如何使用 DriverManager 类建立 JDBC 4.2 的连接。

```
            private static Connection connectViaDM() throws Exception
{
Connection connection = null;
connection = DriverManager.getConnection(CONNECTION_URL);
return connection;
}
```

下面的示例展示了如何使用 `DataSource` 类建立连接。

```
 private static Connection connectViaDS() throws Exception
```

```
{
Connection connection = null;
11
Amazon Redshift JDBC Driver Installation and Configuration Guide
DataSource ds = new com.amazon.redshift.jdbc.DataSource
();
ds.setURL(CONNECTION_URL);
connection = ds.getConnection();
return connection;
}
```

获取 JDBC URL

您需要获知您的 Amazon Redshift 集群的 JDBC URL，才能将 SQL 客户端工具连接到该集群。JDBC URL 采用以下格式：`jdbc:redshift://`*`endpoint`*`:`*`port`*`/`*`database`*。

上述格式的字段具有以下值。

| Field | 值 |
| --- | --- |
| `jdbc` | 连接协议。 |
| `redshift` | 用于指定使用 Amazon Redshift 驱动程序连接到数据库的子协议。 |
| *`endpoint`* | Amazon Redshift 集群的端点。 |
| *`port`* | 您在启动集群时指定的端口号。如果您启用了防火墙，请确保此端口处于打开状态，可供您使用。 |
| *`database`* | 您为集群创建的数据库。 |

以下是一个示例 JDBC URL：`jdbc:redshift://examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com:5439/dev`

如果您的 URL 值包含以下任何 URI 预留字符，则这些值必须经过 URL 编码：

- ;
- +
- {

- }
- [
- ]
- &
- =
- ?
- 一个空格

例如，如果 PWD 的值为 password:password，则使用该值的连接 URL 将如下所示：

```
jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/
dev;UID=amazon;PWD=password%3Apassword
```

有关如何获取 JDBC 连接的信息，请参阅查找集群连接字符串。

如果客户端计算机无法连接到数据库，您可以进行故障排除，解决可能存在的问题。有关更多信息，请参阅 解决 Amazon Redshift 中的连接问题。

构建连接 URL

使用连接 URL 向您正在访问的数据存储提供连接信息。以下是 Amazon Redshift JDBC 驱动程序版本 2.x 的连接 URL 的格式。此处，[主机] 是 Amazon Redshift 服务器的端点，[端口] 是服务器用于侦听客户端请求的传输控制协议 (TCP) 端口的编号。

```
jdbc:redshift://[Host]:[Port]
```

以下是指定某些可选设置的连接 URL 的格式。

```
jdbc:redshift://[Host]:[Port]/[database];[Property1]=[Value];
[Property2]=[Value];
```

如果您的 URL 值包含以下任何 URI 预留字符，则这些值必须经过 URL 编码：

- ;
- +
- {

- }
- [
- ]
- &
- =
- ?
- 一个空格

例如，如果 PWD 的值为 password:password，则使用该值的连接 URL 将如下所示：

```
jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/
dev;UID=amazon;PWD=password%3Apassword
```

例如，假设您希望在 AWS 上连接到位于美国西部（加利福尼亚北部）区域的 Amazon Redshift 集群上的端口 9000。您还希望访问名为 dev 的数据库，并使用数据库用户名和密码对连接进行身份验证。在此情况下，您使用以下连接 URL。

```
jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/
dev;UID=amazon;PWD=amazon
```

您可以使用以下字符将配置选项与 URL 字符串的其余部分分隔开：

- ;
- ?

例如，以下 URL 字符串等效：

```
jdbc:redshift://my_host:5439/dev;ssl=true;defaultRowFetchSize=100
```

```
jdbc:redshift://my_host:5439/dev?ssl=true;defaultRowFetchSize=100
```

您可以使用以下字符将 URL 字符串中的配置选项彼此分隔开：

- ;
- &

例如，以下 URL 字符串等效：

```
jdbc:redshift://my_host:5439/dev;ssl=true;defaultRowFetchSize=100
```

```
jdbc:redshift://my_host:5439/dev;ssl=true&defaultRowFetchSize=100
```

以下 URL 示例指定日志级别 6 和日志的路径。

```
jdbc:redshift://redshift.amazonaws.com:5439/dev;DSILogLevel=6;LogPath=/home/user/logs;
```

不要复制连接 URL 中的属性。

有关可指定的配置选项的完整列表，请参阅 JDBC 驱动程序版本 2.x 配置的选项。

> **Note**
>
> 连接时，请勿使用集群节点的 IP 地址或 VPC 端点的 IP 地址。请务必使用 Redshift 端点以避免不必要的中断。使用端点 URL 的唯一例外是使用自定义域名时。有关更多信息，请参阅使用自定义域名进行客户端连接。

使用 Apache Maven 配置 JDBC 连接

Apache Maven 是一款软件项目管理及理解工具。适用于 Java 的 AWS SDK 支持 Apache Maven 项目。有关更多信息，请参阅《适用于 Java 的 AWS SDK 开发人员指南》中的将开发工具包与 Apache Maven 一起使用。

如果您使用 Apache Maven，可以配置并构建您的项目，以使用 Amazon Redshift JDBC 驱动程序与 Amazon Redshift 集群连接。为此，在项目的 `pom.xml` 文件中将 JDBC 驱动程序添加为依赖项。如果您使用 Maven 生成项目并希望使用 JDBC 连接，请执行以下部分中的步骤。

将 JDBC 驱动程序配置为 Maven 依赖项

1. 将 Amazon 存储库或 Maven Central 存储库添加到您的 `pom.xml` 文件的存储库部分。

   > **Note**
   >
   > 以下代码示例中的 URL 如果用在浏览器中，将返回错误。仅在 Maven 项目的上下文中使用此 URL。

要使用安全套接字层 (SSL) 进行连接，请将以下存储库添加到您的 pom.xml 文件。

```
<repositories>
    <repository>
      <id>redshift</id>
      <url>https://s3.amazonaws.com/redshift-maven-repository/release</url>
    </repository>
</repositories>
```

对于 Maven Central 存储库，请将以下内容添加到您的 pom.xml 文件。

```
<repositories>
    <repository>
      <id>redshift</id>
      <url>https://repo1.maven.org/maven2</url>
    </repository>
</repositories>
```

2. 在 pom.xml 文件的依赖项部分中，声明您要使用的驱动程序版本。

   Amazon Redshift 向与 JDBC 4.2 API 兼容的工具提供驱动程序。有关这些驱动程序支持的功能的信息，请参阅 下载 Amazon Redshift JDBC 驱动程序版本 2.x。

   将以下示例中的 *driver-version* 替换为您的驱动程序版本，例如 2.1.0.1。对于兼容 JDBC 4.2 的驱动程序，请使用以下内容。

   ```
   <dependency>
      <groupId>com.amazon.redshift</groupId>
      <artifactId>redshift-jdbc42</artifactId>
      <version>driver-version</version>
   </dependency>
   ```

   此驱动程序的类名是 com.amazon.redshift.Driver。

当您使用 IAM 数据库身份验证时，Amazon Redshift Maven 驱动程序需要以下可选依赖项。

```
<dependency>
      <groupId>com.amazonaws</groupId>
      <artifactId>aws-java-sdk-core</artifactId>
```

```
      <version>1.12.23</version>
      <scope>runtime</scope>
      <optional>true</optional>
</dependency>
    <dependency>
      <groupId>com.amazonaws</groupId>
      <artifactId>aws-java-sdk-redshift</artifactId>
      <version>1.12.23</version>
      <scope>runtime</scope>
      <optional>true</optional>
</dependency>
<dependency>
      <groupId>com.amazonaws</groupId>
      <artifactId>aws-java-sdk-sts</artifactId>
      <version>1.12.23</version>
      <scope>runtime</scope>
      <optional>true</optional>
</dependency>
```

要将 Amazon Redshift JDBC 驱动程序升级或更改到最新版本，请先将依赖项的版本部分修改为驱动程序的最新版本。然后用 Maven Clean Plugin 清除您的项目，如下所示。

```
mvn clean
```

配置身份验证和 SSL

为了防止数据遭到未经授权的访问，Amazon Redshift 数据存储要求所有连接都使用用户凭据进行身份验证。某些数据存储还要求通过安全套接字层 (SSL) 协议建立连接，无论是否使用单向身份验证。

Amazon Redshift JDBC 驱动程序版本 2.x 提供了对这些身份验证协议的完全支持。

驱动程序支持的 SSL 版本取决于您使用的 JVM 版本。有关每个 Java 版本支持的 SSL 版本的信息，请参阅 Java 平台组产品管理博客上的诊断 TLS、SSL 和 HTTPS。

用于连接的 SSL 版本是驱动程序和服务器都支持的最高版本，该版本在连接时确定。

配置 Amazon Redshift JDBC 驱动程序版本 2.x，以根据要连接到的 Redshift 服务器的安全要求对连接进行身份验证。

要对连接进行身份验证，您必须始终提供您的 Redshift 用户名和密码。根据服务器上是否启用或需要 SSL，您可能还需要将驱动程序配置为通过 SSL 进行连接。或者，您可能要使用单向 SSL 身份验证，以便客户端（驱动程序本身）验证服务器的身份。

在连接 URL 中将配置信息提供给驱动程序。有关连接 URL 的语法的更多信息，请参阅 构建连接 URL。

SSL 表示 TLS/SSL，包括传输层安全性和安全套接字层。驱动程序支持 TLS/SSL 的行业标准版本。

配置 IAM 身份验证

如果您使用 IAM 身份验证连接到 Amazon Redshift 服务器，请将以下属性设置为数据来源连接字符串的一部分。

有关 IAM 身份验证的更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

要使用 IAM 身份验证，请使用以下连接字符串格式之一：

| 连接字符串 | 描述 |
| --- | --- |
| `jdbc:redshift:iam:// [host]:[port]/[db]` | 常规连接字符串。驱动程序从主机推断 ClusterID 和区域。 |
| `jdbc:redshift:iam:// [cluster-id]: [region]/[db]` | 在已知 ClusterID 和区域的情况下，驱动程序将检索主机信息。 |
| `jdbc:redshift:iam:// [host]/[db]` | 驱动程序默认为端口 5439，并从主机推断 ClusterID 和区域。根据您在创建、修改或迁移集群时选择的端口，允许访问所选端口。 |

指定配置文件

如果您使用 IAM 身份验证，则可以在配置文件名称下指定任何其他必需或可选的连接属性。通过这样做，您可以避免将某些信息直接放入连接字符串中。您可以使用 Profile 属性在连接字符串中指定配置文件名称。

配置文件可以添加到 AWS 凭证文件。此文件的默认位置为：`~/.aws/credentials`

您可以通过在以下环境变量中设置路径更改默认值：`AWS_CREDENTIAL_PROFILES_FILE`

有关配置文件的更多信息，请参阅 适用于 Java 的 AWS SDK 中的使用 AWS 凭证。

使用实例配置文件凭证

如果您在与 IAM 角色关联的 Amazon EC2 实例上运行应用程序，则可以使用实例配置文件凭证进行连接。

为此，请使用上表中的 IAM 连接字符串格式之一，并将 dbuser 连接属性设置为您正要以其身份连接的 Amazon Redshift 用户名。

有关实例配置文件的更多信息，请参阅《IAM 用户指南》中的访问控制。

使用凭证提供程序

驱动程序还支持来自以下服务的凭证提供程序插件：

- AWS IAM Identity Center
- Active Directory 联合身份验证服务 (ADFS)
- JSON Web Token (JWT) 服务
- Microsoft Azure Active Directory (AD) 服务和浏览器 Microsoft Azure Active Directory (AD) 服务
- Okta 服务
- PingFederate 服务
- 适用于 SAML 服务（如 Okta、Ping 或 ADFS）的浏览器 SAML

如果您使用上述服务之一，则连接 URL 需要指定以下属性：

- Plugin_Name – 凭证提供程序插件类的完全限定类路径。
- IdP_Host：– 您用于对 Amazon Redshift 进行身份验证的服务的主机。
- IdP_Port – 身份验证服务的主机侦听的端口。Okta 不需要。
- User – idp_host 服务器的用户名。
- Password – 与 idp_host 用户名关联的密码。
- DbUser – 您连接时使用的 Amazon Redshift 用户名。
- SSL_Insecure – 指示是否应验证 IDP 服务器证书。
- Client_ID – 与 Azure AD 门户中的用户名关联的客户端 ID。仅用于 Azure AD。
- Client_Secret – 与 Azure AD 门户中的客户端 ID 关联的客户端密钥。仅用于 Azure AD。
- IdP_Tenant – 适用于您的 Amazon Redshift 应用程序的 Azure AD 租户 ID。仅用于 Azure AD。
- App_ID – 您的 Amazon Redshift 应用程序的 Okta 应用程序 ID。仅适用于 Okta。
- 应用程序名称 – 您的 Amazon Redshift 应用程序的可选 Okta 应用程序名称。仅适用于 Okta。
- Partner_SPID – 可选的合作伙伴 SPID（服务提供商 ID）值。仅用于 PingFederate。
- Idc_Region – AWS IAM Identity Center 实例所在的 AWS 区域。仅用于 AWS IAM Identity Center。
- Issuer_Url – AWS IAM Identity Center 服务器的实例端点。仅用于 AWS IAM Identity Center。

如果您将浏览器插件用于这些服务之一，则连接 URL 还可以包括：

- Login_URL – 通过浏览器插件使用安全断言标记语言 (SAML) 或 Azure AD 服务时，身份提供者网站上的资源的 URL。如果使用浏览器插件，则此参数是必需的。

- Listen_Port – 通过浏览器插件使用 SAML、Azure AD 或 AWS IAM Identity Center 服务时，驱动程序用于从身份提供者获取 SAML 响应的端口。

- IdP_Response_Timeout – 通过浏览器插件使用 SAML、Azure AD 或 AWS IAM Identity Center 服务时，驱动程序等待身份提供者的 SAML 响应的时间（以秒为单位）。

有关其他连接字符串属性的信息，请参阅JDBC 驱动程序版本 2.x 配置的选项。

仅使用用户名和密码

如果要连接的服务器不使用 SSL，则只需提供 Redshift 用户名和密码即可对连接进行身份验证。

仅使用您的 Redshift 用户名和密码配置身份验证

1. 将 UID 属性设置为您的 Redshift 用户名，以便访问 Amazon Redshift 服务器。
2. 将 PWD 属性设置为与您的 Redshift 用户名相对应的密码。

在不验证身份的情况下使用 SSL

如果要连接的服务器使用 SSL 但不需要身份验证，则可以将驱动程序配置为使用非验证 SSL 出厂设置。

要在不进行身份验证的情况下配置 SSL 连接

1. 将 UID 属性设置为您的 Redshift 用户名，以便访问 Amazon Redshift 服务器。
2. 将 PWD 属性设置为与您的 Redshift 用户名相对应的密码。
3. 将 SSLFactory 属性设置为 com.amazon.redshift.ssl.NonValidatingFactory。

使用单向 SSL 身份验证

如果要连接的服务器使用 SSL 并且具有证书，则您可以将驱动程序配置为使用单向身份验证来验证服务器的身份。

单向身份验证需要签名、受信任的 SSL 证书来验证服务器的身份。您可以将驱动程序配置为使用特定证书或访问包含相应证书的 TrustStore。如果您未指定证书或 TrustStore，则驱动程序将使用默认的 Java TrustStore（通常为 `jssecacerts` 或 `cacerts`）。

要配置单向 SSL 身份验证

1. 将 UID 属性设置为您的 Redshift 用户名，以便访问 Amazon Redshift 服务器。
2. 将 PWD 属性设置为与您的 Redshift 用户名相对应的密码。
3. 将 SSL 属性设置为 true。
4. 将 SSLRootCert 属性设置为根 CA 证书的位置。
5. 如果您未使用默认的 Java TrustStore 之一，请执行以下操作之一：
   - 要指定服务器证书，请将 SSLRootCert 属性设置为证书的完整路径。
   - 要指定 TrustStore，请执行以下操作：
     a. 使用 keytool 程序将服务器证书添加到要使用的 TrustStore 中。
     b. 指定使用驱动程序启动 Java 应用程序时要使用的 TrustStore 和密码。例如：

```
-Djavax.net.ssl.trustStore=[TrustStoreName]
-Djavax.net.ssl.trustStorePassword=[TrustStorePassword]
-Djavax.net.ssl.trustStoreType=[TrustStoreType]
```

6. 请选择一种：
   - 要验证证书，请将 SSLMode 属性设置为 verify-ca。
   - 要验证证书并确认证书中的主机名，请将 SSLMode 属性设置为 verify-full。

配置日志记录

您可以在驱动程序中打开日志记录以帮助诊断问题。

您可以使用以下方法记录驱动程序信息：

- 要将记录的信息保存在 .log 文件中，请参阅 使用日志文件。
- 要将记录的信息发送到 DriverManager 中指定的 LogStream 或 LogWriter，请参阅 使用 LogStream 或 LogWriter。

在连接 URL 中将配置信息提供给驱动程序。有关连接 URL 的语法的更多信息，请参阅 构建连接 URL。

使用日志文件

只有打开足够长的日志记录才能捕获问题。日志记录会降低性能，并会占用大量磁盘空间。

在连接 URL 中设置 LogLevel 键以打开日志记录并指定包含在日志文件中的详细信息量。下表列出了 Amazon Redshift JDBC 驱动程序版本 2.x 提供的日志记录级别，按从最不详细到最详细的顺序排列。

| LogLevel 值 | 描述 |
| --- | --- |
| 1 | 记录将导致驱动程序中止的严重错误事件。 |
| 2 | 记录也许不会导致驱动程序中止运行的错误事件。 |
| 3 | 记录在未执行操作时可能导致错误的事件。此日志记录级别和高于此级别的日志记录级别也会记录用户的查询。 |
| 4 | 记录描述驱动程序进程的一般信息。 |
| 5 | 记录用于调试驱动程序的详细信息。 |
| 6 | 记录所有驱动程序活动。 |

要设置使用日志文件的日志记录

1. 将 LogLevel 属性设置为要包含在日志文件中的所需信息级别。
2. 将 LogPath 属性设置为您要保存日志文件的目标文件夹的完整路径。

   例如，以下连接 URL 启用日志记录级别 3 并将日志文件保存在 C:\temp 文件夹中：`jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/Default;DSILogLevel=3;LogPath=C:\temp`
3. 要确保新设置生效，请重新启动 JDBC 应用程序并重新连接到服务器。

   Amazon Redshift JDBC 驱动程序在 LogPath 属性中指定的位置生成以下日志文件：

   - redshift_jdbc.log 文件，该文件记录不特定于连接的驱动程序活动。
   - 与数据库建立的每个连接的 redshift_jdbc_connection_[Number].log 文件，其中 [Number] 是标识每个日志文件的编号。此文件记录特定于连接的驱动程序活动。

如果 LogPath 值无效，则驱动程序将记录的信息发送到标准输出流（System.out）

使用 LogStream 或 LogWriter

只有打开足够长的日志记录才能捕获问题。日志记录会降低性能，并会占用大量磁盘空间。

在连接 URL 中设置 LogLevel 键以打开日志记录，并指定发送到 DriverManager 中指定的 LogStream 或 LogWriter 的详细信息量。

要打开使用 LogStream 或 LogWriter 的日志记录：

1. 要将驱动程序配置为记录描述驱动程序进度的一般信息，请将 LogLevel 属性设置为 1 或 INFO。
2. 要确保新设置生效，请重新启动 JDBC 应用程序并重新连接到服务器。

数据类型转换

Amazon Redshift JDBC 驱动程序版本 2.x 支持许多常见的数据格式，从而可在 Amazon Redshift、SQL 和 Java 数据类型之间进行转换。

下表列出了支持的数据类型映射。

| Amazon Redshift 类型 | SQL 类型 | Java 类型 |
|---|---|---|
| BIGINT | SQL_BIGINT | 长整型 |
| BOOLEAN | SQL_BIT | Boolean |
| CHAR | SQL_CHAR | 字符串 |
| DATE | SQL_TYPE_DATE | java.sql.Date |
| DECIMAL | SQL_NUMERIC | BigDecimal |
| DOUBLE PRECISION | SQL_DOUBLE | Double |
| GEOMETRY | SQL_ LONGVARBINARY | byte[] |
| INTEGER | SQL_INTEGER | 整数 |
| OID | SQL_BIGINT | 长整型 |

| Amazon Redshift 类型 | SQL 类型 | Java 类型 |
| --- | --- | --- |
| SUPER | SQL_LONGVARCHAR | 字符串 |
| REAL | SQL_REAL | Float |
| SMALLINT | SQL_SMALLINT | 短型 |
| TEXT | SQL_VARCHAR | 字符串 |
| TIME | SQL_TYPE_TIME | java.sql.Time |
| TIMETZ | SQL_TYPE_TIME | java.sql.Time |
| TIMESTAMP | SQL_TYPE_ TIMESTAMP | java.sql.Timestamp |
| TIMESTAMPTZ | SQL_TYPE_ TIMESTAMP | java.sql.Timestamp |
| VARCHAR | SQL_VARCHAR | 字符串 |

使用预编译的语句支持

Amazon Redshift JDBC 驱动程序支持预编译的语句。您可以使用预编译的语句来提高需要在同一连接过程中多次运行的参数化查询的性能。

预编译的语句是在服务器端编译但不立即运行的 SQL 语句。在关闭对象或连接之前，已编译的语句作为 PreparedStatement 对象存储在服务器上。当该对象存在时，您可以根据需要使用不同的参数值多次运行预编译的语句，而无需再次编译该语句。此操作降低了开销，可以更快地运行一组查询。

有关预编译语句的更多信息，请参阅来自 Oracle 的 JDBC 基础知识教程中的“使用预编译语句”。

您可以预编译包含多个查询的语句。例如，以下预编译语句包含两个 INSERT 查询：

```
PreparedStatement pstmt = conn.prepareStatement("INSERT INTO
MyTable VALUES (1, 'abc'); INSERT INTO CompanyTable VALUES
(1, 'abc');");
```

请注意，这些查询不依赖于在同一预编译语句内指定的其他查询的结果。由于查询在预编译步骤期间未运行，因此尚未返回结果，并且不可用于同一预编译语句中的其他查询。

例如，不允许使用以下预编译语句，该语句将创建一个表，然后将值插入到该新创建的表中：

```
PreparedStatement pstmt = conn.prepareStatement("CREATE
TABLE MyTable(col1 int, col2 varchar); INSERT INTO myTable
VALUES (1, 'abc');");
```

如果您尝试预编译此语句，服务器将返回一个错误，指出目标表 (myTable) 尚不存在。必须先运行 CREATE 查询，然后才能预编译 INSERT 查询。

JDBC 驱动程序 2.x 版本与 1.x 版本之间的差异

本节介绍 JDBC 驱动程序的 2.x 和 1.x 版本返回的信息的差异。JDBC 驱动程序 1.x 版本已停产。

下表列出了由 getDatabaseProductName() 和 getDatabaseProductVersion() 函数为每个版本的 JDBC 驱动程序返回的 DatabaseMetadata 信息。JDBC 驱动程序版本 2.x 在建立连接时获取值。JDBC 驱动程序版本 1.x 通过查询获取值。

| JDBC 驱动程序版本 | getDatabaseProductName() 结果 | getDatabaseProductVersion() 结果 |
|---|---|---|
| 2.x | Redshift | 8.0.2 |
| 1.x | PostgreSQL | 08.00.0002 |

下表列出了由 getTypeInfo 函数为每个版本的 JDBC 驱动程序返回的 DatabaseMetadata 信息。

| JDBC 驱动程序版本 | getTypeInfo 结果 |
|---|---|
| 2.x | 与 Redshift 数据类型一致 |
| 1.x | 与 PostgreSQL 数据类型保持一致 |

为 JDBC 驱动程序版本 2.x 创建初始化 (.ini) 文件

通过对 Amazon Redshift JDBC 驱动程序版本 2.x 使用初始化 (.ini) 文件，可以指定系统级别的配置参数。例如，联合 IdP 身份验证参数可能因每个应用程序而异。.ini 文件为 SQL 客户端提供了获取所需配置参数的常见位置。

可以创建包含 SQL 客户端配置选项的 JDBC 驱动程序版本 2.x 初始化 (.ini) 文件。文件的默认名称为 `rsjdbc.ini`。JDBC 驱动程序版本 2.x 在按优先级顺序列出的以下位置检查 .ini 文件：

- `IniFile`连接 URL 或 SQL 客户端的连接属性对话框中的 参数。请确保 `IniFile` 参数包含 .ini 文件的完整路径，其中包括文件名。有关 `IniFile` 参数的信息，请参阅 IniFile。如果 `IniFile` 参数错误地指定了 .ini 文件的位置，将显示错误。
- 环境变量，例如具有完整路径 AMAZON_REDSHIFT_JDBC_INI_FILE，包括文件名。您可以使用 `rsjdbc.ini` 或指定文件名。如果 AMAZON_REDSHIFT_JDBC_INI_FILE 环境变量错误地指定了 .ini 文件的位置，将显示一个错误。
- 驱动程序 JAR 文件所在的目录。
- 用户主目录。
- 系统的临时目录。

您可以将 .ini 文件组织成各个部分，例如 [DRIVER]。每个部分都包含键-值对，这些键-值对将指定各种连接参数。您可以使用 `IniSection` 参数来指定 .ini 文件中的部分。有关 `IniSection` 参数的信息，请参阅 IniSection。

以下是 .ini 文件格式的示例，其中包含了 [DRIVER]、[DEV]、[QA] 和 [PROD] 的部分。[DRIVER] 部分可以应用于任何连接。

```
[DRIVER]
key1=val1
key2=val2

[DEV]
key1=val1
key2=val2

[QA]
key1=val1
key2=val2

[PROD]
key1=val1
key2=val2
```

JDBC 驱动程序版本 2.x 从按优先级顺序列出的以下位置加载配置参数：

- 应用程序代码中的默认配置参数。
- .ini 文件中的 [DRIVER] 部分属性（如果包含）。

- 自定义部分配置参数，如果 `IniSection` 选项在连接 URL 或 SQL 客户端的连接属性对话框中提供。
- `getConnection` 调用中指定的连接属性对象的属性。
- 连接 URL 中指定的配置参数。

JDBC 驱动程序版本 2.x 配置的选项

接下来，您可以找到您可以为 Amazon Redshift JDBC 驱动程序 2.x 版本指定的选项的说明。配置选项不区分大小写。

您可以使用连接 URL 设置配置属性。有关更多信息，请参阅 构建连接 URL。

主题

- AccessKeyID
- AllowDBUserOverride
- App_ID
- App_Name
- ApplicationName
- AuthProfile
- AutoCreate
- Client_ID
- Client_Secret
- ClusterID
- 压缩
- connectTimeout
- connectionTimezone
- databaseMetadataCurrentDbOnly
- DbUser
- DbGroups
- DBNAME
- defaultRowFetchSize
- DisableIsValidQuery

- enableFetchRingBuffer
- enableMultiSqlSupport
- EnableTableTypes
- fetchRingBufferSize
- ForceLlowcase
- groupFederation
- HOST
- IAMDisableCache
- IAMDuration
- Idc_Client_Display_Name
- Idc_Region
- IdP_Host
- IdP_Partition
- IdP_Port
- IdP_Tenant
- IdP_Response_Timeout
- IniFile
- IniSection
- isServerless
- Issuer_Url
- Listen_Port
- Login_URL
- loginTimeout
- loginToRp
- LogLevel
- LogPath
- OverrideSchemaPatternType
- Partner_SPID
- 密码
- Plugin_Name

- PORT
- preferPQ
- Preferred_Role
- 配置文件
- PWD
- queryGroup
- readOnly
- Region
- reWriteBatchedInserts
- reWriteBatchedInsertsSize
- roleArn
- roleSessionName
- 范围
- SecretAccessKey
- SessionToken
- serverlessAcctId
- serverlessWorkGroup
- socketFactory
- socketTimeout
- SSL
- SSL_Insecure
- SSLCert
- SSLFactory
- SSLKey
- SSLMode
- SSLPassword
- SSLRootCert
- StsEndpointUrl
- tcpKeepAlive
- 令牌

- token_type
- UID
- 用户
- webIdentityToken

AccessKeyID

- 默认值 – 无
- 数据类型 – 字符串

您可以指定此参数以输入用户或角色的 IAM 访问密钥。您通常可以通过查看现有的字符串或用户配置文件来查找密钥。如果您指定此参数，还必须指定 SecretAccessKey 参数。如果在 JDBC URL 中传递，则 AccessKeyID 必须采用 URL 编码。

此参数为可选的。

AllowDBUserOverride

- 默认值 - 0
- 数据类型 – 字符串

此选项指定驱动程序是使用 SAML 断言中的 DbUser 值，还是连接 URL 的 DbUser 连接属性中指定的值。

此参数为可选的。

1

驱动程序使用 SAML 断言中的 DbUser 值。

如果 SAML 断言没有为 DBUser 指定值，则驱动程序将使用 DBUser 连接属性中指定的值。如果连接属性也没有指定值，则驱动程序将使用连接配置文件中指定的值。

0

驱动程序使用 DBUser 连接属性中指定的 DBUser 值。

如果 DBUser 连接属性没有指定值，则驱动程序将使用连接配置文件中指定的值。如果连接配置文件也没有指定值，则驱动程序将使用 SAML 断言中的值。

App_ID

- 默认值 – 无
- 数据类型 – 字符串

Okta 提供的与您的 Amazon Redshift 应用程序关联的唯一 ID。

如果通过 Okta 服务进行身份验证，则此参数是必需的。

App_Name

- 默认值 – 无
- 数据类型 – 字符串

您用于验证与 Amazon Redshift 的连接的 Okta 应用程序的名称。

此参数为可选的。

ApplicationName

- 默认值 – null
- 数据类型 – 字符串

传递给 Amazon Redshift 以供审计的应用程序的名称。

此参数为可选的。

AuthProfile

- 默认值 – 无
- 数据类型 – 字符串

用于连接到 Amazon Redshift 的身份验证配置文件的名称。

此参数为可选的。

AutoCreate

- 默认值 – false
- 数据类型 – Boolean

此选项指定在指定用户不存在时驱动程序是否导致新用户被创建。

此参数为可选的。

true

如果通过 DBUser 或唯一 ID (UID) 指定的用户不存在，则会创建具有该名称的新用户。

false

驱动程序不会导致新用户被创建。如果指定的用户不存在，则身份验证将失败。

Client_ID

- 默认值 – 无
- 数据类型 – 字符串

使用 Azure AD 服务对连接进行身份验证时使用的客户端 ID。

如果通过 Azure AD 服务进行身份验证，则此参数是必需的。

Client_Secret

- 默认值 – 无
- 数据类型 – 字符串

使用 Azure AD 服务对连接进行身份验证时使用的客户端密钥。

如果通过 Azure AD 服务进行身份验证，则此参数是必需的。

ClusterID

- 默认值 – 无
- 数据类型 – 字符串

要连接到的 Amazon Redshift 集群的名称。驱动程序将尝试从给定主机中检测此参数。如果您使用的是网络负载均衡器 (NLB) 并通过 IAM 进行连接，驱动程序将无法检测到它，因此您可以使用此连接选项进行设置。

此参数为可选的。

压缩

- 默认值 – 关闭
- 数据类型 – 字符串

用于 Amazon Redshift 服务器与客户端或驱动程序之间线路协议通信的压缩方法。

此参数为可选的。

可以指定以下值：

- lz4

  将用于与 Amazon Redshift 通信的线路协议通信的压缩方法设置为 lz4。
- off

  与 Amazon Redshift 通信的线路协议通信不使用压缩。

connectTimeout

- 默认值 – 10
- 数据类型 – 整数

用于套接字连接操作的超时值。如果建立 Amazon Redshift 连接所需的时间超过此值，则系统将该连接视为不可用。超时以秒为单位指定。值为 0 表示未指定超时。

此参数为可选的。

connectionTimezone

- 默认值 – LOCAL
- 数据类型 – 字符串

会话级别的时区。

此参数为可选的。

可以指定以下值：

LOCAL

将会话级别的时区配置为 LOCAL JVM 时区。

SERVER

将会话级别的时区配置为在 Amazon Redshift 服务器上为用户设置的时区。您可以使用以下命令为用户配置会话级别的时区：

```
ALTER USER
[...]
SET TIMEZONE TO [...];
```

databaseMetadataCurrentDbOnly

- 默认值 – true
- 数据类型 – Boolean

此选项指定元数据 API 是从所有可访问的数据库中检索数据，还是仅从连接的数据库检索数据。

此参数为可选的。

可以指定以下值：

true

应用程序从单个数据库中检索元数据。

false

应用程序从所有可访问的数据库中检索元数据。

DbUser

- 默认值 – 无
- 数据类型 – 字符串

用于您的 Amazon Redshift 账户的用户 ID。如果您启用了 AutoCreate 属性，则可以使用当前不存在的 ID。

此参数为可选的。

DbGroups

- 默认值 – PUBLIC
- 数据类型 – 字符串

DBUser 为当前会话加入的现有数据库组名称的逗号分隔列表。

此参数为可选的。

DBNAME

- 默认值 – null
- 数据类型 – 字符串

要连接的数据库的名称。您可以使用此选项在 JDBC 连接 URL 中指定数据库名称。

此参数为必需参数。您必须在连接 URL 或客户端应用程序的连接属性中指定数据库名称。

defaultRowFetchSize

- 默认值 - 0
- 数据类型 – 整数

此选项可指定 getFetchSize 的默认值。

此参数为可选的。

可以指定以下值：

0

在单一操作中取回所有行。

正整数

对于 ResultSet 的每次取回迭代，要从数据库中取回的行数。

DisableIsValidQuery

- 默认值 – False

- 数据类型 – Boolean

此选项指定驱动程序在使用 Connection.isValid() 方法来确定数据库连接是否处于活动状态时是否提交新的数据库查询。

此参数为可选的。

true

驱动程序在使用 Connection.isValid() 方法来确定数据库连接是否处于活动状态时不提交查询。如果数据库服务器意外关闭，这可能会导致驱动程序错误地将数据库连接标识为活动状态。

false

驱动程序在使用 Connection.isValid() 方法来确定数据库连接是否处于活动状态时提交查询。

enableFetchRingBuffer

- 默认值 – true
- 数据类型 – Boolean

此选项指定驱动程序使用单独线程上的环形缓冲区取回行。fetchRingBufferSize 参数指定环形缓冲区的大小。

环形缓冲区在 JDBC 中实施自动内存管理，以防止在数据检索操作期间出现内存不足（OOM）错误。环形缓冲区实时监控缓冲数据的实际大小，确保驱动程序占用的内存总量不超过定义的限制。当缓冲区达到容量上限时，驱动程序会暂停数据提取操作，无需人工干预即可防止内存溢出。这项内置保护措施可以自动消除 OOM 错误，用户无需进行任何配置。

如果事务检测到包含多个 SQL 命令（以分号分隔）的语句，则该事务的提取环形缓冲区将设置为 false。enableFetchRingBuffer 的值不变。

此参数为可选的。

> **Note**
>
> 当禁用环形缓冲区且提取大小配置不当时，可能会出现内存不足（OOM）问题。有关配置提取大小的更多信息，请参阅此处。

enableMultiSqlSupport

- 默认值 – true
- 数据类型 – Boolean

此选项指定是否处理语句中以分号分隔的多个 SQL 命令。

此参数为可选的。

可以指定以下值：

true

驱动程序处理语句对象中多个用分号分隔的 SQL 命令。

false

驱动程序为单个语句中的多个 SQL 命令返回错误。

EnableTableTypes

- 默认值 – true
- 数据类型 – Boolean

此选项指定驱动程序是否在 `getTables` 和 `getTableTypes` `DatabaseMetaData` 方法的结果中识别来自数据来源的详细表类型信息。默认情况下，驱动程序可以识别详细表类型。

此参数为可选的。它在驱动程序版本 2.2.8 和更高版本中可用。

true

驱动程序可识别以下表类型：TABLE、VIEW、SYSTEM TABLE、SYSTEM VIEW、EXTERNAL TABLE 和 LOCAL TEMPORARY。

false

驱动程序将详细表类型信息标准化为泛型 TABLE 和 VIEW 表类型。

fetchRingBufferSize

- 默认值 – 1G

- 数据类型 – 字符串

此选项指定在取回结果集时使用的环形缓冲区的大小。您可以指定以字节为单位的大小，例如 1K 表示 1 KB，5000 代表 5000 字节，1M 表示 1 MB，1G 表示 1 GB，依此类推。您还可以指定堆内存的百分比。驱动程序在达到限制时停止取回行。当应用程序读取行并释放环形缓冲区中的空间时，取回将恢复。

此参数为可选的。

ForceLlowcase

- 默认值 – false
- 数据类型 – Boolean

此选项指定在使用单点登录身份验证时，驱动程序是否会将从身份提供者发送到 Amazon Redshift 的所有数据库组 (DbGroup) 设为小写。

此参数为可选的。

true

驱动程序会小写从身份提供者发送的所有数据库组。

false

驱动程序不会更改数据库组。

groupFederation

- 默认值 – false
- 数据类型 – Boolean

此选项指定是否使用 Amazon Redshift IDP 组。此选项由 GetClusterCredentialsV2 API 支持。

此参数为可选的。

true

使用 Amazon Redshift 身份提供者 (IDP) 组。

false

使用 STS API 和 GetClusterCredentials 进行用户联合身份验证，并明确指定用于连接的 DbGroups。

HOST

- 默认值 – null
- 数据类型 – 字符串

要连接到 Amazon Redshift 服务器的主机名。您可以使用此选项在 JDBC 连接 URL 中指定主机名。

此参数为必需参数。您必须在连接 URL 或客户端应用程序的连接属性中指定主机名。

IAMDisableCache

- 默认值 – false
- 数据类型 – Boolean

此选项指定是否缓存 IAM 凭证。

此参数为可选的。

true

IAM 凭证不会被缓存。

false

IAM 凭证将被缓存。例如，当对 API 网关的请求受到限制时，这会提高性能。

IAMDuration

- 默认值 – 900
- 数据类型 – 整数

临时 IAM 凭证过期之前的时间长度（以秒为单位）。

- 最小值 – 900

- 最大值 – 3,600

此参数为可选的。

Idc_Client_Display_Name

- 默认值 – Amazon Redshift JDBC 驱动程序
- 数据类型 – 字符串

要用于使用 BrowserIdcAuthPlugin 的客户端的显示名称。

此参数为可选的。

Idc_Region

- 默认值 – 无
- 数据类型 – 字符串

IAM Identity Center 实例所在的 AWS 区域。

只有在 plugin_name 配置选项中使用 `BrowserIdcAuthPlugin` 进行身份验证时，才需要此参数。

IdP_Host

- 默认值 – 无
- 数据类型 – 字符串

您用于对 Amazon Redshift 进行身份验证的 IdP（身份提供者）主机。该选项可以在连接字符串或配置文件中指定。

此参数为可选的。

IdP_Partition

- 默认值 – 无
- 数据类型 – 字符串

指定在其中配置身份提供者（IdP）的云分区。这决定了驱动程序连接到哪个 IdP 身份验证端点。

如果将此参数留空，则驱动程序默认为商业分区。可能的值有：

- us-gov：如果在 Azure Government 中配置了 IdP，请使用此值。例如，Azure AD Government 使用端点 login.microsoftonline.us。
- cn：如果在中国云分区中配置了 IdP，请使用此值。例如，Azure AD China 使用 login.chinacloudapi.cn 端点。

此参数为可选的。

IdP_Port

- 默认值 – 无
- 数据类型 – 字符串

IdP（身份提供者）使用的端口。您可以在连接字符串或配置文件中指定端口。默认端口为 5439。根据您在创建、修改或迁移集群时选择的端口，允许访问所选端口。

此参数为可选的。

IdP_Tenant

- 默认值 – 无
- 数据类型 – 字符串

适用于您的 Amazon Redshift 应用程序的 Azure AD 租户 ID。

如果通过 Azure AD 服务进行身份验证，则此参数是必需的。

IdP_Response_Timeout

- 默认值 – 120
- 数据类型 – 整数

当通过浏览器插件使用 SAML 或 Azure AD 服务时，驱动程序等待身份提供者发出 SAML 响应的时间（以秒为单位）。

此参数为可选的。

IniFile

- 默认值 – 无
- 数据类型 – 字符串

.ini 文件的完整路径，包括文件名。例如：

```
IniFile="C:\tools\rsjdbc.ini"
```

有关 .ini 文件的信息，请参阅为 JDBC 驱动程序版本 2.x 创建初始化 (.ini) 文件。

此参数为可选的。

IniSection

- 默认值 – 无
- 数据类型 – 字符串

.ini 文件中包含配置选项的部分的名称。有关 .ini 文件的信息，请参阅为 JDBC 驱动程序版本 2.x 创建初始化 (.ini) 文件。

以下示例指定 .ini 文件的 [Prod] 部分：

```
IniSection="Prod"
```

此参数为可选的。

isServerless

- 默认值 – false
- 数据类型 – Boolean

此选项指定 Amazon Redshift 端点主机是否为无服务器实例。驱动程序将尝试从给定主机中检测此参数。如果您使用的是网络负载均衡器 (NLB)，则驱动程序将无法检测到它，因此您可以在此处对其进行设置。

此参数为可选的。

true

Amazon Redshift 端点主机是一个无服务器实例。

false

Amazon Redshift 端点主机是一个预置集群。

Issuer_Url

- 默认值 – 无
- 数据类型 – 字符串

指向 AWS IAM Identity Center 服务器的实例端点。

只有在 plugin_name 配置选项中使用 `BrowserIdcAuthPlugin` 进行身份验证时，才需要此参数。

Listen_Port

- 默认值：7890
- 数据类型 – 整数

通过浏览器插件使用 SAML、Azure AD 或 AWS Identity Center 服务时，驱动程序用于接收来自身份提供者的 SAML 响应或授权代吗的端口。

此参数为可选的。

Login_URL

- 默认值 – 无
- 数据类型 – 字符串

通过浏览器插件使用 SAML 或 Azure AD 服务时，身份提供者网站上的资源 URL。

如果通过浏览器插件使用 SAML 或 Azure AD 服务进行身份验证，则此参数是必需的。

loginTimeout

- 默认值 - 0
- 数据类型 – 整数

连接服务器并对服务器进行身份验证时发生超时前等待的秒数。如果建立连接的用时长于此阈值，则连接将被中止。

当此属性设置为 0 时，连接不会发生超时。

此参数为可选的。

loginToRp

- 默认值 – `urn:amazon:webservices`
- 数据类型 – 字符串

要用于 AD FS 身份验证类型的信赖方信任。

此参数为可选的。

LogLevel

- 默认值 - 0
- 数据类型 – 整数

使用此属性可以打开或关闭驱动程序中的日志记录并指定包含在日志文件中的详细信息量。

只有启用足够长的日志记录才能捕获问题。日志记录会降低性能，并会占用大量磁盘空间。

此参数为可选的。

将参数设置为以下值之一：

0

禁用所有日志记录。

1

启用 FATAL 级别的日志记录，该级别日志记录将记录导致驱动程序中止的非常严重错误事件。

2

启用 ERROR 级别的日志记录，该级别日志记录将记录可能仍然允许驱动程序继续运行的错误事件。

3

启用 WARNING 级别的日志记录，该级别日志记录将记录在未执行操作时可能导致错误的事件。

4

启用 INFO 级别的日志记录，该级别日志记录将记录描述驱动程序进程的一般信息。

5

启用 DEBUG 级别的日志记录，该级别日志记录将记录对调试驱动程序很有用的详细信息。

6

启用 TRACE 级别的日志记录，该级别日志记录将记录所有驱动程序活动。

启用日志记录后，驱动程序会在 LogPath 属性指定的位置生成以下日志文件：

- **redshift_jdbc.log**：用于记录非特定于连接的驱动程序活动的文件。
- **redshift_jdbc_connection_[Number].log** – 与数据库建立的每个连接的文件，其中 [Number] 是区分每个日志文件和其他日志文件的数字。此文件记录特定于连接的驱动程序活动。

如果 LogPath 值无效，则驱动程序将记录的信息发送到标准输出流 System.out。

LogPath

- 默认值 – 当前工作目录。
- 数据类型 – 字符串

启用 DSILogLevel 属性时驱动程序保存日志文件所在的文件夹的完整路径。

为了确保连接 URL 与所有 JDBC 应用程序兼容，我们建议您通过键入另一个反斜杠来转义文件路径中的反斜杠 (\)。

此参数为可选的。

OverrideSchemaPatternType

- 默认值 – null
- 数据类型 – 整数

此选项指定是否覆盖在 getTables 调用中使用的查询类型。

0

无模式通用查询

1

本地模式查询

2

外部模式查询

此参数为可选的。

Partner_SPID

- 默认值 – 无
- 数据类型 – 字符串

在使用 PingFederate 服务验证连接时使用的合作伙伴 SPID（服务提供商 ID）值。

此参数为可选的。

密码

- 默认值 – 无
- 数据类型 – 字符串

使用 IAM 身份验证通过 IDP 进行连接时，它是 IDP_Host 服务器的密码。使用标准身份验证时，它可用于 Amazon Redshift 数据库密码，而不是 PWD。

此参数为可选的。

Plugin_Name

- 默认值 – 无
- 数据类型 – 字符串

实施特定凭证提供程序插件的完全限定类名称。

此参数为可选的。

以下提供程序选项受支持：

- **AdfsCredentialsProvider** – Active Directory 联合身份验证服务。
- **AzureCredentialsProvider** – Microsoft Azure Active Directory（AD）服务。
- **BasicJwtCredentialsProvider** – JSON Web Token（JWT）服务。
- **BasicSamlCredentialsProvider** – 您可以与许多 SAML 服务提供商一起使用的安全断言标记语言（SAML）凭据。
- **BrowserAzureCredentialsProvider** – 浏览器 Microsoft Azure Active Directory（AD）服务。
- **BrowserAzureOAuth2CredentialsProvider** –用于本机身份验证的浏览器 Microsoft Azure Active Directory（AD）服务。
- **BrowserIdcAuthPlugin** – 使用 AWS IAM Identity Center 的授权插件。
- **BrowserSamlCredentialsProvider** – 用于 SAML 服务（如 Okta、Ping 或 ADFS）的浏览器 SAML。
- **IdpTokenAuthPlugin** – 一种授权插件，可接受 AWS IAM Identity Center 令牌，或者基于 OpenID Connect（OIDC）JSON 的身份令牌（JWT），该令牌由与 AWS IAM Identity Center 关联的任意 Web 身份提供者提供。
- **OktaCredentialsProvider** – Okta 服务。
- **PingCredentialsProvider** – PingFederate 服务。

PORT

- 默认值 – null
- 数据类型 – 整数

要连接到 Amazon Redshift 服务器的端口。您可以使用此选项在 JDBC 连接 URL 中指定端口。

此参数为可选的。

preferPQ

- 默认值 – true
- 数据类型 – Boolean

此选项控制驱动程序是否在 TLS 1.3 ClientHello 中通告后量子混合密钥交换组。当提供后量子组并且服务器也支持它们时，服务器会选择一个。否则，握手会透明地回退到经典密钥交换。

此参数为可选的。它在驱动程序版本 2.2.8 和更高版本中可用。

true

当运行时支持时，驱动程序会通告后量子混合 TLS 密钥交换组。

false

驱动程序不通告后量子组。如果网络中间盒因无法识别驱动程序提供的后量子密钥交换组而阻止 TLS 握手，则使用此设置。

Preferred_Role

- 默认值 – 无
- 数据类型 – 字符串

您希望在 Amazon Redshift 连接期间代入的 IAM 角色。

此参数为可选的。

配置文件

- 默认值 – 无
- 数据类型 – 字符串

用于 IAM 身份验证的配置文件的名称。此配置文件包含未在连接字符串中指定的任何其他连接属性。

此参数为可选的。

PWD

- 默认值 – 无
- 数据类型 – 字符串

与您使用属性 UID 提供的 Amazon Redshift 用户名相对应的密码。

此参数为可选的。

queryGroup

- 默认值 – null

- 数据类型 – 字符串

此选项通过将查询分配到相应的查询组在运行时向队列分配查询。将为会话设置查询组。在连接上运行的所有查询都属于此查询组。

此参数为可选的。

readOnly

- 默认值 – false
- 数据类型 – Boolean

此属性指定驱动程序是否处于只读模式。

此参数为可选的。

true

连接处于只读模式，无法写入数据存储。

false

连接不处于只读模式，可以写入数据存储。

Region

- 默认值 – null
- 数据类型 – 字符串

此选项指定集群所在的 AWS 区域。如果指定 StsEndPoint 选项，则会忽略“区域”选项。Redshift `GetClusterCredentials` API 操作也使用“区域”选项。

此参数为可选的。

reWriteBatchedInserts

- 默认值 – false
- 数据类型 – Boolean

通过此选项，可以优化批处理的兼容 INSERT 语句的重写和合并。

此参数为可选的。

reWriteBatchedInsertsSize

- 默认值 – 128
- 数据类型 – 整数

通过此选项，可以优化批处理的兼容 INSERT 语句的重写和合并。此值必须以 2 为幂呈指数增加。

此参数为可选的。

roleArn

- 默认值 – 无
- 数据类型 – 字符串

角色的 Amazon 资源名称（ARN）。在为 Plugin_Name 选项指定 BasicJwtCredentialsProvider 时，请确保指定此参数。采用以下格式指定 ARN：

`arn:partition:service:region:account-id:resource-id`

如果您为 Plugin_Name 选项指定 BasicJwtCredentialsProvider，则此参数为必需项。

roleSessionName

- 默认值 – jwt_redshift_session
- 数据类型 – 字符串

所代入角色会话的标识符。通常，您可以传递与应用程序用户关联的名称或标识符。您的应用程序使用的临时安全凭证与该用户相关联。在为 Plugin_Name 选项指定 BasicJwtCredentialsProvider 时，您可以指定此参数。

此参数为可选的。

范围

- 默认值 – 无
- 数据类型 – 字符串

用户可以同意的范围列表，以空格分隔。您可以指定此参数，以便 Microsoft Azure 应用程序可以获得您想调用的 API 的同意。在为 Plugin_Name 选项指定 BrowserAzureOAuth2CredentialsProvider 时，您可以指定此参数。

此参数是 BrowserAzureOAuth2CredentialsProvider 插件必需的。

SecretAccessKey

- 默认值 – 无
- 数据类型 – 字符串

用户或角色的 IAM 访问密钥。如果指定了此选项，则还必须指定 AccessKeyID。如果在 JDBC URL 中传递，则 SecretAccessKey 必须采用 URL 编码。

此参数为可选的。

SessionToken

- 默认值 – 无
- 数据类型 – 字符串

与您用于身份验证的 IAM 角色关联的临时 IAM 会话令牌。如果在 JDBC URL 中传递，则临时 IAM 会话令牌必须采用 URL 编码。

此参数为可选的。

serverlessAcctId

- 默认值 – null
- 数据类型 – 字符串

Amazon Redshift Serverless 账户 ID。驱动程序将尝试从给定主机中检测此参数。如果您使用的是网络负载均衡器 (NLB)，则驱动程序将无法检测到它，因此您可以在此处对其进行设置。

此参数为可选的。

serverlessWorkGroup

- 默认值 – null
- 数据类型 – 字符串

Amazon Redshift Serverless 工作组名称。驱动程序将尝试从给定主机中检测此参数。如果您使用的是网络负载均衡器 (NLB)，则驱动程序将无法检测到它，因此您可以在此处对其进行设置。

此参数为可选的。

socketFactory

- 默认值 – null
- 数据类型 – 字符串

此选项指定了用于创建套接字的套接字产生组件。

此参数为可选的。

socketTimeout

- 默认值 - 0
- 数据类型 – 整数

套接字读取操作发生超时前等待的秒数。如果操作用时长于此阈值，则连接将被关闭。当此属性设置为 0 时，连接不会发生超时。

此参数为可选的。

SSL

- 默认值 – TRUE
- 数据类型 – 字符串

使用此属性可以打开或关闭连接的 SSL。

此参数为可选的。

可以指定以下值：

TRUE

驱动程序通过 SSL 连接到服务器。

FALSE

驱动程序在不使用 SSL 的情况下连接到服务器。IAM 身份验证不支持此选项。

或者，您也可以配置 AuthMech 属性。

SSL_Insecure

- 默认值 – true
- 数据类型 – 字符串

此属性指示是否应验证 IDP 主机服务器证书。

此参数为可选的。

可以指定以下值：

true

驱动程序不检查 IDP 服务器证书的真实性。

false

驱动程序检查 IDP 服务器证书的真实性。

SSLCert

- 默认值 – 无
- 数据类型 – 字符串

.pem 或 .crt 文件的完整路径，其中包含在使用 SSL 时用于验证 Amazon Redshift 服务器实例的其他受信任 CA 证书。

如果指定了 SSLKey，则此参数为必填项。

SSLFactory

- 默认值 – 无
- 数据类型 – 字符串

在不使用服务器证书的情况下，通过 TLS/SSL 连接到服务器时使用的 SSL 出厂设置。

SSLKey

- 默认值 – 无

- 数据类型 – 字符串

.der 文件的完整路径，包含用于验证 SSLCert 中指定的证书的 PKCS8 密钥文件。

如果指定了 SSLCert，则此参数为必填项。

SSLMode

- 默认值 – verify-ca
- 数据类型 – 字符串

使用此属性指定启用 TLS/SSL 时驱动程序如何验证证书。

此参数为可选的。

可以指定以下值：

verify-ca

驱动程序验证证书是否来自受信任的证书颁发机构 (CA)。

verify-full

驱动程序验证证书来自受信任的 CA，以及证书中的主机名是否与连接 URL 中指定的主机名匹配。

> **Note**
>
> 我们建议尽可能使用 `verify-full` 进行 SSL 连接。

SSLPassword

- 默认值 - 0
- 数据类型 – 字符串

在 SSLKey 中指定的加密密钥文件的密码。

如果指定了 SSLKey 且对密钥文件进行了加密，则此参数为必填项。

SSLRootCert

- 默认值 – 无
- 数据类型 – 字符串

.pem 或 .crt 文件的完整路径，其中包含在使用 SSL 时用于验证 Amazon Redshift 服务器实例的根 CA 证书。

StsEndpointUrl

- 默认值 – null
- 数据类型 – 字符串

您可以指定 AWS Security Token Service (AWS STS) 端点。如果指定此选项，则“区域”选项将被忽略。您只能为此端点指定安全协议 (HTTPS)。

tcpKeepAlive

- 默认值 – TRUE
- 数据类型 – 字符串

使用此属性可打开或关闭 TCP Keepalive。

此参数为可选的。

可以指定以下值：

TRUE

驱动程序将使用 TCP Keepalive 来防止连接超时。

FALSE

驱动程序不使用 TCP Keepalive。

令牌

- 默认值 – 无
- 数据类型 – 字符串

AWS IAM Identity Center 提供的访问令牌，或 OpenID Connect（OIDC）JSON Web 令牌（JWT，JSON Web Token），该令牌由与 AWS IAM Identity Center 关联的 Web 身份提供者提供。您的应用程序必须通过 AWS IAM Identity Center 或与 AWS IAM Identity Center 关联的身份提供者，对您的应用程序用户进行身份验证，从而生成此令牌。

此参数可用于 `IdpTokenAuthPlugin`。

token_type

- 默认值 – 无
- 数据类型 – 字符串

`IdpTokenAuthPlugin` 中使用的令牌类型。

可以指定以下值：

ACCESS_TOKEN

如果您使用 AWS IAM Identity Center 提供的访问令牌，请输入此项。

EXT_JWT

如果您使用 OpenID Connect（OIDC）JSON Web 令牌（JWT）（该令牌由与 AWS IAM Identity Center 集成的基于 Web 的身份提供者提供），请输入此项。

此参数可用于 `IdpTokenAuthPlugin`。

UID

- 默认值 – 无
- 数据类型 – 字符串

用于访问数据库的数据库用户名。

此参数为必需参数。

用户

- 默认值 – 无

- 数据类型 – 字符串

通过 IDP 使用 IAM 身份验证进行连接时，这是 idp_host 服务器的用户名。使用标准身份验证时，这可用于 Amazon Redshift 数据库用户名。

此参数为可选的。

webIdentityToken

- 默认值 – 无
- 数据类型 – 字符串

身份提供者提供的 OAuth 2.1 访问令牌或 OpenID Connect ID 令牌。您的应用程序必须通过 Web 身份提供者对您的应用程序用户进行身份验证来获取此令牌。在为 Plugin_Name 选项指定 BasicJwtCredentialsProvider 时，请确保指定此参数。

如果您为 Plugin_Name 选项指定 BasicJwtCredentialsProvider，则此参数为必填项。

JDBC 驱动程序版本 2.x 的以前版本

仅当您的工具需要使用某个特定版本的驱动程序时，才能下载 Amazon Redshift JDBC 驱动程序版本 2.x 的以前版本。

以下是以前的与 JDBC 4.2 兼容的 JDBC 驱动程序版本 2.x 驱动程序：

- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.8/redshift-jdbc42-2.2.8.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.7/redshift-jdbc42-2.2.7.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.6/redshift-jdbc42-2.2.6.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.5/redshift-jdbc42-2.2.5.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.4/redshift-jdbc42-2.2.4.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.3/redshift-jdbc42-2.2.3.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.2/redshift-jdbc42-2.2.2.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.1/redshift-jdbc42-2.2.1.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.2.0/redshift-jdbc42-2.2.0.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.34/redshift-jdbc42-2.1.0.34.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.33/redshift-jdbc42-2.1.0.33.zip

- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.32/redshift-jdbc42-2.1.0.32.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.30/redshift-jdbc42-2.1.0.30.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.29/redshift-jdbc42-2.1.0.29.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.28/redshift-jdbc42-2.1.0.28.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.26/redshift-jdbc42-2.1.0.26.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.25/redshift-jdbc42-2.1.0.25.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.24/redshift-jdbc42-2.1.0.24.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.23/redshift-jdbc42-2.1.0.23.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.22/redshift-jdbc42-2.1.0.22.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.21/redshift-jdbc42-2.1.0.21.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.20/redshift-jdbc42-2.1.0.20.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.19/redshift-jdbc42-2.1.0.19.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.18/redshift-jdbc42-2.1.0.18.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.17/redshift-jdbc42-2.1.0.17.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.16/redshift-jdbc42-2.1.0.16.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.15/redshift-jdbc42-2.1.0.15.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.14/redshift-jdbc42-2.1.0.14.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.13/redshift-jdbc42-2.1.0.13.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.12/redshift-jdbc42-2.1.0.12.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.11/redshift-jdbc42-2.1.0.11.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.10/redshift-jdbc42-2.1.0.10.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.8/redshift-jdbc42-2.1.0.8.zip
- https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.7/redshift-jdbc42-2.1.0.7.zip

## Amazon Redshift Python 连接器

通过使用适用于 Python 的 Amazon Redshift 连接器，您可以将工作与适用于 Python 的 AWS SDK（Boto3）以及 Pandas 和 Numerical Python（NumPy）集成。有关 pandas 的更多信息，请参阅 pandas GitHub 存储库。有关 NumPy 的更多信息，请参阅 NumPy GitHub 存储库。

Amazon Redshift Python 连接器提供了一个开源解决方案。您可以浏览源代码、请求增强功能、报告问题和提供文章。

要使用 Amazon Redshift Python 连接器，请确保使用 Python 3.6 或更高版本。有关更多信息，请参阅Amazon Redshift Python 驱动程序许可协议。

Amazon Redshift Python 连接器提供以下内容：

- AWS Identity and Access Management（IAM）身份验证 有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。
- 使用联合 API 访问进行身份提供者身份验证 企业身份提供者支持联合 API 访问，如下所示：
  - Azure AD 有关更多信息，请参阅 AWS 大数据博客文章 Federate Amazon Redshift access with Microsoft Azure AD single sign-on。
  - Active Directory 联合身份验证服务 有关更多信息，请参阅 AWS 大数据博客文章 Federate access to your Amazon Redshift cluster with Active Directory Federation Services (AD FS): Part 1。
  - Okta。有关更多信息，有关更多信息，请参阅 AWS 大数据博客文章 Federate Amazon Redshift access with Okta as an identity provider。
  - PingFederate。有关更多信息，请参阅 PingFederate 站点。
  - JumpCloud。有关的更多信息，请参阅 JumpCloud 站点。
- Amazon Redshift 数据类型。

Amazon Redshift Python 连接器实施 Python 数据库 API 规范 2.0。有关更多信息，请参阅 Python 网站上的 PEP 249 – Python 数据库 API 规范 v2.0。

主题

- 安装 Amazon Redshift Python 连接器
- Amazon Redshift Python 连接器的配置选项
- 导入 Python 连接器
- 将 Python 连接器与 NumPy 集成
- 将 Python 连接器与 pandas 集成
- 使用身份提供者插件
- 使用 Amazon Redshift Python 连接器的示例
- Amazon Redshift Python 连接器的 API 参考

安装 Amazon Redshift Python 连接器

您可以使用以下任意一种方法安装 Amazon Redshift Python 连接器：

- Python 包索引（PyPI）
- Conda
- 克隆 GitHub 存储库

从 PyPI 安装 Python 连接器

要从 Python 包索引（PyPI）安装 Python 连接器，您可以使用 pip。要执行此操作，请运行以下命令。

```
>>> pip install redshift_connector
```

您可以在虚拟环境中安装连接器。要执行此操作，请运行以下命令。

```
>>> pip install redshift_connector
```

或者，你可以用连接器安装 pandas 和 NumPy。

```
>>> pip install 'redshift_connector[full]'
```

有关 pip 的更多信息，请参阅 pip 站点。

从 Conda 安装 Python 连接器

你可以从 Anaconda.org 安装 Python 连接器。

```
>>>conda install -c conda-forge redshift_connector
```

通过从 AWS 中克隆 GitHub 存储库来安装 Python 连接器

要从源代码安装 Python 连接器，请从 AWS 克隆 GitHub 存储库。安装 Python 和 virtualenv 之后，通过运行以下命令来设置环境并安装所需的依赖项。

```
$ git clone https://github.com/aws/amazon-redshift-python-driver.git
```

```
$ cd amazon-redshift-python-driver
$ virtualenv venv
$ . venv/bin/activate
$ python -m pip install -r requirements.txt
$ python -m pip install -e .
$ python -m pip install redshift_connector
```

### Amazon Redshift Python 连接器的配置选项

下面，您可以找到您可以为 Amazon Redshift Python 连接器指定的选项的说明。除非另有说明，否则以下选项适用于最新的可用连接器版本。

access_key_id

- 默认值 – 无
- 数据类型 – 字符串

为 IAM 数据库身份验证配置的 IAM 角色或用户的访问密钥 ID。

此参数为可选的。

allow_db_user_override

- 默认值 – False
- 数据类型 – Boolean

True

指定连接器使用安全断言标记语言（SAML）断言中的 `DbUser` 值。

False

指定使用 `DbUser` 连接参数中的值。

此参数为可选的。

app_name

- 默认值 – 无
- 数据类型 – 字符串

用于身份验证的身份提供者（IdP）应用程序的名称。

此参数为可选的。

application_name

- 默认值 – 无
- 数据类型 – 字符串

传递给 Amazon Redshift 以供审计的客户端应用程序的名称。您提供的应用程序名称显示在 SYS_CONNECTION_LOG 表的“application_name”列中。这有助于在调试问题时跟踪连接源并对其进行故障排除。

此参数为可选的。

auth_profile

- 默认值 – 无
- 数据类型 – 字符串

连接属性为 JSON 的 Amazon Redshift 身份验证配置文件的名称。有关命名连接参数的更多信息，请参阅 `RedshiftProperty` 类。`RedshiftProperty` 类存储由最终用户提供的连接参数，如果适用，在 IAM 身份验证过程中生成（例如，临时 IAM 凭证）。有关更多信息，请参阅 RedshiftProperty 类。

此参数为可选项。

auto_create

- 默认值 – False
- 数据类型 – Boolean

一个值，指示在用户不存在的情况下是否创建用户。

此参数为可选的。

client_id

- 默认值 – 无

- 数据类型 – 字符串

Azure IdP 中的客户端 ID。

此参数为可选的。

client_secret

- 默认值 – 无
- 数据类型 – 字符串

Azure IdP 中的客户端密钥。

此参数为可选的。

cluster_identifier

- 默认值 – 无
- 数据类型 – 字符串

Amazon Redshift 集群的集群标识符。

此参数为可选的。

credentials_provider

- 默认值 – 无
- 数据类型 – 字符串

对 Amazon Redshift 进行身份验证的 IdP。有效值如下所示：

- `AdfsCredentialsProvider`
- `AzureCredentialsProvider`
- `BrowserAzureCredentialsProvider`
- `BrowserAzureOAuth2CredentialsProvider`
- `BrowserIdcAuthPlugin` – 使用 AWS IAM Identity Center 的授权插件。
- `BrowserSamlCredentialsProvider`

- IdpTokenAuthPlugin – 一种授权插件，可接受 AWS IAM Identity Center 令牌，或者基于 OpenID Connect（OIDC）JSON 的身份令牌（JWT），该令牌由与 AWS IAM Identity Center 关联的任意 Web 身份提供者提供。
- PingCredentialsProvider
- OktaCredentialsProvider

此参数为可选的。

database

- 默认值 – 无
- 数据类型 – 字符串

要连接到的数据库的名称。

此参数为必需参数。

database_metadata_current_db_only

- 默认值 – True
- 数据类型 – Boolean

一个值，指示应用程序是否支持多数据库数据共享目录。默认值 True 表示应用程序不支持多数据库数据共享目录以实现向后兼容性。

此参数为可选的。

db_groups

- 默认值 – 无
- 数据类型 – 字符串

DbUser 指示的用户为当前会话加入的现有数据库组名称的逗号分隔列表。

此参数为可选的。

db_user

- 默认值 – 无

- 数据类型 – 字符串

用于 Amazon Redshift 的用户 ID。

此参数为可选的。

enable_table_types

- 默认值 – True
- 数据类型 – Boolean

指定连接器是否在 `get_tables` 和 `get_table_types` 元数据方法的结果中识别来自数据来源的详细表类型信息。默认情况下，连接器可以识别详细表类型。

- True：连接器可识别以下表类型：TABLE、VIEW、SYSTEM TABLE、SYSTEM VIEW、EXTERNAL TABLE 和 LOCAL TEMPORARY。
- False：连接器将详细表类型信息标准化为泛型 TABLE 和 VIEW 表类型。

此参数为可选的。它在连接器版本 2.1.16 及更高版本中可用。

endpoint_url

- 默认值 – 无
- 数据类型 – 字符串

Amazon Redshift 端点 URL。此选项仅供 AWS 内部使用。

此参数为可选的。

group_federation

- 默认值 – False
- 数据类型 – Boolean

此选项指定是否使用 Amazon Redshift IDP 组。

此参数为可选的。

true

使用 Amazon Redshift 身份提供者 (IDP) 组。

false

使用 STS API 和 GetClusterCredentials 进行用户联合身份验证，并指定用于连接的 db_groups。

host

- 默认值 – 无
- 数据类型 – 字符串

Amazon Redshift 集群的主机名。

此参数为可选的。

IAM

- 默认值 – False
- 数据类型 – Boolean

IAM 身份验证已启用。

此参数为必需参数。

iam_disable_cache

- 默认值 – False
- 数据类型 – Boolean

此选项指定是否缓存 IAM 凭证。IAM 凭证将默认被缓存。当对 API 网关的请求受到限制时，这样可以提高性能。

此参数为可选的。

idc_client_display_name

- 默认值 – Amazon Redshift Python 连接器

- 数据类型 – 字符串

要用于使用 BrowserIdcAuthPlugin 的客户端的显示名称。

此参数为可选的。

idc_region

- 默认值 – 无
- 数据类型 – 字符串

AWS IAM Identity Center 实例所在的 AWS 区域。

只有在 credentials_provider 配置选项中使用 `BrowserIdcAuthPlugin` 进行身份验证时，才需要此参数。

idp_partition

- 默认值 – 无
- 数据类型 – 字符串

指定在其中配置身份提供者（IdP）的云分区。这决定了驱动程序连接到哪个 IdP 身份验证端点。

如果将此参数留空，则驱动程序默认为商业分区。可能的值有：

- `us-gov`：如果在 Azure Government 中配置了 IdP，请使用此值。例如，Azure AD Government 使用端点 `login.microsoftonline.us`。
- `cn`：如果在中国云分区中配置了 IdP，请使用此值。例如，Azure AD China 使用 `login.chinacloudapi.cn` 端点。

此参数为可选的。

idpPort

- 默认值 – 7890
- 数据类型 – 整数

IdP 将 SAML 断言发送到的侦听端口。

此参数为必需参数。

idp_response_timeout

- 默认值 – 120
- 数据类型 – 整数

从 IdP 检索 SAML 断言的超时时间。

此参数为必需参数。

idp_tenant

- 默认值 – 无
- 数据类型 – 字符串

IdP 租户。

此参数为可选的。

issuer_url

- 默认值 – 无
- 数据类型 – 字符串

指向 AWS IAM Identity Center 服务器的实例端点。

只有在 credentials_provider 配置选项中使用 `BrowserIdcAuthPlugin` 进行身份验证时，才需要此参数。

listen_port

- 默认值 – 7890
- 数据类型 – 整数

通过浏览器插件使用 SAML、Azure AD 或 AWS IAM Identity Center 服务时，驱动程序用于接收来自身份提供者的 SAML 响应的端口。

此参数为可选的。

login_url

- 默认值 – 无
- 数据类型 – 字符串

IdP 的单点登录 Url。

此参数为可选的。

max_prepared_statements

- 默认值：1000
- 数据类型 – 整数

每个连接将缓存的最大预处理语句数。将该参数设置为 0 将禁用缓存机制。为此参数输入负数会将其设置为默认值。

此参数为可选的。

numeric_to_float

- 默认值 – False
- 数据类型 – Boolean

此选项指定连接器是否将数字数据类型值从 decimal.Decimal 转换为浮点数。默认情况下，连接器将数字数据类型值作为 decimal.Decimal 接收，而不会转换它们。

我们不建议为需要精确度的用例启用 numeric_to_float（数字转浮点），因为结果可能会四舍五入。

有关 decimal.Decimal 以及它与浮点数之间的权衡的更多信息，请参阅 Python 网站上的 decimal — 十进制定点和浮点算术。

此参数为可选的。

partner_sp_id

- 默认值 – 无
- 数据类型 – 字符串

用于 Ping 身份验证的合作伙伴 SP ID。

此参数为可选的。

password

- 默认值 – 无
- 数据类型 – 字符串

用于身份验证的密码。

此参数为可选的。

port

- 默认值 – 5439
- 数据类型 – 整数

Amazon Redshift 集群的端口号。

此参数为必需参数。

preferred_role

- 默认值 – 无
- 数据类型 – 字符串

当前连接首选的 IAM 角色。

此参数为可选的。

principal_arn

- 默认值 – 无
- 数据类型 – 字符串

要为其生成策略的用户或 IAM 角色的 Amazon 资源名称（ARN）。建议您将策略附加到角色，然后将该角色分配给用户以进行访问。

此参数为可选的。

配置文件

- 默认值 – 无
- 数据类型 – 字符串

包含 AWS 凭据的 AWS 凭据文件中的配置文件的名称。

此参数为可选的。

provider_name

- 默认值 – 无
- 数据类型 – 字符串

Redshift 本机验证提供程序的名称。

此参数为可选的。

region

- 默认值 – 无
- 数据类型 – 字符串

集群所在的 AWS 区域。

此参数为可选的。

role_arn

- 默认值 – 无
- 数据类型 – 字符串

调用方承担的角色的 Amazon 资源名称（ARN）。此参数由 `JwtCredentialsProvider` 指示的提供程序使用。

对于 `JwtCredentialsProvider` 提供商，此参数为必填项。否则，此参数位可选项。

role_session_name

- 默认值 – jwt_redshift_session
- 数据类型 – 字符串

所代入角色会话的标识符。通常，您可以传递与使用您的应用程序的用户关联的名称或标识符。您的应用程序使用的临时安全凭证与该用户相关联。此参数由 `JwtCredentialsProvider` 指示的提供程序使用。

此参数为可选的。

范围

- 默认值 – 无
- 数据类型 – 字符串

用户可以同意的范围列表，以空格分隔。您可以指定此参数，以便应用程序可以获得您想调用的 API 的同意。在为 credentials_provider 选项指定 BrowserAzureOAuth2CredentialsProvider 时，您可以指定此参数。

此参数是 BrowserAzureOAuth2CredentialsProvider 插件必需的。

secret_access_key_id

- 默认值 – 无
- 数据类型 – 字符串

为 IAM 数据库身份验证配置的 IAM 角色或用户的秘密访问密钥。

此参数为可选的。

session_token

- 默认值 – 无
- 数据类型 – 字符串

为 IAM 数据库身份验证配置的 IAM 角色或用户的访问密钥 ID。如果使用临时 AWS 凭证，则需要此参数。

此参数为可选的。

serverless_acct_id

- 默认值 – 无
- 数据类型 – 字符串

Amazon Redshift Serverless 账户 ID。

此参数为可选的。

serverless_work_group

- 默认值 – 无
- 数据类型 – 字符串

Amazon Redshift Serverless 工作组名称。

此参数为可选的。

ssl

- 默认值 – True
- 数据类型 – Boolean

已启用安全套接字层（SSL）

此参数为必需参数。

ssl_insecure

- 默认值 – False
- 数据类型 – Boolean

指定是否禁用 IdP 主机的服务器 SSL 证书验证的值。将此参数设置为 True，将禁用对 IdP 主机的服务器 SSL 证书的验证。我们建议您在生产环境中保持默认值 False。

此参数为可选的。

sslmode

- 默认值 – verify-ca
- 数据类型 – 字符串

与 Amazon Redshift 的连接的安全性。您可以指定以下任一值：

- verify-ca
- verify-full

> **Note**
>
> 我们建议尽可能使用 `verify-full` 进行 SSL 连接。

此参数为必需参数。

tcp_keepalive

- 默认值 – True
- 数据类型 – Boolean

是否使用 TCP keepalive 来防止连接超时。可以指定以下值：

- True：驱动程序将使用 TCP keepalive 来防止连接超时。
- False：驱动程序不使用 TCP Keepalive。

此参数为可选的。

tcp_keepalive_count

- 默认值 – 无
- 数据类型 – 整数

所发送的未确认探测器的数量，超过此数量后将连接视为不活动。例如，将该值设置为 3 意味着，驱动程序在发送了 3 个未回应的 keepalive 数据包之后，就会确定连接不再处于活动状态。

如果未指定此参数，Amazon Redshift 将使用系统默认值。

此参数为可选的。

tcp_keepalive_interval

- 默认值 – 无
- 数据类型 – 整数

驱动程序在未收到上一个探测器的确认时，发送后续 keepalive 探测器时的间隔（以秒为单位）。指定此参数时，必须使用正整数。

如果未指定此参数，Amazon Redshift 将使用系统默认值。

此参数为可选的。

tcp_keepalive_idle

- 默认值 – 无
- 数据类型 – 整数

不活动的持续时间（以秒为单位），驱动程序将在此时间之后发送第一个 keepalive 探测器。例如，将该值设置为 120 意味着，对于不活动状态，驱动程序将会等待 2 分钟，然后发送第一个 keepalive 数据包。指定此参数时，必须使用正整数。

如果未指定此参数，Amazon Redshift 将使用系统默认值。

此参数为可选的。

timeout

- 默认值 – 无
- 数据类型 – 整数

连接服务器时发生超时前等待的秒数。

此参数为可选的。

令牌

- 默认值 – 无

- 数据类型 – 字符串

AWS IAM Identity Center 提供的访问令牌，或 OpenID Connect（OIDC）JSON Web 令牌（JWT，JSON Web Token），该令牌由与 AWS IAM Identity Center 关联的 Web 身份提供者提供。您的应用程序必须通过 AWS IAM Identity Center 或与 AWS IAM Identity Center 关联的身份提供者，对您的应用程序用户进行身份验证，从而生成此令牌。

此参数可用于 `IdpTokenAuthPlugin`。

token_type

- 默认值 – 无
- 数据类型 – 字符串

`IdpTokenAuthPlugin` 中使用的令牌类型。

可以指定以下值：

ACCESS_TOKEN

如果您使用 AWS IAM Identity Center 提供的访问令牌，请输入此项。

EXT_JWT

如果您使用 OpenID Connect（OIDC）JSON Web 令牌（JWT）（该令牌由与 AWS IAM Identity Center 集成的基于 Web 的身份提供者提供），请输入此项。

此参数可用于 `IdpTokenAuthPlugin`。

user

- 默认值 – 无
- 数据类型 – 字符串

用于身份验证的用户名。

此参数为可选的。

web_identity_token

- 默认值 – 无

- 数据类型 – 字符串

身份提供者提供的 OAuth 2.0 访问令牌或 OpenID Connect ID 令牌。通过使用 Web 身份提供者对使用您的应用程序的用户进行身份验证，确保您的应用程序获取此令牌。`JwtCredentialsProvider` 指示的提供商使用此参数。

对于 `JwtCredentialsProvider` 提供商，此参数为必填项。否则，此参数位可选项。

导入 Python 连接器

要导入 Python 连接器，请运行以下命令。

```
>>> import redshift_connector
```

要使用 AWS 凭证连接到 Amazon Redshift 集群，请运行以下命令。

```
conn = redshift_connector.connect(
    host='examplecluster.abc123xyz789.us-west-1.redshift.amazonaws.com',
    port=5439,
    database='dev',
    user='awsuser',
    password='my_password'
 )
```

将 Python 连接器与 NumPy 集成

以下是将 Python 连接器与 NumPy 集成的示例。

```
>>>  import numpy
#Connect to the cluster
>>> import redshift_connector
>>> conn = redshift_connector.connect(
     host='examplecluster.abc123xyz789.us-west-1.redshift.amazonaws.com',
     port=5439,
     database='dev',
     user='awsuser',
     password='my_password'
  )

# Create a Cursor object
>>> cursor = conn.cursor()
```

```
# Query and receive result set
cursor.execute("select * from book")

result: numpy.ndarray = cursor.fetch_numpy_array()
print(result)
```

以下是结果。

```
[['One Hundred Years of Solitude' 'Gabriel García Márquez']
['A Brief History of Time' 'Stephen Hawking']]
```

将 Python 连接器与 pandas 集成

以下是将 Python 连接器与 pandas 集成的示例。

```
>>> import pandas

#Connect to the cluster
>>> import redshift_connector
>>> conn = redshift_connector.connect(
     host='examplecluster.abc123xyz789.us-west-1.redshift.amazonaws.com',
     port=5439,
     database='dev',
     user='awsuser',
     password='my_password'
  )

# Create a Cursor object
>>> cursor = conn.cursor()

# Query and receive result set
cursor.execute("select * from book")
result: pandas.DataFrame = cursor.fetch_dataframe()
print(result)
```

使用身份提供者插件

有关如何使用身份提供者插件的一般信息，请参阅用于提供 IAM 凭证的选项。有关管理 IAM 身份的更多信息，包括 IAM 角色的最佳实践，请参阅 Amazon Redshift 中的 Identity and Access Management。

使用 ADFS 身份提供者插件进行身份验证

以下是使用 Active Directory 联合身份验证服务（ADFS）身份提供者插件对连接到 Amazon Redshift 数据库的用户进行身份验证的示例。

```
>>> con = redshift_connector.connect(
    iam=True,
    database='dev',
    host='my-testing-cluster.abc.us-east-2.redshift.amazonaws.com',
    cluster_identifier='my-testing-cluster',
    credentials_provider='AdfsCredentialsProvider',
    user='brooke@myadfshostname.com',
    password='Hunter2',
    idp_host='myadfshostname.com'
)
```

使用 Azure 身份提供者插件进行身份验证

以下是使用 Azure 身份提供者插件进行身份验证的示例。您可以为 `client_id` 和 `client_secret` 创建值用于 Azure 企业应用程序，如下所示。

```
>>>  con = redshift_connector.connect(
    iam=True,
    database='dev',
    host='my-testing-cluster.abc.us-east-2.redshift.amazonaws.com',
    cluster_identifier='my-testing-cluster',
    credentials_provider='AzureCredentialsProvider',
    user='brooke@myazure.org',
    password='Hunter2',
    idp_tenant='my_idp_tenant',
    client_id='my_client_id',
    client_secret='my_client_secret',
    preferred_role='arn:aws:iam:123:role/DataScientist'
)
```

使用 AWS IAM Identity Center 身份提供者插件进行身份验证

以下是使用 AWS IAM Identity Center 身份提供者插件进行身份验证的示例。

```
with redshift_connector.connect(
credentials_provider='BrowserIdcAuthPlugin',
host='my-testing-cluster.abc.us-east-2.redshift.amazonaws.com',
```

```
database='dev',
idc_region='us-east-1',
issuer_url='https://identitycenter.amazonaws.com/ssoins-790723ebe09c86f9',
idp_response_timeout=60,
listen_port=8100,
idc_client_display_name='Test Display Name',
# port value of 5439 is specified by default
)
```

使用 Azure 浏览器身份提供者插件进行身份验证

以下是使用 Azure 浏览器身份提供者插件对连接到 Amazon Redshift 数据库的用户进行身份验证的示例。

浏览器中会发生多重身份验证，其中登录凭证由用户提供。

```
>>>con = redshift_connector.connect(
    iam=True,
    database='dev',
    host='my-testing-cluster.abc.us-east-2.redshift.amazonaws.com',
    cluster_identifier='my-testing-cluster',
    credentials_provider='BrowserAzureCredentialsProvider',
    idp_tenant='my_idp_tenant',
    client_id='my_client_id',
)
```

使用 Okta 身份提供者插件进行身份验证

以下是使用 Okta 身份提供者插件进行身份验证的示例。你可以通过 Okta 应用程序获取 `idp_host`、`app_id` 和 `app_name` 的值。

```
>>> con = redshift_connector.connect(
    iam=True,
    database='dev',
    host='my-testing-cluster.abc.us-east-2.redshift.amazonaws.com',
    cluster_identifier='my-testing-cluster',
    credentials_provider='OktaCredentialsProvider',
    user='brooke@myazure.org',
    password='hunter2',
    idp_host='my_idp_host',
    app_id='my_first_appetizer',
    app_name='dinner_party'
```

```
)
```

使用 JumpCloud 和通用 SAML 浏览器身份提供者插件进行身份验证

以下是使用 JumpCloud 和通用 SAML 浏览器身份提供者插件进行身份验证的示例。

密码参数是必需的。但是，您不必输入此参数，因为浏览器中会发生多重验证。

```
>>> con = redshift_connector.connect(
    iam=True,
    database='dev',
    host='my-testing-cluster.abc.us-east-2.redshift.amazonaws.com',
    cluster_identifier='my-testing-cluster',
    credentials_provider='BrowserSamlCredentialsProvider',
    user='brooke@myjumpcloud.org',
    password='',
    login_url='https://sso.jumpcloud.com/saml2/plustwo_melody'
)
```

使用 Amazon Redshift Python 连接器的示例

以下为如何使用 Amazon Redshift Python 连接器的示例。要运行它们，您必须先安装 Python 连接器。有关安装 Amazon Redshift Python 连接器的更多信息，请参阅安装 Amazon Redshift Python 连接器。有关可以与 Python 连接器一起使用的配置选项的更多信息，请参阅 Amazon Redshift Python 连接器的配置选项。

主题

- 使用 AWS 凭证连接到 Amazon Redshift 集群并进行查询
- 启用自动提交
- 配置游标参数样式
- 使用 COPY 从 Amazon S3 桶中复制数据，然后使用 UNLOAD 将数据写入该桶

使用 AWS 凭证连接到 Amazon Redshift 集群并进行查询

下面的示例将指导您使用 AWS 凭证连接到 Amazon Redshift 集群，然后查询表并检索查询结果。

```
#Connect to the cluster
>>> import redshift_connector
>>> conn = redshift_connector.connect(
```

```
    host='examplecluster.abc123xyz789.us-west-1.redshift.amazonaws.com',
    database='dev',
    port=5439,
    user='awsuser',
    password='my_password'
  )

# Create a Cursor object
>>> cursor = conn.cursor()

# Query a table using the Cursor
>>> cursor.execute("select * from book")

#Retrieve the query result set
>>> result: tuple = cursor.fetchall()
>>> print(result)
 >> (['One Hundred Years of Solitude', 'Gabriel García Márquez'], ['A Brief History of
 Time', 'Stephen Hawking'])
```

### 启用自动提交

根据 Python 数据库 API 规范，默认情况下自动提交属性处于关闭状态。在执行回滚命令后，您可以使用以下命令打开连接的 autocommit 属性，以确保事务不在进行中。

```
#Connect to the cluster
>>> import redshift_connector
>>> conn = redshift_connector.connect(...)

# Run a rollback command
>>>  conn.rollback()

# Turn on autocommit
>>>  conn.autocommit = True
>>>  conn.run("VACUUM")

# Turn off autocommit
>>>  conn.autocommit = False
```

### 配置游标参数样式

可以通过 cursor.paramstyle 修改游标的参数样式。使用的原定设置参数样式是 `format`。参数样式的有效值为 `qmark`、`numeric`、`named`、`format` 和 `pyformat`。

以下是使用各种参数样式将参数传递给示例 SQL 语句的示例。

```
# qmark
redshift_connector.paramstyle = 'qmark'
sql = 'insert into foo(bar, jar) VALUES(?, ?)'
cursor.execute(sql, (1, "hello world"))

# numeric
redshift_connector.paramstyle = 'numeric'
sql = 'insert into foo(bar, jar) VALUES(:1, :2)'
cursor.execute(sql, (1, "hello world"))

# named
redshift_connector.paramstyle = 'named'
sql = 'insert into foo(bar, jar) VALUES(:p1, :p2)'
cursor.execute(sql, {"p1":1, "p2":"hello world"})

# format
redshift_connector.paramstyle = 'format'
sql = 'insert into foo(bar, jar) VALUES(%s, %s)'
cursor.execute(sql, (1, "hello world"))

# pyformat
redshift_connector.paramstyle = 'pyformat'
sql = 'insert into foo(bar, jar) VALUES(%(bar)s, %(jar)s)'
cursor.execute(sql, {"bar": 1, "jar": "hello world"})
```

使用 COPY 从 Amazon S3 桶中复制数据，然后使用 UNLOAD 将数据写入该桶

以下示例说明如何将数据从 Amazon S3 桶复制到表中，然后从该表卸载回到此桶中。

包含以下数据的名为 `category_csv.txt` 的文本文件将上载到 Amazon S3 桶中。

```
12,Shows,Musicals,Musical theatre
13,Shows,Plays,"All ""non-musical"" theatre"
14,Shows,Opera,"All opera, light, and ""rock"" opera"
15,Concerts,Classical,"All symphony, concerto, and choir concerts"
```

以下是 Python 代码的示例，该代码首先连接到 Amazon Redshift 数据库。然后创建一个名为 `category` 的表并将 S3 桶中的 CSV 数据复制到表中。

```
#Connect to the cluster and create a Cursor
```

```
>>> import redshift_connector
>>> with redshift_connector.connect(...) as conn:
>>> with conn.cursor() as cursor:

#Create an empty table
>>>     cursor.execute("create table category (catid int, cargroup varchar, catname
 varchar, catdesc varchar)")

#Use COPY to copy the contents of the S3 bucket into the empty table
>>>     cursor.execute("copy category from 's3://testing/category_csv.txt' iam_role
 'arn:aws:iam::123:role/RedshiftCopyUnload' csv;")

#Retrieve the contents of the table
>>>     cursor.execute("select * from category")
>>>     print(cursor.fetchall())

#Use UNLOAD to copy the contents of the table into the S3 bucket
>>>     cursor.execute("unload ('select * from category') to 's3://testing/
unloaded_category_csv.txt'  iam_role 'arn:aws:iam::123:role/RedshiftCopyUnload' csv;")

#Retrieve the contents of the bucket
>>>     print(cursor.fetchall())
 >> ([12, 'Shows', 'Musicals', 'Musical theatre'], [13, 'Shows', 'Plays', 'All "non-
musical" theatre'], [14, 'Shows', 'Opera', 'All opera, light, and "rock" opera'], [15,
 'Concerts', 'Classical', 'All symphony, concerto, and choir concerts'])
```

如果您未将 autocommit 设置为 true，请在运行 execute() 语句后使用 conn.commit() 提交。

数据卸载到 S3 桶的 unloaded_category_csv.text0000_part00 文件中，内容如下：

```
12,Shows,Musicals,Musical theatre
13,Shows,Plays,"All ""non-musical"" theatre"
14,Shows,Opera,"All opera, light, and ""rock"" opera"
15,Concerts,Classical,"All symphony, concerto, and choir concerts"
```

### Amazon Redshift Python 连接器的 API 参考

您可以在文中找到有关 Amazon Redshift Python 连接器 API 操作的说明。

#### redshift_connector

在下文中，您可以找到有关 redshift_connector API 操作的描述。

`connect(user, database, password[, port, …])`

创建到 Amazon Redshift 集群的连接。此函数可验证用户输入，可以选择使用身份提供者插件进行身份验证，然后构造连接对象。

`apilevel`

支持的 DBAPI 级别，目前为“2.0”。

`paramstyle, str(object='') -> str str(bytes_or_buffer[, encoding[, errors]]) -> str`

要在全局使用的数据库 API 参数样式。

连接

您可以在文中找到有关 Amazon Redshift Python 连接器的连接 API 操作的说明。

`__init__(user, password, database[, host, …])`

初始化原始连接对象。

`cursor`

创建绑定到此连接的游标对象。

`commit`

提交当前数据库事务。

`rollback`

回滚当前的数据库事务。

`close`

关闭数据库连接。

`execute(cursor, operation, vals)`

运行指定的 SQL 命令。您可以将参数作为序列或映射提供，具体取决于 `redshift_connector.paramstyle` 的值。

`run(sql[, stream])`

运行指定的 SQL 命令。或者，您也可以提供与 COPY 命令一起使用的流。

`xid(format_id, global_transaction_id, …)`

创建事务 ID。postgres 中只使用了 `global_transaction_id` 参数。postgres 中不使用 format_id 和 branch_qualifier。`global_transaction_id` 可以是 postgres 支持的任何返回元组的字符串标识符（`format_id`、`global_transaction_id`、`branch_qualifier`）。

`tpc_begin(xid)`

使用由格式 ID、全局事务 ID 和分支限定符组成的事务 ID `xid` 开始 TPC 事务。

`tpc_prepare`

执行以 .tpc_begin 开始的事务的第一阶段。

`tpc_commit([xid])`

在没有参数的情况下进行调用时，.tpc_commit 会提交之前使用 .tpc_prepare() 准备的 TPC 事务。

`tpc_rollback([xid])`

在没有参数的情况下进行调用时，.tpc_rollback 会回滚 TPC 事务。

`tpc_recover`

返回适合与 .tpc_commit(xid) 或 .tpc_rollback(xid) 一起使用的待处理事务 ID 列表。

Cursor

在下文中，您可以找到有关 游标 API 操作的描述。

`__init__(connection[, paramstyle])`

初始化原始游标对象。

`insert_data_bulk(filename, table_name, parameter_indices, column_names, delimiter, batch_size)`

运行批量 INSERT 语句。

`execute(operation[, args, stream, …])`

运行数据库操作。

`executemany(operation, param_sets)`

准备数据库操作，然后为提供的所有参数序列或映射运行该操作。

`fetchone`

获取查询结果集的下一行。

`fetchmany([num])`

获取查询结果的下一个行集。

`fetchall`

获取查询结果的所有剩余行。

`close`

立即关闭光标。

`__iter__`

可以迭代游标对象以从查询中检索行。

`fetch_dataframe([num])`

返回最后查询结果的数据框。

`write_dataframe(df, table)`

将相同的结构数据框写入 Amazon Redshift 数据库。

`fetch_numpy_array([num])`

返回最后查询结果的 NumPy 数组。

`get_catalogs`

Amazon Redshift 不支持来自单个连接的多个目录。Amazon Redshift 只返回当前目录。

`get_tables([catalog, schema_pattern, …])`

返回系统中用户定义的唯一公共表。

`get_columns([catalog, schema_pattern, …])`

返回 Amazon Redshift 数据库中特定表中所有列的列表。

### AdfsCredentialsProvider 插件

以下是 Amazon Redshift Python 连接器的 AdfsCredentialsProvider 插件 API 操作的语法。

```
redshift_connector.plugin.AdfsCredentialsProvider()
```

AzureCredentialsProvider 插件

以下是 Amazon Redshift Python 连接器的 AzureCredentialsProvider 插件 API 操作的语法。

```
redshift_connector.plugin.AzureCredentialsProvider()
```

BrowserAzureCredentialsProvider 插件

以下是 Amazon Redshift Python 连接器的 BrowserAzureCredentialsProvider 插件 API 操作的语法。

```
redshift_connector.plugin.BrowserAzureCredentialsProvider()
```

BrowserSamlCredentialsProvider 插件

以下是 Amazon Redshift Python 连接器的 BrowserSamlCredentialsProvider 插件 API 操作的语法。

```
redshift_connector.plugin.BrowserSamlCredentialsProvider()
```

OktaCredentialsProvider 插件

以下是 Amazon Redshift Python 连接器的 OktaCredentialsProvider 插件 API 操作的语法。

```
redshift_connector.plugin.OktaCredentialsProvider()
```

PingCredentialsProvider 插件

以下是 Amazon Redshift Python 连接器的 PingCredentialsProvider 插件 API 操作的语法。

```
redshift_connector.plugin.PingCredentialsProvider()
```

SamlCredentialsProvider 插件

以下是 Amazon Redshift Python 连接器的 SamlCredentialsProvider 插件 API 操作的语法。

```
redshift_connector.plugin.SamlCredentialsProvider()
```

## 适用于 Apache Spark 的 Amazon Redshift 集成

Apache Spark 是一个分布式处理框架和编程模型，可帮助您进行机器学习、流处理或图形分析。Spark 与 Apache Hadoop 类似，也是一款常用于大数据工作负载的开源、分布式处理系统。Spark 具有优化的有向无环图（DAG）执行引擎，可主动在内存中缓存数据。这可以提高性能，特别适合某些算法和交互式查询。

此集成为您提供了 Spark 连接器，您可以将其用于构建 Apache Spark 应用程序，这些应用程序在 Amazon Redshift 和 Amazon Redshift Serverless 中读取和写入数据。这些应用程序不会影响应用程序性能或数据的事务一致性。此集成自动包括在 Amazon EMR 和 AWS Glue 中，因此您可以立即运行 Apache Spark 作业，在数据摄取和转换管道过程中访问数据并将其加载到 Amazon Redshift 中。

目前，可以使用 Spark 的 3.3.x、3.4.x、3.5.x、4.0.x 和 4.1.x 版本进行此集成。

此集成提供以下内容：

- AWS Identity and Access Management（IAM）身份验证 有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。
- 利用谓词和查询下推来提高性能。
- Amazon Redshift 数据类型。
- 与 Amazon Redshift 和 Amazon Redshift Serverless 的连接。

### 使用 Spark 连接器时的注意事项和限制

- tempdir URI 指向 Amazon S3 位置。此临时目录不会自动清理，可能会增加额外的成本。我们建议使用《Amazon Simple Storage Service 用户指南》中的 Amazon S3 生命周期策略，定义 Amazon S3 存储桶的保留规则。
- 原定设置情况下，如果 S3 桶和 Redshift 集群位于不同的 AWS 区域，则 Amazon S3 和 Redshift 之间的复制不起作用。要使用单独的 AWS 区域，请将 `tempdir_region` 参数设置为对 `tempdir` 使用的 S3 存储桶的区域。
- 如果使用 `tempformat` 参数写入 Parquet 数据，则在 S3 和 Redshift 之间进行跨区域写入。
- 我们建议使用 Amazon S3 服务器端加密以加密使用的 Amazon S3 存储桶。
- 我们建议阻止对 Amazon S3 存储桶的公有访问。
- 我们建议不要公开访问 Amazon Redshift 集群。

- 我们建议启用 Amazon Redshift 审核日志记录。
- 我们建议启用 Amazon Redshift 静态加密。
- 我们建议您为从 Spark on Amazon EMR 到 Amazon Redshift 的 JDBC 连接启用 SSL。
- 我们建议使用参数 `aws_iam_role` 为 Amazon Redshift 身份验证参数传递 IAM 角色。

使用 Spark 连接器进行身份验证

下图描述了 Amazon S3、Amazon Redshift、Spark 驱动程序和 Spark 执行程序之间的身份验证。

Redshift 和 Spark 之间的身份验证

您可以使用 Amazon Redshift 提供的 JDBC 驱动程序版本 2.x 驱动程序，通过指定登录凭证，使用 Spark 连接器连接到 Amazon Redshift。要使用 IAM，请将您的 JDBC url 配置为使用 IAM 身份验证。要从 Amazon EMR 或 AWS Glue 连接到 Redshift 集群，确保您的 IAM 角色具有检索临时 IAM 凭证所必需的权限。以下列表描述了您的 IAM 角色检索凭证和运行 Amazon S3 操作所需的所有权限。

- Redshift:GetClusterCredentials（适用于预置的 Redshift 集群）
- Redshift:DescribeClusters（适用于预置的 Redshift 集群）
- Redshift:GetWorkgroup（适用于 Amazon Redshift Serverless 工作组）
- Redshift:GetCredentials（适用于 Amazon Redshift Serverless 工作组）
- s3:ListBucket
- s3:GetBucket
- s3:GetObject
- s3:PutObject
- s3:GetBucketLifecycleConfiguration

有关 GetClusterCredentials 的更多信息，请参阅 GetClusterCredentials 的 IAM 策略。

您还必须确保 Amazon Redshift 可以在 `COPY` 和 `UNLOAD` 操作期间担任 IAM 角色。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "redshift.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

如果您使用的是最新的 JDBC 驱动程序，则驱动程序将自动管理从 Amazon Redshift 自签名证书到 ACM 证书的转换。但是，您必须为 JDBC url 指定 SSL 选项。

以下是有关如何指定 JDBC 驱动程序 URL 和 aws_iam_role 以连接到 Amazon Redshift 的示例。

```
df.write \
  .format("io.github.spark_redshift_community.spark.redshift ") \
  .option("url", "jdbc:redshift:iam://<the-rest-of-the-connection-string>") \
  .option("dbtable", "<your-table-name>") \
  .option("tempdir", "s3a://<your-bucket>/<your-directory-path>") \
  .option("aws_iam_role", "<your-aws-role-arn>") \
  .mode("error") \
  .save()
```

Amazon S3 和 Spark 之间的身份验证

如果您使用 IAM 角色在 Spark 和 Amazon S3 之间进行身份验证，则使用以下方法之一：

- AWS SDK for Java 会自动尝试使用由 DefaultAWSCredentialsProviderChain 类实施的默认凭证提供程序链来查找 AWS 凭证。有关更多信息，请参阅使用默认凭证提供程序链。
- 您可以通过 Hadoop 配置属性指定 AWS 密钥。例如，如果您的 tempdir 配置指向 s3n:// 文件系统，请在 Hadoop XML 配置文件中设置 fs.s3n.awsAccessKeyId 和 fs.s3n.awsSecretAccessKey 属性或调用 sc.hadoopConfiguration.set() 以更改 Spark 的全局 Hadoop 配置。

例如，如果您使用的是 s3n 文件系统，则添加：

```
sc.hadoopConfiguration.set("fs.s3n.awsAccessKeyId", "YOUR_KEY_ID")
sc.hadoopConfiguration.set("fs.s3n.awsSecretAccessKey", "YOUR_SECRET_ACCESS_KEY")
```

对于 s3a 文件系统，请添加：

```
sc.hadoopConfiguration.set("fs.s3a.access.key", "YOUR_KEY_ID")
sc.hadoopConfiguration.set("fs.s3a.secret.key", "YOUR_SECRET_ACCESS_KEY")
```

如果您使用的是 Python，则使用以下操作：

```
sc._jsc.hadoopConfiguration().set("fs.s3n.awsAccessKeyId", "YOUR_KEY_ID")
sc._jsc.hadoopConfiguration().set("fs.s3n.awsSecretAccessKey",
 "YOUR_SECRET_ACCESS_KEY")
```

- 在 `tempdir` URL 中对身份验证密钥进行编码。例如，URI `s3n://ACCESSKEY:SECRETKEY@bucket/path/to/temp/dir` 对密钥对（`ACCESSKEY`，`SECRETKEY`）进行编码。

Redshift 和 Amazon S3 之间的身份验证

如果您在查询中使用 COPY 和 UNLOAD 命令，则还必须向 Amazon S3 授予访问 Amazon Redshift 的权限，这样才能代表您运行查询。为此，请先授权 Amazon Redshift 访问其他 AWS 服务，然后使用 IAM 角色授权 COPY 和 UNLOAD 操作。

作为最佳实践，我们建议将权限策略附加到 IAM 角色，然后根据需要将其分配给用户和组。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

与 AWS Secrets Manager 集成

您可以从 AWS Secrets Manager 中存储的密钥检索您的 Redshift 用户名和密码凭证。要自动提供 Redshift 凭证，请使用 `secret.id` 参数。有关如何创建 Redshift 凭证密钥的更多信息，请参阅创建 AWS Secrets Manager 数据库密钥。

| GroupID | ArtifactID | 支持的版本 | 说明 |
| --- | --- | --- | --- |
| com.amazonaws.secretsmanager | aws-secretsmanager-jdbc | 1.0.12 | 通过适用于 Java 的 AWS Secrets Manager SQL 连接库，Java 开发人员可以使用存储在 AWS Secrets Manager 中 |

| GroupID | ArtifactID | 支持的版本 | 说明 |
| --- | --- | --- | --- |
| | | | 的密钥轻松连接到 SQL 数据库。 |

> **Note**
>
> 致谢：本文档包含 Apache Software Foundation 根据 Apache 2.0 许可证的许可而开发的示例代码和语言。

通过下推提高性能

Spark 连接器自动应用谓词和查询下推来优化性能。有了这种支持就意味着，如果您在查询中使用支持的函数，Spark 连接器会将该函数转换成 SQL 查询，并在 Amazon Redshift 中运行该查询。这种优化会减少需要检索的数据，因此 Apache Spark 可以处理更少的数据并获得更好的性能。默认情况下，自动激活下推。要停用它，请将 `autopushdown` 设置为 false。

```
import sqlContext.implicits._val
 sample= sqlContext.read
    .format("io.github.spark_redshift_community.spark.redshift")
    .option("url",jdbcURL )
    .option("tempdir", tempS3Dir)
    .option("dbtable", "event")
    .option("autopushdown", "false")
    .load()
```

下推支持以下函数。如果您使用不在此列表中的函数，Spark 连接器将在 Spark 中（而不是在 Amazon Redshift 中）执行此函数，从而导致性能未优化。有关 Spark 中函数的完整列表，请参阅内置函数。

- 聚合函数
  - avg
  - count
  - max
  - min
  - sum
  - stddev_samp

- stddev_pop
- var_samp
- var_pop

- 布尔运算符
  - in
  - isnull
  - isnotnull
  - contains
  - endswith
  - startswith
- 逻辑运算符
  - and
  - or
  - not (or !)
- 数学函数
  - +
  - -
  - *
  - /
  - - (unary)
  - abs
  - acos
  - asin
  - atan
  - ceil
  - cos
  - EXP
  - floor
  - greatest
  - least

- log10
- pi
- pow
- round
- sin
- sqrt
- tan

- 其他函数
  - cast
  - coalesce
  - decimal
  - if
  - in
- 关系运算符
  - !=
  - =
  - >
  - >=
  - <
  - <=
- 字符串函数
  - ascii
  - lpad
  - rpad
  - 翻译
  - upper
  - lower
  - length
  - trim
  - ltrim

- rtrim
- like
- substring
- concat

- 日期和时间函数
  - add_months
  - date
  - date_add
  - date_sub
  - date_trunc
  - timestamp
  - trunc
- 数学运算
  - CheckOverflow
  - PromotePrecision
- 关系运算
  - Aliases（例如，AS）
  - CaseWhen
  - Distinct
  - InSet
  - 联接和交叉联接
  - Limits
  - Unions，union all
  - ScalarSubquery
  - Sorts（升序和降序）
  - UnscaledValue

## 其他配置选项

在本页中，您可以找到可为 Amazon Redshift Spark 连接器指定的选项的说明。

字符串列的最大大小

Redshift 在创建表时将字符串列创建为文本列，它们存储为 VARCHAR(256)。如果您想要支持更大大小的列，则可以使用 maxlength 来指定字符串列的最大长度。下面是说明如何指定 `maxlength` 的示例。

```
columnLengthMap.foreach { case (colName, length) =>
  val metadata = new MetadataBuilder().putLong("maxlength", length).build()
  df = df.withColumn(colName, df(colName).as(colName, metadata))
}
```

列类型

要设置列类型，请使用 `redshift_type` 字段。

```
columnTypeMap.foreach { case (colName, colType) =>
  val metadata = new MetadataBuilder().putString("redshift_type", colType).build()
  df = df.withColumn(colName, df(colName).as(colName, metadata))
}
```

列上的压缩编码

要对列使用特定的压缩编码，请使用编码字段。有关支持的压缩编码的完整列表，请参阅压缩编码。

列的描述

要设置描述，请使用 `description` 字段。

Redshift 和 Amazon S3 之间的身份验证

默认情况下，结果将以 Parquet 格式转存到 Amazon S3。要以竖线分隔的文本文件转存结果，请指定以下选项。

```
.option("unload_s3_format", "TEXT")
```

下推语句

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| spark.datasource.redshift.community. | 否 | True | 指定连接器是否应延时运行下推语句 Redshift。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| autopushdown.lazyMode | | | 如果为 true，则 spark 连接器会在运行查询之前检索所有相关的模型和信息，这通常会产生更好的性能。<br><br>如果为 false，则 spark 连接器会立即在 Spark 驱动程序主线程中运行下推语句，并跨表达式进行序列化。 |

连接器参数

Spark SQL 中的参数映射或 OPTIONS 支持以下设置。

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| dbtable | 是，除非指定了查询 | 不适用 | 要在 Redshift 中创建或读取的表。将数据保存回 Redshift 时需要此参数。 |
| 查询 | 是，除非指定了 dbtable | 不适用 | 要在 Redshift 中读取的查询。 |
| 用户 | 否 | 不适用 | Redshift 用户名。必须与 password 参数一起使用。仅当 user 和 password 不是 URL 中的参数时才有效。同时使用两者会引发错误。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| password | 否 | 不适用 | Redshift 密码。必须与 user 参数一起使用。仅当 user 和 password 不是 URL 中的参数时才有效。同时使用两者会引发错误。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| url | 否 | 不适用 | JDBC URL。格式为 jdbc:subprotocol://host:port/database?user=username&password=password。<br><br>根据您加载的 JDBC 驱动程序，子协议可以是 postgresql 或 Redshift。请注意，类路径中必须有一个兼容 Redshift 的驱动程序并且与此 URL 相匹配。<br><br>Host 和 port 应指向 Redshift 主节点，因此您必须配置安全组和/或 VPC 以允许从您的驱动程序应用程序访问。<br><br>数据库是 Redshift 数据库名称。<br><br>用户名和密码是用于访问数据库的凭证，必须将其嵌入到此 JDBC 的 URL 中，并且您的数据库用户必须具有访问该表所需的权限。 |
| aws_iam_role | 仅在使用 IAM 角色授权 Redshift COPY/UNLOAD 操作时 | 不适用 | 连接到 Redshift 集群的 IAM 角色的完全指定 ARN。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| forward_spark_s3_credentials | 否 | False | 指示此库是否应自动发现 Spark 用于连接到 Amazon S3 的凭证，以及是否通过 JDBC 驱动程序将这些凭证转发给 Redshift。这些凭证作为 JDBC 查询的一部分发送。因此，在使用此选项时，我们建议您启用 JDBC 连接的 SSL 加密。 |
| temporary_aws_access_key_id | 否 | 不适用 | AWS 访问密钥。必须具有 S3 存储桶的写入权限。 |
| temporary_aws_secret_access_key | 否 | 不适用 | AWS对应于访问密钥的 秘密访问密钥。 |
| temporary_aws_session_token | 否 | 不适用 | AWS对应于提供的访问密钥的 会话令牌。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| tempdir | 否 | 不适用 | Amazon S3 中的可写入位置。用于在读取时转存数据，以及在写入时将 Avro 数据加载到 Redshift 中。如果您在常规 ETL 管道中使用适用于 Spark 的 Redshift 数据来源，则在存储桶上设置生命周期策略并将其用作这些数据的临时位置可能会很有用。 |
| jdbcdriver | 否 | 由 JDBC URL 的子协议确定 | 要使用的 JDBC 驱动程序的类名称。此类必须位于类路径上。在大多数情况下，不必指定此选项，因为相应的驱动程序类名应由 JDBC URL 的子协议自动确定。 |
| diststyle | 否 | Even | 创建表时使用的 Redshift 分配方式。有效选项为 EVEN、KEY 或 ALL。使用 KEY 时，还必须使用 distkey 选项设置分配键。 |
| distkey | 否，除非使用 DISTSTYLE_KEY | 不适用 | 表中的列的名称，在创建表时用作分配键。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| sortkeyspec | 否 | 不适用 | 完整的 Redshift 排序键定义。 |
| include_column_list | 否 | False | 指示此库是否应根据列映射选项自动从 Schema 中提取列并将其添加到 COPY 命令。 |
| 描述 | 否 | 不适用 | 表的描述。使用 SQL COMMENT 命令设置描述，并在大多数查询工具中显示。查看 `description` 元数据以设置各个列的描述。 |
| preactions | 否 | 不适用 | 在加载 COPY 命令之前要运行的以分号分隔的 SQL 命令列表。在加载新数据之前运行 DELETE 命令或类似命令时可能会很有用。如果命令包含 %s，则表名将在运行时之前格式化（如果您使用的是暂存表）。如果此命令失败，则将其视为异常。如果您使用的是暂存表，则在预处理失败时恢复更改并还原备份表。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| extracopyoptions | 否 | 不适用 | 加载数据时要附加到 Redshift COPY 命令的额外选项列表（例如 TRUNCATECOLUMNS 或 MAXERROR n）。有关可用参数的完整列表，请参阅可选参数。<br><br>请注意，由于这些选项附加到 COPY 命令的末尾，因此只能使用在命令末尾有意义的选项。这应该涵盖最有可能的使用案例。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| sse_kms_key | 否 | 不适用 | 在 Redshift UNLOAD 操作期间，为 S3 中的服务器端加密使用 AWS KMS 密钥 ID，而不是使用 AWS 默认加密。Redshift IAM 角色必须能够访问 KMS 密钥，这样才能使用它进行写入，而 Spark IAM 角色必须能够访问密钥，这样才能执行读取操作。只要 Spark 的 IAM 角色具有适当的访问权限，读取加密的数据就不需要更改（AWS 处理此任务）。 |
| 临时格式 | 否 | AVRO | 写入 Redshift 时在 Amazon S3 中保存临时文件的格式。有效值为 AVRO、CSV、CSV GZIP（压缩的 CSV）和 PARQUET。 |
| cvsnullstring（实验） | 否 | Null | 使用 CSV 临时格式时要为空值写入的字符串值。这应该是一个不会出现在实际数据中的值。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| autopushdown | 否 | True | 指示是否通过捕获和分析 SQL 操作的 Spark 逻辑计划来应用谓词和查询下推。这些操作转换为 SQL 查询，然后在 Redshift 中运行以提高性能。 |
| autopushdown.s3_result_cache | 否 | False | 缓存查询 SQL 以在内存中转存数据 Amazon S3 路径映射，这样就无需在同一 Spark 会话中再次运行相同的查询。仅在开启自动下推时支持。因为缓存的结果可能包含陈旧信息，我们不建议在混合读取和写入操作时使用此参数。 |
| unload_s3_format | 否 | Parquet | 用于转存查询结果的格式。有效选项为 Parquet 和 Text，后者指定以竖线分隔的文本格式转存查询结果。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| extraunloadoptions | 否 | 不适用 | 附加到 Redshift UNLOAD 命令的额外选项。并非所有选项都能确保正常工作，因为某些选项可能与连接器中设置的其他选项发生冲突。 |
| copydelay | 否 | 30000 | Redshift COPY 操作两次重试之间的延迟（以毫秒为单位）。 |
| copyretrycount | 否 | 2 | 重试 Redshift COPY 操作的次数。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| tempdir_region | 否 | 不适用 | tempdir 所在的 AWS 区域。设置此选项可提高与 tempdir 交互的连接器性能，以及在连接器的读取和写入操作期间，自动提供此值作为 COPY 和 UNLOAD 操作的一部分。<br><br>此设置推荐用于以下情况？<br><br>1) 当连接器在 AWS 外部运行时，此时自动区域发现将失败并对连接器性能产生负面影响。<br><br>2) 当 tempdir 与 Redshift 集群位于不同的区域时，此时使用此设置可以缓解使用 extracopyoptions 和 extraunloadoptions 参数手动提供此区域的需要。当使用 PARQUET 作为 tempformat 时，tempdir 不能位于与 Redshift 集群不同的区域中，即使使用此参数也是如此。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| | | | 3) 当连接器运行在与 tempdir 不同的区域中时，此时可以提高连接器对 tempdir 的访问性能。 |
| secret.id | 否 | 不适用 | 存储在 AWS Secrets Manager 中的密钥的名称或 ARN。您可以使用此参数自动提供 Redshift 凭证，但前提是用户、密码和 DbUser 凭证不会传递到 JDBC URL 或作为其他选项传递。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| secret.region | 否 | 不适用 | 要搜索 `secret.id` 值的主要 AWS 区域，例如美国东部（弗吉尼亚州北部）。<br><br>如果您未指定此区域，连接器将尝试使用原定设置凭证提供者链以解析 `secret.id` 的区域。在某些情况下，例如，如果您在外部使用连接器，则连接器将无法找到该区域。我们建议在以下情况下使用此设置：<br><br>1) 当连接器在 AWS 外部运行时，此时自动区域发现功能将失败并阻止使用 Redshift 进行身份验证<br><br>当连接器在与 `secret.id` 不同的区域中运行时，此时可以提高连接器对密钥的访问性能。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| secret.vpcEndpointUrl | 否 | 不适用 | 覆盖原定设置凭证提供者链时 AWS Secrets Manager 的 PrivateLink DNS 端点 URL。 |
| secret.vpcEndpoint Region | 否 | 不适用 | 覆盖原定设置凭证提供者链时 AWS Secrets Manager 的 PrivateLink DNS 端点区域。 |
| jdbc.* | 否 | 不适用 | 要传递给底层 JDBC 驱动程序的其他参数，其中通配符是 JDBC 参数的名称，例如 jdbc.ssl。请注意，jdbc 前缀在传递给 JDBC 驱动程序之前将被删除。要查看 Redshift JDBC 驱动程序的所有可能选项，请参阅 JDBC 驱动程序版本 2.x 配置的选项。 |

| 参数 | 必需 | 默认值 | 说明 |
| --- | --- | --- | --- |
| 标签 | 否 | " " | 使用连接器运行查询时要包含在查询组集中的标识符。必须为 100 个或更少的字符，并且所有字符都必须是有效的 unicodeId entifierP arts 。如果您的标识符超过 100 个字符，则多余的字符将被删除。使用连接器运行查询时，查询组将设置为 JSON 格式的字符串，例如<br>`{"spark-redshift-connector":{"svc":"","ver":"5.1.0-amzn-1-spark_3.3","op":"Read","lbl":""}}`)<br>。此选项将替换 lbl 密钥的值。 |

> **Note**
>
> 致谢：本文档包含 Apache Software Foundation 根据 Apache 2.0 许可证的许可而开发的示例代码和语言。

### 支持的数据类型

Spark 连接器支持 Amazon Redshift 中的以下数据类型。有关 Amazon Redshift 中支持的数据类型的完整列表，请参阅数据类型。如果某个数据类型不在下表中，则 Spark 连接器不支持该数据类型。

| 数据类型 | 别名 |
| --- | --- |
| SMALLINT | INT2 |
| INTEGER | INT、INT4 |
| BIGINT | INT8 |
| DECIMAL | NUMERIC |
| REAL | FLOAT4 |
| DOUBLE PRECISION | FLOAT8、FLOAT |
| BOOLEAN | BOOL |
| CHAR | CHARACTER、NCHAR、BPCHAR |
| VARCHAR | CHARACTER VARYING、NVARCHAR、TEXT |
| DATE | |
| TIMESTAMP | Timestamp without time zone |
| TIMESTAMPTZ | Timestamp with time zone |
| SUPER | |
| TIME | TIME WITHOUT TIME ZONE |
| TIMETZ | Time with time zone |
| VARBYTE | VARBINARY，BINARY VARYING |

复杂的数据类型

您可以使用 spark 连接器在 Redshift SUPER 数据类型列中读写 Spark 复杂数据类型，如 ArrayType、MapType 和 StructType。如果您在读取操作期间提供架构，则该列中的数据将在 Spark 中转换为相应的复杂类型，包括任何嵌套类型。此外，如果启用 autopushdown，嵌套属性、映射值和数组索引的投影将下推到 Redshift，这样，当只访问一部分数据时，就不再需要卸载整个嵌套数据结构。

从连接器写入 DataFrame 时，任何类型为 MapType（使用 StringType）、StructType 或 ArrayType 的列都会写入 Redshift SUPER 数据类型列。在写入这些嵌套数据结构时，tempformat 参数必须为类型 CSV、CSV GZIP 或 PARQUET。使用 AVRO 将导致异常。写入一个键类型不是 StringType 的 MapType 数据结构也会导致异常。

StructType

以下示例演示如何使用包含结构的 SUPER 数据类型创建表

```
create table contains_super (a super);
```

然后，您可以使用连接器，使用下面示例中的类似架构，从表中的 SUPER 列 a 查询 StringType 字段 hello。

```
import org.apache.spark.sql.types._

val sc = // existing SparkContext
val sqlContext = new SQLContext(sc)

val schema = StructType(StructField("a", StructType(StructField("hello",
 StringType) ::Nil)) :: Nil)

val helloDF = sqlContext.read
.format("io.github.spark_redshift_community.spark.redshift")
.option("url", jdbcURL )
.option("tempdir", tempS3Dir)
.option("dbtable", "contains_super")
.schema(schema)
.load().selectExpr("a.hello")
```

以下示例演示如何向列 a 写入结构。

```
import org.apache.spark.sql.types._
```

```
import org.apache.spark.sql._

val sc = // existing SparkContext
val sqlContext = new SQLContext(sc)

val schema = StructType(StructField("a", StructType(StructField("hello",
 StringType) ::Nil)) :: Nil)
val data = sc.parallelize(Seq(Row(Row("world"))))
val mydf = sqlContext.createDataFrame(data, schema)

mydf.write.format("io.github.spark_redshift_community.spark.redshift").
option("url", jdbcUrl).
option("dbtable", tableName).
option("tempdir", tempS3Dir).
option("tempformat", "CSV").
mode(SaveMode.Append).save
```

### MapType

如果您更喜欢使用 MapType 来表示数据，那么您可以在架构中使用 MapType 数据结构，并检索映射中与键对应的值。请注意，MapType 数据结构中的所有键都必须是 String 类型，并且所有值都必须是相同的类型，例如 int。

以下示例演示如何获取列 a 中键 hello 的值。

```
import org.apache.spark.sql.types._

val sc = // existing SparkContext
val sqlContext = new SQLContext(sc)

val schema = StructType(StructField("a", MapType(StringType, IntegerType))::Nil)

val helloDF = sqlContext.read
    .format("io.github.spark_redshift_community.spark.redshift")
    .option("url", jdbcURL )
    .option("tempdir", tempS3Dir)
    .option("dbtable", "contains_super")
    .schema(schema)
    .load().selectExpr("a['hello']")
```

### ArrayType

如果该列包含数组而不是结构，则可以使用连接器查询数组中的第一个元素。

```
import org.apache.spark.sql.types._

val sc = // existing SparkContext
val sqlContext = new SQLContext(sc)

val schema = StructType(StructField("a", ArrayType(IntegerType)):: Nil)

val helloDF = sqlContext.read
    .format("io.github.spark_redshift_community.spark.redshift")
    .option("url", jdbcURL )
    .option("tempdir", tempS3Dir)
    .option("dbtable", "contains_super")
    .schema(schema)
    .load().selectExpr("a[0]")
```

限制

通过 spark 连接器使用复杂数据类型有以下限制：

- 所有嵌套的结构字段名称和映射键必须为小写。如果查询带有大写字母的复杂字段名称，可以尝试省略架构，并使用 `from_json` spark 函数在本地转换返回的字符串来作为解决方法。
- 在读取或写入操作中使用的任何映射字段都必须只有 `StringType` 键。
- 只有 `CSV`、`CSV GZIP` 和 `PARQUET` 是支持将复杂类型写入 Redshift 的临时格式值。尝试使用 `AVRO` 会引发异常。

## 为 Amazon Redshift 配置 ODBC 驱动程序版本 2.x 连接

您可以使用 ODBC 连接将许多第三方 SQL 客户端工具和应用程序连接到您的 Amazon Redshift 集群。如果您的客户端工具支持 JDBC，您可以选择使用 JDBC 类型的连接，而非 ODBC 连接，因为 JDBC 连接更加易于配置。但是，如果您的客户端工具不支持 JDBC，您可以按照本节中的步骤在您的客户端计算机或 Amazon EC2 实例上设置 ODBC 连接。

Amazon Redshift 提供了适用于 Linux、Windows 和 Mac 操作系统的 64 位 ODBC 驱动程序；32 位 ODBC 驱动程序已停用。除了紧急安全补丁外，不会发布针对 32 位 ODBC 驱动程序的进一步更新。

有关 ODBC 驱动程序更改的最新信息，请参阅更改日志。

主题

- 获取 ODBC URL
- 在 Microsoft Windows 上使用 Amazon Redshift ODBC 驱动程序

- 在 Linux 上使用 Amazon Redshift ODBC 驱动程序
- 在 Apple macOS 上使用 Amazon Redshift ODBC 驱动程序
- 身份验证方法
- 数据类型转换
- ODBC 驱动程序选项
- 早期 ODBC 驱动程序版本
- ODBC 驱动程序 2.x 版本与 1.x 版本之间的差异

获取 ODBC URL

Amazon Redshift 在 Amazon Redshift 控制台中显示您的集群的 ODBC URL。此 URL 包含在您的客户端计算机与数据库之间建立连接所需的信息。

ODBC URL 采用以下格式：

```
Driver={driver}; Server=endpoint_host; Database=database_name; UID=user_name;
 PWD=password; Port=port_number
```

上述格式的字段具有以下值：

| Field | Value |
| --- | --- |
| *Driver* | 要使用的 64 位 ODBC 驱动程序的名称：Amazon Redshift ODBC 驱动程序 (x64) |
| *Server* | Amazon Redshift 集群的端点主机。 |
| *Database* | 您为集群创建的数据库。 |
| *UID* | 有权连接到数据库的数据库用户账户的用户名。虽然此值是数据库级权限，而非集群级权限，但您可以使用您在启动集群时设置的 Redshift 管理员用户账户。 |
| *PWD* | 数据库用户账户用于连接数据库的密码。 |
| *Port* | 您在启动集群时指定的端口号。如果您启用了防火墙，请确保此端口处于打开状态，可供您使用。 |

以下是一个示例 ODBC URL：

```
Driver={Amazon Redshift ODBC Driver (x64)}; Server=examplecluster.abc123xyz789.us-
west-2.redshift.amazonaws.com; Database=dev; UID=adminuser;
 PWD=insert_your_admin_user_password_here; Port=5439
```

有关在何处查找 ODBC URL 的信息，请参阅查找集群连接字符串。

在 Microsoft Windows 上使用 Amazon Redshift ODBC 驱动程序

您必须在可访问 Amazon Redshift 数据仓库的客户端计算机上安装 Amazon Redshift ODBC 驱动程序。对于您要在其上安装该驱动程序的每台计算机，有以下最低要求：

- 计算机上的管理员权限。
- 计算机满足以下系统要求：
  - 以下操作系统之一：
    - Windows 10 或 8.1。
    - Windows Server 2025、2022、2019、2016 或 2012。
  - 100MB 可用磁盘空间。
  - 已安装适用于 64 位 Windows 的 Visual C++ Redistributable for Visual Studio 2015。您可以在 Microsoft 网站上的下载 Visual C++ Redistributable for Visual Studio 2022 下载安装包。

下载并安装 Amazon Redshift ODBC 驱动程序

使用以下过程下载并安装适用于 Windows 操作系统的 Amazon Redshift ODBC 驱动程序。仅在您当前运行的第三方应用程序获得了使用 Amazon Redshift 的认证并且需要不同驱动程序时，才能使用该特定驱动程序。

要下载并安装 ODBC 驱动程序，请执行以下操作：

1. 下载以下驱动程序：64 位 ODBC 驱动程序版本 2.2.4.0

   此驱动程序的名称为 Amazon Redshift ODBC 驱动程序(x64)。
2. 审核 Amazon Redshift ODBC 驱动程序版本 2.x 许可证。
3. 双击 .msi 文件，然后按照向导中的步骤安装驱动程序。

为 ODBC 连接创建系统 DSN 条目

下载并安装 ODBC 驱动程序后，将数据来源名称 (DSN) 条目添加到客户端计算机或 Amazon EC2 实例。SQL 客户端工具可以使用此数据来源连接到 Amazon Redshift 数据库。

建议您创建系统 DSN 而不是用户 DSN。有些应用程序使用不同的数据库用户账户加载数据，因而可能无法检测在其他数据库用户账户下创建的用户 DSN。

> **Note**
>
> 对于使用 AWS Identity and Access Management (IAM) 凭证或身份提供者 (IdP) 凭证进行的身份验证，需要执行其他步骤。有关更多信息，请参阅配置 JDBC 或 ODBC 连接以使用 IAM 凭证。

要为 ODBC 连接创建系统 DSN 条目，请执行以下操作：

1. 在开始菜单上，键入“ODBC 数据来源”。选择 ODBC 数据来源。

   请确保您选择的 ODBC Data Source Administrator 的位数与用于连接 Amazon Redshift 的客户端应用程序的位数相同。
2. 在 ODBC 数据来源管理器中，选择驱动程序选项卡，然后找到以下驱动程序文件夹：Amazon Redshift ODBC 驱动程序(x64)。
3. 选择系统 DSN 选项卡为计算机上的所有用户配置驱动程序，或选择用户 DSN 选项卡仅为您的数据库用户账户配置驱动程序。
4. 选择添加。系统随即打开 Create New Data Source 窗口。
5. 选择 Amazon Redshift ODBC 驱动程序(x64)，然后选择完成。系统随即打开 Amazon Redshift ODBC Driver DSN Setup 窗口。
6. 在连接设置部分下，输入以下信息：
   - Data source name

     输入数据来源的名称。例如，如果您遵循的是《Amazon Redshift 入门指南》，则可键入 `exampleclusterdsn`，以便轻松记住将与此 DSN 关联的集群。
   - 服务器

     为您的 Amazon Redshift 集群指定端点主机。您可以在 Amazon Redshift 控制台中的集群详细信息页面上找到该信息。有关更多信息，请参阅在 Amazon Redshift 中配置连接。

- 端口

  输入数据库使用的端口号。根据您在创建、修改或迁移集群时选择的端口，允许访问所选端口。

- 数据库

  输入 Amazon Redshift 数据库的名称。如果您在未指定数据库名称的情况下启动了集群，请输入 dev。否则，请使用您在启动过程中选择的名称。如果您遵循的是《Amazon Redshift 入门指南》，输入 dev。

7. 在身份验证下，指定配置选项以配置标准或 IAM 身份验证。
8. 选择SSL 选项，然后指定以下项目的值：
   - 身份验证模式

     选择处理安全套接字层 (SSL) 的模式。在测试环境中，可以使用 prefer。但是，对于生产环境以及在需要安全交换数据时，请使用 verify-ca 或 verify-full。

   - 最低 TLS

     （可选）选择驱动程序允许数据存储用来对连接进行加密的最低 TLS/SSL 版本。例如，如果您指定了 TLS 1.2，则无法使用 TLS 1.1 对连接进行加密。默认版本为 TLS 1.2。
9. 在代理选项卡中，指定任何代理连接设置。
10. 在游标选项卡中，指定有关如何将查询结果返回至您的 SQL 客户端工具或应用程序的选项。
11. 在高级选项中，指定 logLevel、logPath、compression 和其他选项的值。
12. 选择测试。如果客户端计算机可以连接到 Amazon Redshift 数据库，将显示以下消息：连接成功。如果客户端计算机无法连接到数据库，您可以通过生成日志文件并联系 AWS 支持部门，对可能的问题进行故障排除。有关生成日志的信息，请参阅 (LINK)。
13. 选择确定。

### 在 Linux 上使用 Amazon Redshift ODBC 驱动程序

您必须在可访问 Amazon Redshift 数据仓库的客户端计算机上安装 Amazon Redshift ODBC 驱动程序。对于您要在其上安装该驱动程序的每台计算机，有以下最低要求：

- 计算机上的根访问权限。
- 以下分发之一：
  - Amazon Linux 2
  - Amazon Linux 2023

  - Red Hat® Enterprise Linux®（RHEL）8 或更高版本
  - CentOS 8 或更高版本
  - Debian 11 或更高版本
- 150MB 可用磁盘空间。
- unixODBC 2.2.14 或更高版本。
- glibc 2.26 或更高版本。

下载并安装 Amazon Redshift ODBC 驱动程序

要下载并安装适用于 Linux 的 Amazon Redshift ODBC 驱动程序版本 2.x，请执行以下操作：

1. 下载以下驱动程序：
   - x86 64 位 RPM 驱动程序版本 2.2.4.0
     - Linux 构件签名密钥：密钥
     - Linux 构件签名哈希：哈希
   - ARM 64 位 RPM 驱动程序版本 2.2.4.0
     - Linux 构件签名密钥：密钥
     - Linux 构件签名哈希：哈希
   - x86 64 位 DEB 驱动程序版本 2.2.4.0
     - Linux 构件签名密钥：密钥
     - Linux 构件签名哈希：哈希
   - ARM 64 位 DEB 驱动程序版本 2.2.4.0
     - Linux 构件签名密钥：密钥
     - Linux 构件签名哈希：哈希

   > **Note**
   >
   > 32 位 ODBC 驱动程序已停用。除了紧急安全补丁外，不会发布进一步的更新。

2. 在安装之前，请验证 AWS 对驱动程序包进行了签名。下载软件包、其签名密钥（证书）及其签名哈希（分离的签名）后，运行以下命令：

```
openssl x509 -in SignatureKeyFile -pubkey -noout > redshift-odbc-pubkey.pem
openssl dgst -sha256 -verify redshift-odbc-pubkey.pem \
```

```
    -signature SignedHashFile \
    PackageFile
```

如果软件包为正版，则 openssl 打印 Verified OK。如果输出为其它内容，请不要安装该软件包。重新下载文件或联系 AWS Support。

例如，要验证 x86 64 位 RPM，请执行以下操作：

```
openssl x509 -in AmazonRedshiftODBC-64-bit-2.2.4.0.x86_64-certificate.pem -pubkey -
noout > redshift-odbc-pubkey.pem
openssl dgst -sha256 -verify redshift-odbc-pubkey.pem \
    -signature AmazonRedshiftODBC-64-bit-2.2.4.0.x86_64-signature.bin \
    AmazonRedshiftODBC-64-bit-2.2.4.0.x86_64.rpm
```

3. 转至您下载程序包的位置，然后运行以下命令之一。使用适用于您的 Linux 发行版的命令。

   在 RHEL 和 CentOS 操作系统（RPM）上，运行以下命令：

   ```
   sudo yum --nogpgcheck localinstall RPMFileName
   ```

   将 RPMFileName 替换为 RPM 包文件名。例如，以下命令将演示如何安装 64 位驱动程序：

   ```
   sudo yum --nogpgcheck localinstall AmazonRedshiftODBC-64-bit-2.x.xx.xxxx.x86_64.rpm
   ```

   在 Debian 和 Ubuntu 操作系统（DEB）上，导航到包含已下载的 .deb 文件的目录并运行以下命令：

   ```
   sudo apt install ./DEBFileName
   ```

   将 DEBFileName 替换为 DEB 软件包文件名。例如：

   ```
   sudo apt install ./AmazonRedshiftODBC-64-bit-2.x.xx.xxxx.amd64.deb
   ```

使用 ODBC 驱动程序管理器配置 ODBC 驱动程序

在 Linux 上，您可以使用 ODBC 驱动程序管理器来配置 ODBC 连接设置。ODBC 驱动程序管理器使用配置文件来定义和配置 ODBC 数据来源和驱动程序。您可以使用的 ODBC 驱动程序管理器取决于您使用的操作系统。

使用 unixODBC 驱动程序管理器配置 ODBC 驱动程序

要配置 Amazon Redshift ODBC 驱动程序，需要以下文件：

- amazon.redshiftodbc.ini
- odbc.ini
- odbcinst.ini

如果您将驱动程序安装在默认位置，则 amazon.redshiftodbc.ini 配置文件将位于 /opt/amazon/redshiftodbcx64 中。

此外，在 /opt/amazon/redshiftodbcx64 下，您可以找到示例 odbc.ini 和 odbcinst.ini 文件。您可以使用这些文件作为配置 Amazon Redshift ODBC 驱动程序和数据来源名称 (DSN) 的示例。

我们不建议使用 Amazon Redshift ODBC 驱动程序安装目录来存储配置文件。安装目录中的示例文件仅用作示例。如果您日后重新安装 Amazon Redshift ODBC 驱动程序，或将其升级到新版本，安装目录会被覆盖。您将丢失对安装目录中的文件所做的所有更改。

为了避免出现这种情况，请将 amazon.redshiftodbc.ini 文件复制到安装目录以外的其他目录中。如果您要将此文件复制到用户的主目录，请在文件名的开头添加一个句点 (.)，使其成为隐藏文件。

对于 odbc.ini 和 odbcinst.ini 文件，应在用户的主目录中使用配置文件，或者在其他目录中创建新版本。默认情况下，您的 Linux 操作系统应在用户的主目录（/home/$USER 或 ~/.）中包含 odbc.ini 文件和 odbcinst.ini 文件。这些默认文件均为隐藏文件（通过在每个文件名的前面添加圆点 (.) 表示）。这些文件仅当您使用 -a 标志列出目录内容时显示。

对于 odbc.ini 和 odbcinst.ini 文件，不管您选择哪个选项，都需对这些文件进行修改，以添加驱动程序和 DSN 配置信息。如果您创建新文件，则还需设置环境变量，以指定这些配置文件的目标存储位置。

默认情况下，ODBC 驱动程序管理器将配置为使用位于主目录中的 odbc.ini 和 odbcinst.ini 配置文件的隐藏版本（名为 .odbc.ini 和 .odbcinst.ini）。它们也被配置为使用驱动程序安装目

录中的 amazon.redshiftodbc.ini 文件。如果您将这些配置文件存储在其他位置，请设置如下所述的环境变量，以便驱动程序管理器能够找到这些文件。

如果您使用的是 unixODBC，请执行以下操作：

- 将 ODBCINI 设置到 odbc.ini 文件的完整路径和文件名。
- 将 ODBCSYSINI 设置到包含 odbcinst.ini 文件的目录的完整路径。
- 将 AMAZONREDSHIFTODBCINI 设置到 amazon.redshiftodbc.ini 文件的完整路径和文件名。

以下是设置上述值的示例：

```
export ODBCINI=/usr/local/odbc/odbc.ini
export ODBCSYSINI=/usr/local/odbc
export AMAZONREDSHIFTODBCINI=/etc/amazon.redshiftodbc.ini
```

在 Linux 上使用数据来源名称 (DSN) 配置连接

在使用数据来源名称 (DSN) 连接到数据存储时，请配置 odbc.ini 文件以定义数据来源名称 (DSN)。在 odbc.ini 文件中设置属性以创建指定数据存储的连接信息的 DSN。

在 Linux 操作系统上，使用以下格式：

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file
Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

以下示例显示了在 Linux 操作系统上使用 64 位 ODBC 驱动程序配置 odbc.ini。

```
[ODBC Data Sources]
Amazon_Redshift_x64=Amazon Redshift ODBC Driver (x64)
```

```
[Amazon_Redshift_x64]
Driver=/opt/amazon/redshiftodbcx64/librsodbc64.so
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932Database=dev
locale=en-US
```

在 Linux 上配置没有 DSN 的连接

要通过不带 DSN 的连接来连接到数据存储，请在 odbcinst.ini 文件中定义驱动程序。然后，在应用程序中提供一个无 DSN 的连接字符串。

在 Linux 操作系统上，使用以下格式：

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/driver_file

...
```

以下示例显示了在 Linux 操作系统上使用 64 位 ODBC 驱动程序配置 odbcinst.ini。

```
[ODBC Drivers]
Amazon Redshift ODBC Driver (x64)=Installed

[Amazon Redshift ODBC Driver (x64)]
Description=Amazon Redshift ODBC Driver (64-bit)
Driver=/opt/amazon/redshiftodbcx64/librsodbc64.so
```

在 Apple macOS 上使用 Amazon Redshift ODBC 驱动程序

您必须在可访问 Amazon Redshift 数据仓库的客户端计算机上安装 Amazon Redshift ODBC 驱动程序。对于您要安装该驱动程序的每台计算机，有以下最低要求：

- 计算机上的根访问权限。

- Apple macOS 系统要求：
  - 需要 64 位版本的 Apple macOS 版本 11.7 或更高版本（例如，Apple macOS Big Sur、Monterey、Ventura 或更高版本）。Redshift ODBC 驱动程序仅支持 64 位客户端应用程序。
  - 150MB 可用磁盘空间。
  - 该驱动程序支持使用 iODBC 3.52.9+ 或 unixODBC 2.3.7+ 构建的应用程序。

下载并安装 Amazon Redshift ODBC 驱动程序

使用以下过程在 Apple macOS 上下载并安装 Amazon Redshift ODBC 驱动程序。仅在您当前运行的第三方应用程序获得了使用 Amazon Redshift 的认证并且需要不同驱动程序时，才能使用该特定驱动程序。

要下载并安装 ODBC 驱动程序，请执行以下操作：

1. 下载以下驱动程序：64 位 ODBC 驱动程序版本 2.2.4.0

   x86_64 和 arm64 架构都支持该驱动程序。此驱动程序的名称为 Amazon Redshift ODBC 驱动程序 (x64)。
2. 审核 Amazon Redshift ODBC 驱动程序版本 2.x 许可证。
3. 双击 .pkg 文件，然后按照向导中的步骤操作来安装该驱动程序。或者，运行以下命令：

   ```
   sudo installer -pkg PKGFileName -target /
   ```

   将 PKGFileName 替换为 pkg 包文件名。例如，以下命令将演示如何安装 64 位驱动程序：

   ```
   sudo installer -pkg ./AmazonRedshiftODBC-64-bit.X.X.XX.X.universal.pkg -target /
   ```

使用 ODBC 驱动程序管理器配置 ODBC 驱动程序

在 Mac 上，您可以使用 ODBC 驱动程序管理器来配置 ODBC 连接设置。ODBC 驱动程序管理器使用配置文件来定义和配置 ODBC 数据来源和驱动程序。您可以使用的 ODBC 驱动程序管理器取决于您使用的操作系统。

使用 iODBC 或 unixODBC 驱动程序管理器配置 ODBC 驱动程序

要配置 Amazon Redshift ODBC 驱动程序，需要以下文件：

- amazon.redshiftodbc.ini
- odbc.ini
- odbcinst.ini

如果您将驱动程序安装在默认位置，则 amazon.redshiftodbc.ini 配置文件将位于 /opt/amazon/redshiftodbcx64 中。

此外，在 /opt/amazon/redshiftodbcx64 下，您可以找到示例 odbc.ini 和 odbcinst.ini 文件。您可以使用这些文件作为配置 Amazon Redshift ODBC 驱动程序和数据来源名称 (DSN) 的示例。安装目录中的示例文件仅用作示例。

我们不建议使用 Amazon Redshift ODBC 驱动程序安装目录来存储配置文件。如果您日后重新安装 Amazon Redshift ODBC 驱动程序，或将其升级到新版本，安装目录会被覆盖。您将丢失对安装目录中的文件所做的所有更改。

为了避免出现这种情况，请将 odbc.ini、odbcinst.ini 和 amazon.redshiftodbc.ini 文件复制到安装目录以外的其他目录中。如果您要将这些文件复制到用户的主目录，请在这些文件名的开头添加一个句点（.），使其成为隐藏文件。

修改文件以添加 DSN 配置信息。在创建新文件时，还需设置环境变量，以指定这些配置文件的目标存储位置。

以下示例演示如何设置环境变量：

```
export ODBCINI=/Library/ODBC/odbc.ini
export ODBCSYSINI=/Library/ODBC
export ODBCINSTINI=${ODBCSYSINI}/odbcinst.ini
```

对于命令行应用程序：将导出命令添加到 Shell 启动文件（例如，~/.bash_profile 或 ~/.zshrc）中。

有关支持的驱动程序管理器版本，请参阅此处

在 Apple macOS 上使用数据来源名称（DSN）配置连接

在使用数据来源名称 (DSN) 连接到数据存储时，请配置 odbc.ini 文件以定义数据来源名称 (DSN)。在 odbc.ini 文件中设置属性以创建一个 DSN，来指定 Redshift 数据仓库的连接信息。

在 Apple macOS 上，使用以下格式：

```
[ODBC Data Sources]
```

```
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file
Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

以下示例展示了在 Apple macOS 上使用 64 位 ODBC 驱动程序配置 odbc.ini。

```
[ODBC Data Sources]
Amazon_Redshift_x64=Amazon Redshift ODBC Driver (x64)

[Amazon_Redshift_x64]
Driver=/opt/amazon/redshiftodbcx64/librsodbc64.dylib
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US
```

在 Apple macOS 上配置没有 DSN 的连接

要通过不带 DSN 的连接来连接到 Redshift 数据仓库，请在 odbcinst.ini 文件中定义驱动程序。然后，在应用程序中提供一个无 DSN 的连接字符串。

在 Apple macOS 上，使用以下格式：

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/driver_file

...
```

以下示例展示了在 Apple macOS 上使用 64 位 ODBC 驱动程序配置 odbcinst.ini。

```
[ODBC Drivers]
Amazon Redshift ODBC Driver (x64)=Installed
```

```
[Amazon Redshift ODBC Driver (x64)]
Description=Amazon Redshift ODBC Driver (64-bit)
Driver=/opt/amazon/redshiftodbcx64/librsodbc64.dylib
```

身份验证方法

为了防止数据遭到未经授权的访问，Amazon Redshift 数据存储要求所有连接都使用用户凭据进行身份验证。

下表说明了可用于连接到 Amazon Redshift ODBC 驱动程序版本 2.x 的每种身份验证方法的必需和可选连接选项：

| 身份验证方法 | 必填 | 可选 |
| --- | --- | --- |
| 标准 | • 主机<br>• 端口<br>• 数据库<br>• UID<br>• 密码 | |
| IAM 配置文件 | • 主机<br>• 端口<br>• 数据库<br>• IAM<br>• 配置文件 | • ClusterID<br>• 区域<br>• AutoCreate<br>• EndpointURL<br>• StsEndpointURL<br>• InstanceProfile<br><br>**Note**<br>如果没有单独设置 ClusterID 和区域，则必须在主机中设置它们。 |
| IAM 凭证 | • 主机 | • ClusterID |

| 身份验证方法 | 必填 | 可选 |
| --- | --- | --- |
| | • 端口<br>• 数据库<br>• IAM<br>• AccessKeyID<br>• SecretAccessKey | • 区域<br>• AutoCreate<br>• EndpointURL<br>• StsEndpointURL<br>• SessionToken<br>• UID<br><br>**Note**<br>如果没有单独设置 ClusterID 和区域，则必须在主机中设置它们。 |
| AD FS | • 主机<br>• 端口<br>• 数据库<br>• IAM<br>• plugin_name<br>• UID<br>• 密码<br>• IdP_Host<br>• IdP_Port | • ClusterID<br>• 区域<br>• AutoCreate<br>• EndpointUrl<br>• StsEndpointUrl<br>• Preferred_Role<br>• loginToRp<br>• SSL_Insecure<br><br>**Note**<br>如果没有单独设置 ClusterID 和区域，则必须在主机中设置它们。 |

| 身份验证方法 | 必填 | 可选 |
| --- | --- | --- |
| Azure AD | • 主机<br>• 端口<br>• 数据库<br>• IAM<br>• plugin_name<br>• UID<br>• 密码<br>• IdP_Tenant<br>• Client_ID<br>• Client_Secret | • ClusterID<br>• 区域<br>• AutoCreate<br>• EndpointUrl<br>• StsEndpointUrl<br>• Preferred_Role<br>• dbgroups_filter<br><br>**Note**<br>如果没有单独设置 ClusterID 和区域，则必须在主机中设置它们。 |
| JWT | • 主机<br>• 端口<br>• 数据库<br>• IAM<br>• plugin_name<br>• web_identity_token | • provider_name |

| 身份验证方法 | 必填 | 可选 |
| --- | --- | --- |
| Okta | • 主机<br>• 端口<br>• 数据库<br>• IAM<br>• plugin_name<br>• UID<br>• 密码<br>• IdP_Host<br>• App_Name<br>• App_ID | • ClusterID<br>• 区域<br>• AutoCreate<br>• EndpointUrl<br>• StsEndpointUrl<br>• Preferred_Role<br><br>**Note**<br>如果没有单独设置 ClusterID 和区域，则必须在主机中设置它们。 |
| Ping Federate | • 主机<br>• 端口<br>• 数据库<br>• IAM<br>• plugin_name<br>• UID<br>• 密码<br>• IdP_Host<br>• IdP_Port | • ClusterID<br>• 区域<br>• AutoCreate<br>• EndpointUrl<br>• StsEndpointUrl<br>• Preferred_Role<br>• SSL_Insecure<br>• partner_spid<br><br>**Note**<br>如果没有单独设置 ClusterID 和区域，则必须在主机中设置它们。 |

| 身份验证方法 | 必填 | 可选 |
| --- | --- | --- |
| 浏览器 Azure AD | • 主机<br>• 端口<br>• 数据库<br>• IAM<br>• plugin_name<br>• IdP_Tenant<br>• Client_ID<br>• UID | • ClusterID<br>• 区域<br>• AutoCreate<br>• EndpointUrl<br>• StsEndpointUrl<br>• Preferred_Role<br>• dbgroups_filter<br>• IdP_Response_Timeout<br>• listen_port<br><br>**Note**<br>如果没有单独设置 ClusterID 和区域，则必须在主机中设置它们。 |

| 身份验证方法 | 必填 | 可选 |
| --- | --- | --- |
| 浏览器 SAML | • 主机<br>• 端口<br>• 数据库<br>• IAM<br>• plugin_name<br>• login_url<br>• UID | • ClusterID<br>• 区域<br>• AutoCreate<br>• EndpointUrl<br>• StsEndpointUrl<br>• Preferred_Role<br>• dbgroups_filter<br>• IdP_Response_Timeout<br>• listen_port<br><br>**Note**<br>如果没有单独设置 ClusterID 和区域，则必须在主机中设置它们。 |
| 身份验证配置文件 | • 主机<br>• 端口<br>• 数据库<br>• AccessKeyID<br>• SecretAccessKey | |

| 身份验证方法 | 必填 | 可选 |
| --- | --- | --- |
| 浏览器 Azure AD OAUTH2 | • 主机<br>• 端口<br>• 数据库<br>• IAM<br>• plugin_name<br>• IdP_Tenant<br>• Client_ID<br>• UID | • ClusterID<br>• 区域<br>• EndpointUrl<br>• IdP_Response_Timeout<br>• listen_port<br>• 范围<br>• provider_name<br><br>**Note**<br>如果没有单独设置 ClusterID 和区域，则必须在主机中设置它们。 |
| AWS IAM Identity Center | • 主机<br>• 数据库<br>• plugin_name<br>• idc_region<br>• issuer_url | • idc_client_display_name<br>• idp_response_timeout<br>• listen_port |

使用外部凭证服务

除了对 AD FS、Azure AD 和 Okta 的内置支持外，Amazon Redshift ODBC 驱动程序的 Windows 版本还提供了对其他凭证服务的支持。该驱动程序可以使用您选择的任何基于 SAML 的凭证提供程序插件对连接进行身份验证。

要在 Windows 上配置外部凭证服务，请执行以下操作：

1. 创建一个 IAM 配置文件，根据需要指定凭证提供程序插件和其他身份验证参数。配置文件必须采用 ASCII 编码，并且必须包含以下键值对，其中 `PluginPath` 是插件应用程序的完整路径：

```
plugin_name = PluginPath
```

例如：

```
plugin_name = C:\Users\kjson\myapp\CredServiceApp.exe
```

有关如何创建配置文件的信息，请参阅《Amazon Redshift 集群管理指南》中的使用配置文件。

2. 将驱动程序配置为使用此配置文件。驱动程序将检测并使用配置文件中指定的身份验证设置。

### 数据类型转换

Amazon Redshift ODBC 驱动程序版本 2.x 支持许多常见的数据格式，从而可在 Amazon Redshift 和 SQL 数据类型之间进行转换。

下表列出了支持的数据类型映射。

| Amazon Redshift 类型 | SQL 类型 |
|---|---|
| BIGINT | SQL_BIGINT |
| BOOLEAN | SQL_BIT |
| CHAR | SQL_CHAR |
| DATE | SQL_TYPE_DATE |
| DECIMAL | SQL_NUMERIC |
| DOUBLE PRECISION | SQL_DOUBLE |
| GEOGRAPHY | SQL_ LONGVARBINARY |
| GEOMETRY | SQL_ LONGVARBINARY |
| INTEGER | SQL_INTEGER |
| REAL | SQL_REAL |
| SMALLINT | SQL_SMALLINT |

| Amazon Redshift 类型 | SQL 类型 |
| --- | --- |
| SUPER | SQL_LONGVARCHAR |
| TEXT | SQL_LONGVARCHAR |
| TIME | SQL_TYPE_TIME |
| TIMETZ | SQL_TYPE_TIME |
| TIMESTAMP | SQL_TYPE_ TIMESTAMP |
| TIMESTAMPTZ | SQL_TYPE_ TIMESTAMP |
| VARBYTE | SQL_LONGVARBINARY |
| VARCHAR | SQL_VARCHAR |

ODBC 驱动程序选项

可以使用驱动程序配置选项来控制 Amazon Redshift ODBC 驱动程序的行为。驱动程序选项不区分大小写。

在 Microsoft Windows 中，您通常可以在配置数据来源名称 (DSN) 时设置驱动程序选项。您还能在以编程方式连接时，或者通过在 HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI*your_DSN* 中添加或更改注册表项来设置驱动程序选项。

在 Linux 中，可以在 odbc.ini 和 amazon.redshiftodbc.ini 文件中设置驱动程序配置选项。在 amazon.redshiftodbc.ini 文件中设置的配置选项适用于所有连接。相反，odbc.ini 文件中的设置配置选项特定于一个连接。在 odbc.ini 中设置的配置选项优先于在 amazon.redshiftodbc.ini 中设置的配置选项。

以下是您可以为 Amazon Redshift ODBC 版本 2.x 驱动程序指定的选项的说明：

主题


- AccessKeyID
- app_id
- ApplicationName
- app_name

- AuthProfile
- AuthType
- AutoCreate
- BoolsAsChar
- CaFile
- client_id
- client_ secret
- ClusterId
- 压缩
- 数据库
- DatabaseMetadataCurrentDbOnly
- dbgroups_filter
- 驱动程序
- DSN
- EnableTableTypes
- EndpointUrl
- 取回
- ForceLowercase
- group_federation
- https_proxy_host
- https_proxy_password
- https_proxy_port
- https_proxy_username
- IAM
- idc_client_display_name
- idc_region
- idp_host
- idp_port
- idp_response_timeout

- idp_tenant
- idp_partition
- idp_use_https_proxy
- InstanceProfile
- issuer_url
- KeepAlive
- KeepAliveCount
- KeepAliveInterval
- KeepAliveIdle
- listen_port
- login_url
- loginToRp
- LogLevel
- LogPath
- LoginTimeout
- MaxLongVarcharSize
- MaxVarcharSize
- Min_TLS
- partner_spid
- 密码 | PWD
- plugin_name
- 端口 | PortNumber
- Prefer_PQ
- preferred_role
- 配置文件
- provider_name
- ProxyHost
- ProxyPort
- ProxyPwd

- ProxyUid
- ReadOnly
- region
- SecretAccessKey
- SessionToken
- 服务器 | HostName | 主机
- ssl_insecure
- SSLMode
- StreamingCursorRows
- StsConnectionTimeout
- StsEndpointUrl
- 令牌
- token_type
- UID | 用户 | LogonID
- UseDeclareFetch
- UseUnicode
- web_identity_token

AccessKeyID

- 默认值 – 无
- 数据类型 – 字符串

用户或角色的 IAM 访问密钥。如果您设置此参数，则还必须指定 SecretAccessKey。

此参数为可选的。

app_id

- 默认值 – 无
- 数据类型 – 字符串

Okta 提供的与您的 Amazon Redshift 应用程序关联的唯一 ID。

此参数为可选的。

ApplicationName

- 默认值 – 无
- 数据类型 – 字符串

传递给 Amazon Redshift 以供审计的客户端应用程序的名称。您提供的应用程序名称显示在 SYS_CONNECTION_LOG 表的“application_name”列中。这有助于在调试问题时跟踪连接源并对其进行故障排除。

此参数为可选的。

app_name

- 默认值 – 无
- 数据类型 – 字符串

您用于验证与 Amazon Redshift 的连接的 Okta 应用程序的名称。

此参数为可选的。

AuthProfile

- 默认值 – 无
- 数据类型 – 字符串

用于管理连接设置的身份验证配置文件。如果您设置此参数，则还必须设置 AccessKeyID 和 SecretAccessKey。

此参数为可选的。

AuthType

- 默认值：标准
- 数据类型 – 字符串

此选项指定在您使用“Amazon Redshift ODBC 驱动程序 DSN 设置”对话框配置 DSN 时驱动程序使用的身份验证模式：

- 标准：使用您的 Amazon Redshift 用户名和密码的标准身份验证。
- AWS 配置文件：使用配置文件的 IAM 身份验证。
- AWS IAM 凭证：使用 IAM 凭证的 IAM 身份验证。
- 身份提供者：AD FS：使用 Active Directory 联合身份验证服务（AD FS）的 IAM 身份验证。
- 身份提供者：身份验证插件：一种授权插件，可接受 AWS IAM Identity Center 令牌，或基于 OpenID Connect（OIDC）JSON 的身份令牌（JWT），该令牌由与 AWS IAM Identity Center 关联的任意 Web 身份提供者提供。
- 身份提供者：Azure AD：使用 Azure AD 门户的 IAM 身份验证。
- 身份提供者：JWT：使用 JSON Web 令牌（JWT）的 IAM 身份验证。
- 身份提供者：Okta：使用 Okta 的 IAM 身份验证。
- 身份提供者：PingFederate：使用 PingFederate 的 IAM 身份验证。

仅当您使用 Windows 驱动程序中的“Amazon Redshift ODBC 驱动程序 DSN 设置”对话框配置 DSN 时，此选项才可用。当您使用连接字符串或非 Windows 计算机配置连接时，驱动程序将根据您指定的凭证，自动确定是使用“标准”、“AWS 配置文件”还是“AWS IAM 凭证”身份验证。要使用身份提供者，您必须设置 plugin_name 属性。

此参数为必需参数。

AutoCreate

- 默认值 - 0
- 数据类型 – Boolean

一个布尔值，用于指定当指定的用户不存在时驱动程序是否创建新用户。

- 1 | TRUE：如果通过 UID 指定的用户不存在，则驱动程序将创建新用户。
- 0 | FALSE：驱动程序不会创建新用户。如果指定的用户不存在，则身份验证将失败。

此参数为可选的。

BoolsAsChar

- 默认值 - 0
- 数据类型 – Boolean

指定驱动程序是否将 BOOLEAN 列报告为字符类型而不是位类型。

- 1 | TRUE：驱动程序将 BOOLEAN 列报告为 SQL_VARCHAR，列大小为 1。驱动程序将 BOOLEAN 值作为字符串“0”和“1”返回。
- 0 | FALSE：驱动程序将 BOOLEAN 列报告为 SQL_BIT。这是默认行为。

如果您的应用程序依赖于作为字符数据返回的 BOOLEAN 列，请使用此选项。

此参数为可选的。此选项在驱动程序版本 2.2.1 及更高版本中可用。

CaFile

- 默认值 – 无
- 数据类型 – 字符串

用于某些形式的 IAM 身份验证的 CA 证书文件的文件路径。

此参数仅在 Linux 上可用。

此参数为可选的。

client_id

- 默认值 – 无
- 数据类型 – 字符串

与 Azure AD 中的 Amazon Redshift 应用程序关联的客户端 ID。

如果通过 Azure AD 服务进行身份验证，则此参数是必需的。

client_ secret

- 默认值 – 无
- 数据类型 – 字符串

与 Azure AD 中的 Amazon Redshift 应用程序关联的秘密密钥。

如果通过 Azure AD 服务进行身份验证，则此参数是必需的。

ClusterId

- 默认值 – 无
- 数据类型 – 字符串

您要连接到的 Amazon Redshift 集群的名称。它将在 IAM 身份验证中使用。集群 ID 不会在服务器参数中指定。

此参数为可选的。

压缩

- 默认值 – 关闭
- 数据类型 – 字符串

用于 Amazon Redshift 服务器与客户端或驱动程序之间线路协议通信的压缩方法。

可以指定以下值：

- lz4：将用于与 Amazon Redshift 进行线路协议通信的压缩方法设置为 `lz4`。
- zstd：将用于与 Amazon Redshift 进行线路协议通信的压缩方法设置为 `zstd`。
- off：与 Amazon Redshift 进行线路协议通信时不使用压缩方法。

此参数为可选的。

数据库

- 默认值 – 无
- 数据类型 – 字符串

您要访问的 Amazon Redshift 数据库的名称。

此参数为必需参数。

DatabaseMetadataCurrentDbOnly

- 默认值：1
- 数据类型 – Boolean

一个布尔值，用于指定驱动程序是否从多个数据库和集群返回元数据。

- 1 | TRUE：驱动程序仅从当前数据库返回元数据。
- 0 | FALSE。驱动程序将跨多个 Amazon Redshift 数据库和集群返回元数据。

此参数为可选的。

dbgroups_filter

- 默认值 – 无
- 数据类型 – 字符串

在使用 Azure、浏览器 Azure 和浏览器 SAML 身份验证类型时，您可以指定的正则表达式，用于筛选出从 Amazon Redshift 的 SAML 响应中收到的数据库组 (DbGroup)。

此参数为可选的。

驱动程序

- 默认值：Amazon Redshift ODBC 驱动程序 (x64)
- 数据类型 – 字符串

驱动程序的名称。唯一受支持的值是 Amazon Redshift ODBC 驱动程序(x64)。

如果您未设置 DSN，则此参数是必需的。

DSN

- 默认值 – 无
- 数据类型 – 字符串

驱动程序数据来源名称的名称。应用程序将在 SQLDriverConnect API 中指定 DSN。

如果您未设置驱动程序，则此参数是必需的。

EnableTableTypes

- 默认值：1
- 数据类型 – Boolean

指定当应用程序调用 SQLTables 目录函数时，驱动程序是否识别数据来源中的详细表类型信息。默认情况下，驱动程序可以识别详细表类型。

- 1 | TRUE：驱动程序可识别以下表类型：TABLE、VIEW、SYSTEM TABLE、SYSTEM VIEW、EXTERNAL TABLE 和 LOCAL TEMPORARY。
- 0 | FALSE：驱动程序将详细表类型信息标准化为泛型 TABLE 和 VIEW 表类型。

此参数为可选的。它在驱动程序版本 2.2.1 和更高版本中可用。

EndpointUrl

- 默认值 – 无
- 数据类型 – 字符串

用于与 Amazon Redshift Coral 服务通信以进行 IAM 身份验证的覆盖端点。

此参数为可选的。

取回

- 默认值 – 100
- 数据类型 – 整数

Declare/Fetch 模式启用后，此选项可控制驱动程序每次往返检索的行数，让您可以根据工作负载调整内存使用量和网络开销之间的平衡。值越大，一次读取的行数越多，可减少往返服务器的次数，代价是每次读取会占用更多内存。值越小，内存使用量越少，但会增加驱动程序返回服务器的频率。它设置在 `UseDeclareFetch` 启用时驱动程序单次返回的行数。有关更多信息，请参阅 UseDeclareFetch。

此参数为可选的。

ForceLowercase

- 默认值 - 0
- 数据类型 – Boolean

一个布尔值，用于指定在使用单点登录身份验证时，驱动程序是否会将从身份提供者发送到 Amazon Redshift 的所有数据库组 (DbGroup) 小写。

- 1 | TRUE：驱动程序会将从身份提供者发送的所有 DbGroup 转换为小写。
- 0 | FALSE：驱动程序不会更改 DbGroup。

此参数为可选的。

group_federation

- 默认值 - 0
- 数据类型 – Boolean

一个布尔值，指定 `getClusterCredentialsWithIAM` API 是否用于在预调配集群中获取临时集群凭证。此选项允许 IAM 用户在预调配集群中与 Redshift 数据库角色进行集成。请注意，此选项不适用于 Redshift Serverless 命名空间。

- 1 | TRUE：驱动程序使用 `getClusterCredentialsWithIAM` API 来获取预调配集群中的临时集群凭证。
- 0 | FALSE：驱动程序使用默认 `getClusterCredentials` API 来获取预调配集群中的临时集群凭证。

此参数为可选的。

https_proxy_host

- 默认值 – 无
- 数据类型 – 字符串

您要通过其来传递 IAM 身份验证过程的代理服务器的主机名或 IP 地址。

此参数为可选的。

https_proxy_password

- 默认值 – 无
- 数据类型 – 字符串

用于访问代理服务器的密码。它将用于 IAM 身份验证。

此参数为可选的。

https_proxy_port

- 默认值 – 无
- 数据类型 – 整数

代理服务器用于侦听客户端连接的端口号。它将用于 IAM 身份验证。

此参数为可选的。

https_proxy_username

- 默认值 – 无
- 数据类型 – 字符串

用于访问代理服务器的用户名。它用于 IAM 身份验证。

此参数为可选的。

IAM

- 默认值 - 0
- 数据类型 – Boolean

一个布尔值，用于指定驱动程序是否使用 IAM 身份验证方法对连接进行身份验证。

- 1 | TRUE：驱动程序将使用某种 IAM 身份验证方法（使用访问密钥和秘密密钥对、配置文件或凭证服务）。
- 0 | FALSE。驱动程序将使用标准身份验证（使用您的数据库用户名和密码）。

此参数为可选的。

idc_client_display_name

- 默认值 – Amazon Redshift ODBC 驱动程序
- 数据类型 – 字符串

要用于使用 BrowserIdcAuthPlugin 的客户端的显示名称。

此参数为可选的。

idc_region

- 默认值 – 无
- 数据类型 – 字符串

AWS IAM Identity Center 实例所在的 AWS 区域。

只有在 plugin_name 配置选项中使用 `BrowserIdcAuthPlugin` 进行身份验证时，才需要此参数。

idp_host

- 默认值 – 无
- 数据类型 – 字符串

您用于对 Amazon Redshift 进行身份验证的 IdP（身份提供者）主机。

此参数为可选的。

idp_port

- 默认值 – 无
- 数据类型 – 整数

您用于对 Amazon Redshift 进行身份验证的 IdP（身份提供者）的端口。根据您在创建、修改或迁移集群时选择的端口，允许访问所选端口。

此参数为可选的。

idp_response_timeout

- 默认值 – 120
- 数据类型 – 整数

当通过浏览器插件使用 SAML 或 Azure AD 服务时，驱动程序等待身份提供者发出 SAML 响应的秒数。

此参数为可选的。

idp_tenant

- 默认值 – 无
- 数据类型 – 字符串

与您的 Amazon Redshift 应用程序关联的 Azure AD 租户 ID。

如果通过 Azure AD 服务进行身份验证，则此参数是必需的。

idp_partition

- 默认值 – 无
- 数据类型 – 字符串

指定在其中配置身份提供者（IdP）的云分区。这决定了驱动程序连接到哪个 IdP 身份验证端点。

如果将此参数留空，则驱动程序默认为商业分区。可能的值有：

- us-gov：如果在 Azure Government 中配置了 IdP，请使用此值。例如，Azure AD Government 使用端点 `login.microsoftonline.us`。
- cn：如果在中国云分区中配置了 IdP，请使用此值。例如，Azure AD China 使用 `login.chinacloudapi.cn` 端点。

此参数为可选的。

idp_use_https_proxy

- 默认值 - 0
- 数据类型 – Boolean

一个布尔值，用于指定驱动程序是否通过代理服务器传递身份提供者（IdP）的身份验证过程。

- 1 | TRUE：驱动程序将通过代理服务器传递 IdP 身份验证过程。
- 0 | FALSE。驱动程序不会通过代理服务器传递 IdP 身份验证过程。

此参数为可选的。

InstanceProfile

- 默认值 - 0
- 数据类型 – Boolean

一个布尔值，用于指定驱动程序在配置为使用配置文件进行身份验证时是否使用 Amazon EC2 实例配置文件。

- 1 | TRUE：驱动程序将使用 Amazon EC2 实例配置文件。
- 0 | FALSE。驱动程序将改用通过“配置文件名称”选项（配置文件）指定的串联角色配置文件。

此参数为可选的。

issuer_url

- 默认值 – 无
- 数据类型 – 字符串

指向 AWS IAM Identity Center 服务器的实例端点。

只有在 plugin_name 配置选项中使用 `BrowserIdcAuthPlugin` 进行身份验证时，才需要此参数。

KeepAlive

- 默认值：1
- 数据类型 – Boolean

一个布尔值，用于指定驱动程序是否使用 TCP keepalive 来防止连接超时。

- 1 | TRUE：驱动程序将使用 TCP keepalive 来防止连接超时。
- 0 | FALSE。驱动程序不会使用 TCP keepalive。

此参数为可选的。

KeepAliveCount

- 默认值 - 0

- 数据类型 – 整数

连接被视为断开前可能丢失的 TCP keepalive 包的数量。当此参数设置为 0 时，驱动程序将使用此设置的系统默认值。

此参数为可选的。

KeepAliveInterval

- 默认值 - 0
- 数据类型 – 整数

两次传输 TCP keepalive 间隔的秒数。当此参数设置为 0 时，驱动程序将使用此设置的系统默认值。

此参数为可选的。

KeepAliveIdle

- 默认值 - 0
- 数据类型 – 整数

驱动程序发送 TCP Keepalive 包前处于不活动状态的秒数。当此参数设置为 0 时，驱动程序将使用此设置的系统默认值。

此参数为可选的。

listen_port

- 默认值：7890
- 数据类型 – 整数

通过浏览器插件使用 SAML、Azure AD 或 AWS IAM Identity Center 服务时，驱动程序用于接收来自身份提供者的 SAML 响应的端口。

此参数为可选的。

login_url

- 默认值 – 无

- 数据类型 – 字符串

在使用通用浏览器 SAML 插件时，身份提供者网站上的资源的 URL。

如果通过浏览器插件使用 SAML 或 Azure AD 服务进行身份验证，则此参数是必需的。

loginToRp

- 默认值：urn:amazon:webservices
- 数据类型 – 字符串

要用于 AD FS 身份验证类型的信赖方信任。

此字符串为可选项。

LogLevel

- 默认值 - 0
- 数据类型 – 整数

> **Note**
>
> `LogLevel` 和 `LogPath` 是全局设置。驱动程序按以下优先顺序读取它们：
>
> 1. 连接字符串 (`LogLevel=6;LogPath=/path;`)：最高优先级，在连接时覆盖所有其它设置。
> 2. 在 Windows 上：“DSN 设置”对话框（“高级”选项卡）。
>
>    在 Linux 和 macOS 上：`odbc.ini` 文件中的 `[ODBC]` 部分。
> 3. `amazon.redshiftodbc.ini` 文件中的 **`[DRIVER]`** 部分：如果未从上述来源启用日志记录，则用作后备选项。
>
> 在单个 DSN 部分（例如 `[MyDSN]`）中设置 `LogLevel` 或 `LogPath` 将没有效果。这些选项不是从每个 DSN 部分中读取的。

使用此属性可以启用或禁用驱动程序中的日志记录，并指定包含在日志中的详细信息量。我们建议您启用日志记录的时长仅足以捕获问题即可，因为日志记录会降低性能，并会占用大量磁盘空间。

将该属性设置为以下值之一：

- 0：关闭。禁用所有日志记录。
- 1：严重。保留用于严重错误。
- 2：错误。记录错误事件，例如连接失败和身份验证错误。
- 3：警告。记录警告和意外情况以及错误。
- 4：信息。记录一般进度信息，例如连接状态和服务器参数。
- 5：调试。记录所有驱动程序活动，包括 ODBC API 函数调用以及参数和返回值。
- 6：跟踪。记录所有驱动程序活动，包括其它低级别协议和内部状态详细信息。

> **Important**
>
> 日志级别 5（调试）或更高级别将捕获所有驱动程序活动，其中可能包括诸如连接凭证和查询数据等敏感信息。仅临时启用这些日志级别以捕获问题，并确保在与支持人员或任何第三方共享日志之前对所有敏感信息进行脱敏。

启用日志记录后，驱动程序将在您在 LogPath 属性中指定的位置生成以下日志文件：

- 一个 `redshift_odbc.log.1` 文件，它将记录连接握手期间发生的驱动程序活动。
- 一个 `redshift_odbc.log` 文件，用于与数据库建立连接后的所有驱动程序活动。

此参数为可选的。

LogPath

- 默认值：特定于操作系统的 TEMP 目录
- 数据类型 – 字符串

> **Note**
>
> `LogLevel` 和 `LogPath` 是全局设置。驱动程序按以下优先顺序读取它们：
>
> 1. 连接字符串 (`LogLevel=6;LogPath=/path;`)：最高优先级，在连接时覆盖所有其它设置。
> 2. 在 Windows 上："DSN 设置"对话框（"高级"选项卡）。

在 Linux 和 macOS 上：odbc.ini 文件中的 [ODBC] 部分。

3. amazon.redshiftodbc.ini 文件中的 **[DRIVER]** 部分：如果未从上述来源启用日志记录，则用作后备选项。

在单个 DSN 部分（例如 [MyDSN]）中设置 LogLevel 或 LogPath 将没有效果。这些选项不是从每个 DSN 部分中读取的。

当 LogLevel 大于 0 时，驱动程序保存日志文件的文件夹的完整路径。

此参数为可选的。

LoginTimeout

- 默认值 - 0
- 数据类型 – 整数

在超时之前，驱动程序等待与服务器建立连接的秒数。值为 0 表示没有超时（驱动程序无限期等待）。

可以在连接字符串（使用 LoginTimeout 或短别名 LT）、DSN 配置中或使用 SQLSetConnectAttr 及 SQL_ATTR_LOGIN_TIMEOUT 以编程方式来设置此选项。在 Windows 上，此选项也可在“DSN 设置”对话框（“高级”选项卡）中找到。

此参数为可选的。

MaxLongVarcharSize

- 默认值：65535
- 数据类型 – 整数

对于被 MaxVarcharSize 重新分类为大文本 (SQL_LONGVARCHAR) 的列，驱动程序报告的列大小。此值替换了元数据响应中该列的实际声明大小。

建议使用默认值 65535，因为某些中间件层（例如 MSDASQL）会拒绝其报告大小介于 8001 到 32768 字节之间的大文本列。通过报告 65535，驱动程序可确保接受重新分类的列而不会出现错误。大多数用户不需要更改此值。

当 MaxVarcharSize 设置为 0（禁用重新分类）时，此参数不起作用。

此参数为可选的。它在驱动程序版本 2.1.16 和更高版本中可用。

MaxVarcharSize

- 默认值 – 255
- 数据类型 – 整数

驱动程序报告为常规文本列的最大列大小 (SQL_VARCHAR)。驱动程序会将任何大于此值的 CHAR 和 VARCHAR 列重新分类为大文本（SQL_LONGVARCHAR，或启用 UseUnicode 时为 SQL_WLONGVARCHAR）。

这种重新分类可以防止应用程序或中间件层出现无法处理超过一定大小的常规文本列的错误。例如，通过 MSDASQL（OLE DB）访问 Amazon Redshift 的 SQL Server 链接服务器无法处理大于约 8000 字节的 SQL_VARCHAR 列，因此，将超过大小的列重新分类为大文本可以避免转换失败。

- 255（默认）：大于 255 个字符的列报告为 SQL_LONGVARCHAR。
- 0：禁用重新分类。无论大小如何，所有列都报告为 SQL_VARCHAR。
- 自定义值：任意正整数。大于指定值的列报告为 SQL_LONGVARCHAR。

对某列进行重新分类时，驱动程序会使用在 MaxLongVarcharSize 中指定的值报告其大小。

此参数为可选的。它在驱动程序版本 2.1.16 和更高版本中可用。

Min_TLS

- 默认值 – 1.2
- 数据类型 – 字符串

驱动程序允许数据存储以用于对连接进行加密的 TLS/SSL 的最低版本。例如，如果指定了 TLS 1.2，则无法使用 TLS 1.1 对连接进行加密。

Min_TLS 接受以下值：

- 1.0：连接必须至少使用 TLS 1.0。
- 1.1：连接必须至少使用 TLS 1.1。
- 1.2：连接必须至少使用 TLS 1.2。

此参数为可选的。

partner_spid

- 默认值 – 无
- 数据类型 – 字符串

在使用 PingFederate 服务验证连接时使用的合作伙伴 SPID（服务提供商 ID）值。

此参数为可选的。

密码 | PWD

- 默认值 – 无
- 数据类型 – 字符串

与您在“用户”字段（UID | 用户| LogonID）中提供的数据库用户名相对应的密码。

此参数为可选的。

plugin_name

- 默认值 – 无
- 数据类型 – 字符串

要用于身份验证的凭证提供程序插件名称。

支持下列值：

- ADFS：使用 Active Directory 联合身份验证服务进行身份验证。
- AzureAD：使用 Microsoft Azure Active Directory (AD) 服务进行身份验证。
- BrowserAzureAD：使用适用于 Microsoft Azure Active Directory (AD) 服务的浏览器插件进行身份验证。
- BrowserIdcAuthPlugin ：使用 AWS IAM Identity Center 的授权插件。
- BrowserSAML：使用适用于 SAML 服务（如 Okta 或 Ping）的浏览器插件进行身份验证。
- IdpTokenAuthPlugin：一种授权插件，可接受 AWS IAM Identity Center 令牌，或者基于 OpenID Connect（OIDC）JSON 的身份令牌（JWT），该令牌由与 AWS IAM Identity Center 关联的任意 Web 身份提供者提供。

- JWT：使用 JSON Web 令牌 (JWT) 进行身份验证。
- Ping：使用 PingFederate 服务进行身份验证。
- Okta：使用 Okta 服务进行身份验证。

此参数为可选的。

端口 | PortNumber

- 默认值：5439
- 数据类型 – 整数

Amazon Redshift 服务器用于侦听客户端连接的 TCP 端口号。

此参数为可选的。

Prefer_PQ

- 默认值：1
- 数据类型 – Boolean

指定驱动程序是否在 TLS 1.3 ClientHello 中通告后量子混合密钥交换组。当提供后量子组并且服务器也支持它们时，服务器会选择一个。否则，握手会透明地回退到经典密钥交换。

- 1 | TRUE：驱动程序通告后量子混合 TLS 密钥交换组。
- 0 | FALSE：驱动程序不通告后量子组。如果网络中间盒因无法识别驱动程序提供的后量子密钥交换组而阻止 TLS 握手，则使用此设置。

此参数为可选的。它在驱动程序版本 2.2.1 和更高版本中可用。

preferred_role

- 默认值 – 无
- 数据类型 – 字符串

您希望在 Amazon Redshift 连接期间担任的角色。它将用于 IAM 身份验证。

此参数为可选的。

配置文件

- 默认值 – 无
- 数据类型 – 字符串

用于在 Amazon Redshift 中进行身份验证的用户 AWS 配置文件的名称。

- 如果“使用实例配置文件”参数（InstanceProfile 属性）设置为 1 | TRUE，则该设置将优先，驱动程序将改用 Amazon EC2 实例配置文件。
- 包含配置文件的凭证文件的默认位置为 `~/.aws/Credentials`。`AWS_SHARED_CREDENTIALS_FILE` 环境变量可用于指向其他凭证文件。

此参数为可选的。

provider_name

- 默认值 – 无
- 数据类型 – 字符串

用户使用 CREATE IDENTITY PROVIDER 查询创建的身份验证提供程序。它将用于本机 Amazon Redshift 身份验证。

此参数为可选的。

ProxyHost

- 默认值 – 无
- 数据类型 – 字符串

要通过其连接的代理服务器的主机名或 IP 地址。

此参数为可选的。

ProxyPort

- 默认值 – 无
- 数据类型 – 整数

代理服务器用于侦听客户端连接的端口号。

此参数为可选的。

ProxyPwd

- 默认值 – 无
- 数据类型 – 字符串

用于访问代理服务器的密码。

此参数为可选的。

ProxyUid

- 默认值 – 无
- 数据类型 – 字符串

用于访问代理服务器的用户名。

此参数为可选的。

ReadOnly

- 默认值 - 0
- 数据类型 – Boolean

一个布尔值，用于指定驱动程序是否处于只读模式。

- 1 | TRUE：连接处于只读模式，无法写入数据存储。
- 0 | FALSE：连接不处于只读模式，可以写入数据存储。

此参数为可选的。

region

- 默认值 – 无
- 数据类型 – 字符串

您的集群所在的 AWS 区域。

此参数为可选的。

SecretAccessKey

- 默认值 – 无
- 数据类型 – 字符串

用户或角色的 IAM 秘密密钥。如果您设置此参数，则还必须设置 AccessKeyID。

此参数为可选的。

SessionToken

- 默认值 – 无
- 数据类型 – 字符串

与您用于身份验证的 IAM 角色关联的临时 IAM 会话令牌。

此参数为可选的。

服务器 | HostName | 主机

- 默认值 – 无
- 数据类型 – 字符串

要连接到的端点服务器。

此参数为必需参数。

ssl_insecure

- 默认值 - 0
- 数据类型 – Boolean

一个布尔值，用于指定驱动程序是否检查 IdP 服务器证书的真实性。

- 1 | TRUE：驱动程序不检查 IdP 服务器证书的真实性。

- 0 | FALSE：驱动程序检查 IdP 服务器证书的真实性

此参数为可选的。

SSLMode

- 默认值 – `verify-ca`
- 数据类型 – 字符串

在连接到 Amazon Redshift 时要使用的 SSL 证书验证模式。以下是可能的值：

- `verify-full`：仅使用 SSL、受信任的证书颁发机构和与证书匹配的服务器名称进行连接。
- `verify-ca`：仅使用 SSL 和受信任的证书颁发机构进行连接。
- `require`：仅使用 SSL 进行连接。
- `prefer`：使用 SSL 进行连接（如果可用）。否则，将在不使用 SSL 的情况下进行连接。
- `allow`：默认情况下，不使用 SSL 进行连接。如果服务器需要 SSL 连接，则使用 SSL。
- `disable`：不使用 SSL 进行连接。

> **Note**
>
> 我们建议尽可能使用 `verify-full` 进行 SSL 连接。

此参数为可选的。

StreamingCursorRows

- 默认值 - 0
- 数据类型 – 整数

限制驱动程序针对仅向前游标在内存中同时保存的行数，这样大型结果集就不必同时加载至内存。

将该属性设置为以下值之一：

- 0：驱动程序先将整个结果集加载到内存中，然后再将行返回到应用程序（默认）。大型结果集可能会占用大量内存。

- 正整数：对于仅向前游标，驱动程序一次只能在内存中保留该数量的行。它会将这些行返回到应用程序，释放它们，待应用程序处理完当前批次后，再获取下一批。

此选项对可滚动游标没有影响，可滚动游标始终采用内存中行为。请注意，在 Windows 上，Amazon Redshift ODBC Driver DSN 设置对话框会将此字段的值预先填充为 100。

流式传输游标模式需要仅向前游标。如果启用了客户端游标（CSC）选项，则优先使用该选项，并禁用流式传输游标模式。

> **Note**
>
> `StreamingCursorRows` 和 `UseDeclareFetch` 均采用分批处理结果的方式，而非将整个结果集加载到内存。`UseDeclareFetch` 使用服务器端游标，并且通常可提供显著更高的吞吐量，因此当提取性能为首要考量时，优先选用该选项。`StreamingCursorRows` 不需要事务封装或服务器端游标；当这些属性比峰值速度更重要（即内存占用有限的仅向前流）时可选择它。如果 `UseDeclareFetch` 已启用，则它的优先级更高并强制将 `StreamingCursorRows` 设为 0。有关更多信息，请参阅 UseDeclareFetch。

此参数为可选的。

StsConnectionTimeout

- 默认值 - 0
- 数据类型 – 整数

IAM 连接的最长等待时间（以秒为单位）。如果设置为 0 或未指定，驱动程序将为每次 AWS STS 调用等待 60 秒。

此参数为可选的。

StsEndpointUrl

- 默认值 – 无
- 数据类型 – 字符串

此选项指定用于与 AWS Security Token Service (AWS STS) 通信的覆盖端点。

此参数为可选的。

令牌

- 默认值 – 无
- 数据类型 – 字符串

AWS IAM Identity Center 提供的访问令牌，或 OpenID Connect（OIDC）JSON Web 令牌（JWT，JSON Web Token），该令牌由与 AWS IAM Identity Center 关联的 Web 身份提供者提供。您的应用程序必须通过 AWS IAM Identity Center 或与 AWS IAM Identity Center 关联的身份提供者，对您的应用程序用户进行身份验证，从而生成此令牌。

此参数可用于 `IdpTokenAuthPlugin`。

token_type

- 默认值 – 无
- 数据类型 – 字符串

`IdpTokenAuthPlugin` 中使用的令牌类型。

可以指定以下值：

ACCESS_TOKEN

如果您使用 AWS IAM Identity Center 提供的访问令牌，请输入此项。

EXT_JWT

如果您使用 OpenID Connect（OIDC）JSON Web 令牌（JWT）（该令牌由与 AWS IAM Identity Center 集成的基于 Web 的身份提供者提供），请输入此项。

此参数可用于 `IdpTokenAuthPlugin`。

UID | 用户 | LogonID

- 默认值 – 无
- 数据类型 – 字符串

用于访问 Amazon Redshift 服务器的用户名。

如果使用数据库身份验证，则此参数是必需项。

UseDeclareFetch

- 默认值 - 0
- 数据类型 – Boolean

当查询返回的结果集过大，无法全部存入内存时，启用该选项可让驱动程序按固定大小批次处理结果，而非一次性缓冲整个结果。无论查询返回多少行数据，该方式都能让客户端内存占用可控且可预测。它是一个布尔值，用于指定驱动程序是否使用 Declare/Fetch 模式，一次返回设定数量的行。要设置每批次返回的行数，请使用 `Fetch` 选项。有关更多信息，请参阅 取回。

- 1 | TRUE：驱动程序使用 Declare/Fetch 模式，一次返回 `Fetch` 选项指定的行数。
- 0 | FALSE：驱动程序一次性返回全部查询结果。

此参数为可选的。

> **Note**
>
> 如果 `UseDeclareFetch` 已启用，则无论您的 DSN 或连接字符串中设置的值如何，驱动程序都会禁用流式游标模式，并将 `StreamingCursorRows` 的值覆盖为 0。有关更多信息，请参阅 StreamingCursorRows。

UseUnicode

- 默认值 - 0
- 数据类型 – Boolean

一个布尔值，用于指定驱动程序是以 Unicode 还是常规 SQL 类型返回 Redshift 数据。

- 1 | TRUE：驱动程序返回字符数据类型的宽 SQL 类型。
  - 返回 SQL_WCHAR 而不是 SQL_CHAR。
  - 返回 SQL_WVARCHAR 而不是 SQL_VARCHAR。
  - 返回 SQL_WLONGVARCHAR 而不是 SQL_LONGVARCHAR。
- 0 | FALSE：驱动程序返回字符数据类型的正常 SQL 类型。
  - 返回 SQL_CHAR 而不是 SQL_WCHAR。
  - 返回 SQL_VARCHAR 而不是 SQL_WVARCHAR。

- 返回 SQL_LONGVARCHAR 而不是 SQL_WLONGVARCHAR。

此参数为可选的。它在驱动程序版本 2.1.15 和更高版本中可用。

web_identity_token

- 默认值 – 无
- 数据类型 – 字符串

身份提供者提供的 OAUTH 令牌。它将在 JWT 插件中使用。

如果您将 plugin_name 参数设置为 BasicJwtCredentialsProvider，则此参数是必需项。

早期 ODBC 驱动程序版本

仅当您的工具需要使用某个特定版本的驱动程序时，才能下载 Amazon Redshift ODBC 驱动程序版本 2.x 的以前版本。

使用适用于 Microsoft Windows 的以前 ODBC 驱动程序版本

以下是适用于 Microsoft Windows 的旧 Amazon Redshift ODBC 驱动程序版本 2.x：

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.3.0/AmazonRedshiftODBC64-2.2.3.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.2.0/AmazonRedshiftODBC64-2.2.2.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.1.0/AmazonRedshiftODBC64-2.2.1.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.0.0/AmazonRedshiftODBC64-2.2.0.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.17.0/AmazonRedshiftODBC64-2.1.17.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.16.0/AmazonRedshiftODBC64-2.1.16.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.15.0/AmazonRedshiftODBC64-2.1.15.0.msi

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.14.0/AmazonRedshiftODBC64-2.1.14.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.13.0/AmazonRedshiftODBC64-2.1.13.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.12.0/AmazonRedshiftODBC64-2.1.12.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.11.0/AmazonRedshiftODBC64-2.1.11.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.10.0/AmazonRedshiftODBC64-2.1.10.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.9.0/AmazonRedshiftODBC64-2.1.9.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.9.0/AmazonRedshiftODBC64-2.1.9.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.8.0/AmazonRedshiftODBC64-2.1.8.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.7.0/AmazonRedshiftODBC64-2.1.7.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.6.0/AmazonRedshiftODBC64-2.1.6.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.4.0/AmazonRedshiftODBC64-2.1.4.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.3.0/AmazonRedshiftODBC64-2.1.3.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.2.0/AmazonRedshiftODBC64-2.1.2.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.1.0/AmazonRedshiftODBC64-2.1.1.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.0.0/AmazonRedshiftODBC64-2.1.0.0.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.1.0/AmazonRedshiftODBC64-2.0.1.0.msi

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.11/AmazonRedshiftODBC64-2.0.0.11.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.9/AmazonRedshiftODBC64-2.0.0.9.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.8/AmazonRedshiftODBC64-2.0.0.8.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.7/AmazonRedshiftODBC64-2.0.0.7.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.6/AmazonRedshiftODBC64-2.0.0.6.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.5/AmazonRedshiftODBC64-2.0.0.5.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.3/AmazonRedshiftODBC64-2.0.0.3.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.1/AmazonRedshiftODBC64-2.0.0.1.msi

使用适用于 Linux 的以前 ODBC 驱动程序版本

以下是适用于 Linux 的旧 Amazon Redshift ODBC 驱动程序版本 2.x：

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.3.0/AmazonRedshiftODBC-64-bit-2.2.3.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.3.0/AmazonRedshiftODBC-64-bit-2.2.3.0.aarch64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.3.0/AmazonRedshiftODBC-64-bit-2.2.3.0.amd64.deb
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.3.0/AmazonRedshiftODBC-64-bit-2.2.3.0.arm64.deb
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.2.0/AmazonRedshiftODBC-64-bit-2.2.2.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.2.0/AmazonRedshiftODBC-64-bit-2.2.2.0.aarch64.rpm

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.2.0/AmazonRedshiftODBC-64-bit-2.2.2.0.amd64.deb
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.2.0/AmazonRedshiftODBC-64-bit-2.2.2.0.arm64.deb
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.1.0/AmazonRedshiftODBC-64-bit-2.2.1.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.1.0/AmazonRedshiftODBC-64-bit-2.2.1.0.aarch64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.1.0/AmazonRedshiftODBC-64-bit-2.2.1.0.amd64.deb
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.1.0/AmazonRedshiftODBC-64-bit-2.2.1.0.arm64.deb
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.0.0/AmazonRedshiftODBC-64-bit-2.2.0.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.0.0/AmazonRedshiftODBC-64-bit-2.2.0.0.aarch64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.0.0/AmazonRedshiftODBC-64-bit-2.2.0.0.amd64.deb
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.0.0/AmazonRedshiftODBC-64-bit-2.2.0.0.arm64.deb
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.17.0/AmazonRedshiftODBC-64-bit-2.1.17.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.17.0/AmazonRedshiftODBC-64-bit-2.1.17.0.aarch64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.16.0/AmazonRedshiftODBC-64-bit-2.1.16.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.16.0/AmazonRedshiftODBC-64-bit-2.1.16.0.aarch64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.15.0/AmazonRedshiftODBC-64-bit-2.1.15.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.15.0/AmazonRedshiftODBC-64-bit-2.1.15.0.aarch64.rpm

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.14.0/AmazonRedshiftODBC-64-bit-2.1.14.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.14.0/AmazonRedshiftODBC-64-bit-2.1.14.0.aarch64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.13.0/AmazonRedshiftODBC-64-bit-2.1.13.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.13.0/AmazonRedshiftODBC-64-bit-2.1.13.0.aarch64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.12.0/AmazonRedshiftODBC-64-bit-2.1.12.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.12.0/AmazonRedshiftODBC-64-bit-2.1.12.0.aarch64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.11.0/AmazonRedshiftODBC-64-bit-2.1.11.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.11.0/AmazonRedshiftODBC-64-bit-2.1.11.0.aarch64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.10.0/AmazonRedshiftODBC-64-bit-2.1.10.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.9.0/AmazonRedshiftODBC-64-bit-2.1.9.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.8.0/AmazonRedshiftODBC-64-bit-2.1.8.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.7.0/AmazonRedshiftODBC-64-bit-2.1.7.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.6.0/AmazonRedshiftODBC-64-bit-2.1.6.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.4.0/AmazonRedshiftODBC-64-bit-2.1.4.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.3.0/AmazonRedshiftODBC-64-bit-2.1.3.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.2.0/AmazonRedshiftODBC-64-bit-2.1.2.0.x86_64.rpm

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.1.0/AmazonRedshiftODBC-64-bit-2.1.1.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.0.0/AmazonRedshiftODBC-64-bit-2.1.0.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.1.0/AmazonRedshiftODBC-64-bit-2.0.1.0.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.11/AmazonRedshiftODBC-64-bit-2.0.0.11.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.9/AmazonRedshiftODBC-64-bit-2.0.0.9.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.8/AmazonRedshiftODBC-64-bit-2.0.0.8.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.7/AmazonRedshiftODBC-64-bit-2.0.0.7.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.6/AmazonRedshiftODBC-64-bit-2.0.0.6.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.5/AmazonRedshiftODBC-64-bit-2.0.0.5.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.3/AmazonRedshiftODBC-64-bit-2.0.0.3.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.0.0.1/AmazonRedshiftODBC-64-bit-2.0.0.1.x86_64.rpm

使用适用于 Apple macOS 的早期 ODBC 驱动程序版本

以下是适用于 Apple macOS 的旧 Amazon Redshift ODBC 驱动程序版本 2.x：

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.3.0/AmazonRedshiftODBC-64-bit.2.2.3.0.universal.pkg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.2.0/AmazonRedshiftODBC-64-bit.2.2.2.0.universal.pkg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.1.0/AmazonRedshiftODBC-64-bit.2.2.1.0.universal.pkg

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.2.0.0/AmazonRedshiftODBC-64-bit.2.2.0.0.universal.pkg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.17.0/AmazonRedshiftODBC-64-bit.2.1.17.0.universal.pkg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.16.0/AmazonRedshiftODBC-64-bit.2.1.16.0.universal.pkg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.15.0/AmazonRedshiftODBC-64-bit.2.1.15.0.universal.pkg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.14.0/AmazonRedshiftODBC-64-bit.2.1.14.0.universal.pkg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.13.0/AmazonRedshiftODBC-64-bit.2.1.13.0.universal.pkg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/2.1.12.0/AmazonRedshiftODBC-64-bit.2.1.12.0.universal.pkg

### ODBC 驱动程序 2.x 版本与 1.x 版本之间的差异

本节介绍 ODBC 1.x 驱动程序和 ODBC 2.x 驱动程序之间的差异，并提供有关如何迁移到 2.x 驱动程序的指导。它介绍了可能影响应用程序的更改以及如何解决这些更改。

#### 需要解决的主要差异

以下操作可以解决最常见的迁移问题。大多数应用程序只需要其中的一部分操作。

#### DSN 和连接字符串

- 查看您的 DSN，了解可能已移除或重命名的驱动程序选项。ODBC 2.x 驱动程序不支持某些 ODBC 1.x 驱动程序选项。其它选项已重命名。有关详细信息，请参阅连接字符串和 DSN 选项。
- 如果您的应用程序依赖于宽字符类型代码（**SQL_WVARCHAR**、**SQL_WCHAR**），请设置 **UseUnicode=true**。在 ODBC 2.x 驱动程序中，UseUnicode 默认为 `false`，这会报告窄字符类型。
- 将日志记录设置移至 **[ODBC]** 部分。在 Linux 和 macOS 上，必须在 `odbc.ini` 的全局 [ODBC] 部分中设置 LogLevel 和 LogPath，而不是在单个 DSN 部分中进行设置。

查询和架构

- 将 **EXTERNAL TABLE** 包含在 **SQLTables** 类型筛选条件中。在 ODBC 2.x 中，Amazon Redshift Spectrum 表和数据共享表均报告为 EXTERNAL TABLE。如果您按类型筛选 SQLTables，请添加 EXTERNAL TABLE 以继续查看这些对象。或者，将 EnableTableTypes 选项设置为 0，以便将详细表类型信息标准化为泛型 TABLE 和 VIEW 表类型，如以下项目所述。
- 默认情况下，详细表类型处于启用状态。默认情况下，ODBC 2.x 启用 EnableTableTypes 选项。默认情况下，1.x 驱动程序禁用了此选项。默认使用 2.x 的情况下，SQLTables 会报告详细类型，例如 SYSTEM TABLE、SYSTEM VIEW、EXTERNAL TABLE 和 LOCAL TEMPORARY。要将每个表报告为泛型类型 TABLE，将每个视图报告为 VIEW，请将 EnableTableTypes 设置为 0。
- 对于不支持间隔数据类型的应用程序，将 **INTERVAL** 强制转换为 **VARCHAR**。某些客户端不支持 ODBC 间隔类型。对于这些客户端，请在您的查询中强制转换以下列：SELECT col::VARCHAR FROM ...

应用程序代码

- 验证查询超时设置。ODBC 2.x 按照 ODBC 规范正确执行 SQL_ATTR_QUERY_TIMEOUT。ODBC 1.x 无提示忽略了此设置。以前成功的长时间运行的查询现在可能会因超时错误而失败。根据需要查看并调整超时值。
- 为执行时数据参数提供数据长度。2.x 将 SQL_NEED_LONG_DATA_LEN 报告为 Y（1.x 报告 N）。绑定 SQL_DATA_AT_EXEC 参数的应用程序现在必须在 StrLen_or_IndPtr 中预先提供总数据长度。

连接字符串和 DSN 选项

下表显示了 ODBC 1.x 驱动程序选项，这些选项在 ODBC 2.x 驱动程序中要么已重命名，要么具有直接等效项。

| 1.x 选项 | 2.x 等效项 | 备注 |
| --- | --- | --- |
| MaxLongVarChar | MaxLongVarcharSize | 默认值从 8190 更改为 65535。 |
| ConnectionTimeout | LoginTimeout | 相同的连接超时，已重命名。默认为 0（无超时）。 |

| 1.x 选项 | 2.x 等效项 | 备注 |
| --- | --- | --- |
| VpcEndpointUrl | vpc_endpoint_url | |
| SSLCertPath | TrustStore 或 CaFile | 用于验证服务器的 CA 证书的路径。在 Windows 上，在 DSN 设置对话框的“信任存储”字段中对此进行设置；该对话框没有 CaFile 字段。如果两者均已设置，则 TrustStore 优先。 |

当前的 2.x 驱动程序不支持以下 1.x 选项。2.x 驱动程序会忽略它们，因此它们不会影响您的连接。将它们从 DSN 中移除是可选的，但为了避免混淆，建议您这样做。

- SingleRowMode：要限制客户端内存，请改用 StreamingCursorRows 或 UseDeclareFetch。
- UseSystemTrustStore：不支持。在 Windows 上，1.x 驱动程序可以根据 Windows 系统证书存储验证服务器证书。2.x 驱动程序根据 CA 证书文件进行验证：默认情况下，它使用捆绑的 Amazon Redshift 根证书，或者使用您在 TrustStore 或 CaFile 中指定的文件。
- TextAsLongVarchar、CheckCertRevocation、EnableAwsSdkLogs、UseLogPrefix、Locale、当前版本中没有等效项。Amazon Redshift 团队正在评估未来版本中这些选项的等效项或替代项。

有关支持的 2.x 选项的完整列表，请参阅 ODBC 驱动程序选项。

数据类型更改

下表显示了 Amazon Redshift 数据类型映射，这些映射在 ODBC 2.x 中进行了更改以符合 ODBC 规范。大多数应用程序不受影响，因为它们按索引或名称而不是按类型代码绑定列。

| Amazon Redshift 数据类型 | ODBC 1.x | ODBC 2.x |
| --- | --- | --- |
| DOUBLE PRECISION | SQL_FLOAT (6) | SQL_DOUBLE (8) |
| INTERVAL YEAR TO MONTH | SQL_VARCHAR (12)，文本 | SQL_INTERVAL_YEAR_TO_MONTH (107)，间隔构造 |

| Amazon Redshift 数据类型 | ODBC 1.x | ODBC 2.x |
| --- | --- | --- |
| INTERVAL DAY TO SECOND | SQL_VARCHAR (12)，文本 | SQL_INTERVAL_DAY_TO_SECOND (110)，间隔构造 |
| VARCHAR / CHAR / TEXT | 宽类型（SQL_WVARCHAR 、SQL_WCHAR 、SQL_WLONGVARCHAR ）；默认 UseUnicode=true | 窄类型（SQL_VARCHAR 、SQL_CHAR、SQL_LONGVARCHAR ）。设置 UseUnicode=true 以还原宽类型。 |
| GEOMETRY/GEOGRAPHY （用作 SQL_C_BINARY ） | 十六进制编码的 ASCII 字符串 | 原始二进制字节 |

对于 GEOMETRY、GEOGRAPHY 和 SUPER 数据类型，SQLColumns 返回的 COLUMN_SIZE 值也发生了变化，它们现在根据 ODBC 规范返回 NULL 以表示未指定大小的列。基于 COLUMN_SIZE 分配缓冲区的应用程序必须使用默认缓冲区大小处理 NULL。

迁移后故障排除

下表描述了迁移后可能遇到的常见问题及其解决方法。

| 症状 | 原因 | 操作 |
| --- | --- | --- |
| 外部表在架构浏览器中不可见 | Spectrum 表和数据共享表报告为 EXTERNAL TABLE | 在您的 SQLTables 类型筛选条件中包含 EXTERNAL TABLE，或者将 EnableTableTypes 设置为 0，以将详细表类型标准化为泛型 TABLE 和 VIEW 类型。 |
| 间隔列上出现 pyodbc 错误 | pyodbc 不支持 ODBC 间隔类型 | 在查询中将间隔强制转换为 VARCHAR。 |
| 字符数据显示为意外字符（乱码） | UseUnicode 默认值更改为 false。需要宽字符（UTF-16）数据的应用程序可能会 | 在您的 DSN 中设置 UseUnicode=true ，或者更新您的应用程序以 |

| 症状 | 原因 | 操作 |
| --- | --- | --- |
| | 将窄字符字节误解释为宽对，从而产生乱码输出。数据本身没有变化。 | 将列绑定为 SQL_C_CHAR ，而不是 SQL_C_WCHAR 。 |
| 长时间运行的查询返回超时错误 | 现已强制执行 SQL_ATTR_QUERY_TIMEOUT 。ODBC 1.x 无提示忽略了此设置。 | 在 DSN 中增加或移除 QueryTimeout ，或者在应用程序中将 SQL_ATTR_QUERY_TIMEOUT 设置为 0。 |
| 设置 SQL_ATTR_CURRENT_CATALOG 会返回 HY011 | 无法在打开的连接上设置该属性。不支持在打开的连接上切换数据库。 | 在连接之前设置该属性，或者关闭并重新打开目标数据库上的连接。 |
| 连接在双栈环境中挂起 | 版本 <= 2.1.16 时的限制 | 升级至 2.1.17 或更高版本。 |
| 错误消息解析返回意外结果 | 消息格式已更改；SQLSTATE 代码未更改 | 解析 SQLSTATE 代码而不是消息文本。 |
| 执行时数据参数绑定的行为不同 | SQL_NEED_LONG_DATA_LEN 已从 FALSE 更改为 TRUE；使用 SQL_DATA_AT_EXEC 的应用程序现在必须预先提供总数据长度 | 当绑定 SQL_DATA_AT_EXEC 参数时，在 StrLen_or_IndPtr 中设置长度值。 |
| 未生成驱动程序日志 | LogLevel 和 LogPath 必须位于 odbc.ini 的 [ODBC] 全局部分，而不是在各个 DSN 部分中 | 将日志记录设置从您的 DSN 移至 [ODBC] 部分。 |

从较旧的 ODBC 2.x 版本升级

如果您已经在运行较旧的 ODBC 2.x 版本，请升级到最新版本。有关所有版本的增强功能和错误修复的完整列表，请参阅 GitHub 上的 Amazon Redshift ODBC 驱动程序更改日志。

更多信息

- 为 Amazon Redshift 配置 ODBC 驱动程序版本 2.x 连接

- ODBC 驱动程序选项
- Amazon Redshift ODBC 驱动程序更改日志

## 配置 ODBC 驱动程序版本 1.x 连接

您可以使用 ODBC 连接将许多第三方 SQL 客户端工具和应用程序连接到您的 Amazon Redshift 集群。为此，请在您的客户端计算机或 Amazon EC2 实例上设置连接。如果您的客户端工具支持 JDBC，您可以选择使用 JDBC 连接而非 ODBC 连接，因为 JDBC 连接更加易于配置。但是，如果您的客户端工具不支持 JDBC，请按此部分中的步骤配置 ODBC 连接。

Amazon Redshift 提供了适用于 Linux、Windows 和 macOS 操作系统的 64 位 ODBC 驱动程序。32 位 ODBC 驱动程序已停产。除了紧急安全补丁外，不会发布进一步的更新。

有关 ODBC 驱动程序功能和先决条件的最新信息，请参阅 Amazon Redshift ODBC 驱动程序发布说明。

有关 Amazon Redshift ODBC 驱动程序的安装和配置信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南。

主题

- 获取 ODBC URL
- 在 Microsoft Windows 上使用 Amazon Redshift ODBC 驱动程序
- 在 Linux 上使用 Amazon Redshift ODBC 驱动程序
- 在 macOS 上使用 Amazon Redshift ODBC 驱动程序
- ODBC 驱动程序选项
- 早期 ODBC 驱动程序版本

### 获取 ODBC URL

Amazon Redshift 在 Amazon Redshift 控制台中显示您的集群的 ODBC URL。此 URL 包含在客户端计算机与数据库之间建立连接时所需的信息。

ODBC URL 采用以下格式：`Driver={driver};Server=endpoint;Database=database_name;UID=user_name;PWD=pass`

前面显示的格式的字段具有以下值。

| 字段 | Value |
| --- | --- |
| Driver | 要使用的 64 位 ODBC 驱动程序的名称：Amazon Redshift (x64)。<br><br>32 位 ODBC 驱动程序的名称：Amazon Redshift (x86)。 |
| Server | Amazon Redshift 集群的端点。 |
| Database | 您为集群创建的数据库。 |
| UID | 有权连接到数据库的用户账户的用户名。该值是数据库权限，而非 Amazon Redshift 权限，但是您可以使用您在启动集群时设置的管理员用户账户。 |
| PWD | 用户账户用于连接数据库的密码。 |
| Port | 您在启动集群时指定的端口号。如果您启用了防火墙，请确保此端口处于打开状态，可供您使用。 |

上表中的字段可以包含以下特殊字符：

```
[]{}(),;?*=!@
```

如果您使用这些特殊字符，则必须用大括号括起值。例如，连接字符串 `PWD={Your;password123};` 中的密码值表示为 `Your;password123`。

由于 `Field=value` 对使用分号分隔，因此中间具有任意数量空格的 `}` 和 `;` 组合将被视为 `Field={value};` 对的结尾。我们建议您避免在字段值中使用序列 `};`。例如，如果您将密码值设置为 `PWD={This is a passwor} ;d};`，则密码将为 `This is a passwor} ;`，URL 将出错。

以下是一个示例 ODBC URL。

```
Driver={Amazon Redshift (x64)};
                    Server=examplecluster.abc123xyz789.us-
west-2.redshift.amazonaws.com;
                    Database=dev;
                    UID=adminuser;
                    PWD=insert_your_admin_user_password_here;
                    Port=5439
```

有关如何获取 ODBC 连接的信息，请参阅 查找集群连接字符串。

在 Microsoft Windows 上使用 Amazon Redshift ODBC 驱动程序

您可以在可访问 Amazon Redshift 数据仓库的客户端计算机上安装 Amazon Redshift ODBC 驱动程序。在其上安装该驱动程序的每台计算机都必须满足一系列最低系统要求。有关最低系统要求的信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南。

主题

- 下载并安装 Amazon Redshift ODBC 驱动程序
- 为 ODBC 连接创建系统 DSN 条目

下载并安装 Amazon Redshift ODBC 驱动程序

使用以下过程下载适用于 Windows 操作系统的 Amazon Redshift ODBC 驱动程序。仅在您当前运行的第三方应用程序获得了使用 Amazon Redshift 的认证并且需要特定的驱动程序时，才使用上述驱动程序之外的驱动程序。

安装 ODBC 驱动程序

1. 根据您的 SQL 客户端工具或应用程序的系统架构，下载以下驱动程序之一：

   - 64 位 ODBC 驱动程序版本 1.6.3

     此驱动程序的名称为 Amazon Redshift (x64)。

   - 32 位 ODBC 驱动程序版本 1.4.52

     此驱动程序的名称为 Amazon Redshift (x86)。32 位 ODBC 驱动程序已停产。除了紧急安全补丁外，不会发布进一步的更新。

   > **Note**
   >
   > 根据您的 SQL 客户端工具或应用程序的系统架构，下载相应的 MSI 包。例如，如果您的 SQL 客户端工具是 64 位，则安装 64 位驱动程序。

   然后，下载并查看 Amazon Redshift ODBC 和 JDBC 驱动程序许可证协议。

2. 双击 .msi 文件，然后按照向导中的步骤安装驱动程序。

为 ODBC 连接创建系统 DSN 条目

下载并安装 ODBC 驱动程序后，将数据来源名称 (DSN) 条目添加到客户端计算机或 Amazon EC2 实例。SQL 客户端工具将使用此数据源连接到 Amazon Redshift 数据库。

建议您创建系统 DSN 而不是用户 DSN。一些应用程序使用不同的用户账户加载数据。这些应用程序可能无法检测在其他用户账户下创建的用户 DSN。

> **Note**
>
> 对于使用 AWS Identity and Access Management (IAM) 凭证或身份提供者 (IdP) 凭证进行的身份验证，需要执行其他步骤。有关更多信息，请参阅 步骤 5：配置 JDBC 或 ODBC 连接以使用 IAM 凭证。

有关如何创建系统 DSN 条目的信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南。

在 Windows 上为 ODBC 连接创建系统 DSN 条目

1. 在开始菜单上，打开ODBC 数据源。

   请确保您选择的 ODBC Data Source Administrator 的位数与用于连接 Amazon Redshift 的客户端应用程序的位数相同。
2. 在 ODBC 数据数据源管理器中，选择驱动程序选项卡并找到驱动程序文件夹。

   - Amazon Redshift ODBC 驱动程序（64 位）
   - Amazon Redshift ODBC 驱动程序（32 位）
3. 选择系统 DSN 选项卡为计算机上的所有用户配置驱动程序，或选择用户 DSN 选项卡仅为您的用户账户配置驱动程序。
4. 选择添加。系统随即打开 Create New Data Source 窗口。
5. 选择 Amazon Redshift ODBC 驱动程序，然后选择完成。系统随即打开 Amazon Redshift ODBC Driver DSN Setup 窗口。
6. 在 Connection Settings 下，输入以下信息：

   Data source name

   输入数据源的名称。在稍后创建到集群的连接时，您可以使用任何名称来标识该数据源。例如，如果您遵循的是《Amazon Redshift 入门指南》，则可键入 `exampleclusterdsn`，以便轻松记住将与此 DSN 关联的集群。

服务器

为您的 Amazon Redshift 集群指定端点。您可以在 Amazon Redshift 控制台中的集群详细信息页面上找到该信息。有关更多信息，请参阅 在 Amazon Redshift 中配置连接。

端口

输入数据库使用的端口号。使用集群在启动或修改时配置为使用的端口。

数据库

输入 Amazon Redshift 数据库的名称。如果您在未指定数据库名称的情况下启动了集群，请输入 *dev*。否则，请使用您在启动过程中选择的名称。如果您遵循的是《Amazon Redshift 入门指南》，输入 *dev*。

7. 在身份验证下，指定配置选项以配置标准或 IAM 身份验证。有关身份验证选项的信息，请参阅《Amazon Redshift ODBC 连接器安装和配置指南》中的“在 Windows 上配置身份验证”。
8. 在 SSL Settings 下，指定以下项目的值：

   SSL authentication

   选择处理安全套接字层 (SSL) 的模式。在测试环境中，可以使用 `prefer`。但是，对于生产环境以及在需要安全交换数据时，请使用 `verify-ca` 或 `verify-full`。有关在 Windows 上使用 SSL 的更多信息，请参阅《Amazon Redshift ODBC 连接器安装和配置指南》中的“在 Windows 上配置 SSL 验证”。
9. 在其他选项下，指定有关如何将查询结果返回至您的 SQL 客户端工具或应用程序的选项。有关更多信息，请参阅《Amazon Redshift ODBC 连接器安装和配置指南》中的“在 Windows 上配置其他选项”。
10. 在日志记录选项中，指定日志记录选项的值。有关更多信息，请参阅《Amazon Redshift ODBC 连接器安装和配置指南》中的“在 Windows 上配置日志记录选项”。

    然后选择确定。
11. 在数据类型选项 下，指定数据类型的值。有关更多信息，请参阅《Amazon Redshift ODBC 连接器安装和配置指南》中的“在 Windows 上配置数据类型选项”。

    然后选择确定。
12. 选择测试。如果客户端计算机可以连接到 Amazon Redshift 数据库，您会看到以下消息：连接成功。

如果客户端计算机无法连接到数据库，您可以进行故障排除，解决可能存在的问题。有关更多信息，请参阅 解决 Amazon Redshift 中的连接问题。

13. 在 Windows 上配置 TCP Keepalive 以防止连接超时。有关如何在 Windows 上配置 TCP Keepalive 的信息，请参阅《Amazon Redshift ODBC 连接器安装和配置指南》。
14. 要帮助进行故障排除，请配置日志记录。有关如何在 Windows 上配置日志记录的信息，请参阅《Amazon Redshift ODBC 连接器安装和配置指南》。

## 在 Linux 上使用 Amazon Redshift ODBC 驱动程序

您可以在可访问 Amazon Redshift 数据仓库的客户端计算机上安装 Amazon Redshift ODBC 驱动程序。在其上安装该驱动程序的每台计算机都必须满足一系列最低系统要求。有关最低系统要求的信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南。

主题

- 下载并安装 Amazon Redshift ODBC 驱动程序
- 使用 ODBC 驱动程序管理器配置驱动程序

### 下载并安装 Amazon Redshift ODBC 驱动程序

按照此部分中的步骤在受支持的 Linux 发行版上下载并安装 Amazon Redshift ODBC 驱动程序。在安装过程中，驱动程序文件将被安装到以下目录中：

- `/opt/amazon/redshiftodbc/lib/64`（对于 64 位驱动程序）
- `/opt/amazon/redshiftodbc/ErrorMessages`
- `/opt/amazon/redshiftodbc/Setup`
- `/opt/amazon/redshiftodbc/lib/32`（对于 32 位驱动程序）

#### 要安装 Amazon Redshift ODBC 驱动程序

1. 根据您的 SQL 客户端工具或应用程序的系统架构，下载以下驱动程序之一：
   - 64 位 RPM 驱动程序版本 1.6.3
   - 64 位 Debian 驱动程序版本 1.6.3
   - 32 位驱动程序版本 1.4.52

这些驱动程序的名称均是 Amazon Redshift ODBC 驱动程序。32 位 ODBC 驱动程序已停产。除了紧急安全补丁外，不会发布进一步的更新。

> **Note**
>
> 下载适用于您的 SQL 客户端工具或应用程序系统架构的程序包。例如，如果您的客户端工具是 64 位，则安装 64 位驱动程序。

然后，下载并查看 Amazon Redshift ODBC 和 JDBC 驱动程序许可证协议。

2. 转至您下载程序包的位置，然后运行以下命令之一。使用适用于您的 Linux 发行版的命令。

   - 在 RHEL 和 CentOS 操作系统上，运行以下命令。

     ```
     yum -nogpgcheck localinstall RPMFileName
     ```

     将 *RPMFileName* 替换为 RPM 包文件名。例如，以下命令将演示如何安装 64 位驱动程序。

     ```
     yum -nogpgcheck localinstall AmazonRedshiftODBC-64-bit-1.x.xx.xxxx-x.x86_64.rpm
     ```

   - 在 SLES 上，运行以下命令。

     ```
     zypper install RPMFileName
     ```

     将 *RPMFileName* 替换为 RPM 包文件名。例如，以下命令将演示如何安装 64 位驱动程序。

     ```
     zypper install AmazonRedshiftODBC-1.x.x.xxxx-x.x86_64.rpm
     ```

   - 在 Debian 上，运行以下命令。

     ```
     sudo apt install ./DEBFileName.deb
     ```

     将 *DEBFileName.deb* 替换为 Debian 包文件名。例如，以下命令将演示如何安装 64 位驱动程序。

     ```
     sudo apt install ./AmazonRedshiftODBC-1.x.x.xxxx-x.x86_64.deb
     ```

> **Important**
>
> 安装完驱动程序后，请对其进行配置以在您的系统上使用。有关驱动程序配置的更多信息，请参阅使用 ODBC 驱动程序管理器配置驱动程序。

使用 ODBC 驱动程序管理器配置驱动程序

在 Linux 操作系统上，您可以使用 ODBC 驱动程序管理器来配置 ODBC 连接设置。ODBC 驱动程序管理器使用配置文件来定义和配置 ODBC 数据来源和驱动程序。您可以使用的 ODBC 驱动程序管理器取决于您使用的操作系统。对于 Linux，则是 unixODBC 驱动程序管理器。

有关用来配置 Amazon Redshift ODBC 驱动程序的受支持的 ODBC 驱动程序管理器的更多信息，请参阅 在 Linux 上使用 Amazon Redshift ODBC 驱动程序（适用于 Linux 操作系统）。另请参阅 Amazon Redshift ODBC 连接器安装和配置指南中的“在非 Windows 计算机上指定 ODBC 驱动程序管理器”。。

要配置 Amazon Redshift ODBC 驱动程序，需要提供以下三个文件：`amazon.redshiftodbc.ini`、`odbc.ini` 和 `odbcinst.ini`。

如果您将驱动程序安装在默认位置，`amazon.redshiftodbc.ini` 配置文件则位于以下目录之一：

- `/opt/amazon/redshiftodbc/lib/64`（适用于 Linux 操作系统上的 64 位驱动程序）
- `/opt/amazon/redshiftodbc/lib/32`（适用于 Linux 操作系统上的 32 位驱动程序）

此外，在 Linux 上的 `/opt/amazon/redshiftodbc/Setup` 下，还有示例 `odbc.ini` 和 `odbcinst.ini` 文件。您可以使用这些文件作为配置 Amazon Redshift ODBC 驱动程序和数据来源名称 (DSN) 的示例。

我们不建议使用 Amazon Redshift ODBC 驱动程序安装目录来存储配置文件。`Setup` 目录中的示例文件仅用作示例。如果您日后重新安装 Amazon Redshift ODBC 驱动程序，或将其升级到新版本，安装目录会被覆盖。之后，您将丢失对这些文件所做的全部更改。

为了避免出现这种情况，请将 `amazon.redshiftodbc.ini` 文件复制到安装目录以外的其他目录中。如果您要将此文件复制到用户的主目录，请在文件名的开头添加一个句点 (.)，使其成为隐藏文件。

对于 `odbc.ini` 和 `odbcinst.ini` 文件，应在用户的主目录中使用配置文件，或者在其他目录中创建新版本。默认情况下，您的 Linux 操作系统应在用户的主目录（`/home/$USER` 或 `~/`）中包含 `odbc.ini` 文件和 `odbcinst.ini` 文件。这些默认文件均为隐藏文件（通过在每个文件名的前面添加圆点 (.) 表示）。这些文件仅当您使用 `-a` 标志列出目录内容时显示。

对于 odbc.ini 和 odbcinst.ini 文件，不管您选择哪个选项，都需对这些文件进行修改，以添加驱动程序和 DSN 配置信息。如果您创建新文件，则还需设置环境变量，以指定这些配置文件的目标存储位置。

默认情况下，ODBC 驱动程序管理器将配置为使用主目录中隐藏的 odbc.ini 和 odbcinst.ini 配置文件版本（名为 .odbc.ini 和 .odbcinst.ini）。它们也被配置为使用驱动程序安装目录的 amazon.redshiftodbc.ini 子文件夹中的 /lib 文件。如果您将这些配置文件存储在其他位置，请设置如下所述的环境变量，以便驱动程序管理器能够找到这些文件。有关更多信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南中的“指定驱动程序配置文件的位置”。。

在 Linux 操作系统上创建数据来源名称

在使用数据源名称 (DSN) 连接到数据存储时，请配置 odbc.ini 文件以定义 DSN。在 odbc.ini 文件中设置属性以创建指定数据存储的连接信息的 DSN。

有关如何配置 odbc.ini 文件的信息，请参阅 Amazon Redshift ODBC connector installation and configuration guide 中的“Creating a Data Source Name on a Non-Windows Machine”

请在 Linux 操作系统上使用以下格式。

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file

Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

以下示例显示了在 Linux 操作系统上使用 64 位 ODBC 驱动程序配置 odbc.ini。

```
[ODBC Data Sources]
Amazon_Redshift_x64=Amazon Redshift (x64)

[Amazon Redshift (x64)]
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US
```

以下示例显示了在 Linux 操作系统上使用 32 位 ODBC 驱动程序配置 odbc.ini。

```
[ODBC Data Sources]
Amazon_Redshift_x32=Amazon Redshift (x86)

[Amazon Redshift (x86)]
Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.so
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US
```

在 Linux 操作系统上配置没有 DSN 的连接

要通过不带 DSN 的连接来连接到数据存储，请在 odbcinst.ini 文件中定义驱动程序。然后，在应用程序中提供一个无 DSN 的连接字符串。

有关在这种情况下如何配置 odbcinst.ini 文件的信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南中的“在非 Windows 计算机上配置没有 DSN 的连接”。。

请在 Linux 操作系统上使用以下格式。

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/driver_file

...
```

以下示例显示了安装在 Linux 操作系统默认目录中的 64 位驱动程序的 odbcinst.ini 配置。

```
[ODBC Drivers]
Amazon Redshift (x64)=Installed

[Amazon Redshift (x64)]
Description=Amazon Redshift ODBC Driver (64-bit)
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so
```

以下示例显示了安装在 Linux 操作系统默认目录中的 32 位驱动程序的 odbcinst.ini 配置。

```
[ODBC Drivers]
Amazon Redshift (x86)=Installed

[Amazon Redshift (x86)]
Description=Amazon Redshift ODBC Driver (32-bit)
Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.so
```

配置环境变量

使用正确的 ODBC 驱动程序管理器加载正确的驱动程序。为此，请设置库路径环境变量。有关更多信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南中的“在非 Windows 计算机上指定 ODBC 驱动程序管理器”。。

默认情况下，ODBC 驱动程序管理器将配置为使用主目录中隐藏的 `odbc.ini` 和 `odbcinst.ini` 配置文件版本（名为 `.odbc.ini` 和 `.odbcinst.ini`）。它们也被配置为使用驱动程序安装目录的 `/lib` 子文件夹中的 `amazon.redshiftodbc.ini` 文件。如果您将这些配置文件存储在其他位置，请设置环境变量，以便驱动程序管理器能够找到这些文件。有关更多信息，请参阅《Amazon Redshift ODBC 连接器安装和配置指南中的“指定驱动程序配置文件的位置”。

配置连接功能

可以为 ODBC 设置配置以下连接功能：

- 配置 ODBC 驱动程序以提供凭证并验证与 Amazon Redshift 数据库的连接。
- 如果要连接到启用了 SSL 的 Amazon Redshift 服务器，请将 ODBC 驱动程序配置为连接到启用了安全套接字层 (SSL) 的套接字。
- 将 ODBC 驱动程序配置为通过代理服务器连接到 Amazon Redshift。
- 将 ODBC 驱动程序配置为使用查询处理模式来防止查询占用过多内存。
- 将 ODBC 驱动程序配置为通过代理服务器传递 IAM 身份验证过程。
- 将 ODBC 驱动程序配置为使用 TCP Keepalive 来防止连接超时。

有关这些连接功能的信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南。

在 macOS 上使用 Amazon Redshift ODBC 驱动程序

您可以在可访问 Amazon Redshift 数据仓库的客户端计算机上安装驱动程序。在其上安装该驱动程序的每台计算机都必须满足一系列最低系统要求。有关最低系统要求的信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南。

主题


- 下载并安装 Amazon Redshift ODBC 驱动程序
- 使用 ODBC 驱动程序管理器配置驱动程序


## 下载并安装 Amazon Redshift ODBC 驱动程序

按照此部分中的步骤在受支持的 macOS 版本上下载并安装 Amazon Redshift ODBC 驱动程序。在安装过程中，驱动程序文件将被安装到以下目录中：

- `/opt/amazon/redshift/lib/universal`
- `/opt/amazon/redshift/ErrorMessages`
- `/opt/amazon/redshift/Setup`

### 在 macOS 上安装 Amazon Redshift ODBC 驱动程序

1. 要在 macOS 上安装 Amazon Redshift ODBC 驱动程序，请下载 macOS 驱动程序版本 1.6.3。

   然后，下载并查看 Amazon Redshift ODBC 和 JDBC 驱动程序许可证协议。
2. 双击 AmazonRedshiftODBC.pkg 以运行安装程序。
3. 按照安装程序中的步骤完成驱动程序的安装过程。要执行安装，需同意许可协议的条款。

> **Important**
>
> 安装完驱动程序后，请对其进行配置以在您的系统上使用。有关驱动程序配置的更多信息，请参阅使用 ODBC 驱动程序管理器配置驱动程序。

## 使用 ODBC 驱动程序管理器配置驱动程序

在 macOS 操作系统上，您可以使用 ODBC 驱动程序管理器来配置 ODBC 连接设置。ODBC 驱动程序管理器使用配置文件来定义和配置 ODBC 数据来源和驱动程序。您可以使用的 ODBC 驱动程序管理器取决于您使用的操作系统。对于 macOS 操作系统，则是 iODBC 驱动程序管理器。

有关用来配置 Amazon Redshift ODBC 驱动程序的受支持的 ODBC 驱动程序管理器的更多信息，请参阅在 macOS 上使用 Amazon Redshift ODBC 驱动程序（适用于 macOS 操作系统）。另请参阅 Amazon Redshift ODBC 连接器安装和配置指南中的“在非 Windows 计算机上指定 ODBC 驱动程序管理器”。。

要配置 Amazon Redshift ODBC 驱动程序，需要提供以下三个文件：amazon.redshiftodbc.ini、odbc.ini 和 odbcinst.ini。

如果您将驱动程序安装在默认位置，则 amazon.redshiftodbc.ini 配置文件将位于 /opt/amazon/redshift/lib 中。

此外，在 macOS 上的 /opt/amazon/redshift/Setup 下，还有示例 odbc.ini 和 odbcinst.ini 文件。您可以使用这些文件作为配置 Amazon Redshift ODBC 驱动程序和数据来源名称 (DSN) 的示例。

我们不建议使用 Amazon Redshift ODBC 驱动程序安装目录来存储配置文件。Setup 目录中的示例文件仅用作示例。如果您日后重新安装 Amazon Redshift ODBC 驱动程序，或将其升级到新版本，安装目录会被覆盖。之后，您将丢失对这些文件所做的全部更改。

为了避免出现这种情况，请将 amazon.redshiftodbc.ini 文件复制到安装目录以外的其他目录中。如果您要将此文件复制到用户的主目录，请在文件名的开头添加一个句点 (.)，使其成为隐藏文件。

对于 odbc.ini 和 odbcinst.ini 文件，应在用户的主目录中使用配置文件，或者在其他目录中创建新版本。默认情况下，您的 macOS 操作系统应在用户的主目录（/home/$USER 或 ~/）中具有 odbc.ini 文件和 odbcinst.ini 文件。这些默认文件均为隐藏文件（通过在每个文件名的前面添加圆点 (.) 表示）。这些文件仅当您使用 -a 标志列出目录内容时显示。

对于 odbc.ini 和 odbcinst.ini 文件，不管您选择哪个选项，都需对这些文件进行修改，以添加驱动程序和 DSN 配置信息。如果您创建新文件，则还需设置环境变量，以指定这些配置文件的目标存储位置。

默认情况下，ODBC 驱动程序管理器将配置为使用主目录中隐藏的 odbc.ini 和 odbcinst.ini 配置文件版本（名为 .odbc.ini 和 .odbcinst.ini）。它们也被配置为使用驱动程序安装目录的 amazon.redshiftodbc.ini 子文件夹中的 /lib 文件。如果您将这些配置文件存储在其他位置，请设置如下所述的环境变量，以便驱动程序管理器能够找到这些文件。有关更多信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南中的“指定驱动程序配置文件的位置”。。

在 macOS X 操作系统上创建数据来源名称

在使用数据源名称 (DSN) 连接到数据存储时，请配置 odbc.ini 文件以定义 DSN。在 odbc.ini 文件中设置属性以创建指定数据存储的连接信息的 DSN。

有关如何配置 odbc.ini 文件的信息，请参阅 Amazon Redshift ODBC connector installation and configuration guide 中的“Creating a Data Source Name on a Non-Windows Machine”。

在 macOS 操作系统上使用以下格式。

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/lib/amazonredshiftodbc.dylib

Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

以下示例显示了 macOS 操作系统上 odbc.ini 的配置。

```
[ODBC Data Sources]
Amazon_Redshift_dylib=Amazon Redshift DSN for macOS

[Amazon Redshift DSN for macOS]
Driver=/opt/amazon/redshift/lib/amazonredshiftodbc.dylib
Host=examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com
Port=5932
Database=dev
locale=en-US
```

在 macOS 操作系统上配置没有 DSN 的连接

要通过不带 DSN 的连接来连接到数据存储，请在 odbcinst.ini 文件中定义驱动程序。然后，在应用程序中提供一个无 DSN 的连接字符串。

有关在这种情况下如何配置 odbcinst.ini 文件的信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南中的“在非 Windows 计算机上配置没有 DSN 的连接”。。

在 macOS 操作系统上使用以下格式。

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
```

```
Driver=path/lib/amazonredshiftodbc.dylib

...
```

以下示例显示了安装在 macOS 操作系统默认目录中的驱动程序的 odbcinst.ini 配置。

```
[ODBC Drivers]
Amazon RedshiftODBC DSN=Installed

[Amazon RedshiftODBC DSN]
Description=Amazon Redshift ODBC Driver for macOS
Driver=/opt/amazon/redshift/lib/amazonredshiftodbc.dylib
```

配置环境变量

使用正确的 ODBC 驱动程序管理器加载正确的驱动程序。为此，请设置库路径环境变量。有关更多信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南中的“在非 Windows 计算机上指定 ODBC 驱动程序管理器”。。

默认情况下，ODBC 驱动程序管理器将配置为使用主目录中隐藏的 odbc.ini 和 odbcinst.ini 配置文件版本（名为 .odbc.ini 和 .odbcinst.ini）。它们也被配置为使用驱动程序安装目录的 /lib 子文件夹中的 amazon.redshiftodbc.ini 文件。如果您将这些配置文件存储在其他位置，请设置环境变量，以便驱动程序管理器能够找到这些文件。有关更多信息，请参阅《Amazon Redshift ODBC 连接器安装和配置指南中的“指定驱动程序配置文件的位置”。

配置连接功能

可以为 ODBC 设置配置以下连接功能：

- 配置 ODBC 驱动程序以提供凭证并验证与 Amazon Redshift 数据库的连接。
- 如果要连接到启用了 SSL 的 Amazon Redshift 服务器，请将 ODBC 驱动程序配置为连接到启用了安全套接字层 (SSL) 的套接字。
- 将 ODBC 驱动程序配置为通过代理服务器连接到 Amazon Redshift。
- 将 ODBC 驱动程序配置为使用查询处理模式来防止查询占用过多内存。
- 将 ODBC 驱动程序配置为通过代理服务器传递 IAM 身份验证过程。
- 将 ODBC 驱动程序配置为使用 TCP Keepalive 来防止连接超时。

有关这些连接功能的信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南。

ODBC 驱动程序选项

可以使用配置选项来控制 Amazon Redshift ODBC 驱动程序的行为。

在 Microsoft Windows 中，您通常可以在配置数据源名称 (DSN) 时设置驱动程序选项。您还能在以编程方式连接时，或者通过在 HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI*your_DSN* 中添加或更改注册表项来设置驱动程序选项。有关配置 DSN 的更多信息，请参阅在 Microsoft Windows 上使用 Amazon Redshift ODBC 驱动程序。

在 macOS 中，您可以在 odbc.ini 和 amazon.redshiftodbc.ini 文件中设置驱动程序配置选项，如使用 ODBC 驱动程序管理器配置驱动程序中所述。在 amazon.redshiftodbc.ini 文件中设置的配置选项适用于所有连接。相反，odbc.ini 文件中的设置配置选项特定于一个连接。在 odbc.ini 中设置的配置选项优先于在 amazon.redshiftodbc.ini 中设置的配置选项。

有关如何设置 ODBC 驱动程序配置选项的信息，请参阅 Amazon Redshift ODBC 连接器安装和配置指南。

早期 ODBC 驱动程序版本

仅当您的工具需要使用某个特定版本的驱动程序时，才能下载 Amazon Redshift ODBC 驱动程序的以前版本。

适用于 Windows 的早期 ODBC 驱动程序版本

以下是 64 位驱动程序：

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.3.1006/AmazonRedshiftODBC64-1.6.3.1006.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.1.1000/AmazonRedshiftODBC64-1.6.1.1000.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.20.1024/AmazonRedshiftODBC64-1.5.20.1024.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.16.1019/AmazonRedshiftODBC64-1.5.16.1019.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.9.1011/AmazonRedshiftODBC64-1.5.9.1011.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.7.1007/AmazonRedshiftODBC64-1.5.7.1007.msi

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.65.1000/AmazonRedshiftODBC64-1.4.65.1000.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.62.1000/AmazonRedshiftODBC64-1.4.62.1000.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.59.1000/AmazonRedshiftODBC64-1.4.59.1000.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.56.1000/AmazonRedshiftODBC64-1.4.56.1000.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.53.1000/AmazonRedshiftODBC64-1.4.53.1000.msi
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.52.1000/AmazonRedshiftODBC64-1.4.52.1000.msi

32 位驱动程序已停用，并且早期版本不受支持。

适用于 Linux 的早期 ODBC 驱动程序版本

以下是 64 位驱动程序版本：

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.3.1006/AmazonRedshiftODBC-64-bit-1.6.3.1006-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.1.1000/AmazonRedshiftODBC-64-bit-1.6.1.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.20.1024/AmazonRedshiftODBC-64-bit-1.5.20.1024-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.16.1019/AmazonRedshiftODBC-64-bit-1.5.16.1019-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.9.1011/AmazonRedshiftODBC-64-bit-1.5.9.1011-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.7.1007/AmazonRedshiftODBC-64-bit-1.5.7.1007-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.65.1000/AmazonRedshiftODBC-64-bit-1.4.65.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.62.1000/AmazonRedshiftODBC-64-bit-1.4.62.1000-1.x86_64.rpm

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.59.1000/AmazonRedshiftODBC-64-bit-1.4.59.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.59.1000/AmazonRedshiftODBC-64-bit-1.4.59.1000-1.x86_64.deb
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.56.1000/AmazonRedshiftODBC-64-bit-1.4.56.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.56.1000/AmazonRedshiftODBC-64-bit-1.4.56.1000-1.x86_64.deb
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.52.1000/AmazonRedshiftODBC-64-bit-1.4.52.1000-1.x86_64.rpm
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.52.1000/AmazonRedshiftODBC-64-bit-1.4.52.1000-1.x86_64.deb

32 位驱动程序已停用，并且早期版本不受支持。

适用于 macOS 的早期 ODBC 驱动程序版本

以下是适用于 macOS 的 Amazon Redshift ODBC 驱动程序版本：

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.3.1006/AmazonRedshiftODBC-64-bit.1.6.3.1006.universal.pkg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.6.1.1000/AmazonRedshiftODBC-64-bit.1.6.1.1000.universal.pkg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.20.1024/AmazonRedshiftODBC-1.5.20.1024.arm64.dmg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.20.1024/AmazonRedshiftODBC-1.5.20.1024.x86_64.dmg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.16.1019/AmazonRedshiftODBC-1.5.16.1019.x86_64.dmg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.9.1011/AmazonRedshiftODBC-1.5.9.1011.x86_64.dmg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.5.7.1007/AmazonRedshiftODBC-1.5.7.1007.x86_64.dmg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.65.1000/AmazonRedshiftODBC-1.4.65.1000.dmg

- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.62.1000/AmazonRedshiftODBC-1.4.62.1000.dmg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.59.1000/AmazonRedshiftODBC-1.4.59.1000.dmg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.56.1000/AmazonRedshiftODBC-1.4.56.1000.dmg
- https://s3.amazonaws.com/redshift-downloads/drivers/odbc/1.4.52.1000/AmazonRedshiftODBC-1.4.52.1000.dmg

# 用于发现元数据的最佳实践

Amazon Redshift 提供了多种发现元数据的方式。对于应用程序和工具，包括使用人工智能代理构建的应用程序和工具，我们建议您使用驱动程序元数据 API 或 SHOW 命令来进行一致、可靠和高性能的元数据发现。

主题


- 将 Amazon Redshift 驱动程序元数据 API 用于应用程序和工具
- 使用 SHOW 命令
- 查询系统表
- 设置应用程序名称连接属性


## 将 Amazon Redshift 驱动程序元数据 API 用于应用程序和工具

对于连接到 Amazon Redshift 的应用程序和工具，例如商业智能工具或查询编辑器，我们建议您使用 Amazon Redshift JDBC 2.x、ODBC 2.x 或 Python 驱动程序提供的驱动程序元数据 API 来发现有关数据仓库对象的元数据，包括数据库、架构、表、列和数据类型。作为替代方案，您可以使用 Amazon Redshift SHOW 命令。

使用驱动程序元数据 API 可获得以下优势：

- 符合规范。JDBC 和 ODBC 驱动程序实施了标准的元数据接口（在 JDBC 为 `DatabaseMetaData`，在 ODBC 中为 `SQLTables` 和 `SQLColumns`）。由于 Python 的 DB-API（PEP 249）未定义元数据 API 规范，因此 Amazon Redshift Python 驱动程序遵循 JDBC DatabaseMetaData 规范，提供了等效的方法，例如 `get_tables()`、`get_columns()` 和 `get_schemas()`。这些 API 遵循明确定义的规范，因此您的集成代码是可移植的。随着 Amazon Redshift 不断发展其内部系统表，您的应用程序不需要更改。

- 性能优化。驱动程序元数据 API 经过优化，可高效地返回元数据。AWS 继续投资改进驱动程序元数据 API 性能。
- 向前兼容。Amazon Redshift 遵循 JDBC、ODBC 和 Python 连接器规范。当您根据这些标准 API 进行编码时，应用程序受到保护，而不受底层系统目录结构更改所影响。

示例：使用 JDBC DatabaseMetaData.getTables() 检索表元数据

```
DatabaseMetaData dbmd = connection.getMetaData();

// getTables(catalog, schemaPattern, tableNamePattern, types)
//   catalog:          "test" – filters to the database named "test"
//   schemaPattern:    "test_pattern" – filters schemas matching this pattern (supports
 SQL wildcards % and _)
//   tableNamePattern: null – no filter, returns all table names
//   types:            {"TABLE", "EXTERNAL TABLE"} – only return regular tables and
 external tables
ResultSet rs = dbmd.getTables("test", "test_pattern", null, new String[] {"TABLE",
 "EXTERNAL TABLE"});
```

示例：使用 Python cursor.get_columns() 检索列元数据

```
cursor: redshift_connector.Cursor = conn.cursor()

# get_columns(catalog, schema_pattern, table_name_pattern, column_name_pattern)
#   catalog:             'test' – filters to the database named "test"
#   schema_pattern:      'test_pattern' – filters schemas matching this pattern
 (supports SQL wildcards % and _)
#   table_name_pattern:  'testabc' – filters to the table named "testabc"
#   column_name_pattern: '%' – wildcard, returns all columns in the matching table
result: tuple = cursor.get_columns('test', 'test_pattern', 'testabc', '%')
```

示例：使用 ODBC SQLPrimaryKeys() 检索主键元数据

```
// SQLPrimaryKeys(hstmt, catalog, catalog_len, schema, schema_len, table, table_len)
//   catalog: "test"         – filters to the database named "test"
//   schema:  "test_schema"  – filters to the schema named "test_schema"
//   table:   "test_table"   – retrieves primary key columns for this table
// Note: Unlike getTables/getColumns, SQLPrimaryKeys does NOT support wildcard
 patterns.
retcode = SQLPrimaryKeys(hstmt, (SQLCHAR *)"test", SQL_NTS, (SQLCHAR *)"test_schema",
 SQL_NTS, (SQLCHAR *)"test_table", SQL_NTS);
```

```
while (SQL_SUCCEEDED(retcode = SQLFetch(hstmt))) {
    for (i = 1; i <= columns; i++) {
        retcode = SQLGetData(hstmt, i, SQL_C_CHAR, buf, sizeof(buf), &indicator);
    }
}
```

示例：使用 ODBC SQLTables() 列出数据库和架构

ODBC API 未提供用于列出目录或架构的单独函数。相反，您可以使用 SQLTables() 的特殊调用约定来检索此信息。

列出所有数据库（目录）

在 CatalogName 设置为 SQL_ALL_CATALOGS 的情况下调用 SQLTables()。将 SchemaName 和 TableName 设置为空字符串。结果集仅在 TABLE_CAT 列中返回有效值。所有其它列都包含 NULL。

```
// List all catalogs (databases) available on the data source.
retcode = SQLTables(hstmt,
    (SQLCHAR *)SQL_ALL_CATALOGS, SQL_NTS, // CatalogName = "%" (SQL_ALL_CATALOGS)
    (SQLCHAR *)"", 0,                     // SchemaName = "" (empty string)
    (SQLCHAR *)"", 0,                     // TableName  = "" (empty string)
    NULL, 0);                             // TableType  = NULL (not filtered)
```

列出所有架构

在 SchemaName 设置为 SQL_ALL_SCHEMAS 的情况下调用 SQLTables()。将 CatalogName 和 TableName 设置为空字符串。

```
// List all schemas available on the data source.
retcode = SQLTables(hstmt,
    (SQLCHAR *)"", 0,                     // CatalogName = "" (empty string)
    (SQLCHAR *)SQL_ALL_SCHEMAS, SQL_NTS,  // SchemaName = "%" (SQL_ALL_SCHEMAS)
    (SQLCHAR *)"", 0,                     // TableName  = "" (empty string)
    NULL, 0);                             // TableType  = NULL (not filtered)
```

> **Note**
>
> ODBC 规范仅将 TABLE_SCHEM 定义为对架构枚举有效。Amazon Redshift 还会填充 TABLE_CAT，因为它支持跨数据库元数据发现，而且每个架构的范围都限于特定的数据库。

## 使用 SHOW 命令

如果驱动程序元数据 API 未涵盖您的使用案例，则可以使用 Amazon Redshift SHOW 命令来检索元数据。SHOW 命令经过优化，可快速检索元数据。当您在交互式 SQL 会话中需要元数据时，或者当您的应用程序通过不公开驱动程序元数据 API 的客户端进行连接时，它们很有用。

常见的发现操作支持以下 SHOW 命令：

- SHOW DATABASES
- SHOW SCHEMAS
- SHOW TABLES
- SHOW COLUMNS
- SHOW CONSTRAINTS
- SHOW GRANTS
- SHOW FUNCTIONS
- SHOW PROCEDURES
- SHOW PARAMETERS

示例

```
SHOW SCHEMAS FROM DATABASE dev;

database_name |     schema_name      | schema_owner | schema_type |         schema_acl
         | source_database | schema_option
--------------+----------------------+--------------+-------------
+-----------------------------+-----------------+--------------
dev           | pg_automv            |            1 | local       |
         |                 |
dev           | pg_catalog           |            1 | local       | jpuser=UC/
jpuser~=U/jpuser  |                 |
dev           | public               |            1 | local       | jpuser=UC/
jpuser~=UC/jpuser |                 |
dev           | information_schema   |            1 | local       | jpuser=UC/
jpuser~=U/jpuser  |                 |
dev           | schemad79cd6d93bf043 |            1 | local       |
         |                 |
```

```
SHOW TABLES FROM SCHEMA dev.s1 LIKE '%view' LIMIT 1;
```

```
database_name | schema_name |    table_name     | table_type |              table_acl
             | remarks | owner | last_altered_time | last_modified_time | dist_style |
  table_subtype
---------------+-------------+-------------------+------------
+-------------------------------------+---------+-------+------------------
+--------------------+------------+------------------
dev            | s1          | late_binding_view | VIEW       | {alice=arwdRxtDPA/
alice,bob=d/alice} |         | alice |                   |                    |
    | LATE BINDING VIEW
```

```
SHOW COLUMNS FROM TABLE second_db.public.t22;

 database_name | schema_name | table_name | column_name | ordinal_position |
 column_default | is_nullable |          data_type          | character_maximum_length
 | numeric_precision | numeric_scale | remarks | sort_key_type | sort_key | dist_key |
 encoding | collation
---------------+-------------+------------+-------------+------------------
+----------------+-------------+-----------------------------
+-------------------------+------------------+--------------+---------
+---------------+----------+----------+----------+-----------
 second_db     | public      | t22        | col1        |                1 |
     | YES         | integer                     |                           |
      32 |             0 |         | INTERLEAVED   |       -1 |          | mostly8  |
 second_db     | public      | t22        | col2        |                2 |
      | YES         | character varying           |                      100 |
          |               |         | INTERLEAVED   |        2 |          | text255  |
 default
 second_db     | public      | t22        | col3        |                3 |
     | YES         | timestamp without time zone |                           |
        |               |         |               |        0 |          | raw      |
 second_db     | public      | t22        | col4        |                4 |
     | YES         | numeric                     |                           |
      10 |             2 |         |               |        0 |          | az64     |
```

有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 SHOW。

## 查询系统表

如果驱动程序元数据 API 和 SHOW 命令都无法满足您的需要，则可以直接查询 PostgreSQL 目录表 (pg_*)、信息架构 (information_schema) 或 Amazon Redshift 系统视图（例如 SVV_TABLE_INFO）。

这种方法对于通过交互式 SQL 查询运行的临时探索、调试或管理任务非常有用，例如在 Amazon Redshift 查询编辑器 V2 或 psql 中。但是，系统目录结构可能会在 Amazon Redshift 版本之间发生变化。我们不建议您依赖直接系统表查询作为生产应用程序集成的元数据接口。

查询系统表时，请遵循以下实践：

- 指定数据库、架构和表。为了获得最佳性能，请在查询筛选条件中包含数据库名称、架构名称和表名称。缩小范围可以减少 Amazon Redshift 需要扫描和返回的元数据量。
- 使用等号或 LIKE 筛选条件。在 WHERE 子句中使用等号 (=) 或 LIKE 谓词来缩小结果范围。避免对大型目录视图进行未筛选的查询。在包含许多对象的集群上，此类查询可能会很慢。
- 避免在筛选条件谓词中使用函数。筛选系统表查询时，请勿在 WHERE 子句中使用诸如 LOWER( ) 或字符串连接之类的函数。这会妨碍进行高效的元数据查找，并可能显著降低查询性能。

反模式与推荐查询示例

```
-- Avoid:
SELECT * FROM svv_all_columns;

-- Recommended:
SELECT * FROM svv_all_columns WHERE database_name = 'dev' and schema_name = 'public'
 AND table_name = 'sales';
```

```
-- Avoid:
SELECT * FROM svv_all_columns WHERE LOWER(schema_name) = 'public';

-- Recommended:
SELECT * FROM svv_all_columns WHERE schema_name = 'public' AND table_name LIKE 'sales
%';
```

```
-- Avoid:
SELECT * FROM svv_all_columns WHERE database_name || '.' || schema_name || '.' ||
 table_name = 'dev.public.sales';

-- Recommended:
SELECT * FROM svv_all_columns WHERE database_name = 'dev' and schema_name = 'public'
 AND table_name = 'sales';
```

```
-- Avoid:
```

```
SELECT * FROM svv_all_columns WHERE CONCAT(database_name, '.', schema_name) =
 'dev.public';

-- Recommended:
SELECT * FROM svv_all_columns WHERE database_name = 'dev' and schema_name = 'public';
```

## 设置应用程序名称连接属性

当您连接到 Amazon Redshift 时，请设置可选的应用程序名称驱动程序属性。此属性可确定每个连接是由哪个应用程序或工具建立的，从而可以更轻松地监控、审计您的 Amazon Redshift 环境并对其进行故障排除。

您提供的应用程序名称显示在 SYS_CONNECTION_LOG 的 application_name 列中。您可以使用它来执行以下操作：

- 跟踪连接来源。确定哪些应用程序、服务或工具连接到您的集群。
- 性能故障排除。将查询与原始应用程序关联以隔离性能问题。
- 审计访问权限。了解数据仓库中的应用程序级使用模式。

下表显示了如何为每个 Amazon Redshift 驱动程序设置应用程序名称。

| 驱动程序 | 属性 | 示例 |
| --- | --- | --- |
| JDBC | ApplicationName | jdbc:redshift:// *endpoint*:*port*/*databas* ApplicationName=my-etl-pipeline |
| ODBC | ApplicationName | 在 DSN 或连接字符串中设置 ApplicationName=my-bi-tool。 |
| Python | application_name | redshift_connector.connect(..., application_name='my-data-catalog') |

有关每个驱动程序的应用程序名称属性的更多信息，请参阅以下内容：

- JDBC：ApplicationName
- ODBC：ApplicationName
- Python：application_name

> **Note**
>
> 对应用程序名称使用描述性、一致的命名约定，如 *team*-*service*-*environment*（例如 `quicksight-analytics-dashboard-prod`）。采用这种方法，可以更轻松地筛选和分析 `SYS_CONNECTION_LOG` 数据。

## 配置连接的安全选项

Amazon Redshift 支持安全套接字层 (SSL) 连接来加密数据和服务器证书，以验证客户端连接到的服务器证书。

### SSL

为了支持 SSL 连接，Amazon Redshift 会在每个集群中创建并安装 AWS Certificate Manager (ACM) 颁发的 SSL 证书。ACM 证书受到大多数操作系统、Web 浏览器和客户端的公开信任。如果您的 SQL 客户端或应用程序使用 SSL 连接到 Amazon Redshift，并且 `sslmode` 连接选项设置为 `require`、`verify-ca` 或 `verify-full`，则您可能需要下载证书捆绑包。如果您的客户需要，Amazon Redshift 会提供如下捆绑证书：

> **Note**
>
> 我们建议尽可能使用 `verify-full` 进行 SSL 连接。

- 捆绑包下载地址：https://s3.amazonaws.com/redshift-downloads/amazon-trust-ca-bundle.crt。
  - 预期的 MD5 校验码为 418dea9b6d5d5d5de7a8f1ac42e164cdcf。
  - sha256 校验码为 36dba8e4b8041cd14b9d601593963301bcbb92e1c456847784de2acb5bd550。

  不要使用位于 `https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt` 的之前的证书捆绑包。

- 在中国 AWS 区域，请从以下网址下载捆绑包：https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/amazon-trust-ca-bundle.crt。
  - 预期的 MD5 校验码为 418dea9b6d5d5d5de7a8f1ac42e164cdcf。
  - sha256 校验码是 36dba8e4b8041cd14b9d601593963301bcbb92e1c456847784de2acb5bd550。

  不要使用位于 `https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/redshift-ca-bundle.crt` 和 `https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/redshift-ssl-ca-cert.pem` 的以前的证书捆绑包

> **Important**
>
> Amazon Redshift 改变了我们管理 SSL 证书的方法。您可能需要更新当前的信任根 CA 证书，才能继续使用 SSL 连接集群。有关更多信息，请参阅 将 SSL 连接过渡到 ACM 证书。

默认情况下，无论连接是否使用 SSL，集群数据库都会接受该连接。要将您的集群配置为需要使用 SSL 连接，请在与集群关联的参数组中将 `require_SSL` 参数设置为 `true`。

Amazon Redshift 支持符合联邦信息处理标准 (FIPS) 140-2 的 SSL 模式。符合 FIPS 的 SSL 模式在默认情况下处于禁用状态。

> **Important**
>
> 仅在您的系统需要与 FIPS 兼容时才启用与 FIPS 兼容的 SSL 模式。

要启用符合 FIPS 的 SSL 模式，请在与 Amazon Redshift 集群或 Redshift Serverless 工作组关联的参数组中，将 `use_fips_ssl` 参数和 `require_SSL` 参数设置为 `true`。有关修改集群上的参数组的信息，请参阅 Amazon Redshift 参数组。有关修改工作组上的参数组的信息，请参阅配置与 Amazon Redshift Serverless 的符合 FIPS 的 SSL 连接。

Amazon Redshift 支持 Elliptic Curve Diffie—Hellman Ephemeral (ECDHE) 密钥协商协议。ECDHE 使得客户端和服务器各有一个椭圆曲线公有/私有密钥对，用来在不安全的通道上创建共享密钥。您无需在 Amazon Redshift 中进行任何配置即可启用 ECDHE。如果您从使用 ECDHE 来加密客户端与服务器间通信的 SQL 客户端工具进行连接，则 Amazon Redshift 将使用提供的密码列表来建立合适的连接。有关更多信息，请参阅 Wikipedia 上的 Elliptic curve diffie—hellman 和 OpenSSL 网站上的密码。

## ODBC 中的 SSL 和信任 CA 证书

如果您使用最新 Amazon Redshift ODBC 驱动程序 (1.3.7.1000 版或更高版本) 连接，则可以跳过此部分。要下载最新驱动程序，请参阅为 Amazon Redshift 配置 ODBC 驱动程序版本 2.x 连接。

您可能需要更新当前的信任根 CA 证书，才能继续使用 SSL 连接集群。有关更多信息，请参阅 SSL。

您可以验证下载的证书是否与此预期的 MD5 校验和码匹配。为此，您可以在 Linux 操作系统上使用 Md5sum 程序，或者在 Windows 和 macOS 操作系统上使用其它工具。

ODBC DSN 中的 `sslmode` 设置用于确定如何处理客户端连接加密和服务器证书验证。Amazon Redshift 支持客户端连接中的以下 `sslmode` 值：

- `disable`

  SSL 处于禁用状态，且连接未加密。
- `allow`

  如果服务器需要，则使用 SSL。
- `prefer`

  如果服务器支持，则使用 SSL。Amazon Redshift 支持 SSL，因此您可以在将 `sslmode` 设置为 `prefer` 时使用 SSL。
- `require`

  需要使用 SSL。
- `verify-ca`

  必须使用 SSL，且必须验证服务器证书。
- `verify-full`

  必须使用 SSL。必须验证服务器证书，且服务器主机名必须与证书上的主机名属性一致。

您可以确定是否使用了 SSL 以及是否在客户端和服务器之间的连接中验证了服务器证书。为此，您需要查看客户端上 ODBC DSN 的 `sslmode` 设置以及服务器上 Amazon Redshift 集群的 `require_SSL` 设置。下表介绍了各种客户端与服务器设置组合的加密结果：

| sslmode（客户端） | require_SSL（服务器） | 结果 |
| --- | --- | --- |
| disable | false | 连接未加密。 |
| disable | true | 由于服务器需要使用 SSL，但客户端针对该连接禁用了 SSL，因此无法建立连接。 |
| allow | true | 连接经过加密。 |
| allow | false | 连接未加密。 |
| prefer 或者 require | true | 连接经过加密。 |
| prefer 或者 require | false | 连接经过加密。 |
| verify-ca | true | 连接经过加密，且验证了服务器证书。 |
| verify-ca | false | 连接经过加密，且验证了服务器证书。 |
| verify-full | true | 连接经过加密，且验证了服务器证书和主机名。 |
| verify-full | false | 连接经过加密，且验证了服务器证书和主机名。 |

在 Microsoft Windows 上使用服务器证书和 ODBC 进行连接

如果要使用 SSL 和服务器证书连接到集群，请先将证书下载到客户端计算机或 Amazon EC2 实例。然后配置 ODBC DSN。

1. 将 Amazon Redshift 证书颁发机构包下载到客户端计算机上的驱动程序安装目录中的 `lib` 文件夹，然后将该文件另存为 `root.crt`。有关下载信息，请参阅SSL。

2. 打开 ODBC Data Source Administrator，然后添加或编辑 ODBC 连接的系统 DSN 条目。对于 SSL Mode，如果不使用 DNS 别名，请选择 `verify-full`。如果您使用 DNS 别名，请选择 `verify-ca`。然后选择保存。

   有关配置 ODBC DSN 的更多信息，请参阅为 Amazon Redshift 配置 ODBC 驱动程序版本 2.x 连接。

## Java 中的 SSL 和服务器证书

SSL 通过加密可在您的客户端与集群之间移动的数据来提供一个安全层。使用服务器证书可验证集群是否为 Amazon Redshift 集群，从而提供一个额外的安全层。具体方法是，检查自动安装在您预配置的所有集群上的服务器证书。有关将服务器证书 JDBC 一起使用的更多信息，请转到 PostgreSQL 文档中的配置客户端。

在 Java 中使用信任 CA 证书以进行连接

> **Important**
>
> Amazon Redshift 改变了我们管理 SSL 证书的方法。您可能需要更新当前的信任根 CA 证书，才能继续使用 SSL 连接集群。有关更多信息，请参阅 SSL。

使用信任 CA 证书进行连接

您可以使用 `redshift-keytool.jar` 文件将 Amazon Redshift 证书颁发机构包中的 CA 证书导入 Java 信任存储库或您的私有信任存储库中。

1. 如果您使用 Java 命令行 `-Djavax.net.ssl.trustStore` 选项，请尽量将其从命令行中删除。
2. 下载 redshift-keytool.jar。
3. 请执行下列操作之一：

   - 要将 Amazon Redshift 证书颁发机构捆绑包导入 Java 信任存储库中，请运行以下命令。

     ```
     java -jar redshift-keytool.jar -s
     ```

   - 要将 Amazon Redshift 证书颁发机构捆绑包导入您的私有信任存储库中，请运行以下命令：

     ```
     java -jar redshift-keytool.jar -k <your_private_trust_store> -
     p <keystore_password>
     ```

# 将 SSL 连接过渡到 ACM 证书

Amazon Redshift 正将您的集群中的 SSL 证书替换为 AWS Certificate Manager (ACM) 颁发的证书。ACM 是一个可信的公有证书颁发机构 (CA)，受当前大多数系统信任。您可能需要更新当前的信任根 CA 证书，才能继续使用 SSL 连接集群。

仅当满足以下所有条件时，此更改才会影响到您：

- 您的 SQL 客户端或应用程序使用 SSL 连接 Amazon Redshift 集群，并将 `sslMode` 连接选项设置为 `require`、`verify-ca` 或 `verify-full` 配置选项。
- 您不使用 Amazon Redshift ODBC 或 JDBC 驱动程序，或者您使用的是 ODBC 版本 1.3.7.1000 或 JDBC 版本 1.2.8.1005 之前的 Amazon Redshift 驱动程序。

如果此更改影响到您的 Amazon Redshift 商业区域，那么您必须在 2017 年 10 月 23 日之前更新当前的信任根 CA 证书。Amazon Redshift 会在现在至 2017 年 10 月 23 日之间将您的集群过渡为使用 ACM 证书。该更改对您集群的性能或可用性应该影响不大或者没有任何影响。

如果此更改对 AWS GovCloud (US)（美国）区域产生影响，则必须在 2020 年 4 月 1 日之前更新当前的信任根 CA 证书，以避免服务中断。从该日期开始，使用 SSL 加密连接来连接到 Amazon Redshift 集群的客户端需要额外的受信任的证书颁发机构 (CA)。当客户端连接到 Amazon Redshift 集群时，将使用受信任的证书颁发机构来确认该集群的身份。您需要执行相应操作来更新 SQL 客户端和应用程序，以使用包含新的受信任 CA 的更新的证书包。

> **Important**
>
> 2021 年 1 月 5 日，在中国区域，Amazon Redshift 正在将您的集群中的 SSL 证书替换为 AWS Certificate Manager (ACM) 颁发的证书。如果此更改对中国（北京）区域或中国（宁夏）区域产生影响，则必须在 2021 年 1 月 5 日之前更新当前的信任根 CA 证书，以避免服务中断。从该日期开始，使用 SSL 加密连接来连接到 Amazon Redshift 集群的客户端需要额外的受信任的证书颁发机构 (CA)。当客户端连接到 Amazon Redshift 集群时，将使用受信任的证书颁发机构来确认该集群的身份。您需要执行相应操作来更新 SQL 客户端和应用程序，以使用包含新的受信任 CA 的更新的证书包。

- 使用最新的 Amazon Redshift ODBC 或 JDBC 驱动程序
- 使用较早的 Amazon Redshift ODBC 或 JDBC 驱动程序
- 使用其他 SSL 连接类型

使用最新的 Amazon Redshift ODBC 或 JDBC 驱动程序

首选方法是使用最新的 Amazon Redshift ODBC 或 JDBC 驱动程序。从 ODBC 版本 1.3.7.1000 和 JDBC 版本 1.2.8.1005 开始的 Amazon Redshift 驱动程序可自动管理从 Amazon Redshift 自签名证书到 ACM 证书的过渡。要下载最新驱动程序，请参阅为 Amazon Redshift 配置 JDBC 驱动程序版本 2.x 连接。

如果您使用的是最新的 Amazon Redshift JDBC 驱动程序，最好不要使用 JVM 选项中的 `-Djavax.net.ssl.trustStore`。如果必须使用 `-Djavax.net.ssl.trustStore`，请将 Redshift 证书颁发机构包导入到它指向的信任存储库中。有关下载信息，请参阅SSL。有关更多信息，请参阅将 Amazon Redshift 证书颁发机构捆绑包导入信任存储库。

使用较早的 Amazon Redshift ODBC 或 JDBC 驱动程序

- 如果您的 ODBC DSN 配置了 `SSLCertPath`，请覆盖指定路径中的证书文件。
- 如果未设置 `SSLCertPath`，则请覆盖驱动程序 DLL 位置中名为 `root.crt` 的证书文件。

如果必须使用版本 1.2.8.1005 之前的 Amazon Redshift JDBC 驱动程序，请执行以下操作之一：

- 如果您的 JDBC 连接字符串使用 `sslCert` 选项，请删除 `sslCert` 选项。然后，将 Redshift 证书颁发机构包导入到 Java 信任存储库中。有关下载信息，请参阅SSL。有关更多信息，请参阅 将 Amazon Redshift 证书颁发机构捆绑包导入信任存储库。
- 如果您使用 Java 命令行 `-Djavax.net.ssl.trustStore` 选项，请尽量将其从命令行中删除。然后，将 Redshift 证书颁发机构包导入到 Java 信任存储库中。有关下载信息，请参阅SSL。有关更多信息，请参阅 将 Amazon Redshift 证书颁发机构捆绑包导入信任存储库。

将 Amazon Redshift 证书颁发机构捆绑包导入信任存储库

您可以使用 `redshift-keytool.jar` 将 Amazon Redshift 证书颁发机构包中的 CA 证书导入 Java 信任存储库或您的私有信任存储库中。

要将 Amazon Redshift 证书颁发机构捆绑包导入信任存储库

1. 下载 redshift-keytool.jar。
2. 请执行下列操作之一：
    - 要将 Amazon Redshift 证书颁发机构捆绑包导入 Java 信任存储库中，请运行以下命令。

```
java -jar redshift-keytool.jar -s
```

- 要将 Amazon Redshift 证书颁发机构捆绑包导入您的私有信任存储库中，请运行以下命令：

```
java -jar redshift-keytool.jar -k <your_private_trust_store> -
p <keystore_password>
```

使用其他 SSL 连接类型

如果您使用以下任何方式进行连接，请执行本部分中的步骤：

- 开源 ODBC 驱动程序
- 开源 JDBC 驱动程序
- Amazon Redshift RSQL 命令行界面
- 基于 libpq 的任何语言绑定，如 psycopg2 (Python) 和 ruby-pg (Ruby)

将 ACM 证书与其他 SSL 连接类型结合使用：

1. 下载 Amazon Redshift 证书颁发机构捆绑包。有关下载信息，请参阅SSL。
2. 将包中的证书放入您的 `root.crt` 文件中。
   - 在 Linux 和 macOS 操作系统中，该文件为 `~/.postgresql/root.crt`。
   - 在 Microsoft Windows 上，该文件为 `%APPDATA%\postgresql\root.crt`。

## 通过客户端工具和代码连接

Amazon Redshift 提供 Amazon Redshift 查询编辑器 v2，用于连接到您的集群和工作组。有关更多信息，请参阅 使用查询编辑器 v2 查询数据库。

此部分提供了一些用于第三方工具连接的选项。另外，此部分还介绍了如何以编程方式连接到您的集群。

主题

- 使用 Amazon Redshift RSQL 连接
- 使用 Amazon Redshift RSQL 连接到集群

- Amazon Redshift RSQL 元命令
- Amazon Redshift RSQL 变量
- Amazon Redshift RSQL 错误代码
- Amazon Redshift RSQL 环境变量

## 使用 Amazon Redshift RSQL 连接

Amazon Redshift RSQL 是一个命令行客户端，用于与 Amazon Redshift 集群和数据库进行交互。您可以连接到 Amazon Redshift 集群、描述数据库对象、查询数据以及查看各种输出格式的查询结果。

Amazon Redshift RSQL 支持 PostgreSQL psql 命令行工具的功能以及特定于 Amazon Redshift 的额外功能。这些功能包括：

- 您可以使用 AD FS、PingIdentity、Okta、Azure ADm 或其它基于 SAML/JWT 的身份提供者实现单点登录身份验证。您还可以使用基于浏览器的 SAML 身份提供者进行多重身份验证（MFA）。
- 您可以描述 Amazon Redshift 对象的特性或属性，例如表分发键、表排序键、后期绑定视图（LBV）和实体化视图。您还可以描述 AWS Glue 目录或 Apache Hive Metastore 中外部表的特性或属性、Amazon RDS for PostgreSQL、Amazon Aurora PostgreSQL 兼容版、RDS for MySQL（预览版）和 Amazon Aurora MySQL 兼容版（预览版）中的外部数据库，以及使用 Amazon Redshift 数据共享共享的表。
- 您还可以使用增强的控制流命令，例如 `IF`（`\ELSEIF`、`\ELSE`, `\ENDIF`）`\GOTO` 和 `\LABEL`。

借助 Amazon Redshift RSQL 批处理模式（运行作为输入参数传递的脚本），您可以运行包含 SQL 和复杂业务逻辑的脚本。如果您有现有的自我管理的本地数据仓库，您可以使用 Amazon Redshift RSQL 替换现有的提取、转换、加载（ETL）和自动化脚本，例如 Teradata BTEQ 脚本。使用 RSQL 有助于避免以过程语言手动重新实现脚本。

Amazon Redshift RSQL 适用于 Linux、Windows 和 macOS 操作系统。

如需报告 Amazon Redshift RSQL 的问题，请发送邮件至 `<redshift-rsql-support@amazon.com>`。

主题

- Amazon Redshift RSQL 入门
- Amazon Redshift RSQL 更改日志

## Amazon Redshift RSQL 入门

在具有 Linux、macOS 或微软 Windows 操作系统的计算机上安装 Amazon Redshift RSQL。

### 下载 RSQL

- Linux 64 位 RPM：RSQL 版本 1.1.2
  - Linux 构件签名密钥：密钥
  - Linux 构件签名哈希：哈希
- Mac OS 64 位 PKG：RSQL 版本 1.1.2
- Windows 64 位 MSI：RSQL 版本 1.1.2

请在 Amazon Redshift RSQL 更改日志中参阅以前版本的更改日志和下载。

### 安装适用于 Linux 的 RSQL

请按照以下步骤，安装适用于 Linux 的 RSQL。

1. 使用以下命令安装驱动程序：

```
sudo yum install unixODBC
```

2. 安装 ODBC 驱动程序：下载并安装 Amazon Redshift ODBC 驱动程序。
3. 将 ini 文件复制到您的主目录：

```
cp /opt/amazon/redshiftodbcx64/odbc.ini ~/.odbc.ini
```

4. 将环境变量设置为指向文件的位置：

```
export ODBCINI=~/.odbc.ini
export ODBCSYSINI=/opt/amazon/redshiftodbcx64/
export AMAZONREDSHIFTODBCINI=/opt/amazon/redshiftodbcx64/amazon.redshiftodbc.ini
```

5. 现在，您可以通过运行以下命令安装 RSQL。

```
sudo rpm -i AmazonRedshiftRsql-<version>.rhel.x86_64.rpm
```

### 安装适用于 Mac 的 RSQL

请按照以下步骤操作，安装适用于 Mac OSX 的 RSQL。

1. 使用以下命令安装驱动程序：

```
brew install unixodbc --build-from-source
```

2. 安装 ODBC 驱动程序：下载并安装 Amazon Redshift ODBC 驱动程序。
3. 将 ini 文件复制到您的主目录：

```
cp /opt/amazon/redshift/Setup/odbc.ini ~/.odbc.ini
```

4. 将环境变量设置为指向文件的位置：

```
export ODBCINI=~/.odbc.ini
export ODBCSYSINI=/opt/amazon/redshift/Setup
export AMAZONREDSHIFTODBCINI=/opt/amazon/redshift/lib/amazon.redshiftodbc.ini
```

5. 如果它不在/usr/local/lib 中，则将 DYLD_LIBRARY_PATH 设置为 libodbc.dylib 的位置。

```
export DYLD_LIBRARY_PATH=$DYLD_LIBRARY_PATH:/usr/local/lib
```

6. 双击 pkg 文件运行安装程序。
7. 按照安装向导中的步骤完成安装。同意许可协议条款。

### 安装适用于 Windows 的 RSQL

请按照以下步骤操作，安装适用于 Windows 的 RSQL。

1. 安装 ODBC 驱动程序：下载并安装 Amazon Redshift ODBC 驱动程序。
2. 请双击 RSQL 下载文件，以运行安装程序，然后按照提示完成安装。

### Amazon Redshift RSQL 更改日志

#### 1.1.2（2025-12-11）

**错误修复**

- 修复了一个错误，该错误会导致 \goto 和 \label 命令失败。

- 修复了一个错误，当变量被引号引起来时，该错误会导致 RSQL 无法打印变量值。
- 修复了一个错误，在启用 UseDeclareFetch 后，如果查询结果超过 ODBC DSN Fetch 参数大小，该错误会导致 RSQL 发生崩溃。
- 修复了一个问题，即便已启用分页程序，该问题也会导致一次性返回多个结果页。
- 修复了一个错误，在事务块内的查询失败时，该错误会导致 RSQL 发生崩溃。

1.1.1（2025-11-20）

错误修复

- 修复了一个问题，在使用 -c 标志时，该问题会导致 RSQL 错误地解析查询。此修复适用于所有平台。
- 修复了 Mac 上的一个错误，该错误导致用户无法在 RSQL 中使用 \s 命令。

1.1.0（2025-11-11）

错误修复

- 解决了导致 RSQL 中意外发生崩溃的内存泄漏问题。
- 从 RSQL 中移除了 OpenSSL 依赖项。
- 修复了同一环境中的 libpq/psql 安装存在的链接冲突问题。
- 提升了针对 Amazon Linux 2023、Windows 和 macOS 的平台兼容性。
- 修复了一个问题，当输出超出当前显示大小时，该问题会导致输出被截断。

1.0.8 (2023-06-19)

错误修复

- 修复了使用 SHOW 命令时会截断输出的问题。
- 为 \de 添加了用于描述外部 Kinesis 流和 Kafka 主题的支持。

1.0.7 (2023-03-22)

错误修复

- 修复了 RSQL 无法描述实体化视图的问题。

- 修复了使用 Amazon Redshift Serverless 时 stl_connection_log 上出现的权限被拒绝错误。
- 修复了 RSQL 可能不正确地处理 \GOTO 标签的问题。
- 修复了在静默模式下输出 SSL 消息的问题。
- 修复了描述存储过程时显示随机字符的问题。
- 修复了重复输出 ERROR/INFO 消息的问题。

New

- RSQL 现在可以直接从 ODBC 驱动程序获取 SSL 信息。

1.0.6 (2023-02-21)

错误修复

- 修复了在 Redshift 补丁 1.0.46086 (P173) 上，\d 引发错误（整数的输入语法无效："xid"）的问题。

New

- 已重命名安装文件来体现支持的架构。

1.0.5（2022 年 6 月 27 日）

错误修复

- 将 SQL 错误消息发送到标准错误 (stderr)。
- 修复了使用 ON_ERROR_STOP 时退出代码的问题。现在，脚本将在遇到错误后终止，并返回正确的退出代码。
- Maxerror 现在不区分大小写。

New

- 增加了对 ODBC 2.x 驱动程序的支持。

1.0.4 (2022-03-19)

- 添加对 RSPASSWORD 环境变量的支持。设置密码以连接到 Amazon Redshift。例如 `export RSPASSWORD=TestPassw0rd`。

1.0.3（2021 年 12 月 8 日）

错误修复

- 修复了在 Windows 操作系统中使用 `\c` 或 `\logon` 切换数据库时弹出的对话框。
- 修复了检查 ssl 信息时发生的崩溃。

Amazon Redshift RSQL 的以前版本

请根据您的操作系统，选择其中一个链接以下载所需的 Amazon Redshift RSQL 版本。

Linux 64 位 RPM

- RSQL 版本 1.1.1
- RSQL 版本 1.1.0
- RSQL 版本 1.0.8
- RSQL 版本 1.0.7
- RSQL 版本 1.0.6
- RSQL 版本 1.0.5
- RSQL 版本 1.0.4
- RSQL 版本 1.0.3
- RSQL 版本 1.0.1

Mac OS 64 位 DMG/PKG

- RSQL 版本 1.1.1
- RSQL 版本 1.1.0
- RSQL 版本 1.0.8
- RSQL 版本 1.0.7
- RSQL 版本 1.0.6
- RSQL 版本 1.0.5

- RSQL 版本 1.0.4
- RSQL 版本 1.0.3
- RSQL 版本 1.0.1

Windows 64 位 MSI

- RSQL 版本 1.1.1
- RSQL 版本 1.1.0
- RSQL 版本 1.0.8
- RSQL 版本 1.0.7
- RSQL 版本 1.0.6
- RSQL 版本 1.0.5
- RSQL 版本 1.0.4
- RSQL 版本 1.0.3
- RSQL 版本 1.0.1

## 使用 Amazon Redshift RSQL 连接到集群

使用 Amazon Redshift，您可以连接到集群并使用 RSQL 与之交互。这是一个命令行工具，提供了一种安全的方式来查询数据、创建数据库对象和管理 Amazon Redshift 集群。以下各节将指导您完成使用 RSQL（带或不带数据来源名称（DSN））与集群建立连接的步骤。

在没有 DSN 的情况下连接

1. 在 Amazon Redshift 控制台上，选择要连接到的集群，然后记下端点、数据库和端口。
2. 在命令提示符下，使用命令行参数指定连接信息。

   ```
   rsql -h <endpoint> -U <username> -d <databasename> -p <port>
   ```

   在这里，以下内容适用：

   - *<endpoint>* 是您在上一步中记录的端点。
   - *<userid>* 是有权连接到集群的用户。
   - *<databasename>* 是您在上一步中记录的数据库名称。
   - *<port>* 是您在上一步中记录的端口。*<port>* 是一个可选参数。

以下为示例。

```
rsql -h testcluster.example.amazonaws.com -U user1 -d dev -p 5439
```

3. 在密码提示符处，输入 *<username>* 用户的密码。

   成功的连接响应如下所示。

```
% rsql -h testcluster.example.com -d dev -U user1 -p 5349
Password for user user1:
DSN-less Connected
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
Redshift Version: 1.0.29306
Type "help" for help.

(testcluster) user1@dev=#
```

要连接的命令在 Linux、Mac OS 和 Windows 上具有相同的参数。

在使用 DSN 的情况下连接

可以使用 DSN 将 RSQL 连接到 Amazon Redshift，以简化连接属性的整理方式。本主题包括 ODBC 驱动程序安装说明和 DSN 属性的描述。

使用带密码的 DSN 连接

下面显示使用密码的 DSN 连接配置的示例。对于 Mac OSX 而言，默认 <path to driver> 为 /opt/amazon/redshift/lib/libamazonredshiftodbc.dylib，对于 Linux 而言，则为 /opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so。

```
[testuser]
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so
SSLMode=verify-ca
Min_TLS=1.2
boolsaschar=0
Host=<server endpoint>
Port=<database port>
```

```
Database=<dbname>
UID=<username>
PWD=<password>
sslmode=prefer
```

连接成功后，输出结果如下。

```
% rsql -D testuser
DSN Connected
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
Redshift Version: 1.0.29306
Type "help" for help.

(testcluster) user1@dev=#
```

使用单点登录 DSN

您可以配置 DSN 以进行单点登录身份验证。下面显示使用 Okta 单点登录的 DSN 连接配置的示例。

```
[testokta]
Driver=<path to driver>
SSLMode=verify-ca
Min_TLS=1.2
boolsaschar=0
Host=<server endpoint>
clusterid=<cluster id>
region=<region name>
Database=<dbname>
locale=en-US
iam=1
plugin_name=<plugin name>
uid=<okta username>
pwd=<okta password>
idp_host=<idp endpoint>
app_id=<app id>
app_name=<app name>
preferred_role=<role arn>
```

成功连接的输出示例。

```
% rsql -D testokta
DSN Connected
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
Redshift Version: 1.0.29306
Type "help" for help.

(testcluster) user1@dev=#
```

下面显示使用 Azure 单点登录的 DSN 连接配置的示例。

```
[testazure]
Driver=<path to driver>
SSLMode=verify-ca
Min_TLS=1.2
boolsaschar=0
Host=<server endpoint>
Port=<cluster port>
clusterid=<cluster id>
region=<region name>
Database=<dbname>
locale=en-us
iam=1
plugin_name=<plugin name>
uid=<azure username>
pwd=<azure password>
idp_tenant=<Azure idp tenant uuid>
client_id=<Azure idp client uuid>
client_secret=<Azure idp client secret>
```

将 DSN 连接与 IAM 配置文件结合使用

您可以使用配置的 IAM 配置文件连接到 Amazon Redshift。IAM 配置文件必须具有调用 `GetClusterCredentials` 的权限。以下示例演示了要使用的 DSN 属性。仅当 `Host` 不是 Amazon 提供的端点（如 `examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com`）时，才需要 `ClusterID` 和 `Region` 参数。

```
[testiam]
Driver=Default
Host=testcluster.example.com
```

```
Database=dev
DbUser=testuser
ClusterID=rsqltestcluster
Region=us-east-1
IAM=1
Profile=default
```

Profile 密钥的值是您从 AWS CLI 凭证中选择的命名配置文件。此示例显示名为 default 的配置文件的凭证。

```
$ cat .aws/credentials
[default]
aws_access_key_id = ASIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

下面显示的是连接响应。

```
$ rsql -D testiam
DSN Connected
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
Redshift Version: 1.0.29306
Type "help" for help.

(testcluster) testuser@dev=>
```

### 将 DSN 连接与实例配置文件结合使用

您可以使用 Amazon EC2 实例配置文件连接到 Amazon Redshift。实例配置文件必须具有调用 GetClusterCredentials 的权限。有关要使用的 DSN 属性，请参阅下面的示例。仅当 Host 不是 Amazon 提供的端点（如 examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com）时，才需要 ClusterID 和 Region 参数。

```
[testinstanceprofile]
Driver=Default
Host=testcluster.example.com
Database=dev
DbUser=testuser
ClusterID=rsqltestcluster
```

```
Region=us-east-1
IAM=1
Instanceprofile=1
```

下面显示的是连接响应。

```
$ rsql -D testinstanceprofile
DSN Connected
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
Redshift Version: 1.0.29306
Type "help" for help.

(testcluster) testuser@dev=>
```

将 DSN 连接与默认凭证提供程序链结合使用

要使用默认凭证提供程序链进行连接，请仅指定 IAM 属性，Amazon Redshift RSQL 将尝试按照适用于 Java 的 AWS SDK 中使用 AWS 凭证所述的顺序获取凭证。链中必须至少一个提供程序具有 `GetClusterCredentials` 权限。例如，这对于从 ECS 容器进行连接非常有用。

```
[iamcredentials]
Driver=Default
Host=testcluster.example.com
Database=dev
DbUser=testuser
ClusterID=rsqltestcluster
Region=us-east-1
IAM=1
```

## Amazon Redshift RSQL 元命令

Amazon Redshift RSQL 元命令返回有关数据库或特定数据库对象的信息性记录。结果可以包括各种列和元数据。其他命令会执行特定操作。这些命令的前面都有反斜杠。

\d[S+]

列出本地用户创建的表、常规视图、后期绑定视图和实体化视图。`\dS` 还列出了表和视图，如 `\d`，但系统对象包含在返回的记录中。`+` 为所有列出的对象生成额外的元数据列 `description`。下面显示的是作为命令返回的示例记录。

```
List of relations
 schema |   name    | type  |  owner
--------+-----------+-------+---------
 public | category  | table | awsuser
 public | date      | table | awsuser
 public | event     | table | awsuser
 public | listing   | table | awsuser
 public | sales     | table | awsuser
 public | users     | table | awsuser
 public | venue     | table | awsuser
(7 rows)
```

\d[S+] NAME

描述表、视图或索引。包括列名称和类型。它还提供了 diststyle、备份配置、创建日期（2018 年 10 月之后创建的表）和约束。例如，\dS+ sample 返回对象属性。附加 S+ 会导致返回的记录中包含其它列。

```
Table "public.sample"
 Column |            Type             |   Collation    | Nullable | Default Value |
 Encoding  | DistKey | SortKey
--------+-----------------------------+----------------+----------+---------------
+-----------+---------+---------
 col1   | smallint                    |                | NO       |               |
 none      | t       | 1
 col2   | character(100)              | case_sensitive | YES      |               |
 none      | f       | 2
 col3   | character varying(100)      | case_sensitive | YES      |               |
 text32k   | f       | 3
 col4   | timestamp without time zone |                | YES      |               |
 runlength | f       | 0
 col5   | super                       |                | YES      |               |
 zstd      | f       | 0
 col6   | bigint                      |                | YES      |               |
 az64      | f       | 0

Diststyle: KEY
Backup: YES
Created: 2021-07-20 19:47:27.997045
Unique Constraints:
    "sample_pkey" PRIMARY KEY (col1)
    "sample_col2_key" UNIQUE (col2)
Foreign-key constraints:
```

```
    "sample_col2_fkey" FOREIGN KEY (col2) REFERENCES lineitem(l_orderkey)
```

表的分布样式或 Diststyle 可以是 KEY、AUTO、EVEN 或 ALL。

备份指示在拍摄快照时是否备份了表。有效值为 YES 或 NO。

已创建是创建表的时间的时间戳。创建日期不适用于 2018 年 11 月之前创建的 Amazon Redshift 表。在此日期之前创建的表显示 n/a（不可用）。

独特约束列出了表中的唯一的主键约束。

外键约束列出了表中的外键越塾。

\dC[+] [PATTERN]

列出转换。包括源类型、目标类型以及转换是否为隐式。

以下显示来自 \dC+ 的结果子集：

```
List of casts
         source type         |         target type         |      function       |
 implicit?   | description
-----------------------------+-----------------------------+---------------------
+---------------+-------------
 "char"                      | character                   | bpchar              | in
 assignment |
 "char"                      | character varying           | text                | in
 assignment |
 "char"                      | integer                     | int4                | no
          |
 "char"                      | text                        | text                | yes
          |
 "path"                      | point                       | point               | no
          |
 "path"                      | polygon                     | polygon             | in
 assignment |
 abstime                     | date                        | date                | in
 assignment |
 abstime                     | integer                     | (binary coercible)  | no
          |
 abstime                     | time without time zone      | time                | in
 assignment |
```

```
abstime                     | timestamp with time zone    | timestamptz         | yes
         |
abstime                     | timestamp without time zone | timestamp           | yes
         |
bigint                      | bit                         | bit                 | no
         |
bigint                      | boolean                     | bool                | yes
         |
bigint                      | character                   | bpchar              | in
assignment |
bigint                      | character varying           | text                | in
assignment |
bigint                      | double precision            | float8              | yes
         |
bigint                      | integer                     | int4                | in
assignment |
bigint                      | numeric                     | numeric             | yes
         |
bigint                      | oid                         | oid                 | yes
         |
bigint                      | real                        | float4              | yes
         |
bigint                      | regclass                    | oid                 | yes
         |
bigint                      | regoper                     | oid                 | yes
         |
bigint                      | regoperator                 | oid                 | yes
         |
bigint                      | regproc                     | oid                 | yes
         |
bigint                      | regprocedure                | oid                 | yes
         |
bigint                      | regtype                     | oid                 | yes
         |
bigint                      | smallint                    | int2                | in
assignment |
bigint                      | super                       | int8_partiql        | in
assignment |
```

\dd[S] [PATTERN]

显示在别处未显示的对象描述。

\de

列出外部表。这包括 AWS Glue Data Catalog 中的表、Hive Metastore 和来自 Amazon RDS/Aurora MySQL、Amazon RDS/Aurora PostgreSQL 和 Amazon Redshift 数据共享表的联合表。

\de NAME

描述外部表。

以下示例显示 AWS Glue 外部表。

```
# \de spectrum.lineitem
                           Glue External table "spectrum.lineitem"
     Column      | External Type | Redshift Type | Position | Partition Key | Nullable
-----------------+---------------+---------------+----------+---------------+----------
 l_orderkey      | bigint        | bigint        | 1        | 0             |
 l_partkey       | bigint        | bigint        | 2        | 0             |
 l_suppkey       | int           | int           | 3        | 0             |
 l_linenumber    | int           | int           | 4        | 0             |
 l_quantity      | decimal(12,2) | decimal(12,2) | 5        | 0             |
 l_extendedprice | decimal(12,2) | decimal(12,2) | 6        | 0             |
 l_discount      | decimal(12,2) | decimal(12,2) | 7        | 0             |
 l_tax           | decimal(12,2) | decimal(12,2) | 8        | 0             |
 l_returnflag    | char(1)       | char(1)       | 9        | 0             |
 l_linestatus    | char(1)       | char(1)       | 10       | 0             |
 l_shipdate      | date          | date          | 11       | 0             |
 l_commitdate    | date          | date          | 12       | 0             |
 l_receiptdate   | date          | date          | 13       | 0             |
 l_shipinstruct  | char(25)      | char(25)      | 14       | 0             |
 l_shipmode      | char(10)      | char(10)      | 15       | 0             |
 l_comment       | varchar(44)   | varchar(44)   | 16       | 0             |

Location: s3://redshiftbucket/kfhose2019/12/31
Input_format: org.apache.hadoop.mapred.TextInputFormat
Output_format: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Serialization_lib: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
Serde_parameters: {"field.delim":"|","serialization.format":"|"}
Parameters:
 {"EXTERNAL":"TRUE","numRows":"178196721475","transient_lastDdlTime":"1577771873"}
```

Hive Metastore 表。

```
# \de emr.lineitem
```

```
                    Hive Metastore External Table "emr.lineitem"
     Column      | External Type | Redshift Type | Position | Partition Key | Nullable
-----------------+---------------+---------------+----------+---------------+----------
 l_orderkey      | bigint        | bigint        | 1        | 0             |
 l_partkey       | bigint        | bigint        | 2        | 0             |
 l_suppkey       | int           | int           | 3        | 0             |
 l_linenumber    | int           | int           | 4        | 0             |
 l_quantity      | decimal(12,2) | decimal(12,2) | 5        | 0             |
 l_extendedprice | decimal(12,2) | decimal(12,2) | 6        | 0             |
 l_discount      | decimal(12,2) | decimal(12,2) | 7        | 0             |
 l_tax           | decimal(12,2) | decimal(12,2) | 8        | 0             |
 l_returnflag    | char(1)       | char(1)       | 9        | 0             |
 l_linestatus    | char(1)       | char(1)       | 10       | 0             |
 l_commitdate    | date          | date          | 11       | 0             |
 l_receiptdate   | date          | date          | 12       | 0             |
 l_shipinstruct  | char(25)      | char(25)      | 13       | 0             |
 l_shipmode      | char(10)      | char(10)      | 14       | 0             |
 l_comment       | varchar(44)   | varchar(44)   | 15       | 0             |
 l_shipdate      | date          | date          | 16       | 1             |

Location: s3://redshiftbucket/cetas
Input_format: org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat
Output_format: org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat
Serialization_lib: org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe
Serde_parameters: {"serialization.format":"1"}
Parameters: {"EXTERNAL":"TRUE", "numRows":"4307207",
 "transient_lastDdlTime":"1626990007"}
```

PostgreSQL 外部表。

```
# \de pgrsql.alltypes
                                Postgres Federated Table "pgrsql.alltypes"
 Column |        External Type        |        Redshift Type        | Position |
 Partition Key | Nullable
--------+-----------------------------+-----------------------------+----------
+---------------+----------
 col1   | bigint                      | bigint                      | 1        | 0
        |
 col2   | bigint                      | bigint                      | 2        | 0
        |
 col5   | boolean                     | boolean                     | 3        | 0
        |
```

```
col6   | box                        | varchar(65535)             | 4        | 0
       |
col7   | bytea                      | varchar(65535)             | 5        | 0
       |
col8   | character(10)              | character(10)              | 6        | 0
       |
col9   | character varying(10)      | character varying(10)      | 7        | 0
       |
col10  | cidr                       | varchar(65535)             | 8        | 0
       |
col11  | circle                     | varchar(65535)             | 9        | 0
       |
col12  | date                       | date                       | 10       | 0
       |
col13  | double precision           | double precision           | 11       | 0
       |
col14  | inet                       | varchar(65535)             | 12       | 0
       |
col15  | integer                    | integer                    | 13       | 0
       |
col16  | interval                   | varchar(65535)             | 14       | 0
       |
col17  | json                       | varchar(65535)             | 15       | 0
       |
col18  | jsonb                      | varchar(65535)             | 16       | 0
       |
col19  | line                       | varchar(65535)             | 17       | 0
       |
col20  | lseg                       | varchar(65535)             | 18       | 0
       |
col21  | macaddr                    | varchar(65535)             | 19       | 0
       |
col22  | macaddr8                   | varchar(65535)             | 20       | 0
       |
col23  | money                      | varchar(65535)             | 21       | 0
       |
col24  | numeric                    | numeric(38,20)             | 22       | 0
       |
col25  | path                       | varchar(65535)             | 23       | 0
       |
col26  | pg_lsn                     | varchar(65535)             | 24       | 0
       |
col28  | point                      | varchar(65535)             | 25       | 0
       |
```

```
 col29  | polygon                     | varchar(65535)              | 26       | 0
        |
 col30  | real                        | real                        | 27       | 0
        |
 col31  | smallint                    | smallint                    | 28       | 0
        |
 col32  | smallint                    | smallint                    | 29       | 0
        |
 col33  | integer                     | integer                     | 30       | 0
        |
 col34  | text                        | varchar(65535)              | 31       | 0
        |
 col35  | time without time zone      | varchar(65535)              | 32       | 0
        |
 col36  | time with time zone         | varchar(65535)              | 33       | 0
        |
 col37  | timestamp without time zone | timestamp without time zone | 34       | 0
        |
 col38  | timestamp with time zone    | timestamp with time zone    | 35       | 0
        |
 col39  | tsquery                     | varchar(65535)              | 36       | 0
        |
 col40  | tsvector                    | varchar(65535)              | 37       | 0
        |
 col41  | txid_snapshot               | varchar(65535)              | 38       | 0
        |
 col42  | uuid                        | varchar(65535)              | 39       | 0
        |
 col43  | xml                         | varchar(65535)              | 40       | 0
        |
```

\df[anptw][S+] [PATTERN]

列出各种类型的函数。例如，命令 \df 返回函数列表。结果包括名称、返回的数据类型、访问权限和其它元数据等属性。函数类型可以包括触发器、存储过程、窗口函数和其它类型。当您将 S+ 附加到命令（例如 \dfantS+）时，会包含其它元数据列，例如 owner、security 和 access privileges。

\dL[S+] [PATTERN]

列出与数据库关联的过程语言的数据。信息包括名称（例如 plpgsql）和其它元数据，其中包括名称是否受信任、访问权限和描述。例如，示例调用是 \dLS+，它列出了语言及其属性。当您将 S+ 附加到命令时，会包含其它元数据列，例如 `call handler` 和 `access privileges`。

示例结果：

```
List of languages
   name    | trusted | internal language |      call handler       |
      validator                          | access privileges |          description

-----------+---------+-------------------+-------------------------
+-------------------------------------------------------------+-------------------
+--------------------------------
 c         | f       | t                 | -                       |
 fmgr_c_validator(oid)                                         |                   |
 Dynamically-loaded C functions
 exfunc    | f       | f                 | exfunc_call_handler()   | -
                                         | rdsdb=U/rdsdb     |
 internal  | f       | t                 | -                       |
 fmgr_internal_validator(oid)                                  |                   |
 Built-in functions
 mlfunc    | f       | f                 | mlfunc_call_handler()   | -
                                         | rdsdb=U/rdsdb     |
 plpgsql   | t       | f                 | plpgsql_call_handler()  |
 plpgsql_validator(oid)                                        |                   |
 plpythonu | f       | f                 | plpython_call_handler() |
 plpython_compiler(cstring,cstring,cstring,cstring,cstring) | rdsdb=U/rdsdb     |
 sql       | t       | t                 | -                       |
 fmgr_sql_validator(oid)                                       | =U/rdsdb          | SQL-
language functions
```

\dm[S+] [PATTERN]

列出实体化视图。例如，\dmS+ 列出了实体化视图及其属性。当您将 S+ 附加到命令时，会包含其它元数据列。

\dn[S+] [PATTERN]

列出架构。当您将 S+ 附加到命令（例如 \dnS+）时，会包含其它元数据列，例如 description 和 access privileges。

\dp [PATTERN]

列出表、视图和序列访问权限。

\dt[S+] [PATTERN]

列出表 当您将 S+ 附加到命令（例如 \dtS+）时，会包含其它元数据列，例如本例中的 description。

\du

列出数据库的用户。包括他们的姓名和角色，例如超级用户和属性。

\dv[S+] [PATTERN]

列出视图。包括架构、类型和拥有者数据。当您将 S+ 附加到命令（例如 \dvS+）时，会包含其它元数据列。

\H

打开 HTML 输出。这对于快速返回格式化的结果非常有用。例如，select * from sales; \H 以 HTML 格式返回销售表中的结果。要切换回表格结果，请使用 \q，或者无提示。

\i

从文件中运行命令。例如，假设您的工作目录中有 rsql_steps.sql，以下将运行文件中的命令：\i rsql_steps.sql。

\l[+] [PATTERN]

列出数据库。包括所有者、编码和其它信息。

\q

退出，或者 \q 命令，注销数据库会话并退出 RSQL。

\sv[+] VIEWNAME

显示视图的定义。

\timing

例如，显示查询的运行时间。

\z [PATTERN]

输出与\ dp 相同。

\?

显示帮助信息。可选的参数指定要解释的项目。

\EXIT

注销所有数据库会话并退出 Amazon Redshift RSQL。此外，您可以指定可选的退出代码。例如，\EXIT 15 将退出 Amazon Redshift RSQL 终端并返回退出代码 15。

以下示例显示来自连接的输出并从 RSQL 退出。

```
% rsql -D testuser
DSN Connected
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.34.1000
Rsql Version: 1.0.1
Redshift Version: 1.0.29306
Type "help" for help.

(testcluster) user1@dev=# \exit 15

% echo $?
15
```

\EXPORT

指定导出文件的名称，RSQL 将该文件用于存储后续 SQL SELECT 语句返回的数据库信息。

export_01.sql

```
\export report file='E:\\accounts.out'
\rset rformat off
\rset width 1500
\rset heading "General Title"
\rset titledashes on
```

```
select * from td_dwh.accounts;
\export reset
```

控制台输出

```
Rformat is off.
Target width is 1500.
Heading is set to: General Title
Titledashes is on.
(exported 40 rows)
```

\LOGON

连接到数据库。您可以使用位置语法指定连接参数或将其指定作为连接字符串。

命令语法如下所示：\logon {[DBNAME|- USERNAME|- HOST|- PORT|- [PASSWORD]] | conninfo}

DBNAME 是要连接的数据库的名称。USERNAME 是连接的用户名。默认 HOST 为 localhost。默认 PORT 为 5439。

当在 \LOGON 命令中指定主机名时，它将成为其它 \LOGON 命令的默认主机名。要更改默认主机名，请在附加 \LOGON 命令中指定新的 HOST。

user1 的 \LOGON 命令输出示例如下。

```
(testcluster) user1@redshiftdb=# \logon dev
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
You are now connected to database "dev" as user "user1".
(testcluster) user1@dev=#
```

user2 的示例输出。

```
(testcluster) user1@dev=# \logon dev user2 testcluster2.example.com
Password for user user2:
DBMS Name: Amazon Redshift
Driver Name: Amazon Redshift ODBC Driver
Driver Version: 1.4.27.1000
Rsql Version: 1.0.1
```

```
You are now connected to database "dev" as user "user2" on host
 "testcluster2.example.com" at port "5439".
(testcluster2) user2@dev=#
```

\REMARK

\echo 命令的扩展。\REMARK 将指定的字符串打印到输出流。\REMARK 通过添加在单独的行上中断输出的能力来扩展 \echo。

以下示例显示了命令的输出。

```
(testcluster) user1@dev=# \remark 'hello//world'
hello
world
```

\RSET

该 \rset 命令将设置命令参数和变量。\rset 还具有交互式模式和批处理模式。它不支持 bash 选项（例如 -x）或参数（例如 --<arg>）。

它设置如下所示的变量：

- ERRORLEVEL
- HEADING 和 RTITLE
- RFORMAT
- MAXERROR
- TITLEDASHES
- WIDTH

以下示例指定一个标题。

```
\rset heading "Winter Sales Report"
```

有关如何使用 \rset 的更多示例，您可以在 Amazon Redshift RSQL 变量 主题中查找示例。

\RUN

运行包含在指定文件中的 Amazon Redshift RSQL 脚本。\RUN 通过添加一个选项来跳过文件中的标题行以扩展 \i 命令。

如果文件名包含逗号、分号或空格，请用单引号将其括起来。此外，如果文本在文件名后面，请将其括在引号中。在 UNIX 中，文件名区分大小写。在 Windows 中，文件名不区分大小写。

以下示例显示了命令的输出。

```
(testcluster) user1@dev=# \! cat test.sql
select count(*) as lineitem_cnt from lineitem;
select count(*) as customer_cnt from customer;
select count(*) as orders_cnt from orders;



(testcluster) user1@dev=# \run file=test.sql
 lineitem_cnt
--------------
      4307207
(1 row)

 customer_cnt
--------------
     37796166
(1 row)

 orders_cnt
------------
          0
(1 row)


(testcluster) user1@dev=# \run file=test.sql skip=2
2 records skipped in RUN file.
 orders_cnt
------------
          0
(1 row)
```

\OS

\! 命令的别名。\OS 运行作为参数传递的操作系统命令。命令运行后，控制将返回到 Amazon Redshift RSQL。例如，您可以运行以下命令以打印当前系统日期时间并返回到 RSQL 终端：\os date。

```
(testcluster) user1@dev=# \os date
```

```
Tue Sep 7 20:47:54 UTC 2021
```

\GOTO

适用于 Amazon Redshift RSQL 的新命令。\GOTO 跳过所有干预命令并在指定的 \LABEL 继续处理。\LABEL 必须是前向引用。您不能跳转到词汇上位于 \GOTO 之前的 \LABEL。

下面显示了示例输出。

```
(testcluster) user1@dev=# \! cat test.sql
select count(*) as cnt from lineitem \gset
select :cnt as cnt;
\if :cnt > 100
    \goto LABELB
\endif

\label LABELA
\remark 'this is label LABELA'
\label LABELB
\remark 'this is label LABELB'


(testcluster) user1@dev=# \i test.sql
   cnt
---------
 4307207
(1 row)

\label LABELA ignored
\label LABELB processed
this is label LABELB
```

\LABEL

适用于 Amazon Redshift RSQL 的新命令。\LABEL 建立运行程序的入口点，作为 \GOTO 命令的目标。

以下显示了命令的输出。

```
(testcluster) user1@dev=# \! cat test.sql
select count(*) from lineitem limit 5;
\goto LABELB
```

```
\remark "this step was skipped by goto label";
\label LABELA
\remark 'this is label LABELA'
\label LABELB
\remark 'this is label LABELB'



(testcluster) user1@dev=# \i testgoto.sql
  count
 4307193
(1 row)

\label LABELA ignored
\label LABELB processed
this is label LABELB
```

### \IF (\ELSEIF, \ELSE, \ENDIF)

\IF 和相关命令有条件地运行部分输入脚本。PSQL \if ( \elif、\else、\endif ) 命令的扩展。\IF 和 \ELSEIF 支持布尔表达式，包括 AND、OR 和 NOT 条件。

以下显示了命令的示例输出。

```
(testcluster) user1@dev=# \! cat test.sql
SELECT query FROM stv_inflight LIMIT 1 \gset
select :query as query;
\if :query > 1000000
    \remark 'Query id is greater than 1000000'
\elseif :query = 1000000
    \remark 'Query id is equal than 1000000'
\else
    \remark 'Query id is less than 1000000'
\endif


(testcluster) user1@dev=# \i test.sql
 query
--------
 994803
(1 row)

Query id is less than 1000000
```

在您的分支逻辑中使用 ERRORCODE。

```
\if :'ERRORCODE' = '00000'
    \remark 'The statement was executed without error'
\else
    \remark :LAST_ERROR_MESSAGE
\endif
```

在 \IF 块内使用 \GOTO 以控制代码的运行方式。

## Amazon Redshift RSQL 变量

有些关键字在 RSQL 中充当变量。您可以将每个值设置为特定值，也可以重新设置值。大多数值都由 \rset 设置，它具有交互式模式和批处理模式。命令可以用小写或大写定义。

### ACTIVITYCOUNT

指示受上次提交的请求影响的行数。对于数据返回请求，这是从数据库返回到 RSQL 的行数。该值为 0 或正整数。最大值为 18,446,744,073,709,551,615。

经过特殊处理的变量 ACTIVITYCOUNT 类似于变量 ROW_COUNT。但是，在命令完成时，ROW_COUNT 不会向客户端应用程序报告受影响的 SELECT、COPY 或 UNLOAD 行数。而 ACTIVITYCOUNT 会报告相关行数。

activitycount_01.sql：

```
select viewname, schemaname
from pg_views
where schemaname = 'not_existing_schema';
\if :ACTIVITYCOUNT = 0
\remark 'views do not exist'
\endif
```

控制台输出：

```
viewname | schemaname
----------+------------
(0 rows)

views do not exist
```

ERRORLEVEL

为错误分配严重性级别。使用严重性级别来确定适当操作。如果尚未使用 ERRORLEVEL 命令，那么默认情况下，它的值为 ON。

errorlevel_01.sql：

```
\rset errorlevel 42P01 severity 0

select * from tbl;

select 1 as col;

\echo exit
\quit
```

控制台输出：

```
Errorlevel is on.
rsql: ERROR: relation "tbl" does not exist
(1 row)

col
1

exit
```

HEADING 和 RTITLE

允许用户指定显示在报告顶部的标题。使用 RSET RTITLE 命令指定的标题会自动包含客户端电脑的当前系统日期。

rset_heading_rtitle_02.rsql 内容：

```
\remark Starting...
\rset rtitle "Marketing Department||Confidential//Third Quarter//Chicago"
\rset width 70
\rset rformat on
select * from rsql_test.tbl_currency order by id limit 2;
\exit
\remark Finishing...
```

控制台输出：

```
Starting...
Rtitle is set to: &DATE||Marketing Department||Confidential//Third Quarter//Chicago
 (Changes will take effect after RFORMAT is
switched ON)
Target width is 70.
Rformat is on.
09/11/20       Marketing       Department Confidential
                  Third Quarter
                     Chicago
id  | bankid  | name |      start_date
100 |       1 | USD | 2020-09-11 10:51:39.106905
110 |       1 | EUR | 2020-09-11 10:51:39.106905
(2 rows)

Press any key to continue . . .
```

MAXERROR

指定 RSQL 终止任务处理的最大错误严重性级别。在完成每个作业或任务后，返回代码是 RSQL 返回到客户端操作系统的整数值。返回代码的值表示作业或任务的完成状态。如果脚本包含生成错误严重性级别大于指定 maxerror 值的语句，RSQL 将立即退出。因此，要将 RSQL 退出的错误严重性级别设为 8，请使用 RSET MAXERROR 7。

maxerror_01.sql 内容：

```
\rset maxerror 0

select 1 as col;

\quit
```

控制台输出：

```
Maxerror is default.
(1 row)

col
1
```

RFORMAT

允许用户指定是否应用格式化命令的设置。

rset_rformat.rsql 内容：

```
\remark Starting...
\pset border 2
\pset format wrapped
\pset expanded on
\pset title 'Great Title'
select * from rsql_test.tbl_long where id = 500;
\rset rformat
select * from rsql_test.tbl_long where id = 500;
\rset rformat off
select * from rsql_test.tbl_long where id = 500;
\rset rformat on
select * from rsql_test.tbl_long where id = 500;
\exit
\remark Finishing...
```

控制台输出：

```
Starting...
Border style is 2. (Changes will take effect after RFORMAT is switched ON)
Output format is wrapped. (Changes will take effect after RFORMAT is switched ON)
Expanded display is on. (Changes will take effect after RFORMAT is switched ON)
Title is "Great Title". (Changes will take effect after RFORMAT is switched ON)
id  |                                                                long_string
500 | In general, the higher the number the more borders and lines the tables will
 have, but details depend on the particular
format.
(1 row)

Rformat is on.
Great Title
+-[ RECORD
 1 ]+-------------------------------------------------------------------------------------------
-----------+
| id          | 500
|
| long_string | In general, the higher the number the more borders and lines the tables
 will have, but details depend on the
```

```
particular format. |
+-------------
+-----------------------------------------------------------------------------------------------
-----------+

Rformat is off.
id  |                                                                  long_string
500 | In general, the higher the number the more borders and lines the tables will
 have, but details depend on the particular format.
(1 row)

Rformat is on.
Great Title
+-[ RECORD
 1 ]+-----------------------------------------------------------------------------------------------
-----------+
| id          | 500
|
| long_string | In general, the higher the number the more borders and lines the tables
 will have, but details depend on the
particular format. |
+-------------
+-----------------------------------------------------------------------------------------------
-----------+
Press any key to continue . . .
```

### ROW_COUNT

获取受以前查询影响的记录数。它通常用于检查结果，如以下代码片段所示：

```
SET result = ROW_COUNT;

IF result = 0
...
```

### TITLEDASHES

使用此控件，用户可以指定是否要在为 SQL 语句返回的列数据的上方打印一行破折号字符。

示例：

```
\rset titledashes on
select dept_no, emp_no, salary from rsql_test.EMPLOYEE
```

```
where dept_no = 100;
\rset titledashes off
select dept_no, emp_no, salary from rsql_test.EMPLOYEE
where dept_no = 100;
```

控制台输出：

```
dept_no      emp_no          salary
------------ ------------ --------------------
100          1000346         1300.00
100          1000245         5000.00
100          1000262         2450.00

dept_no     emp_no          salary
100          1000346         1300.00
100          1000245         5000.00
100          1000262         2450.00
```

WIDTH

将输出格式设置为换行并指定报告中每行的目标宽度。如果没有参数，它将返回格式和目标宽度的当前设置。

rset_width_01.rsql 内容：

```
\echo Starting...
\rset width
\rset width 50
\rset width
\quit
\echo Finishing...
```

控制台输出：

```
Starting...
Target width is 75.
Target width is 50.
Target width is 50.
Press any key to continue . . .
```

使用参数的示例：

```
\echo Starting...
\rset rformat on
\pset format wrapped
select * from rsql_test.tbl_long where id = 500;
\rset width 50
select * from rsql_test.tbl_long where id = 500;
\quit
\echo Finishing...
```

控制台输出：

```
Starting...
Rformat is on.
Output format is wrapped.
id  |                                          long_string
500 | In general, the higher the number the more borders and lines the ta.
    |.bles will have, but details depend on the particular format.
(1 row)

Target width is 50.
id  |                                          long_string
500 | In general, the higher the number the more.
    |. borders and lines the tables will have, b.
    |.ut details depend on the particular format.
    |..
(1 row)
Press any key to continue . . .
```

## Amazon Redshift RSQL 错误代码

成功消息、警告和异常：

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 00000 | 00 类：成功完成 | successful_completion |
| 01000 | 01 类：警告 | warning |
| 0100C | 01 类：警告 | dynamic_result_sets_returned |
| 01008 | 01 类：警告 | implicit_zero_bit_padding |

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 01003 | 01 类：警告 | null_value_eliminated_in_set_function |
| 01007 | 01 类：警告 | privilege_not_granted |
| 01006 | 01 类：警告 | privilege_not_revoked |
| 01004 | 01 类：警告 | string_data_right_truncation |
| 01P01 | 01 类：警告 | deprecated_feature |
| 02000 | 02 类：无数据 | no_data |
| 02001 | 02 类：无数据 | no_additional_dynamic_result_sets_returned |
| 03000 | 03 类：SQL 语句尚未结束 | sql_statement_not_yet_complete |
| 08000 | 08 类：连接异常 | connection_exception |
| 08003 | 08 类：连接异常 | connection_does_not_exist |
| 08006 | 08 类：连接异常 | connection_failure |
| 08001 | 08 类：连接异常 | sqlclient_unable_to_establish_sqlconnection |
| 08004 | 08 类：连接异常 | sqlserver_rejected_establishment_of_sqlconnection |
| 08007 | 08 类：连接异常 | transaction_resolution_unknown |
| 08P01 | 08 类：连接异常 | protocol_violation |
| 09000 | 09 类：已触发的操作异常 | triggered_action_exception |
| 0A000 | 0A 类：功能不受支持 | feature_not_supported |
| 0A000 | 0A 类：功能不受支持 | feature_not_supported |
| 0B000 | 0B 类：事务启动无效 | invalid_transaction_initiation |
| 0F000 | 0F 类：定位器异常 | locator_exception |

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 0F001 | 0F 类：定位器异常 | invalid_locator_specification |
| 0L000 | 0L 类：授予者无效 | invalid_grantor |
| 0LP01 | 0L 类：授予者无效 | invalid_grant_operation |
| 0P000 | 0P 类：角色规范无效 | invalid_role_specification |
| 0Z000 | 0Z 类：诊断异常 | diagnostics_exception |
| 0Z002 | 0Z 类：诊断异常 | stacked_diagnostics_accessed_without_active_handler |
| 20000 | 20 类：未找到案例 | case_not_found |
| 21000 | 21 类：基数违规 | cardinality_violation |

数据异常：

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 22000 | 22 类：数据异常 | data_exception |
| 2202E | 22 类：数据异常 | array_subscript_error |
| 22021 | 22 类：数据异常 | character_not_in_repertoire |
| 22008 | 22 类：数据异常 | datetime_field_overflow |
| 22012 | 22 类：数据异常 | division_by_zero |
| 22005 | 01 类：警告 | error_in_assignment |
| 2200B | 01 类：警告 | escape_character_conflict |
| 22022 | 01 类：警告 | indicator_overflow |
| 22015 | 01 类：警告 | interval_field_overflow |

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 2201E | 01 类：警告 | invalid_argument_for_logarithm |
| 2201F | 01 类：警告 | invalid_argument_for_power_function |
| 2201G | 01 类：警告 | invalid_argument_for_width_bucket_function |
| 22018 | 01 类：警告 | invalid_character_value_for_cast |
| 22007 | 01 类：警告 | invalid_datetime_format |
| 22019 | 01 类：警告 | invalid_escape_character |
| 2200D | 01 类：警告 | invalid_escape_octet |
| 22025 | 01 类：警告 | invalid_escape_sequence |
| 22P06 | 01 类：警告 | nonstandard_use_of_escape_character |
| 22010 | 01 类：警告 | invalid_indicator_parameter_value |
| 22023 | 01 类：警告 | invalid_parameter_value |
| 2201B | 01 类：警告 | invalid_regular_expression |
| 22009 | 01 类：警告 | invalid_time_zone_displacement_value |
| 2200C | 01 类：警告 | invalid_use_of_escape_character |
| 2200G | 01 类：警告 | most_specific_type_mismatch |
| 22004 | 01 类：警告 | null_value_not_allowed |
| 22002 | 01 类：警告 | null_value_no_indicator_parameter |
| 22003 | 01 类：警告 | numeric_value_out_of_range |
| 22026 | 01 类：警告 | string_data_length_mismatch |
| 22001 | 01 类：警告 | string_data_right_truncation |

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 22011 | 01 类：警告 | substring_error |
| 22027 | 01 类：警告 | trim_error |
| 22024 | 01 类：警告 | unterminated_c_string |
| 2200F | 01 类：警告 | zero_length_character_string |
| 22P01 | 01 类：警告 | floating_point_exception |
| 22P02 | 01 类：警告 | invalid_text_representation |
| 22P03 | 01 类：警告 | invalid_binary_representation |
| 22P04 | 01 类：警告 | bad_copy_file_format |
| 22P05 | 01 类：警告 | untranslatable_character |

违反完整性约束：

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 23000 | 23 类：违反完整性约束 | integrity_constraint_violation |
| 23001 | 23 类：违反完整性约束 | restrict_violation |
| 23502 | 23 类：违反完整性约束 | not_null_violation |
| 23503 | 23 类：违反完整性约束 | foreign_key_violation |
| 23505 | 23 类：违反完整性约束 | unique_violation |
| 23514 | 23 类：违反完整性约束 | check_violation |
| 24000 | 24 类：游标状态无效 | invalid_cursor_state |
| 01004 | 01 类：警告 | string_data_right_truncation |

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 25000 | 25 类：事务状态无效 | invalid_transaction_state |
| 25001 | 25 类：事务状态无效 | active_sql_transaction |
| 25002 | 25 类：事务状态无效 | invalid_transaction_state |
| 25008 | 25 类：事务状态无效 | held_cursor_requires_same_isolation_level |
| 25003 | 25 类：事务状态无效 | inappropriate_access_mode_for_branch_transaction |
| 25004 | 25 类：事务状态无效 | inappropriate_isolation_level_for_branch_transaction |
| 25005 | 25 类：事务状态无效 | no_active_sql_transaction_for_branch_transaction |
| 25006 | 25 类：事务状态无效 | read_only_sql_transaction |
| 25007 | 25 类：事务状态无效 | no_active_sql_transaction_for_branch_transaction |
| 25P01 | 25 类：事务状态无效 | no_active_sql_transaction |
| 25P02 | 25 类：事务状态无效 | in_failed_sql_transaction |
| 26000 | 26 类：SQL 语句名称无效 | invalid_sql_statement_name |
| 28000 | 28 类：授权规范无效 | invalid_authorization_specification |
| 2B000 | 2B 类：关联权限描述符依然存在 | dependent_privilege_descriptors_still_exist |
| 2BP01 | 2B 类：关联权限描述符依然存在 | dependent_objects_still_exist |
| 2D000 | 2D 类：事务终止无效 | invalid_transaction_termination |
| 2F000 | 2F 类：SQL 例程异常 | sql_routine_exception |

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 2F005 | 2F 类：SQL 例程异常 | function_executed_no_return_statement |
| 2F002 | 2F 类：SQL 例程异常 | modifying_sql_data_not_permitted |
| 2F003 | 2F 类：SQL 例程异常 | prohibited_sql_statement_attempted |
| 2F004 | 2F 类：SQL 例程异常 | reading_sql_data_not_permitted |
| 34000 | 34 类：游标名称无效 | invalid_cursor_name |
| 38000 | 38 类：外部例程异常 | external_routine_exception |
| 38001 | 38 类：外部例程异常 | containing_sql_not_permitted |
| 38002 | 38 类：外部例程异常 | modifying_sql_data_not_permitted |
| 38003 | 38 类：外部例程异常 | prohibited_sql_statement_attempted |
| 38004 | 38 类：外部例程异常 | reading_sql_data_not_permitted |
| 39000 | 39 类：外部例程调用异常 | external_routine_invocation_exception |
| 39001 | 39 类：外部例程调用异常 | invalid_sqlstate_returned |
| 39004 | 39 类：外部例程调用异常 | null_value_not_allowed |
| 39P01 | 39 类：外部例程调用异常 | trigger_protocol_violated |
| 39P02 | 39 类：外部例程调用异常 | srf_protocol_violated |
| 3D000 | 3D 类：目录名称无效 | invalid_catalog_name |
| 3F000 | 3F 类：架构名称无效 | invalid_schema_name |
| 42000 | 42 类：语法错误或访问规则冲突 | syntax_error_or_access_rule_violation |
| 42601 | 42 类：语法错误或访问规则冲突 | syntax_error |

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 42501 | 42 类：语法错误或访问规则冲突 | insufficient_privilege |
| 42846 | 42 类：语法错误或访问规则冲突 | cannot_coerce |
| 42803 | 42 类：语法错误或访问规则冲突 | grouping_error |
| 42830 | 42 类：语法错误或访问规则冲突 | invalid_foreign_key |
| 42602 | 42 类：语法错误或访问规则冲突 | invalid_name |
| 42622 | 42 类：语法错误或访问规则冲突 | name_too_long |
| 42939 | 42 类：语法错误或访问规则冲突 | reserved_name |
| 42804 | 42 类：语法错误或访问规则冲突 | datatype_mismatch |
| 42P18 | 42 类：语法错误或访问规则冲突 | indeterminate_datatype |
| 42809 | 42 类：语法错误或访问规则冲突 | wrong_object_type |
| 42703 | 42 类：语法错误或访问规则冲突 | undefined_column |
| 42883 | 42 类：语法错误或访问规则冲突 | undefined_function |
| 42P01 | 42 类：语法错误或访问规则冲突 | undefined_table |

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 42P02 | 42 类：语法错误或访问规则冲突 | undefined_parameter |
| 42704 | 42 类：语法错误或访问规则冲突 | undefined_object |
| 42701 | 42 类：语法错误或访问规则冲突 | duplicate_column |
| 42P03 | 42 类：语法错误或访问规则冲突 | duplicate_cursor |
| 42P04 | 42 类：语法错误或访问规则冲突 | duplicate_database |
| 42723 | 42 类：语法错误或访问规则冲突 | duplicate_function |
| 42P05 | 42 类：语法错误或访问规则冲突 | duplicate_prepared_statement |
| 42P06 | 42 类：语法错误或访问规则冲突 | duplicate_schema |
| 42P07 | 42 类：语法错误或访问规则冲突 | duplicate_table |
| 42712 | 42 类：语法错误或访问规则冲突 | duplicate_alias |
| 42710 | 42 类：语法错误或访问规则冲突 | duplicate_object |
| 42702 | 42 类：语法错误或访问规则冲突 | ambiguous_column |
| 42725 | 42 类：语法错误或访问规则冲突 | ambiguous_function |

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 42P08 | 42 类：语法错误或访问规则冲突 | ambiguous_parameter |
| 42P09 | 42 类：语法错误或访问规则冲突 | ambiguous_alias |
| 42P10 | 42 类：语法错误或访问规则冲突 | invalid_column_reference |
| 42611 | 42 类：语法错误或访问规则冲突 | invalid_column_definition |
| 42P11 | 42 类：语法错误或访问规则冲突 | invalid_cursor_definition |
| 42P12 | 42 类：语法错误或访问规则冲突 | invalid_database_definition |
| 42P13 | 42 类：语法错误或访问规则冲突 | invalid_function_definition |
| 42P14 | 42 类：语法错误或访问规则冲突 | invalid_prepared_statement_definition |
| 42P15 | 42 类：语法错误或访问规则冲突 | invalid_schema_definition |
| 42P16 | 42 类：语法错误或访问规则冲突 | invalid_table_definition |
| 42P17 | 42 类：语法错误或访问规则冲突 | invalid_object_definition |
| 44000 | 44 类：WITH CHECK OPTION 违规 | with_check_option_violation |
| 53000 | 53 类：资源不足 | insufficient_resources |

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 53100 | 53 类：资源不足 | disk_full |
| 53200 | 53 类：资源不足 | out_of_memory |
| 53300 | 53 类：资源不足 | too_many_connections |
| 54000 | 54 类：已超出程序限制 | program_limit_exceeded |
| 54001 | 54 类：已超出程序限制 | statement_too_complex |
| 54011 | 54 类：已超出程序限制 | too_many_columns |
| 54023 | 54 类：已超出程序限制 | too_many_arguments |
| 55000 | 55 类：对象不在先决条件状态 | object_not_in_prerequisite_state |
| 55006 | 55 类：对象不在先决条件状态 | object_in_use |
| 55P02 | 55 类：对象不在先决条件状态 | cant_change_runtime_param |
| 55P03 | 55 类：对象不在先决条件状态 | lock_not_available |
| 57000 | 57 类：操作员干预 | operator_intervention |
| 57014 | 57 类：操作员干预 | query_canceled |
| 57P01 | 57 类：操作员干预 | admin_shutdown |
| 57P02 | 57 类：操作员干预 | crash_shutdown |
| 57P03 | 57 类：操作员干预 | cannot_connect_now |
| 58000 | 58 类：系统错误（PostgreSQL 外部的错误） | system_error |
| 58030 | 58 类：系统错误（PostgreSQL 外部的错误） | io_error |
| 58P01 | 58 类：系统错误（PostgreSQL 外部的错误） | undefined_file |

| 错误代码 | 错误类 | 条件名称 |
| --- | --- | --- |
| 58P02 | 58 类：系统错误（PostgreSQL 外部的错误） | duplicate_file |
| F0000 | F0 类：配置文件错误 | duplicate_file |
| F0001 | F0 类：配置文件错误 | lock_file_exists |
| P0000 | P0 类：PL/pgSQL 错误 | plpgsql_error |
| P0001 | P0 类：PL/pgSQL 错误 | raise_exception |
| P0002 | P0 类：PL/pgSQL 错误 | no_data_found |
| P0003 | P0 类：PL/pgSQL 错误 | too_many_rows |
| XX000 | XX 类：内部错误 | internal_error |
| XX001 | XX 类：内部错误 | data_corrupted |
| XX002 | XX 类：内部错误 | index_corrupted |

## Amazon Redshift RSQL 环境变量

Amazon Redshift RSQL 可以使用环境变量来选择默认参数值。

RSPASSWORD

> **Important**
>
> 出于安全原因，我们不建议使用此环境变量，因为有些操作系统允许非管理用户查看进程环境变量。

为 Amazon Redshift RSQL 设置连接到 Amazon Redshift 时要使用的密码。此环境变量需要 Amazon Redshift RSQL 1.0.4 及更高版本。

如果设置了 RSPASSWORD，RSQL 会优先考虑它。如果未设置 RSPASSWORD 并且您使用 DSN 进行连接，则 RSQL 将从 DSN 文件的参数中获取密码。最后，如果未设置 RSPASSWORD 且您没有使用 DSN，则 RSQL 会在尝试连接后提供密码提示。

下面为设置 RSPASSWORD 的示例：

```
export RSPASSWORD=TestPassw0rd
```

# 使用身份验证配置文件连接到 Amazon Redshift

如果您与 Amazon Redshift 有很多连接，则可能很难管理所有这些连接的设置。通常，每个 JDBC 或 ODBC 连接都使用特定的配置选项。借助身份验证配置文件，可以将连接选项存储在一起。这样，用户可以选择要连接的配置文件，避免管理单个选项的设置。配置文件可以应用于各种场景和用户类型。

创建身份验证配置文件后，用户可以将即用型配置文件添加到连接字符串。这样，他们即可使用针对每个角色和使用案例的正确设置连接到 Amazon Redshift。

有关 Amazon Redshift API 的信息，请参阅 CreateAuthenticationProfile。

## 创建身份验证配置文件

通过 AWS CLI，您可以使用 `create-authentication-profile` 命令创建身份验证配置文件。这假定您已有一个 Amazon Redshift 集群和现有数据库。您的凭证必须具有连接到 Amazon Redshift 数据库的权限以及获取身份验证配置文件的权限。您可以将配置选项作为 JSON 字符串提供，或者引用包含 JSON 字符串的文件。

```
create-authentication-profile --authentication-profile-name<value: String> --
authentication-profile-content<value: String>
```

以下示例创建一个名为 `ExampleProfileName` 的配置文件。在这里，您可以将定义集群名称和其它选项设置的键和值作为 JSON 字符串添加。

```
create-authentication-profile --authentication-profile-name "ExampleProfileName"
 --authentication-profile-content "{\"AllowDBUserOverride\":\"1\",\"Client_ID
\":\"ExampleClientID\",\"App_ID\":\"ExampleAppID\",\"AutoCreate\":false,
\"enableFetchRingBuffer\":true,\"databaseMetadataCurrentDbOnly\":true}"
}
```

此命令使用指定的 JSON 设置创建配置文件。返回以下内容，表示已创建配置文件。

```
{"AuthenticationProfileName": "ExampleProfileName",
"AuthenticationProfileContent": "{\"AllowDBUserOverride\":\"1\",
\"Client_ID\":\"ExampleClientID\",\"App_ID\":\"ExampleAppID\",
\"AutoCreate\":false,\"enableFetchRingBuffer\":true,
\"databaseMetadataCurrentDbOnly\":true}" }
```

创建身份验证配置文件的限制和配额

每位客户的配额为十（10）个身份验证配置文件。

身份验证配置文件可能发生某些错误 例如，如果您使用现有名称创建新的配置文件，或者您是否超过了配置文件配额。有关更多信息，请参阅 CreateAuthenticationProfile。

您无法在身份验证配置文件存储中存储 JDBC、ODBC 和 Python 连接字符串的某些选项键和值：

- AccessKeyID
- access_key_id
- SecretAccessKey
- secret_access_key_id
- PWD
- Password
- password

对于 JDBC 或 ODBC 连接字符串，您不能在配置文件存储中存储键或值 AuthProfile。对于 Python 连接，您无法存储 auth_profile。

身份验证配置文件存储在 Amazon DynamoDB 中并由 AWS 管理。

## 使用身份验证配置文件进行连接

创建身份验证配置文件后，您可以包含配置文件名称作为 JDBC 版本 2.x AuthProfile 的连接选项。使用此连接选项可以检索存储的设置。

```
jdbc:redshift:iam://endpoint:port/database?AuthProfile=<Profile-
Name>&AccessKeyID=<Caller-Access-Key>&SecretAccessKey=<Caller-Secret-Key>
```

以下是 JDBC URL 字符串示例。

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?
AuthProfile="ExampleProfile"&AccessKeyID="AKIAIOSFODNN7EXAMPLE"&SecretAccessKey="wJalrXUtnFEMI/
K7MDENG/bPxRfiCYEXAMPLEKEY"
```

在 JDBC URL 中指定 AccessKeyID 和 SecretAccessKey，以及身份验证配置文件名称。

在下面的示例中，您还可以使用分号分隔符分隔配置选项，其中包括日志记录选项。

```
jdbc:redshift:iam://my_redshift_end_point:5439/dev?LogLevel=6;LogPath=/
tmp;AuthProfile=my_profile;AccessKeyID="AKIAIOSFODNN7EXAMPLE";SecretAccessKey="wJalrXUtnFEMI/
K7MDENG/bPxRfiCYEXAMPLEKEY"
```

> **Note**
>
> 请勿向身份验证配置文件中添加机密信息。例如，不要在身份验证配置文件中存储 AccessKeyID 或 SecretAccessKey 值。身份验证配置文件存储具有禁止存储密钥的规则。如果尝试存储与敏感信息相关的密钥和值，则会出错。

## 获取身份验证配置文件

要列出现有身份验证配置文件，请调用以下命令。

```
describe-authentication-profiles --authentication-profile-name <value: String>
```

下面的示例显示了两个检索配置文件。如果不指定配置文件名称，则返回所有配置文件。

```
{ "AuthenticationProfiles": [ { "AuthenticationProfileName":
"testProfile1", "AuthenticationProfileContent": "{\"AllowDBUserOverride
\":\"1\",\"Client_ID\":\"ExampleClientID\",\"App_ID\":\"ExampleAppID
\",\"AutoCreate\":false,\"enableFetchRingBuffer\":true,
\"databaseMetadataCurrentDbOnly\":true}" }, { "AuthenticationProfileName":
"testProfile2", "AuthenticationProfileContent": "{\"AllowDBUserOverride
\":\"1\",\"Client_ID\":\"ExampleClientID\",\"App_ID\":\"ExampleAppID
\",\"AutoCreate\":false,\"enableFetchRingBuffer\":true,
\"databaseMetadataCurrentDbOnly\":true}" } ] }
```

# 解决 Amazon Redshift 中的连接问题

如果您在从 SQL 客户端工具连接到集群时遇到问题，可以从几个方面进行排查以缩小问题范围。如果您使用了 SSL 或服务器证书，请在开始排查连接问题之前将其删除以降低复杂性。然后，在找出解决方案之后再添加。有关更多信息，请参阅 配置连接的安全选项。

有关 Amazon Redshift 功能中可能影响应用程序的行为更改的信息，请参阅 Amazon Redshift 中的行为更改。

> **Important**
>
> Amazon Redshift 改变了我们管理 SSL 证书的方法。如果您在使用 SSL 连接时遇到问题，您可能需要更新当前的信任根 CA 证书。有关更多信息，请参阅 将 SSL 连接过渡到 ACM 证书。

以下部分提供了一些针对连接问题的示例错误消息和可能的解决方案。不同 SQL 客户端工具提供的错误消息不同，因此，虽然此非完整列表，但却是不错的问题排查切入点。

## 从 Amazon EC2 外部连接时遇到防火墙超时问题

在运行 COPY 命令等较长的查询时，客户端到数据库的连接会挂起或超时。此时，您可能会发现，Amazon Redshift 控制台显示查询已完成，而客户端工具仍然显示正在运行查询。查询结果可能会丢失或不完整，具体取决于连接停止的时间。

### 可能的解决方案

当您从 Amazon EC2 实例以外的计算机连接到 Amazon Redshift 时，会发生此问题。在此情况下，空闲连接将在处于不活动状态一段时间后被防火墙等中间网络组件终止。当您从 Virtual Private Network（VPN）或本地网络登录时，通常会出现这种行为。

为避免出现此类超时，建议您执行以下更改：

- 提高用于处理 TCP/IP 超时的客户端系统值。在用来连接集群的计算机上进行这些更改。根据您的客户端和网络调整超时期限。有关更多信息，请参阅 更改 TCP/IP 超时设置。
- （可选）在 DSN 级别设置 Keepalive 行为。有关更多信息，请参阅 更改 DSN 超时设置。

### 更改 TCP/IP 超时设置

要更改 TCP/IP 超时设置，请根据您用来连接集群的操作系统配置超时设置。

- Linux — 如果您的客户端在 Linux 上运行，请以根用户身份运行以下命令来更改当前会话的超时设置：

```
/sbin/sysctl -w net.ipv4.tcp_keepalive_time=200 net.ipv4.tcp_keepalive_intvl=200
 net.ipv4.tcp_keepalive_probes=5
```

要保存设置，请使用以下值创建或修改文件 /etc/sysctl.conf，然后重新启动您的系统。

```
net.ipv4.tcp_keepalive_time=200
net.ipv4.tcp_keepalive_intvl=200
net.ipv4.tcp_keepalive_probes=5
```

- Windows — 如果您的客户端在 Windows 上运行，请在 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\ 下编辑以下注册表设置的值：
  - KeepAliveTime：30000
  - KeepAliveInterval：1000
  - TcpMaxDataRetransmissions：10

这些设置使用 DWORD 数据类型。如果它们不在注册表路径下，您可以创建设置并指定这些建议值。有关编辑 Windows 注册表的更多信息，请参阅 Windows 文档。

在设置好这些值之后，重启您的计算机以使更改生效。

- Mac — 如果您的客户端在 Mac 上运行，请运行以下命令来更改当前会话的超时设置：

```
sudo sysctl net.inet.tcp.keepintvl=200000
sudo sysctl net.inet.tcp.keepidle=200000
sudo sysctl net.inet.tcp.keepinit=200000
sudo sysctl net.inet.tcp.always_keepalive=1
```

要保存设置，请使用以下值创建或修改文件 /etc/sysctl.conf：

```
net.inet.tcp.keepidle=200000
net.inet.tcp.keepintvl=200000
net.inet.tcp.keepinit=200000
net.inet.tcp.always_keepalive=1
```

重新启动计算机，然后运行以下命令来验证值是否已设置。

```
sysctl net.inet.tcp.keepidle
sysctl net.inet.tcp.keepintvl
sysctl net.inet.tcp.keepinit
sysctl net.inet.tcp.always_keepalive
```

更改 DSN 超时设置

如果需要，您可以在 DSN 级别设置 Keepalive 行为。在 odbc.ini 文件中添加或修改以下参数即可实现：

KeepAlivesCount

连接被视为断开前可能丢失的 TCP keepalive 包的数量。

KeepAlivesIdle

驱动程序发送 TCP Keepalive 包前处于不活动状态的秒数。

KeepAlivesInterval

两次传输 TCP keepalive 间隔的秒数。

如果这些参数不存在，或者其值为 0，则系统将使用为 TCP/IP 指定的 Keepalive 参数来确定 DSN Keepalive 行为。在 Windows 上，您可以在注册表中的 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\` 中查找 TCP/IP 参数。在 Linux 和 macOS 上，可以在 sysctl.conf 文件中查找 TCP/IP 参数。

## 连接被拒绝或失败

当连接被拒绝或失败时，您可能会收到类似于以下其中一种的错误。

- “无法建立到 *<endpoint>* 的连接。”
- “无法连接到服务器：连接超时。服务器是否在主机 *'<endpoint>'* 上运行，是否接受端口 *'<port>'* 上的 TCP/IP 连接？”
- “连接被拒。请检查主机名和端口是否正确无误，以及邮件管理员是否接受 TCP/IP 连接。”

可能的解决方案

通常，当您收到指示连接建立失败的错误消息时，是因为访问集群的权限或到达集群的网络流量存在问题。

要从集群所在的网络外部的客户端工具连接到集群，请将入站规则添加到集群的安全组中。规则配置取决于是否在 Virtual Private Cloud (VPC) 中创建了 Amazon Redshift 集群：

- 如果您已在基于 Amazon VPC 的虚拟私有云 (VPC) 中创建 Amazon Redshift 集群，请在 Amazon VPC 中向指定客户端 CIDR/IP 地址的 VPC 安全组添加一条入站规则。有关为集群配置 VPC 安全组和可公开访问的选项的更多信息，请参阅VPC 中的 Redshift 资源。
- 如果您已在 VPC 外部创建 Amazon VPC 集群，则需要将您的客户端 CIDR/IP 地址添加到 Amazon Redshift 中的集群安全组。有关配置集群安全组的更多信息，请参阅Amazon Redshift 安全组。

如果您尝试从在 Amazon EC2 实例上运行的客户端工具连接到集群，也需要添加入站规则。在这种情况下，请向集群安全组添加规则。该规则必须指定与客户端工具的 Amazon EC2 实例关联的 Amazon EC2 安全组。

在某些情况下，客户端和服务器之间可能有一个层，例如防火墙。在这些情况下，请确保防火墙接受通过为集群配置的端口的入站连接。

## 客户端和驱动程序不兼容

如果客户端和驱动程序不兼容，可能会收到“指定的 DSN 包含驱动程序和应用程序之间的架构不匹配”的错误信息。

可能的解决方案

如果您在尝试连接时收到架构不匹配的错误，则说明客户端工具与驱动程序不兼容。此情况是由于其系统架构不匹配导致的。例如，如果您在 32 位客户端工具上安装 64 位驱动程序版本，便会发生这种情况。有时，64 位客户端工具可以使用 32 位驱动程序，但无法在 32 位应用程序上使用 64 位驱动程序。请确保驱动程序和客户端工具使用的是相同版本的系统架构。

## 查询似乎挂起，有时无法连接到集群

您在完成查询时遇到问题，即在 SQL 客户端工具中，查询显示为正在进行，但实则处于挂起状态。有时，查询无法显示在集群中，如系统表或 Amazon Redshift 控制台中。

可能的解决方案

此问题可能是由于数据包丢失导致的。在此情况下，两个 Internet 协议 (IP) 主机在网络路径中的最大传输单位 (MTU) 大小不同。MTU 大小用于确定可通过网络连接在单个以太网帧中传输的数据包的最大大小（以字节为单位）。在 AWS 中，一些 Amazon EC2 实例类型支持 1500 MTU（以太网 v2 帧），其他实例类型支持 9001 MTU（TCP/IP 巨型帧）。

为避免因 MTU 大小不同可能导致的各种问题，建议您执行以下操作之一：

- 如果集群使用 EC2-VPC 平台，则使用将返回 `Destination Unreachable` 的入站自定义 互联网控制消息协议 (ICMP) 规则配置 Amazon VPC 安全组。因此，该规则指示原始主机沿网络路径使用最小的 MTU 大小。有关此方法的详细信息，请参阅配置安全组以允许 ICMP“无法到达目标”。
- 如果您的集群使用 EC2-Classic 平台，或者您无法允许 ICMP 入站规则，请禁用 TCP/IP 巨型帧以便使用以太网 v2 帧。有关此方法的详细信息，请参阅配置实例的 MTU。

配置安全组以允许 ICMP“无法到达目标”

当两个主机在网络中的 MTU 大小存在差异时，请先确保您的网络设置不会阻止路径 MTU 发现 (PMTUD)。PMTUD 使接收主机能够使用以下 ICMP 消息响应原始主机：`Destination Unreachable: fragmentation needed and DF set (ICMP Type 3, Code 4)`。此消息将指示原始主机在网络路径中使用最低 MTU 大小重新发送请求。若无此协商，当请求过大，导致接收主机无法接收时，数据包可能会丢失。有关此 ICMP 消息的更多信息，请转至 国际互联网工程任务组 (IETF) 网站上的 RFC792。

如果您没有为 Amazon VPC 安全组明确配置此 ICMP 入站规则，PMTUD 则将被阻止。在 AWS 中，安全组是虚拟防火墙，用于为到实例的入站和出站流量指定规则。有关 Amazon Redshift 集群安全组的信息，请参阅 Amazon Redshift 安全组。对于使用 EC2-VPC 平台的集群，Amazon Redshift 将使用 VPC 安全组来允许或拒绝到集群的流量。默认情况下，安全组处于锁定状态，会拒绝所有入站流量。有关如何为 EC2-Classic 或 EC2-VPC 实例设置入站和出站规则的信息，请参阅《Amazon EC2 用户指南》中的 EC2-Classic 与 VPC 中的实例间的差异。

有关如何向 VPC 安全组添加规则的更多信息，请参阅VPC 安全组。有关此规则中需要的特定 PMTUD 设置的更多信息，请参阅《Amazon EC2 用户指南》中的路径 MTU 发现。

配置实例的 MTU

在某些情况下，您的集群可能会使用 EC2-Classic 平台，或者您不能允许对入站流量使用自定义 ICMP 规则。在这些情况下，建议您在从中连接到 Amazon Redshift 集群的 EC2 实例的网络接口 (NIC) 上将 MTU 调整为 1500。此调整将禁用 TCP/IP 巨型帧，以确保连接始终使用同一数据包大小。但是，此选

项将从整体上降低实例的最大网络吞吐量，而不仅仅是到 Amazon Redshift 的连接。有关更多信息，请参阅以下流程。

在 Microsoft Windows 操作系统上设置 MTU

如果客户端在 Microsoft Windows 操作系统上运行，则可使用 netsh 命令查看和设置以太网适配器的 MTU 值。

1. 运行以下命令可确定当前 MTU 值：

```
netsh interface ipv4 show subinterfaces
```

2. 在输出中查看 MTU 适配器的 Ethernet 值。
3. 如果值不是 1500，则运行以下命令设置它：

```
netsh interface ipv4 set subinterface "Ethernet" mtu=1500 store=persistent
```

设置好此值后，重启您的计算机以使更改生效。

在 Linux 操作系统上设置 MTU

如果客户端在 Linux 操作系统上运行，则可使用 ip 命令查看和设置 MTU 值。

1. 运行以下命令可确定当前 MTU 值：

```
$ ip link show eth0
```

2. 查看输出中 mtu 后面的值。
3. 如果值不是 1500，则运行以下命令设置它：

```
$ sudo ip link set dev eth0 mtu 1500
```

在 Mac 操作系统上设置 MTU

- 请按照 macOS 支持网站上关于 How to change the MTU for troubleshooting purposes 的说明操作。有关详细信息，请搜索 支持网站。

## 设置 JDBC 提取大小参数

默认情况下，Redshift JDBC 驱动程序使用环形缓冲区来高效管理内存，并防止出现内存不足错误。仅当明确禁用环形缓冲区时，提取大小参数才适用。有关更多信息，请查看链接。在此配置中，您应设置提取大小以控制每个批次中检索的行数。

何时使用提取大小？

在以下情况下使用提取大小参数：

- 您需要对基于行的批处理实施精细控制
- 使用需要传统提取大小行为的旧式应用程序

设置提取大小

在已禁用环形缓冲区的情况下，JDBC 驱动程序会默认一次性收集查询的所有结果。返回大型结果集的查询可能会占用过量内存。要分批而非一次性检索结果集，请在应用程序中设置 JDBC 提取大小参数。

> **Note**
>
> ODBC 不支持提取大小。

为获得最佳性能，请将提取大小设置为不会导致内存不足错误的最大值。较小的提取大小值会导致更多的服务器通信，从而延长执行时间。服务器会预留资源，包括 WLM 查询槽和关联内存，直到客户端检索到整个结果集或查询取消为止。如果适当优化提取大小，则可以更快释放这些资源，使其能够供其他查询使用。

> **Note**
>
> 如果需要提取大型数据集，建议使用 UNLOAD 语句将数据传输到 Amazon S3。使用 UNLOAD 时，计算节点并行工作，以加快数据的传输。

有关设置 JDBC 提取大小参数的更多信息，请参阅 PostgreSQL 文档中的基于光标获取结果。

# 使用 Amazon Redshift 数据 API

Amazon Redshift 数据 API 消除了管理数据库驱动程序、连接、网络配置、数据缓冲、凭证等的需求，从而简化了对 Amazon Redshift 数据仓库的访问。您可以通过 AWS SDK，使用数据 API 操作来运行 SQL 语句。有关数据 API 操作的更多信息，请参阅 Amazon Redshift 数据 API 参考。

数据 API 不需要与数据库的持久连接。相反，它提供了安全 HTTP 端点以及与 AWS 开发工具包的集成。您可以使用终端节点运行 SQL 语句，而无需管理连接。对数据 API 的调用是异步的。数据 API 可以使用存储在 AWS Secrets Manager 中的凭证或临时数据库凭证。您无需使用任何一种授权方法在 API 调用中传递密码。有关 AWS Secrets Manager 的更多信息，请参阅《AWS Secrets Manager 用户指南》中的什么是 AWS Secrets Manager？ 您也可以使用 AWS IAM Identity Center 进行授权。

使用数据 API，您可以通过基于 Web 服务的应用程序（包括 AWS Lambda、Amazon SageMaker AI 笔记本和 AWS Cloud9），以编程方式访问 Amazon Redshift 数据。有关这些应用程序的更多信息，请参阅 AWS Lambda、Amazon SageMaker AI 和 AWS Cloud9。

要了解有关数据 API 的更多信息，请参阅 AWS 大数据博客中的 Get started with the Amazon Redshift Data API。

## 使用 Amazon Redshift 数据 API

在您使用 Amazon Redshift 数据 API 之前，请查看以下步骤：

1. 确定您作为数据 API 的调用者是否获得授权。有关授权的更多信息，请参阅授予对 Amazon Redshift 数据 API 的访问权限。
2. 确定是否计划使用来自 Secrets Manager 的身份验证凭证、临时凭证或使用 AWS IAM Identity Center 来调用数据 API。有关更多信息，请参阅 在调用 Amazon Redshift 数据 API 时选择数据库身份验证凭证。
3. 如果您将 Secrets Manager 用于身份验证凭证，请设置密码。有关更多信息，请参阅 在 AWS Secrets Manager 中存储数据库凭证。
4. 查看调用数据 API 时的注意事项和限制。有关更多信息，请参阅 调用 Amazon Redshift 数据 API 时的注意事项。
5. 从 AWS Command Line Interface (AWS CLI)、您自己的代码中调用数据 API，或使用 Amazon Redshift 控制台中的查询编辑器。有关从 AWS CLI 进行调用的示例，请参阅调用 Data API。

## 调用 Amazon Redshift 数据 API 时的注意事项

调用数据 API 时，请注意以下事项：

- Amazon Redshift 数据 API 可以访问 Amazon Redshift 预置集群和 Redshift Serverless 工作组中的数据库。有关可以使用数据 API 的 AWS 区域列表，请参阅《Amazon Web Services 一般参考》中为 Redshift 数据 API 列出的端点。
- 查询的最长持续时间为 24 小时。
- 每个 Amazon Redshift 集群的最大活动查询（STARTED 和 SUBMITTED 查询）数量为 500 个。
- 最大查询结果大小为 500 MB（gzip 压缩后）。如果调用返回的响应数据超过 500 MB，则调用将结束。
- 查询结果的最长保留时间为 24 小时。
- 最大查询语句大小为 200 KB。
- 数据 API 可用于查询所有无服务器工作组，以及以下节点类型的预调配单节点和多节点集群：
  - dc2.large
  - dc2.8xlarge
  - rg.large
  - rg.xlarge
  - rg.4xlarge
  - rg.12xlarge
  - ra3.large
  - ra3.xlplus
  - ra3.4xlarge
  - ra3.16xlarge
- 集群必须在基于 Amazon VPC 服务的 Virtual Private Cloud (VPC) 中。
- 默认情况下，具有与 ExecuteStatement 或 BatchExecuteStatement API 操作运行者相同的 IAM 角色的用户可以使用 CancelStatement、DescribeStatement、GetStatementResult、GetStatementResultV2 和 ListStatements API 操作对同一个语句进行操作。要从另一个用户对同一 SQL 语句执行操作，该用户必须能够代入运行 SQL 语句的用户的 IAM 角色。有关如何代入角色的更多信息，请参阅授予对 Amazon Redshift 数据 API 的访问权限。
- BatchExecuteStatement API 操作的 Sqls 参数中的 SQL 语句，将按数组顺序串行运行。后续的 SQL 语句要等到前一条语句完成后才会开始。默认情况下，所有 SQL 语句都作为单个事务运行。如果任何 SQL 语句失败，所有工作都将回滚。使用 ExecutionMode 参数来控制事务行为：
  - TRANSACTION（默认）：该服务将所有 SQL 语句作为单个事务运行，并统一进行提交或回滚。
  - AUTO_COMMIT：每个 SQL 语句都单独提交。一个语句的失败不会影响其它语句。

有关示例，请参阅运行多条 SQL 语句。

- 在 ExecuteStatement 或 BatchExecuteStatement API 操作中使用的客户端令牌的最长保留时间为 8 小时。
- 当操作提交新查询或针对正在进行的查询时，可选的 WaitTimeSeconds 参数可让数据 API 最多等待 30 秒。这减少了轮询请求的数量，从而减少了往返次数并降低了延迟。有关更多信息，请参阅 使用长轮询减少 API 调用。
- 如果使用客户自主管理型密钥对 Amazon Redshift 预调配集群和 Redshift Serverless 工作组进行加密，则 Redshift 将创建一项授权，允许 Redshift Data API 将该密钥用于其操作。有关更多信息，请参阅 结合使用 AWS KMS 与 Amazon Redshift Data API。
- Redshift 数据 API 中的每个 API 都有每秒事务次数配额，超过该配额会对请求节流。有关配额，请参阅 Amazon Redshift 数据 API 的配额。如果请求速率超过配额，则会返回 ThrottlingException 以及“HTTP 状态代码：400”。要应对节流的情况，请使用重试策略，如《AWS SDK 和工具参考指南》中的重试行为所述。在某些 AWS SDK 中，会针对节流错误自动实施这一策略。

> **Note**
>
> 在 AWS Step Functions 中，默认情况下未启用重试功能。如果您需要在 Step Functions 状态机中调用 Redshift 数据 API，请在 Redshift 数据 API 调用中包括 ClientToken 幂等性参数。ClientToken 的值需要在重试之间保持不变。在以下 ExecuteStatement API 请求的示例片段中，表达式 States.ArrayGetItem(States.StringSplit($$.Execution.Id, ':'), 7) 使用内置函数提取 $$.Execution.Id 的 UUID 部分，这一部分在状态机的每次执行中都是唯一的。有关更多信息，请参阅《AWS Step Functions 开发人员指南》中的内置函数。
>
> ```
> {
>   "Database": "dev",
>   "Sql": "select 1;",
>   "ClusterIdentifier": "MyCluster",
>   "ClientToken.$": "States.ArrayGetItem(States.StringSplit($$.Execution.Id,
>  ':'), 7)"
> }
> ```

# 在调用 Amazon Redshift 数据 API 时选择数据库身份验证凭证

当您调用数据 API 时，您对某些 API 操作使用以下身份验证方法之一。每种方法都需要不同的参数组合。

AWS IAM Identity Center

在 AWS IAM Identity Center 中注册的单点登录用户可以访问数据 API。有关设置 IAM Identity Center 的步骤的信息，请参阅将数据 API 与可信身份传播结合使用。

AWS Secrets Manager

使用此方法，提供存储在 AWS Secrets Manager 中密钥的 `secret-arn`，其中包括了 `username` 和 `password`。指定的密钥包含用于连接到您指定的 `database` 的凭证。在连接到集群时，如果您提供了集群标识符（`dbClusterIdentifier`），则还可以提供数据库名称，该名称必须与密钥中存储的集群标识符相匹配。在连接到无服务器工作组时，也要提供数据库名称。有关更多信息，请参阅 在 AWS Secrets Manager 中存储数据库凭证。

使用此方法，您还可以提供 `region` 值，该值指定您的数据所在的 AWS 区域。

临时凭证

使用此方法时，请选择以下选项之一：

- 在连接到无服务器工作组时，需指定工作组名称和数据库名称。数据库用户名派生自 IAM 身份。例如，`arn:iam::123456789012:user:foo` 具有数据库用户名 `IAM:foo`。此外，需具有调用 `redshift-serverless:GetCredentials` 操作的权限。
- 以 IAM 身份连接到集群时，需要指定集群标识符和数据库名称。数据库用户名派生自 IAM 身份。例如，`arn:iam::123456789012:user:foo` 具有数据库用户名 `IAM:foo`。此外，需具有调用 `redshift:GetClusterCredentialsWithIAM` 操作的权限。
- 以数据库用户的身份连接到机群时，需要指定集群标识符、数据库名称和数据库用户名。此外，需具有调用 `redshift:GetClusterCredentials` 操作的权限。有关使用此方法进行连接时如何加入数据库组的信息，请参阅连接到集群时加入数据库组。

使用此方法，您还可以提供 `region` 值，该值指定您的数据所在的 AWS 区域。

# 调用 Amazon Redshift 数据 API 时映射 JDBC 数据类型

下表将 Java 数据库连接 (JDBC) 数据类型映射到您在数据 API 调用中指定的数据类型。

| JDBC 数据类型 | Data API 数据类型 |
| --- | --- |
| INTEGER, SMALLINT, BIGINT | LONG |
| FLOAT, REAL, DOUBLE | DOUBLE |
| DECIMAL | STRING |
| BOOLEAN, BIT | BOOLEAN |
| BLOB, BINARY, LONGVARBINARY | BLOB |
| VARBINARY | STRING |
| CLOB | STRING |
| 其他类型（包括与日期和时间有关的类型） | STRING |

字符串值将传递到 Amazon Redshift 数据库并隐式转换为数据库数据类型。

> **Note**
>
> 目前，数据 API 不支持通用唯一标识符 (UUID) 的数组。

## 在调用 Amazon Redshift 数据 API 时运行带有参数的 SQL 语句

您可以通过使用 SQL 语句部分的参数调用数据 API 操作来控制提交到数据库引擎的 SQL 文本。命名参数提供了一种灵活的方式来传入参数，而无需在 SQL 文本中对参数进行硬编码。它们可以帮助您重复使用 SQL 文本并避免 SQL 注入问题。

以下示例显示 `execute-statement` 或 `batch-execute-statement` AWS CLI 命令的 `parameters` 字段的命名参数。

```
--parameters "[{\"name\": \"id\", \"value\": \"1\"},{\"name\": \"address\", \"value\":
 \"Seattle\"}]"
```

使用命名参数时，请注意以下事项：

- 命名参数只能用于替换 SQL 语句中的值。

- 您可以替换 INSERT 语句中的值，例如 `INSERT INTO mytable VALUES(:val1)`。

  命名参数可以按任意顺序排列，并且参数可以在 SQL 文本中多次使用。前面示例中显示的参数选项，值 1 和 `Seattle` 插入到表列 `id` 和 `address` 中。在 SQL 文本中，您可以按如下方式指定命名参数：

  ```
  --sql "insert into mytable values (:id, :address)"
  ```

- 您可以替换条件子句中的值，例如 `WHERE attr >= :val1`、`WHERE attr BETWEEN :val1 AND :val2` 和 `HAVING COUNT(attr) > :val`。
- 您无法替换 SQL 语句中的列名，例如 `SELECT column-name`、`ORDER BY column-name` 或 `GROUP BY column-name`。

  例如，以下 SELECT 语句将失败，出现语法无效错误。

  ```
  --sql "SELECT :colname, FROM event" --parameters "[{\"name\": \"colname\", \"value
  \": \"eventname\"}]"
  ```

  如果您说明（`describe-statement` 操作）语句有语法错误，则返回的 `QueryString` 不会用列名代替参数（`"QueryString": "SELECT :colname, FROM event"`），并且会报告错误（错误：在 \"FROM\"\ 处或附近出现语法错误 \n 位置：12）。

- 您无法替换聚合函数中的列名，例如 `COUNT(column-name)`、`AVG(column-name)` 或 `SUM(column-name)`。
- 您无法替换 JOIN 子句中的列名。

- SQL 运行时，数据将隐式转换为数据类型。有关数据类型转换的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的数据类型。
- 您不能将值设置为 NULL。数据 API 将其解释为文本字符串 `NULL`。以下示例将 `id` 替换为文本字符串 `null`。不是 SQL NULL 值。

  ```
  --parameters "[{\"name\": \"id\", \"value\": \"null\"}]"
  ```

- 无法设置零长度值。数据 API SQL 语句失败。下面的示例尝试设置 `id` 长度值为零，导致 SQL 语句失败。

  ```
  --parameters "[{\"name\": \"id\", \"value\": \"\"}]"
  ```

- 不能在带有参数的 SQL 语句中设置表名。数据 API 遵循 JDBC `PreparedStatement` 的规则。

- describe-statement 操作的输出返回 SQL 语句的查询参数。
- 仅 execute-statement 和 batch-execute-statement 操作支持带有参数的 SQL 语句。使用 batch-execute-statement 时，参数将在批处理中的所有 SQL 语句之间共享。每个 SQL 语句都可以引用所提供参数的子集，但每个参数都必须由至少一个 SQL 语句使用。

## 在调用 Amazon Redshift 数据 API 时运行带有幂等性令牌的 SQL 语句

当您发出变更 API 请求时，该请求通常会在操作的异步工作流完成之前返回结果。即使请求已经返回结果，操作在完成之前也可能会超时或遇到其他服务器问题。这样就很难确定请求是否成功，并且会导致进行多次重试以确保操作成功完成。但是，如果原始请求和后续重试成功，则会多次完成操作。这意味着您可能会更新比预期更多的资源。

幂等性确保 API 请求完成不超过一次。对于幂等性请求，如果原始请求成功完成，则后续重试也会成功完成，而不会执行任何后续操作。数据 API ExecuteStatement 和 BatchExecuteStatement 操作具有可选的 ClientToken 幂等性参数。ClientToken 在 8 小时后过期。

> **Important**
>
> 如果您从 AWS SDK 调用 ExecuteStatement 和 BatchExecuteStatement 操作，它会自动生成客户端令牌以供重试时使用。在这种情况下，我们不建议将 client-token 参数与 ExecuteStatement 和 BatchExecuteStatement 操作一起使用。查看 CloudTrail 日志以查看 ClientToken。有关 CloudTrail 日志示例，请参阅 Amazon Redshift 数据 API 示例。

以下 execute-statement AWS CLI 命令说明了幂等性的可选 client-token 参数。

```
aws redshift-data execute-statement
    --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
    --cluster-identifier mycluster-test
    --sql "select * from stl_query limit 1"
    --database dev
    --client-token b855dced-259b-444c-bc7b-d3e8e33f94g1
```

下表显示了幂等性 API 请求可能获得的一些常见响应，并提供了重试建议。

| 响应 | 建议 | 评论 |
| --- | --- | --- |
| 200（正常） | 不重试 | 原始请求成功完成。成功返回任何后续重试。 |
| 400 系列响应代码 | 不重试 | 请求存在问题，原因如下：<br>• 它包括无效的参数或参数组合。<br>• 它使用您没有权限的操作或资源。<br>• 它使用正在改变状态的资源。<br><br>如果请求涉及正在改变状态的资源，则重试请求可能会成功。 |
| 500 系列响应代码 | 重试 | 该错误是由 AWS 服务器端问题引起，通常是暂时性的。使用适当的退避策略重复发出请求。 |

有关 Amazon Redshift 响应代码的信息，请参阅《Amazon Redshift API 参考》中的常见错误。

## 在调用 Amazon Redshift 数据 API 时运行带有会话重用的 SQL 语句

当您发出 API 请求来运行 SQL 语句时，在其中运行 SQL 的会话通常在 SQL 完成后终止。为了在指定的秒数内使会话保持活动状态，数据 API `ExecuteStatement` 和 `BatchExecuteStatement` 操作具有可选的 `SessionKeepAliveSeconds` 参数。`SessionId` 响应字段包含会话的身份，随后可以在后续 `ExecuteStatement` 和 `BatchExecuteStatement` 操作中使用该身份。在后续调用中，可以指定另一个 `SessionKeepAliveSeconds` 来更改空闲超时时间。如果未更改 `SessionKeepAliveSeconds`，则会保留初始空闲超时设置。使用会话重用时，请考虑以下事项：

- `SessionKeepAliveSeconds` 的最大值为 24 小时。
- 会话最多可持续 24 小时。24 小时后，会话将被强制关闭，正在进行的查询将被终止。
- 每个 Amazon Redshift 集群或 Redshift Serverless 工作组的最大会话数为 500 个。
- 在会话中，一次只能运行一个查询。您需要等到查询完成后，才能在同一个会话中运行下一个查询。也就是说，不能在提供的会话中并行运行查询。
- 数据 API 无法对给定会话的查询进行排队。

要检索调用 ExecuteStatement 和 BatchExecuteStatement 操作时所使用的 SessionId，请调用 DescribeStatement 和 ListStatements 操作。要列出会话及其状态，请使用 ListSessions 操作。有关更多信息，请参阅 列出会话。

以下示例演示如何使用 SessionKeepAliveSeconds 和 SessionId 参数来使会话保持活动状态并重用。首先，在可选 session-keep-alive-seconds 参数设置为 2 的情况下调用 execute-statement AWS CLI 命令。

```
aws redshift-data execute-statement
    --session-keep-alive-seconds 2
    --sql "select 1"
    --database dev
    --workgroup-name mywg
```

响应将包含会话标识符。

```
{
    "WorkgroupName": "mywg",
    "CreatedAt": 1703022996.436,
    "Database": "dev",
    "DbUser": "awsuser",
    "Id": "07c5ffea-76d6-4786-b62c-4fe3ef529680",
    "SessionId": "5a254dc6-4fc2-4203-87a8-551155432ee4"
}
```

然后，使用从第一个调用返回的 SessionId 来调用 execute-statement AWS CLI 命令。或者，指定将 session-keep-alive-seconds 参数设置为 10 以更改空闲超时值。

```
aws redshift-data execute-statement
    --sql "select 1"
    --session-id 5a254dc6-4fc2-4203-87a8-551155432ee4
    --session-keep-alive-seconds 10
```

## 获取 SQL 语句的结果

根据结果格式，您可以使用不同的数据 API 操作来获取 SQL 结果。在调用 ExecuteStatement 和 BatchExecuteStatement 操作时，可以指定结果的格式是 JSON 还是 CSV。如果未指定，则默认

格式为 JSON。要检索 JSON 结果，请使用 GetStatementResult 操作。要检索 CSV 结果，请使用 GetStatementResultV2 操作。

以 JSON 格式返回的结果是包含有关每列的元数据的记录。每条记录都是 JSON 格式。例如，来自 GetStatementResult 的响应与以下内容相似：

```
{
   "ColumnMetadata": [
      {
         "isCaseSensitive": false,
         "isCurrency": false,
         "isSigned": true,
         "label": "?column?",
         "name": "?column?",
         "nullable": 1,
         "precision": 10,
         "scale": 0,
         "schemaName": "",
         "tableName": "",
         "typeName": "int4",
         "length": 0
      }
   ],
   "NextToken": "<token>",
   "Records": [
        [
            {
                "longValue": 1
            }
        ]
    ],
   "TotalNumRows": <number>
}
```

以 CSV 格式返回的结果是包含有关每列的元数据的记录。结果以 1 MB 的分块形式返回，其中每个分块能够以 CSV 格式存储任意数量的行。每个请求最多返回 15 MB 的结果。如果结果大于 15 MB，则返回下一页令牌以继续检索结果。例如，来自 GetStatementResultV2 的响应与以下内容相似：

```
{
    "ColumnMetadata": [
        {
            "isCaseSensitive": false,
```

```
            "isCurrency": false,
            "isSigned": true,
            "label": "?column?",
            "name": "?column?",
            "nullable": 1,
            "precision": 10,
            "scale": 0,
            "schemaName": "",
            "tableName": "",
            "typeName": "int4",
            "length": 0
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "?column?",
            "name": "?column?",
            "nullable": 1,
            "precision": 10,
            "scale": 0,
            "schemaName": "",
            "tableName": "",
            "typeName": "int4",
            "length": 0
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "?column?",
            "name": "?column?",
            "nullable": 1,
            "precision": 10,
            "scale": 0,
            "schemaName": "",
            "tableName": "",
            "typeName": "int4",
            "length": 0
        }
    ],
    "NextToken": "<token>",
    "Records": [
        [
```

```
            {
                "CSVRecords":"1,2,3\r\n4,5,6\r\n7,8,9\r\n, ... 1MB" // First 1MB Chunk
            },
            {
                "CSVRecords":"1025,1026,1027\r\n1028,1029,1030\r\n...2MB" // Second 1MB
 chunk
            }
            ...
        ]
    ],
    "ResultFormat" : "CSV",
    "TotalNumRows": <number>
}
```

# 授予对 Amazon Redshift 数据 API 的访问权限

用户必须获得授权才能访问数据 API。您可以通过将托管式策略（预定义的 AWS Identity and Access Management (IAM) 策略）添加给用户，授予该用户访问数据 API 的权限。作为最佳实践，我们建议将权限策略附加到 IAM 角色，然后根据需要将其分配给用户和组。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。要查看托管式策略允许和拒绝的权限，请参阅 IAM 控制台 (https://console.aws.amazon.com/iam/)。

## 配置 IAM 权限

Amazon Redshift 提供 `AmazonRedshiftDataFullAccess` 托管式策略。此策略提供了对 Amazon Redshift 数据 API 操作的完全访问。此策略还允许将访问权限限定为特定 Amazon Redshift、AWS Secrets Manager 以及对 Amazon Redshift 集群或 Redshift Serverless 进行身份验证和访问所需的 IAM API 操作。

此外，您还可以创建自己的 IAM 策略，以允许对特定资源的访问。要创建策略，请使用 `AmazonRedshiftDataFullAccess` 策略作为起始模板。在创建策略后，将该策略添加到需要访问数据 API 的每个用户。

考虑与用户关联的 IAM 策略的以下要求：

- 如果您使用 AWS Secrets Manager 进行身份验证，请确认策略允许使用 `secretsmanager:GetSecretValue` 操作来检索使用键 `RedshiftDataFullAccess` 标记的密钥。
- 如果您使用临时凭证对集群进行身份验证，请确认该策略允许将 `redshift:GetClusterCredentials` 操作用于集群中任何数据库的数据库用户名 `redshift_data_api_user`。此用户名必须已在数据库中创建。

- 如果您使用临时凭证对无服务器工作组进行身份验证，请确认该策略允许使用 `redshift-serverless:GetCredentials` 操作来检索使用键 `RedshiftDataFullAccess` 标记的工作组。数据库用户按 1:1 的比例映射到源 AWS Identity and Access Management (IAM) 身份。例如，用户 sample_user 映射到数据库用户 `IAM:sample_user`，而 IAM 角色 sample_role 映射到 `IAMR:sample_role`。有关 IAM 身份的更多信息，请参阅《IAM 用户指南》中的 IAM 身份（用户、组和角色）。
- IAM 操作 `redshift-data:GetStatementResult` 同时支持访问 `GetStatementResult` 和 `GetStatementResultV2` API 操作。

以下链接提供了《IAM 用户指南》中有关 AWS Identity and Access Management 的更多信息。

- 有关创建 IAM 角色的信息，请参阅创建 IAM 角色。
- 有关创建 IAM 策略的更多信息，请参阅创建 IAM 策略。
- 有关将 IAM 策略添加到用户的信息，请参阅添加和删除 IAM 身份权限。

在其它账户拥有的集群上运行查询

要对其他账户拥有的集群运行查询，拥有账户必须提供一个 IAM 角色，数据 API 可以在调用账户时代入该角色。例如，假设账户 B 拥有账户 A 需要访问的集群。账户 B 可以将 AWS 托管式策略 `AmazonRedshiftDataFullAccess` 附加到账户 B 的 IAM 角色。然后，账户 B 使用信任策略信任账户 A，如下所示：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:role/someRoleA"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

最后，账户 A 的 IAM 角色需要能够代入账户 B 的 IAM 角色。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
        "Resource": "arn:aws:iam::111122223333:role/someRoleB"
    }
}
```

指定一个 IAM 角色，该角色将资源限制为 AWS 账户中的 Redshift Serverless 工作组和 Amazon Redshift 集群

可以在基于身份的策略中指定资源 ARN，以便控制对 AWS 账户中 Redshift Serverless 工作组和 Amazon Redshift 集群的访问权限。此示例显示如何创建一个策略，该策略只支持针对指定 AWS 账户中的工作组和集群访问数据 API。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "redshift-data:CancelStatement",
                "redshift-data:DescribeStatement",
                "redshift-data:GetStatementResult",
                "redshift-data:ListStatements"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "redshift-data:*",
            "Resource": [
                "arn:aws:redshift:us-east-1:111122223333:workgroup/*",
```

```
                "arn:aws:redshift:us-east-1:111122223333:cluster:*"
            ]
        }
    ]
}
```

配置 IAM 策略，来限制只有语句所有者才能访问 SQL 语句信息

默认情况下，Amazon Redshift 数据 API 将调用 `ExecuteStatement` 和 `BatchExecuteStatement` 时使用的 IAM 角色视为 SQL 语句的所有者。任何获准代入该角色的人都可以访问有关 SQL 语句的信息，包括其结果。要将 SQL 语句信息访问权限限制为与特定所有者之间的 IAM 角色会话，请添加条件 `redshift-data:statement-owner-iam-userid: "${aws:userid}"`。以下 IAM 策略会限制访问权限。

JSON

```
{
"Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "redshift-data:CancelStatement",
                "redshift-data:DescribeStatement",
                "redshift-data:GetStatementResult",
                "redshift-data:ListStatements"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "redshift-data:statement-owner-iam-userid": "${aws:userid}"
                }
            }
        }
    ]
}
```

可以将条件 statement-owner-iam-userid 与 CancelStatement、DescribeStatement、GetStatementResult 和 ListStatements 结合使用。有关更多信息，请参阅 Amazon Redshift Data API 定义的操作。

配置 IAM 策略，来限制只有会话所有者才能访问 SQL 结果

默认情况下，Amazon Redshift 数据 API 将调用 ExecuteStatement 和 BatchExecuteStatement 时使用的 IAM 角色视为运行 SQL 语句的数据库会话的所有者。任何获准代入该角色的人都可以提交对数据库会话的查询。要将会话访问权限限制为与特定所有者之间的 IAM 角色会话，请添加条件 redshift-data:session-owner-iam-userid: "${aws:userid}"。以下 IAM 策略会限制访问权限。

以下 IAM 策略仅支持会话所有者获取语句结果。条件 session-owner-iam-userid 用于将资源访问权限限制为指定的 userid。

JSON

```
{
"Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "redshift-data:ExecuteStatement",
                "redshift-data:BatchExecuteStatement"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "redshift-data:session-owner-iam-userid": "${aws:userid}"
                }
            }
        }
    ]
}
```

可以将条件 session-owner-iam-userid 与 ExecuteStatement 和 BatchExecuteStatement 结合使用。有关更多信息，请参阅 Amazon Redshift Data API 定义的操作。

## 在 AWS Secrets Manager 中存储数据库凭证

在调用数据 API 时，您可以使用 AWS Secrets Manager 中的密钥传递集群或无服务器工作组的凭证。要通过此方式传递凭证，您需要指定密钥的名称或密钥的 Amazon 资源名称（ARN）。

要使用 Secrets Manager 存储凭证，您需要 `SecretManagerReadWrite` 托管式策略权限。有关最低权限的更多信息，请参阅《AWS Secrets Manager 用户指南》中的使用 AWS Secrets Manager 创建和管理密钥。

要将凭证存储在 Amazon Redshift 集群的密钥中

1. 使用 AWS Secrets Manager 控制台创建包含集群凭证的密钥：

   - 当您选择存储新密钥时，选择Redshift 集群的凭证。
   - 将用户名（数据库用户）、密码和 数据库集群（集群标识符）的值存储在您的密钥中。
   - 使用键 `RedshiftDataFullAccess` 标记密钥。AWS 托管式策略 `AmazonRedshiftDataFullAccess` 只支持对使用键 `RedshiftDataFullAccess` 进行标记的密钥执行操作 `secretsmanager:GetSecretValue`。

   有关说明，请参阅《AWS Secrets Manager 用户指南》中的创建基本密钥。
2. 使用 AWS Secrets Manager 控制台查看您创建的密钥的详细信息，或运行 `aws secretsmanager describe-secret` AWS CLI 命令。

   记下密钥的名称和 ARN。您可以将其用于对数据 API 的调用中。

将凭证存储在无服务器工作组的密钥中

1. 使用 AWS Secrets Manager AWS CLI 命令存储包含无服务器工作组凭证的密钥：

   - 在文件中创建密钥，例如名为 `mycreds.json` 的 JSON 文件。在文件中提供用户名（数据库用户）和密码的值。

     ```
     {
           "username": "myusername",
           "password": "mypassword"
     }
     ```
   - 将值存储在密钥中，并使用键 `RedshiftDataFullAccess` 标记密钥。

```
aws secretsmanager create-secret --name MyRedshiftSecret  --tags
 Key="RedshiftDataFullAccess",Value="serverless" --secret-string file://
mycreds.json
```

下面显示了输出。

```
{
    "ARN":
 "arn:aws:secretsmanager:region:accountId:secret:MyRedshiftSecret-mvLHxf",
    "Name": "MyRedshiftSecret",
    "VersionId": "a1603925-e8ea-4739-9ae9-e509eEXAMPLE"
}
```

有关更多信息，请参阅《AWS Secrets Manager User Guide》中的 Creating a Basic Secret with AWS CLI。

2. 使用 AWS Secrets Manager 控制台查看您创建的密钥的详细信息，或运行 `aws secretsmanager describe-secret` AWS CLI 命令。

   记下密钥的名称和 ARN。您可以将其用于对数据 API 的调用中。

## 为数据 API 创建 Amazon VPC 终端节点 (AWS PrivateLink)

借助 Amazon Virtual Private Cloud (Amazon VPC)，您可以在 Virtual Private Cloud (VPC) 中启动 AWS 资源（例如 Amazon Redshift 集群和应用程序）。AWS PrivateLink 在亚马逊网络上提供了 Virtual Private Cloud (VPC) 和 AWS 服务之间的私有连接。通过使用 AWS PrivateLink，您可以创建 VPC 终端节点，这可让您根据 Amazon VPC 跨不同的账户和 VPC 连接到服务。有关 AWS PrivateLink 的更多信息，请参阅《Amazon Virtual Private Cloud 用户指南》中的 VPC 终端节点服务 (AWS PrivateLink)。

您可以使用 Amazon VPC 终端节点调用数据 API。使用 Amazon VPC 终端节点可保留 Amazon VPC 中应用程序间的流量与AWS网络中的 Data API，而无需使用公有 IP 地址。Amazon VPC 终端节点可帮助您遵守与管理公共互联网连接有关的合规性和法规要求。例如，如果您使用 Amazon VPC 终端节点，则可保持 Amazon EC2 实例上运行的应用程序和包含终端节点的 VPC 中的 Data API 之间的流量。

创建 Amazon VPC 终端节点后，便能开始使用此终端节点，而无需在应用程序中进行任何代码或配置更改。

为 Data API 创建 Amazon VPC 终端节点

1. 登录到AWS 管理控制台并打开 Amazon VPC 控制台，网址：https://console.aws.amazon.com/vpc/。
2. 选择端点，然后选择创建端点。
3. 在创建终端节点页面上，为服务类别选择 AWS 服务。对于服务名称，选择 redshift-data (`com.amazonaws.region.redshift-data`)。
4. 对于 VPC，选择要在其中创建终端节点的 VPC。

   选择包含用于进行 Data API 调用的应用程序的 VPC。
5. 对于子网，请为运行应用程序的 AWS 服务所使用的每个可用区 (AZ) 选择子网。

   要创建 Amazon VPC 终端节点，请指定端点可在其中访问的私有 IP 地址范围。为此，请为每个可用区选择子网。执行此操作会将 VPC 端点限制为特定于每个可用区的私有 IP 地址范围，并且还会在每个可用区中创建一个 Amazon VPC 端点。
6. 对于启用 DNS 名称，选择为此终端节点启用。

   私有 DNS 会将标准 Data API DNS 主机名 (`https://redshift-data.region.amazonaws.com`) 解析为与特定于 Amazon VPC 终端节点的 DNS 主机名关联的私有 IP 地址。因此，您可以使用 AWS CLI 或 AWS 开发工具包访问 Data API VPC 终端节点，而无需进行任何代码或配置更改来更新 Data API 终端节点 URL。
7. 对于安全组，选择要与 Amazon VPC 终端节点关联的安全组。

   选择允许访问运行应用程序的 AWS 服务的安全组。例如，如果 Amazon EC2 实例正在运行您的应用程序，请选择允许访问 Amazon EC2 实例的安全组。利用安全组，您可以控制从 VPC 中的资源发送到 Amazon VPC 终端节点的流量。
8. 选择Create endpoint。

创建终端节点后，选择 AWS 管理控制台中的链接以查看终端节点详细信息。

终端节点详细信息选项卡将显示在创建 Amazon VPC 终端节点时生成的 DNS 主机名。

您可以使用标准终端节点 (`redshift-data.region.amazonaws.com`) 或特定于 VPC 的终端节点之一来调用 Amazon VPC 中的 Data API。标准 Data API 端点将自动路由到 Amazon VPC 端点。发生此路由的原因是，在创建 Amazon VPC 终端节点时启用了私有 DNS 主机名。

在 Data API 调用中使用 Amazon VPC 终端节点时，应用程序和 Data API 之间的所有流量将在包含它们的 Amazon VPC 中保留。可以将 Amazon VPC 终端节点用于任何类型的 Data API 调用。有关调用 Data API 的信息，请参阅调用 Amazon Redshift 数据 API 时的注意事项。

## 连接到集群时加入数据库组

数据库组是数据库用户的集合。可以向组授予数据库权限。管理员可以配置 IAM 角色，以便在使用数据 API 运行您的 SQL 时，将这些数据库组考虑在内。有关数据库组的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的组。

您可以配置数据 API 调用者的 IAM 角色，以便在数据 API 连接到集群时，在调用中指定的数据库用户加入数据库组。只有在连接到预置集群时才支持此功能。连接到 Redshift Serverless 工作组时不支持此功能。数据 API 调用方的 IAM 角色还必须允许 `redshift:JoinGroup` 操作。

通过向 IAM 角色添加标签来对此进行配置。调用方 IAM 角色的管理员添加以 `RedshiftDbGroups` 为键、以数据库组列表为键值的标签。该值是以冒号 (:) 分隔的数据库组名称的列表，总长度不超过 256 个字符。必须事先在连接的数据库中定义数据库组。如果在数据库中找不到任何指定的组，则将其忽略。例如，对于数据库组 `accounting` 和 `retail`，键/值为 `accounting:retail`。标签键/值对 `{"Key":"RedshiftDbGroups","Value":"accounting:retail"}` 由数据 API 用于确定哪些数据库组与调用数据 API 时提供的数据库用户相关联。

加入数据库组

1. 登录AWS 管理控制台，然后通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在控制台的导航窗格中，选择角色，然后选择要编辑的角色的名称。
3. 选择标签选项卡，然后选择管理标签。
4. 选择添加标签，然后添加键 RedshiftDbGroups 以及一个值，该值是 *`database-groups-colon-separated`* 的列表。
5. 选择保存更改。

   现在，当 IAM 主体（附加了此 IAM 角色）调用数据 API 时，指定的数据库用户将加入在 IAM 角色中指定的数据库组。

有关如何将标签附加至主体（包括 IAM 角色和 IAM 用户）的更多信息，请参阅《IAM 用户指南》中的标记 IAM 资源部分。

# 将数据 API 与可信身份传播结合使用

作为 Amazon Redshift 账户管理员，您可以将 Amazon Redshift 集群或工作组与 AWS IAM Identity Center 集成，这有助于管理您的员工通过单点登录访问 Amazon Redshift 的权限。有关更多信息，请参阅 设置 AWS IAM Identity Center 与 Amazon Redshift 的集成。Amazon Redshift 数据 API 支持将 IAM Identity Center 用户身份传播到 Amazon Redshift 集群或工作组以及链下游的其它服务，例如 AWS Lake Formation。您可以按照 Access AWS services programmatically using trusted identity propagation 中的步骤操作，使用数据 API 进行设置和查询。

当您在身份增强型 IAM 角色会话中使用 IAM Identity Center 用户身份调用数据 API 时，您只能使用相同的 IAM Identity Center 用户访问生成的语句和语句结果。例如，以下 AWS CLI 命令调用 `execute-statement` 操作，以便使用可信身份传播来运行 SQL 命令。

```
aws redshift-data execute-statement
--sql "select current_user;"
--cluster-id mycluster
--database dev
```

以下 AWS CLI 命令调用 `batch-execute-statement` 操作来运行两个 SQL 命令。

```
aws redshift-data batch-execute-statement
--sqls  "select current_user;"  "select current_date;"
--cluster-id mycluster
--database dev
```

要访问由身份增强型 IAM 角色会话提交的具有 `cancel-statement`、`describe-statement`、`get-statement-result` 和 `get-statement-result-v2` 的语句，IAM Identity Center 用户和 IAM 角色必须与用于运行 `execute-statement` 或 `batch-execute-statement` 的凭证相匹配。例如，以下 AWS CLI 命令会获取 SQL 语句的结果。

```
aws redshift-data get-statement-result
--id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

要列出语句，必须提供 `cluster-identifier` 或 `workgroup-name` 参数，以确保 IAM Identity Center 用户只能访问分配给他们的 Amazon Redshift IAM Identity Center 应用程序。例如，以下 AWS CLI 命令会列出特定集群的语句。

```
aws redshift-data list-statements
```

```
--cluster-identifier mycluster
```

您还可以调用数据 API 操作，这些操作使用可信身份传播访问集群或工作组中的数据库对象。这包括 `list-databases`、`list-schemas`、`list-tables` 和 `describe-table` 操作。

可以在 AWS CloudTrail 中跟踪由 IAM Identity Center 用户发出的 API 调用。CloudTrail 事件的 `onBehalfOf` 部分显示了 IAM Identity Center 用户 ID 和身份存储 ARN。以下示例显示了 CloudTrail 事件的一个片段，其中显示了 `onBehalfOf` 部分，IAM Identity Center ID 为 `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`，身份存储 ARN 为 `arn:aws:identitystore::123456789012:identitystore/d-9067bc44d2`。

```
{
            "eventVersion":"1.10",
            "userIdentity":{
            "type":"AssumedRole",
            ...
            },
            "onBehalfOf":{
            "userId":"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "identityStoreArn":"arn:aws:identitystore::123456789012:identitystore/
d-9067bc44d2"
            }
            },
            "eventTime":"2025-01-13T04:46:27Z",
            "eventSource":"redshift-data.amazonaws.com",
            "eventName":"ExecuteStatement",
            "awsRegion":"us-east-1"
            }
```

您可以运行以下 SQL 命令来检查由 IAM Identity Center 用户提交的查询。在此示例中，在 Identity Center 中注册的电子邮件为 `username@example.com`。

```
SELECT
    h.query_id,
    h.database_name,
    h.status,
    h.query_text,
    u.usename,
    h.start_time,
    h.end_time
FROM
    sys_query_history h
```

```
LEFT JOIN
    pg_user u
ON
    h.user_id = u.usesysid
where u.usename='awsidc:username@example.com'
ORDER BY
    h.start_time DESC;
```

## 调用 Data API

您可以调用数据 API 或 AWS CLI 以在集群或无服务器工作组上运行 SQL 语句。运行 SQL 语句的主要操作是《Amazon Redshift Data API Reference》中的 `ExecuteStatement` 和 `BatchExecuteStatement`。数据 API 支持 AWS 开发工具包所支持的编程语言。有关它们的更多信息，请参阅用于在 AWS 上构建的工具。

要查看调用 Data API 的代码示例，请参阅 GitHub 中的 Redshift 数据 API入门。此存储库包含使用 AWS Lambda，从 Amazon EC2、AWS Glue Data Catalog 和 Amazon SageMaker Runtime 访问 Amazon Redshift 数据的示例。示例编程语言包括 Python、Go、Java 和 Javascript。

您可以使用 AWS CLI 调用 Data API。

以下示例使用 AWS CLI 调用数据 API。要运行示例，请编辑参数值以匹配您的环境。在许多示例中，会提供 `cluster-identifier` 以针对集群运行。在针对无服务器工作组运行时，需改为提供 `workgroup-name`。这些示例演示了一些数据 API 操作。有关更多信息，请参阅 AWS CLI 命令参考。

以下示例中的命令已被拆分和格式化以便于阅读。并非所有示例都显示了所有参数和响应。有关完整请求语法、请求参数、响应语法和响应元素的 API 定义，请参阅 Amazon Redshift Data API Reference。

### 将 SQL 语句传递到 Amazon Redshift 数据仓库

本页中的示例讲述了将 SQL 语句传递到数据仓库的不同方法

运行 SQL 语句

要运行 SQL 语句，请使用 `aws redshift-data execute-statement` AWS CLI 命令。

（可选）使用 `WaitTimeSeconds` 参数以留出额外的时间，让语句在 API 返回响应之前完成。语句完成后，响应还包括 `Status`、`RedshiftPid` 和 `HasResultSet`。有关更多信息，请参阅 使用长轮询减少 API 调用。

以下 AWS CLI 命令针对集群运行一个 SQL 语句，并返回一个标识符来获取结果。此示例使用 AWS Secrets Manager 身份验证方法。

```
aws redshift-data execute-statement
    --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
    --cluster-identifier mycluster-test
    --sql "select * from stl_query limit 1"
    --database dev
```

以下为响应示例。

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": 1598323175.823,
    "Database": "dev",
    "Id": "c016234e-5c6c-4bc5-bb16-2c5b8ff61814",
    "SecretArn": "arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn"
}
```

以下 AWS CLI 命令针对集群运行一个 SQL 语句，并返回一个标识符来获取结果。此示例使用临时凭证身份验证方法。

```
aws redshift-data execute-statement
    --db-user myuser
    --cluster-identifier mycluster-test
    --database dev
    --sql "select * from stl_query limit 1"
```

以下为响应示例。

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": 1598306924.632,
    "Database": "dev",
    "DbUser": "myuser",
    "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
}
```

以下 AWS CLI 命令针对无服务器工作组运行一个 SQL 语句，并返回一个标识符来获取结果。此示例使用临时凭证身份验证方法。

```
aws redshift-data execute-statement
    --database dev
    --workgroup-name myworkgroup
    --sql "select 1;"
```

以下为响应示例。

```
{
 "CreatedAt": "2022-02-11T06:25:28.748000+00:00",
 "Database": "dev",
 "DbUser": "IAMR:RoleName",
 "Id": "89dd91f5-2d43-43d3-8461-f33aa093c41e",
 "WorkgroupName": "myworkgroup"
}
```

以下 AWS CLI 命令针对集群运行一个 SQL 语句，并返回一个标识符来获取结果。此示例使用 AWS Secrets Manager 身份验证方法和幂等性令牌。

```
aws redshift-data execute-statement
    --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
    --cluster-identifier mycluster-test
    --sql "select * from stl_query limit 1"
    --database dev
    --client-token b855dced-259b-444c-bc7b-d3e8e33f94g1
```

以下为响应示例。

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": 1598323175.823,
    "Database": "dev",
    "Id": "c016234e-5c6c-4bc5-bb16-2c5b8ff61814",
    "SecretArn": "arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn"
}
```

运行多条 SQL 语句

要使用一个命令运行多个 SQL 语句，请使用 `aws redshift-data batch-execute-statement` AWS CLI 命令。

（可选）使用 `WaitTimeSeconds` 参数以留出额外的时间，让整个批次在 API 返回响应之前完成。有关更多信息，请参阅 使用长轮询减少 API 调用。

以下 AWS CLI 命令针对集群运行三个 SQL 语句，并返回一个标识符来获取结果。此示例使用临时凭证身份验证方法。

```
aws redshift-data batch-execute-statement
    --db-user myuser
    --cluster-identifier mycluster-test
    --database dev
    --sqls "set timezone to BST" "select * from mytable" "select * from another_table"
```

以下为响应示例。

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": 1598306924.632,
    "Database": "dev",
    "DbUser": "myuser",
    "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
}
```

以下 AWS CLI 命令使用 `AUTO_COMMIT` 执行模式在单独的事务中运行三个 SQL 语句。每个语句都是独立提交的，即一条语句失败不会阻止其余语句运行，也不会影响已经完成的语句。

```
aws redshift-data batch-execute-statement
    --db-user myuser
    --cluster-identifier mycluster-test
    --database dev
    --sqls "insert into mytable values (1)" "invalid sql statement" "insert into
 mytable values (3)"
    --execution-mode AUTO_COMMIT
```

以下为响应示例。

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": 1623979777.126,
    "Duration": 6591877,
    "HasResultSet": false,
    "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652",
    "RedshiftPid": 31459,
    "RedshiftQueryId": 0,
    "Status": "FAILED",
    "SubStatements": [
        {
            "CreatedAt": 1623979777.274,
            "Duration": 3396637,
            "HasResultSet": false,
            "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:1",
            "QueryString": "insert into mytable values (1)",
            "RedshiftQueryId": -1,
            "Status": "FINISHED",
            "UpdatedAt": 1623979777.903
        },
        {
            "CreatedAt": 1623979777.274,
            "Duration": 1195240,
            "HasResultSet": false,
            "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2",
            "QueryString": "invalid sql statement",
            "RedshiftQueryId": -1,
            "Error": "ERROR: syntax error at or near \"invalid\"",
            "Status": "FAILED",
            "UpdatedAt": 1623979777.503
        },
        {
            "CreatedAt": 1623979777.274,
            "Duration": 1095981,
            "HasResultSet": false,
            "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:3",
            "QueryString": "insert into mytable values (3)",
            "RedshiftQueryId": -1,
            "Status": "FINISHED",
            "UpdatedAt": 1623979777.903
        }
    ],
    "UpdatedAt": 1623979778.183
```

```
}
```

运行带有参数的语句

您可以将命名参数与 execute-statement 和 batch-execute-statement 操作一起使用。使用 batch-execute-statement 时，参数将在批处理中的所有 SQL 语句之间共享。每个 SQL 语句都可以引用所提供参数的子集，但每个参数都必须由至少一个 SQL 语句使用。

以下 AWS CLI 命令针对集群运行一个 SQL 语句，并返回一个标识符来获取结果。此示例使用 AWS Secrets Manager 身份验证方法。SQL 文本具有命名参数 distance。在这种情况下，在谓词中使用的距离是 5。在 SELECT 语句中，列名的命名参数只能在谓词中使用。SQL 语句的命名参数值在 parameters 选项中指定。

```
aws redshift-data execute-statement
    --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
    --cluster-identifier mycluster-test
    --sql "SELECT ratecode FROM demo_table WHERE trip_distance > :distance"
    --parameters "[{\"name\": \"distance\", \"value\": \"5\"}]"
    --database dev
```

以下为响应示例。

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": 1598323175.823,
    "Database": "dev",
    "Id": "c016234e-5c6c-4bc5-bb16-2c5b8ff61814",
    "SecretArn": "arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn"
}
```

以下示例使用示例数据库中的 EVENT 表。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 EVENT 表。

如果您的数据库中没有 EVENT 表，则可以使用数据 API 创建一个，如下所示：

```
aws redshift-data execute-statement
--database dev
--cluster-id mycluster-test
```

```
--db-user awsuser
--sql "create table event( eventid integer not null distkey,
                           venueid smallint not null,
                           catid smallint not null,
                           dateid smallint not null sortkey,
                           eventname varchar(200),
                           starttime timestamp)"
```

以下命令将一个行插入 EVENT 表。

```
aws redshift-data execute-statement
--database dev
--cluster-id mycluster-test
--db-user awsuser
--sql "insert into event
 values(:eventid, :venueid::smallint, :catid, :dateid, :eventname, :starttime)"
--parameters "[{\"name\": \"eventid\", \"value\": \"1\"}, {\"name\": \"venueid\",
 \"value\": \"1\"},
               {\"name\": \"catid\", \"value\": \"1\"},
               {\"name\": \"dateid\", \"value\": \"1\"},
               {\"name\": \"eventname\", \"value\": \"event 1\"},
               {\"name\": \"starttime\", \"value\": \"2022-02-22\"}]"
```

以下命令将另一个行插入 EVENT 表。该示例演示以下内容：

- 名为 id 参数在 SQL 文本中使用了四次。
- 插入参数 starttime 时自动应用隐式类型转换。
- venueid 列是转换为 SMALLINT 数据类型的类型。
- 表示 DATE 数据类型的字符串将隐式转换为 TIMESTAMP 数据类型。
- 注释可以在 SQL 文本中使用。

```
aws redshift-data execute-statement
--database dev
--cluster-id mycluster-test
--db-user awsuser
--sql "insert into event values(:id, :id::smallint, :id, :id, :eventname, :starttime) /
*this is comment, and it won't apply parameterization for :id, :eventname or :starttime
 here*/"
```

```
--parameters "[{\"name\": \"eventname\", \"value\": \"event 2\"},
               {\"name\": \"starttime\", \"value\": \"2022-02-22\"},
               {\"name\": \"id\", \"value\": \"2\"}]"
```

下面显示了两个插入的行：

```
 eventid | venueid | catid | dateid | eventname |      starttime
---------+---------+-------+--------+-----------+---------------------
       1 |       1 |     1 |      1 | event 1   | 2022-02-22 00:00:00
       2 |       2 |     2 |      2 | event 2   | 2022-02-22 00:00:00
```

以下命令使用 WHERE 子句中的命名参数来检索 eventid 为 1 的行。

```
aws redshift-data execute-statement
--database dev
--cluster-id mycluster-test
--db-user awsuser
--sql "select * from event where eventid=:id"
--parameters "[{\"name\": \"id\", \"value\": \"1\"}]"
```

运行以下命令以获取上一个 SQL 语句的 SQL 结果：

```
aws redshift-data get-statement-result --id 7529ad05-b905-4d71-9ec6-8b333836eb5a
```

提供以下结果：

```
{
    "Records": [
        [
            {
                "longValue": 1
            },
            {
                "longValue": 1
```

```
            },
            {
                "longValue": 1
            },
            {
                "longValue": 1
            },
            {
                "stringValue": "event 1"
            },
            {
                "stringValue": "2022-02-22 00:00:00.0"
            }
        ]
    ],
    "ColumnMetadata": [
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "eventid",
            "length": 0,
            "name": "eventid",
            "nullable": 0,
            "precision": 10,
            "scale": 0,
            "schemaName": "public",
            "tableName": "event",
            "typeName": "int4"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "venueid",
            "length": 0,
            "name": "venueid",
            "nullable": 0,
            "precision": 5,
            "scale": 0,
            "schemaName": "public",
            "tableName": "event",
            "typeName": "int2"
        },
```

```
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "catid",
            "length": 0,
            "name": "catid",
            "nullable": 0,
            "precision": 5,
            "scale": 0,
            "schemaName": "public",
            "tableName": "event",
            "typeName": "int2"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "dateid",
            "length": 0,
            "name": "dateid",
            "nullable": 0,
            "precision": 5,
            "scale": 0,
            "schemaName": "public",
            "tableName": "event",
            "typeName": "int2"
        },
        {
            "isCaseSensitive": true,
            "isCurrency": false,
            "isSigned": false,
            "label": "eventname",
            "length": 0,
            "name": "eventname",
            "nullable": 1,
            "precision": 200,
            "scale": 0,
            "schemaName": "public",
            "tableName": "event",
            "typeName": "varchar"
        },
        {
            "isCaseSensitive": false,
```

```
            "isCurrency": false,
            "isSigned": false,
            "label": "starttime",
            "length": 0,
            "name": "starttime",
            "nullable": 1,
            "precision": 29,
            "scale": 6,
            "schemaName": "public",
            "tableName": "event",
            "typeName": "timestamp"
        }
    ],
    "TotalNumRows": 1
}
```

以下 AWS CLI 命令使用 batch-execute-statement，对集群运行两个具有共享命名参数的 SQL 语句。SQL 文本具有命名参数 id 和 name。第一个 SQL 语句使用 id 参数，第二个 SQL 语句使用 name 参数。

```
aws redshift-data batch-execute-statement
    --db-user myuser
    --cluster-identifier mycluster-test
    --database dev
    --sqls "SELECT * FROM mytable WHERE id = :id" "SELECT * FROM mytable WHERE name
 = :name"
    --parameters "[{\"name\": \"id\", \"value\": \"1\"},{\"name\": \"name\", \"value\":
 \"Alice\"}]"
```

以下为响应示例。

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": 1598306924.632,
    "Database": "dev",
    "DbUser": "myuser",
    "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
}
```

以下 AWS CLI 命令运行两个 SQL 语句，这些语句使用相同的参数。名为 id 的参数用在 INSERT 语句和 SELECT 语句中。

```
aws redshift-data batch-execute-statement
    --db-user myuser
    --cluster-identifier mycluster-test
    --database dev
    --sqls "INSERT INTO mytable VALUES(:id, :name, :address)" "SELECT * FROM mytable
 WHERE id = :id"
    --parameters "[{\"name\": \"id\", \"value\": \"1\"},{\"name\": \"name\", \"value\":
 \"Alice\"},{\"name\": \"address\", \"value\": \"Seattle\"}]"
```

以下为响应示例。

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": 1598306924.632,
    "Database": "dev",
    "DbUser": "myuser",
    "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
}
```

## 列出有关 SQL 语句的元数据

要列出有关 SQL 语句的元数据，请使用 `aws redshift-data list-statements` AWS CLI 命令。运行此命令的授权基于调用者的 IAM 权限。

以下 AWS CLI 命令列出了运行的 SQL 语句。

```
aws redshift-data list-statements
    --status ALL
```

以下为响应示例。

```
{
    "Statements": [
        {
            "CreatedAt": 1598306924.632,
            "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766",
            "QueryString": "select * from stl_query limit 1",
            "Status": "FINISHED",
            "UpdatedAt": 1598306926.667
```

```
        },
        {
            "CreatedAt": 1598311717.437,
            "Id": "e0ebd578-58b3-46cc-8e52-8163fd7e01aa",
            "QueryString": "select * from stl_query limit 1",
            "Status": "FAILED",
            "UpdatedAt": 1598311719.008
        },
        {
            "CreatedAt": 1598313683.65,
            "Id": "c361d4f7-8c53-4343-8c45-6b2b1166330c",
            "QueryString": "select * from stl_query limit 1",
            "Status": "ABORTED",
            "UpdatedAt": 1598313685.495
        },
        {
            "CreatedAt": 1598306653.333,
            "Id": "a512b7bd-98c7-45d5-985b-a715f3cfde7f",
            "QueryString": "select 1",
            "Status": "FINISHED",
            "UpdatedAt": 1598306653.992
        }
    ]
}
```

## 列出会话

要列出您在过去 24 小时内创建的会话，请使用 `aws redshift-data list-sessions` AWS CLI 命令。默认情况下，该命令仅返回状态为 AVAILABLE 或 BUSY 的会话。您可以按会话状态、计算目标（集群或无服务器工作组）或数据库筛选结果。您的 IAM 权限决定您是否可以运行此命令。有关会话的更多信息，请参阅在调用 Amazon Redshift 数据 API 时运行带有会话重用的 SQL 语句。

以下 AWS CLI 命令列出了在集群上运行的会话：

```
aws redshift-data list-sessions
   --cluster-identifier mycluster-test
```

以下为响应示例：

```
{
    "Sessions": [
```

```
        {
            "SessionId": "5a254dc6-4fc2-4203-87a8-551155432ee4",
            "Status": "AVAILABLE",
            "CreatedAt": 1703022996.436,
            "UpdatedAt": 1703023010.221,
            "Database": "dev",
            "DbUser": "awsuser",
            "ClusterIdentifier": "mycluster-test",
            "SessionAliveSeconds": 10,
            "SessionTtl": 1703023020.436
        },
        {
            "SessionId": "8b365ed7-5gd7-5314-98b9-662266543ff5",
            "Status": "BUSY",
            "CreatedAt": 1703023100.512,
            "UpdatedAt": 1703023150.874,
            "Database": "dev",
            "DbUser": "awsuser",
            "ClusterIdentifier": "mycluster-test",
            "SessionAliveSeconds": 60,
            "SessionTtl": 1703023210.512,
            "CurrentStatementId": "c016234e-5c6c-4bc5-bb16-2c5b8ff61814"
        }
    ]
}
```

以下 AWS CLI 命令通过筛选会话状态列出无服务器工作组上的 CLOSED 会话：

```
aws redshift-data list-sessions
    --workgroup-name myworkgroup
    --status CLOSED
```

以下为响应示例：

```
{
    "Sessions": [
        {
            "SessionId": "07c5ffea-76d6-4786-b62c-4fe3ef529680",
            "Status": "CLOSED",
            "CreatedAt": 1703019996.436,
            "UpdatedAt": 1703020056.436,
            "Database": "dev",
```

```
            "DbUser": "IAMR:RoleName",
            "WorkgroupName": "myworkgroup",
            "SessionAliveSeconds": 60,
            "SessionTtl": 1703020056.436
        }
    ]
}
```

> **Note**
>
> 如果您使用身份增强型 IAM 角色会话，则必须提供 `cluster-identifier` 或 `workgroup-name` 参数。使用此参数，您可以确保 AWS IAM Identity Center 用户只能访问分配给他们的 Amazon Redshift IAM Identity Center 应用程序。有关更多信息，请参阅 将数据 API 与可信身份传播结合使用。

## 描述有关 SQL 语句的元数据

要获取 SQL 语句的元数据描述，请使用 `aws redshift-data describe-statement` AWS CLI 命令。还可以使用此命令通过提供子语句 ID 来描述批次中的各个子语句。运行此命令的授权基于调用者的 IAM 权限。

或者，要减少轮询请求的数量，请使用 `WaitTimeSeconds` 参数留出额外的时间，以便在此 API 调用中更新操作的状态。有关更多信息，请参阅 使用长轮询减少 API 调用。

以下 AWS CLI 命令描述 SQL 语句。

```
aws redshift-data describe-statement
    --id d9b6c0c9-0747-4bf4-b142-e8883122f766
```

以下为响应示例。

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": 1598306924.632,
    "Duration": 1095981511,
    "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766",
    "QueryString": "select * from stl_query limit 1",
    "RedshiftPid": 20859,
```

```
    "RedshiftQueryId": 48879,
    "ResultRows": 1,
    "ResultSize": 4489,
    "Status": "FINISHED",
    "UpdatedAt": 1598306926.667
}
```

以下是使用多个 SQL 语句运行 batch-execute-statement 命令后的 describe-statement 响应示例。

```
{
    "ClusterIdentifier": "mayo",
    "CreatedAt": 1623979777.126,
    "Duration": 6591877,
    "HasResultSet": true,
    "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652",
    "RedshiftPid": 31459,
    "RedshiftQueryId": 0,
    "ResultRows": 2,
    "ResultSize": 22,
    "Status": "FINISHED",
    "SubStatements": [
        {
            "CreatedAt": 1623979777.274,
            "Duration": 3396637,
            "HasResultSet": true,
            "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:1",
            "QueryString": "select 1;",
            "RedshiftQueryId": -1,
            "ResultRows": 1,
            "ResultSize": 11,
            "Status": "FINISHED",
            "UpdatedAt": 1623979777.903
        },
        {
            "CreatedAt": 1623979777.274,
            "Duration": 3195240,
            "HasResultSet": true,
            "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2",
            "QueryString": "select 2;",
            "RedshiftQueryId": -1,
            "ResultRows": 1,
            "ResultSize": 11,
```

```
            "Status": "FINISHED",
            "UpdatedAt": 1623979778.076
        }
    ],
    "UpdatedAt": 1623979778.183
}
```

以下 AWS CLI 命令通过提供子语句 ID 来描述批次中的特定子语句。

```
aws redshift-data describe-statement
    --id b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2
```

以下为响应示例。父语句状态为 STARTED，因为批次尚未完成。SubStatements 数组仅包含请求的子语句。

```
{
    "ClusterIdentifier": "mayo",
    "CreatedAt": 1623979777.126,
    "Duration": 0,
    "HasResultSet": false,
    "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652",
    "RedshiftPid": 31459,
    "RedshiftQueryId": 0,
    "Status": "STARTED",
    "SubStatements": [
        {
            "CreatedAt": 1623979777.274,
            "Duration": 3195240,
            "HasResultSet": true,
            "Id": "b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2",
            "QueryString": "select 2;",
            "RedshiftQueryId": -1,
            "ResultRows": 1,
            "ResultSize": 11,
            "Status": "FINISHED",
            "UpdatedAt": 1623979778.076
        }
    ],
    "UpdatedAt": 1623979777.503
}
```

## 获取 SQL 语句的结果

要从运行的 SQL 语句中获取结果，请使用 `redshift-data get-statement-result` 或 `redshift-data get-statement-result-v2` AWS CLI 命令。来自 `get-statement-result` 的结果采用 JSON 格式。来自 `get-statement-result-v2` 的结果采用 CSV 格式。您可以提供响应 `execute-statement` 或者 `batch-execute-statement` 而收到的 `Id`。可以在 `describe-statement` 的结果中检索 `batch-execute-statement` 运行的 SQL 语句的 `Id` 值，并以冒号和 `b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2` 等序号为该值添加后缀。如果使用 `batch-execute-statement` 运行多个 SQL 语句，则每个 SQL 语句都有一个 `Id` 值，如 `describe-statement` 中所示。运行此命令的授权基于调用者的 IAM 权限。

（可选）要减少轮询请求的数量，请使用 `WaitTimeSeconds` 参数以留出额外的时间，让语句在 API 返回语句结果之前完成。有关更多信息，请参阅 使用长轮询减少 API 调用。

以下语句返回 `execute-statement` 运行的 SQL 语句的结果，该语句让 `ResultFormat` 默认为 `JSON`。要检索结果，请调用 `get-statement-result` 操作。

```
aws redshift-data get-statement-result
    --id d9b6c0c9-0747-4bf4-b142-e8883122f766
```

以下语句返回 `batch-execute-statement` 运行的第二个 SQL 语句的结果。

```
aws redshift-data get-statement-result
    --id b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2
```

以下是调用 `get-statement-result` 的响应示例，其中在响应的 `Records` 键中以 JSON 格式返回 SQL 结果。

```
{
    "ColumnMetadata": [
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "userid",
            "length": 0,
            "name": "userid",
            "nullable": 0,
            "precision": 10,
```

```
            "scale": 0,
            "schemaName": "",
            "tableName": "stll_query",
            "typeName": "int4"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "query",
            "length": 0,
            "name": "query",
            "nullable": 0,
            "precision": 10,
            "scale": 0,
            "schemaName": "",
            "tableName": "stll_query",
            "typeName": "int4"
        },
        {
            "isCaseSensitive": true,
            "isCurrency": false,
            "isSigned": false,
            "label": "label",
            "length": 0,
            "name": "label",
            "nullable": 0,
            "precision": 320,
            "scale": 0,
            "schemaName": "",
            "tableName": "stll_query",
            "typeName": "bpchar"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "xid",
            "length": 0,
            "name": "xid",
            "nullable": 0,
            "precision": 19,
            "scale": 0,
            "schemaName": "",
```

```
            "tableName": "stll_query",
            "typeName": "int8"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "pid",
            "length": 0,
            "name": "pid",
            "nullable": 0,
            "precision": 10,
            "scale": 0,
            "schemaName": "",
            "tableName": "stll_query",
            "typeName": "int4"
        },
        {
            "isCaseSensitive": true,
            "isCurrency": false,
            "isSigned": false,
            "label": "database",
            "length": 0,
            "name": "database",
            "nullable": 0,
            "precision": 32,
            "scale": 0,
            "schemaName": "",
            "tableName": "stll_query",
            "typeName": "bpchar"
        },
        {
            "isCaseSensitive": true,
            "isCurrency": false,
            "isSigned": false,
            "label": "querytxt",
            "length": 0,
            "name": "querytxt",
            "nullable": 0,
            "precision": 4000,
            "scale": 0,
            "schemaName": "",
            "tableName": "stll_query",
            "typeName": "bpchar"
```

```
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": false,
            "label": "starttime",
            "length": 0,
            "name": "starttime",
            "nullable": 0,
            "precision": 29,
            "scale": 6,
            "schemaName": "",
            "tableName": "stll_query",
            "typeName": "timestamp"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": false,
            "label": "endtime",
            "length": 0,
            "name": "endtime",
            "nullable": 0,
            "precision": 29,
            "scale": 6,
            "schemaName": "",
            "tableName": "stll_query",
            "type": 93,
            "typeName": "timestamp"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "aborted",
            "length": 0,
            "name": "aborted",
            "nullable": 0,
            "precision": 10,
            "scale": 0,
            "schemaName": "",
            "tableName": "stll_query",
            "typeName": "int4"
        },
```

```
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "insert_pristine",
            "length": 0,
            "name": "insert_pristine",
            "nullable": 0,
            "precision": 10,
            "scale": 0,
            "schemaName": "",
            "tableName": "stll_query",
            "typeName": "int4"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "concurrency_scaling_status",
            "length": 0,
            "name": "concurrency_scaling_status",
            "nullable": 0,
            "precision": 10,
            "scale": 0,
            "schemaName": "",
            "tableName": "stll_query",
            "typeName": "int4"
        }
    ],
    "Records": [
        [
            {
                "longValue": 1
            },
            {
                "longValue": 3
            },
            {
                "stringValue": "health"
            },
            {
                "longValue": 1023
            },
            {
```

```
                "longValue": 15279
            },
            {
                "stringValue": "dev"
            },
            {
                "stringValue": "select system_status from stv_gui_status;"
            },
            {
                "stringValue": "2020-08-21 17:33:51.88712"
            },
            {
                "stringValue": "2020-08-21 17:33:52.974306"
            },
            {
                "longValue": 0
            },
            {
                "longValue": 0
            },
            {
                "longValue": 6
            }
        ]
    ],
    "TotalNumRows": 1
}
```

以下示例显示了 execute-statement 运行的一条 SQL 语句，该语句以 JSON 格式返回结果。表 testingtable 有三个整数列（col1、col2、col3），还有三行的值分别为（1、2、3）、（4、5、6）和（7、8、9）。

```
aws redshift-data execute-statement
    --database dev
    --sql "SELECT col1, col2, col3 FROM testingtable"
    --cluster-id mycluster-test
    --result-format JSON
```

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": "2024-04-02T16:45:25.144000+00:00",
```

```
    "Database": "dev",
    "DbUser": "IAMR:Administrator",
    "Id": "d468d942-6df9-4f85-8ae3-bac01a61aec3"
}
```

以下是调用 get-statement-result 的响应示例，其中在响应的 Records 键中以 JSON 格式返回 SQL 结果。

```
aws redshift-data get-statement-result
    --id d468d942-6df9-4f85-8ae3-bac01a61aec3
```

```
{
    "Records": [
        [
            {
                "longValue": 1
            },
            {
                "longValue": 2
            },
            {
                "longValue": 3
            }
        ],
        [
            {
                "longValue": 4
            },
            {
                "longValue": 5
            },
            {
                "longValue": 6
            }
        ],
        [
            {
                "longValue": 7
            },
            {
                "longValue": 8
            },
```

```
            {
                "longValue": 9
            }
        ]
    ],
    "ColumnMetadata": [
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "col1",
            "name": "col1",
            "nullable": 1,
            "precision": 10,
            "scale": 0,
            "schemaName": "public",
            "tableName": "testingtable",
            "typeName": "int4",
            "length": 0
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "col2",
            "name": "col2",
            "nullable": 1,
            "precision": 10,
            "scale": 0,
            "schemaName": "public",
            "tableName": "testingtable",
            "typeName": "int4",
            "length": 0
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "col3",
            "name": "col3",
            "nullable": 1,
            "precision": 10,
            "scale": 0,
            "schemaName": "public",
```

```
            "tableName": "testingtable",
            "typeName": "int4",
            "length": 0
        }
    ],
    "TotalNumRows": 3
}
```

以下示例显示了 execute-statement 运行的一条 SQL 语句，该语句以 CSV 格式返回结果。表 testingtable 有三个整数列（col1、col2、col3），还有三行的值分别为（1、2、3）、（4、5、6）和（7、8、9）。

```
aws redshift-data execute-statement
    --database dev
    --sql "SELECT col1, col2, col3 FROM testingtable"
    --cluster-id mycluster-test
    --result-format CSV
```

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": "2024-04-02T16:45:25.144000+00:00",
    "Database": "dev",
    "DbUser": "IAMR:Administrator",
    "Id": "d468d942-6df9-4f85-8ae3-bac01a61aec3"
}
```

以下是调用 get-statement-result-v2 的响应示例，其中在响应的 Records 键中以 CSV 格式返回 SQL 结果。行由回车符和换行符（\r\n）分隔。Records 中返回的第一行是列标题。以 CSV 格式返回的结果以 1 MB 为单位返回，其中每个分块可以存储任意数量的行，最大为 1MB。

```
aws redshift-data get-statement-result-v2
    --id d468d942-6df9-4f85-8ae3-bac01a61aec3
```

```
{
    "Records": [
        {
            "CSVRecords": "col1,col2,col3\r\n1,2,3\r\n4,5,6\r\n7,8,9\r\n"
        }
    ],
```

```
    "ColumnMetadata": [
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "col1",
            "name": "col1",
            "nullable": 1,
            "precision": 10,
            "scale": 0,
            "schemaName": "public",
            "tableName": "testingtable",
            "typeName": "int4",
            "length": 0
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "col2",
            "name": "col2",
            "nullable": 1,
            "precision": 10,
            "scale": 0,
            "schemaName": "public",
            "tableName": "testingtable",
            "typeName": "int4",
            "length": 0
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": true,
            "label": "col3",
            "name": "col3",
            "nullable": 1,
            "precision": 10,
            "scale": 0,
            "schemaName": "public",
            "tableName": "testingtable",
            "typeName": "int4",
            "length": 0
        }
    ],
```

```
    "TotalNumRows": 3,
    "ResultFormat": "csv"
}
```

## 使用长轮询减少 API 调用

默认情况下，数据 API 操作会立即返回响应。如果操作针对正在进行的语句，则必须反复轮询，直到该语句达到终止状态。使用长轮询时，将 `WaitTimeSeconds` 设置为介于 1 和 30 之间的值。API 将延迟返回同步响应，直到语句完成或达到 `WaitTimeSeconds`（以较早者为准）。

DescribeStatement、ExecuteStatement、BatchExecuteStatement、GetStatementResult 和 GetStatementResultV2 支持长轮询。

当操作使用 `WaitTimeSeconds` 来提交新语句或等待现有语句时：

- 如果语句在等待时间内完成，则响应将包括终止状态或结果数据，具体取决于操作。
- 如果等待时间到期时语句仍在运行，则响应将返回当前正在进行状态。
- 如果语句在长轮询请求发出之前已经完成，则 API 会立即返回终止状态。
- 如果省略了 `WaitTimeSeconds`，则保留默认的异步行为。

### 常见使用案例

以下是长轮询可能有用的示例：

- 提交简短的查询并等待完成：将 `ExecuteStatement` 与 `WaitTimeSeconds` 结合使用。当查询完成或等待时间到期时，响应会返回语句 `Status`、`RedshiftPid` 和 `HasResultSet`。要获取完整的语句元数据，请调用 `DescribeStatement`。如果 `HasResultSet` 为 `true`，则调用 `GetStatementResult` 以提取结果。
- 长轮询以获取状态：使用 `ExecuteStatement` 以提交查询，然后使用 `WaitTimeSeconds` 调用 `DescribeStatement`。查询完成后，`DescribeStatement` 返回完整的语句元数据，包括状态、持续时间和错误详情。
- 长轮询以获取结果：使用 `ExecuteStatement` 以提交查询，然后使用 `WaitTimeSeconds` 调用 `GetStatementResult`。查询完成后，将直接返回结果数据。如果等待时间到期时查询仍在运行，则会返回 `ResourceNotFoundException`，表示尚无可用的结果。
- 提交多个查询并等待整个批次完成：将 `BatchExecuteStatement` 与 `WaitTimeSeconds` 结合使用。该调用会保持打开状态，直至所有子语句执行完毕，然后返回批次父 ID 和整体状态。

- 为子语句提交多个查询和长轮询状态：将 DescribeStatement 或 GetStatementResult 与子语句 ID 和 WaitTimeSeconds 结合使用。该子语句完成后立即返回，无需等待批次的其余部分。

## 描述表

要获取描述表的元数据，请使用 aws redshift-data describe-table AWS CLI 命令。

以下 AWS CLI 命令针对集群运行一个 SQL 语句，并返回描述表的元数据。此示例使用 AWS Secrets Manager 身份验证方法。

```
aws redshift-data describe-table
    --cluster-identifier mycluster-test
    --database dev
    --schema information_schema
    --table sql_features
    --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
```

以下为响应示例。

```
{
    "ColumnList": [
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": false,
            "length": 2147483647,
            "name": "feature_id",
            "nullable": 1,
            "precision": 2147483647,
            "scale": 0,
            "schemaName": "information_schema",
            "tableName": "sql_features",
            "typeName": "character_data"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": false,
            "length": 2147483647,
            "name": "feature_name",
```

```
            "nullable": 1,
            "precision": 2147483647,
            "scale": 0,
            "schemaName": "information_schema",
            "tableName": "sql_features",
            "typeName": "character_data"
        }
    ]
}
```

以下 AWS CLI 命令针对集群运行一个 SQL 语句来描述表。此示例使用临时凭证身份验证方法。

```
aws redshift-data describe-table
    --db-user myuser
    --cluster-identifier mycluster-test
    --database dev
    --schema information_schema
    --table sql_features
```

以下为响应示例。

```
{
    "ColumnList": [
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": false,
            "length": 2147483647,
            "name": "feature_id",
            "nullable": 1,
            "precision": 2147483647,
            "scale": 0,
            "schemaName": "information_schema",
            "tableName": "sql_features",
            "typeName": "character_data"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": false,
            "length": 2147483647,
            "name": "feature_name",
```

```
            "nullable": 1,
            "precision": 2147483647,
            "scale": 0,
            "schemaName": "information_schema",
            "tableName": "sql_features",
            "typeName": "character_data"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": false,
            "length": 2147483647,
            "name": "sub_feature_id",
            "nullable": 1,
            "precision": 2147483647,
            "scale": 0,
            "schemaName": "information_schema",
            "tableName": "sql_features",
            "typeName": "character_data"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": false,
            "length": 2147483647,
            "name": "sub_feature_name",
            "nullable": 1,
            "precision": 2147483647,
            "scale": 0,
            "schemaName": "information_schema",
            "tableName": "sql_features",
            "typeName": "character_data"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": false,
            "length": 2147483647,
            "name": "is_supported",
            "nullable": 1,
            "precision": 2147483647,
            "scale": 0,
            "schemaName": "information_schema",
            "tableName": "sql_features",
```

```
            "typeName": "character_data"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": false,
            "length": 2147483647,
            "name": "is_verified_by",
            "nullable": 1,
            "precision": 2147483647,
            "scale": 0,
            "schemaName": "information_schema",
            "tableName": "sql_features",
            "typeName": "character_data"
        },
        {
            "isCaseSensitive": false,
            "isCurrency": false,
            "isSigned": false,
            "length": 2147483647,
            "name": "comments",
            "nullable": 1,
            "precision": 2147483647,
            "scale": 0,
            "schemaName": "information_schema",
            "tableName": "sql_features",
            "typeName": "character_data"
        }
    ]
}
```

## 列出集群中的数据库

要列出集群中的数据库，请使用 `aws redshift-data list-databases` AWS CLI 命令。

以下 AWS CLI 命令针对集群运行一个 SQL 语句来列出数据库。此示例使用 AWS Secrets Manager 身份验证方法。

```
aws redshift-data list-databases

    --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
```

```
    --cluster-identifier mycluster-test
    --database dev
```

以下为响应示例。

```
{
    "Databases": [
        "dev"
    ]
}
```

以下 AWS CLI 命令针对集群运行一个 SQL 语句来列出数据库。此示例使用临时凭证身份验证方法。

```
aws redshift-data list-databases
    --db-user myuser
    --cluster-identifier mycluster-test
    --database dev
```

以下为响应示例。

```
{
    "Databases": [
        "dev"
    ]
}
```

## 列出数据库中的架构

要列出数据库中的 schema，请使用 `aws redshift-data list-schemas` AWS CLI 命令。

以下 AWS CLI 命令针对集群运行一个 SQL 语句来列出数据库中的架构。此示例使用 AWS Secrets Manager 身份验证方法。

```
aws redshift-data list-schemas
    --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
    --cluster-identifier mycluster-test
    --database dev
```

以下为响应示例。

```
{
    "Schemas": [
        "information_schema",
        "pg_catalog",
        "pg_internal",
        "public"
    ]
}
```

以下 AWS CLI 命令针对集群运行一个 SQL 语句来列出数据库中的架构。此示例使用临时凭证身份验证方法。

```
aws redshift-data list-schemas
    --db-user mysuser
    --cluster-identifier mycluster-test
    --database dev
```

以下为响应示例。

```
{
    "Schemas": [
        "information_schema",
        "pg_catalog",
        "pg_internal",
        "public"
    ]
}
```

## 列出数据库中的表

要列出数据库中的表，请使用 `aws redshift-data list-tables` AWS CLI 命令。

以下 AWS CLI 命令针对集群运行一个 SQL 语句来列出数据库中的表。此示例使用 AWS Secrets Manager 身份验证方法。

```
aws redshift-data list-tables
```

```
    --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
    --cluster-identifier mycluster-test
    --database dev
    --schema information_schema
```

以下为响应示例。

```
{
    "Tables": [
        {
            "name": "sql_features",
            "schema": "information_schema",
            "type": "SYSTEM TABLE"
        },
        {
            "name": "sql_implementation_info",
            "schema": "information_schema",
            "type": "SYSTEM TABLE"
        }
}
```

以下 AWS CLI 命令针对集群运行一个 SQL 语句来列出数据库中的表。此示例使用临时凭证身份验证方法。

```
aws redshift-data list-tables

     --db-user myuser
     --cluster-identifier mycluster-test
     --database dev
     --schema information_schema
```

以下为响应示例。

```
{
    "Tables": [
        {
            "name": "sql_features",
            "schema": "information_schema",
```

```
            "type": "SYSTEM TABLE"
        },
        {
            "name": "sql_implementation_info",
            "schema": "information_schema",
            "type": "SYSTEM TABLE"
        }
    ]
}
```

# Amazon Redshift 数据 API 的问题排查

使用以下部分（标题为常见错误消息）帮助解决 Data API 问题。

主题

- 用于查询的包太大
- 数据库响应超出大小限制

## 用于查询的包太大

如果您看到一条错误，指示查询的数据包过大，则为行返回的结果集通常过大。数据库返回的结果集中的 Data API 大小限制为每行 64 KB。

要解决此问题，请确保结果集中的每一行都小于或等于 64 KB。

## 数据库响应超出大小限制

如果您看到一条错误，指示数据库响应超出了大小限制，则数据库返回的结果集的大小通常太大。数据库返回的结果集中的数据 API 限制为 500 MB。

要解决此问题，请确保对数据 API 的调用返回 500 MB 或更少的数据。如果需要返回 500 MB 以上的数据，则可以在查询中将多个语句调用与 `LIMIT` 子句结合使用。

# 使用 Amazon EventBridge 计划 Amazon Redshift 数据 API 操作

您可以创建规则来匹配选定的事件，并将它们路由到目标以采取操作。此外，您还可以使用规则对预定的计划采取操作。有关更多信息，请参阅 Amazon EventBridge 用户指南。

要使用 EventBridge 计划数据 API 操作，关联的 IAM 角色必须信任 CloudWatch Events (events.amazonaws.com) 的委托人。此角色应附加相当于托管式策略

AmazonEventBridgeFullAccess 的策略。它还应具有 AmazonRedshiftDataFullAccess 策略权限，这些权限由数据 API 管理。您可以在 IAM 控制台上创建具有这些权限的 IAM 角色。在 IAM 控制台上创建角色时，为 CloudWatch Events 选择 AWS 服务可信任实体。在 EventBridge 目标的 RoleArn JSON 值中指定 IAM 角色。有关创建 IAM 角色的更多信息，请参阅《IAM 用户指南》中的为 AWS 服务创建角色（控制台）。

您在 Amazon EventBridge 中创建的规则的 name 必须与 RedshiftDataParameters 中的 StatementName 相匹配。

以下示例显示了使用单个或多个 SQL 语句创建 EventBridge 规则的变体，并将 Amazon Redshift 集群或 Amazon Redshift Serverless 工作组作为数据仓库。

对集群执行包含单个 SQL 语句的调用

以下示例使用 AWS CLI 创建 EventBridge 规则，用于对 Amazon Redshift 集群运行 SQL 语句。

```
aws events put-rule
--name test-redshift-cluster-data
--schedule-expression "rate(1 minute)"
```

然后创建一个 EventBridge 目标，以按照规则中指定的计划运行。

```
aws events put-targets
--cli-input-json file://data.json
```

输入 data.json 文件如下。Sql JSON 键表示只有一个 SQL 语句。Arn JSON 值包含集群标识符。RoleArn JSON 值包含用于运行 SQL 的 IAM 角色，如前所述。

```
{
    "Rule": "test-redshift-cluster-data",
    "EventBusName": "default",
    "Targets": [
        {
            "Id": "2",
            "Arn": "arn:aws:redshift:us-east-1:123456789012:cluster:mycluster",
            "RoleArn": "arn:aws:iam::123456789012:role/Administrator",
            "RedshiftDataParameters": {
                "Database": "dev",
                "DbUser": "root",
                "Sql": "select 1;",
                "StatementName": "test-redshift-cluster-data",
```

```
                "WithEvent": true
            }
        }
    ]
}
```

使用单个 SQL 语句和工作组进行调用

以下示例使用 AWS CLI 创建 EventBridge 规则，用于对 Amazon Redshift Serverless 工作组运行一个 SQL 语句。

```
aws events put-rule
--name  test-redshift-serverless-workgroup-data
--schedule-expression "rate(1 minute)"
```

然后创建一个 EventBridge 目标，以按照规则中指定的计划运行。

```
aws events put-targets
--cli-input-json file://data.json
```

输入 data.json 文件如下。`Sql` JSON 键表示只有一个 SQL 语句。`Arn` JSON 值包含工作组名称。`RoleArn` JSON 值包含用于运行 SQL 的 IAM 角色，如前所述。

```
{
    "Rule": "test-redshift-serverless-workgroup-data",
    "EventBusName": "default",
    "Targets": [
        {
            "Id": "2",
            "Arn": "arn:aws:redshift-serverless:us-east-1:123456789012:workgroup/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "RoleArn": "arn:aws:iam::123456789012:role/Administrator",
            "RedshiftDataParameters": {
                "Database": "dev",
                "Sql": "select 1;",
                "StatementName": "test-redshift-serverless-workgroup-data",
                "WithEvent": true
            }
        }
    ]
}
```

对集群执行包含多个 SQL 语句的调用

以下示例使用 AWS CLI 创建 EventBridge 规则，用于对 Amazon Redshift 集群运行多个 SQL 语句。

```
aws events put-rule
--name  test-redshift-cluster-data
--schedule-expression "rate(1 minute)"
```

然后创建一个 EventBridge 目标，以按照规则中指定的计划运行。

```
aws events put-targets
--cli-input-json file://data.json
```

输入 data.json 文件如下。`Sqls` JSON 键表示有多个 SQL 语句。`Arn` JSON 值包含集群标识符。`RoleArn` JSON 值包含用于运行 SQL 的 IAM 角色，如前所述。

```
{
    "Rule": "test-redshift-cluster-data",
    "EventBusName": "default",
    "Targets": [
        {
            "Id": "2",
            "Arn": "arn:aws:redshift:us-east-1:123456789012:cluster:mycluster",
            "RoleArn": "arn:aws:iam::123456789012:role/Administrator",
            "RedshiftDataParameters": {
                "Database": "dev",
                "Sqls": ["select 1;", "select 2;", "select 3;"],
                "StatementName": "test-redshift-cluster-data",
                "WithEvent": true
            }
        }
    ]
}
```

使用多个 SQL 语句和工作组进行调用

以下示例使用 AWS CLI 创建 EventBridge 规则，用于对 Amazon Redshift Serverless 工作组运行多个 SQL 语句。

```
aws events put-rule
--name  test-redshift-serverless-workgroup-data
```

```
--schedule-expression "rate(1 minute)"
```

然后创建一个 EventBridge 目标，以按照规则中指定的计划运行。

```
aws events put-targets
--cli-input-json file://data.json
```

输入 data.json 文件如下。Sqls JSON 键表示有多个 SQL 语句。Arn JSON 值包含工作组名称。RoleArn JSON 值包含用于运行 SQL 的 IAM 角色，如前所述。

```
{
    "Rule": "test-redshift-serverless-workgroup-data",
    "EventBusName": "default",
    "Targets": [
        {
            "Id": "2",
            "Arn": "arn:aws:redshift-serverless:us-east-1:123456789012:workgroup/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "RoleArn": "arn:aws:iam::123456789012:role/Administrator",
            "RedshiftDataParameters": {
                "Database": "dev",
                "Sqls": ["select 1;", "select 2;", "select 3;"],
                "StatementName": "test-redshift-serverless-workgroup-data",
                "WithEvent": true
            }
        }
    ]
}
```

## 监控数据 API

监控是保持数据 API 和您的其他 AWS 解决方案的可靠性、可用性和性能的重要方面。AWS 提供了以下一些监控工具来监控数据 API、在出现错误时进行报告并适时自动采取措施：

- 您可以使用 Amazon EventBridge 自动执行您的 AWS 服务并自动响应系统事件，例如应用程序可用性问题或资源更改。AWS 服务中的事件将近乎实时地传输到 EventBridge。您可以编写简单的规则来指示您关注的事件，并指示要在事件匹配规则时执行的自动化操作。有关更多信息，请参阅 Amazon EventBridge 用户指南。
- AWS CloudTrail 捕获由您的 AWS 账户或代表该账户发出的 API 调用和相关事件，并将日志文件传输到您指定的 Amazon S3 存储桶。您可以标识哪些用户和账户调用了 AWS、发出调用的源 IP 地址

以及调用的发生时间。要了解有关 Amazon Redshift 如何与 AWS CloudTrail 集成的更多信息，请参阅使用 Cloudtrail 进行日志记录。有关 CloudTrail 的更多信息，请参阅《AWS CloudTrail 用户指南》https://docs.aws.amazon.com/awscloudtrail/latest/userguide/。

主题

- 在 Amazon EventBridge 中监控 Amazon Redshift 数据 API 的事件

## 在 Amazon EventBridge 中监控 Amazon Redshift 数据 API 的事件

您可以在 EventBridge 中监控数据 API，这将从您自己的应用程序、软件即服务 (SaaS) 应用程序和 AWS 服务传输实时数据流。EventBridge 将该数据路由到诸如 AWS Lambda 和 Amazon SNS 之类的目标。这些事件与 CloudWatch Events 中出现的事件相同，可提供近乎实时的系统事件流，这些系统事件描述 AWS 资源的变化。事件将发送到包含 Amazon Redshift 数据库的账户。例如，如果您担任另一个账户中的角色，则事件将发送到该账户。有关更多信息，请参阅《Amazon EventBridge 用户指南》中的 Amazon EventBridge 事件。

当 `ExecuteStatement` 或 `BatchExecuteStatement` API 操作将 `WithEvent` 选项设置为 `true` 时，将发送数据 API 事件。事件的 `state` 字段包含以下值之一：

- 中止 – 用户停止了查询运行。
- FAILED – 查询运行失败。
- FINISHED – 查询已完成运行。

事件会确保送达。有关更多信息，请参阅《Amazon EventBridge 用户指南》中的 Events from AWS services。

数据 API 完成事件示例

以下示例在 `ExecuteStatement` API 操作完成时显示数据 API 的一个事件。在以下示例中，名为 `test.testtable` 的语句运行完成。

```
{
    "version": "0",
    "id": "18e7079c-dd4b-dd64-caf9-e2a31640dab0",
    "detail-type": "Redshift Data Statement Status Change",
    "source": "aws.redshift-data",
    "account": "123456789012",
    "time": "2020-10-01T21:14:26Z",
```

```
    "region": "us-east-1",
    "resources": [
        "arn:aws:redshift:us-east-1:123456789012:cluster:redshift-cluster-1"
    ],
    "detail": {
        "principal": "arn:aws:iam::123456789012:user/myuser",
        "statementName": "test.testtable",
        "statementId": "dd2e1ec9-2ee3-49a0-819f-905fa7d75a4a",
        "redshiftQueryId": -1,
        "state": "FINISHED",
        "rows": 1,
        "expireAt": 1601673265
    }
}
```

# 结合使用 AWS KMS 与 Amazon Redshift Data API

在使用客户自主管理型密钥对 Amazon Redshift 集群或 Redshift Serverless 工作组进行加密时，Amazon Redshift Data API 会使用相同的客户自主管理型密钥来存储和加密您的查询与结果。

默认情况下，Data API 会加密您的数据以保护敏感信息，例如查询文本和查询结果。它使用 AWS 拥有的 AWS KMS 加密密钥来实施此保护措施。

静态数据的默认加密可降低保护敏感数据时的运营开销和复杂性。此方法有助于构建符合严格加密合规性和监管要求的安全应用程序。

## 使用 AWS KMS 中的授权

Data API 需要授权才能使用客户自主管理型密钥。

在对已使用客户自主管理型密钥加密的集群调用 `ExecuteStatement` 或 `BatchExecuteStatement` 时，Amazon Redshift 会通过向 AWS KMS 发送 `CreateGrant` 请求来代表您创建授权。AWS KMS 通过授权向数据 API 授予对您账户中的 KMS 密钥的访问权限。

Data API 需要授权，才能将客户自主管理型密钥用于以下操作：

- 向 AWS KMS 发送 `Encrypt` 请求，以使用客户自主管理型密钥对查询元数据进行加密。
- 向 AWS KMS 发送 `GenerateDataKey` 请求，以生成由客户自主管理型密钥加密的数据密钥。
- 向 AWS KMS 发送 `Decrypt` 请求，以对已加密的数据密钥进行解密，从而使用这些密钥对您的数据进行加密。

您可以随时撤销对授权的访问权限，或者删除 Amazon Redshift 对客户自主管理型密钥的访问权限。如果您这样做，Data API 将不再能访问由客户自主管理型密钥加密的数据，这会影响依赖于该数据的操作。例如，如果您在撤销授权后尝试检索查询结果或跟踪查询状态，则 Data API 会返回 `AccessDeniedException`。

### 客户自主管理型密钥的密钥策略

密钥政策控制对客户托管密钥的访问。每个客户托管式密钥必须只有一个密钥策略，其中包含确定谁可以使用密钥以及如何使用密钥的声明。创建客户托管式密钥时，可以指定密钥策略。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的客户托管密钥。

要结合使用您的客户自主管理型密钥与 Data API，您必须先允许访问 Amazon Redshift。密钥策略中必须允许以下 API 操作：

- `kms:CreateGrant`：向客户托管密钥添加授权。这些授权控制对指定 AWS KMS 密钥的访问权限，从而允许访问 Amazon Redshift 所需的授权操作。有关更多信息，请参阅在 AWS KMS 中使用授权。

以下是示例密钥策略：

```
"Statement":[
   {
      "Sid":"Allow access to principals authorized to use Amazon Redshift",
      "Effect":"Allow",
      "Principal":{
         "AWS":"*"
      },
      "Action":[
         "kms:DescribeKey",
         "kms:CreateGrant"
      ],
      "Resource":"*",
      "Condition":{
         "StringEquals":{
            "kms:ViaService":"redshift.amazonaws.com",
            "kms:CallerAccount":"111122223333"
         }
      }
   },
   {
      "Sid":"AllowKeyAdministratorsAccess",
```

```
      "Effect":"Allow",
      "Principal":{
         "AWS":"arn:aws:iam::111122223333:role/ExampleAdminRole"
      },
      "Action":"kms:*",
      "Resource":"*"
   },
   {
      "Sid":"AllowKeyUseForExampleRole",
      "Effect":"Allow",
      "Principal":{
         "AWS":"arn:aws:iam::111122223333:role/ExampleUserRole"
      },
      "Action":[
         "kms:Encrypt",
         "kms:Decrypt",
         "kms:ReEncrypt*",
         "kms:GenerateDataKey*",
         "kms:DescribeKey"
      ],
      "Resource":"*"
   }
]
```

## Data API 加密上下文

加密上下文是一组可选的键值对，包含有关数据的其他上下文信息。AWS KMS 会将加密上下文用作其他已经过验证的数据以支持经过身份验证的加密。在请求中包含加密上下文以加密数据时，AWS KMS 将加密上下文绑定到加密的数据。要解密数据，您必须在请求中包含相同的加密上下文。

Data API 在预调配集群的所有 AWS KMS 加密操作中使用相同的三个加密上下文键值对：

- `aws:redshift:arn`：集群的 Amazon 资源名称（ARN）
- `aws:redshift:createtime`：请求创建集群时的时间戳
- `serviceName` – `RedshiftDataAPI`

```
"EncryptionContextSubset": {
    "aws:redshift:arn": "arn:aws:redshift:us-east-1:123456789012:cluster:redshift-
cluster",
    "aws:redshift:createtime": "20250815T0000Z",
    "serviceName": "RedshiftDataAPI",
```

```
}
```

Data API 在无服务器工作组的所有 AWS KMS 加密操作中使用两个加密上下文键值对：

- `aws:redshift-serverless:arn`：命名空间的 Amazon 资源名称（ARN）
- `serviceName`：RedshiftDataAPI

```
"EncryptionContextSubset": {
    "aws:redshift-serverless:arn": "arn:aws:redshift-serverless:us-
east-1:123456789012:namespace:12345678-1234-1234-1234-123456789012",
    "serviceName": "RedshiftDataAPI"
}
```

有关加密的更多信息，请参阅 AWS KMS 的加密详细信息简介。有关 Amazon Redshift 和 AWS KMS 集成的更多信息，请参阅 Amazon Redshift 使用 AWS KMS 的方式。

# 使用 Amazon SageMaker 融通式合作开发工作室在 Amazon Redshift 和 SageMaker 智能湖仓中查询您的数据库

Amazon SageMaker 融通式合作开发工作室提供一个非控制台开发环境，并支持对 SageMaker 智能湖仓、Amazon Redshift 和 Amazon Athena 中的数据进行 SQL 分析来实现 SQL 分析。使用管理员提供的 URL 导航至 Amazon SageMaker 融通式合作开发工作室，然后使用您的 SSO 或 AWS 凭证进行登录。有关设置第一个项目的更多信息，请参阅《Amazon SageMaker Unified Studio User Guide》中的 Getting started。

在 Amazon SageMaker 融通式合作开发工作室中，可以通过使用 query editor 运行 Amazon Redshift 和 Amazon Athena 来执行 SQL analytics。使用查询编辑器来编写和运行查询、查看结果以及与团队共享您的工作。在您的 AWS 账户中（在同一账户内和跨其它 AWS 账户）针对 Redshift 数据仓库运行查询，使用相同的界面为 Redshift 和 Athena 构建 SQL 查询，并使用 Amazon Managed Workflows for Apache Airflow 安排 SQL 查询。还可以使用 Amazon Q 生成式 SQL，根据自然语言生成 SQL。

# Amazon Redshift 参数组

在 Amazon Redshift 中，您可以将参数组与创建的每个集群相关联。参数组是一组适用于您在集群中创建的所有数据库的参数。这些参数可用于配置数据库设置，例如查询超时和日期样式。当您启动集群时，您必须将它与一个参数组相关联。如果您在以后要更改参数组，可以修改集群并选择其他参数组。

每个参数组都拥有几个可用于配置数据库设置的参数。可用参数列表取决于参数组所属的参数组系列。默认参数组系列为 `redshift-2.0`。

Amazon Redshift 为每个参数组系列提供一个默认的参数组。默认参数组中的每个参数都有预设值，并且无法修改。默认参数组名称的格式为 `default.`*`parameter_group_family`*。例如，`redshift-2.0` 参数组系列的默认参数组为 `default.redshift-2.0`。

如果您想要使用与默认参数组不同的参数值，则必须创建自定义参数组，然后将其与您的集群相关联。最初，自定义参数组中的参数值与默认参数组中的参数值相同。由于值由 Amazon Redshift 预先设置，因此所有参数的初始 `source` 均为 `engine-default`。在您更改参数值之后，`source` 将变为 `user`，表明默认值被修改为现在的值。

> **Note**
>
> Amazon Redshift 控制台不会显示每个参数的 `source`。您必须使用Amazon Redshift API、AWS CLI、或其中一个 AWS 开发工具包才能查看 `source`。

对于您创建的参数组，您可以随时修改参数值，也可以将所有参数值重置为默认值。您还可以将其他参数组与某个集群相关联。在某些情况下，您可以修改与集群关联的参数组中的参数值，或者将其他参数组与集群关联。在这些情况下，您可能需要重新启动集群以使更新后的参数值生效。如果该集群发生故障并由 Amazon Redshift 重新启动，那么系统将在此时应用您所做的更改。如果集群在维护期间重启，则无法应用更改。有关更多信息，请参阅 WLM 动态和静态属性。

## 默认参数值

> **Note**
>
> 自 2025 年 1 月 10 日起，`require_ssl` 参数的默认值为 true。如果您不想您的集群使用 SSL，则可以在创建集群时使用自定义参数组，或者在使用默认参数组创建集群后修改集群以使其与自定义参数组关联。

您可以通过下面的表一目了然地了解默认的参数值，并借助提供的相应链接详细了解每个参数。下面显示的是 redshift-2.0 参数组系列的默认值。

| 参数名称 | 值 | 更多信息 |
| --- | --- | --- |
| auto_analyze | true | 《Amazon Redshift 数据库开发人员指南》中的 auto_analyze |
| auto_mv | true | 《Amazon Redshift 数据库开发人员指南》中的自动实体化视图 |
| datestyle | ISO、MDY | 《Amazon Redshift 数据库开发人员指南》中的 datestyle |
| enable_case_sensitive_identifier | false | 《Amazon Redshift 数据库开发人员指南》中的 enable_case_sensitive_identifier |
| enable_user_activity_logging | false | 数据库审计日志记录本指南中的 |
| extra_float_digits | 0 | 《Amazon Redshift 数据库开发人员指南》中的 extra_float_digits |
| max_concurrency_scaling_clusters | 1 | 《Amazon Redshift 数据库开发人员指南》中的 max_concurrency_scaling_clusters |
| query_group | 默认 | 《Amazon Redshift 数据库开发人员指南》中的 query_group |
| require_ssl | true | 配置连接的安全选项本指南中的 |
| search_path | $user、public | 《Amazon Redshift 数据库开发人员指南》中的 search_path |
| statement_timeout | 0 | 《Amazon Redshift 数据库开发人员指南》中的 statement_timeout |
| wlm_json_configuration | [{"auto_wlm":true}] | 工作负载管理本指南中的 |

| 参数名称 | 值 | 更多信息 |
| --- | --- | --- |
| use_fips_ssl | false | 仅在您的系统需要与 FIPS 兼容时才启用与 FIPS 兼容的 SSL 模式。 |

> **Note**
>
> max_cursor_result_set_size 参数已弃用。有关游标结果集大小的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的游标约束。

您可以使用数据库中的 SET 命令临时覆盖参数。SET 命令仅覆盖当前会话持续时间内的参数。除了上表中列出的参数之外，您还可以通过在数据库中设置 wlm_query_slot_count 来临时调整槽位计数。wlm_query_slot_count 参数不适用于参数组中的配置。有关调整槽数的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 wlm_query_slot_count。有关临时覆盖其他参数的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的修改服务器配置。

# 工作负载管理

在 Amazon Redshift 中，您可以使用工作负载管理 (WLM) 来定义可用的查询队列的数量，并定义如何将查询路由至这些队列以进行处理。WLM 是参数组配置的一部分。集群使用其关联的参数组中指定的 WLM 配置。

当您创建参数组时，默认 WLM 配置中包含一个队列，该队列最多可并发运行五个查询。如果您想要更好地控制查询处理，则可以添加更多队列并在每个队列中配置 WLM 属性。除非对其属性进行配置，否则您添加的每个队列都具有相同的默认 WLM 配置。

当您添加更多队列时，配置中的最后一个队列是默认队列。除非根据 WLM 配置中的标准将查询路由至另一个队列，否则该查询将由默认队列进行处理。您可以为默认队列指定模式和并发级别（查询槽），但不能为默认队列指定用户组或查询组。

与其他参数一样，您无法修改默认参数组中的 WLM 配置。与默认参数组相关联的集群始终使用默认的 WLM 配置。要修改 WLM 配置，请创建一个新的参数组，然后将该参数组与需要自定义 WLM 配置的所有集群相关联。

## WLM 动态和静态属性

WLM 配置属性可以是动态的，也可以是静态的。您可以将动态属性应用于数据库而无需重新启动集群，但静态属性需要重新启动集群才能够使更改生效。有关静态和动态属性的更多信息，请参阅 WLM 动态和静态配置属性。

## WLM 配置参数的属性

您可以使用 Amazon Redshift 控制台、AWS CLI、Amazon Redshift API 或 AWS 开发工具包之一配置 WLM。WLM 配置使用多个用于定义队列行为的属性，例如队列之间的内存分配、队列中可以并发运行的查询数量等。

> **Note**
>
> 下列属性和其 Amazon Redshift 控制台名称一起显示，具体说明中提供了对应的 JSON 属性名称。

下表总结属性是适用于自动 WLM 还是手动 WLM。

| WLM 属性 | 自动 WLM | 手动 WLM |
| --- | --- | --- |
| 自动 WLM | 是 | 是 |
| 启用短查询加速 | 支持 | 是 |
| 短查询的最大运行时间 | 支持 | 是 |
| 优先级 | 是 | 否 |
| 队列类型 | 支持 | 是 |
| 队列名称 | 支持 | 是 |
| 并发扩展模式 | 支持 | 是 |
| 并发 | 否 | 是 |
| 用户组 | 支持 | 是 |

| WLM 属性 | 自动 WLM | 手动 WLM |
| --- | --- | --- |
| 用户组通配符 | 支持 | 是 |
| 查询组 | 支持 | 是 |
| 查询组通配符 | 支持 | 是 |
| 用户角色 | 支持 | 是 |
| 用户角色通配符 | 支持 | 是 |
| 超时 | 否 | 弃用 |
| 内存 | 否 | 是 |
| 查询监控规则 | 支持 | 是 |

下面的列表说明了可以配置的 WLM 属性。

自动 WLM

自动 WLM 设置为 `true` 以启动自动 WLM。自动 WLM 将主要并发和内存(%) 的值设置为 `Auto`。Amazon Redshift 管理查询并发性和内存分配。默认值为 `true`。

JSON 属性：`auto_wlm`

启用短查询加速

短查询加速 (SQA) 让选定的短时查询优先于长时查询。SQA 在专用空间中执行短时查询，因此 SQA 查询不会被迫排在队列中的长时查询后面等待。使用 SQA，短时查询会更快地开始执行，用户会更快地看到结果。当您启用 SQA 时，还可以为短时查询指定最大运行时间。要启用 SQA，请指定 `true`。默认为 `false`。此设置适用于每个参数组，而不是适用于队列。

JSON 属性：`short_query_queue`

短查询的最大运行时间

当您启用 SQA 时，可以指定 0 以允许 WLM 动态设置短查询的最大运行时间。或者，您也可以指定一个介于 1 到 20 秒之间的值（以毫秒为单位）。默认值为 `0`。

JSON 属性：`max_execution_time`

优先级：

优先级设置队列中运行的查询的优先级。要设置优先级，WLM 模式必须设置为自动 WLM；也即 auto_wlm 必须为 true。优先级值可以是 highest、high、normal、low 和 lowest。默认值为 normal。

JSON 属性：priority

队列类型

队列类型指定自动 WLM 或手动 WLM 使用的队列。将 queue_type 设置为 auto 或 manual。如果未指定，则默认值为 manual。

JSON 属性：queue_type

队列名称

队列的名称。可以根据业务需求设置队列的名称。队列名称在 WLM 配置中必须唯一，最多为 64 个字母数字字符、下划线或空格，并且不能包含引号。例如，如果您有一个 ETL 查询队列，则可将该队列命名为 ETL queue。此名称在指标、系统表值和 Amazon Redshift 控制台中用来标识队列。使用这些源中的名称的查询和报表需要能够处理名称的更改。以前，队列名称由 Amazon Redshift 生成。队列的默认名称为 Queue 1、Queue 2，依此类推，直至最后一个名为 Default queue 的队列。

> **Important**
>
> 如果您更改队列名称，则 WLM 队列指标（例如 WLMQueueLength、WLMQueueWaitTime、WLMQueriesCompletedPerSecond、WLMQueryDurat 等）的 QueueName 维度值也会发生更改。因此，如果您更改队列的名称，则可能需要更改已设置的 CloudWatch 警报。

JSON 属性：name

并发扩展模式

要在队列上启用并发扩展，请将并发扩展模式设置为 auto。当路由到队列的查询数超过队列的已配置并发数时，符合条件的查询将转到扩展集群。当有槽位可用时，将在主集群上运行查询。默认值为 off。

JSON 属性：concurrency_scaling

并发

手动 WLM 队列中可以并发运行的查询的数量。此属性仅适用于手动 WLM。如果启用了并发扩展，则当队列达到并发级别（查询槽）时，符合条件的查询将转到扩展集群。如果未启用并发扩展，则查询将在队列中等待，直到有槽位可用。范围介于 1-50 之间。

JSON 属性：query_concurrency

用户组

逗号分隔的用户组名称列表。当用户组成员在数据库中运行查询时，其查询将路由至与相应用户组相关联的队列。

JSON 属性：user_group

用户组通配符

用来指示是否对用户组启用通配符的布尔值。如果该值为 0，则禁用通配符；如果该值为 1，则启用通配符。启用通配符后，您可以在运行查询时使用“*”或“?”指定多个用户组。有关更多信息，请参阅通配符。

JSON 属性：user_group_wild_card

查询组

逗号分隔的查询组列表。当查询组成员在数据库中运行查询时，其查询将路由至与相应查询组相关联的队列。

JSON 属性：query_group

Query Group Wildcard

用来指示是否对查询组启用通配符的布尔值。如果该值为 0，则禁用通配符；如果该值为 1，则启用通配符。启用通配符后，您可以在运行查询时使用“*”或“?”指定多个查询组。有关更多信息，请参阅通配符。

JSON 属性：query_group_wild_card

用户角色

以逗号分隔的用户角色列表。当具有该用户角色的成员在数据库中运行查询时，其查询将路由至与其用户角色相关联的队列。有关用户角色的更多信息，请参阅基于角色的访问控制 (RBAC)。

JSON 属性：user_role

用户角色通配符

用来指示是否对查询组启用通配符的布尔值。如果该值为 0，则禁用通配符；如果该值为 1，则启用通配符。启用通配符后，您可以在运行查询时使用“*”或“?”指定多个查询组。有关更多信息，请参阅通配符。

JSON 属性：`user_role_wild_card`

超时 (ms)

WLM 超时 (`max_execution_time`) 已弃用。使用自动 WLM 时不可用。相反，使用 `query_execution_time` 创建查询监控规则 (QMR) 来限制经过的查询执行时间。有关更多信息，请参阅 WLM 查询监控规则。

查询在取消之前可以运行的最长时间，以毫秒为单位。在一些情况下，只读查询（例如 SELECT 语句）可能会由于 WLM 超时而取消。在这些情况下，WLM 会尝试根据 WLM 队列分配规则将该查询路由到下一个匹配的队列。如果查询不匹配任何其他队列定义，则取消查询；系统不会将其分配给默认队列。有关更多信息，请参阅 WLM 查询队列跳过。WLM 超时不适用于已进入 `returning` 状态的查询。要查看查询的状态，请参阅 STV_WLM_QUERY_STATE 系统表。

JSON 属性：`max_execution_time`

“Memory”

分配给队列的内存百分比。如果您要为至少一个队列指定内存百分比，则必须为所有其他队列指定内存百分比，且所有队列的百分比合计不超过 100%。如果所有队列上的内存分配低于 100%，则未分配的内存将由服务管理。服务可以临时将此未分配内存提供给请求额外内存进行处理的队列。

JSON 属性：`memory_percent_to_use`

查询监控规则

您可以使用 WLM 查询监控规则，根据标准或谓词持续监控您的 WLM 队列中有无您指定的查询。例如，您可监控倾向于使用额外系统资源的查询，然后在查询超过您指定的性能界限时启动指定的操作。

> **Note**
>
> 如果您选择以编程方式创建规则，强烈建议您使用控制台生成包含在参数组定义中的 JSON。

您将一个查询监控规则与特定的查询队列相关联。您可以让每个队列有最多 25 个规则，并且所有队列的总限制为 25 个规则。

JSON 属性：rules

JSON 属性层次结构：

```
rules
    rule_name
    predicate
        metric_name
        operator
        value
    action
        value
```

对于每个规则，您可以指定以下属性：

- rule_name – 规则名称必须在 WLM 配置中是唯一的。规则名称最多可包含 32 个字母数字字符或下划线，且不能包含空格或引号。
  - predicate – 您最多可以为每个规则设置 3 个谓词。对于每个谓词，指定以下属性。
    - metric_name – 有关指标的列表，请参阅《Amazon Redshift 数据库开发人员指南》中的查询监控指标。
    - operator – 操作包括 =、< 和 >。
    - value – 指定指标的可触发操作的阈值。
- action – 每个规则都与一个操作相关联。有效的操作是：
  - log
  - hop（仅适用于手动 WLM）
  - abort
  - change_query_priority（仅适用于自动 WLM）

以下示例显示名为 rule_1 的 WLM 查询监控规则的 JSON，带有两个谓词以及操作 hop。

```
"rules": [
          {
            "rule_name": "rule_1",
            "predicate": [
              {
                "metric_name": "query_execution_time",
```

```
            "operator": ">",
            "value": 100000
          },
          {
            "metric_name": "query_blocks_read",
            "operator": ">",
            "value": 1000
          }
        ],
        "action": "hop"
      }
    ]
```

有关这些属性以及用于配置查询队列的策略的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的实施工作负载管理。

## 使用 AWS CLI 配置 WLM 参数

要配置 WLM，您需要修改 `wlm_json_configuration` 参数。`wlm_json_configuration` 属性值的最大大小为 8000 个字符。相应值采用 JavaScript 对象表示法 (JSON) 格式。如果您想要使用 AWS CLI、Amazon Redshift API 或其中一个 AWS 开发工具包来配置 WLM，则可以使用此部分的其余内容来了解如何为 `wlm_json_configuration` 参数构建 JSON 结构。

> **Note**
>
> 如果您使用 Amazon Redshift 控制台来配置 WLM，则无需了解 JSON 格式，因为您可以通过控制台轻松添加队列并配置其属性。有关使用控制台配置 WLM 的更多信息，请参阅修改参数组。

示例

下面的示例是默认的 WLM 配置，定义了使用自动 WLM 的队列。

```
{
   "auto_wlm": true
}
```

示例

下面的示例是一个自定义 WLM 配置，它定义了一个并发级别（查询槽）为五的手动 WLM 队列。

```
{
   "query_concurrency":5
}
```

语法

默认的 WLM 配置非常简单，只有一个队列和一个属性。您可以添加更多队列并为 JSON 结构中的每个队列配置多种属性。以下语法表示您用于配置具有多种属性的多个队列的 JSON 结构：

```
[
   {
      "ParameterName":"wlm_json_configuration", "ParameterValue":
         "[
             {
                "q1_first_property_name":"q1_first_property_value",
                "q1_second_property_name":"q1_second_property_value",
                ...
             },
             {
                "q2_first_property_name":"q2_first_property_value",
                "q2_second_property_name":"q2_second_property_value",
                ...
             }
             ...

         ]"
   }
]
```

在上述示例中，以 q1 开头的代表性属性是第一个队列的数组中的对象。其中每个对象都是名称/值对；name 和 value 共同设置第一个队列的 WLM 属性。以 q2 开头的代表性属性是第二个队列的数组中的对象。如果您需要更多队列，则可以为每个增加的队列添加另一个数组并为每个对象设置属性。

修改 WLM 配置时，您必须涵盖队列的整个结构，即使您只是想更改队列中的一个属性也是如此。这是因为整个 JSON 结构以字符串的形式传递为 wlm_json_configuration 参数的值。

## 设置 AWS CLI 命令的格式

使用 wlm_json_configuration 时，您需要为 AWS CLI 参数指定具体格式。您使用的格式取决于您的客户端操作系统。操作系统可以采用不同的方式将 JSON 结构括起来，因此它可以根据命令行正

确地进行传输。要详细了解如何在 Linux、macOS 和 Windows 操作系统中构造相应的命令，请参阅以下部分。有关封闭 AWS CLI 中的 JSON 数据结构的差异的更多信息，一般情况下，请参阅《AWS Command Line Interface 用户指南》中的引用字符串。

示例

以下示例命令为名为 example-parameter-group 的参数组配置手动 WLM。该配置可启用短查询加速，其中短查询的最大运行时间设置为 0，这指示 WLM 可动态设置该值。ApplyType 设置为 dynamic。此设置表示对参数中的动态属性做出的所有更改都将立即应用，除非同时对配置做出了其他静态更改。该配置定义下列三个队列：

- 通过第一个队列，用户可以将 report 指定为其查询中的标签（在 query_group 属性中所指定的），以帮助他们将查询路由至该队列。对于 report* 标签，通配符搜索处于启用状态，因此标签无需完全相同，查询也可路由至该队列。例如，reports 和 reporting 都与此查询组匹配。此队列将分配所有队列总内存的 25%，并且最多可同时运行四个查询。查询限制为最大时间 20000 毫秒 (ms)。模式设置为自动，因此当队列的查询槽位已满时，符合条件的查询将发送到扩展集群。
- 通过第二个队列，作为数据库中 admin 或 dba 组成员的用户可以将其查询路由至队列以进行处理。对于用户组，通配符搜索处于禁用状态，因此用户必须与数据库中的组完全匹配，其查询才能路由至该队列。此队列将跨所有队列分配总内存的 40%，并且最多可同时运行五个查询。模式设置为关闭，因此管理员或 dba 组的成员发送的所有查询都在主集群上运行。
- 配置中的最后一个队列是默认队列。此队列将分配所有队列总内存的 35%，并且最多可同时处理五个查询。模式设置为自动。

> **Note**
>
> 为了便于演示，该示例分为多行显示出来。实际命令中不得使用换行符。

```
aws redshift modify-cluster-parameter-group
--parameter-group-name example-parameter-group
--parameters
'[
  {
    "query_concurrency": 4,
    "max_execution_time": 20000,
    "memory_percent_to_use": 25,
    "query_group": ["report"],
    "query_group_wild_card": 1,
```

```
    "user_group": [],
    "user_group_wild_card": 0,
    "user_role": [],
    "user_role_wild_card": 0,
    "concurrency_scaling": "auto",
    "queue_type": "manual"
  },
  {
    "query_concurrency": 5,
    "memory_percent_to_use": 40,
    "query_group": [],
    "query_group_wild_card": 0,
    "user_group": [
      "admin",
      "dba"
    ],
    "user_group_wild_card": 0,
    "user_role": [],
    "user_role_wild_card": 0,
    "concurrency_scaling": "off",
    "queue_type": "manual"
  },
  {
    "query_concurrency": 5,
    "query_group": [],
    "query_group_wild_card": 0,
    "user_group": [],
    "user_group_wild_card": 0,
    "user_role": [],
    "user_role_wild_card": 0,
    "concurrency_scaling": "auto",
    "queue_type": "manual"
  },
  {"short_query_queue": true}
]'
```

以下是为自动 WLM 配置配置 WLM 查询监控规则的示例。该示例创建一个名为 example-monitoring-rules 的参数组。该配置定义了和上一示例相同的三个队列，但不再指定 query_concurrency 和 memory_percent_to_use。该配置还添加了以下规则和查询优先级：

- 第一个队列定义名为 rule_1 的规则。该规则有两个谓词：query_cpu_time > 10000000 和 query_blocks_read > 1000。规则操作是 log。此队列的优先级为 Normal。

- 第二个队列定义名为 rule_2 的规则。该规则有两个谓词：query_execution_time > 600000000 和 scan_row_count > 1000000000。规则操作是 abort。此队列的优先级为 Highest。
- 配置中的最后一个队列是默认队列。此队列的优先级为 Low。

> **Note**
>
> 为了便于演示，该示例分为多行显示出来。实际命令中不得使用换行符。

```
aws redshift modify-cluster-parameter-group
--parameter-group-name example-monitoring-rules
--parameters
'[ {
  "query_group" : [ "report" ],
  "query_group_wild_card" : 1,
  "user_group" : [ ],
  "user_group_wild_card" : 0,
  "user_role": [ ],
  "user_role_wild_card": 0,
  "concurrency_scaling" : "auto",
  "rules" : [{
    "rule_name": "rule_1",
    "predicate": [{
      "metric_name": "query_cpu_time",
      "operator": ">",
      "value": 1000000 },
      { "metric_name": "query_blocks_read",
      "operator": ">",
      "value": 1000
    } ],
    "action" : "log"
  } ],
   "priority": "normal",
   "queue_type": "auto"
}, {
  "query_group" : [ ],
  "query_group_wild_card" : 0,
  "user_group" : [ "admin", "dba" ],
  "user_group_wild_card" : 0,
  "user_role": [ ],
```

```
  "user_role_wild_card": 0,
  "concurrency_scaling" : "off",
  "rules" : [ {
    "rule_name": "rule_2",
    "predicate": [
      {"metric_name": "query_execution_time",
      "operator": ">",
      "value": 600000000},
      {"metric_name": "scan_row_count",
      "operator": ">",
      "value": 1000000000}],
      "action": "abort"}],
   "priority": "high",
   "queue_type": "auto"

}, {
  "query_group" : [ ],
  "query_group_wild_card" : 0,
  "user_group" : [ ],
  "user_group_wild_card" : 0,
  "user_role": [ ],
  "user_role_wild_card": 0,
  "concurrency_scaling" : "auto",
  "priority": "low",
  "queue_type": "auto",
  "auto_wlm": true
}, {
  "short_query_queue" : true
} ]'
```

在命令行中使用 AWS CLI 和 JSON 文件以配置 WLM

您可以使用 `wlm_json_configuration` 修改 AWS CLI 参数，并将 `parameters` 参数值作为 JSON 文件传递。

```
aws redshift modify-cluster-parameter-group --parameter-group-name
 myclusterparaametergroup --parameters file://modify_pg.json
```

`--parameters` 的参数存储在 `modify_pg.json` 文件中。文件位置是使用操作系统格式指定的。有关更多信息，请参阅从文件中加载参数。下面显示了 `modify_pg.json` JSON 文件的内容示例。

```
[
    {
        "ParameterName": "wlm_json_configuration",
        "ParameterValue": "[{\"user_group\":\"example_user_group1\",\"query_group\":
 \"example_query_group1\", \"query_concurrency\":7},{\"query_concurrency\":5}]"
    }
]
```

```
[
    {
        "ParameterName": "wlm_json_configuration",
        "ParameterValue": "[{\"query_group\":[\"reports\"],\"query_group_wild_card\":0,
\"query_concurrency\":4,\"max_execution_time\":20000,\"memory_percent_to_use\":25},
{\"user_group\":[\"admin\",\"dba\"],\"user_group_wild_card\":1,\"query_concurrency\":5,
\"memory_percent_to_use\":40},{\"query_concurrency\":5,\"memory_percent_to_use\":35},
{\"short_query_queue\": true, \"max_execution_time\": 5000 }]",
        "ApplyType": "dynamic"
    }
]
```

在 Linux 和 macOS 操作系统上的命令行中使用 AWS CLI 配置 WLM 的规则

请按照以下规则在一行中使用参数运行 AWS CLI 命令：

- 整个 JSON 结构必须使用单引号 (') 和一组方括号 ([ ]) 括起来。
- 所有参数名称和参数值必须使用双引号 (") 括起来。
- 在 ParameterValue 值中，必须使用双引号 (") 和方括号 ([ ]) 将整个嵌套结构括起来。
- 在嵌套结构中，必须使用大括号 ({ }) 将每个队列的每个属性和值括起来。
- 在嵌套结构中，必须在每个双引号 (") 前面使用反斜杠 (\) 转义符。
- 对于名称/值对，使用冒号 (:) 将每个属性与其值分隔开来。
- 可使用逗号 (,) 将各个名称/值对彼此分隔开来。
- 多个队列可通过在一个队列结束的右大括号 (}) 和下一个队列开始的左大括号 ({) 之间使用逗号 (,) 分隔开来。

在 Microsoft Windows 操作系统上的 Windows PowerShell 中使用 AWS CLI 配置 WLM 的规则

请按照以下规则在一行中使用参数运行 AWS CLI 命令：

- 整个 JSON 结构必须使用单引号 (') 和一组方括号 ([ ]) 括起来。
- 所有参数名称和参数值必须使用双引号 (") 括起来。
- 在 ParameterValue 值中，必须使用双引号 (") 和方括号 ([ ]) 将整个嵌套结构括起来。
- 在嵌套结构中，必须使用大括号 ({ }) 将每个队列的每个属性和值括起来。
- 在嵌套结构中，必须在每个双引号 (") 及其反斜杠 (\) 转义符前面使用反斜杠 (\) 转义符。该要求意味着，您将使用三个反斜杠和一个双引号以确保正确传递属性 (\\\")。
- 对于名称/值对，使用冒号 (:) 将每个属性与其值分隔开来。
- 可使用逗号 (,) 将各个名称/值对彼此分隔开来。
- 多个队列可通过在一个队列结束的右大括号 (}) 和下一个队列开始的左大括号 ({) 之间使用逗号 (,) 分隔开来。

在 Windows 操作系统上使用命令提示符配置 WLM 的规则

请按照以下规则在一行中使用参数运行 AWS CLI 命令：

- 整个 JSON 结构必须使用双引号 (") 和一组方括号 ([ ]) 括起来。
- 所有参数名称和参数值必须使用双引号 (") 括起来。
- 在 ParameterValue 值中，必须使用双引号 (") 和方括号 ([ ]) 将整个嵌套结构括起来。
- 在嵌套结构中，必须使用大括号 ({ }) 将每个队列的每个属性和值括起来。
- 在嵌套结构中，必须在每个双引号 (") 及其反斜杠 (\) 转义符前面使用反斜杠 (\) 转义符。该要求意味着，您将使用三个反斜杠和一个双引号以确保正确传递属性 (\\\")。
- 对于名称/值对，使用冒号 (:) 将每个属性与其值分隔开来。
- 可使用逗号 (,) 将各个名称/值对彼此分隔开来。
- 多个队列可通过在一个队列结束的右大括号 (}) 和下一个队列开始的左大括号 ({) 之间使用逗号 (,) 分隔开来。

# 创建参数组

如果您要设置与默认参数组不同的参数值，您可以创建您自己的参数组。

创建参数组

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。

2. 在导航菜单上，选择 Configurations（配置），然后选择 Workload management（工作负载管理），以显示 Workload management（工作负载管理）页面。
3. 选择 Create (创建) 显示 Create parameter group (创建参数组) 窗口。
4. 为 Parameter group name (参数组名称) 和 Description (说明) 输入一个值。
5. 选择 Create (创建) 以创建参数组。

# 修改参数组

您可以查看任意参数组，查看参数值的摘要以及工作负载管理 (WLM) 配置。您可以修改参数来更改参数设置和 WLM 配置属性。

> **Note**
>
> 您无法修改默认参数组。

AWS 管理控制台

在控制台中，组参数显示在参数选项卡中，工作负载队列显示在工作负载管理选项卡中。

修改参数组

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Configurations（配置），然后选择 Workload management（工作负载管理），以显示 Workload management（工作负载管理）页面。
3. 选择要修改的参数组以便显示详细信息页面，该页面带有 Parameters (参数) 和 Workload management (工作负载管理) 选项卡。
4. 选择 Parameters (参数) 选项卡查看当前参数设置。
5. 选择 Edit parameters (编辑参数) 以便允许更改以下参数的设置：

   - `auto_analyze`
   - `auto_mv`
   - `datestyle`
   - `enable_case_sensitive_identifier`

- enable_user_activity_logging
- extra_float_digits
- max_concurrency_scaling_clusters
- max_cursor_result_set_size
- query_group
- require_ssl
- search_path
- statement_timeout
- use_fips_ssl

有关这些参数的更多信息，请参阅 Amazon Redshift 参数组。

6. 输入所做的更改，然后选择 Save (保存) 更新参数组。

修改参数组的 WLM 配置

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Configurations（配置），然后选择 Workload management（工作负载管理），以显示 Workload management（工作负载管理）页面。
3. 选择要修改的参数组以便显示详细信息页面，该页面带有 Parameters (参数) 和 Workload management (工作负载管理) 选项卡。
4. 选择 Workload management (工作负载管理) 选项卡查看当前 WLM 配置。
5. 选择编辑工作负载队列以编辑 WLM 配置。
6. （可选）选择 Enable short query acceleration (启用短查询加速) 以便启用短查询加速 (SQA)。

   当您启用 SQA 时，原定设置情况下 Maximum run time for short queries (1 to 20 seconds) (短查询的最大运行时间 (1 到 20 秒)) 将设置为 Dynamic (动态)。要将最大运行时设置为固定值，请选择一个介于 1–20 之间的值。
7. 执行以下一项或多项操作来修改队列配置：

   - 选择 Switch WLM mode (切换 WLM 模式) 可在 Automatic WLM (自动 WLM) 和 Manual WLM (手动 WLM) 之间选择。

使用 Automatic WLM (自动 WLM) 时，Memory (内存) 和 Concurrency on main (主集群上的并发) 值设置为 auto (自动)。

- 要创建队列，请选择 Edit workload queues (编辑工作负载队列)，然后选择 Add Queue (添加队列)。
- 要修改队列，更改表中的属性值。根据队列类型，属性可能包括：
  - 可以更改 Queue name (队列名称)。
  - “Memory”
  - 主集群上的并发
  - 并发扩展模式可以为关闭或自动
  - 超时 (ms)
  - 用户组
  - 查询组
  - 用户角色

  有关这些属性的更多信息，请参阅WLM 配置参数的属性。

  > **Important**
  >
  > 如果您更改队列名称，则 WLM 队列指标（例如 WLMQueueLength、WLMQueueWaitTime、WLMQueriesCompletedPerSecond、WLMQuer 等）的 QueueName 维度值也会发生更改。因此，如果您更改队列的名称，则可能需要更改已设置的 CloudWatch 警报。

- 要更改队列顺序，请选择 Up (向上) 和 Down (向下) 箭头按钮。
- 要删除队列，在表中选择改队列所在行中的 Delete (删除)。

8. （可选）选择 Defer dynamic changes until reboot (推迟动态更改，直到重新启动) 以便在下次重启集群后对其应用更改。

   > **Note**
   >
   > 对于某些设置，无论此项设置为何，都要求在集群重启之后才生效。有关更多信息，请参阅 WLM 动态和静态属性。

9. 选择 Save。

### AWS CLI

要使用 AWS CLI 配置 Amazon Redshift 参数，可以对特定的参数组使用 modify-cluster-parameter-group 命令。您可以在 `parameter-group-name` 中指定要修改的参数组。您可以使用 `parameters` 参数（针对 `modify-cluster-parameter-group` 命令）指定希望在参数组中修改的每个参数的名称/值对。

> **Note**
>
> 这些是使用 `wlm_json_configuration` 配置 AWS CLI 参数时需要考虑的一些特殊注意事项。此部分中的示例适用于除 `wlm_json_configuration` 之外的所有参数。有关使用 `wlm_json_configuration` 配置 AWS CLI 的更多信息，请参阅工作负载管理。

修改参数值之后，您必须重新启动与修改后的参数组相关联的所有集群。当系统正在应用值时，集群状态将针对 `ParameterApplyStatus` 显示 `applying`；当应用了这些值之后，该状态则变为 `pending-reboot`。重新启动之后，您集群中的数据库便开始使用新的参数值。有关重新启动集群的更多信息，请参阅重新引导集群。

> **Note**
>
> `wlm_json_configuration` 参数中包含一些动态属性，您无需重新启动相关联的集群即可应用更改。有关动态属性和静态属性的更多信息，请参阅 WLM 动态和静态属性。

下面的语法显示了如何使用 `modify-cluster-parameter-group` 命令配置参数。您可以指定 *`parameter_group_name`*，并使用实际参数来替换 *`parameter_name`* 和 *`parameter_value`*，以修改相应参数的值。如果您想同时修改多个参数，请使用空格将每个参数和值集合彼此分隔开来。

```
aws redshift modify-cluster-parameter-group --parameter-group-
name parameter_group_name --parameters
 ParameterName=parameter_name,ParameterValue=parameter_value
```

下面的示例显示了如何配置 `myclusterparametergroup` 参数组的 `statement_timeout` 和 `enable_user_activity_logging` 参数。

> **Note**
>
> 为了便于阅读，该示例分为多行显示出来；但在实际的 AWS CLI 中，该示例显示为一行。

```
aws redshift modify-cluster-parameter-group
--parameter-group-name myclusterparametergroup
--parameters ParameterName=statement_timeout,ParameterValue=20000
 ParameterName=enable_user_activity_logging,ParameterValue=true
```

# 创建查询监控规则

您可以使用 Amazon Redshift 控制台创建和修改 WLM 查询监控规则。查询监控规则是一个参数组的 WLM 配置参数的一部分。如果修改查询监控规则 (QMR)，无需修改集群即可自动进行更改。有关更多信息，请参阅 WLM 查询监控规则。

创建规则时，您要定义规则名称、一个或多个谓词以及一个操作。

保存包含规则的 WLM 配置时，您可以将规则定义的 JSON 代码视为 WLM 配置参数的 JSON 的一部分。

创建查询监控规则

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Configurations（配置），然后选择 Workload management（工作负载管理），以显示 Workload management（工作负载管理）页面。
3. 选择要修改的参数组以便显示详细信息页面，该页面带有 Parameters (参数) 和 Workload management (工作负载管理) 选项卡。
4. 选择工作负载管理选项卡，然后选择编辑工作负载队列以编辑 WLM 配置。
5. 使用预定义模板或从头开始添加新规则。

   要使用预定义模板，请执行以下操作：

   1. 选择 Query monitoring rules（查询监控规则）组中的 Add rule from template（从模板中添加规则）。此时将显示规则模板的列表

2. 选择一个或多个规则模板。选择 Save (保存) 后，WLM 将为您选择的每个模板创建一个规则。
3. 输入或确认规则的值，其中包括 Rule names (规则名称)、Predicates (谓词) 和 Actions (操作)。
4. 选择保存。

要从头开始添加新规则，请执行以下操作：

1. 要添加其他谓词，请选择 Add predicate (添加谓词)。您最多可以为每个规则设置 3 个谓词。如果满足所有谓词，WLM 会触发关联操作。
2. 选择 Action。每个规则具有一个操作。
3. 选择保存。

Amazon Redshift 将生成 JSON 格式的 WLM 配置参数，并在 JSON 部分中显示它。

# 删除参数组

如果您不再需要某个参数组而且它并没有与任何集群相关联，您可以删除该参数组。您只能删除自定义参数组。

删除参数组

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Configurations（配置），然后选择 Workload management（工作负载管理），以显示 Workload management（工作负载管理）页面。
3. 对于 Parameter groups,，选择您要修改的参数组。

   > **Note**
   >
   > 您无法删除默认参数组。

4. 选择 Delete (删除) 并确认要删除参数组。

# 将 Amazon Redshift 与 AWS 合作伙伴集成

通过使用 Amazon Redshift，可以在 Amazon Redshift 控制台上与 AWS 合作伙伴集成。从 Cluster details（集群详细信息）页面上，您可以使用 AWS 合作伙伴应用程序加速数据到 Amazon Redshift 数据仓库的载入。您还可以联接和分析来自不同来源的数据以及集群中的现有数据。在完成与 Informatica 的集成之前，必须先将合作伙伴的 IP 地址添加到入站流量的允许列表中。以下 AWS 合作伙伴可以与 Amazon Redshift 集成：

- Datacoral
- Etleap
- Fivetran
- SnapLogic
- Stitch
- Upsolver
- Matillion（预览版）
- Sisense（预览版）
- Thoughtspot

AWS 合作伙伴可以使用 AWS CLI 或 Amazon Redshift API 操作与 Amazon Redshift 集成。有关更多信息，请参阅 Amazon Redshift API 参考上的 AWS CLI 命令参考。

使用以下过程将集群与 AWS 合作伙伴集成。

要将 Amazon Redshift 集群与 AWS 合作伙伴集成

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。
3. 选择要集成的集群。
4. 选择 Add partner integration（添加合作伙伴集成）。Choose partner（选择合作伙伴）页面将打开，其中包含有关可用 AWS 合作伙伴的详细信息。
5. 选择一位 AWS 合作伙伴，然后选择 Next（下一步）。

有关选择的 AWS 合作伙伴的更多详细信息出现，还包括您正在集成的集群的详细信息。集群详细信息部分包含您在 AWS 合作伙伴网站上提供的信息，例如集群标识符、端点、数据库名称和用户名（这是一个数据库用户名）。此信息将发送到您选择的合作伙伴。

6. 选择 Add partner（添加合作伙伴）以打开 AWS 合作伙伴的网站。
7. 在合作伙伴的网站上配置与您的 Amazon Redshift 集群的集成。在合作伙伴的网站上，您可以选择并配置加载到 Amazon Redshift 集群的数据源。您还可以定义其他数据提取、加载和转换 (ELT) 转换，以处理业务数据、将其与其他数据集联接并构建用于分析和报告的合并视图。

您可以查看和管理集群详细信息 Properties（属性）选项卡中的 AWS 合作伙伴集成。集成部分列出合作伙伴名称（您可以使用该名称链接到 AWS 合作伙伴网站）、集成的状态、接收数据的数据库，以及可能已更新集群的上一次成功连接。

可能的状态值如下所示：

- Active（活动）– AWS 合作伙伴可以连接到集群并完成配置的任务。
- Inactive（非活动）– AWS 合作伙伴集成不存在。
- Runtime failure（运行时失败）– AWS 合作伙伴可以连接到集群并无法完成配置的任务。
- Connection failure（连接失败）– AWS 合作伙伴无法连接到集群。

删除来自 Amazon Redshift 的 AWS 合作伙伴集成，数据会继续流入您的集群。在合作伙伴网站上完成删除操作。

# 加载 AWS 合作伙伴的数据

除了将合作伙伴与 Amazon Redshift 集群集成外，您还可以使用我们合作伙伴的数据加载工具，将来自 30 多个源的数据移动到您的 Amazon Redshift 集群中。在此之前，必须将合作伙伴的 IP 地址（见下文）添加到入站规则的允许列表中。有关向 Amazon EC2 安全组添加规则的更多信息，请参阅《Amazon EC2 用户指南》中的为您的实例授权入站流量。请注意，虽然 Informatica 数据加载器工具是免费的，但可能会收取数据传入费用，具体取决于您选择的数据来源和目标。

您可以加载来自以下合作伙伴的数据。

- Informatica

通过合作伙伴将数据加载到 Amazon Redshift 集群中

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 AWS 合作伙伴集成，然后选择要与您的集群集成的合作伙伴。
3. 选择 Complete <partner-name> integration（完成 <partner-name> 集成）。您将重定向至合作伙伴的集成站点。
4. 在合作伙伴的网站上输入必要的详细信息并完成集成。

# 预留节点

在 AWS 中，您使用 Amazon Redshift 产生的费用取决于计算节点。每个计算节点均按每小时费率计费。每小时费率因诸多因素而言，如区域、节点类型以及相应节点使用的是按需节点定价还是预留节点定价。

按需节点定价费用最高，但在 Amazon Redshift 中灵活性最高。使用按需费率，您只需为正在运行的集群中的计算节点付费。如果您关闭或删除某个集群，则无需再为该集群中的计算节点付费。您只需为所使用的计算节点付费，无需支付任何其他费用。您为每个计算节点支付的每小时费率因区域和节点类型等因素而异。

预留节点定价比按需定价的费用低一些，因为计算节点以打折后的每小时费率计费。不过，要享受此类打折费率，您必须购买预留节点产品。购买产品即进行预留。这种预留为您针对预留持续时间预留的每个节点设置一个打折费率。产品的打折费率因区域、节点类型、持续时间以及付款选项等因素而异。

您可以通过调用 `PurchaseReservedNodeOffering` API 操作或在 Amazon Redshift 控制台上选择 Purchase reserved nodes (购买预留节点)，将节点指定为预留节点。购买预留节点时，您必须为适用的预留节点类型指定 AWS 区域、节点类型、期限、节点数量和产品类型。预留节点只能用于指定的 AWS 区域。

本主题讨论了有哪些预留节点产品，以及如何购买此类产品才能减少运行 Amazon Redshift 集群的费用。本主题粗略介绍了按需费率和打折费率，以便您可以了解一些定价概念以及定价如何影响计费。有关具体费率的更多信息，请转到 Amazon Redshift 定价。

## 预留节点产品

如果您打算让 Amazon Redshift 集群持续运行很长一段时间，则不妨考虑购买预留节点产品。与按需定价相比，购买这些产品可以节省大量费用，但您必须预留计算节点并承诺在一年或三年的期限里为所使用的节点支付相应费用。

预留节点是一个计费概念，完全用于确定您为节点付费所使用的费率。实际上，预留节点并不会为您创建任何节点。无论如何使用，您都需要为预留节点付费，也就是说，您必须为预留持续时间预留的每个节点付费，无论正在运行的集群中是否有任何节点可以使用打折费率。

在项目评估阶段或运行概念验证时，您可以通过按需定价灵活地按需付费，仅为使用的服务付费，并随时关闭或删除集群以停止付费。满足了生产环境的需求并进入实施阶段后，不妨考虑购买一个或多个产品来预留计算节点。

一个产品适用于一个或多个计算节点。当您购买产品时，需指定要预留的计算节点的数量。您可以选择为多个计算节点购买一个产品，也可以选择购买多个产品并在每个产品中指定一定数量的计算节点。

例如，您可以采用以下有效方法为三个计算节点购买产品：

- 购买一个产品并指定三个计算节点。
- 购买两个产品，为第一个产品指定一个计算节点，为第二个产品指定两个计算节点。
- 购买三个产品并为这三个产品各指定一个计算节点。

## 比较不同预留节点产品的定价

Amazon Redshift 提供多种产品付款选项。您选择的付款选项会影响您针对相应预留付费的付款计划和打折费率。您为预留预先支付的费用越多，总体节省的费用就越多。

针对产品付费可使用下列付款选项。这些产品以与按需费率相比，节省的费用从低到高列出。

> **Note**
>
> 无论您是否使用预留节点，都需要为指定预留持续时间里的每个小时支付相应的每小时费率。付款选项仅确定付款频率以及应用的折扣。有关更多信息，请参阅 预留节点产品。

| 付款选项 | 付款计划 | 相对节省的费用 | Duration | 预付费用 | 定期支付的月度费用 |
|---|---|---|---|---|---|
| 无费用预付 | 在预留持续时间里按月分期付款。不预付任何费用。 | 与按需费率相比，节省 20%。 | 一年期或三年期 | 无 | 是 |
| 预付部分费用 | 预付部分费用，然后在预留持续时间里按月分期付款。 | 根据持续时间，节省 41% 到 73%。 | 一年期或三年期 | 支持 | 是 |
| 预付全部费用 | 针对预留预付全费。无月度费用。 | 根据持续时间，节省 42% 到 76%。 | 一年期或三年期 | 是 | 无 |

具体选项和持续时间视可用情况而定。

> **Note**
>
> 如果您之前购买的是 Amazon Redshift 的高利用率产品，则可比较的产品是预付部分费用产品。

## 预留节点的工作方式

使用预留节点产品，您需要根据上一部分所述的付款方式付费。无论您已有正在运行的集群，还是在进行预留后启动集群，都需要按这种方式付费。

当您购买产品时，您的预留将处于 payment-pending 状态，直到预留得到处理。如果预留未得到处理，则该状态将显示为 payment-failed，直到您再次尝试进行此流程。预留成功得到处理后，该状态将变为 active。在该状态变为 active 之前，您的预留适用的打折费率将不会应用于您的账单。预留持续时间过后，该状态将变为 retired，但您可以继续出于历史目的访问预留的相关信息。当预留处于 retired 状态时，您的集群将继续运行，但您可能要以按需费率付费，除非您进行另一次预留，针对节点应用打折定价。

预留节点专用于您在其中购买产品的区域。如果您使用 Amazon Redshift 控制台来购买产品，请选择您要在其中购买产品的 AWS 区域，然后完成预留流程。如果您以编程方式购买产品，则区域由您连接到的 Amazon Redshift 端点确定。有关 Amazon Redshift 区域的更多信息，请转到《Amazon Web Services 一般参考》中的区域和端点。

为确保相应的打折费率在您启动集群时应用于所有节点，请确保区域、节点类型以及您选择的节点数量与一个或多个有效预留匹配。否则，对于与有效预留不匹配的节点，将以按需费率对您进行收费。

在正在运行的集群中，如果您超出预留的节点数量，则您将以按需费率为超出预留数量的节点付费。这种费用意味着，您可能以不同的费率为同一个集群中的节点付费，具体取决于您预留的节点数量。您可以再购买一个产品来涵盖超出预留数量的这部分节点，然后在预留状态变为 active 之后，相应的打折费率将在剩余持续时间里应用于这部分节点。

如果您调整集群的大小以采用其他节点类型，并且您没有预留此类型的节点，则您将以按需费率付费。如果您希望大小经过调整后的集群享受到打折费率，则可以采用新的节点类型再购买一个产品。不过，您还将继续为原始预留付费，直到其持续时间过去。如果您需要在期限到期前更改预留，请使用 AWS 控制台创建支持案例。

> **Note**
>
> 控制台显示已使用和未使用的预留节点的数量。但是，控制台仅将当前用户账户使用的节点数量显示为已使用。如果同一付款人账户下的另一个用户账户使用节点，则控制台会将这些节点显示为未使用。
> 示例
>
> - 付款人账户预留 20 个节点
> - 当前用户账户使用六个节点
> - 同一付款人账户下的另一个用户账户也使用六个节点
>
> 在此示例中，控制台仅显示六个已使用的节点和十四个未使用的节点。

## 预留节点和整合账单

当购买账户属于在一个整合账单付款人账户之下开具账单的一套账户的一部分时，预留节点的定价优惠可以共享。将所有子账户的小时使用量每月聚合到付款人账户。这通常对具有不同职能团队或团体的公司很有用；然后，将应用正常的预留节点逻辑来计算账单。有关更多信息，请参阅 AWS Billing 用户指南中的整合账单。

## 在使用预留实例的集群上更改节点类型之前需要考虑的事项

您可以使用调整大小功能（弹性或经典），通过更改节点系列或节点大小来更改集群的节点类型。更改节点类型时，与集群关联的预留节点购买不会自动更改。现有预留保持与原始节点类型绑定。在预留期限结束之前，您仍需为这些预留继续付费。与此同时，新类型的节点可能会采用按需费率计费。

因此，对于对预留实例进行的更改，我们建议您在更改使用预留节点的集群的节点类型之前，先联系您的 AWS 账户团队或 AWS 账单支持部门。他们可以为您提供适当的指导。

## 预留节点示例

本部分中的情景演示了节点如何根据按需费率和打折费率（后者的预留详细信息如下所示）产生费用：

- 区域：美国西部（俄勒冈）
- 节点类型：ra3.xlplus
- 付款选项：No Upfront

- 持续时间：一年
- 预留节点的数量：16

## 示例 1

您在美国西部（俄勒冈州）区域有一个集群，其中有 20 个节点。

在此情景中，其中 16 个节点享受预留打折费率，但集群中的另外 4 个节点以按需费率计费。

## 示例 2

您在美国西部（俄勒冈州）区域有一个集群，其中有 12 个节点。

在此情景中，集群中的全部 12 个节点都享受预留打折费率。不过，您还需要为预留中的剩余预留节点付费，即使您当前并没有正在运行的集群可供这些预留节点应用于其中。

## 示例 3

您在美国西部（俄勒冈州）区域有一个集群，其中有 12 个节点。您采用此配置运行该集群几个月时间，然后您需要向其中添加节点。您调整了集群的大小，选择相同的节点类型并指定共计 16 个节点。

在此情景中，您将以打折费率为这 16 个节点付费。在整年的持续时间里您的费用保持不变，因为集群中的节点数量与您预留的节点数量相同。

## 示例 4

您在美国西部（俄勒冈州）区域有一个集群，其中有 16 个节点。您采用此配置运行该集群几个月时间，然后您需要添加节点。您调整了集群的大小，选择相同的节点类型并指定共计 20 个节点。

在此情景中，您将以打折费率为调整大小之前的所有节点付费。调整大小之后，您将在一年中剩下的时间以打折费率为 16 个节点付费，并以按需费率为添加到集群中的另外 4 个节点付费。

## 示例 5

您在美国西部（俄勒冈州）区域有两个集群。一个集群中有 6 个节点，另一个集群中有 10 个节点。

在此情景中，您将以打折费率为所有节点付费，因为两个集群中的节点总数与您预留的节点数量相同。

## 示例 6

您在美国西部（俄勒冈州）区域有两个集群。一个集群中有 4 个节点，另一个集群中有 6 个节点。

在此情景中，您将以打折费率为正在运行的集群中的 10 个节点付费；此外，您还需以打折费率为预留的额外 6 个节点，即使您当前并没有任何正在运行的集群可供这些预留节点应用于其中。

# 购买预留节点

可以使用 AWS 管理控制台或 AWS CLI 购买预留节点产品，并查看当前和过去的预留。

AWS 管理控制台

要购买预留节点

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Clusters（集群），然后选择 Reserved nodes（预留节点）以显示预留节点列表。
3. 选择 Purchase reserved nodes（购买预留节点）显示页面，以选择要购买的节点的属性。
4. 输入节点的属性，然后选择 Purchase reserved nodes（购买预留节点）。

在您购买产品之后，Reserved Node 列表将显示您的预留和每个预留的详细信息，例如节点类型、节点数量和预留的状态。有关预留的更多信息，如需了解更多信息，请参阅预留节点的工作方式。

要升级预留节点，请使用 AWS CLI。

您无法将所有节点类型都转换为预留节点，也有可能现有预留节点无法续订。这可能是因为该节点类型已停用。联系客户支持部门，以续订已停用的节点类型。

AWS CLI

要利用 AWS CLI 升级预留节点预留

1. 获取 ReservedNodeOfferingID 的列表以找到可满足您对支付类型、期限和费用的要求的产品。以下示例说明了该步骤。

```
aws redshift get-reserved-node-exchange-offerings --reserved-node-id xxxxxxxx-
xxxx-xxxx-xxxx-xxxxxxxxxxxx
{
    "ReservedNodeOfferings": [
        {
            "Duration": 31536000,
```

```
            "ReservedNodeOfferingId": "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy",
            "UsagePrice": 0.0,
            "NodeType": "dc2.large",
            "RecurringCharges": [
                {
                    "RecurringChargeFrequency": "Hourly",
                    "RecurringChargeAmount": 0.2
                }
            ],
            "CcurrencyCode": "USD",
            "OfferingType": "No Upfront",
            "ReservedNodeOfferingType": "Regular",
            "FixedPrice": 0.0
        }
    ]
}
```

2. 调用 accept-reserved-node-exchange 并为要与在上一步中获取的 ReservedNodeOfferingID 一起交换的 DC1 预留节点提供 ID。

   以下示例说明了该步骤。

```
aws redshift accept-reserved-node-exchange --reserved-node-id xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxxx --target-reserved-node-offering-id yyyyyyyy-yyyy-yyyy-
yyyy-yyyyyyyyyyyy
{
    "ExchangedReservedNode": {
        "UsagePrice": 0.0,
        "OfferingType": "No Upfront",
        "State": "exchanging",
        "FixedPrice": 0.0,
        "CurrencyCode": "USD",
        "ReservedNodeId": "zzzzzzzz-zzzz-zzzz-zzzz-zzzzzzzzzzzz",
        "NodeType": "dc2.large",
        "NodeCount": 1,
        "RecurringCharges": [
            {
                "RecurringChargeFrequency": "Hourly",
                "RecurringChargeAmount": 0.2
            }
        ],
        "ReservedNodeOfferingType": "Regular",
```

```
        "StartTime": "2018-06-27T18:02:58Z",
        "ReservedNodeOfferingId": "yyyyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyyy",
        "Duration": 31536000
    }
}
```

您可以通过调用 describe-reserved-nodes 并检查 Node type 的值来确认该交换是否完成。

# Amazon Redshift 中的安全性

AWS的云安全性的优先级最高。作为 AWS 客户，您将从专为满足大多数安全敏感型企业的要求而打造的数据中心和网络架构中受益。

安全性是 AWS 和您的共同责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云的安全性 – AWS 负责保护在 AWS 云中运行 AWS 服务的基础设施。AWS 还向您提供可安全使用的服务。作为 AWS 合规计划的一部分，我们的安全有效性会定期由第三方审核员进行测试和验证。要了解适用于 Amazon Redshift 的合规性计划，请参阅合规性计划范围内的 AWS 服务。
- 云中的安全性 - 您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您组织的要求以及适用的法律法规。

您可以在以下四个级别控制对 Amazon Redshift 资源的访问：

- 集群管理 – 创建、配置和删除集群的能力由授予给与您的 AWS 安全凭证关联的用户或账户的权限进行控制。具有适当权限的用户可以使用 AWS 管理控制台、AWS Command Line Interface（CLI）或 Amazon Redshift 应用程序编程接口（API）来管理其集群。可以借助 IAM 策略对这种访问加以管理。

  > **Important**
  >
  > Amazon Redshift 提供了有关管理权限、身份和安全访问的一系列最佳实践。我们建议您在开始使用 Amazon Redshift 时熟悉这些内容。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

- 集群连接 – Amazon Redshift 安全组用于指定有权连接到无类别域间路由 (CIDR) 格式的 Amazon Redshift 集群的 AWS 实例。有关创建 Amazon Redshift、Amazon EC2 和 Amazon VPC 安全组以及将其与集群关联的信息，请参阅Amazon Redshift 安全组。
- 数据库访问 – 访问数据库对象（如表和视图）的能力由 Amazon Redshift 数据库中的数据库用户账户控制。用户只能访问其用户账户有权访问的数据库中的资源。您可以创建这些 Amazon Redshift 用户账户并使用 CREATE USER、CREATE GROUP、GRANT 和 REVOKE SQL 语句管理权限。有关更多信息，请参阅 Amazon Redshift 数据库开发人员指南中的管理数据库安全。
- 临时数据库凭证和单点登录 –除了使用 SQL 命令（如 CREATE USER 和 ALTER USER）创建和管理数据库用户之外，您还可以使用自定义 Amazon Redshift JDBC 或 ODBC 驱动程序配置 SQL 客户端。这些驱动程序将创建数据库用户和临时密码的过程作为数据库登录过程的一部分进行管理。

这些驱动程序将基于 AWS Identity and Access Management (IAM) 身份验证来验证数据库用户的身份。如果您已在 AWS 外部管理用户身份，则可以使用符合 SAML 2.0 标准的身份提供者 (IdP) 来管理对 Amazon Redshift 资源的访问。您使用 IAM 角色将 IdP 和 AWS 配置为允许联合身份用户生成临时数据库凭证并登录 Amazon Redshift 数据库。有关更多信息，请参阅 使用 IAM 身份验证生成数据库用户凭证。

此文档将帮助您了解如何在使用 Amazon Redshift 时应用责任共担模式。以下主题说明如何配置 Amazon Redshift 以实现您的安全性和合规性目标。您还会了解如何使用其他 AWS 服务以帮助您监控和保护 Amazon Redshift 资源。

主题


- Amazon Redshift 中的数据保护
- Amazon Redshift 中的 Identity and Access Management
- 使用 AWS Secrets Manager 管理 Amazon Redshift 管理员密码
- Amazon Redshift 中的日志记录和监控
- Amazon Redshift 的合规性验证
- Amazon Redshift 中的恢复能力
- Amazon Redshift 中的基础设施安全性
- Amazon Redshift 中的配置和漏洞分析


# Amazon Redshift 中的数据保护

AWS 责任共担模式适用于 Amazon Redshift 中的数据保护。如该模式中所述，AWS 负责保护运行所有 AWS Cloud 的全球基础结构。您负责维护对托管在此基础结构上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅通用数据保护条例（GDPR）中心。

出于数据保护目的，建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management（IAM）设置单个用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证（MFA）。
- 使用 SSL/TLS 与 AWS 资源进行通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。

- 使用 AWS CloudTrail 设置 API 和用户活动日记账记录。有关使用 CloudTrail 跟踪来捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的使用 CloudTrail 跟踪。
- 使用 AWS 加密解决方案以及 AWS 服务中的所有默认安全控制。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果在通过命令行界面或 API 访问 AWS 时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-3 版》https://aws.amazon.com/compliance/fips/。

强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括当您通过控制台、API、AWS CLI 或 AWS SDK 使用 Amazon Redshift 或其他 AWS 服务时。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供 URL，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

## 数据加密

数据保护指在数据传输（发往和离开 Amazon Redshift 时）和处于静态（存储在 Amazon Redshift 数据中心的磁盘上时）期间保护数据。可以使用 SSL 或使用客户端加密保护传输中的数据。您可以通过以下选项在 Amazon Redshift 中保护静态数据。

- 使用服务器端加密 – 您请求 Amazon Redshift 在将数据保存到数据中心的磁盘上之前加密对象，并在下载对象时进行解密。
- 使用客户端加密 – 您可以在客户端加密数据并将加密的数据上载到 Amazon Redshift。在这种情况下，您需要管理加密过程、加密密钥和相关的工具。

### 静态加密

服务器端加密是静态数据加密，即，Amazon Redshift 在将数据写入其数据中心时选择性地对其进行加密，并在您访问时进行解密。只要您验证了您的请求并且拥有访问权限，您访问加密和未加密数据的方式就没有区别。

Amazon Redshift 通过加密为静态数据提供保护。（可选）您可以通过高级加密标准 AES-256，为存储在集群中磁盘上的所有数据以及 Amazon S3 中的所有备份提供保护。

要管理用于加密和解密 Amazon Redshift 资源的密钥，您可以使用 AWS Key Management Service (AWS KMS)。AWS KMS 将安全、高度可用的硬件和软件结合起来，提供可扩展到云的密钥管理系

统。利用 AWS KMS，您可创建加密密钥并定义控制这些密钥的使用方式的策略。AWS KMS 支持 AWS CloudTrail，因此，您可审核密钥使用情况以验证密钥是否使用得当。您可将 AWS KMS 密钥与 Amazon Redshift 和支持的 AWS 服务结合使用。有关支持 AWS KMS 的服务的列表，请参阅《AWS Key Management Service 开发人员指南》中的 AWS 服务如何使用 AWS KMS。

如果您选择使用 AWS Secrets Manager 来管理预置集群或无服务器命名空间的管理员密码，Amazon Redshift 还可接受额外的 AWS KMS 密钥，可供 AWS Secrets Manager 用于加密您的凭证。此额外的密钥可以是从 AWS Secrets Manager 自动生成的密钥，也可以是您提供的自定义密钥。

Amazon Redshift 查询编辑器 v2 安全存储输入到查询编辑器中的信息，如下所示：

- 用于加密查询编辑器 v2 数据的 KMS 密钥的 Amazon 资源名称 (ARN)。
- 数据库连接信息。
- 文件和文件夹的名称和内容。

Amazon Redshift 查询编辑器 v2 使用 KMS 密钥或服务账户 KMS 密钥的数据块级加密来加密信息。Amazon Redshift 数据的加密由 Amazon Redshift 集群属性控制。

主题

- Amazon Redshift 数据库加密

## Amazon Redshift 数据库加密

在 Amazon Redshift 中，默认对数据库进行加密以保护静态数据。数据库加密将应用于集群及其快照。

您可以修改未加密的集群以使用 AWS Key Management Service（AWS KMS）加密。为此，您可以使用 AWS 拥有的密钥或客户托管式密钥。当您修改集群以启用 AWS KMS 加密时，Amazon Redshift 会自动将您的数据迁移到新加密的集群。从加密集群创建的快照也会被加密。您还可以通过修改集群和更改 Encrypt database（加密数据库）选项将加密的集群迁移到未加密的集群。有关更多信息，请参阅更改集群加密。

尽管在创建集群后，您仍可以将默认加密集群转换为未加密集群，但建议您将包含敏感数据的集群保持加密状态。此外，根据管理数据的准则或法规，您可能需要使用加密。例如，支付卡行业数据安全标准 (PCI DSS)、萨班斯-奥克斯利法 (SOX)、健康保险流通与责任法案 (HIPAA) 以及其他此类法规为处理特定类型数据提供了准则。

Amazon Redshift 使用加密密钥层次结构来加密数据库。您可以使用 AWS Key Management Service (AWS KMS) 或硬件安全模块 (HSM) 来管理该层次结构中的顶级加密密钥。Amazon Redshift 用于加密

的流程因您管理密钥的方式而异。Amazon Redshift 自动集成了 AWS KMS 但没有集成 HSM。当您使用 HSM 时，必须使用客户端和服务器证书在 Amazon Redshift 和 HSM 之间配置受信任的连接。

> **Important**
>
> 禁用客户自主管理型 KMS 密钥后，Amazon Redshift 可能会失去对预置集群或无服务器命名空间的 KMS 密钥的访问权限。在这些情况下，Amazon Redshift 对 Amazon Redshift 数据仓库进行备份，并将其置于 `inaccessible-kms-key` 状态达 14 天。如果您在这段时间内恢复 KMS 密钥，Amazon Redshift 将恢复访问权限，仓库将正常运行。如果 14 天的期限结束时没有恢复 KMS 密钥，Amazon Redshift 将删除该数据仓库。当仓库处于 `inaccessible-kms-key` 状态时，仓库具有以下特征：
>
> - 不能对数据仓库运行任何查询。
> - 如果数据仓库是数据共享的生产者仓库，则无法从使用者仓库对其运行数据共享查询。
> - 无法创建跨区域快照副本。
>
> 有关恢复已禁用的 KMS 密钥的信息，请参阅《AWS Key Management Service Developer Guide》中的 Enable and disable keys。如果删除了仓库的 KMS 密钥，则可以在删除 `inaccessible-kms-key` 状态的仓库之前使用备份创建新的数据仓库。

改进加密过程以提高性能和可用性

RG 和 RA3 节点的加密

对 RG 和 RA3 节点加密过程的更新大幅改进了体验。读取和写入查询都可以在此过程中运行，而加密对性能的影响较小。此外，加密完成的速度要快得多。更新的过程步骤包括还原操作，以及将集群元数据迁移到目标集群的操作。例如，改进的体验适用于诸如 AWS KMS 的加密类型。当您拥有 PB 级的数据量时，操作时间从几周缩短到几天。

在加密集群之前，如果您计划继续运行数据库工作负载，则可以通过添加具有弹性大小调整功能的节点来提高性能并加快此过程。在加密过程中不能使用弹性大小调整功能，因此，请在加密之前执行此操作。请注意，添加节点通常会导致成本增加。

使用其他节点类型加密

当您加密具有 DC2 节点的集群时，您无法像使用 RG 或 RA3 节点那样运行写入查询。只能运行读取查询。

## RG 和 RA3 节点加密的用法说明

以下见解和资源有助于您做好准备进行加密并监控整个加密过程。

- 开始加密后运行查询 - 加密开始后，大约十五分钟内即可进行读取和写入。完成完整加密过程需要的时间取决于集群上的数据量和工作负载级别。
- 加密需要多长时间？ – 加密数据的时间取决于多个因素：包括正在运行的工作负载数量、所用的计算资源、节点的数量，以及节点的类型。我们建议您首先在测试环境中执行加密。根据经验，如果您处理的数据量达到 PB 级，则可能需要 1-3 天才能完成加密。
- 我如何知道加密已完成？ – 启用加密功能后，第一张快照的完成即确认加密已完成。
- 回滚加密 - 如果您需要回滚加密操作，最好是从加密启动之前进行的最新备份中还原。在最后一次备份之后，必须重新应用任何新的更新（更新/删除/插入）。
- 执行表还原 – 请注意，您无法将表从未加密集群还原到加密集群。
- 加密单节点集群 – 加密单节点集群具有性能限制。此过程所花的时间比多节点集群加密要长。
- 加密后创建备份 – 加密集群中的数据时，只有在集群完全加密后才会创建备份。此过程所需的时间量可能会有所不同。备份所需的时间可能为数小时到数天，具体取决于集群大小。加密完成后，可能有一段延迟，然后才会创建备份。

  注意，由于在加密过程中发生备份和还原操作，因此不会保留使用 `BACKUP NO` 创建的任何表或实体化视图。有关更多信息，请参阅 CREATE TABLE 或 CREATE MATERIALIZED VIEW。

### 主题


- 使用 AWS KMS 的加密
- 使用硬件安全模块的加密
- 加密密钥轮换
- 更改集群加密
- 迁移到 HSM 加密的集群
- 轮换加密密钥


## 使用 AWS KMS 的加密

当您选择 AWS KMS 对 Amazon Redshift 进行密钥管理时，加密密钥层次结构有四层。按层次顺序，这些密钥是根密钥、集群加密密钥（CEK）、数据库加密密钥（DEK）和数据加密密钥。

当您启动集群时，Amazon Redshift 会返回 Amazon Redshift 或您的 AWS 账户所创建或有权在 AWS KMS 中使用的 AWS KMS keys 的列表。在加密层次结构中选择要用作根密钥的 KMS 密钥。

默认情况下，Amazon Redshift 会选择自动生成的 AWS 拥有的密钥作为您的 AWS 账户的根密钥，以便在 Amazon Redshift 中使用。

如果不希望使用默认密钥，在 Amazon Redshift 中启动集群之前，您必须在 AWS KMS 中单独拥有（或创建）一个客户托管式 KMS 密钥。客户托管式密钥可为您提供更大灵活度，包括创建、转动、禁用、定义访问控制，以及审计用于保护数据的加密密钥的能力。有关创建 KMS 密钥的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的创建密钥。

如果您希望使用其他 AWS 账户中的 AWS KMS 密钥，则您必须有权使用该密钥并在 Amazon Redshift 中指定其 Amazon Resource Name (ARN)。有关在 AWS KMS 中访问密钥的更多信息，请参阅 AWS Key Management Service 开发人员指南中的控制对您的密钥的访问。

当您选择根密钥之后，Amazon Redshift 会请求 AWS KMS 生成数据密钥，并使用所选的根密钥对其进行加密。此数据密钥在 Amazon Redshift 中用作 CEK。AWS KMS 将已加密的 CEK 导出到 Amazon Redshift 中，该 CEK 以及对 KMS 密钥的授权和 CEK 的加密上下文在独立于该集群的网络中的磁盘内部存储。只有加密的 CEK 才会导出到 Amazon Redshift；KMS 密钥仍保留在 AWS KMS 中。Amazon Redshift 还会通过安全通道将已加密的 CEK 传递到集群中，并将其加载到内存中。然后，Amazon Redshift 会调用 AWS KMS 以解密 CEK，并将已解密的 CEK 加载到内存中。有关授予、加密上下文和其他 AWS KMS 相关概念，请参阅 AWS Key Management Service 开发人员指南中的概念。

接着，Amazon Redshift 会随机生成一个密钥以用作 DEK，并将其加载到集群的内存中。已解密的 CEK 用于加密 DEK，然后，Amazon Redshift 便会通过安全通道从该集群传递 DEK，以使其在独立于该集群的网络中的磁盘内部存储。与 CEK 一样，DEK 的加密版本和解密版本都会加载到集群中的内存中。然后，DEK 的解密版本将用于加密为数据库中的每个数据块随机生成的各个加密密钥。

当该集群重启时，Amazon Redshift 先使用内部存储的加密版 CEK 和 DEK，将其重新加载到内存中，然后调用 AWS KMS 以再次使用 KMS 密钥解密 CEK，以便其加载到内存中。然后使用解密的 CEK 再次解密 DEK，并将解密的 DEK 加载到内存中，并根据需要用于加密和解密数据块密钥。

有关创建使用 AWS KMS 密钥加密的 Amazon Redshift 集群的更多信息，请参阅创建集群。

将 AWS KMS 加密的快照复制到另一个 AWS 区域

AWS KMS 密钥特定于 AWS 区域。如果要允许将 Amazon Redshift 快照从一个加密的源集群复制到另一个 AWS 区域，并为目标中的快照使用您自己的 AWS KMS 密钥，您需要为 Amazon Redshift 配置

授予，以便在目标 AWS 区域中使用根密钥。此授予使 Amazon Redshift 能够在目标 AWS 区域中加密快照。如果您希望通过 AWS 区域拥有的密钥对目标中的快照进行加密，则无需在目标 AWS 区域中配置任何授予。有关跨区域快照复制的更多信息，请参阅将快照复制到其它 AWS 区域。

> **Note**
>
> 如果您启用已加密集群中的快照复制功能，并针对根密钥使用 AWS KMS，则无法重命名集群，因为集群名称是加密上下文的一部分。如果您必须重命名集群，则可以禁用在源 AWS 区域中复制快照，重命名集群，然后再次配置并启用快照复制。

配置复制快照的授权的过程如下所示。

1. 在目标 AWS 区域中，通过执行以下操作创建快照复制授权：
   - 如果您没有 AWS KMS 密钥可用，请创建一个。有关创建 AWS KMS 密钥的更多信息，请参阅 AWS Key Management Service 开发人员指南中的创建密钥。
   - 指定快照复制授权的名称。此名称必须在您的 AWS 账户的 AWS 区域中唯一。
   - 为您创建的授权指定 AWS KMS 密钥 ID。如果您未指定密钥 ID，则该授予会应用于您的默认密钥。
2. 在源 AWS 区域中，启用快照复制并指定您在目标 AWS 区域中创建的快照复制授权的名称。

前述流程仅适用于您使用 AWS CLI、Amazon Redshift API 或开发工具包启用快照复制功能的情况。如果您使用控制台，在您启用跨区域快照复制时，Amazon Redshift 将提供相应的工作流程以配置授权。有关使用控制台为 AWS KMS 加密的集群配置跨区域快照复制的更多信息，请参阅为 AWS KMS 加密的集群配置跨区域快照副本。

快照复制到目标 AWS 区域之前，Amazon Redshift 使用源 AWS 区域中的根密钥解密快照并使用 Amazon Redshift 内部管理的随机生成的 RSA 密钥临时对其进行重新加密。然后，Amazon Redshift 将快照通过安全通道复制到目标 AWS 区域，使用内部托管式 RSA 密钥解密快照，然后使用目标 AWS 区域中的根密钥重新加密快照。

使用硬件安全模块的加密

如果您不使用 AWS KMS 管理密钥，则可以使用硬件安全模块 (HSM) 在 Amazon Redshift 中管理密钥。

> **Important**
>
> DC2、RG 和 RA3 节点类型不支持 HSM 加密。

HSM 是直接控制密钥生成和管理的设备。HSM 使密钥管理独立于应用程序和数据库层，以提供更强的安全性。Amazon Redshift 支持 AWS CloudHSM Classic 进行密钥管理。使用 HSM 管理加密密钥的加密流程与 AWS KMS 不同。

> **Important**
>
> Amazon Redshift 仅支持 AWS CloudHSM Classic。我们不支持较新的 AWS CloudHSM 服务。
> AWS CloudHSM Classic 不对新客户开放。有关更多信息，请参阅 CloudHSM Classic 定价。AWS CloudHSMClassic 并非在所有 AWS 区域可用。有关可用 AWS 区域的更多信息，请参阅 AWS 区域列表。

当您配置集群以使用 HSM 时，Amazon Redshift 会向 HSM 发出请求，以生成并存储要用作 CEK 的密钥。不过，与 AWS KMS 不同，HSM 不会将 CEK 导出到 Amazon Redshift 中。Amazon Redshift 会在集群中随机生成 DEK，并将其传递到 HSM 中，由 CEK 进行加密。HSM 会将已加密的 DEK 返回到 Amazon Redshift 中，由随机生成的内部根密钥进行加密并在独立于该集群的网络中的磁盘内部存储。Amazon Redshift 还会将解密版 DEK 加载到集群的内存中，以便 DEK 可用于加密和解密数据块的各个密钥。

如果该集群重启，则 Amazon Redshift 会使用内部根密钥解密内部存储的双重加密 DEK，以将内部存储的 DEK 还原为 CEK 加密状态。然后，CEK 加密的 DEK 会传递到 HSM 进行加密并传递回 Amazon Redshift，以便再次加载到内存中用于各个数据块密钥。

在 Amazon Redshift 和 HSM 之间配置受信任的连接

如果您选择使用 HSM 管理集群密钥，则需要在 Amazon Redshift 和 HSM 之间配置受信任的网络连接。执行此操作需要配置客户端和服务器证书。在加密和解密操作过程中，受信任的连接用于在 HSM 和 Amazon Redshift 之间传递加密密钥。

Amazon Redshift 会通过随机生成的私有和公有键前缀创建公有客户端证书。这些都进行了加密，并在内部存储。您可以在 HSM 中下载并注册公有客户端证书，并将其分配给适用的 HSM 分区。

您需要为 Amazon Redshift 提供 HSM IP 地址、HSM 分区名称、HSM 分区密码以及公有 HSM 服务器证书，该证书使用内部根密钥进行了加密。Amazon Redshift 会完成配置流程并验证其能否连接到

HSM。如果无法使用，则集群将进入 INCOMPATIBLE_HSM 状态，并且不会创建集群。在这种情况下，您必须删除不完整的集群，然后重试。

> **Important**
>
> 当您将集群修改为使用其他 HSM 分区时，Amazon Redshift 将验证它是否能连接到新分区，但不会验证是否存在有效的加密密钥。在使用新分区之前，您必须将密钥复制到新分区。如果集群已重新启动，而 Amazon Redshift 无法找到有效密钥，重新启动将失败。有关更多信息，请参阅在 HSM 中复制密钥。

初始配置之后，如果 Amazon Redshift 无法连接到 HSM，则系统会记录一个事件。有关这些事件的更多信息，请参阅 Amazon Redshift 事件通知。

加密密钥轮换

在 Amazon Redshift 中，您可以轮换已加密集群的加密密钥。当您启动密钥轮换流程时，Amazon Redshift 会轮换指定集群以及该集群的任何自动或手动快照的 CEK。Amazon Redshift 还会轮换指定集群的 DEK，但不会轮换快照的 DEK，而这些快照在 Amazon Simple Storage Service（Amazon S3）内部存储并使用现有 DEK 进行了加密。

在轮换进行过程中，该集群将处于 ROTATING_KEYS 状态一直到轮换完成为止，届时，该集群将还原为 AVAILABLE 状态。Amazon Redshift 会在密钥轮换过程中处理解密和重新加密。

> **Note**
>
> 如果没有源集群，则无法轮换快照的密钥。在删除集群之前，请考虑其快照是否依赖密钥轮换。

由于集群在密钥轮换过程中暂时不可用，因此您应该只在数据需求需要时或在怀疑密钥可能已被破坏时轮换密钥。作为最佳实践，您应该查看存储的数据的类型，并计划对该数据进行加密的密钥的轮换频率。轮换密钥的频率取决于贵公司的数据安全策略以及关于敏感数据和监管合规的任何行业标准。确保您的计划平衡安全需求和集群的可用性考虑因素。

有关轮换密钥的更多信息，请参阅轮换加密密钥。

## 更改集群加密

可以使用 AWS 拥有的密钥或客户自主管理型密钥来修改未加密的集群，以使用 AWS Key Management Service（AWS KMS）加密。当您修改集群以启用 AWS KMS 加密时，Amazon Redshift 会自动将您的数据迁移到新加密的集群。还可以通过使用 AWS CLI（而不是 AWS 管理控制台）修改集群，来将加密的集群迁移到未加密的集群。

在迁移操作过程中，您的集群在只读模式下可用，并且集群状态显示为调整大小。

如果您的集群配置为启用跨 AWS 区域快照副本，您必须在更改加密之前将其禁用。有关更多信息，请参阅将快照复制到其它 AWS 区域和为 AWS KMS 加密的集群配置跨区域快照副本。您无法通过修改集群启用硬件安全模块 (HSM) 加密。而是创建一个新的 HSM 加密集群，并将您的数据迁移到新集群。有关更多信息，请参阅 迁移到 HSM 加密的集群。

### Amazon Redshift console

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Clusters（集群），然后选择要修改加密的集群。
3. 选择 Properties (属性)。
4. 在 Database configurations（数据库配置）部分，选择 Edit（编辑），然后选择 Edit encryption（编辑加密）。
5. 选择其中一个加密选项，然后选择 Save changes（保存更改）。

### AWS CLI

若要修改未加密的集群以使用 AWS KMS，请运行 `modify-cluster` CLI 命令并指定 `--encrypted`，如下所示。预设情况下，使用默认 KMS 密钥。要指定客户托管式密钥，请包含 `--kms-key-id` 选项。

```
aws redshift modify-cluster --cluster-identifier <value> --encrypted --kms-key-id
 <value>
```

要从集群中删除加密，请运行以下 CLI 命令。

```
aws redshift modify-cluster --cluster-identifier <value> --no-encrypted
```

迁移到 HSM 加密的集群

要将未加密的集群迁移到使用硬件安全模块 (HSM) 加密的集群，请创建新的加密集群并将数据移动到新集群。您不能通过修改集群迁移到 HSM 加密的集群。

要从未加密的集群迁移到 HSM 加密的集群，首先从现有的源集群中卸载数据。然后使用所选的加密设置将数据重新加载到新的目标集群中。有关启动加密集群的更多信息，请参阅Amazon Redshift 数据库加密。

在迁移过程中，源集群可用于只读查询，直到最后一步。最后一步是重命名目标集群和源集群，用于切换端点，以便将所有流量路由到新的目标集群。在重命名后重新启动之前，目标集群不可用。在传输数据时，暂停源集群上的所有数据加载和其他写入操作。

准备迁移

1. 确定与 Amazon Redshift 交互的所有关联系统，例如业务情报 (BI) 工具以及提取、转换和加载 (ETL) 系统。
2. 确定用于测试迁移的验证查询。

   例如，您可以使用以下查询来查找用户定义表的数目。

   ```
   select count(*)
   from pg_table_def
   where schemaname != 'pg_catalog';
   ```

   以下查询返回所有用户定义表的列表以及每个表中的行数。

   ```
   select "table", tbl_rows
   from svv_table_info;
   ```

3. 选择迁移的好时机。要查找集群使用率最低的时间，请监控 CPU 利用率和数据库连接数量等集群指标。有关更多信息，请参阅 查看集群性能数据。
4. 删除未使用的表。

   要创建表列表和查询每个表的次数，请运行以下查询。

   ```
   select database,
   schema,
   table_id,
   "table",
   ```

```
round(size::float/(1024*1024)::float,2) as size,
sortkey1,
nvl(s.num_qs,0) num_qs
from svv_table_info t
left join (select tbl,
perm_table_name,
count(distinct query) num_qs
from stl_scan s
where s.userid > 1
and   s.perm_table_name not in ('Internal worktable','S3')
group by tbl,
perm_table_name) s on s.tbl = t.table_id
where t."schema" not in ('pg_internal');
```

5. 启动新的加密集群。

   对目标集群使用与源集群相同的端口号。有关启动加密集群的更多信息，请参阅Amazon Redshift 数据库加密。

6. 设置卸载和加载过程。

   您可以使用 Amazon Redshift 卸载/复制实用工具来帮助您在集群之间迁移数据。该实用工具会将源集群的数据导出到 Simple Storage Service（Amazon S3）上的某个位置。使用 AWS KMS 加密数据。然后，该实用工具会自动将数据导入到目标中。或者，您可以使用该实用程序在迁移完成后清除 Simple Storage Service（Amazon S3）。

7. 运行测试以验证您的过程并估计必须暂停写入操作的时间。

   在卸载和加载操作期间，通过暂停数据加载和其他写入操作来维护数据一致性。使用最大的表之一，运行卸载和加载过程，以帮助您估计时间。

8. 创建数据库对象，如 schema、视图和表。要帮助您生成必要的数据定义语言 (DDL) 语句，您可以使用 AWS GitHub 存储库中的 AdminViews 中的脚本。

要迁移集群

1. 停止源集群上的所有 ETL 流程。

   要确认进程中未执行任何写入操作，请使用 Amazon Redshift 管理控制台来监控写入 IOPS。有关更多信息，请参阅 查看集群性能数据。

2. 运行您之前确定的验证查询，以便在迁移前收集有关未加密源集群的信息。

3. （可选）创建一个工作负载管理 (WLM) 队列，以使用源集群和目标集群中的最大可用资源。例如，创建名为 `data_migrate` 的队列并将队列配置为内存 95%，并发性为 4。有关更多信息，请参阅 Amazon Redshift 数据库开发人员指南中的基于用户组和查询组将查询路由至队列中。
4. 使用 `data_migrate` 队列，运行 UnloadCopyUtility。

   使用 Amazon Redshift 控制台监控 UNLOAD 和 COPY 进程。
5. 再次运行验证查询，并验证结果是否与源集群的结果匹配。
6. 重命名源集群和目标集群以交换端点。为避免中断，请在工作时间以外执行此操作。
7. 验证您是否可以使用所有 SQL 客户端（如 ETL 和报告工具）连接到目标集群。
8. 关闭未加密的源集群。

### 轮换加密密钥

您可以借助 Amazon Redshift 控制台以使用以下过程来轮换加密密钥。

要为集群轮换加密密钥

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Clusters（集群），然后选择要更新加密密钥的集群。
3. 对于 Actions（操作），选择 Rotate encryption（轮换加密）以显示 Rotate encryption keys（轮换加密密钥）页面。
4. 正在 Rotate encryption keys（轮换加密密钥）页面上，选择 Rotate encryption keys（轮换加密密钥）。

## 传输中加密

可以配置环境来保护传输中数据的机密性和完整性。

以下详细信息适用于加密在 Amazon Redshift 集群和 SQL 客户端之间通过 JDBC/ODBC 传输的数据：

- 您可以通过 Java 数据库连接 (JDBC) 和开放式数据库连接 (ODBC) 这两种连接将 SQL 客户端工具连接到 Amazon Redshift 集群。
- Amazon Redshift 支持安全套接字层 (SSL) 连接来加密数据和服务器证书，以验证客户端连接到的服务器证书。客户端连接到 Amazon Redshift 集群的领导节点。有关更多信息，请参阅 配置连接的安全选项。

- 为了支持 SSL 连接，Amazon Redshift 会在每个集群中创建并安装 AWS Certificate Manager (ACM) 颁发的证书。有关更多信息，请参阅 将 SSL 连接过渡到 ACM 证书。
- 为保护AWS云中的传输中数据，Amazon Redshift 使用硬件加速的 SSL 与 Amazon S3 或 Amazon DynamoDB 通信以执行 COPY、UNLOAD、备份和还原操作。

以下详细信息适用于加密在 Amazon Redshift 集群与 Amazon S3 或 DynamoDB 之间传输的数据：

- Amazon Redshift 使用硬件加速的 SSL 与 Amazon S3 或 DynamoDB 通信以执行 COPY、UNLOAD、备份和还原操作。
- Redshift Spectrum 支持使用受 AWS Key Management Service (KMS) 管理的账户默认密钥的 Amazon S3 服务器端加密 (SSE)。
- 可以使用 Amazon S3 和 AWS KMS 来加密 Amazon Redshift 加载。有关更多信息，请参阅使用 Amazon S3 和 AWS KMS 加密您的 Amazon Redshift 加载。

以下详细信息适用于对在 AWS CLI、SDK 或 API 客户端与 Amazon Redshift 端点之间传输的数据进行加密和签名：

- Amazon Redshift 为加密传输中的数据提供了 HTTPS 端点。
- 为了保护向 Amazon Redshift 出的 API 请求的完整性，API 调用必须由调用者签名。调用由 X.509 证书或客户的 AWS 秘密访问密钥根据前面版本 4 签名流程 (Sigv4) 签名。有关更多信息，请参阅《AWS 一般参考》中的签名版本 4 签名流程。
- 使用 AWS CLI 或 AWS 开发工具包之一向 AWS 发出请求。这些工具会自动使用您在配置工具时指定的访问密钥为您签署请求。

以下详细信息适用于加密在 Amazon Redshift 集群与 Amazon Redshift 查询编辑器 V2 之间传输的数据：

- 数据通过 TLS-加密通道在查询编辑器 V2 与 Amazon Redshift 集群之间传输。

通过 Amazon Redshift 执行 VPC 加密控制功能

Amazon Redshift 支持 VPC 加密控制，这是一项安全功能，可助您对某个区域中的 VPC 内部以及跨 VPC 的所有流量强制执行传输中加密。本文档介绍如何在 Amazon Redshift 集群和无服务器工作组中执行 VPC 加密控制功能。

VPC 加密控制功能提供了集中化控制能力，以便在您的 VPC 内部监控并强制执行传输中加密。在强制模式下启用此功能后，它可确保所有网络流量都在硬件层（使用 AWS Nitro System）或应用层（使用 TLS/SSL）进行加密。

Amazon Redshift 与 VPC 加密控制功能集成，助您满足医疗保健（HIPAA）、政府（FedRAMP）和金融（PCI DSS）等行业的合规要求。

VPC 加密控制功能与 Amazon Redshift 的协作方式

VPC 加密控制功能在两种模式下运行：

- 监控模式：可清晰呈现流量的加密状态，并帮助识别允许非加密流量的资源。
- 强制模式：禁止在 VPC 中创建或使用允许非加密流量的资源。所有流量都必须在硬件层（基于 Nitro 的实例）或应用层（TLS/SSL）进行加密。

使用 VPC 加密控制功能的要求

实例类型要求

Amazon Redshift 要求使用基于 Nitro 的实例来支持 VPC 加密控制功能。所有现代 Redshift 实例类型都支持必要的加密功能。

SSL/TLS 要求

在强制模式下启用 VPC 加密控制功能时，require_ssl 参数必须设置为 true 且不能禁用。这可确保所有客户端连接均采用加密的 TLS 连接。

迁移到 VPC 加密控制功能

对于现有集群和工作组

在包含现有 Redshift 集群或无服务器工作组的 VPC 上，无法在强制模式下启用 VPC 加密控制功能。如果您已有集群或工作组，请按以下步骤操作来使用加密控制功能：

1. 创建现有集群或命名空间的快照
2. 创建一个新的 VPC，并在强制模式下启用 VPC 加密控制功能
3. 通过使用下列操作之一，从快照还原至新 VPC：
    - 对于预调配集群：使用 `restore-from-cluster-snapshot` 操作
    - 对于无服务器：在工作组上使用 `restore-from-snapshot` 操作

在 VPC 中创建新的集群或工作组并启用加密控制功能时，必须将 require_ssl 参数设置为 true。

Amazon Redshift 要求使用基于 Nitro 的实例来支持 VPC 加密控制功能。所有现代 Redshift 实例类型都支持必要的加密功能。

SSL/TLS 要求

在强制模式下启用 VPC 加密控制功能时，require_ssl 参数必须设置为 true 且不能禁用。这可确保所有客户端连接均采用加密的 TLS 连接。

注意事项和限制

在 Amazon Redshift 中使用 VPC 加密控制功能时，需注意以下几点：

VPC 状态限制

- 当 VPC 加密控制功能处于 `enforce-in-progress` 状态时，集群和工作组创建操作将被阻止
- 您必须等到 VPC 进入 `enforce` 模式后，再创建新的资源

SSL 配置

- require_ssl 参数：对于在已实施加密的 VPC 中创建的集群与工作组，该参数的值必须始终为 `true`
- 在已实施加密的 VPC 中创建集群或工作组后，`require_ssl` 在其生命周期内不能禁用

区域可用性

在以下区域内，Amazon Redshift Serverless 不支持在强制模式下启用 VPC 加密控制功能：

- 南美洲（圣保罗）
- 欧洲（苏黎世）

## 密钥管理

您可以配置环境以使用密钥保护数据：

- Amazon Redshift 自动与 AWS Key Management Service (AWS KMS) 集成以进行密钥管理。AWS KMS 使用信封加密。有关更多信息，请参阅信封加密。
- 在 AWS KMS 中管理加密密钥时，Amazon Redshift 使用基于密钥的四层架构进行加密。此架构包括随机生成的 AES-256 数据加密密钥、数据库密钥、集群密钥和根密钥。有关更多信息，请参阅 Amazon Redshift 如何使用 AWS KMS。

- 您可以在 AWS KMS 中创建自己的客户托管式密钥。有关更多信息，请参阅创建 KMS 密钥和授权将 AWS KMS 客户托管密钥与 Amazon Redshift（预调配）和 Amazon Redshift Serverless 配合使用。
- 您也可为新 AWS KMS keys 导入自己的密钥材料。有关更多信息，请参阅将密钥材料导入 AWS Key Management Service (AWS KMS) 中。
- Amazon Redshift 支持在外部硬件安全模块 (HSM) 中管理加密密钥。HSM 可以为本地，也可以为 AWS CloudHSM。当您使用 HSM 时，必须使用客户端和服务器证书在 Amazon Redshift 和 HSM 之间配置受信任的连接。Amazon Redshift 仅支持 AWS CloudHSM Classic 进行密钥管理。有关更多信息，请参阅 使用硬件安全模块的加密。有关 AWS CloudHSM 的信息，请参阅什么是 AWS CloudHSM？
- 您可以轮换已加密集群的加密密钥。有关更多信息，请参阅 加密密钥轮换。

## 授权将 AWS KMS 客户托管密钥与 Amazon Redshift（预调配）和 Amazon Redshift Serverless 配合使用

Amazon Redshift 与 AWS Key Management Service（AWS KMS）集成，可以加密您的静态数据。当您使用客户托管密钥时，可以完全控制密钥策略、密钥轮换，以及启用或禁用密钥。有关客户托管密钥的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的客户托管密钥。

当 Amazon Redshift 在加密操作中使用客户托管密钥时，它将代表创建或者修改 Amazon Redshift 预调配集群或 Amazon Redshift Serverless 资源的用户。

所需的权限

要创建和使用通过客户托管密钥加密的 Amazon Redshift 预调配集群或 Amazon Redshift Serverless 资源，用户必须具备针对相应客户托管密钥调用以下操作的权限：

- `kms:CreateGrant`
- `kms:GenerateDataKey`
- `kms:Encrypt`
- `kms:Decrypt`
- `kms:DescribeKey`

客户托管密钥的所有者或管理员可以在客户托管密钥的密钥策略中为特定用户添加这些权限；若密钥策略允许，也可在 IAM 策略中配置。

Amazon Redshift 通过 AWS KMS 授权访问客户托管密钥。请确保您的密钥策略不会阻止 Amazon Redshift 创建或使用对密钥的授权。例如，显式 Deny 语句，或 Allow 语句中的限制性条件，都可能会阻止授权的创建并导致操作失败。

作为一项安全最佳实践，我们建议您仅授予使用案例所需的权限。例如，您可以使用 `kms:GrantIsForAWSResource` 这类条件键来缩小 `kms:CreateGrant` 权限的适用范围。

有关控制 AWS KMS 密钥访问权限的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的 AWS KMS 中的密钥策略和允许其他账户中的用户使用 KMS 密钥。

## 数据令牌化

令牌化是为了数据安全目的将实际值替换为不透明值的过程。安全敏感型应用程序使用令牌化来替换敏感数据，如个人身份信息 (PII) 或受保护健康信息 (PHI)，以降低安全风险。取消令牌化使用适当的安全策略为授权用户使用实际值来反转令牌。

为了与第三方令牌化服务集成，您可以使用 Amazon Redshift 用户定义的函数 (UDF)，这些函数使用 AWS Lambda 创建。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Lambda 用户定义的函数。例如，请参阅 Protegrity。

Amazon Redshift 将令牌化请求发送到通过 REST API 或预定义端点访问的令牌化服务器。两个或多个互补的 Lambda 函数处理令牌化和取消令牌化请求。对于此处理，您可以使用第三方令牌化提供程序提供的 Lambda 函数。您还可以使用在 Amazon Redshift 中注册为 Lambda UDF 的 Lambda 函数。

例如，假设提交的查询在列上调用令牌化或取消令牌化 UDF。Amazon Redshift 集群可以后台处理适用的参数行，并将这些行分批并行发送到 Lambda 函数。Amazon Redshift 计算节点和 Lambda 之间的数据传输是在客户端无法访问的独立隔离网络连接中进行的。Lambda 函数将数据传递到令牌化服务器端点。令牌化服务器根据需要对数据进行令牌化或取消令牌化，并返回数据。然后，Lambda 函数将结果传输到 Amazon Redshift 集群进行进一步处理（如有必要），然后返回查询结果。

## 在 Amazon Redshift 中路由互联网络流量

您可以通过 Amazon Redshift 中的已知私有网络路由来路由流量。本页介绍如何在企业网络上以及同一 AWS 区域的资源之间路由流量。

要在 Amazon Redshift 与公司网络上的客户端和应用程序之间路由流量，请执行以下操作：

- 在 Virtual Private Cloud (VPC) 和公司网络之间建立私有连接。通过互联网设置 IPsec VPN 连接或使用 Direct Connect 连接设置专用物理连接。Direct Connect 使您可以在本地网络与 Amazon VPC 之间直接建立一个专用虚拟接口，从而在您的网络和 VPC 之间提供一个专用的高带宽网络连接。

借助多个虚拟接口，您甚至可以在保持网络隔离性的同时，与多个 VPC 建立专用连接。有关更多信息，请参阅什么是 AWS Site-to-Site VPN？和什么是 Direct Connect？

要在 VPC 中的 Amazon Redshift 集群与相同 AWS 区域中的 Amazon S3 桶之间路由流量，请执行以下操作：

- 设置 Amazon S3 私有 VPC 端点以从 ETL 负载或分载私密访问 Amazon S3 数据。有关更多信息，请参阅用于 Amazon S3 的端点。
- 为 Amazon Redshift 集群启用“增强型 VPC 路由”，指定目标 Amazon S3 VPC 端点。Amazon Redshift COPY、UNLOAD 或 CREATE LIBRARY 命令生成的流量随后路由通过私有端点。有关更多信息，请参阅 启用增强型 VPC 路由。

# Amazon Redshift 中的 Identity and Access Management

访问 Amazon Redshift 时需要可供 AWS 用来验证您的请求的凭证。这些凭证必须有权访问 AWS 资源，如 Amazon Redshift 集群。下面几节提供详细信息来说明如何使用 AWS Identity and Access Management (IAM) 和 Amazon Redshift 控制谁能访问您的资源，从而对这些资源进行保护：

- 使用身份进行身份验证
- 访问控制

> **Important**
>
> 本主题包含有关管理权限、身份和安全访问的一系列最佳实践。我们建议您熟悉掌握将 IAM 与 Amazon Redshift 结合使用的最佳实践。这些最佳实践包括使用 IAM 角色应用权限。充分了解这些部分有助于您维护更安全的 Amazon Redshift 数据仓库。

## 使用身份进行身份验证

身份验证是您使用身份凭证登录 AWS 的方法。您必须作为 AWS 账户根用户、IAM 用户或通过担任 IAM 角色进行身份验证。

您可以使用来自 AWS IAM Identity Center（IAM Identity Center）等身份源的凭证、单点登录身份验证或 Google/Facebook 凭证，以联合身份进行登录。有关登录的更多信息，请参阅《AWS 登录 用户指南》中的如何登录您的 AWS 账户。

对于编程访问，AWS 提供了 SDK 和 CLI 来对请求进行加密签名。有关更多信息，请参阅《IAM 用户指南》中的适用于 API 请求的 AWS 签名版本 4。

### AWS 账户 根用户

当您创建 AWS 账户 时，最初使用的是一个对所有 AWS 服务和资源拥有完全访问权限的登录身份（称为 AWS 账户根用户）。我们强烈建议不要使用根用户进行日常任务。有关要求根用户凭证的任务，请参阅《IAM 用户指南》中的需要根用户凭证的任务。

### IAM 用户和群组

IAM 用户是对某个人员或应用程序具有特定权限的一个身份。建议使用临时凭证，而非具有长期凭证的 IAM 用户。有关更多信息，请参阅《IAM 用户指南》中的要求人类用户使用带有身份提供商的联合身份验证才能使用临时凭证访问 AWS。

IAM 组指定一组 IAM 用户，便于更轻松地对大量用户进行权限管理。有关更多信息，请参阅《IAM 用户指南》中的 IAM 用户使用案例。

### IAM 角色

IAM 角色是具有特定权限的身份，可提供临时凭证。您可以通过从用户切换到 IAM 角色（控制台）或调用 AWS CLI 或 AWS API 操作来担任角色。有关更多信息，请参阅《IAM 用户指南》中的担任角色的方法。

IAM 角色对于联合用户访问、临时 IAM 用户权限、跨账户访问、跨服务访问以及在 Amazon EC2 上运行的应用程序非常有用。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

> **Note**
>
> Redshift 中的转发访问会话（FAS，Forward Access Session）的有效期仅为 12 小时。在此期限之后，使用 FAS 与其他服务集成的任何连接会话都必须重建。

## 防止跨服务混淆座席

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在 AWS 中，跨服务模拟可能会导致混淆代理问题。一个服务（呼叫服务）调用另一项服务（所谓的服务）时，可能会发生跨服务模拟。可以操纵调用服务，使用其权限以在其他情况下该服务不应有访问权限的方式对另一个客户的资源进行操作。为防止这种情况，AWS 提供可帮助您保护所有服务的

数据的工具，而这些服务中的服务主体有权限访问账户中的资源。有关更多信息，请参阅《IAM 用户指南》中的防止跨服务混淆代理。

我们建议在基于资源的策略中使用 aws:SourceArn 和 aws:SourceAccount 全局条件上下文键，以限制 Amazon Redshift 为其它服务提供的资源访问权限。如果 aws:SourceArn 值不包含账户 ID，例如 Amazon S3 存储桶 ARN，您必须使用两个全局条件上下文密钥来限制权限。如果同时使用全局条件上下文密钥和包含账户 ID 的 aws:SourceArn 值，则 aws:SourceAccount 值和 aws:SourceArn 值中的账户在同一策略语句中使用时，必须使用相同的账户 ID。

以下示例显示了一个策略，您可以应用该策略来限制 Amazon Redshift 出现混淆代理问题。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionForRedshift",
    "Effect": "Allow",
    "Principal": {
      "Service": "redshift.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:redshift:us-east-1:123456789012:cluster:my-cluster"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

以下示例显示 Amazon Redshift Serverless 的策略。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionForRedshiftServerless",
    "Effect": "Allow",
    "Principal": {
      "Service": "redshift-serverless.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
```

```
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:redshift-serverless:us-east-1:123456789012:workgroup/
my-workgroup"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

## 访问控制

您可以使用有效的凭证来对自己的请求进行身份验证，但您还必须拥有权限才能创建或访问 Amazon Redshift 资源。例如，您必须拥有权限才能创建 Amazon Redshift 集群、创建快照、添加事件订阅等。

下面几节介绍如何管理 Amazon Redshift 的权限。我们建议您先阅读概述。

- 管理 Amazon Redshift 资源的访问权限的概览
- 将基于身份的策略（IAM 策略）用于 Amazon Redshift

## 管理 Amazon Redshift 资源的访问权限的概览

每个 AWS 资源都归某个 AWS 账户所有，创建或访问这些资源的权限由权限策略控制。账户管理员可以向 IAM 身份（即：用户、组和角色）附加权限策略，某些服务（如 AWS Lambda）也支持向资源附加权限策略。

> **Note**
>
> 账户管理员（或管理员用户）是具有管理员权限的用户。有关更多信息，请参阅 IAM 用户指南中的 IAM 最佳实践。

在授予权限时，您要决定谁获得权限，获得对哪些资源的权限，以及您允许对这些资源执行的具体操作。

## Amazon Redshift 资源和操作

Amazon Redshift 提供服务特定的资源、操作和条件上下文键以在 IAM 权限策略中使用。

Amazon Redshift、Amazon Redshift Serverless、Amazon Redshift 数据 API 和Amazon Redshift 查询编辑器 v2 访问权限

在设置 访问控制 时，您编写可附加到 IAM 身份的权限策略（基于身份的策略）。有关详细参考信息，请参阅《服务授权参考》中的以下主题：

- 对于 Amazon Redshift，请参阅 Amazon Redshift 的操作、资源和条件键，它们使用 `redshift:` 前缀。
- 对于 Amazon Redshift Serverless，请参阅 Amazon Redshift Serverless 的操作、资源和条件键，它们使用 `redshift-serverless:` 前缀。
- 对于 Amazon Redshift 数据 API，请参阅 Amazon Redshift 数据 API 的操作、资源和条件键，它们使用 `redshift-data:` 前缀。
- 对于 Amazon Redshift 查询编辑器 v2，请参阅 AWS SQL Workbench（Amazon Redshift 查询编辑器 v2）的操作、资源和条件键，它们使用 `sqlworkbench:` 前缀。

  查询编辑器 v2 包括仅限权限操作，这些操作不会直接响应 API 操作。《Service Authorization Reference》中用 `[permission only]` 指明了这些操作。

《服务授权参考》包含有关可在 IAM 策略中使用的 API 操作的信息。它还包括您可以授予权限的 AWS 资源以及您可以针对细粒度访问控制包含的条件键。有关条件的更多信息，请参阅“使用 IAM 策略条件进行精细访问控制”。

您需要在策略的 `Action` 字段中指定操作、在策略的 `Resource` 字段中指定资源值、在策略的 `Condition` 字段中指定条件。要为 Amazon Redshift 指定操作，请在 API 操作名称之前使用 `redshift:` 前缀（例如 `redshift:CreateCluster`）。

## 了解资源所有权

资源所有者是创建资源的 AWS 账户。也就是说，资源拥有者是 主体实体（根账户、IAM 用户或 IAM 角色）的 AWS 账户，该账户会对创建资源的请求进行身份验证。以下示例说明了它的工作原理：

- 如果您使用 AWS 账户的根账户凭证创建数据库集群，则您的 AWS 账户即为该 Amazon Redshift 资源的拥有者。

- 如果您在您的 AWS 账户中创建具有创建 Amazon Redshift 资源的权限的 IAM 角色，则能够担任该角色的任何人都可以创建 Amazon Redshift 资源。该角色所属的您的 AWS 账户拥有这些 Amazon Redshift 资源。
- 如果您在 AWS 账户中创建 IAM 用户并对该用户授予创建 Amazon Redshift 资源的权限，则该用户也可以创建 Amazon Redshift 资源。但是，该用户所属的 AWS 账户拥有这些 Amazon Redshift 资源。在大多数情况下，不建议使用此方法。我们建议创建 IAM 角色并向该角色附加权限，然后将该角色分配给用户。

## 管理对资源的访问

权限策略规定谁可以访问哪些内容。下一节介绍创建权限策略时的可用选项。

> **Note**
>
> 本节讨论如何在 Amazon Redshift 范围内使用 IAM。这里不提供有关 IAM 服务的详细信息。有关完整的 IAM 文档，请参阅 IAM 用户指南中的什么是 IAM？。有关 IAM 策略语法和说明的信息，请参阅 IAM 用户指南中的 AWS IAM 策略参考。

附加到 IAM 身份的策略称作基于身份 的策略（IAM 策略），附加到资源的策略称作基于资源 的策略。Amazon Redshift 只支持基于身份的策略（IAM 策略）。

基于身份的策略（IAM 策略）

您可以通过将策略附加到 IAM 角色，然后将该角色分配给用户或组来分配权限。下面的示例策略包含为您的 AWS 账户创建、删除、修改和重启 Amazon Redshift 集群的权限。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"AllowManageClusters",
      "Effect":"Allow",
      "Action": [
        "redshift:CreateCluster",
        "redshift:DeleteCluster",
```

```
        "redshift:ModifyCluster",
        "redshift:RebootCluster"
      ],
      "Resource":"*"
    }
  ]
}
```

有关将基于身份的策略与 Amazon Redshift 结合使用的更多信息，请参阅将基于身份的策略（IAM 策略）用于 Amazon Redshift。有关用户、组、角色和权限的更多信息，请参阅 IAM 用户指南中的身份（用户、组和角色）。

基于资源的策略

其他服务（如 Amazon S3）还支持基于资源的权限策略。例如，您可以将策略附加到 S3 桶以管理对该桶的访问权限。Amazon Redshift 不支持基于资源的策略。

## 指定策略元素：操作、效果、资源和主体

对于每个 Amazon Redshift 资源（请参阅 Amazon Redshift 资源和操作），该服务都定义了一组 API 操作（请参阅操作）。为授予这些 API 操作的权限，Amazon Redshift 定义了一组您可以在策略中指定的操作。执行一个 API 操作可能需要多个操作的权限。

以下是基本的策略元素：

- 资源 - 在策略中，您可以使用 Amazon 资源名称（ARN）标识策略应用到的资源。有关更多信息，请参阅 Amazon Redshift 资源和操作。
- 操作 – 您可以使用操作关键字标识要允许或拒绝的资源操作。例如，`redshift:DescribeClusters` 权限允许执行 Amazon Redshift `DescribeClusters` 操作的用户权限。
- 效果 — 您可以指定当用户请求特定操作（可以是允许或拒绝）时的效果。如果没有显式授予（允许）对资源的访问权限，则隐式拒绝访问。您也可显式拒绝对资源的访问，这样可确保用户无法访问该资源，即使有其他策略授予了访问权限的情况下也是如此。
- 主体：在基于身份的策略（IAM 策略）中，附加了策略的用户是隐式主体。对于基于资源的策略，您可以指定要接收权限的用户、账户、服务或其他实体 (仅适用于基于资源的策略)。Amazon Redshift 不支持基于资源的策略。

有关 IAM 策略语法和描述的更多信息，请参阅 IAM 用户指南中的 AWS IAM 策略参考。

有关显示所有 Amazon Redshift API 操作及其适用的资源的表，请参阅Amazon Redshift、Amazon Redshift Serverless、Amazon Redshift 数据 API 和Amazon Redshift 查询编辑器 v2 访问权限。

## 在策略中指定条件

当您授予权限时，可使用访问策略语言来指定规定策略何时生效的条件。例如，您可能希望策略仅在特定日期后应用。有关使用访问策略语言指定条件的更多信息，请参阅 IAM 用户指南中的 IAM JSON 策略元素：条件。

要确定权限策略适用的条件，请在 IAM 权限策略中包含元素 `Condition`。例如，您可以创建一个策略，允许用户使用 `redshift:CreateCluster` 操作来创建集群，也可以添加 `Condition` 元素来限制用户只能在特定的区域中创建集群。有关详细信息，请参阅使用 IAM 策略条件进行精细访问控制。有关所有条件键值及其适用的 Amazon Redshift 操作和资源的列表，请参阅Amazon Redshift、Amazon Redshift Serverless、Amazon Redshift 数据 API 和Amazon Redshift 查询编辑器 v2 访问权限。

使用 IAM 策略条件进行精细访问控制

在 Amazon Redshift 中，您可以根据资源的标签使用条件键来限制对资源的访问。以下是常见的 Amazon Redshift 条件键。

| 条件键 | 描述 |
| --- | --- |
| `aws:RequestTag` | 要求用户在创建资源时添加标签密钥（名称）和值。有关更多信息，请参阅 IAM 用户指南中的 aws:RequestTag。 |
| `aws:ResourceTag` | 根据特定标签密钥和值限制用户对资源的访问。有关更多信息，请参阅 IAM 用户指南中的 aws:ResourceTag。 |
| `aws:TagKeys` | 使用此键可将请求中的标签键与您在策略中指定的键进行比较。有关更多信息，请参阅 IAM 用户指南中的 aws:TagKeys。 |

有关标签的信息，请参阅在 Amazon Redshift 中为资源添加标签。

有关支持 `redshift:RequestTag` 和 `redshift:ResourceTag` 条件键的 API 操作的列表，请参阅Amazon Redshift、Amazon Redshift Serverless、Amazon Redshift 数据 API 和Amazon Redshift 查询编辑器 v2 访问权限。

以下条件键可用于 Amazon Redshift GetClusterCredentials 操作。

| 条件键 | 描述 |
| --- | --- |
| redshift:DurationSeconds | 限制可为持续时间指定的秒数。 |
| redshift:DbName | 限制可指定的数据库名称。 |
| redshift:DbUser | 限制可指定的数据库用户名。 |

示例 1：使用 aws:ResourceTag 条件键限制访问

使用以下 IAM 策略，仅允许用户针对位于 us-west-2 区域的特定 AWS 账户修改标签名为 environment、值为 test 的 Amazon Redshift 集群。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Sid":"AllowModifyTestCluster",
        "Effect": "Allow",
        "Action": "redshift:ModifyCluster",
        "Resource": "arn:aws:redshift:us-west-2:123456789012:cluster:*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/environment": "test"
            }
        }
    }
}
```

示例 2：使用 aws:RequestTag 条件键限制访问

使用以下 IAM 策略，仅允许用户在创建集群的命令包含名为 usage、值为 production 的标签时创建 Amazon Redshift 集群。带有 aws:TagKeys 和 ForAllValues 修饰符的条件指定只能在请求中指定 costcenter 和 usage 键。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Sid":"AllowCreateProductionCluster",
        "Effect": "Allow",
        "Action": [
            "redshift:CreateCluster",
            "redshift:CreateTags"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/usage": "production"
            },
            "ForAllValues:StringEquals": {
                "aws:TagKeys": [
                    "costcenter",
                    "usage"
                ]
            }
        }
    }
}
```

## 将基于身份的策略（IAM 策略）用于 Amazon Redshift

本主题提供了基于身份的策略的示例，在这些策略中，账户管理员可以向 IAM 身份（即：用户、组和角色）附加权限策略。

> **Important**
>
> 我们建议您首先阅读以下介绍性主题，这些主题讲解了管理 Amazon Redshift 资源访问的基本概念和选项。有关更多信息，请参阅 管理 Amazon Redshift 资源的访问权限的概览。

下面介绍权限策略示例。该权限允许用户创建、删除、修改和重启所有集群，但拒绝删除或修改 AWS 区域 `us-west-2` 和 AWS 账户 `123456789012` 中集群标识符以 `production` 开头的任何集群的权限。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"AllowClusterManagement",
      "Action": [
        "redshift:CreateCluster",
        "redshift:DeleteCluster",
        "redshift:ModifyCluster",
        "redshift:RebootCluster"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Sid":"DenyDeleteModifyProtected",
      "Action": [
        "redshift:DeleteCluster",
        "redshift:ModifyCluster"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:cluster:production*"
      ],
      "Effect": "Deny"
    }
  ]
}
```

该策略包含两条语句：

- 第一条语句授予用户创建、删除、修改和重启集群的权限。该语句指定通配符 (*) 作为 Resource 的值，因此，该策略适用于归根 AWS 账户所有的一切 Amazon Redshift 资源。
- 第二条语句拒绝删除或修改集群的权限。该语句指定包含通配符 (*) 的集群 Amazon 资源名称（ARN）作为 Resource 的值。因此，该语句适用于归根 AWS 账户所有的一切 Amazon Redshift 集群（集群标识符以 production 开头）。

## AWS适用于 Amazon Redshift 的 托管式策略

AWS通过提供由创建和管理的独立 IAM 策略来满足许多常用案例的要求。AWS托管策略可针对常见使用案例授予必要权限，因此，您无需自行调查具体需要哪些权限。有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管式策略。

此外，您还可以创建自定义 IAM 策略，以授予账户 Amazon Redshift API 操作和资源访问的权限。您可以将这些自定义策略附加到需要这些权限的 IAM 角色或组。

以下部分描述 Amazon Redshift 特有的 AWS 托管式策略（可附加至您账户中的用户）：

## AWS 托管式策略的 Amazon Redshift 更新

查看有关 Amazon Redshift（自从其开始跟踪更新更改以来）的 AWS 托管式策略更新的详细信息。有关此页面更改的自动提示，请订阅 Amazon Redshift 文档历史记录页面上的 RSS 源。

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftFederatedAuthorization - 新策略 | Amazon Redshift 新增了一项易用性策略，用于通过 Amazon Redshift 联合身份验证授权功能执行查询。 | 2025 年 11 月 21 日 |
| AmazonRedshiftServiceLinkedRolePolicy：对现有策略的更新 | 托管式策略增加了 `lakeformation:GetDataAccess` 操作的权限。添加此权限可授予从 AWS Lake Formation 中获取联合目录信息的权限。<br><br>操作 `glue:GetCatalog` 和 `glue:GetCatalogs` 的其它条件将添加到托管式策略中。 | 2025 年 3 月 13 日 |
| AmazonRedshiftServiceLinkedRolePolicy：对现有策略的更新 | 操作 `glue:GetCatalog` 和 `glue:GetCatalogs` 的权限将添加到托管式策略中。添加这些权限可授予从 AWS Glue 中获取目录信息的权限。 | 2024 年 12 月 3 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftServiceLinkedRolePolicy：对现有策略的更新 | 托管式策略增加了 `servicequotas:GetServiceQuota` 操作的权限。这为访问配额或限制提供了权限。 | 2024 年 3 月 8 日 |
| AmazonRedshiftQueryEditorV2FullAccess：对现有策略的更新 | 操作 `redshift-serverless:ListNamespaces` 和 `redshift-serverless:ListWorkgroups` 的权限将添加到托管式策略中。添加它们可授予在 Amazon Redshift 数据仓库中列出无服务器命名空间和无服务器工作组的权限。 | 2024 年 2 月 21 日 |
| AmazonRedshiftQueryEditorV2NoSharing：对现有策略的更新 | 操作 `redshift-serverless:ListNamespaces` 和 `redshift-serverless:ListWorkgroups` 的权限将添加到托管式策略中。添加它们可授予在 Amazon Redshift 数据仓库中列出无服务器命名空间和无服务器工作组的权限。 | 2024 年 2 月 21 日 |
| AmazonRedshiftQueryEditorV2ReadSharing：对现有策略的更新 | 操作 `redshift-serverless:ListNamespaces` 和 `redshift-serverless:ListWorkgroups` 的权限将添加到托管式策略中。添加它们可授予在 Amazon Redshift 数据仓库中列出无服务器命名空间和无服务器工作组的权限。 | 2024 年 2 月 21 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftQueryEditorV2ReadWriteSharing：对现有策略的更新 | 操作 `redshift-serverless:ListNamespaces` 和 `redshift-serverless:ListWorkgroups` 的权限将添加到托管式策略中。添加它们可授予在 Amazon Redshift 数据仓库中列出无服务器命名空间和无服务器工作组的权限。 | 2024 年 2 月 21 日 |
| AmazonRedshiftReadOnlyAccess：对现有策略的更新 | 托管式策略增加了 `redshift:ListRecommendations` 操作的权限。这将授予列出 Amazon Redshift Advisor 建议的权限。 | 2024 年 2 月 7 日 |
| AmazonRedshiftServiceLinkedRolePolicy：对现有策略的更新 | 操作 `ec2:AssignIpv6Addresses` 和 `ec2:UnassignIpv6Addresses` 的权限将添加到托管式策略中。添加这些权限可授予分配和取消分配 IP 地址的权限。 | 2023 年 10 月 31 日 |
| AmazonRedshiftQueryEditorV2NoSharing：对现有策略的更新 | 操作 `sqlworkbench:GetAutocompletionMetadata` 和 `sqlworkbench:GetAutocompletionResource` 的权限将添加到托管式策略中。添加它们可授予生成和检索数据库信息的权限，以便在编辑查询时自动完成 SQL。 | 2023 年 8 月 16 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftQueryEditorV2ReadSharing：对现有策略的更新 | 操作 `sqlworkbench:GetAutocompletionMetadata` 和 `sqlworkbench:GetAutocompletionResource` 的权限将添加到托管式策略中。添加它们可授予生成和检索数据库信息的权限，以便在编辑查询时自动完成 SQL。 | 2023 年 8 月 16 日 |
| AmazonRedshiftQueryEditorV2ReadWriteSharing：对现有策略的更新 | 操作 `sqlworkbench:GetAutocompletionMetadata` 和 `sqlworkbench:GetAutocompletionResource` 的权限将添加到托管式策略中。添加它们可授予生成和检索数据库信息的权限，以便在编辑查询时自动完成 SQL。 | 2023 年 8 月 16 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftServiceLinkedRolePolicy：对现有策略的更新 | 在托管式策略中增加了在 AWS Secrets Manager 中执行密钥创建和管理操作的权限。添加的权限如下：<br>• `secretsmanager:GetRandomPassword`<br>• `secretsmanager:DescribeSecret`<br>• `secretsmanager:PutSecretValue`<br>• `secretsmanager:UpdateSecret`<br>• `secretsmanager:UpdateSecretVersionStage`<br>• `secretsmanager:RotateSecret`<br>• `secretsmanager:DeleteSecret` | 2023 年 8 月 14 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftServiceLinkedRolePolicy：对现有策略的更新 | 现在已从托管式策略删除了在 Amazon EC2 上执行操作以创建和管理安全组和路由规则的权限。这些权限与创建子网和 VPC 有关。移除的权限如下：<br>• ec2:AuthorizeSecurityGroupEgress<br>• ec2:AuthorizeSecurityGroupIngress<br>• ec2:UpdateSecurityGroupRuleDescriptionsEgress<br>• ec2:ReplaceRouteTableAssociation<br>• ec2:CreateRouteTable<br>• ec2:AttachInternetGateway<br>• ec2:UpdateSecurityGroupRuleDescriptionsIngress<br>• ec2:AssociateRouteTable<br>• ec2:RevokeSecurityGroupIngress<br>• ec2:CreateRoute<br>• ec2:CreateSecurityGroup<br>• ec2:RevokeSecurityGroupEgress | 2023 年 5 月 8 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| | • `ec2:ModifyVpcAttribute`<br>• `ec2:CreateSubnet`<br>• `ec2:CreateInternetGateway`<br>• `ec2:CreateVpc`<br><br>它们与 Purpose:RedshiftMigrateToVpc 资源标签有关联。该标签将权限范围限制为从 Amazon EC2 Classic 到 Amazon EC2 VPC 迁移任务。有关更多信息，请参阅控制对使用标签的 AWS 资源进行访问。 | |
| AmazonRedshiftDataFullAccess：对现有策略的更新 | 托管式策略增加了 `redshift:GetClusterCredentialsWithIAM` 操作的权限。增加此权限会授予获取增强型临时凭证以通过指定的 AWS 账户访问 Amazon Redshift 数据库的权限。 | 2023 年 4 月 7 日 |
| AmazonRedshiftServiceLinkedRolePolicy：对现有策略的更新 | 在托管式策略中，现在添加了在 Amazon EC2 上执行操作以创建和管理安全组规则的权限。这些安全组和规则特别关联到 Amazon Redshift `aws:RequestTag/Redshift` 资源标签。这样可以将权限的范围限制为特定 Amazon Redshift 资源。 | 2023 年 4 月 6 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftQueryEditorV2NoSharing：对现有策略的更新 | 托管式策略增加了 `sqlworkbench:GetSchemaInference` 操作的权限。添加此项会授予权限，以获取从文件推断的列和数据类型 | 2023 年 3 月 21 日 |
| AmazonRedshiftQueryEditorV2ReadSharing：对现有策略的更新 | 托管式策略增加了 `sqlworkbench:GetSchemaInference` 操作的权限。添加此项会授予权限，以获取从文件推断的列和数据类型 | 2023 年 3 月 21 日 |
| AmazonRedshiftQueryEditorV2ReadWriteSharing：对现有策略的更新 | 托管式策略增加了 `sqlworkbench:GetSchemaInference` 操作的权限。添加此项会授予权限，以获取从文件推断的列和数据类型 | 2023 年 3 月 21 日 |
| AmazonRedshiftQueryEditorV2NoSharing：对现有策略的更新 | 托管式策略增加了 `sqlworkbench:AssociateNotebookWithTab` 操作的权限。添加它会授予相关权限，以创建和更新链接到用户自己的笔记本的选项卡。 | 2023 年 2 月 2 日 |
| AmazonRedshiftQueryEditorV2ReadSharing：对现有策略的更新 | 托管式策略增加了 `sqlworkbench:AssociateNotebookWithTab` 操作的权限。添加它会授予相关权限，以创建和更新链接到用户自己的笔记本或链接到与其共享的笔记本的选项卡。 | 2023 年 2 月 2 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftQueryEditorV2ReadWriteSharing：对现有策略的更新 | 托管式策略增加了 `sqlworkbench:AssociateNotebookWithTab` 操作的权限。添加它会授予相关权限，以创建和更新链接到用户自己的笔记本或链接到与其共享的笔记本的选项卡。 | 2023 年 2 月 2 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftQueryEditorV2NoSharing：对现有策略的更新 | 为了授予使用笔记本的权限，Amazon Redshift 添加了以下操作的权限：<br>• sqlworkbench:ListNotebooks<br>• sqlworkbench:CreateNotebook<br>• sqlworkbench:DuplicateNotebook<br>• sqlworkbench:CreateNotebookFromVersion<br>• sqlworkbench:ImportNotebook<br>• sqlworkbench:GetNotebook<br>• sqlworkbench:UpdateNotebook<br>• sqlworkbench:DeleteNotebook<br>• sqlworkbench:CreateNotebookCell<br>• sqlworkbench:DeleteNotebookCell<br>• sqlworkbench:UpdateNotebookCellContent<br>• sqlworkbench:UpdateNotebookCellLayout | 2022 年 10 月 17 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| | • sqlworkbench:BatchGetNotebookCell<br>• sqlworkbench:ListNotebookVersions<br>• sqlworkbench:CreateNotebookVersion<br>• sqlworkbench:GetNotebookVersion<br>• sqlworkbench:DeleteNotebookVersion<br>• sqlworkbench:RestoreNotebookVersion<br>• sqlworkbench:ExportNotebook | |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftQueryEditorV2ReadSharing：对现有策略的更新 | 为了授予使用笔记本的权限，Amazon Redshift 添加了以下操作的权限：<br>• sqlworkbench:ListNotebooks<br>• sqlworkbench:CreateNotebook<br>• sqlworkbench:DuplicateNotebook<br>• sqlworkbench:CreateNotebookFromVersion<br>• sqlworkbench:ImportNotebook<br>• sqlworkbench:GetNotebook<br>• sqlworkbench:UpdateNotebook<br>• sqlworkbench:DeleteNotebook<br>• sqlworkbench:CreateNotebookCell<br>• sqlworkbench:DeleteNotebookCell<br>• sqlworkbench:UpdateNotebookCellContent<br>• sqlworkbench:UpdateNotebookCellLayout | 2022 年 10 月 17 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| | • sqlworkbench:BatchGetNotebookCell<br>• sqlworkbench:ListNotebookVersions<br>• sqlworkbench:CreateNotebookVersion<br>• sqlworkbench:GetNotebookVersion<br>• sqlworkbench:DeleteNotebookVersion<br>• sqlworkbench:RestoreNotebookVersion<br>• sqlworkbench:ExportNotebook | |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftQueryEditorV2ReadWriteSharing：对现有策略的更新 | 为了授予使用笔记本的权限，Amazon Redshift 添加了以下操作的权限：<br>• sqlworkbench:ListNotebooks<br>• sqlworkbench:CreateNotebook<br>• sqlworkbench:DuplicateNotebook<br>• sqlworkbench:CreateNotebookFromVersion<br>• sqlworkbench:ImportNotebook<br>• sqlworkbench:GetNotebook<br>• sqlworkbench:UpdateNotebook<br>• sqlworkbench:DeleteNotebook<br>• sqlworkbench:CreateNotebookCell<br>• sqlworkbench:DeleteNotebookCell<br>• sqlworkbench:UpdateNotebookCellContent<br>• sqlworkbench:UpdateNotebookCellLayout | 2022 年 10 月 17 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| | • sqlworkbench:BatchGetNotebookCell<br>• sqlworkbench:ListNotebookVersions<br>• sqlworkbench:CreateNotebookVersion<br>• sqlworkbench:GetNotebookVersion<br>• sqlworkbench:DeleteNotebookVersion<br>• sqlworkbench:RestoreNotebookVersion<br>• sqlworkbench:ExportNotebook | |
| AmazonRedshiftServiceLinkedRolePolicy：对现有策略的更新 | Amazon Redshift 添加了命名空间 AWS/Redshift 以允许向 CloudWatch 发布指标。 | 2022 年 9 月 7 日 |
| AmazonRedshiftQueryEditorV2NoSharing：对现有策略的更新 | Amazon Redshift 添加了用于执行 sqlworkbench:ListQueryExecutionHistory 和 sqlworkbench:GetQueryExecutionHistory 操作的权限。这授予了查看查询历史记录的权限。 | 2022 年 8 月 30 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftQueryEditorV2ReadSharing：对现有策略的更新 | Amazon Redshift 添加了用于执行 sqlworkbench:ListQueryExecutionHistory 和 sqlworkbench:GetQueryExecutionHistory 操作的权限。这授予了查看查询历史记录的权限。 | 2022 年 8 月 30 日 |
| AmazonRedshiftQueryEditorV2ReadWriteSharing：对现有策略的更新 | Amazon Redshift 添加了用于执行 sqlworkbench:ListQueryExecutionHistory 和 sqlworkbench:GetQueryExecutionHistory 操作的权限。这授予了查看查询历史记录的权限。 | 2022 年 8 月 30 日 |
| AmazonRedshiftFullAccess：对现有策略的更新 | Amazon Redshift Serverless 的权限已添加到现有的 AmazonRedshiftFullAccess 托管式策略中。 | 2022 年 7 月 22 日 |
| AmazonRedshiftDataFullAccess：对现有策略的更新 | Amazon Redshift 已将标签 aws:ResourceTag/RedshiftDataFullAccess 权限的 redshift-serverless:GetCredentials 默认作用域条件从 StringEquals 更新为 StringLike ，以授予对使用标签键 RedshiftDataFullAccess 和任何标签值标记的资源的访问权限。 | 2022 年 7 月 11 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftDataFullAccess：对现有策略的更新 | Amazon Redshift 添加了新权限，以允许 `redshift-serverless:GetCredentials` 获得对 Amazon Redshift Serverless 的临时凭证。 | 2022 年 7 月 8 日 |
| AmazonRedshiftQueryEditorV2NoSharing：对现有策略的更新 | Amazon Redshift 添加了用于执行 `sqlworkbench:GetAccountSettings` 操作的权限。这将授予获取账户设置的权限。 | 2022 年 6 月 15 日 |
| AmazonRedshiftQueryEditorV2ReadSharing：对现有策略的更新 | Amazon Redshift 添加了用于执行 `sqlworkbench:GetAccountSettings` 操作的权限。这将授予获取账户设置的权限。 | 2022 年 6 月 15 日 |
| AmazonRedshiftQueryEditorV2ReadWriteSharing：对现有策略的更新 | Amazon Redshift 添加了用于执行 `sqlworkbench:GetAccountSettings` 操作的权限。这将授予获取账户设置的权限。 | 2022 年 6 月 15 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftServiceLinkedRolePolicy：对现有策略的更新 | 为了允许对新的 Amazon Redshift Serverless 端点进行公有访问，Amazon Redshift 在客户账户中分配弹性 IP 地址并将其关联到 VPC 端点的弹性网络接口。它通过由服务相关角色提供的权限来完成此操作。要启用此使用案例，分配和释放弹性 IP 地址的操作将添加到 Amazon Redshift Serverless 服务相关角色中。 | 2022 年 5 月 26 日 |
| AmazonRedshiftQueryEditorV2FullAccess：对现有策略的更新 | 操作 `sqlworkbench:ListTaggedResources` 的权限。它专门适用于 Amazon Redshift 查询编辑器 v2 资源。此策略更新授予仅可通过查询编辑器 v2 调用 `tag:GetResources` 的权限。 | 2022 年 2 月 22 日 |
| AmazonRedshiftQueryEditorV2NoSharing：对现有策略的更新 | 操作 `sqlworkbench:ListTaggedResources` 的权限。它专门适用于 Amazon Redshift 查询编辑器 v2 资源。此策略更新授予仅可通过查询编辑器 v2 调用 `tag:GetResources` 的权限。 | 2022 年 2 月 22 日 |
| AmazonRedshiftQueryEditorV2ReadSharing：对现有策略的更新 | 操作 `sqlworkbench:ListTaggedResources` 的权限。它专门适用于 Amazon Redshift 查询编辑器 v2 资源。此策略更新授予仅可通过查询编辑器 v2 调用 `tag:GetResources` 的权限。 | 2022 年 2 月 22 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftQueryEditorV2ReadWriteSharing：对现有策略的更新 | 操作 `sqlworkbench:ListTaggedResources` 的权限。它专门适用于 Amazon Redshift 查询编辑器 v2 资源。此策略更新授予仅可通过查询编辑器 v2 调用 `tag:GetResources` 的权限。 | 2022 年 2 月 22 日 |
| AmazonRedshiftQueryEditorV2ReadSharing：对现有策略的更新 | 托管式策略增加了 `sqlworkbench:AssociateQueryWithTab` 操作的权限。增加此权限将允许客户创建编辑器选项卡，以链接到与其共享的查询。 | 2022 年 2 月 22 日 |
| AmazonRedshiftServiceLinkedRolePolicy – 对现有策略的更新 | Amazon Redshift 添加了新操作的权限，允许对 Amazon Redshift 网络和 VPC 资源进行管理。 | 2021 年 11 月 22 日 |
| AmazonRedshiftAllCommandsFullAccess – 新策略 | Amazon Redshift 添加了一项新策略，允许使用从 Amazon Redshift 控制台创建的 IAM 角色，并将其设置为默认角色，以便集群从 Amazon S3 运行 COPY、UNLOAD、CREATE EXTERNAL SCHEMA、CREATE EXTERNAL FUNCTION 和 CREATE MODEL 和 CREATE LIBRARY 命令。 | 2021 年 11 月 18 日 |
| AmazonRedshiftServiceLinkedRolePolicy – 对现有策略的更新 | Amazon Redshift 添加了新操作的权限，允许管理 Amazon Redshift CloudWatch 日志组和日志流，包括审计日志导出。 | 2021 年 11 月 15 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftFullAccess – 对现有策略的更新 | Amazon Redshift 添加了新权限。以允许模型可解释性、DynamoDB、Redshift Spectrum 和 Amazon RDS 联合身份验证等功能。 | 2021 年 10 月 7 日 |
| AmazonRedshiftQueryEditorV2FullAccess – 新策略 | Amazon Redshift 添加了一项新策略，允许对 Amazon Redshift 查询编辑器 v2 的完全访问。 | 2021 年 9 月 24 日 |
| AmazonRedshiftQueryEditorV2NoSharing – 新策略 | Amazon Redshift 添加了一项新策略，允许在不共享资源的情况下使用 Amazon Redshift 查询编辑器 v2。 | 2021 年 9 月 24 日 |
| AmazonRedshiftQueryEditorV2ReadSharing – 新策略 | Amazon Redshift 添加了一项新策略，允许在 Amazon Redshift 查询编辑器 v2 中进行读取共享。 | 2021 年 9 月 24 日 |
| AmazonRedshiftQueryEditorV2ReadWriteSharing – 新策略 | Amazon Redshift 添加了一项新策略，允许在 Amazon Redshift 查询编辑器 v2 中进行读取和更新共享。 | 2021 年 9 月 24 日 |
| AmazonRedshiftFullAccess – 对现有策略的更新 | Amazon Redshift 添加了新的权限，以允许 `sagemaker:*Job*` 。 | 2021 年 8 月 18 日 |
| AmazonRedshiftDataFullAccess – 对现有策略的更新 | Amazon Redshift 添加了新的权限，以允许 `AuthorizeDataShare` 。 | 2021 年 8 月 12 日 |

| 更改 | 描述 | 日期 |
| --- | --- | --- |
| AmazonRedshiftDataFullAccess – 对现有策略的更新 | Amazon Redshift 添加了新的权限，以允许 `BatchExecuteStatement` 。 | 2021 年 7 月 27 日 |
| Amazon Redshift 开始跟踪更改 | Amazon Redshift 开始跟踪其 AWS 托管式策略的更改。 | 2021 年 7 月 27 日 |

## AmazonRedshiftReadOnlyAccess

授予 AWS 账户对所有 Amazon Redshift 资源的只读访问权限。

您可以在 IAM 控制台上找到 AmazonRedshiftReadOnlyAccess 策略，可以在《AWS 托管式策略参考指南》中找到 AmazonRedshiftReadOnlyAccess 相关信息。

## AmazonRedshiftFullAccess

授予 AWS 账户对所有 Amazon Redshift 资源的完全访问权限。此外，此策略还授予对所有 Amazon Redshift Serverless 资源的完全访问权限。

您可以在 IAM 控制台上找到 AmazonRedshiftFullAccess 策略，在《AWS 托管式策略参考指南》中找到 AmazonRedshiftFullAccess。

## AmazonRedshiftQueryEditor

授予账户对 Amazon Redshift 控制台查询编辑器的完全访问权限。

您可以在 IAM 控制台上找到 AmazonRedshiftQueryEditor 策略，在《AWS 托管式策略参考指南》中找到 AmazonRedshiftQueryEditor。

## AmazonRedshiftDataFullAccess

授予 AWS 账户对所有 Amazon Redshift 数据 API 操作和资源的完全访问权限。

您可以在 IAM 控制台上找到 AmazonRedshiftDataFullAccess 策略，在《AWS 托管式策略参考指南》中找到 AmazonRedshiftDataFullAccess。

### AmazonRedshiftQueryEditorV2FullAccess

授予账户对 Amazon Redshift 查询编辑器 v2 的完全访问权限。此策略还授予访问其它所需服务的权限。

您可以在 IAM 控制台上找到 AmazonRedshiftQueryEditorV2FullAccess 策略，在《AWS 托管式策略参考指南》中找到 AmazonRedshiftQueryEditorV2FullAccess。

### AmazonRedshiftQueryEditorV2NoSharing

授予账户使用 Amazon Redshift 查询编辑器 v2 的能力（资源不共享）。此策略还授予访问其它所需服务的权限。使用此策略的主体无法标记其资源（例如查询），因此无法与同一 AWS 账户 中的其它主体共享这些资源。

您可以在 IAM 控制台上找到 AmazonRedshiftQueryEditorV2NoSharing 策略，在《AWS 托管式策略参考指南》中找到 AmazonRedshiftQueryEditorV2NoSharing。

### AmazonRedshiftQueryEditorV2ReadSharing

授予账户使用 Amazon Redshift 查询编辑器 v2 的能力（资源共享受限）。此策略还授予访问其它所需服务的权限。使用此策略的主体可以标记其资源（例如查询），以便与同一 AWS 账户 中的其它主体共享这些资源。获得授权的主体可读取其与团队共享的资源，但无法更新。

您可以在 IAM 控制台上找到 AmazonRedshiftQueryEditorV2ReadSharing 策略，在《AWS 托管式策略参考指南》中找到 AmazonRedshiftQueryEditorV2ReadSharing。

### AmazonRedshiftQueryEditorV2ReadWriteSharing

授予账户使用 Amazon Redshift 查询编辑器 v2 共享资源的能力。此策略还授予访问其它所需服务的权限。使用此策略的主体可以标记其资源（例如查询），以便与同一 AWS 账户 中的其它主体共享这些资源。获得授权的主体可以读取和更新其与团队共享的资源。

您可以在 IAM 控制台上找到 AmazonRedshiftQueryEditorV2ReadWriteSharing 策略，在《AWS 托管式策略参考指南》中找到 AmazonRedshiftQueryEditorV2ReadWriteSharing。

### AmazonRedshiftServiceLinkedRolePolicy

您无法将 AmazonRedshiftServiceLinkedRolePolicy 策略附加至您的 IAM 实体。把此策略附加至服务相关的角色，该角色允许 Amazon Redshift 访问账户资源。有关更多信息，请参阅使用面向 Amazon Redshift 的服务相关角色。

您可以在 IAM 控制台上找到 AmazonRedshiftServiceLinkedRolePolicy 策略，在《AWS 托管式策略参考指南》中找到 AmazonRedshiftServiceLinkedRolePolicy。

### AmazonRedshiftAllCommandsFullAccess

授予账户使用从 Amazon Redshift 控制台创建的 IAM 角色的能力，并将其设置为默认角色，以便集群从 Amazon S3 运行 COPY、UNLOAD、CREATE EXTERNAL SCHEMA、CREATE EXTERNAL FUNCTION 和 CREATE MODEL 命令。该策略还授予账户为相关服务运行 SELECT 语句的权限，例如 Amazon S3、CloudWatch Logs、Amazon SageMaker AI 或 AWS Glue。

您可以在 IAM 控制台上找到 AmazonRedshiftAllCommandsFullAccess 策略，在《AWS 托管式策略参考指南》中找到 AmazonRedshiftAllCommandsFullAccess。

### AmazonRedshiftFederatedAuthorization

该策略整合了使用 Amazon Redshift 联合身份验证权限对 Glue Data Catalog 数据库运行查询所需的 IAM 操作。此类查询通过 AWS Glue 执行，因此需要对目录对象执行 Get 操作来发现这些对象，并需要对目录对象执行 Create、Update、Rename 和 Delete 操作来修改这些对象。请注意，资源由 Amazon Redshift 管理，因此主体还需获得 Redshift 权限才能完成查询。`glue:FederateAuthorization` 操作允许 AWS Glue 将目录对象的授权决策委派给 Amazon Redshift。

该策略允许主体使用 Amazon Redshift 联合身份验证权限对目录运行查询，但禁止向 AWS Glue 注册和取消注册 Amazon Redshift 命名空间。请参阅有关介绍 Amazon Redshift 联合身份验证权限设置的 IAM 策略要求的文档。

您可以在 IAM 控制台上找到 AmazonRedshiftFederatedAuthorization 策略，在《AWS 托管式策略参考指南》中找到 AmazonRedshiftFederatedAuthorization。

此外，您还可以创建自定义 IAM 策略，以授予账户 Amazon Redshift API 操作和资源访问的权限。您可以将这些自定义策略附加到需要这些权限的 IAM 角色或组。

## 使用 Redshift Spectrum 所需的权限

Amazon Redshift Spectrum 需要其他 AWS 服务访问资源的权限。有关 Redshift Spectrum 的 IAM 策略中的权限的详细信息，请参阅 Amazon Redshift 数据库开发人员指南中的 Amazon Redshift Spectrum 的 IAM 策略。

## 使用 Amazon Redshift 控制台所需的权限

用户若要能够使用 Amazon Redshift 控制台，则必须拥有一组最低的权限来允许用户为自己的 AWS 账户描述 Amazon Redshift 资源。这些权限还必须允许用户描述其他相关信息（包括 Amazon EC2 安全、Amazon CloudWatch、Amazon SNS 和网络信息）。

如果创建比必需的最低权限更为严格的 IAM 策略，对于附加了该 IAM 策略的用户，控制台无法按预期正常运行。要确保这些用户仍可使用 Amazon Redshift 控制台，也可向用户附加 `AmazonRedshiftReadOnlyAccess` 托管式策略。其操作方法，请见AWS适用于 Amazon Redshift 的 托管式策略所述。

有关授予用户访问 Amazon Redshift 控制台上的查询编辑器的权限，请参阅使用 Amazon Redshift 控制台查询编辑器所需的权限。

对于只需要调用 AWS CLI 或 Amazon Redshift API 的用户，无需为其提供最低限度的控制台权限。

## 使用 Amazon Redshift 控制台查询编辑器所需的权限

一个用户若要使用 Amazon Redshift 查询编辑器，该用户必须具有一组 Amazon Redshift 和 Amazon Redshift 数据 API 操作的最低权限。要使用密钥连接到数据库，您还必须具有 Secrets Manager 权限。

要授予用户对 Amazon Redshift 控制台上的查询编辑器的访问权限，请附加 `AmazonRedshiftQueryEditor` 和 `AmazonRedshiftReadOnlyAccess` AWS 托管式策略。`AmazonRedshiftQueryEditor` 策略允许用户只检索其自己的 SQL 语句的结果。也就是说，相同 `aws:userid` 提交的语句，如 `AmazonRedshiftQueryEditor` AWS 托管式策略的此部分所示。

```
{
    "Sid":"DataAPIIAMStatementPermissionsRestriction",
    "Action": [
        "redshift-data:GetStatementResult",
        "redshift-data:CancelStatement",
        "redshift-data:DescribeStatement",
        "redshift-data:ListStatements"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "redshift-data:statement-owner-iam-userid": "${aws:userid}"
        }
    }
```

```
}
```

要允许用户检索同一 IAM 角色中其他用户的 SQL 语句的结果，请创建您自己的策略，而不需要限制对当前用户的访问权限的条件。同时限制管理员更改策略的访问权限。

## 使用查询编辑器 v2 所需的权限

需要使用 Amazon Redshift 查询编辑器 v2 的用户必须拥有一组 Amazon Redshift、查询编辑器 v2 操作和其它操作的最低权限，以及其它 AWS 服务的权限，诸如 AWS Key Management Service、AWS Secrets Manager和标记服务。

要授予用户对查询编辑器 v2 的完全访问权，请附上 `AmazonRedshiftQueryEditorV2FullAccess` AWS 托管式策略。这些 `AmazonRedshiftQueryEditorV2FullAccess` 策略授权用户与同一团队中的其它人共享查询编辑器 v2 资源（例如查询）。有关如何控制查询编辑器 v2 资源访问的详细信息，请参阅 IAM 控制台上查询编辑器 v2 的特定托管式策略的定义。

一些 Amazon Redshift 查询编辑器 v2 AWS 托管式策略在条件中使用 AWS 标签，以限定对资源的访问。在查询编辑器 v2 中，共享查询的基础是附加至主体（IAM 角色）的 IAM 策略中的标签键和值 `"aws:ResourceTag/sqlworkbench-team": "${aws:PrincipalTag/sqlworkbench-team}"`。同样 AWS 账户 的主体具有相同标记值（例如 `accounting-team`），在查询编辑器 v2 中处于同一个团队。您同时只能与一个团队关联。具有管理权限的用户可以在 IAM 控制台上设置团队，方法是为所有团队成员提供相同的 `sqlworkbench-team` 标记值。如果标记值 `sqlworkbench-team` 已为 IAM 用户或 IAM 角色更改，可能要经过延迟后，更改才会反映在共享资源中。如果资源（例如查询）的标记值发生了更改，在更改生效之前可能再次出现延迟。团队成员还必须拥有 `tag:GetResources` 权限才能分享。

示例：添加 IAM 角色的 **`accounting-team`** 标签

1. 登录AWS 管理控制台，然后通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/iam/。
2. 在控制台的导航窗格中，选择角色，然后选择要编辑的角色的名称。
3. 选择标签选项卡，然后选择添加标签。
4. 添加标记密钥 sqlworkbench-team 和值 `accounting-team`。
5. 选择保存更改。

   现在，当 IAM 主体（附加了此 IAM 角色）与团队共享查询时，其它具有同样 `accounting-team` 标记值的主体可以查看查询。

有关如何将标签附加至主体（包括 IAM 角色和 IAM 用户）的更多信息，请参阅《IAM 用户指南》中的标记 IAM 资源部分。

您还可以使用身份提供者 (IdP) 在会话级别设置团队。这允许使用同一 IAM 角色的多个用户拥有不同的团队。IAM 角色信任策略必须允许 `sts:TagSession` 操作。有关更多信息，请参阅《IAM 用户指南》中的添加会话标签所需的权限。将主标签属性添加到 IdP 提供的 SAML 断言中。

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:sqlworkbench-
team">
    <AttributeValue>accounting-team</AttributeValue>
</Attribute>
```

按照身份提供者 (IdP) 提供的说明使用来自目录的内容填充 SAML 属性。有关身份提供者 (IdP) 和 Amazon Redshift 的更多信息，请参阅《IAM 用户指南》 中的使用 IAM 身份验证生成数据库用户凭证和身份提供者和联合身份验证。

`sqlworkbench:CreateNotebookVersion` 授予在账户上获取笔记本单元格的当前内容和创建笔记本版本的权限。这意味着，在创建版本时，笔记本的当前内容与该版本的内容相同。稍后，在更新当前笔记本时，版本中单元格的内容保持不变。`sqlworkbench:GetNotebookVersion` 授予获取笔记本版本的权限。没有 `sqlworkbench:BatchGetNotebookCell` 权限但拥有笔记本的 `sqlworkbench:CreateNotebookVersion` 和 `sqlworkbench:GetNotebookVersion` 权限的用户可以访问该版本中的笔记本单元格。此用户没有 `sqlworkbench:BatchGetNotebookCell` 权限，但仍然能够通过先创建一个版本，然后获取这个已创建版本来检索笔记本单元格的内容。

## 使用 Amazon Redshift 调度程序所需的权限

使用 Amazon Redshift 计划程序时，您应设置与 Amazon Redshift 计划程序 (**`scheduler.redshift.amazonaws.com`**) 具有信任关系的 IAM 角色，以便允许该计划程序代表您承担权限。您还可以为要计划的 Amazon Redshift API 操作将策略（权限）附加到角色。

以下示例演示 JSON 格式的策略文档，该策略用于在 Amazon Redshift 调度程序和 Amazon Redshift 之间设置信任关系。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "scheduler.redshift.amazonaws.com",
                    "redshift.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

有关信任实体更多信息，请参阅 IAM 用户指南中的创建向 AWS 服务委派权限的角色。

您还必须为要计划的 Amazon Redshift 操作添加权限。

为使计划程序使用 `ResizeCluster` 操作，请为 IAM 策略添加如下权限。根据您的环境，您可能希望使策略限制更严格。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "redshift:ResizeCluster",
            "Resource": "*"
        }
    ]
}
```

有关为 Amazon Redshift 调度程序创建角色的步骤，请参阅《IAM 用户指南》中的为 AWS 服务创建角色（控制台）。在 IAM 控制台中创建角色时，请选择以下选项：

- 对于选择将使用此角色的服务：选择 Redshift。
- 对于选择您的使用案例，选择 Redshift - 计划程序。

- 为允许计划的 Amazon Redshift 操作的角色创建或附加策略。选择创建策略或修改角色以便附加策略。输入计划操作的 JSON 策略。
- 在创建角色后，编辑 IAM 角色的信任关系以便包含服务 `redshift.amazonaws.com`。

您创建的 IAM 角色具有信任实体 `scheduler.redshift.amazonaws.com` 和 `redshift.amazonaws.com`。它还具有附加策略，该策略允许支持的 Amazon Redshift API 操作，例如 `"redshift:ResizeCluster"`。

## 使用 Amazon EventBridge 调度程序所需的权限

使用 Amazon EventBridge 调度程序时，您应设置与 EventBridge 计划程序 (**`events.amazonaws.com`**) 具有信任关系的 IAM 角色，以便允许该计划程序代表您承担权限。您还可以为要计划的 Amazon Redshift 数据 API 操作将策略（权限）附加到角色，并附加 Amazon EventBridge 操作的策略。

当您使用控制台上的 Amazon Redshift 查询编辑器创建计划查询时，您可以使用 EventBridge 调度程序。

您可以创建 IAM 角色以在 IAM 控制台上运行计划查询。在此 IAM 角色中，附加 `AmazonEventBridgeFullAccess` 和 `AmazonRedshiftDataFullAccess`。

以下示例演示 JSON 格式的策略文档，该策略用于设置与EventBridge 调度程序的信任关系。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "events.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

有关信任实体更多信息，请参阅 IAM 用户指南中的创建向 AWS 服务委派权限的角色。

有关为 EventBridge 调度程序创建角色的步骤，请参阅《IAM 用户指南》中的为 AWS 服务创建角色（控制台）。在 IAM 控制台中创建角色时，请选择以下选项：

- 在选择将使用此角色的服务下，选择 CloudWatch Events。
- 对于选择您的使用案例：选择 CloudWatch Events。
- 附加以下权限策略：`AmazonEventBridgeFullAccess` 和 `AmazonRedshiftDataFullAccess`。

您创建的 IAM 角色具有信任实体 `events.amazonaws.com`。它还具有附加策略，该策略允许支持的 Amazon Redshift 数据 API 操作，例如 `"redshift-data:*"`。

## 使用 Amazon Redshift 机器学习 (ML) 所需的权限

接下来，您可以找到使用 Amazon Redshift 机器学习 (ML) 所需权限的描述（针对不同的使用案例）。

为了让用户在 Amazon SageMaker AI 中使用 Amazon Redshift ML，请创建一个 IAM 角色，该角色拥有的策略比默认策略更具限制性。您可以使用以下策略。您还可以修改此策略以满足您的需求。

以下策略显示了从 Amazon Redshift 运行 SageMaker AI Autopilot（具备模型可解释性）所需的权限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateTrainingJob",
                "sagemaker:CreateAutoMLJob",
                "sagemaker:CreateCompilationJob",
                "sagemaker:CreateEndpoint",
                "sagemaker:DescribeAutoMLJob",
                "sagemaker:DescribeTrainingJob",
                "sagemaker:DescribeCompilationJob",
                "sagemaker:DescribeProcessingJob",
                "sagemaker:DescribeTransformJob",
                "sagemaker:ListCandidatesForAutoMLJob",
                "sagemaker:StopAutoMLJob",
```

```
                "sagemaker:StopCompilationJob",
                "sagemaker:StopTrainingJob",
                "sagemaker:DescribeEndpoint",
                "sagemaker:InvokeEndpoint",
                "sagemaker:StopProcessingJob",
                "sagemaker:CreateModel",
                "sagemaker:CreateProcessingJob"
            ],
            "Resource": [
                "arn:aws:sagemaker:*:*:model/*redshift*",
                "arn:aws:sagemaker:*:*:training-job/*redshift*",
                "arn:aws:sagemaker:*:*:automl-job/*redshift*",
                "arn:aws:sagemaker:*:*:compilation-job/*redshift*",
                "arn:aws:sagemaker:*:*:processing-job/*redshift*",
                "arn:aws:sagemaker:*:*:transform-job/*redshift*",
                "arn:aws:sagemaker:*:*:endpoint/*redshift*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/sagemaker/Endpoints/*redshift*",
                "arn:aws:logs:*:*:log-group:/aws/sagemaker/ProcessingJobs/
*redshift*",
                "arn:aws:logs:*:*:log-group:/aws/sagemaker/TrainingJobs/
*redshift*",
                "arn:aws:logs:*:*:log-group:/aws/sagemaker/TransformJobs/
*redshift*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                "cloudwatch:namespace": [
                    "SageMaker",
                    "/aws/sagemaker/Endpoints",
                    "/aws/sagemaker/ProcessingJobs",
                    "/aws/sagemaker/TrainingJobs",
                    "/aws/sagemaker/TransformJobs"
                ]
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "ecr:BatchCheckLayerAvailability",
            "ecr:BatchGetImage",
            "ecr:GetAuthorizationToken",
            "ecr:GetDownloadUrlForLayer"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetObject",
            "s3:GetBucketAcl",
            "s3:GetBucketCors",
            "s3:GetEncryptionConfiguration",
            "s3:GetBucketLocation",
            "s3:ListBucket",
            "s3:ListAllMyBuckets",
            "s3:ListMultipartUploadParts",
            "s3:ListBucketMultipartUploads",
            "s3:PutObject",
            "s3:PutBucketAcl",
            "s3:PutBucketCors",
            "s3:DeleteObject",
            "s3:AbortMultipartUpload",
            "s3:CreateBucket"
        ],
        "Resource": [
            "arn:aws:s3:::redshift-downloads",
            "arn:aws:s3:::redshift-downloads/*",
            "arn:aws:s3:::*redshift*",
            "arn:aws:s3:::*redshift*/*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetBucketAcl",
                "s3:GetBucketCors",
                "s3:GetEncryptionConfiguration",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListAllMyBuckets",
                "s3:ListMultipartUploadParts",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject",
                "s3:PutBucketAcl",
                "s3:PutBucketCors",
                "s3:DeleteObject",
                "s3:AbortMultipartUpload",
                "s3:CreateBucket"
            ],
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "s3:ExistingObjectTag/Redshift": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "redshift.amazonaws.com",
                        "sagemaker.amazonaws.com"
                    ]
                }
            }
        }
    ]
```

```
}
```

以下策略显示了允许访问 Amazon DynamoDB、Redshift Spectrum 和 Amazon RDS 联合身份验证的完全最低权限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sagemaker:CreateTrainingJob",
                "sagemaker:CreateAutoMLJob",
                "sagemaker:CreateCompilationJob",
                "sagemaker:CreateEndpoint",
                "sagemaker:DescribeAutoMLJob",
                "sagemaker:DescribeTrainingJob",
                "sagemaker:DescribeCompilationJob",
                "sagemaker:DescribeProcessingJob",
                "sagemaker:DescribeTransformJob",
                "sagemaker:ListCandidatesForAutoMLJob",
                "sagemaker:StopAutoMLJob",
                "sagemaker:StopCompilationJob",
                "sagemaker:StopTrainingJob",
                "sagemaker:DescribeEndpoint",
                "sagemaker:InvokeEndpoint",
                "sagemaker:StopProcessingJob",
                "sagemaker:CreateModel",
                "sagemaker:CreateProcessingJob"
            ],
            "Resource": [
                "arn:aws:sagemaker:*:*:model/*redshift*",
                "arn:aws:sagemaker:*:*:training-job/*redshift*",
                "arn:aws:sagemaker:*:*:automl-job/*redshift*",
                "arn:aws:sagemaker:*:*:compilation-job/*redshift*",
                "arn:aws:sagemaker:*:*:processing-job/*redshift*",
                "arn:aws:sagemaker:*:*:transform-job/*redshift*",
                "arn:aws:sagemaker:*:*:endpoint/*redshift*"
            ]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/sagemaker/Endpoints/*redshift*",
                "arn:aws:logs:*:*:log-group:/aws/sagemaker/ProcessingJobs/
*redshift*",
                "arn:aws:logs:*:*:log-group:/aws/sagemaker/TrainingJobs/
*redshift*",
                "arn:aws:logs:*:*:log-group:/aws/sagemaker/TransformJobs/
*redshift*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "SageMaker",
                        "/aws/sagemaker/Endpoints",
                        "/aws/sagemaker/ProcessingJobs",
                        "/aws/sagemaker/TrainingJobs",
                        "/aws/sagemaker/TransformJobs"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetAuthorizationToken",
                "ecr:GetDownloadUrlForLayer"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetBucketAcl",
                "s3:GetBucketCors",
                "s3:GetEncryptionConfiguration",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListAllMyBuckets",
                "s3:ListMultipartUploadParts",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject",
                "s3:PutBucketAcl",
                "s3:PutBucketCors",
                "s3:DeleteObject",
                "s3:AbortMultipartUpload",
                "s3:CreateBucket"
            ],
            "Resource": [
                "arn:aws:s3:::redshift-downloads",
                "arn:aws:s3:::redshift-downloads/*",
                "arn:aws:s3:::*redshift*",
                "arn:aws:s3:::*redshift*/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetBucketAcl",
                "s3:GetBucketCors",
                "s3:GetEncryptionConfiguration",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListAllMyBuckets",
                "s3:ListMultipartUploadParts",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject",
                "s3:PutBucketAcl",
                "s3:PutBucketCors",
```

```
                "s3:DeleteObject",
                "s3:AbortMultipartUpload",
                "s3:CreateBucket"
            ],
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "s3:ExistingObjectTag/Redshift": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:Scan",
                "dynamodb:DescribeTable",
                "dynamodb:Getitem"
            ],
            "Resource": [
                "arn:aws:dynamodb:*:*:table/*redshift*",
                "arn:aws:dynamodb:*:*:table/*redshift*/index/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:ListInstances"
            ],
            "Resource": [
                "arn:aws:elasticmapreduce:*:*:cluster/*redshift*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:ListInstances"
            ],
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "elasticmapreduce:ResourceTag/Redshift": "true"
                }
            }
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": "arn:aws:lambda:*:*:function:*redshift*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "glue:CreateDatabase",
                "glue:DeleteDatabase",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:UpdateDatabase",
                "glue:CreateTable",
                "glue:DeleteTable",
                "glue:BatchDeleteTable",
                "glue:UpdateTable",
                "glue:GetTable",
                "glue:GetTables",
                "glue:BatchCreatePartition",
                "glue:CreatePartition",
                "glue:DeletePartition",
                "glue:BatchDeletePartition",
                "glue:UpdatePartition",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:BatchGetPartition"
            ],
            "Resource": [
                "arn:aws:glue:*:*:table/*redshift*/*",
                "arn:aws:glue:*:*:catalog",
                "arn:aws:glue:*:*:database/*redshift*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds"
            ],
```

```
            "Resource": [
                "arn:aws:secretsmanager:*:*:secret:*redshift*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetRandomPassword",
                "secretsmanager:ListSecrets"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "secretsmanager:ResourceTag/Redshift": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "redshift.amazonaws.com",
                        "glue.amazonaws.com",
                        "sagemaker.amazonaws.com",
                        "athena.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

或者，要使用 AWS KMS 密钥用于加密，将以下权限添加到策略中。

```
{
    "Effect": "Allow",
    "Action": [
```

```
        "kms:CreateGrant",
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": [
        "arn:aws:kms:<your-region>:<your-account-id>:key/<your-kms-key>"
    ]
}
```

要支持 Amazon Redshift 和 SageMaker AI 代入先前的 IAM 角色以便与其它服务交互，请将以下信任策略添加到 IAM 角色中。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "redshift.amazonaws.com",
          "sagemaker.amazonaws.com",
          "forecast.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

先前的 Amazon S3 桶 `redshift-downloads/redshift-ml/`，是存储用于其他步骤和示例的示例数据的位置。如果您不需要从 Amazon S3 加载数据，您可以将桶删除。或者，将其替换为用于将数据加载到 Amazon Redshift 的其它 Amazon S3 桶。

**`your-account-id`**、**`your-role`** 和 **`your-s3-bucket`** 值是您在 CREATE MODEL（创建模型）命令中指定的账户 ID、角色和桶。

（可选）如果您为使用 Amazon Redshift ML 指定了一个 AWS KMS 密钥，请使用示例策略的 AWS KMS 密钥部分。**your-kms-key** 值是作为 CREATE MODEL 命令一部分使用的键。

如果您为超参数优化任务指定了一个私有的 Virtual Private Cloud (VPC)，请添加以下权限：

```
{
            "Effect": "Allow",
            "Action": [
            "ec2:CreateNetworkInterface",
            "ec2:CreateNetworkInterfacePermission",
            "ec2:DeleteNetworkInterface",
            "ec2:DeleteNetworkInterfacePermission",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeVpcs",
            "ec2:DescribeDhcpOptions",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
            ]
}
```

要使用模型解释，请确保您具备调用 SageMaker AI API 操作的权限。建议使用 `AmazonSageMakerFullAccess` 托管式策略。如果您要使用更具限制性的策略创建 IAM 角色，您可以使用以下策略。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sagemaker:CreateEndpoint",
        "sagemaker:CreateEndpointConfig",
        "sagemaker:DeleteEndpoint",
        "sagemaker:DeleteEndpointConfig",
        "sagemaker:DescribeEndpoint",
        "sagemaker:DescribeEndpointConfig",
        "sagemaker:DescribeModel",
        "sagemaker:InvokeEndpoint",
        "sagemaker:ListTags"
      ],
```

```
      "Resource": "*"
    }
  ]
}
```

有关 AmazonSageMakerFullAccess 托管式策略的更多信息，请参阅《Amazon SageMaker AI 开发人员指南》中的 AmazonSageMakerFullAccess。

如果您想创建预测模型，我们建议您使用 AmazonForecastFullAccess 托管式策略。如果您要使用更具限制性的策略，请将以下策略添加到您的 IAM 角色中。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "forecast:CreateAutoPredictor",
                "forecast:CreateDataset",
                "forecast:CreateDatasetGroup",
                "forecast:CreateDatasetImportJob",
                "forecast:CreateForecast",
                "forecast:CreateForecastExportJob",
                "forecast:DeleteResourceTree",
                "forecast:DescribeAutoPredictor",
                "forecast:DescribeDataset",
                "forecast:DescribeDatasetGroup",
                "forecast:DescribeDatasetImportJob",
                "forecast:DescribeForecast",
                "forecast:DescribeForecastExportJob",
                "forecast:StopResource",
                "forecast:TagResource",
                "forecast:UpdateDatasetGroup"
            ],
            "Resource": "*"
        }
    ]
}
```

如果您想创建 Amazon Bedrock 模型，我们建议您使用 AmazonBedrockFullAccess 托管式策略。如果您要使用更具限制性的策略，请将以下策略添加到您的 IAM 角色中。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "bedrock:InvokeModel",
            "Resource": [
                "*",
                "arn:aws:bedrock:us-east-1::foundation-model/*"
            ]
        }
    ]
}
```

有关 Amazon Redshift ML 的更多信息，请参阅在 Amazon Redshift 中使用机器学习、CREATE MODEL 或 CREATE EXTERNAL MODEL。

## 串流摄取的权限

串流摄取适用于两项服务。这两项服务是 Kinesis Data Streams 和 Amazon MSK。

在 Kinesis Data Streams 中使用串流摄取所需的权限

有关托管式策略示例的过程，请参阅开始使用 Amazon Kinesis Data Streams 串流摄取。

在 Amazon MSK 中使用串流摄取所需的权限

有关托管式策略示例的过程，请参阅开始使用 Amazon Managed Streaming for Apache Kafka 串流摄取。

## 使用数据共享 API 操作所需的权限

要控制对数据共享 API 操作的访问，请使用基于 IAM 操作的策略。有关如何管理 IAM 策略的信息，请参阅《IAM 用户指南》中的管理 IAM 策略。

当创建者集群管理员需要使用 AuthorizeDataShare 调用来授权 AWS 账户 账户外数据共享出口的时候，更是如此。在这种情况下，您可以设置基于 IAM 操作的策略来授予此权限。使用 DeauthorizeDataShare 调用以撤消出口。

使用基于 IAM 操作的策略时，您还可以在策略中指定 IAM 资源，例如 DataShareARN。下面显示了 DataShareARN 的格式和示例。

```
arn:aws:redshift:region:account-id:datashare:namespace-guid/datashare-name
arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-e2e24359e9a8/
SalesShare
```

您可以通过在 IAM 策略中指定数据更新名称来限制对特定数据共享的 AuthorizeDataShare 访问权限。

```
{
  "Statement": [
    {
      "Action": [
        "redshift:AuthorizeDataShare",
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-
e2e24359e9a8/SalesShare"
      ],
      "Effect": "Deny"
    }
  ]
}
```

您还可以将 IAM 策略限制为特定创建器集群拥有的所有数据共享。若要执行此操作，请将策略中的 **datashare-name** 值替换为通配符或星号。保留集群的 namespace-guid 值。

```
arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-e2e24359e9a8/
*
```

以下 IAM 策略防止实体面向特定创建者集群拥有的数据共享调用 AuthorizeDataShare。

```
{
  "Statement": [
    {
      "Action": [
```

```
        "redshift:AuthorizeDataShare",
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-
e2e24359e9a8/*"
      ],
      "Effect": "Deny"
    }
  ]
}
```

DataShareARN 会根据数据共享名称和所拥有的全局唯一的集群命名空间 ID (GUID) 来限制访问。它通过将名称指定为星号来完成此操作。

## GetClusterCredentials 的资源策略

要使用 IAM 数据库凭证通过 JDBC 或 ODBC 连接来连接到集群数据库，或以编程方式调用 GetClusterCredentials 操作，您需要拥有调用 redshift:GetClusterCredentials 操作的权限以及对 dbuser 资源的访问权限。

如果使用 JDBC 或 ODBC 连接，您可以指定 server 和 port 来代替 cluster_id 和 region，为此，您的策略必须允许能够访问 redshift:DescribeClusters 资源的 cluster 操作。

如果您使用可选参数 Autocreate、DbGroups 和 DbName 调用 GetClusterCredentials 操作，您还必须允许这些操作，并允许访问下表中列出的资源。

| GetClusterCredentials 参数 | 操作 | 资源 |
| --- | --- | --- |
| Autocreate | redshift:CreateClusterUser | dbuser |
| DbGroups | redshift:JoinGroup | dbgroup |
| DbName | NA | dbname |

有关资源的更多信息，请参阅 Amazon Redshift 资源和操作。

您还可以在策略中包括以下条件：

- `redshift:DurationSeconds`
- `redshift:DbName`
- `redshift:DbUser`

> **Important**
>
> 对于 SAML SSO 集成，您可能需要使用 `${redshift:DbUser}` 变量指定 IAM 策略。在此类情况下，我们强烈建议您使用条件语句，以确保调用方无法获取与其 AWS userid 不匹配的用户的凭证。例如，`"StringEquals": {"aws:userid":"AIDIODR4TAW7CSEXAMPLE:${redshift:DbUser}"}`。请参阅示例 8：使用 GetClusterCredentials 的 IAM 策略。有关条件的更多信息，请参阅在策略中指定条件

## 客户管理型策略示例

本节的用户策略示例介绍如何授予各 Amazon Redshift 操作的权限。当您使用 Amazon Redshift API、AWS 开发工具包或 AWS CLI 时，可以使用这些策略。

> **Note**
>
> 所有示例都使用美国西部（俄勒冈州）区域 (`us-west-2`) 并且包含虚构的账户 ID。

示例 1：为用户授予所有 Amazon Redshift 操作和资源的完全访问权限

以下策略允许访问所有资源上的所有 Amazon Redshift 操作。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"AllowRedshift",
      "Action": [
```

```
        "redshift:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

`Action` 元素中的 `redshift:*` 值指示 Amazon Redshift 中的所有操作。

示例 2：拒绝用户访问一组 Amazon Redshift 操作

默认情况下，所有权限都将被拒绝。不过，有时您需要明确拒绝对某个或某组操作的访问。以下策略允许访问所有 Amazon Redshift 操作，但明确拒绝对名称以 `Delete` 开头的任何 Amazon Redshift 操作的访问。该策略适用于 `us-west-2` 中的所有 Amazon Redshift 资源。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"AllowUSWest2Region",
      "Action": [
        "redshift:*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:redshift:us-east-1:111122223333:*"
    },
   {
     "Sid":"DenyDeleteUSWest2Region",
     "Action": [
        "redshift:Delete*"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:redshift:us-east-1:111122223333:*"
   }
  ]
}
```

示例 3：允许用户管理集群

以下策略允许用户创建、删除、修改和重启所有集群，但拒绝删除名称以 protected 开头的任何集群的权限。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"AllowClusterManagement",
      "Action": [
        "redshift:CreateCluster",
        "redshift:DeleteCluster",
        "redshift:ModifyCluster",
        "redshift:RebootCluster"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Sid":"DenyDeleteProtected",
      "Action": [
        "redshift:DeleteCluster"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:cluster:protected*"
      ],
      "Effect": "Deny"
    }
  ]
}
```

示例 4：允许用户授予和撤销快照访问权限

以下策略允许用户（如用户 A）执行以下操作：

- 授予对从名为 shared 的集群中创建的任何快照的访问权限。

- 撤消对从快照名称以 shared 开头的 revokable 集群中创建的任何快照的访问权限。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"AllowSharedSnapshots",
      "Action": [
        "redshift:AuthorizeSnapshotAccess"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:shared/*"
      ],
      "Effect": "Allow"
    },
    {
      "Sid":"AllowRevokableSnapshot",
      "Action": [
        "redshift:RevokeSnapshotAccess"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:snapshot:*/revokable*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

如果用户 A 允许用户 B 访问快照，则用户 B 必须拥有以下某项策略才能从该快照还原集群。以下策略允许用户 B 描述集群、从快照还原集群以及创建集群。这些集群的名称必须以 from-other-account 开头。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
```

```
    {
      "Sid":"AllowDescribeSnapshots",
      "Action": [
        "redshift:DescribeClusterSnapshots"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Sid":"AllowUserRestoreFromSnapshot",
      "Action": [
        "redshift:RestoreFromClusterSnapshot"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:snapshot:*/*",
        "arn:aws:redshift:us-west-2:444455556666:cluster:from-other-account*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

示例 5：允许用户复制集群快照以及从快照中还原集群

以下策略允许用户复制从名为 big-cluster-1 的集群中创建的任何快照，以及还原名称以 snapshot-for-restore 开头的任何快照。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"AllowCopyClusterSnapshot",
      "Action": [
        "redshift:CopyClusterSnapshot"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:snapshot:big-cluster-1/*"
      ],
```

```
      "Effect": "Allow"
    },
    {
      "Sid":"AllowRestoreFromClusterSnapshot",
      "Action": [
        "redshift:RestoreFromClusterSnapshot"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:snapshot:*/snapshot-for-
restore*",
        "arn:aws:redshift:us-west-2:123456789012:cluster:*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

示例 6：允许用户访问 Amazon Redshift 以及相关 AWS 服务的常见操作和资源

以下示例策略允许访问 Amazon Redshift、Amazon Simple Notification Service (Amazon SNS) 和 Amazon CloudWatch 的所有操作和资源。它还允许对账户下的所有相关 Amazon EC2 资源执行指定的操作。

> **Note**
>
> 此示例策略中指定的 Amazon EC2 操作不支持资源级权限。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"AllowRedshift",
      "Effect": "Allow",
      "Action": [
        "redshift:*"
      ],
      "Resource": [
        "*"
```

```
      ]
    },
    {
      "Sid":"AllowSNS",
      "Effect": "Allow",
        "Action": [
          "sns:*"
        ],
        "Resource": [
          "*"
        ]
      },
    {
      "Sid":"AllowCloudWatch",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid":"AllowEC2Actions",
      "Effect": "Allow",
      "Action": [
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AttachNetworkInterface",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAddresses",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeInternetGateways",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

### 示例 7：允许用户使用 Amazon Redshift 控制台标记资源

以下示例策略允许用户使用 AWS Resource Groups 通过 Amazon Redshift 控制台对资源进行标记。此策略可附加到调用新的或原始 Amazon Redshift 控制台的用户角色。有关标记的更多信息，请参阅在 Amazon Redshift 中为资源添加标签。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid":"TaggingPermissions",
            "Effect": "Allow",
            "Action": [
                "redshift:DeleteTags",
                "redshift:CreateTags",
                "redshift:DescribeTags",
                "tag:UntagResources",
                "tag:TagResources"
            ],
            "Resource": "*"
        }
    ]
}
```

### 示例 8：使用 GetClusterCredentials 的 IAM 策略

以下策略使用这些示例参数值：

- 区域：us-west-2
- AWS 账户：123456789012
- 集群名称：examplecluster

以下策略启用 GetCredentials、CreateClusterUser 和 JoinGroup 操作。仅在 AWS 用户 ID 与 "AIDIODR4TAW7CSEXAMPLE:${redshift:DbUser}@yourdomain.com" 匹配时，该策略才使用条件键以允许 GetClusterCredentials 和 CreateClusterUser 操作。IAM 访问权限仅对 "testdb" 数据库是必需的。该策略还允许用户参与名为 "common_group" 的组。

JSON

```
{
"Version":"2012-10-17",
  "Statement": [
    {
     "Sid":"GetClusterCredsStatement",
      "Effect": "Allow",
      "Action": [
        "redshift:GetClusterCredentials"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/
${redshift:DbUser}",
        "arn:aws:redshift:us-west-2:123456789012:dbname:examplecluster/testdb",
        "arn:aws:redshift:us-west-2:123456789012:dbgroup:examplecluster/
common_group"
      ],
        "Condition": {
           "StringEquals": {
           "aws:userid":"AIDIODR4TAW7CSEXAMPLE:${redshift:DbUser}@yourdomain.com"
           }
        }
    },
    {
      "Sid":"CreateClusterUserStatement",
      "Effect": "Allow",
      "Action": [
        "redshift:CreateClusterUser"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/
${redshift:DbUser}"
      ],
      "Condition": {
        "StringEquals": {
          "aws:userid":"AIDIODR4TAW7CSEXAMPLE:${redshift:DbUser}@yourdomain.com"
        }
      }
    },
    {
      "Sid":"RedshiftJoinGroupStatement",
      "Effect": "Allow",
```

```
      "Action": [
        "redshift:JoinGroup"
      ],
      "Resource": [
        "arn:aws:redshift:us-west-2:123456789012:dbgroup:examplecluster/
common_group"
      ]
    }
  ]
}
```

以下示例显示允许 IAM 角色调用 GetClusterCredentials 操作的策略。该策略指定，对于名为 examplecluster 的集群上的名为 temp_creds_user 的数据库用户，Amazon Redshift dbuser 资源为其授予角色访问权限。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "redshift:GetClusterCredentials",
    "Resource": "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/
temp_creds_user"
  }
}
```

您可使用通配符 (*) 替换完整或部分集群名称、用户名和数据库组名称。以下示例允许指定账户中的任何集群上所有以 temp_ 开头的用户名调用。

> **Important**
>
> 以下示例中的语句指定通配符 (*) 作为资源的值，以便策略允许以指定字符开头的任何资源。在 IAM 策略使用通配符可能过于宽松。作为最佳实践，我们建议对您的业务应用程序使用最严格的可行策略。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "redshift:GetClusterCredentials",
    "Resource": "arn:aws:redshift:us-west-2:123456789012:dbuser:*/temp_*"
  }
}
```

以下示例显示一个允许 IAM 角色调用 GetClusterCredentials 操作的策略，并提供了自动创建新用户和指定用户在登录时加入的组的选项。"Resource": "*" 子句向角色授予对任何资源 (包括集群、数据库用户或用户组) 的访问权限。

> **Important**
>
> 以下示例中的语句指定通配符 ( * ) 作为给定操作的资源，使策略能够允许访问任何集群和数据库用户，并支持创建任何用户。在 IAM 策略使用通配符可能过于宽松。作为最佳实践，我们建议对您的业务应用程序使用最严格的可行策略。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
             "redshift:GetClusterCredentials",
             "redshift:CreateClusterUser",
  "redshift:JoinGroup"
            ],
    "Resource": "*"
  }
}
```

有关更多信息，请参阅 Amazon Redshift ARN 语法。

# Amazon Redshift 的原生身份提供者 (IdP) 联合身份验证

借助原生身份提供者联合身份验证，管理 Amazon Redshift 的身份和权限变得更加容易，因为它利用现有身份提供者来简化身份验证和管理权限。它可以通过使身份提供者向 Redshift 共享身份元数据来实现这一目标。对于此功能的第一次迭代，受支持的身份提供者是 Microsoft Azure Active Directory (Azure AD)。

要将 Amazon Redshift 配置为对来自第三方身份提供者的身份进行身份验证，您可以向 Amazon Redshift 注册身份提供者。这样做可以让 Redshift 对身份提供者定义的用户和角色进行身份验证。这样，您可以避免必须同时在第三方身份提供者和 Amazon Redshift 中执行精细的身份管理，因为身份信息是共享的。

有关使用从身份提供者 (IdP) 组转移的会话角色的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 PG_GET_SESSION_ROLES 。

## 原生身份提供者（IdP）联合身份验证

要完成身份提供者和 Amazon Redshift 之间的初步设置，您需要执行以下几个步骤：首先，向身份提供者注册 Amazon Redshift 作为第三方应用程序，同时请求必要的 API 权限。然后在身份提供者中创建用户和组。最后，您可以使用 SQL 语句向 Amazon Redshift 注册身份提供者，这些语句设置身份提供者特有的身份验证参数。在向 Redshift 注册身份提供者的过程中，您分配一个命名空间以确保正确地对用户和角色进行分组。

在向 Amazon Redshift 注册身份提供者后，Redshift 和身份提供者之间就建立了通信。然后，客户端可以作为身份提供者实体传递令牌并向 Redshift 进行身份验证。Amazon Redshift 使用 IdP 组成员资格信息来映射到 Redshift 角色。如果用户之前不存在于 Redshift 中，则会创建该用户。创建映射到身份提供者组的角色（如果它们不存在）。Amazon Redshift 管理员授予对角色的权限，而用户可以运行查询和执行其他数据库任务。

以下步骤概述当用户登录时原生身份提供者联合身份验证的工作方式：

1. 当用户从客户端使用原生 IdP 选项登录时，身份提供者令牌将从客户端发送到驱动程序。
2. 对用户进行身份验证。如果 Amazon Redshift 中不存在此用户，则将创建一个新用户。Redshift 将用户的身份提供者组映射到 Redshift 角色。
3. 权限是根据用户的 Redshift 角色分配的。管理员将这些权限授予用户和角色。
4. 用户可以查询 Redshift。

## 桌面客户端工具

有关如何使用原生身份提供者联合身份验证通过 Power BI 连接到 Amazon Redshift 的说明，请参阅博客文章将 Amazon Redshift 原生 IdP 联合身份验证与 Microsoft Azure Active Directory (AD) 和 Power BI 集成。它介绍了使用 Azure AD 实施 Amazon Redshift 原生 IdP 设置的按步骤过程。它详细介绍了为 Power BI 桌面或 Power BI 服务设置客户端连接的步骤。这些步骤包括应用程序注册、配置权限和配置凭证。

要了解如何使用 Power BI 桌面和 JDBC 客户端 SQL WorkBench/J 将 Amazon Redshift 原生 IdP 联合身份验证与 Azure AD 集成，请观看以下视频：

有关如何使用原生身份提供者联合身份验证通过 SQL 客户端（特别是 DBEaver 或 SQL WorkBench/J）连接到 Amazon Redshift 的说明，请参阅博客文章使用 SQL 客户端将 Amazon Redshift 原生 IdP 联合身份验证与 Microsoft Azure AD 集成。

## 限制

以下限制适用：

- Amazon Redshift 驱动程序从以下版本开始支持 `BrowserIdcAuthPlugin`：
  - Amazon Redshift JDBC 驱动程序 v2.1.0.30
  - Amazon Redshift ODBC 驱动程序 v2.1.3
  - Amazon Redshift Python 驱动程序 v2.1.3
- Amazon Redshift 驱动程序从以下版本开始支持 `IdpTokenAuthPlugin`：
  - Amazon Redshift JDBC 驱动程序 v2.1.0.19
  - Amazon Redshift ODBC 驱动程序 v2.0.0.9
  - Amazon Redshift Python 驱动程序 v2.0.914
- 不支持增强型 VPC 路由：使用原生身份提供商联合身份验证时，不支持增强型 VPC 路由。有关增强型 VPC 路由的更多信息，请参阅 Amazon Redshift 中的增强型 VPC 路由。
- AWS IAM Identity Center 缓存 – AWS IAM Identity Center 会缓存会话信息。当您尝试通过 Redshift 查询编辑器 v2 连接到 Redshift 数据库时，这可能会导致无法预测的访问问题。这是因为查询编辑器 v2 中相关联的 AWS IAM Identity Center 会话即使在数据库用户退出 AWS 控制台的情况下仍然有效。缓存会在一小时后过期，这通常可以解决任何问题。

## 在 Amazon Redshift 上设置身份提供者

本节介绍配置身份提供者和 Amazon Redshift 以便为原生身份提供者联合身份验证建立通信的步骤。您需要身份提供者的有效账户。在配置 Amazon Redshift 之前，您可以向身份提供者注册 Redshift 作为应用程序，并授予管理员同意。

在 Amazon Redshift 中，完成以下步骤：

1. 运行 SQL 语句来注册身份提供者，包括 Azure 应用程序元数据的说明。要在 Amazon Redshift 中创建身份提供者，请在替换参数值 issuer、client_id、client_secret 和 audience 后运行以下命令。这些参数特定于 Microsoft Azure AD。将身份提供者名称替换为您选择的名称，然后用唯一名称替换命名空间，以包含来自身份提供者目录中的用户和角色。

```
CREATE IDENTITY PROVIDER oauth_standard TYPE azure
NAMESPACE 'aad'
PARAMETERS '{
"issuer":"https://sts.windows.net/2sdfdsf-d475-420d-b5ac-667adad7c702/",
"client_id":"<client_id>",
"client_secret":"BUAH~ewrqewrqwerUUY^%tHe1oNZShoiU7",
"audience":["https://analysis.windows.net/powerbi/connector/AmazonRedshift"]
}'
```

   类型 `azure` 表示该提供商特意加强了与 Microsoft Azure AD 的通信。这是目前唯一受支持的第三方身份提供者。

   - issuer - 收到令牌时要信任的发布者 ID。tenant_id 的唯一标识符附加到发布者之后。
   - client_id - 向身份提供者注册的应用程序的唯一公有标识符。这可以称为应用程序 ID。
   - client_secret - 只有身份提供者和注册的应用程序才知道的秘密标识符或密码。
   - audience - 在 Azure 中分配给应用程序的应用程序 ID。

   您可以在创建身份提供者时设置参数来指定证书、私有密钥和私有密钥密码，而不使用共享客户端密钥。

```
CREATE IDENTITY PROVIDER example_idp TYPE azure
NAMESPACE 'example_aad'
PARAMETERS '{"issuer":"https://sts.windows.net/2sdfdsf-d475-420d-
b5ac-667adad7c702/",
"client_id":"<client_id>",
```

```
"audience":["https://analysis.windows.net/powerbi/connector/AmazonRedshift"],
"client_x5t":"<certificate thumbprint>",
"client_pk_base64":"<private key in base64 encoding>",
"client_pk_password":"test_password"}';
```

私有密钥密码 client_pk_password 为可选项。

2. 可选：在 Amazon Redshift 中运行 SQL 命令以预先创建用户和角色。这有助于提前授予权限。Amazon Redshift 中的角色名称如下所示：<Namespace>:<GroupName on Azure AD>。例如，当您在 Microsoft Azure AD 中创建一个名为 `rsgroup` 的组和一个名为 `aad` 的命名空间时，角色名称为 `aad:rsgroup`。从身份提供者命名空间中的这些用户名和组成员资格定义 Amazon Redshift 中的用户和角色名称。

   角色和用户的映射包括验证他们的 `external_id` 值，以确保它是最新的。外部 ID 将映射到身份提供者中组或用户的标识符。例如，角色的外部 ID 将映射到相应的 Azure AD 组 ID。同样，每个用户的外部 ID 都将映射到他们在身份提供者中的 ID。

   ```
   create role "aad:rsgroup";
   ```

3. 根据您的要求向角色授予相关权限。例如：

   ```
   GRANT SELECT on all tables in schema public to role "aad:rsgroup";
   ```

4. 您还可向特定用户授予权限。

   ```
   GRANT SELECT on table foo to aad:alice@example.com
   ```

   请注意，对于联合身份验证的外部用户，其角色成员资格仅在该用户的会话中可用。这对创建数据库对象有影响。例如，当联合身份验证的外部用户创建任何视图或存储过程时，同一个用户无法将这些对象的权限委托给其他用户和角色。

命名空间的解释

命名空间将用户或角色映射到特定身份提供者。例如，在 AWS IAM 中创建的用户的前缀是 `iam:`。此前缀可防止用户名冲突，并使支持多个身份存储成为可能。如果用户 alice@example.com 来自向 aad 命名空间注册的身份源，则当此用户登录时，会在 Redshift 中创建用户 `aad:alice@example.com`（如果此用户尚未存在）。请注意，用户和角色命名空间与 Amazon Redshift 集群命名空间具有不同的功能，后者是与集群关联的唯一标识符。

## 自动为身份提供者创建 Amazon Redshift 角色

借助此特征，您可以根据身份提供者（IdP）的组成员资格在 Redshift 中自动创建角色。自动创建角色支持具有本机 IdP 集成的 Azure Active Directory。

自动创建角色有几个好处。自动创建角色时，Redshift 会在您的 IdP 中创建具有组成员资格的角色，因此您可以避免繁琐的手动角色创建和维护。还可以选择筛选映射到 Redshift 角色的组。

工作原理

当您以 IdP 用户身份登录 Redshift 时，会发生以下一系列事件：

1. Redshift 从 IdP 那里检索您的组成员资格。
2. Redshift 使用角色格式 *idp_namespace*:*rolename* 自动创建映射到这些组的角色。
3. Redshift 授予您使用映射角色的权限。

每次用户登录时，都会自动创建目录中不存在但该用户所属的每个组。您可以选择设置包括和排除筛选条件，从而控制哪些 IdP 组已创建 Redshift 角色。

配置自动创建角色

使用 CREATE IDENTITY PROVIDER 和 ALTER IDENTITY PROVIDER 命令启用和配置自动角色创建。

```
-- Create a new IdP with auto role creation enabled
CREATE IDENTITY PROVIDER <idp_name> TYPE azure
  NAMESPACE '<namespace>'
  APPLICATION_ARN 'app_arn'
  IAM_ROLE 'role_arn'
  AUTO_CREATE_ROLES TRUE;

-- Enable on existing IdP
ALTER IDENTITY PROVIDER <idp_name>
  AUTO_CREATE_ROLES TRUE;

-- Disable
ALTER IDENTITY PROVIDER <idp_name>
  AUTO_CREATE_ROLES FALSE;
```

筛选组

您可以选择使用 INCLUDE 和 EXCLUDE 模式筛选映射到 Redshift 角色的 IdP 组。当模式冲突时，EXCLUDE 优先于 INCLUDE。

```
-- Only create roles for groups with 'dev'
CREATE IDENTITY PROVIDER <idp_name> TYPE azure
  ...
  AUTO_CREATE_ROLES TRUE
  INCLUDE GROUPS LIKE '%dev%';

-- Exclude 'test' groups
ALTER IDENTITY PROVIDER <idp_name>
  AUTO_CREATE_ROLES TRUE
  EXCLUDE GROUPS LIKE '%test%';
```

示例

以下示例说明了如何启用自动创建角色而不进行筛选。

```
CREATE IDENTITY PROVIDER prod_idc TYPE azure ...
  AUTO_CREATE_ROLES TRUE;
```

以下示例包括开发组，不包括测试组。

```
ALTER IDENTITY PROVIDER prod_idc
  AUTO_CREATE_ROLES TRUE
  INCLUDE GROUPS LIKE '%dev%'
  EXCLUDE GROUPS LIKE '%test%';
```

最佳实践

为角色启用自动创建功能时，请考虑遵循以下最佳实践：

- 使用 INCLUDE 和 EXCLUDE 筛选条件来控制哪些组获得角色。
- 定期审核角色并清理未使用的角色。
- 利用 Redshift 角色层次结构来简化权限管理。

## 将 Redshift 与 AWS IAM Identity Center 连接来提供单点登录体验

您可以通过可信身份传播来管理用户和组对 Amazon Redshift 数据仓库的访问权限。

Trusted identity propagation 是一项 AWS IAM Identity Center 功能，已连接的 AWS 服务的管理员可以使用它来授予和审计对服务数据的访问权限。对这些数据的访问权限基于用户属性，例如组关联。设置可信身份传播要求已连接的 AWS 服务的管理员和 IAM Identity Center 管理员之间进行协作。有关更多信息，请参阅 Prerequisites and considerations。

以一个端到端案例来说明，您可以使用 Amazon Quick 控制面板或 Amazon Redshift 查询编辑器 v2 来访问 Redshift。在这种情况下，访问权限基于 AWS IAM Identity Center 组。Redshift 可以确定用户的身份以及他们的组成员资格。AWSIAM Identity Center 还允许通过 Okta 或 PingOne 等第三方身份提供者 (IdP) 连接和管理身份。

管理员设置 Redshift 与 AWS IAM Identity Center 之间的连接后，他们可以根据身份提供者的组来配置精细访问权限，以授权用户访问数据。

> **Important**
>
> 从 AWS IAM Identity Center 或连接的身份提供者（IdP）目录中删除用户时，该用户不会自动从 Amazon Redshift 目录中删除。要从 Amazon Redshift 目录中手动删除用户，请运行 DROP USER 命令以完全删除已从 AWS IAM Identity Center 或 IdP 中移除的用户。有关如何删除用户的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 DROP USER。

Redshift 与 AWS IAM Identity Center 集成的优势

将 AWS IAM Identity Center 与 Redshift 结合使用，可以在以下方面为企业带来益处：

- Amazon Quick 中的控制面板作者无需重新输入密码，也无需要求管理员设置具有复杂权限的 IAM 角色，即可连接 Redshift 数据来源。
- AWS IAM Identity Center 为您在 中的员工用户提供了一个中心位置。AWS您可以直接在 AWS IAM Identity Center 创建用户和群，也可以连接您在基于标准的身份提供者中管理的现有用户和组，例如 Okta、PingOne 或 Microsoft Entra ID（Azure AD）。AWSIAM Identity Center 将身份验证定向到您为用户和组选择的信任源，并维护一个用户和组的目录以供 Redshift 访问。有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的管理您的身份源和支持的身份提供者。
- 通过简单的自动发现和连接功能，您便可与多个 Redshift 集群和工作组共享一个 AWS IAM Identity Center 实例。这样可以快速添加集群，而无需额外地为每个集群配置 AWS IAM Identity Center 连接，还可确保所有集群和工作组都能一致地查看用户、其属性和组。请注意，您企业的 AWS IAM Identity Center 实例必须与要连接的任意 Redshift 数据共享位于同一区域。
- 由于用户身份已知并与数据访问记录在一起，因此您可以通过在 AWS CloudTrail 中审计用户的访问，更轻松地满足合规性监管要求。

用于连接应用程序的管理员角色

在将分析应用程序连接到 AWS IAM Identity Center 托管的 Redshift 应用程序的过程中，以下角色非常关键：

- 应用程序管理员 – 创建应用程序并配置要与哪些服务启用身份令牌交换。此管理员还需要指定哪些用户或组有权访问应用程序。
- 数据管理员 – 配置对数据的精细访问权限。AWS IAM Identity Center 中的用户和组可以映射到特定权限。

使用 AWS IAM Identity Center 通过 Amazon Quick 连接到 Amazon Redshift

以下内容演示在连接到 Redshift 并通过 AWS IAM Identity Center 管理访问权限时，如何使用 Quick 通过 Redshift 进行身份验证：授权从 Quick 到 Amazon Redshift 集群的连接。这些步骤也适用于 Amazon Redshift Serverless。

使用 AWS IAM Identity Center 通过 Amazon Redshift 查询编辑器 v2 连接到 Amazon Redshift

完成设置 AWS IAM Identity Center 与 Redshift 连接的步骤后，用户可以通过基于 AWS IAM Identity Center 且以命名空间为前缀的身份，访问数据库以及数据库中的相应对象。有关使用查询编辑器 v2 登录身份连接到 Redshift 数据库的更多信息，请参阅使用查询编辑器 v2 查询数据库。

跨多个 AWS 区域使用 AWS IAM Identity Center

Amazon Redshift 在多个 AWS 区域中支持 AWS IAM Identity Center。您可以将 AWS IAM Identity Center 从您的主 AWS 区域扩展到其它区域，从而通过靠近用户和提高可靠性来提高性能。在 AWS IAM Identity Center 中添加新区域后，您无需从主区域中复制身份，即可在新区域中创建 Redshift IAM Identity Center 应用程序。您可以在新区域中使用 AWS IAM Identity Center 设置 Amazon Redshift 联合权限，在该新区域中，您可以启用行级、列级和掩蔽控件。有关在多个区域中开始使用 AWS IAM Identity Center 的更多详细信息，请参阅《AWS IAM Identity Center 用户指南》中的在多个 AWS 区域中管理 AWS IAM Identity Center。

设置 AWS IAM Identity Center 与 Amazon Redshift 的集成

您的 Amazon Redshift 集群管理员或 Amazon Redshift Serverless 管理员必须执行多个步骤，以将 Redshift 配置为启用了 AWS IAM Identity Center 的应用程序。这样，Redshift 便可以自动发现并连接到 AWS IAM Identity Center 来获取登录和用户目录服务。在此之后，当 Redshift 管理员创建集群或工作组时，他们可以允许新的数据仓库使用 AWS IAM Identity Center 来管理数据库访问。

启用 Redshift 作为 AWS IAM Identity Center 托管应用程序的意义在于，您可以从 AWS IAM Identity Center 中或与之集成的第三方身份提供者控制用户和组的权限。当数据库用户（例如分析师或数据科学家）登录 Redshift 数据库时，服务会在 AWS IAM Identity Center 中检查他们所属的组，这些组与 Redshift 中的角色名称相匹配。通过这种方式，举例而言，定义了 Redshift 数据库角色名称的组可以访问一组表以便进行销售分析。以下部分将介绍如何完成此设置。

先决条件

将 AWS IAM Identity Center 与 Amazon Redshift 集成的先决条件如下所示：

- 账户配置 – 如果您计划使用跨账户应用场景，或者如果您在具有相同 AWS IAM Identity Center 实例的不同账户中使用 Redshift 集群，则必须在 AWS Organizations 管理账户中配置 AWS IAM Identity Center。这包括配置您的身份源。有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的入门、员工身份和支持的身份提供者。您必须确保已在 AWS IAM Identity Center 中创建用户或组，或者已从您的身份源同步用户和组，然后才能在 Redshift 中将他们指定到 Redshift 中的数据。

  > **Note**
  >
  > 您可以选择使用 AWS IAM Identity Center 的账户实例，前提是 Redshift 和 AWS IAM Identity Center 位于同一个账户中。在创建和配置 Redshift 集群或工作组时，您可以使用小组件创建此实例。

- 配置可信令牌发布者 – 在某些情况下，您可能需要使用可信令牌发布者，这是可以发布和验证信任令牌的实体。在执行此操作之前，需要先执行预备步骤，然后配置 AWS IAM Identity Center 集成的 Redshift 管理员才能选择可信令牌发布者，并添加必要的属性以完成配置。这些步骤可能包括在 AWS IAM Identity Center 控制台中，将外部身份提供者配置为可信令牌发布者并添加其属性。要完成这些步骤，请参阅 Using applications with a trusted token issuer。

  > **Note**
  >
  > 并非所有外部连接都需要设置可信令牌发布者。使用 Amazon Redshift 查询编辑器 v2 连接到您的 Redshift 数据库时，不需要配置可信令牌发布者。但可能需要对第三方应用程序进行配置，例如通过您的身份提供者进行身份验证的控制面板或自定义应用程序。

- 配置 IAM 角色 – 后面的部分提到了必须配置的权限。您必须按照 IAM 最佳实践添加权限。后面的步骤将会详细介绍具体权限。
- 增强型 VPC 路由的网络配置：如果开启了增强型 VPC 路由，则您的 VPC 必须能够访问 AWS IAM Identity Center 服务。接口 VPC 端点是提供这种连接的推荐方式。

> **Note**
>
> 请在用户登录之前设置此连接。有关将增强型 VPC 路由与 AWS IAM Identity Center 结合使用的更多信息，请参阅将 AWS IAM Identity Center 身份验证与增强型 VPC 路由结合使用。

有关更多信息，请参阅 Getting Started with AWS IAM Identity Center。

配置您的身份提供者以使用 AWS IAM Identity Center

控制用户和组身份管理的第一步是连接到 AWS IAM Identity Center 并配置您的身份提供者。您可以使用 AWS IAM Identity Center 本身作为身份提供者，也可以连接第三方身份存储，例如 Okta。有关设置与身份提供者的连接和进行配置的更多信息，请参阅《AWS IAM Identity Center 用户指南》中的连接到外部身份提供者。请确保在此过程结束时，您已将一小批用户和组添加到 AWS IAM Identity Center 以用于测试目的。

管理权限

Redshift/AWS IAM Identity Center 应用程序生命周期管理所需的权限

您必须创建一个 IAM 身份供 Redshift 管理员用来配置 Redshift，以便与 AWS IAM Identity Center 结合使用。常见的做法是，创建一个具有相应权限的 IAM 角色，并根据需要将其分配给其他身份。该角色必须具有执行以下操作的权限。

创建 Redshift/AWS IAM Identity Center 应用程序

- `sso:PutApplicationAssignmentConfiguration` – 用于安全设置。
- `sso:CreateApplication` – 用于创建 AWS IAM Identity Center 应用程序。
- `sso:PutApplicationAuthenticationMethod` – 授予 Redshift 身份验证访问权限。
- `sso:PutApplicationGrant` – 用于更改可信令牌发布者信息。
- `sso:PutApplicationAccessScope` – 用于 Redshift AWS IAM Identity Center 应用程序设置。这包括用于 AWS Lake Formation 和用于 Amazon S3 Access Grants。
- `redshift:CreateRedshiftIdcApplication` – 用于创建 Redshift AWS IAM Identity Center 应用程序。

描述 Redshift/AWS IAM Identity Center 应用程序

- `sso:GetApplicationGrant` – 用于列出可信令牌发布者信息。
- `sso:ListApplicationAccessScopes` – 用于 Redshift AWS IAM Identity Center 应用程序设置以列出下游集成，例如用于 AWS Lake Formation 和 S3 Access Grants。
- `redshift:DescribeRedshiftIdcApplications` – 用于描述现有 AWS IAM Identity Center 应用程序。

更改 Redshift/AWS IAM Identity Center 应用程序

- `redshift:ModifyRedshiftIdcApplication` – 用于更改现有的 Redshift 应用程序。
- `sso:UpdateApplication` – 用于更新 AWS IAM Identity Center 应用程序。
- `sso:GetApplicationGrant` – 获取可信令牌颁发者信息。
- `sso:ListApplicationAccessScopes` – 用于 Redshift AWS IAM Identity Center 应用程序设置。
- `sso:DeleteApplicationGrant` – 删除可信令牌颁发者信息。
- `sso:PutApplicationGrant` – 用于更改可信令牌发布者信息。
- `sso:PutApplicationAccessScope` – 用于 Redshift AWS IAM Identity Center 应用程序设置。这包括用于 AWS Lake Formation 和用于 Amazon S3 Access Grants。
- `sso:DeleteApplicationAccessScope` – 用于删除 Redshift AWS IAM Identity Center 应用程序设置。这包括用于 AWS Lake Formation 和用于 Amazon S3 Access Grants。

删除 Redshift/AWS IAM Identity Center 应用程序

- `sso:DeleteApplication` – 用于删除 AWS IAM Identity Center 应用程序。
- `redshift:DeleteRedshiftIdcApplication` – 提供删除现有 Redshift AWS IAM Identity Center 应用程序的功能。

Redshift/查询编辑器 v2 应用程序生命周期管理所需的权限

您必须创建一个 IAM 身份供 Redshift 管理员用来配置 Redshift，以便与 AWS IAM Identity Center 结合使用。常见的做法是，创建一个具有相应权限的 IAM 角色，并根据需要将其分配给其他身份。该角色必须具有执行以下操作的权限。

创建查询编辑器 v2 应用程序

- `redshift:CreateQev2IdcApplication` – 用于创建 QEV2 应用程序。

- `sso:CreateApplication` – 提供创建 AWS IAM Identity Center 应用程序的功能。
- `sso:PutApplicationAuthenticationMethod` – 授予 Redshift 身份验证访问权限。
- `sso:PutApplicationGrant` – 用于更改可信令牌发布者信息。
- `sso:PutApplicationAccessScope` – 用于 Redshift AWS IAM Identity Center 应用程序设置。这包括查询编辑器 v2。
- `sso:PutApplicationAssignmentConfiguration` – 用于安全设置。

描述查询编辑器 v2 应用程序

- `redshift:DescribeQev2IdcApplications` – 用于描述 AWS IAM Identity Center QEV2 应用程序。

更改查询编辑器 v2 应用程序

- `redshift:ModifyQev2IdcApplication` – 用于更改 AWS IAM Identity Center QEV2 应用程序。
- `sso:UpdateApplication` – 用于更改 AWS IAM Identity Center QEV2 应用程序。

删除查询编辑器 v2 应用程序

- `redshift:DeleteQev2IdcApplication` – 用于删除 QEV2 应用程序。
- `sso:DeleteApplication` – 用于删除 QEV2 应用程序。

> **Note**
>
> 在 Amazon Redshift 开发工具包中，以下 API 不可用：
>
> - CreateQev2IdcApplication
> - DescribeQev2IdcApplications
> - ModifyQev2IdcApplication
> - DeleteQev2IdcApplication
>
> 这些操作专门用于在 AWS 控制台中执行 AWS IAM Identity Center 与 Redshift QEV2 的集成。有关更多信息，请参阅 Amazon Redshift 定义的操作。

数据库管理员在控制台中连接新资源所需的权限

在创建过程中，需要这些权限才能连接新预置的集群或 Amazon Redshift Serverless 工作组。如果您具有这些权限，则控制台中会显示一个选项，供您选择连接到 AWS IAM Identity Center 托管的 Redshift 应用程序。

- `redshift:DescribeRedshiftIdcApplications`
- `sso:ListApplicationAccessScopes`
- `sso:GetApplicationAccessScope`
- `sso:GetApplicationGrant`

作为最佳实践，我们建议将权限策略附加到 IAM 角色，然后根据需要将其分配给用户和组。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

使用 AWS IAM Identity Center 将 Redshift 设置为 AWS 托管应用程序

Redshift 管理员必须完成以下步骤，使 Redshift 成为 AWS IAM Identity Center 托管应用程序，然后 AWS IAM Identity Center 才能管理 Amazon Redshift 预置集群或 Amazon Redshift Serverless 工作组的身份：

1. 在 Amazon Redshift 或 Amazon Redshift Serverless 控制台菜单中选择 AWS IAM Identity Center 集成，然后选择连接到 AWS IAM Identity Center。在其中，您可以逐步完成一系列选择来填充 AWS IAM Identity Center 集成的属性。
2. 为 Redshift 的 AWS Identity Center 托管应用程序选择显示名称和唯一名称。
3. 为您的组织指定命名空间。这通常是组织名称的缩写。它已作为前缀添加到 Redshift 数据库中 AWS IAM Identity Center 管理的用户和角色。
4. 选择要使用的 IAM 角色。此 IAM 角色应与用于 Redshift 的其他角色分开，建议不要将该角色其用于其他目的。所需的特定策略权限如下所示：

   - `sso:DescribeApplication` – 需要在目录中创建身份提供者 (IdP) 条目。
   - `sso:DescribeInstance` – 用于手动创建 IdP 联合身份验证角色或用户。
5. 配置客户端连接和可信令牌发布者。配置可信令牌发布者，通过设置与外部身份提供者的关系来协调可信身份的传播。例如，通过身份传播，用户可以登录一个应用程序并访问另一个应用程序中的特定数据。这样用户便可以更无缝地从不同位置收集数据。在此步骤中，您可以在控制台中为每个可信令牌发布者设置属性。这些属性包括名称和受众群体声明（简写为 aud claim），您可能需要从工具或服务的配置属性中获取这些信息。您可能还需要从第三方工具的 JSON Web 令牌 (JWT, JSON Web Token) 提供应用程序名称。

> **Note**
>
> 每个第三方工具或服务要求的 `aud claim` 可能会有所不同，具体取决于令牌类型，这些可以是身份提供者发布的访问令牌，也可以是 ID 令牌等其他类型。每个供应商都可能不同。在实施可信身份传播并与 Redshift 集成时，您需要为第三方工具发送到 AWS 的令牌类型提供正确的 aud 值。请查看您的工具或服务供应商的建议。

有关可信身份传播的详细信息，请参阅《AWS IAM Identity Center 用户指南》中的 Trusted identity propagation overview。

Redshift 管理员完成步骤并保存配置后，AWS IAM Identity Center 属性将显示在 Redshift 控制台中。您也可以查询系统视图 SVV_IDENTITY_PROVIDERS 来验证应用程序的属性。这些属性包括应用程序名称和命名空间。您可以使用命名空间作为与应用程序关联的 Redshift 数据库对象的前缀。完成这些任务后，Redshift 便成为支持 AWS IAM Identity Center 的应用程序。控制台中的属性包括集成状态。集成完成后，它会显示已启用。完成此流程后，可以在每个新集群上启用 AWS IAM Identity Center 集成。

完成配置后，您可以在 Redshift 中使用 AWS IAM Identity Center 的用户和组，方法是选择用户或组选项卡，然后选择分配。

为 Amazon Redshift 集群或 Amazon Redshift Serverless 工作组启用 AWS IAM Identity Center 集成

您的数据库管理员配置新的 Redshift 资源，使其与 AWS IAM Identity Center 配合使用，从而简化登录和数据访问。此过程包括在创建预置集群或无服务器工作组的步骤中。任何有权创建 Redshift 资源的人员都可以执行这些 AWS IAM Identity Center 集成任务。当您创建预置集群时，请首先在 Amazon Redshift 控制台中选择创建集群。以下步骤显示如何为数据库启用 AWS IAM Identity Center 管理。（其中并未包括创建集群的所有步骤。）

1. 在创建集群步骤中，在 IAM Identity Center 集成部分中，选择为 <您的集群名称> 启用。
2. 在启用集成时，流程中有一个步骤。您可以在控制台中选择启用 IAM Identity Center 集成来完成此步骤。
3. 对于新的集群或工作组，使用 SQL 命令在 Redshift 中创建数据库角色。命令如下：

```
CREATE ROLE <idcnamespace:rolename>;
```

命名空间和角色名称如下：

- IAM Identity Center 命名空间前缀 – 这是您在设置 AWS IAM Identity Center 与 Redshift 之间的连接时，定义的命名空间。
- 角色名称 – 此 Redshift 数据库角色必须与 AWS IAM Identity Center 中的组名匹配。

Redshift 连接到 AWS IAM Identity Center，然后提取创建数据库角色并将其映射到 AWS IAM Identity Center 组所需的信息。

请注意，创建新的数据仓库时，为 AWS IAM Identity Center 集成指定的 IAM 角色将自动附加到预置集群或 Amazon Redshift Serverless 工作组。完成所需集群元数据的输入和资源创建后，您可以在属性中查看 AWS IAM Identity Center 集成的状态。如果您在 AWS IAM Identity Center 中的组名有空格，则在创建匹配角色时需要在 SQL 中使用引号。

启用 Redshift 数据库并创建角色后，就可以使用 Amazon Redshift 查询编辑器 v2 或 Amazon Quick 连接到数据库。详细信息将在后面的部分中进一步说明。

使用 API 设置默认 **`RedshiftIdcApplication`**

由您的身份管理员完成设置。使用 API，您可以创建并填充 `RedshiftIdcApplication`，它代表 AWS IAM Identity Center 中的 Redshift 应用程序。

1. 首先，您可以创建用户，并将这些用户添加到 AWS IAM Identity Center 的组中。您可以在 AWS 控制台中为 AWS IAM Identity Center 执行此操作。
2. 调用 `create-redshift-idc-application` 以创建 AWS IAM Identity Center 应用程序，使其与 Redshift 的使用兼容。您可以通过填充所需的值来创建应用程序。显示名称是在 AWS IAM Identity Center 控制面板上显示的名称。IAM 角色 ARN 是拥有 AWS IAM Identity Center 权限的 ARN，同样也可由 Redshift 代入。

```
aws redshift create-redshift-idc-application
--idc-instance-arn 'arn:aws:sso:::instance/ssoins-1234a01a1b12345d'
--identity-namespace 'MYCO'
--idc-display-name 'TEST-NEW-APPLICATION'
--iam-role-arn 'arn:aws:redshift:us-east-1:012345678901:role/TestRedshiftRole'
--redshift-idc-application-name 'myredshiftidcapplication'
```

以下示例显示了对 `create-redshift-idc-application` 的调用返回的示例 `RedshiftIdcApplication` 响应。

```
"RedshiftIdcApplication": {
                "IdcInstanceArn": "arn:aws:sso:::instance/ssoins-1234a01a1b12345d",
                "RedshiftIdcApplicationName": "test-application-1",
                "RedshiftIdcApplicationArn": "arn:aws:redshift:us-
east-1:012345678901:redshiftidcapplication:12aaa111-3ab2-3ab1-8e90-b2d72aea588b",
                "IdentityNamespace": "MYCO",
                "IdcDisplayName": "Redshift-Idc-Application",
                "IamRoleArn": "arn:aws:redshift:us-east-1:012345678901:role/
TestRedshiftRole",
                "IdcManagedApplicationArn": "arn:aws:sso::012345678901:application/
ssoins-1234a01a1b12345d/apl-12345678910",
                "IdcOnboardStatus": "arn:aws:redshift:us-
east-1:123461817589:redshiftidcapplication",
                "RedshiftIdcApplicationArn": "Completed",
                "AuthorizedTokenIssuerList": [
                       "TrustedTokenIssuerArn": ...,
                       "AuthorizedAudiencesList": [...]...
                ]}
```

3. 您可以使用 `create-application-assignment`，将特定组或个人用户分配给 AWS IAM Identity Center 中的托管应用程序。采用这种做法，您可以指定通过 AWS IAM Identity Center 管理的组。如果数据库管理员在 Redshift 中创建数据库角色，则 AWS IAM Identity Center 中的组名会映射到 Redshift 中的角色名称。数据库中的角色控制权限。有关更多信息，请参阅Assign user access to applications in the AWS IAM Identity Center console。
4. 启用应用程序后，调用 `create-cluster` 并附上 AWS IAM Identity Center 中的 Redshift 托管应用程序 ARN。通过这样做，可以将集群与 AWS IAM Identity Center 中的托管应用程序关联起来。

### 将 AWS IAM Identity Center 应用程序与现有集群或工作组关联

如果您要为现有的集群或工作组启用 AWS IAM Identity Center 集成，可以通过运行 SQL 命令来实现。您也可以运行 SQL 命令来更改集成的设置。有关更多信息，请参阅 ALTER IDENTITY PROVIDER。

还可以删除现有身份提供者。以下示例演示如何对附加到身份提供者的用户和角色执行级联删除。

```
DROP IDENTITY PROVIDER
<provider_name> [ CASCADE ]
```

设置用户权限

管理员根据用户的身份属性和组成员资格，在其身份提供者内部或直接在 AWS IAM Identity Center 中配置对各种资源的权限。例如，身份提供者管理员可以将数据库工程师添加到适合其角色的组中。此组名映射到 Redshift 数据库角色名称。该角色提供或限制对 Redshift 中特定表或视图的访问权限。

将 AWS IAM Identity Center 身份验证与增强型 VPC 路由结合使用

开启增强型 VPC 路由后，Redshift 会通过您的虚拟私有云（VPC）路由流量，而非通过互联网。这包括 Redshift 为对 AWS IAM Identity Center 用户进行身份验证而发出的调用。无论用户是以交互方式登录，还是使用通过可信身份传播获得的 AWS IAM Identity Center 令牌进行连接，均适用此规则。要使这些连接获得成功，Redshift 必须能够从您的 VPC 访问 AWS IAM Identity Center。接口 VPC 端点是提供这种连接的推荐方式。它们可让您的预调配集群或 Amazon Redshift Serverless 工作组通过 AWS 网络访问 AWS IAM Identity Center。

> **Important**
>
> 这些要求适用于任何通过 AWS IAM Identity Center 进行身份验证的连接。IAM 用户和本地数据库用户通过不同的机制进行身份验证，不受影响。如果关闭增强型 VPC 路由，则不需要这些端点。

> **Note**
>
> 更改增强型 VPC 路由会重新启动预调配集群。有关开启增强型 VPC 路由的更多信息，请参阅启用增强型 VPC 路由。

接口 VPC 端点

在您的集群或工作组所在的 VPC 中创建以下接口 VPC 端点（AWS PrivateLink）。这两个端点都是必需的。如果其中任一端点缺失或无法访问，AWS IAM Identity Center 登录将失败。

`com.amazonaws.region.sso-oauth`

验证用户在登录时出示的 AWS IAM Identity Center 访问令牌，并将其交换为限定于 Redshift 会话的令牌。

`com.amazonaws.region.identitystore`

从 AWS IAM Identity Center 身份存储中解析用户及其 AWS IAM Identity Center 组成员资格。

有关创建接口 VPC 端点的信息，请参阅《Amazon VPC 用户指南》中的创建 VPC 端点。

在创建端点之前，请确认为您的集群或工作组在其中运行的 VPC 开启 `DNS hostnames` 和 `DNS resolution` 属性。私有 DNS 名称取决于这两个属性。有关更多信息，请参阅《Amazon VPC 用户指南》中的 VPC 的 DNS 属性。

增强型 VPC 路由要求您的预调配集群或 Redshift Serverless 工作组不可公开访问。

创建每个端点时，请执行以下操作：

- 开启私有 DNS 名称：Redshift 使用这些服务的标准 DNS 名称连接到这些服务。如果没有私有 DNS 名称，这些名称将继续解析为公有地址，调用也不会路由到您的端点。
- 在 AWS IAM Identity Center 区域中创建端点：AWS IAM Identity Center 是一项区域性服务。在您的 AWS IAM Identity Center 实例运行所在的区域中创建这些端点。如果您的集群或工作组在其它区域运行，且您未使用 AWS IAM Identity Center 多区域复制，请使用跨区域端点选项，以便端点能够访问 AWS IAM Identity Center 区域。
- 选择相同可用区中的子网：在您的集群或工作组使用的可用区中选择子网。
- 支持入站 HTTPS：附加到每个端点的安全组必须支持 TCP 443 端口上来自您的集群或工作组子网的入站流量。
- 保留默认端点策略：默认策略支持完全访问。如果您的组织需要限制性端点策略，请按主体而非操作列表来限定其范围。然后，验证登录是否仍然成功。

### 使用 AWS Network Firewall

AWS Network Firewall 为可选的。如果您使用 AWS Network Firewall 而非使用接口 VPC 端点来筛选出站流量，则防火墙策略必须支持 AWS IAM Identity Center 服务域。否则，当 Redshift 无法访问这些域时，登录将失败。将以下域添加到防火墙的允许列表中，将 *region* 替换为您的 AWS IAM Identity Center 区域：

`oidc.`*region*`.amazonaws.com`

验证 AWS IAM Identity Center 访问令牌，并将其交换为限定于您的 Redshift 会话的令牌。

`.sso.`*region*`.amazonaws.com`

访问 AWS IAM Identity Center 服务，以解析 AWS IAM Identity Center 应用程序及其分配。

`identitystore.`*region*`.amazonaws.com`

解析用户及其 AWS IAM Identity Center 组成员资格。

如果您的防火墙使用默认拒绝策略，则还必须支持 Redshift 在增强型 VPC 路由下用于常规操作的域，例如 Amazon S3；如果您查询数据湖，则还必须支持 AWS Glue。有关更多信息，请参阅 Amazon Redshift 中的增强型 VPC 路由。

如接口 VPC 端点中所述，使用接口 VPC 端点可将此流量保持在 AWS 网络上，且无需将相关域加入防火墙允许列表。

自动为 AWS IAM Identity Center 创建 Amazon Redshift 角色

此特征与 AWS IAM Identity Center 集成，让您可以根据组成员资格在 Redshift 中自动创建角色。

自动创建角色有几个好处。自动创建角色时，Redshift 会在您的 IdP 中创建具有组成员资格的角色，因此您可以避免繁琐的手动角色创建和维护。您还可以选择使用包括和排除模式来筛选映射到 Redshift 角色的组。

工作原理

当您以 IdP 用户身份登录 Redshift 时，会发生以下一系列事件：

1. Redshift 从 IdP 那里检索您的组成员资格。
2. Redshift 使用角色格式 *idp_namespace*:*rolename* 自动创建映射到这些组的角色。
3. Redshift 授予您使用映射角色的权限。

每次用户登录时，都会自动创建目录中不存在但该用户所属的每个组。您可以选择设置包括和排除筛选条件，从而控制哪些 IdP 组已创建 Redshift 角色。

配置自动创建角色

使用 CREATE IDENTITY PROVIDER 和 ALTER IDENTITY PROVIDER 命令启用和配置自动角色创建。

```
-- Create a new IdP with auto role creation enabled
CREATE IDENTITY PROVIDER <idp_name> TYPE AWSIDC
  NAMESPACE '<namespace>'
  APPLICATION_ARN 'app_arn'
  IAM_ROLE 'role_arn'
  AUTO_CREATE_ROLES TRUE;

-- Enable on existing IdP
ALTER IDENTITY PROVIDER <idp_name>
```

```
  AUTO_CREATE_ROLES TRUE;

-- Disable
ALTER IDENTITY PROVIDER <idp_name>
  AUTO_CREATE_ROLES FALSE;
```

筛选组

您可以选择使用 INCLUDE 和 EXCLUDE 模式筛选映射到 Redshift 角色的 IdP 组。当模式冲突时，EXCLUDE 优先于 INCLUDE。

```
-- Only create roles for groups with 'dev'
CREATE IDENTITY PROVIDER <idp_name> TYPE AWSIDC
  ...
  AUTO_CREATE_ROLES TRUE
  INCLUDE GROUPS LIKE '%dev%';

-- Exclude 'test' groups
ALTER IDENTITY PROVIDER <idp_name>
  AUTO_CREATE_ROLES TRUE
  EXCLUDE GROUPS LIKE '%test%';
```

示例

以下示例说明了如何启用自动创建角色而不进行筛选。

```
CREATE IDENTITY PROVIDER prod_idc TYPE AWSIDC  ...
  AUTO_CREATE_ROLES TRUE;
```

以下示例包括开发组，不包括测试组。

```
ALTER IDENTITY PROVIDER prod_idc
  AUTO_CREATE_ROLES TRUE
  INCLUDE GROUPS LIKE '%dev%'
  EXCLUDE GROUPS LIKE '%test%';
```

最佳实践

为角色启用自动创建功能时，请考虑遵循以下最佳实践：

- 使用 INCLUDE 和 EXCLUDE 筛选条件来控制哪些组获得角色。

- 定期审核角色并清理未使用的角色。
- 利用 Redshift 角色层次结构来简化权限管理。

### Amazon Redshift 与 Amazon S3 访问权限管控的集成

通过与 Amazon S3 访问权限管控集成，可以无缝传播 IAM Identity Center 身份，来控制对 Amazon S3 数据的访问权限。此集成支持您根据 IAM Identity Center 用户和组授权 Amazon S3 数据访问权限。

有关 Amazon S3 访问权限管控的信息，请参阅使用 S3 访问权限管控管理访问权限。

使用 Amazon S3 访问权限管控可为应用程序带来以下优势：

- 基于 IAM Identity Center 身份，对 Amazon S3 数据进行精细的访问控制。
- 跨 Amazon Redshift 和 Amazon S3 集中管理 IAM Identity Center 身份。
- 您可以避免为 Amazon S3 访问权限管理单独的 IAM 权限。

### 工作原理

要将应用程序与 Amazon S3 访问权限管控集成，请执行以下操作：

- 首先，使用 AWS 管理控制台或 AWS CLI 将 Amazon Redshift 配置为与 Amazon S3 访问权限管控集成。
- 接下来，具有 IdC 管理员权限的用户使用 Amazon S3 访问权限管控服务，向特定 IdC 用户/组授予对 Amazon S3 存储桶或前缀的访问权限。有关更多信息，请参阅在 S3 访问权限管控中处理授权。
- 当经过 Redshift 身份验证的 IdC 用户运行访问 S3 的查询（如 COPY、UNLOAD 或 Spectrum 操作）时，Amazon Redshift 会从 Amazon S3 访问权限管控服务中检索该 IdC 身份范围内的临时 S3 访问凭证。
- 然后，Amazon Redshift 使用检索到的临时凭证，来访问该查询的经授权的 Amazon S3 位置。

### 设置与 Amazon S3 访问权限管控的集成

要为 Amazon Redshift 设置与 Amazon S3 访问权限管控的集成，请执行以下操作：

主题

- 使用 AWS 管理控制台设置与 Amazon S3 访问权限管控的集成

- 使用 AWS CLI 启用与 Amazon S3 访问权限管控的集成

使用 AWS 管理控制台设置与 Amazon S3 访问权限管控的集成

1. 打开 Amazon Redshift 控制台。
2. 从集群窗格中选择您的集群。
3. 在集群的详细信息页面的身份提供者集成部分，启用与 S3 访问权限管控服务的集成。

> **Note**
>
> 如果您未配置 IAM Identity Center，则不会显示身份提供者集成部分。有关更多信息，请参阅 Enabling AWS IAM Identity Center。

使用 AWS CLI 启用与 Amazon S3 访问权限管控的集成

1. 要创建一个启用了 S3 集成的新 Amazon Redshift IdC 应用程序，请执行以下操作：

```
aws redshift create-redshift-idc-application <other parameters>
  --service-integrations '[ {"S3AccessGrants": [{"ReadWriteAccess":
 {"Authorization": "Enabled"}}]} ]'
```

2. 要修改现有应用程序以启用 S3 访问权限管控集成，请执行以下操作：

```
aws redshift modify-redshift-idc-application <other parameters>
  --service-integrations '[ {"S3AccessGrants": [{"ReadWriteAccess":
 {"Authorization": "Enabled"}}]} ]'
```

3. 要修改现有应用程序以禁用 S3 访问权限管控集成，请执行以下操作：

```
aws redshift modify-redshift-idc-application <other parameters>
  --service-integrations '[ {"S3AccessGrants": [{"ReadWriteAccess":
 {"Authorization": "Disabled"}}]} ]'
```

使用与 S3 访问权限管控的集成

配置 S3 访问权限管控集成后，访问 S3 数据的查询（例如 COPY、UNLOAD 或 Spectrum 查询）将使用 IdC 身份进行授权。未使用 IdC 进行身份验证的用户也可以运行这些查询，但是这些用户账户无法利用 IdC 提供的集中式管理。

以下示例显示了通过 S3 访问权限管控集成运行的查询：

```
COPY table FROM 's3://mybucket/data';  // -- Redshift uses IdC identity
UNLOAD ('SELECT * FROM table') TO 's3://mybucket/unloaded/'    // -- Redshift uses IdC
 identity
```

## 通过 AWS Lake Formation 查询数据

使用 AWS Lake Formation 可以更轻松地集中管理和保护您的数据湖，并可用于提供数据访问。通过对 AWS IAM Identity Center 和 Redshift 进行配置，将身份传播到 Lake Formation，这样管理员就可以根据组织的身份提供者（IdP）组，实现对 Amazon S3 数据湖精细的访问控制。这些组通过 AWS IAM Identity Center 进行管理。此部分介绍如何配置几个应用场景，用于从数据湖中进行查询和从数据共享中进行查询，以演示如何将 AWS IAM Identity Center 与 Redshift 结合使用，来连接到由 Lake Formation 管理的资源。

使用 AWS IAM Identity Center 和 Redshift 连接来查询数据湖

这些步骤涵盖的应用场景是，您使用 AWS IAM Identity Center 连接到 Redshift，以便查询 Lake Formation 管理的数据湖。

先决条件：

此过程有多个先决条件步骤：

1. AWS IAM Identity Center 必须设置为支持 Redshift 的身份验证和身份管理。您可以从控制台启用 AWS IAM Identity Center 并选择身份提供者（IdP）来源。之后，将您的一组 IdP 用户与 AWS IAM Identity Center 同步。您还必须按照本文档前面详述的步骤，在 AWS IAM Identity Center 与 Redshift 之间建立连接。
2. 创建新的 Amazon Redshift 集群，并在配置步骤中启用通过 AWS IAM Identity Center 进行身份管理。
3. 为 Lake Formation 创建托管 AWS IAM Identity Center 应用程序并对其进行配置。此配置应在设置了 AWS IAM Identity Center 与 Redshift 之间的连接之后进行。步骤如下：
   a. 在 AWS CLI 中，使用 `modify-redshift-idc-application` 命令启用 Lake Formation 服务与 AWS IAM Identity Center 托管的 Redshift 应用程序的集成。此调用包括 `service-integrations` 参数，该参数设置为启用对 Lake Formation 进行授权的配置字符串值。
   b. 使用 `create-lake-formation-identity-center-configuration` 命令配置 Lake Formation。这将创建一个适用于 Lake Formation 的 AWS IAM Identity Center 应用程序，该应用程序在 AWS IAM Identity Center 门户中可见。管理员必须设置 `--cli-input-json` 参数，

其值是 JSON 文件的路径，该文件使用所有 AWS CLI API 调用的标准格式。您必须包括以下各项的值：

- `CatalogId` – Lake Formation 目录 ID。
- `InstanceArn` – AWS IAM Identity Center 实例 ARN 值。

管理员完成先决条件配置后，数据库管理员可以创建用于查询数据湖的外部架构。

1. 管理员创建外部架构 – Redshift 数据库管理员使用以下 SQL 语句连接到数据库并创建外部架构：

```
CREATE EXTERNAL SCHEMA if not exists my_external_schema from DATA CATALOG database
 'my_lf_integrated_db' catalog_id '12345678901234';
```

   请注意，在这种情况下不需要指定 IAM 角色，因为访问权限是通过 AWS IAM Identity Center 管理的。

2. 管理员授予权限 – 管理员向 AWS IAM Identity Center 组授予使用权限，这会授予对 Redshift 资源的权限。此步骤通过运行如下所示的 SQL 语句完成：

```
GRANT USAGE ON SCHEMA "my_external_schema" to "MYCO:sales";
```

   随后，管理员根据组织的要求，使用 AWS CLI 授予 Lake Formation 在对象上的权限：

```
aws lakeformation grant-permissions ...
```

3. 用户运行查询 – 举例说明，此时销售人员组中的 AWS IAM Identity Center 用户，可以通过查询编辑器 v2 登录到 Redshift 数据库。然后，他们可以运行访问外部架构中的表的查询，如以下示例：

```
SELECT * from my_external_schema.table1;
```

使用 AWS IAM Identity Center 和 Redshift 连接来连接到数据共享

通过 AWS IAM Identity Center 管理访问权限时，您可以从不同的 Redshift 数据仓库访问数据共享。为此，您需要运行查询来设置外部数据库。完成这些步骤的要求是，假定您已在 Redshift 和 AWS IAM Identity Center 之间建立连接，并且您已经创建了 AWS Lake Formation 应用程序，如前面的过程所述。

1. 创建外部数据库 – 管理员创建外部数据库用于数据共享，并通过其 ARN 进行引用。以下是演示如何操作的示例：

```
CREATE DATABASE "redshift_external_db" FROM ARN 'arn:aws:glue:us-
east-1:123456789012:database/redshift_external_db-iad' WITH NO DATA CATALOG SCHEMA;
```

在本使用场景中，您将 AWS IAM Identity Center 与 Redshift 用于身份管理，但不包括 IAM 角色。

2. 管理员设置权限 – 创建数据库后，管理员向 AWS IAM Identity Center 组授予使用权限。这将授予对 Redshift 资源的权限：

```
GRANT USAGE ON DATABASE "my_external_db" to "MYCO:sales";
```

管理员还使用 AWS CLI 授予 Lake Formation 在对象上的权限：

```
aws lakeformation grant-permissions ...
```

3. 用户运行查询 – 销售组中的用户可以基于分配的权限查询数据库中的表：

```
select * from redshift_external_db.public.employees;
```

有关授予数据湖权限和授予数据共享权限的更多信息，请参阅 Granting permissions to users and groups。有关向架构或数据库授予使用权限的更多信息，请参阅 GRANT。

## 使用可信令牌发布者将应用程序或工具与 OAuth 集成

您可以向所创建的客户端工具中添加功能，以通过 AWS IAM Identity Center 连接来连接到 Redshift。如果您已经配置了 Redshift 与 AWS IAM Identity Center 的集成，请使用此部分中详述的属性来设置连接。

用于使用 AWS IAM Identity Center 连接到 Redshift 的身份验证插件

您可以使用以下驱动程序插件，通过 AWS IAM Identity Center 连接到 Amazon Redshift：

- `BrowserIdcAuthPlugin` – 此插件促进实现与 AWS IAM Identity Center 的无缝单点登录集成。插件会创建一个浏览器窗口，供用户使用其企业身份提供者中定义的用户凭证进行登录。
- `IdpTokenAuthPlugin` – 此插件应由想要自行管理身份验证流程，而不是让 Amazon Redshift 驱动程序打开浏览器窗口进行 AWS IAM Identity Center 身份验证的应用程序使用。插件接受 AWS IAM Identity Center 提供的访问令牌，或 OpenID Connect（OIDC）JSON Web 令牌（JWT），该令牌由与 AWS IAM Identity Center 关联的任意 Web 身份提供者提供，例如 Okta、PingOne 和 Microsoft Entra ID（Azure AD）。客户端应用程序负责生成此必需的访问令牌/JWT。

### 使用 **BrowserIdcAuthPlugin** 进行身份验证

根据您的 Amazon Redshift 驱动程序，使用以下插件名称，通过 BrowserIdcAuthPlugin 进行连接。

| 驱动程序 | 连接选项键 | 值 | 备注 |
| --- | --- | --- | --- |
| JDBC | plugin_name | com.amazon.redshift.plugin.BrowserIdcAuthPlugin | 在连接时，您必须输入插件的完全限定类名。 |
| ODBC | plugin_name | BrowserIdcAuthPlugin | |
| Python | credentials_provider | BrowserIdcAuthPlugin | Python 驱动程序没有 plugin_name 选项可用。请改用credentials_provider 。 |

BrowserIdcAuthPlugin 插件具有以下额外的连接选项：

| 选项名称 | 必填？ | 说明 | 示例 |
| --- | --- | --- | --- |
| idc_region | 必填 | AWS IAM Identity Center 实例所在的 AWS 区域。 | us-east-1 |
| issuer_url | 必填 | AWS IAM Identity Center 服务器的实例端点。您可以使用 AWS IAM Identity Center 控制台查找此值。 | https://identitycenter.amazonaws.com/ssoins-g5j2k70sn4yc5nsc |
| listen_port | 可选 | Amazon Redshift 驱动程序用来通过浏览器重定向接收来自 AWS IAM Identity Center 的 | 7890 |

| 选项名称 | 必填？ | 说明 | 示例 |
| --- | --- | --- | --- |
| | | auth_code 响应的端口。 | |
| idc_client_display_name | 可选 | 在 AWS IAM Identity Center 的单点登录同意弹出窗口中，AWS IAM Identity Center 客户端为应用程序使用的名称。 | Amazon Redshift 驱动程序 |
| idp_response_timeout | 可选 | Redshift 驱动程序等待身份验证流程完成的时间，以秒为单位。 | 60 |

您必须输入自己创建并用于连接的工具的这些连接属性值。有关更多信息，请参阅相应的各个驱动程序的连接选项文档：

- JDBC 驱动程序版本 2.x 配置的选项
- ODBC 驱动程序选项
- Amazon Redshift Python 连接器的配置选项

使用 **IdpTokenAuthPlugin** 进行身份验证

根据您的 Amazon Redshift 驱动程序，使用以下插件名称，通过 IdpTokenAuthPlugin 进行连接。

| 驱动程序 | 连接选项键 | 值 | 备注 |
| --- | --- | --- | --- |
| JDBC | plugin_name | com.amazon.redshift.plugin.IdpTokenAuthPlugin | 在连接时，您必须输入插件的完全限定类名。 |
| ODBC | plugin_name | IdpTokenAuthPlugin | |
| Python | credentials_provider | IdpTokenAuthPlugin | Python 驱动程序没有 plugin_na |

| 驱动程序 | 连接选项键 | 值 | 备注 |
| --- | --- | --- | --- |
| | | | me 选项可用。请改用credentials_provider 。 |

IdpTokenAuthPlugin 插件具有以下额外的连接选项：

| 选项名称 | 必填？ | 说明 |
| --- | --- | --- |
| 令牌 | 必填 | AWS IAM Identity Center 提供的访问令牌，或 OpenID Connect（OIDC）JSON Web 令牌（JWT，JSON Web Token），该令牌由与 AWS IAM Identity Center 连接的 Web 身份提供者提供。您的应用程序必须通过 AWS IAM Identity Center 或与 AWS IAM Identity Center 连接的身份提供者，对您的应用程序用户进行身份验证，从而生成此令牌。 |
| token_type | 必填 | 用于 IdpTokenAuthPlugin 的令牌类型。可能的值包括：<br>• ACCESS_TOKEN – 如果您使用 AWS IAM Identity Center 提供的访问令牌，请输入此项。<br>• EXT_JWT – 如果您使用 OpenID Connect（OIDC）JSON Web 令牌（JWT，JSON Web Token）（该令牌由与 AWS |

| 选项名称 | 必填？ | 说明 |
| --- | --- | --- |
| | | IAM Identity Center 关联的基于 Web 的身份提供者提供），请输入此项。 |

您必须输入自己创建并用于连接的工具的这些连接属性值。有关更多信息，请参阅相应的各个驱动程序的连接选项文档：

- JDBC 驱动程序版本 2.x 配置的选项
- ODBC 驱动程序选项
- Amazon Redshift Python 连接器的配置选项

## 排除 Amazon Redshift 查询编辑器 v2 连接故障

此列表详细列出了经常出现的错误，有助于您使用 AWS IAM Identity Center 身份通过查询编辑器 v2 连接到 Redshift 数据库。

- 错误：连接问题：没有可用的身份中心会话信息。– 出现此错误时，请检查浏览器的安全和隐私设置。这些浏览器设置，尤其是安全 Cookie 设置（如 Firefox 的全面 Cookie 保护功能），可能会导致 Amazon Redshift 查询编辑器 v2 与 Redshift 数据库的连接尝试受阻。请按照您浏览器的详细修复步骤进行修复：
  - Firefox – 目前，默认情况下会阻止第三方 Cookie。单击浏览器地址栏中的盾牌，然后切换开关以关闭针对查询编辑器 v2 的增强跟踪保护。
  - Chrome 隐身模式 – 默认情况下，Chrome 隐身模式会阻止第三方 Cookie。单击地址栏中的眼睛图标以允许查询编辑器 v2 使用第三方 Cookie。将设置更改为允许 Cookie 后，您可能在地址栏上看不到眼睛图标。
  - Safari – 在 Mac 上，打开 Safari 应用程序。选择设置，然后选择高级。切换为关闭：阻止所有 Cookie。
  - Edge – 选择设置，然后选择 Cookie 和站点权限。然后选择管理和删除 Cookie 和站点数据，并关闭阻止第三方 Cookie。

  如果您在更改设置后尝试连接，但继续收到错误消息连接问题：没有可用的身份中心会话信息，我们建议您刷新与 AWS IAM Identity Center 的连接。为此，请右键单击您的 Redshift 数据库实例并选择刷新。此时将出现一个新窗口，您可以使用该窗口进行身份验证。

- 错误：连接问题：身份中心会话已过期或无效。– 将 Redshift 预置集群或 Serverless 工作组与 AWS IAM Identity Center 集成后，用户在尝试从查询编辑器 v2 连接到 Redshift 数据库时可能会收到此错误。这可能是在成功连接尝试之后发生的。在这种情况下，我们建议您重新进行身份验证。为此，请右键单击您的 Redshift 数据库实例并选择刷新。此时将出现一个新窗口，您可以使用该窗口进行身份验证。
- 错误：范围无效。用户凭证无权连接到 Redshift。– 将 Redshift 预置集群或 Serverless 工作组与 AWS IAM Identity Center 集成用于身份管理后，用户在尝试从查询编辑器 v2 连接到 Redshift 数据库时可能会收到此错误。在这种情况下，为了使查询编辑器 v2 成功连接并通过 AWS IAM Identity Center 对用户进行身份验证，以便访问正确的资源，管理员必须通过 Redshift 控制台将用户分配给 Redshift AWS IAM Identity Center 应用程序。这是在 IAM Identity Center 连接下完成的。完成这些设置后，用户可以在一个小时之后成功建立连接，这段时间是 AWS IAM Identity Center 会话缓存造成的限制。
- 错误：无法列出数据库。致命：集群自动暂停时查询失败。– 当 Amazon Redshift Serverless 数据库处于空闲状态，未处理任何工作负载时，如果您使用 AWS IAM Identity Center 身份进行连接，它可能会保持暂停状态。要解决此问题，请使用其它身份验证方法登录来恢复无服务器工作组。然后使用您的 AWS IAM Identity Center 身份连接到该数据库。
- 错误：尝试与 AWS IAM Identity Center 联合时出错。Amazon Redshift 管理员必须使用 Redshift 控制台删除并重新创建 AWS IAM Identity Center QEV2 应用程序。– 此错误通常发生在删除与查询编辑器 v2 相关联的 AWS IAM Identity Center 应用程序实例时。要解决这个问题，Amazon Redshift 管理员必须为 AWS IAM Identity Center 删除并重新创建 Redshift 和查询编辑器 v2 应用程序。可以在 Redshift 控制台上或使用 https://docs.aws.amazon.com/cli/latest/reference/redshift/delete-redshift-idc-application.html CLI 命令执行此操作。

## 对 Amazon Redshift 使用服务相关角色

Amazon Redshift 使用 AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种独特类型的 IAM 角色，它与 Amazon Redshift 直接相关。服务相关角色由 Amazon Redshift 预定义，具有服务代表您的 Amazon Redshift 集群调用 AWS 服务所需的所有权限。

服务相关角色可让您更轻松地设置 Amazon Redshift，因为您不必手动添加必要权限。该角色与 Amazon Redshift 使用案例相关联并且具有预定义的权限。只有 Amazon Redshift 可以代入该角色，并且只有服务相关角色可以使用预定义的权限策略。Amazon Redshift 会在您首次创建集群或 Redshift 托管的 VPC 端点时，在您的账户中创建服务相关角色。只有删除您账户中的所有 Amazon Redshift 集群或 Redshift 托管的 VPC 端点之后，您才可以删除服务相关角色。这将保护您的 Amazon Redshift 资源，因为您不会无意中删除访问资源所需的权限。

Amazon Redshift 支持在该服务可用的所有区域中使用服务相关角色。有关更多信息，请参阅 AWS 区域和终端节点。

有关支持服务相关角色的其它服务的信息，请参阅使用 IAM 的 AWS 服务，并查找服务相关角色列中显示为是的服务。选择是和链接，查看该服务的服务相关角色文档。

## Amazon Redshift 的服务相关角色权限

Amazon Redshift 使用名为 AWSServiceRoleForRedshift 的服务相关角色 – 允许 Amazon Redshift 代表您调用 AWS 服务。此服务相关角色附加至以下托管式策略上：AmazonRedshiftServiceLinkedRolePolicy。有关此策略的更新，请参阅适用于 Amazon Redshift 的 AWS 托管式策略（预定义）。

AWSServiceRoleForRedshift 服务相关角色仅信任 **redshift.amazonaws.com** 来代入该角色。

AWSServiceRoleForRedshift 服务相关角色权限策略允许 Amazon Redshift 对所有相关资源完成以下操作：

- ec2:DescribeVpcs
- ec2:DescribeSubnets
- ec2:DescribeNetworkInterfaces
- ec2:DescribeAddress
- ec2:AssociateAddress
- ec2:DisassociateAddress
- ec2:CreateNetworkInterface
- ec2:DeleteNetworkInterface
- ec2:ModifyNetworkInterfaceAttribute
- ec2:CreateVpcEndpoint
- ec2:DeleteVpcEndpoints
- ec2:DescribeVpcEndpoints
- ec2:ModifyVpcEndpoint
- ec2:DescribeVpcAttribute
- ec2:DescribeSecurityGroups
- ec2:DescribeInternetGateways

- ec2:DescribeSecurityGroupRules
- ec2:DescribeAvailabilityZones
- ec2:DescribeNetworkAcls
- ec2:DescribeRouteTables
- ec2:AssignIpv6Addresses
- ec2:UnassignIpv6Addresses

对网络资源的权限

以下权限允许在 Amazon EC2 上执行操作来创建和管理安全组规则。这些安全组和规则特别关联到 Amazon Redshift aws:RequestTag/Redshift 资源标签。这样可以将权限的范围限制为特定 Amazon Redshift 资源。

- ec2:CreateSecurityGroup
- ec2:AuthorizeSecurityGroupEgress
- ec2:AuthorizeSecurityGroupIngress
- ec2:RevokeSecurityGroupEgress
- ec2:RevokeSecurityGroupIngress
- ec2:ModifySecurityGroupRules
- ec2:DeleteSecurityGroup

服务配额权限

以下权限支持调用方获得服务配额。

servicequotas:GetServiceQuota

以下 JSON 代码段显示了服务配额的操作和资源范围。

```
{
   "Sid": "ServiceQuotasToCheckCustomerLimits",
   "Effect": "Allow",
   "Action": [
      "servicequotas:GetServiceQuota"
   ],
```

```
   "Resource": [
      "arn:aws:servicequotas:*:*:ec2/L-0263D0A3",
      "arn:aws:servicequotas:*:*:vpc/L-29B6F2EB"
   ]
}
```

配额代码如下：

- L-0263D0A3 – EC2-VPC 弹性 IP 的配额代码。
- L-29B6F2EB – 每个 VPC 的接口 VPC 端点的配额代码。

有关更多信息，请参阅 AWS 服务配额。

审计日志记录的操作

列出的带 `logs` 前缀的操作与审计日志记录和相关功能有关。具体来说是创建和管理日志组和日志流的操作。

- `logs:CreateLogGroup`
- `logs:PutRetentionPolicy`
- `logs:CreateLogStream`
- `logs:PutLogEvents`
- `logs:DescribeLogStreams`
- `logs:GetLogEvents`

以下 JSON 向 Amazon Redshift 显示了用于审计日志记录的操作和资源范围。

```
[
    {
        "Sid": "EnableCreationAndManagementOfRedshiftCloudwatchLogGroups",
        "Effect": "Allow",
        "Action": [
            "logs:CreateLogGroup",
            "logs:PutRetentionPolicy"
        ],
        "Resource": [
            "arn:aws:logs:*:*:log-group:/aws/redshift/*"
        ]
```

```
    },
    {
        "Sid": "EnableCreationAndManagementOfRedshiftCloudwatchLogStreams",
        "Effect": "Allow",
        "Action": [
            "logs:CreateLogStream",
            "logs:PutLogEvents",
            "logs:DescribeLogStreams",
            "logs:GetLogEvents"
        ],
        "Resource": [
            "arn:aws:logs:*:*:log-group:/aws/redshift/*:log-stream:*"
        ]
    }
]
```

有关服务相关角色及它们在 AWS 中的作用的更多信息，请参阅使用服务相关角色。有关 Amazon Redshift 的特定操作和其它 IAM 资源的更多信息，请参阅 Amazon Redshift 的操作、资源和条件键。

使用 AWS Secrets Manager 管理管理员凭证的操作

所列出的带有 `secretsmanager` 前缀的操作与使用 Amazon Redshift 管理您的管理员凭证有关。Amazon Redshift 通过这些操作，使用 AWS Secrets Manager 来创建和管理您的管理员凭证密钥。

以下 JSON 显示了操作和资源范围，可供 Amazon Redshift 通过 AWS Secrets Manager 管理管理员凭证。

```
[
    {
        "Effect": "Allow",
        "Action": [
            "secretsmanager:DescribeSecret",
            "secretsmanager:DeleteSecret",
            "secretsmanager:PutSecretValue",
            "secretsmanager:UpdateSecret",
            "secretsmanager:UpdateSecretVersionStage",
            "secretsmanager:RotateSecret"
        ],
        "Resource": [
            "arn:aws:secretsmanager:*:*:secret:redshift!*"
        ],
        "Condition": {
```

```
            "StringEquals": {
                "secretsmanager:ResourceTag/aws:secretsmanager:owningService":
 "redshift"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "secretsmanager:GetRandomPassword"
        ],
        "Resource": "*"
    }
]
```

用于将集群和无服务器命名空间注册到 AWS Glue Data Catalog 的操作

以 glue 前缀列出的操作涉及访问 AWS Glue Data Catalog 中的目录，这些目录是通过注册预置集群或无服务器命名空间创建的。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Apache Iceberg compatibility for Amazon Redshift。

以下 JSON 显示了 Amazon Redshift 用于访问 AWS Glue Data Catalog 中目录的操作和资源范围：

```
[
    {
        "Sid": "DiscoverRedshiftCatalogs",
        "Effect": "Allow",
        "Action": [
            "glue:GetCatalogs",
            "glue:GetCatalog"
        ],
        "Resource": [
            "arn:aws:glue:*:*:catalog",
            "arn:aws:glue:*:*:catalog/*"
        ],
   "Condition":
    {
        "Bool":
        {
            "glue:EnabledForRedshiftAutoDiscovery": "true"
        },
        "StringEquals": {
             "aws:ResourceAccount": "${aws:PrincipalAccount}"
```

```
        }
    }
 },
 {
    "Sid": "LakeFormationGetMetadataAccessForFederatedCatalogs",
    "Effect": "Allow",
    "Action": [
        "lakeformation:GetDataAccess"
    ],
    "Resource": [ "*" ],
    "Condition":
    {
        "Bool":
        {
            "lakeformation:EnabledOnlyForMetaDataAccess":"true"
        },
        "StringEquals": {
             "aws:ResourceAccount": "${aws:PrincipalAccount}"
        },
        "ForAnyValue:StringEquals":
        {
            "aws:CalledVia": "glue.amazonaws.com"
        }
    }
 }
    }
]
```

glue:GetCatalog 和 glue:GetCatalogs 权限具有条件 glue:EnabledForRedshiftAutoDiscovery:true，这意味着 Amazon Redshift 向 IAM 授予自动发现目录的访问权限。要选择退出，请添加 AWS Glue 账户级资源策略，以便选择性地拒绝服务相关角色访问目录。由于服务相关角色在策略中已经有显式允许操作，因此选择退出策略需要显式拒绝该操作。考虑下面的例子，其中一个额外的策略拒绝 Amazon Redshift 的自动发现功能：

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : {
            "Effect": "Deny",
            "Action": [
                "glue:GetCatalog",
```

```
                "glue:GetCatalogs"
            ],
            "Principal" : {
            "AWS" : "arn:aws:iam::111122223333:role/aws-service-role/
redshift.amazonaws.com/AWSServiceRoleForRedshift"
            },
            "Resource": [
                "arn:aws:glue:*:*:catalog/<s3_table_catalog_name>",
                "arn:aws:glue:*:*:catalog/<s3_table_catalog_name>/*"
            ]
        }
}
```

要允许 IAM 实体创建 AWSServiceRoleForRedshift 服务相关角色

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole"
    ],
    "Resource": "arn:aws:iam::<AWS-account-ID>:role/aws-service-role/
redshift.amazonaws.com/AWSServiceRoleForRedshift",
    "Condition": {"StringLike": {"iam:AWSServiceName": "redshift.amazonaws.com"}}
}
```

要允许 IAM 实体删除 AWSServiceRoleForRedshift 服务相关角色

向该 IAM 实体的权限中添加以下策略声明：

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::<AWS-account-ID>:role/aws-service-role/
redshift.amazonaws.com/AWSServiceRoleForRedshift",
    "Condition": {"StringLike": {"iam:AWSServiceName": "redshift.amazonaws.com"}}
}
```

或者，您可以使用 AWS 托管式策略提供对 Amazon Redshift 的完全访问权限。

### 为 Amazon Redshift 创建服务相关角色

您无需手动创建 AWSServiceRoleForRedshift 服务相关角色。Amazon Redshift 将为您创建服务相关角色。如果已从您的账户中删除 AWSServiceRoleForRedshift 服务相关角色，Amazon Redshift 将在您启动新 Amazon Redshift 集群时创建该角色。

> **Important**
>
> 如果您在 2017 年 9 月 18 日（从此时开始支持服务相关角色）之前已使用 Amazon Redshift 服务，则 Amazon Redshift 已在您的账户中创建 AWSServiceRoleForRedshift 角色。要了解更多信息，请参阅我的 IAM 账户中出现新角色。

### 为 Amazon Redshift 编辑服务相关角色

Amazon Redshift 不允许您编辑 AWSServiceRoleForRedshift 服务相关角色。在创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。不过，您可以使用 IAM 控制台、AWS Command Line Interface (AWS CLI) 或 IAM API 编辑角色描述。有关更多信息，请参阅 IAM 用户指南中的修改角色。

### 删除 Amazon Redshift 的服务相关角色

如果您不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样您就没有未被主动监控或维护的未使用实体。

在删除账户的服务相关角色之前，您需要关闭并删除该账户中的所有集群。有关更多信息，请参阅 关闭和删除集群。

您可以使用 IAM 控制台、AWS CLI 或 IAM API 删除服务相关角色。有关更多信息，请参阅 IAM 用户指南中的删除服务相关角色。

## 使用 IAM 身份验证生成数据库用户凭证

您可以根据通过 AWS Identity and Access Management (IAM) 权限策略授予的权限生成临时数据库凭证，以管理用户的 Amazon Redshift 数据库访问权限。

通常，Amazon Redshift 数据库用户提供数据库用户名和密码以登录到数据库。不过，您不必在 Amazon Redshift 数据库中维护用户名和密码。作为替代方法，您可以将系统配置为允许用户创建用户凭证，并根据其 IAM 凭证登录到数据库。

Amazon Redshift 提供 GetClusterCredentials API 操作以生成临时数据库用户凭证。您可使用 Amazon Redshift JDBC 或 ODBC 驱动程序来配置您的 SQL 客户端，这些驱动程序可管理调用 GetClusterCredentials 操作的过程。它们通过检索数据库用户凭证并在您的 SQL 客户端和您的 Amazon Redshift 数据库之间建立连接来完成此操作。您也可以使用数据库应用程序以编程方式调用 GetClusterCredentials 操作，检索数据库用户凭证并连接到数据库。

如果您已在 AWS 外部管理用户身份，您可以使用与安全断言标记语言 (SAML) 2.0 兼容的身份提供者 (IdP) 管理对 Amazon Redshift 资源的访问。配置您的 IdP 以允许联合身份用户访问 IAM 角色。通过使用该 IAM 角色，您可以生成临时数据库凭证并登录到 Amazon Redshift 数据库。

SQL 客户端需要具有权限才能为您调用 GetClusterCredentials 操作。您将通过创建 IAM 角色并附加 IAM 权限策略（该策略授权或限制对 GetClusterCredentials 操作和相关操作的访问）来管理这些权限。作为最佳实践，我们建议将权限策略附加到 IAM 角色，然后根据需要将其分配给用户和组。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

该策略还授权或限制对特定资源 (例如，Amazon Redshift 集群、数据库、数据库用户名和用户组名称) 的访问。

> **Note**
>
> 我们建议使用 Amazon Redshift JDBC 或 ODBC 驱动程序来管理调用 GetClusterCredentials 操作并登录数据库的过程。为简单起见，我们在整个主题中假定您将 SQL 客户端与 JDBC 或 ODBC 驱动程序结合使用。
> 有关使用 GetClusterCredentials 操作或并行 get-cluster-credentials CLI 命令的具体详细信息和示例，请参阅 GetClusterCredentials 和 get-cluster-credentials。

为了集中管理身份验证和授权，Amazon Redshift 支持使用 IAM 进行数据库身份验证，从而通过企业联合进行用户身份验证。您也可以不创建用户，而是使用来自 AWS Directory Service、您的企业用户目录或 Web 身份提供者的现有身份。这些用户被称为联合身份用户。在通过 IdP 请求访问权限时，AWS 将为联合身份用户分配角色。

要为组织中的用户或客户端应用程序提供联合访问权限以调用 Amazon Redshift API 操作，您还可以使用具有 SAML 2.0 支持的 JDBC 或 ODBC 驱动程序请求从组织 IdP 中进行身份验证。在这种情况下，组织的用户没有 Amazon Redshift 的直接访问权限。

有关更多信息，请参阅 IAM 用户指南中的身份提供者和联合。

## 创建临时 IAM 凭证

在本节中，您可以了解如何配置系统以生成基于 IAM 的临时数据库用户凭证，并使用新凭证登录到数据库。

概括来说，流程如下所示：

1. 步骤 1：创建用于 IAM 单点登录访问的 IAM 角色

   （可选）您可以将 IAM 身份验证和第三方身份提供者 (IdP) 进行集成，对访问 Amazon Redshift 数据库的用户进行身份验证。

2. 步骤 2：为 IdP 配置 SAML 断言

   （可选）要使用 IdP 进行 IAM 身份验证，您需要在 IdP 应用程序中定义一个声明规则，将组织中的用户或组映射到 IAM 角色。或者，您可以包含属性元素以设置 `GetClusterCredentials` 参数。

3. 步骤 3：创建有权调用 GetClusterCredentialsWithIAM 或 GetClusterCredentials 的 IAM 角色

   您的 SQL 客户端应用程序在调用 `GetClusterCredentials` 操作时代入用户。如果您创建了用于身份提供者访问的 IAM 角色，则可向该角色添加必要权限。

4. 步骤 4：创建数据库用户和数据库组

   （可选）默认情况下，`GetClusterCredentials` 返回凭证；如果用户名不存在，则会创建一个新用户。您也可以选择指定用户在登录时加入的用户组。默认情况下，数据库用户加入 PUBLIC 组。

5. 步骤 5：配置 JDBC 或 ODBC 连接以使用 IAM 凭证

   要连接到您的 Amazon Redshift 数据库，可将 SQL 客户端配置为使用 Amazon Redshift JDBC 或 ODBC 驱动程序。

### 步骤 1：创建用于 IAM 单点登录访问的 IAM 角色

如果您未使用身份提供者进行单点登录访问，则可跳过此步骤。

如果您已在 AWS 外部管理用户身份，则可将 IAM 身份验证和第三方 SAML-2.0 身份提供者 (IdP) 集成，对访问 Amazon Redshift 数据库的用户进行身份验证。

有关更多信息，请参阅 IAM 用户指南中的身份提供者和联合。

在使用 Amazon Redshift IdP 身份验证之前，请先创建一个 AWS SAML 身份提供者。您可以在 IAM 控制台中创建一个 IdP，以向 AWS 通知该 IdP 及其配置。这样将在您的 AWS 账户和 IdP 之间建立信任。有关创建角色的步骤，请参阅《IAM 用户指南》中的创建用于 SAML 2.0 联合的角色（控制台）。

步骤 2：为 IdP 配置 SAML 断言

在创建 IAM 角色后，您可以在 IdP 应用程序中定义一个声明规则，以将组织中的用户或组映射到 IAM 角色。有关更多信息，请参阅 IAM 用户指南中的为身份验证响应配置 SAML 断言。

如果选择使用可选的 `GetClusterCredentials` 参数 `DbUser`、`AutoCreate` 和 `DbGroups`，您有两个选择。您可以使用 JDBC 或 ODBC 连接设置这些参数的值，也可以将 SAML 属性元素添加到 IdP 以设置这些值。有关 `DbUser`、`AutoCreate` 和 `DbGroups` 参数的更多信息，请参阅步骤 5：配置 JDBC 或 ODBC 连接以使用 IAM 凭证。

> **Note**
>
> 如果您使用 IAM 策略变量 `${redshift:DbUser}`（如GetClusterCredentials 的资源策略中所述），则 `DbUser` 的值将被替换为由 API 操作的请求上下文检索的值。Amazon Redshift 驱动程序使用由连接 URL 提供的 `DbUser` 变量的值，而不是作为 SAML 属性提供的值。
> 为了帮助保护该配置，我们建议您在 IAM 策略中使用一个条件以通过 `DbUser` 验证 `RoleSessionName` 值。您可以在 示例 8：使用 GetClusterCredentials 的 IAM 策略 中找到如何使用 IAM 策略设置条件。

要配置 IdP 以设置 `DbUser`、`AutoCreate` 和 `DbGroups` 参数，请包含以下 `Attribute` 元素：

- `Attribute` 属性设置为“https://redshift.amazon.com/SAML/Attributes/DbUser”的 `Name` 元素

  将 `AttributeValue` 元素设置为要连接到 Amazon Redshift 数据库的用户的名称。

  `AttributeValue` 元素中的值必须为小写形式，以字母开头，仅包含字母数字字符、下划线 (_)、加号 (+)、圆点 (.)、@ 符号或连字符 (-)，并且应少于 128 个字符。通常，用户名为用户 ID (例如，bobsmith) 或电子邮件地址 (例如，bobsmith@example.com)。此值不能包含空格 (例如，像 Bob Smith 这样的用户显示名称)。

  ```
  <Attribute Name="https://redshift.amazon.com/SAML/Attributes/DbUser">
      <AttributeValue>user-name</AttributeValue>
  </Attribute>
  ```

- Name 属性设置为“https://redshift.amazon.com/SAML/Attributes/AutoCreate”的 Attribute 元素

可以将 AttributeValue 元素设置为 true 以创建新的数据库用户（如果不存在）。将 AttributeValue 设置为 false 可指定数据库用户必须存在于 Amazon Redshift 数据库中。

```
<Attribute Name="https://redshift.amazon.com/SAML/Attributes/AutoCreate">
    <AttributeValue>true</AttributeValue>
</Attribute>
```

- `Attribute` 属性设置为“https://redshift.amazon.com/SAML/Attributes/DbGroups”的 `Name` 元素

  该元素包含一个或多个 `AttributeValue` 元素。将每个 `AttributeValue` 元素设置为一个数据库组名称，`DbUser` 将在连接到 Amazon Redshift 数据库的会话期间加入该组。

```
<Attribute Name="https://redshift.amazon.com/SAML/Attributes/DbGroups">
    <AttributeValue>group1</AttributeValue>
    <AttributeValue>group2</AttributeValue>
    <AttributeValue>group3</AttributeValue>
</Attribute>
```

步骤 3：创建有权调用 GetClusterCredentialsWithIAM 或 GetClusterCredentials 的 IAM 角色

您的 SQL 客户端需要授权才能代表您调用 `GetClusterCredentialsWithIAM` 或 `GetClusterCredentials` 操作。要提供该授权，可以创建用户或角色并附加用于授予必要权限的策略。这两项操作都可用于获取集群凭证，但它们的身份验证方法不同。`GetClusterCredentialsWithIAM` 使用 IAM 角色进行身份验证，这将自动创建一个映射到该角色的数据库用户，从而帮助管理 IAM 角色级别的权限，而 `GetClusterCredentials` 则会为数据库中的给定用户名提供凭证。

创建有权调用 GetClusterCredentialsWithIAM 的 IAM 角色

1. 利用 IAM 服务，创建用户或角色。您也可以使用现有用户或角色。例如，如果您创建了一个用于身份提供者访问的 IAM 角色，则可向该角色附加必要的 IAM 策略。
2. 附加有权调用 `redshift:GetClusterCredentialsWithIAM` 操作的权限策略。以下策略示例说明了允许对特定集群和数据库、集群中的任意数据库以及任意集群中的任意数据库执行操作的选项。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "SpecificClusterAndDBName",
            "Effect": "Allow",
            "Action": "redshift:GetClusterCredentialsWithIAM",
            "Resource": [
                "arn:aws:redshift:us-east-1:123456789012:dbname:testcluster/
testdatabase"
            ]
        },
        {
            "Sid": "SpecificClusterAndAnyDBName",
            "Effect": "Allow",
            "Action": "redshift:GetClusterCredentialsWithIAM",
            "Resource": "arn:aws:redshift:us-
east-1:123456789012:dbname:examplecluster/*",
        },
        {
            "Sid": "AnyClusterAnyDatabase",
            "Effect": "Allow",
            "Action": "redshift:GetClusterCredentialsWithIAM",
            "Resource": "*"
        }
    ]
}
```

创建有权调用 GetClusterCredentials 的 IAM 角色

1. 利用 IAM 服务，创建用户或角色。您也可以使用现有用户或角色。例如，如果您创建了一个用于身份提供者访问的 IAM 角色，则可向该角色附加必要的 IAM 策略。
2. 附加有权调用 `redshift:GetClusterCredentials` 操作的权限策略。根据指定的可选参数，您还可在策略中允许或限制其他操作和资源：

   - 要允许您的 SQL 客户端检索 Redshift 集群资源的集群 ID、AWS 区域和端口，可包含调用 `redshift:DescribeClusters` 操作的权限。
   - 如果您使用 AutoCreate 选项，请包含对 `redshift:CreateClusterUser` 资源调用 dbuser 的权限。以下 Amazon 资源名称（ARN）指定 Amazon Redshift dbuser。将 *region*、*account-id* 和 *cluster-name* 替换为您的 AWS 区域、账户和集群的相应值。对于 *dbuser-name*，指定用于登录到集群数据库的用户名。

     ```
     arn:aws:redshift:region:account-id:dbuser:cluster-name/dbuser-name
     ```

- （可选）添加一个 ARN，以使用以下格式指定 Amazon Redshift dbname 资源。将 *region*、*account-id* 和 *cluster-name* 替换为您的 AWS 区域、账户和集群的相应值。对于 *database-name*，指定用户将登录到的数据库的名称。

```
arn:aws:redshift:region:account-id:dbname:cluster-name/database-name
```

- 如果您使用 DbGroups 选项，请使用以下格式包含对 Amazon Redshift dbgroup 资源调用 redshift:JoinGroup 操作的权限。将 *region*、*account-id* 和 *cluster-name* 替换为您的 AWS 区域、账户和集群的相应值。对于 *dbgroup-name*，请指定用户在登录时加入的用户组的名称。

```
arn:aws:redshift:region:account-id:dbgroup:cluster-name/dbgroup-name
```

有关更多信息以及示例，请参阅 GetClusterCredentials 的资源策略。

步骤 4：创建数据库用户和数据库组

（可选）您可以创建一个用于登录到集群数据库的数据库用户。如果您为现有用户创建临时用户凭证，则可禁用此用户的密码以强制用户使用临时密码进行登录。（可选）您可以使用 GetClusterCredentials 选项自动创建新的数据库用户。

您可以创建希望 IAM 数据库用户在登录时加入的数据库用户组并提供相应的权限。在调用 GetClusterCredentials 操作时，您可以指定新用户在登录时加入的用户组名称列表。这些组成员资格仅对特定请求生成的凭证所创建的会话有效。

创建数据库用户和数据库组

1. 登录到 Amazon Redshift 数据库，并使用 CREATE USER 创建数据库用户或使用 ALTER USER 修改现有用户。
2. （可选）指定 PASSWORD DISABLE 选项可阻止用户使用密码。在禁用用户的密码后，用户只能使用临时凭证进行登录。如果未禁用密码，用户可以使用密码或临时凭证进行登录。您不能禁用超级用户的密码。

   如果用户需要在 AWS 管理控制台之外与 AWS 交互，则需要编程式访问权限。授予编程式访问权限的方法取决于访问 AWS 的用户类型。

   要向用户授予编程式访问权限，请选择以下选项之一。

| 哪个用户需要编程式访问权限？ | 目的 | 方式 |
| --- | --- | --- |
| IAM | （推荐）使用控制台凭证作为临时凭证来签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照您希望使用的界面的说明进行操作。<br>• 对于 AWS CLI，请参阅《AWS Command Line Interface用户指南》中的登录 AWS 以进行本地开发。<br>• 对于 AWS SDK，请参阅《AWS SDK 和工具参考指南》中的登录 AWS 以进行本地开发。 |
| 人力身份<br>（在 IAM Identity Center 中管理的用户） | 使用临时凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照您希望使用的界面的说明进行操作。<br>• 有关 AWS CLI 的更多信息，请参阅《AWS Command Line Interface 用户指南》中的配置 AWS CLI 以使用 AWS IAM Identity Center。<br>• 有关 AWS SDK、工具和 AWS API 的更多信息，请参阅《AWS SDK 和工具参考指南》中的 IAM Identity Center 身份验证。 |
| IAM | 使用临时凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照《IAM 用户指南》中将临时凭证用于 AWS 资源中的说明进行操作。 |

| 哪个用户需要编程式访问权限？ | 目的 | 方式 |
| --- | --- | --- |
| IAM | （不推荐使用）<br>使用长期凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照您希望使用的界面的说明进行操作。<br>• 有关 AWS CLI 的更多信息，请参阅《AWS Command Line Interface 用户指南》中的使用 IAM 用户凭证进行身份验证。<br>• 有关 AWS SDK 和工具的更多信息，请参阅《AWS SDK 和工具参考指南》中的使用长期凭证进行身份验证。<br>• 有关 AWS API 的更多信息，请参阅《IAM 用户指南》中的管理 IAM 用户的访问密钥。 |

以下示例创建一个已禁用密码的用户。

```
create user temp_creds_user password disable;
```

以下示例禁用现有用户的密码。

```
alter user temp_creds_user password disable;
```

3. 使用 CREATE GROUP 创建数据库用户组。
4. 使用 GRANT 命令为组定义访问权限。

步骤 5：配置 JDBC 或 ODBC 连接以使用 IAM 凭证

您可以使用 Amazon Redshift JDBC 或 ODBC 驱动程序配置 SQL 客户端。此驱动程序管理创建数据库用户凭证以及在 SQL 客户端和 Amazon Redshift 数据库之间建立连接的过程。

如果您使用身份提供者进行身份验证，请指定凭证提供商插件名称。Amazon Redshift JDBC 和 ODBC 驱动程序包括以下基于 SAML 的身份提供者的插件：

- Active Directory 联合身份验证服务 (AD FS)
- PingOne
- Okta
- Microsoft Azure AD

  有关将 Microsoft Azure AD 设置为身份提供者的步骤，请参阅设置 JDBC 或 ODBC 单点登录身份验证。

配置 JDBC 连接以使用 IAM 凭证

1. 从 为 Amazon Redshift 配置 JDBC 驱动程序版本 2.x 连接 页面下载最新的 Amazon Redshift JDBC 驱动程序。
2. 使用下列格式之一创建包含 IAM 凭证选项的 JDBC URL。要使用 IAM 身份验证，请将 `iam:` 添加到 Amazon Redshift JDBC URL 中的 `jdbc:redshift:` 后面，如以下示例中所示。

   ```
   jdbc:redshift:iam://
   ```

   添加 `cluster-name`、`region` 和 `account-id`。JDBC 驱动程序使用您的 IAM 账户信息和集群名称来检索集群 ID和 AWS 区域。为此，您的用户或角色必须有权针对指定的集群调用 `redshift:DescribeClusters` 操作。如果您的用户或角色无权调用 `redshift:DescribeClusters` 操作，请包含集群 ID、AWS 区域和端口，如以下示例中所示。端口号是可选的。

   ```
   jdbc:redshift:iam://examplecluster.abc123xyz789.us-
   west-2.redshift.amazonaws.com:5439/dev
   ```

3. 添加 JDBC 选项可提供 IAM 凭证。使用不同的 JDBC 选项组合可提供 IAM 凭证。有关更多信息，请参阅 用于创建数据库用户凭证的 JDBC 和 ODBC 选项。

   以下 URL 为用户指定 AccessKeyID 和 SecretAccessKey。

   ```
   jdbc:redshift:iam://examplecluster:us-west-2/dev?
   AccessKeyID=AKIAIOSFODNN7EXAMPLE&SecretAccessKey=wJalrXUtnFEMI/K7MDENG/
   bPxRfiCYEXAMPLEKEY
   ```

以下示例指定包含 IAM 凭证的命名配置文件。

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?Profile=user2
```

4. 添加 JDBC 驱动程序用来调用 GetClusterCredentials API 操作的 JDBC 选项。如果您以编程方式调用 GetClusterCredentials API 操作，请不要包括这些选项。

   以下示例包括 JDBC GetClusterCredentials 选项。

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?
plugin_name=com.amazon.redshift.plugin.AzureCredentialsProvider&UID=user&PWD=password&idp_t
```

### 配置 ODBC 连接以使用 IAM 凭证

在以下过程中，您只能找到用于配置 IAM 身份验证的步骤。有关使用标准身份验证 (采用数据库用户名和密码) 的步骤，请参阅为 Amazon Redshift 配置 ODBC 驱动程序版本 2.x 连接。

1. 为您的操作系统安装和配置最新的 Amazon Redshift OBDC 驱动程序。有关更多信息，请参阅为 Amazon Redshift 配置 ODBC 驱动程序版本 2.x 连接页面。

   > **Important**
   >
   > Amazon Redshift ODBC 驱动程序必须为版本 1.3.6.1000 或更高版本。

2. 根据您的操作系统选择配置连接设置需要遵循的步骤。
3. 在 Microsoft Windows 操作系统上，访问“Amazon Redshift ODBC 驱动程序 DSN 设置”窗口。

   a. 在连接设置下，输入以下信息：

      - 数据源名称
      - 服务器（可选）
      - 端口（可选）
      - 数据库

      如果您的用户或角色有权调用 redshift:DescribeClusters 操作，仅需要数据源名称和数据库。Amazon Redshift 通过调用 DescribeCluster 操作使用 ClusterId 和区域来获取服务器和端口。

如果您的用户或角色无权调用 `redshift:DescribeClusters` 操作，请指定服务器和端口。

b. 在身份验证下，为身份验证类型选择一个值。

对于每种身份验证类型，请输入如下列出的值：

AWS 配置文件

输入以下信息：

- ClusterID
- 区域
- 配置文件名称

  输入 AWS config 文件中包含 ODBC 连接选项值的配置文件的名称。有关更多信息，请参阅 使用配置文件。

（可选）提供 ODBC 驱动程序用于调用 `GetClusterCredentials` API 操作的选项的详细信息：

- DbUser
- User AutoCreate
- DbGroups

  有关更多信息，请参阅 用于创建数据库用户凭证的 JDBC 和 ODBC 选项。

IAM 凭证

输入以下信息：

- ClusterID
- 区域
- AccessKeyID 和 SecretAccessKey

  为 IAM 数据库身份验证配置的 IAM 角色或用户的访问密钥 ID 和秘密访问密钥。

- SessionToken

  对于具有临时凭证的 IAM 角色，SessionToken 是必填的。有关更多信息，请参阅临时安全凭证。

提供 ODBC 驱动程序用于调用 `GetClusterCredentials` API 操作的选项的详细信息：

- DbUser（必填）
- User AutoCreate（可选）
- DbGroups（可选）

有关更多信息，请参阅 用于创建数据库用户凭证的 JDBC 和 ODBC 选项。

Identity Provider: AD FS

对于使用 AD FS 的 Windows 集成身份验证，请将 User 和 Password 保留为空。

提供 IdP 详细信息：

- IdP Host

  企业身份提供者主机的名称。此名称不应包含任何斜线 (/)。

- IdP Port（可选）

  身份提供者使用的端口。默认值为 443。

- Preferred Role

  SAML 断言中的 `AttributeValue` 属性的多值 `Role` 元素中的 IAM 角色 Amazon 资源名称（ARN）。请与 IdP 管理员一起查找适合首选角色的值。有关更多信息，请参阅 步骤 2：为 IdP 配置 SAML 断言。

（可选）提供 ODBC 驱动程序用于调用 `GetClusterCredentials` API 操作的选项的详细信息：

- DbUser
- User AutoCreate
- DbGroups

有关更多信息，请参阅 用于创建数据库用户凭证的 JDBC 和 ODBC 选项。

Identity Provider: PingFederate

对于用户和密码，键入您的 IdP 用户名和密码。

提供 IdP 详细信息：

- IdP Host

  企业身份提供者主机的名称。此名称不应包含任何斜线 (/)。

- IdP Port（可选）

  身份提供者使用的端口。默认值为 443。

- Preferred Role

  SAML 断言中的 `AttributeValue` 属性的多值 `Role` 元素中的 IAM 角色 Amazon 资源名称（ARN）。请与 IdP 管理员一起查找适合首选角色的值。有关更多信息，请参阅 步骤 2：为 IdP 配置 SAML 断言。

（可选）提供 ODBC 驱动程序用于调用 `GetClusterCredentials` API 操作的选项的详细信息：

- DbUser
- User AutoCreate
- DbGroups

有关更多信息，请参阅 用于创建数据库用户凭证的 JDBC 和 ODBC 选项。

Identity Provider: Okta

对于用户和密码，键入您的 IdP 用户名和密码。

提供 IdP 详细信息：

- IdP Host

  企业身份提供者主机的名称。此名称不应包含任何斜线 (/)。

- IdP Port

  Okta 不使用此值。

- Preferred Role

  SAML 断言中的 `AttributeValue` 属性的 `Role` 元素中的 IAM 角色 Amazon 资源名称（ARN）。请与 IdP 管理员一起查找适合首选角色的值。有关更多信息，请参阅 步骤 2：为 IdP 配置 SAML 断言。

- Okta App ID

Okta 应用程序的 ID。应用程序 ID 的值位于 Okta 应用程序嵌入链接中的“amazon_aws”之后。与您的 IdP 管理员一起获取此值。

（可选）提供 ODBC 驱动程序用于调用 `GetClusterCredentials` API 操作的选项的详细信息：

- DbUser
- User AutoCreate
- DbGroups

有关更多信息，请参阅 用于创建数据库用户凭证的 JDBC 和 ODBC 选项。

身份提供者：Azure AD

对于用户和密码，键入您的 IdP 用户名和密码。

对于集群 ID 和区域，输入 Amazon Redshift 集群的集群 ID 和 AWS 区域。

对于数据库，输入您为 Amazon Redshift 集群创建的数据库。

提供 IdP 详细信息：

- IdP 租户

  用于 Azure AD 的租户。

- Azure 客户端密钥

  Azure 中的 Amazon Redshift 企业应用程序的客户端密钥。

- Azure 客户端 ID

  Azure 中的 Amazon Redshift 企业应用程序的客户端 ID（应用程序 ID）。

（可选）提供 ODBC 驱动程序用于调用 `GetClusterCredentials` API 操作的选项的详细信息：

- DbUser
- User AutoCreate
- DbGroups

有关更多信息，请参阅 用于创建数据库用户凭证的 JDBC 和 ODBC 选项。

## 用于提供 IAM 凭证的选项

要为 JDBC 或 ODBC 连接提供 IAM 凭证，请选择以下选项之一。

- AWS 配置文件

  作为以 JDBC 或 ODBC 设置形式提供凭证值的替代方案，您可在命名配置文件中放置这些值。有关更多信息，请参阅 使用配置文件。

- IAM 凭证

  以 JDBC 或 ODBC 设置形式提供 AccessKeyID 和 SecretAccessKey 的值，还可以选择提供 SessionToken 的值。SessionToken 仅对于具有临时凭证的 IAM 角色是必填的。有关更多信息，请参阅 用于提供 IAM 凭证的 JDBC 和 ODBC 选项。

- 身份提供者联合

  在使用身份提供者联合以允许身份提供者中的用户在 Amazon Redshift 中进行身份验证时，请指定凭证提供商插件的名称。有关更多信息，请参阅 凭证提供程序插件。

  Amazon Redshift JDBC 和 ODBC 驱动程序包括以下基于 SAML 的联合身份验证凭证提供商的插件：

  - Microsoft Active Directory 联合身份验证服务 (AD FS)
  - PingOne
  - Okta
  - Microsoft Azure Active Directory (Azure AD)

  您可以 JDBC 或 ODBC 设置格式提供或使用配置文件提供插件名称和相关值。有关更多信息，请参阅 JDBC 驱动程序版本 2.x 配置的选项。

有关更多信息，请参阅 步骤 5：配置 JDBC 或 ODBC 连接以使用 IAM 凭证。

使用配置文件

您可提供 IAM 凭证选项和 `GetClusterCredentials` 选项作为 AWS 配置文件中的命名配置文件的设置。要提供配置文件名称，请使用配置文件 JDBC 选项。该配置存储在名为 `config` 的文件或位于主目录下面的名为 `credentials` 的文件夹中的名为 `.aws` 的文件中。

对于 Amazon Redshift JDBC 或 ODBC 驱动程序附带的基于 SAML 的凭证提供商插件，您可以使用 凭证提供程序插件 中前面所述的设置。如果未使用 `plugin_name`，则忽略其他选项。

下例所示为一个有两个配置文件的 ~/.aws/credentials 文件。

```
[default]
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

[user2]
aws_access_key_id=AKIAI44QH8DHBEXAMPLE
aws_secret_access_key=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
session_token=AQoDYXdzEPT//////////
wEXAMPLEtc764bNrC9SAPBSM22wDOk4x4HIZ8j4FZTwdQWLWsKWHGBuFqwAeMicRXmxfpSPfIeoIYRqTflfKD8YUuwthAx7
qkPpKPi/kMcGd
QrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburEDy0KPkyQDYwT7WZ0wq5VSXDvp75YU
9HFvlRd8Tx6q6fE8YQcHNVXAkiY9q6d+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
+scqKmlzm8FDrypNC9Yjc8fPOLn9FX9KSYvKTr4rvx3iSIlTJabIQwj2ICCR/oLxBA==
```

要使用 user2 示例的凭证，请在 JDBC URL 中指定 Profile=user2。

有关使用配置文件的更多信息，请参阅《AWS Command Line Interface 用户指南》中的配置和凭证文件设置。

有关为 JDBC 驱动程序使用配置文件的更多信息，请参阅指定配置文件。

有关为 ODBC 驱动程序使用配置文件的更多信息，请参阅身份验证方法。

用于提供 IAM 凭证的 JDBC 和 ODBC 选项

下表列出了用于提供 IAM 凭证的 JDBC 和 ODBC 选项。

| 选项 | 描述 |
| --- | --- |
| Iam | 仅用于 ODBC 连接字符串。设置为 1 可使用 IAM 身份验证。 |
| AccessKeyID<br><br>SecretAccessKey<br><br>SessionToken | 为 IAM 数据库身份验证配置的 IAM 角色或用户的访问密钥 ID 和秘密访问密钥。SessionToken 仅对具有临时凭证的 IAM 角色是必需的。SessionToken 不用于用户。有关更多信息，请参阅临时安全凭证。 |

| 选项 | 描述 |
| --- | --- |
| plugin_name | 实施凭证提供商的类的完全限定名称。Amazon Redshift JDBC 驱动程序包括基于 SAML 的凭证提供商插件。如果提供 plugin_name ，您还可以提供其他相关选项。有关更多信息，请参阅 凭证提供程序插件。 |
| Profile | AWS 凭证中配置文件的名称或包含 JDBC 连接选项值的 config 文件的名称。有关更多信息，请参阅 使用配置文件。 |

用于创建数据库用户凭证的 JDBC 和 ODBC 选项

要使用 Amazon Redshift JDBC 或 ODBC 驱动程序创建数据库用户凭证，请提供数据库用户名作为 JDBC 或 ODBC 选项。（可选）您可以让驱动程序创建新的数据库用户（如果不存在），并且可以指定用户在登录时加入的数据库用户组列表。

如果您使用身份提供者 (IdP)，请与 IdP 管理员一起确定这些选项的正确值。您的 IdP 管理员还可配置 IdP 来提供这些选项，这样一来，您将无需提供它们作为 JDBC 或 ODBC 选项。有关更多信息，请参阅 步骤 2：为 IdP 配置 SAML 断言。

> **Note**
>
> 如果您使用 IAM 策略变量 ${redshift:DbUser}（如GetClusterCredentials 的资源策略中所述），则 DbUser 的值将被替换为由 API 操作的请求上下文检索的值。Amazon Redshift 驱动程序使用由连接 URL 提供的 DbUser 变量的值，而不是作为 SAML 属性提供的值。
> 为帮助保护此配置，我们建议您在 IAM 策略中使用一个使用来通过 DbUser 验证 RoleSessionName 值。您可以在 示例 8：使用 GetClusterCredentials 的 IAM 策略 中找到如何使用 IAM 策略设置条件。

下表列出了用于创建数据库用户凭证的选项。

| 选项 | 描述 |
| --- | --- |
| DbUser | 数据库用户的名称。如果数据库中存在名为 DbUser 的用户，则临时用户凭证具有与现有用户相同的权限。如果数据库中不存在 DbUser 且 AutoCreate 为 true，则创建一个名为 DbUser 的新用户。（可选）禁用现有用户的密码。有关更多信息，请参阅 ALTER_USER |

| 选项 | 描述 |
| --- | --- |
| AutoCreate | 指定 `true` 以使用为 DbUser 指定的名称创建数据库用户（如果不存在）。默认为 false。 |
| `DbGroups` | 数据库用户在当前会话中加入的一个或多个现有数据库组的名称的逗号分隔列表。默认情况下，新用户仅添加到 PUBLIC。 |

凭证提供程序插件

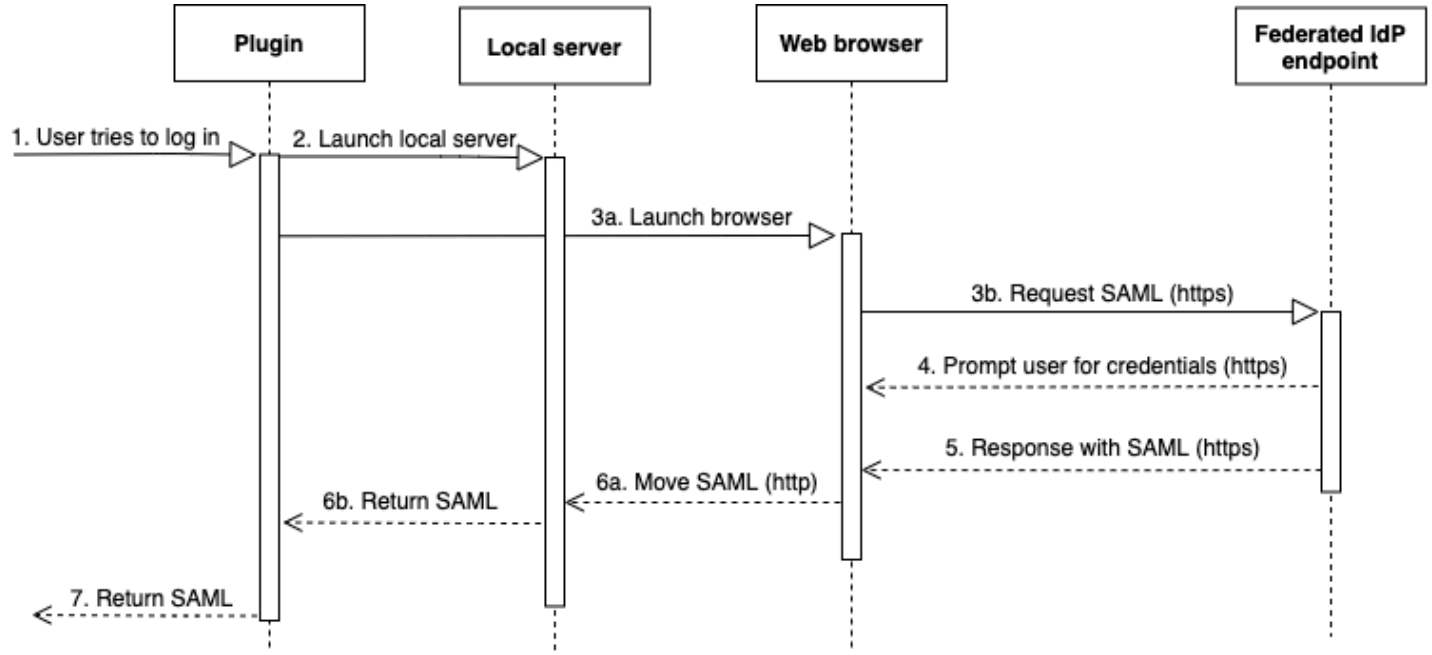
Amazon Redshift 使用凭证提供商插件进行单点登录身份验证。

为了支持单点登录身份验证，Amazon Redshift 提供了适用于 Microsoft Azure Active Directory 的 Azure AD 插件。有关如何配置该插件的信息，请参阅设置 JDBC 或 ODBC 单点登录身份验证。

多重身份验证
多重身份验证

为了支持多重身份验证（MFA），Amazon Redshift 提供了基于浏览器的插件。请使用适用于 Okta、PingOne 的的浏览器 SAML 插件，以及适用于 Microsoft Azure Active Directory 的浏览器 Azure AD 插件。

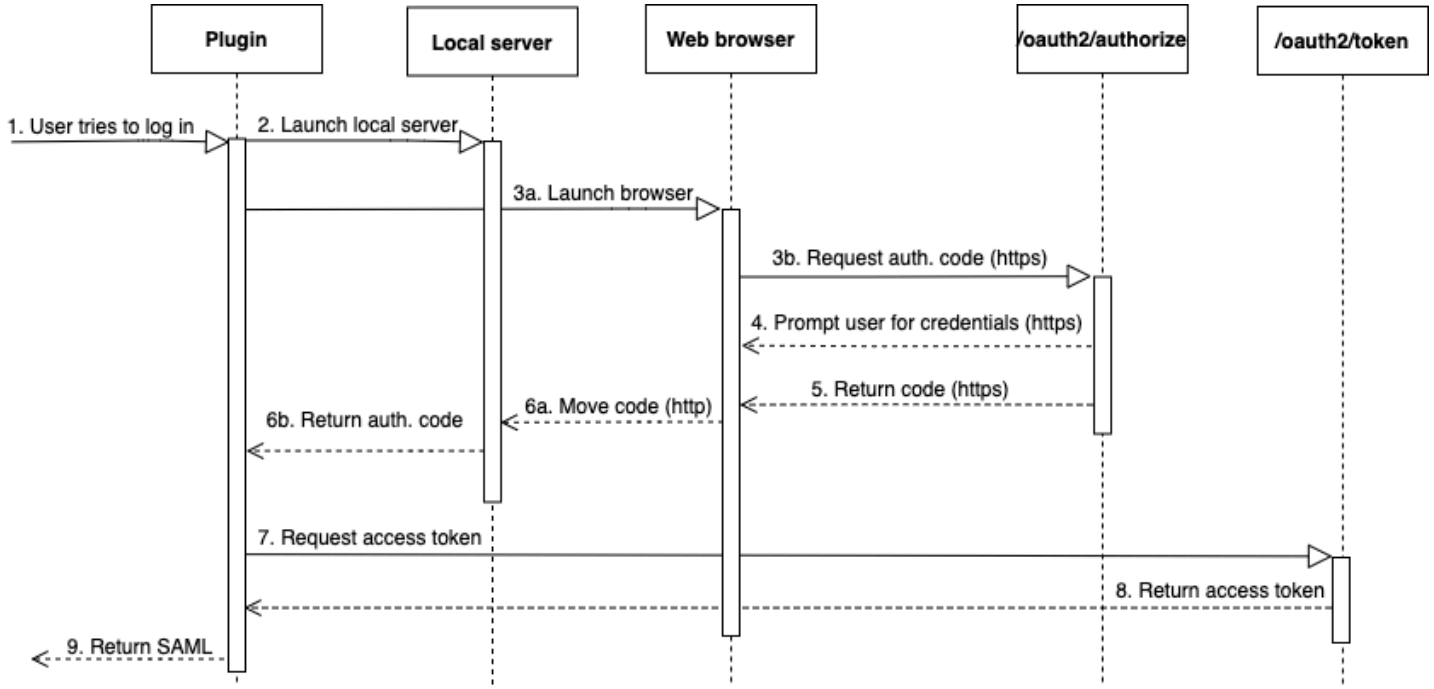
在使用浏览器 SAML 插件时，OAuth 身份验证流程如下所示：




1. 用户尝试登录。

2. 该插件启动本地服务器以侦听 localhost 上的传入连接。
3. 该插件启动 Web 浏览器，以通过 HTTPS 从指定的单点登录 URL 联合身份提供者端点请求 SAML 响应。
4. Web 浏览器访问该链接，并提示用户输入凭证。
5. 在用户进行身份验证并获得允许后，联合身份提供者端点通过 HTTPS 向 `redirect_uri` 指示的 URI 返回 SAML 响应。
6. Web 浏览器将包含 SAML 响应的响应消息传送到指示的 `redirect_uri`。
7. 本地服务器接受传入连接，该插件检索 SAML 响应并将其传送到 Amazon Redshift。

在使用浏览器 Azure AD 插件时，SAML 身份验证流程如下所示：




1. 用户尝试登录。
2. 该插件启动本地服务器以侦听 localhost 上的传入连接。
3. 该插件启动 Web 浏览器，以从 Azure AD `oauth2/authorize` 端点请求授权代码。
4. Web 浏览器通过 HTTPS 访问生成的链接，并提示用户输入凭证。该链接是使用配置属性（例如 tenant 和 client_id）生成的。
5. 在用户进行身份验证并获得允许后，Azure AD `oauth2/authorize` 端点通过 HTTPS 返回包含授权代码的响应并将其发送到指示的 `redirect_uri`。
6. Web 浏览器将包含 SAML 响应的响应消息传送到指示的 `redirect_uri`。

7. 本地服务器接受传入连接，该插件请求和检索授权代码，并将 POST 请求发送到 Azure AD `oauth2/token` 端点。
8. Azure AD `oauth2/token` 端点将包含访问令牌的响应返回到指示的 `redirect_uri`。
9. 该插件检索 SAML 响应并将其传送到 Amazon Redshift。

参阅以下部分：

- Active Directory 联合身份验证服务 (AD FS)

  有关更多信息，请参阅 设置 JDBC 或 ODBC 单点登录身份验证。
- PingOne (Ping)

  只有使用 Forms 身份验证预先确定的 PingOne IdP 适配器支持 Ping。

  有关更多信息，请参阅 设置 JDBC 或 ODBC 单点登录身份验证。
- Okta

  仅与 AWS 管理控制台 一起使用的 Okta 提供的应用程序支持 Okta。

  有关更多信息，请参阅 设置 JDBC 或 ODBC 单点登录身份验证。
- Microsoft Azure Active Directory

  有关更多信息，请参阅 设置 JDBC 或 ODBC 单点登录身份验证。

插件选项
插件选项

要使用基于 SAML 的凭证提供商插件，请使用 JBDC 或 ODBC 选项，或在命名配置文件中指定以下选项。如果未指定 `plugin_name`，则忽略其他选项。

| 选项 | 描述 |
| --- | --- |
| `plugin_name` | 对于 JDBC，为实施凭证提供商的类名称。指定下列项之一：<br>• 对于 Active Directory 联合身份验证服务<br>`com.amazon.redshift.plugin.AdfsCredentialsProvider`<br>• 对于 Okta |

| 选项 | 描述 |
| --- | --- |
| | `com.amazon.redshift.plugin.OktaCredentialsProvider`<br>• 对于 PingFederate<br>`com.amazon.redshift.plugin.PingCredentialsProvider`<br>• 对于 Microsoft Azure Active Directory<br>`com.amazon.redshift.plugin.AzureCredentialsProvider`<br>• 对于 SAML MFA<br>`com.amazon.redshift.plugin.BrowserSamlCredentialsProvider`<br>• 对于具有 MFA 的 Microsoft Azure Active Directory 单点登录<br>`com.amazon.redshift.plugin.BrowserAzureCredentialsProvider`<br><br>对于 ODBC，指定下列项之一：<br>• 对于 Active Directory 联合身份验证服务：`adfs`<br>• 对于 Okta：`okta`<br>• 对于 PingFederate：`ping`<br>• 对于 Microsoft Azure Active Directory：`azure`<br>• 对于 SAML MFA：`browser saml`<br>• 对于具有 MFA 的 Microsoft Azure Active Directory 单点登录： `browser azure ad` |
| `idp_host` | 企业身份提供者主机的名称。此名称不应包含任何斜线 (“/”)。对于 Okta 身份提供者，`idp_host` 的值应以 `.okta.com` 结尾。 |
| `idp_port` | 身份提供者使用的端口。默认值为 443。Okta 将忽略该端口。 |

| 选项 | 描述 |
| --- | --- |
| preferred_role | SAML 断言中的 AttributeValue 属性的 Role 元素中的角色 Amazon 资源名称（ARN）。请与 IdP 管理员一起查找适合首选角色的值。有关更多信息，请参阅 步骤 2：为 IdP 配置 SAML 断言。 |
| user | 企业用户名，包括域 (如果适用)。例如，对于 Active Directory，域名的格式需为 domain\username。 |
| 密码 | 企业用户的密码。我们建议不使用此选项，而是使用您的 SQL 客户端提供密码。 |
| app_id | Okta 应用程序的 ID。仅用于 Okta。app_id 的值位于 Okta 应用程序嵌入链接中的 amazon_aws 后面。与 IdP 管理员一起获取该值。以下是应用程序嵌入链接的示例：https://example.okta.com/home/amazon_aws/0oa2hylwrpM8U Gehd1t7/272 |
| idp_tenant | 用于 Azure AD 的租户。仅与 Azure 一起使用。 |
| client_id | Azure AD 中的 Amazon Redshift 企业应用程序的客户端 ID。仅与 Azure 一起使用。 |

使用 Amazon Redshift CLI 或 API 生成 IAM 身份的数据库凭证

为了以编程方式生成临时数据库用户凭证，Amazon Redshift 提供适用于 AWS Command Line Interface (AWS CLI) 和 GetClusterCredentials API 操作的 get-cluster-credentials 命令。或者，您可以使用 Amazon Redshift JDBC 或 ODBC 驱动程序配置您的 SQL 客户端，这些驱动程序用于管理调用 GetClusterCredentials 操作，检索数据库用户凭证，并在您的 SQL 客户端与您的 Amazon Redshift 数据库之间建立连接的过程。有关更多信息，请参阅 用于创建数据库用户凭证的 JDBC 和 ODBC 选项。

> **Note**
>
> 我们建议使用 Amazon Redshift JDBC 或 ODBC 驱动程序生成数据库用户凭证。

在此部分中，您可以找到用于以编程方式调用 GetClusterCredentials 操作或 get-cluster-credentials 命令，检索数据库用户凭证并连接到数据库的步骤。

生成和使用临时数据库凭证

1. 创建或修改具有所需权限的用户或角色。有关 IAM 权限的更多信息，请参阅步骤 3：创建有权调用 GetClusterCredentialsWithIAM 或 GetClusterCredentials 的 IAM 角色。
2. 以您在上一步骤中授权的用户或角色的身份，运行 get-cluster-credentials CLI 命令或调用 `GetClusterCredentials` API 操作并提供下列值：
   - 集群标识符 – 包含数据库的集群的名称。
   - 数据库用户名 – 现有的或新的数据库用户的名称。
     - 如果数据库中不存在此用户且 AutoCreate 为 true，则将创建一个已禁用 PASSWORD 的新用户。
     - 如果此用户不存在且 AutoCreate 为 false，则请求会失败。
     - 在此示例中，数据库用户名为 `temp_creds_user`。
   - Autocreate –（可选）如果数据库用户名不存在，则创建新用户。
   - 数据库名称 –（可选）授权用户登录的数据库的名称。如果未指定数据库名称，则用户可以登录到任何集群数据库。
   - 数据库组 –（可选）现有数据库用户组的列表。在成功登录后，数据库用户将添加到指定的用户组中。如果未指定组，则用户仅具有 PUBLIC 权限。此用户组名称必须与在附加到用户或角色的 IAM 策略中指定的 dbgroup 资源 ARN 匹配。
   - 过期时间 –（可选）临时凭证过期之前经历的时间（以秒为单位）。您可指定一个介于 900 秒 (15 分钟) 和 3600 秒 (60 分钟) 之间的值。默认值为 900 秒。
3. Amazon Redshift 确认用户是否有权使用指定资源来调用 `GetClusterCredentials` 操作。
4. Amazon Redshift 返回临时密码和数据库用户名。

   以下示例使用 Amazon Redshift CLI 为名为 `temp_creds_user` 的现有用户生成临时数据库凭证。

```
aws redshift get-cluster-credentials --cluster-identifier examplecluster --db-user
 temp_creds_user --db-name exampledb --duration-seconds 3600
```

   结果如下所示。

```
{
  "DbUser": "IAM:temp_creds_user",
  "Expiration": "2016-12-08T21:12:53Z",
```

```
  "DbPassword": "EXAMPLEjArE3hcnQj8zt4XQj9Xtma8oxYEM8OyxpDHwXVPyJYBDm/
gqX2Eeaq6P3DgTzgPg=="
}
```

以下示例使用 Amazon Redshift CLI 与 autocreate 为新用户生成临时数据库凭证并将此用户添加到组 example_group。

```
aws redshift get-cluster-credentials --cluster-identifier examplecluster --db-user
 temp_creds_user --auto-create --db-name exampledb --db-groups example_group --
duration-seconds 3600
```

结果如下所示。

```
{
  "DbUser": "IAMA:temp_creds_user:example_group",
  "Expiration": "2016-12-08T21:12:53Z",
  "DbPassword": "EXAMPLEjArE3hcnQj8zt4XQj9Xtma8oxYEM8OyxpDHwXVPyJYBDm/
gqX2Eeaq6P3DgTzgPg=="
}
```

5. 建立到 Amazon Redshift 集群的安全套接字层 (SSL) 身份验证连接，并发送包含 GetClusterCredentials 响应中的用户名和密码的登录请求。请在用户名中包含 IAM: 或 IAMA: 前缀，例如，IAM:temp_creds_user 或 IAMA:temp_creds_user。

   > **Important**
   >
   > 将您的 SQL 客户端配置为需要 SSL。否则，如果您的 SQL 客户端自动尝试使用 SSL 进行连接，则可能会在出现任何故障时回退到非 SSL。在这种情况下，首次连接尝试可能因凭证过期或无效而失败，随后的另一次连接尝试可能因连接不是 SSL 而失败。如果出现这种情况，第一条错误消息可能会丢失。有关使用 SSL 连接到集群的更多信息，请参阅配置连接的安全选项。

6. 如果连接未使用 SSL，则连接尝试可能失败。
7. 集群将向 SQL 客户端发送 authentication 请求。
8. 随后，SQL 客户端会向集群发送临时密码。
9. 如果密码有效且尚未到期，集群将完成连接。

设置 JDBC 或 ODBC 单点登录身份验证

您可以利用外部身份提供者（IdP）对访问 Amazon Redshift 集群的用户进行身份验证和授权，从而简化用户管理并增强安全性。这就实现了集中式用户管理、基于角色的访问控制以及跨多个服务的审计功能。常见应用场景包括简化不同用户组的身份验证、执行一致的访问策略以及满足监管要求。

以下页面将指导您配置 IdP 与 Redshift 集群的集成。有关配置 AWS 作为 IdP 的服务提供商的更多信息，请参阅 IAM 用户指南中的通过信赖方信任和添加陈述来配置 SAML 2.0 IdP。

AD FS

本教程将向您展示如何将 AD FS 用作身份提供者（IdP）来访问 Amazon Redshift 集群。

第 1 步：设置 AD FS 和您的 AWS 账户相互信任

以下过程介绍如何建立信任关系。

1. 创建或使用现有的 Amazon Redshift 集群，以使 AD FS 用户连接到该集群。要配置连接，需要此集群的某些属性，例如集群标识符。有关更多信息，请参阅创建集群。
2. 在 Microsoft 管理控制台上设置 AD FS 以控制 Amazon Redshift 访问：

    1. 选择 ADFS 2.0，然后选择添加信赖方信任。在添加信赖方信任向导页面上，选择开始。
    2. 在选择数据源页面上，选择导入有关在线或在本地网络上发布的信赖方的数据。
    3. 对于联合元数据地址(主机名或 URL)，请输入 **`https://signin.aws.amazon.com/saml-metadata.xml`**。元数据 XML 文件是将 AWS 描述为信赖方的标准 SAML 元数据文档。
    4. 在指定显示名称页面上，输入显示名称的值。
    5. 在选择颁发授权规则页面上，选择一个颁发授权规则以允许或拒绝所有用户访问该信赖方。
    6. 在准备好添加信任页面上，查看您的设置。
    7. 在完成页面上，选择向导关闭时打开此信赖方信任的“编辑声明规则”对话框。
    8. 在上下文（右键单击）菜单上，选择信赖方信任。
    9. 对于您的信赖方，打开上下文（右键单击）菜单，然后选择编辑声明规则。在编辑声明规则页面上，选择添加规则。
    10. 对于声明规则模板，请选择转换传入声明，然后在编辑规则 - NameId 页面上执行以下操作：
        - 对于声明规则名称，请输入 NameId。
        - 对于传入声明名称，请选择 Windows 账户名。
        - 对于传出声明名称，请选择名称 ID。
        - 对于传出名称 ID 格式，请选择持久性标识符。

- 选择传递所有声明值。

11.在编辑声明规则页面上，选择添加规则。在选择规则模板页面上，对于声明规则模板，请选择以声明方式发送 LDAP 特性。

12.在配置规则页面上，执行以下操作：

- 对于声明规则名称，请输入 RoleSessionName。
- 对于特性存储，请选择 Active Directory。
- 对于 LDAP 特性，请选择电子邮件地址。
- 对于传出声明类型，选择 https://aws.amazon.com/SAML/Attributes/RoleSessionName。

13.在编辑声明规则页面上，选择添加规则。在选择规则模板页面上，对于声明规则模板，请选择使用自定义规则发送声明。

14.在编辑规则 - 获取 AD 组页面上，对于声明规则名称，请输入获取 AD 组。

15.对于自定义规则，请输入以下内容。

```
c:[Type ==
                                    "http://schemas.microsoft.com/ws/2008/06/
identity/claims/windowsaccountname",
                                    Issuer == "AD AUTHORITY"] => add(store =
 "Active Directory",
                                    types = ("http://temp/variable"), query =
 ";tokenGroups;{0}",
                                    param = c.Value);
```

16.在编辑声明规则页面上，选择添加规则。在选择规则模板页面上，对于声明规则模板，请选择使用自定义规则发送声明。

17.在编辑规则 - 角色页面上，对于声明规则名称，请键入角色。

18.对于自定义规则，请输入以下内容。

```
c:[Type == "http://temp/variable", Value =~ "(?i)^AWS-"] =>
 issue(Type = "https://aws.amazon.com/SAML/Attributes/Role", Value =
 RegExReplace(c.Value, "AWS-", "arn:aws:iam::123456789012:saml-provider/
ADFS,arn:aws:iam::123456789012:role/ADFS-"));
```

请注意 SAML 提供商的 ARN 和要担任的角色。在该示例中，`arn:aws:iam:123456789012:saml-provider/ADFS` 是 SAML 提供商的 ARN，`arn:aws:iam:123456789012:role/ADFS-` 是角色的 ARN。

3. 确保您已下载 `federationmetadata.xml` 文件。检查以确认文档内容不包含无效的字符。这是在配置与 AWS 的信任关系时使用的元数据文件。
4. 在 IAM 控制台上创建 IAM SAML 身份提供者。您提供的元数据文档是您在设置 Azure 企业应用程序时保存的联合元数据 XML 文件。有关详细步骤，请参阅《IAM 用户指南》中的创建和管理 IAM 身份提供者（控制台）。
5. 在 IAM 控制台上为 SAML 2.0 联合身份创建 IAM 角色。有关详细步骤，请参阅 IAM 用户指南中的创建用于 SAML 的角色。
6. 创建一个 IAM 策略，您可以将其附加到您在 IAM 控制台上为 SAML 2.0 联合身份验证创建的 IAM 角色。有关详细步骤，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。有关 Azure AD 示例，请参阅设置 JDBC 或 ODBC 单点登录身份验证。

第 2 步：设置 JDBC 或 ODBC，以便向 AD FS 进行身份验证

JDBC

以下过程介绍如何设置与 AD FS 的 JDBC 关系。

- 将数据库客户端配置为通过 JDBC 并使用 AD FS 单点登录连接到集群。

  您可以使用任何采用 JDBC 驱动程序的客户端通过 AD FS 单点登录进行连接，也可以使用像 Java 这样的语言通过脚本进行连接。有关安装和配置信息，请参阅为 Amazon Redshift 配置 JDBC 驱动程序版本 2.x 连接。

  例如，您可以使用 SQLWorkbench/J 作为客户端。当您配置 SQLWorkbench/J 时，数据库的 URL 使用以下格式。

  ```
  jdbc:redshift:iam://cluster-identifier:us-west-1/dev
  ```

  如果您使用 SQLWorkbench/J 作为客户端，请执行以下步骤：

  a. 启动 SQL Workbench/J。在选择连接配置文件页面中，添加一个配置文件组，例如，**ADFS**。
  b. 对于连接配置文件，请输入您的连接配置文件名称，例如 **ADFS**。
  c. 选择管理驱动程序，然后选择 Amazon Redshift。选择库旁边的打开文件夹图标，然后选择适当的 JDBC .jar 文件。
  d. 在选择连接配置文件页面上，向连接配置文件添加信息，如下所示：

- 对于用户，请输入您的 AD FS 用户名。这是您用于单点登录的账户的用户名，该账户有权限访问您在尝试进行身份验证时使用的集群。
- 对于密码，请输入您的 AD FS 密码。
- 对于驱动程序，请选择 Amazon Redshift (com.amazon.redshift.jdbc.Driver)。
- 对于 URL，请输入 **jdbc:redshift:iam://*your-cluster-identifier*:*your-cluster-region*/*your-database-name***。

e. 选择扩展属性。对于 plugin_name，请输入 **com.amazon.redshift.plugin.AdfsCredentialsProvider**。该值指定驱动程序将 AD FS 单点登录作为身份验证方法。

ODBC

设置 ODBC 以在 AD FS 中进行身份验证

- 将数据库客户端配置为通过 ODBC 并使用 AD FS 单点登录连接到集群。

  Amazon Redshift 提供了适用于 Linux、Windows 和 macOS 操作系统的 ODBC 驱动程序。在安装 ODBC 驱动程序之前，请确定您的 SQL 客户端工具是 32 位还是 64 位。安装符合 SQL 客户端工具要求的 ODBC 驱动程序。

  在 Windows 上的 Amazon Redshift ODBC 驱动程序 DSN 安装页的连接设置下，输入以下信息：

  - 对于数据源名称，输入 ***your-DSN***。这将指定用作 ODBC 配置文件名称的数据源名称。
  - 对于身份验证类型，选择身份提供者：SAML。这是 ODBC 驱动程序在通过 AD FS 单点登录进行身份验证时使用的身份验证方法。
  - 对于集群 ID，请输入 ***your-cluster-identifier***。
  - 对于区域，请输入 ***your-cluster-region***。
  - 对于数据库，请输入 ***your-database-name***。
  - 对于用户，请输入 ***your-adfs-username***。这是您用于单点登录的 AD FS 账户的用户名，该账户有权访问您在尝试进行身份验证时使用的集群。请仅将其用于身份验证类型为身份提供者: SAML 的情况。
  - 对于密码，请输入 ***your-adfs-password***。请仅将其用于 Auth type (身份验证类型) 为 Identity Provider: SAML (身份提供者: SAML) 的情况。

在 macOS 和 Linux 上，按如下方式编辑 odbc.ini 文件：

> **Note**
>
> 所有条目不区分大小写。

- 对于 clusterid，请输入 ***your-cluster-identifier***。这是已创建的 Amazon Redshift 集群的名称。
- 对于区域，请输入 ***your-cluster-region***。这是已创建的 Amazon Redshift 集群的 AWS 区域。
- 对于数据库，请输入 ***your-database-name***。这是您尝试在 Amazon Redshift 集群上访问的数据库的名称。
- 对于区域设置，请输入 **en-us**。这是显示错误消息的语言。
- 对于 iam，请输入 **1**。此值指定要使用 IAM 凭证进行身份验证的驱动程序。
- 对于 plugin_name，请执行以下操作之一：
  - 对于具有 MFA 的 AD FS 单点登录配置，请输入 **BrowserSAML**。这是 ODBC 驱动程序在 AD FS 中进行身份验证时使用的身份验证方法。
  - 对于 AD FS 单点登录配置，请输入 **ADFS**。这是 ODBC 驱动程序在通过 Azure AD 单点登录进行身份验证时使用的身份验证方法。
- 对于 uid，请输入 ***your-adfs-username***。这是您用于单点登录的 Microsoft Azure 账户的用户名，该账户有权限访问您在尝试进行身份验证时使用的集群。请仅将其用于 plugin_name 为 ADFS 的情况。
- 对于密码，请输入 ***your-adfs-password***。请仅将其用于 plugin_name 为 ADFS 的情况。

在 macOS 和 Linux 上，还要编辑配置文件设置以添加以下导出。

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

Azure

您可以使用 Microsoft Azure AD 作为身份提供者 (IdP) 来访问 Amazon Redshift 集群。本教程将向您展示如何将 Azure 用作身份提供者（IdP）来访问 Amazon Redshift 集群。

观看以下视频，了解如何通过 Microsoft Azure AD 单点登录进行联合 Amazon Redshift 访问：通过 Microsoft Azure AD 单点登录进行联合 Amazon Redshift 访问。

第 1 步：设置 Azure 和您的 AWS 账户相互信任

以下过程介绍如何建立信任关系。

将 Azure AD 和您的 AWS 账户设置为彼此信任

1. 创建或使用现有的 Amazon Redshift 集群，以使 Azure AD 用户连接到该集群。要配置连接，需要此集群的某些属性，例如集群标识符。有关更多信息，请参阅创建集群。
2. 在 Microsoft Azure 门户上设置用于 AWS 的 Azure Active Directory、组、用户。
3. 在 Microsoft Azure 门户上添加 Amazon Redshift 以作为企业应用程序，以用于 AWS 控制台单点登录和 Amazon Redshift 联合登录。选择企业应用程序。
4. 选择 + 新建应用程序。将显示“添加应用程序”页面。
5. 在搜索字段中搜索 **AWS**。
6. 选择 Amazon Web Services (AWS)，然后选择添加。将会创建 AWS 应用程序。
7. 在管理下方，选择单点登录。
8. 选择 SAML。将显示 Amazon Web Services (AWS) | 基于 SAML 的登录页面。
9. 选择是以转到“使用 SAML 设置单点登录”页面。该页面显示预配置的单点登录相关属性的列表。
10. 对于基本 SAML 配置，请选择编辑图标，然后选择保存。
11. 在配置多个应用程序时，请提供一个标识符值。例如，输入 ***https://signin.aws.amazon.com/saml#2***。请注意，从第二个应用程序开始，将该格式与 # 符号一起使用以指定唯一的 SPN 值。
12. 在用户属性和声明部分中，选择编辑图标。

    默认情况下，预配置了唯一用户标识符 (UID)、角色、RoleSessionName 和 SessionDuration 声明。
13. 选择 + 添加新的声明，以便为数据库用户添加声明。

    对于名称，请输入 **DbUser**。

    对于命名空间，请输入 **https://redshift.amazon.com/SAML/Attributes**。

对于源，请选择属性。

对于源属性，请选择 user.userprincipalname。然后，选择保存。

14. 选择 + 添加新的声明，以便为 AutoCreate 添加声明。

对于名称，请输入 **AutoCreate**。

对于命名空间，请输入 **https://redshift.amazon.com/SAML/Attributes**。

对于源，请选择属性。

对于源属性，请选择 "true"。然后，选择保存。

此处，*123456789012* 是您的 AWS 账户，*AzureSSO* 是您创建的 IAM 角色，*AzureADProvider* 是 IAM 提供商。

| 声明名称 | 值 |
| --- | --- |
| 唯一用户标识符（名称 ID） | user.userprincipalname |
| https://aws.amazon.com/SAML/Attributes/SessionDuration | "900" |
| https://aws.amazon.com/SAML/Attributes/Role | arn:aws:iam::*123456789012* :role/*AzureSSO*,arn:aws:iam::*123456789012* :saml-provider/*AzureADProvider* |
| https://aws.amazon.com/SAML/Attributes/RoleSessionName | user.userprincipalname |
| https://redshift.amazon.com/SAML/Attributes/AutoCreate | "true" |
| https://redshift.amazon.com/SAML/Attributes/DbGroups | user.assignedroles |
| https://redshift.amazon.com/SAML/Attributes/DbUser | user.userprincipalname |

15. 在应用程序注册 > ***your-application-name*** > 身份验证 下，添加移动和桌面应用程序。将 URL 指定为 http://localhost/redshift/。
16. 在 SAML 签名证书 部分中，选择下载以下载并保存联合元数据 XML 文件，以便在创建 IAM SAML 身份提供者时使用。该文件用于创建单点登录联合身份。
17. 在 IAM 控制台上创建 IAM SAML 身份提供者。您提供的元数据文档是您在设置 Azure 企业应用程序时保存的联合元数据 XML 文件。有关详细步骤，请参阅《IAM 用户指南》中的创建和管理 IAM 身份提供者（控制台）。
18. 在 IAM 控制台上为 SAML 2.0 联合身份创建 IAM 角色。有关详细步骤，请参阅 IAM 用户指南中的创建用于 SAML 的角色。
19. 创建一个 IAM 策略，您可以将其附加到您在 IAM 控制台上为 SAML 2.0 联合身份验证创建的 IAM 角色。有关详细步骤，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。

    为您的环境修改以下策略（JSON 格式）：

    - 将集群的 AWS 区域替换为 *us-west-1*。
    - 将您的 AWS 账户替换为 *123456789012*。
    - 将您的集群标识符（或对于所有集群，则为 *）替换为 *cluster-identifier*。
    - 将您的数据库（或对于所有集群，则为 *）替换为 *dev*。
    - 将 IAM 角色的唯一标识符替换为 *AROAJ2UCCR6DPCEXAMPLE*。
    - 将您的租户或公司电子邮件域替换为 *example.com*。
    - 将您计划为其分配用户的数据库组替换为 *my_dbgroup*。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "redshift:GetClusterCredentials",
            "Resource": [
                "arn:aws:redshift:us-west-1:123456789012:dbname:cluster-
identifier/dev",
                "arn:aws:redshift:us-west-1:123456789012:dbuser:cluster-identifier/
${redshift:DbUser}",
                "arn:aws:redshift:us-west-1:123456789012:cluster:cluster-
identifier"
            ],
            "Condition": {
```

```
                "StringEquals": {
                    "aws:userid": "AROAJ2UCCR6DPCEXAMPLE:
${redshift:DbUser}@example.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "redshift:CreateClusterUser",
            "Resource": "arn:aws:redshift:us-west-1:123456789012:dbuser:cluster-
identifier/${redshift:DbUser}"
        },
        {
            "Effect": "Allow",
            "Action": "redshift:JoinGroup",
            "Resource": "arn:aws:redshift:us-west-1:123456789012:dbgroup:cluster-
identifier/my_dbgroup"
        },
        {
            "Effect": "Allow",
            "Action": [
                "redshift:DescribeClusters",
                "iam:ListRoles"
            ],
            "Resource": "*"
        }
    ]
}
```

此策略授予权限，如下所示：

- 第一部分授予执行 GetClusterCredentials API 操作的权限，以获取指定集群的临时凭证。在此示例中，资源是 *cluster-identifier*，所在的数据库为 *dev*，所在的账户为 *123456789012*，所在的 AWS 区域为 *us-west-1*。${redshift:DbUser} 子句仅允许与在 Azure AD 中指定的 DbUser 值匹配的用户进行连接。
- 条件子句会强制只有特定用户能够获得临时凭证。这些用户是由角色唯一 ID *AROAJ2UCCR6DPCEXAMPLE* 指定的角色下的用户，该 ID 位于您公司的电子邮件域中的电子邮件地址所标识的 IAM 账户中。有关唯一 ID 的更多信息，请参阅 IAM 用户指南中的唯一 ID。

  您使用 IdP（在本例中为 Azure AD）进行的设置决定了条件子句的写入方式。如果员工的电子邮件是 johndoe@example.com，请首先将 ${redshift:DbUser} 设置为与

员工用户名 johndoe 匹配的超级用户字段。然后，要使该条件有效，请将 AWS SAML RoleSessionName 字段设置为与员工电子邮件 johndoe@example.com 匹配的超级用户字段。使用这种方法时，请考虑以下几点：

- 如果您将 ${redshift:DbUser} 设置为员工的电子邮件，则删除示例 JSON 中的 @example.com 以匹配 RoleSessionName。
- 如果您只是将 RoleSessionId 设置为员工的用户名，则删除示例中的 @example.com 以匹配 RoleSessionName。
- 在示例 JSON 中，${redshift:DbUser} 和 RoleSessionName 都已设置为员工电子邮件。此示例 JSON 使用 Amazon Redshift 数据库用户名以及 @example.com 让用户登录以访问集群。

- 第二部分授予在指定集群中创建 dbuser 名称的权限。在此示例 JSON 中，它将创建限制为 ${redshift:DbUser}。
- 第三部分授予指定用户可以加入 dbgroup 的权限。在此示例 JSON 中，用户可以加入指定集群中的 my_dbgroup 组。
- 第四部分授予用户可以对所有资源执行的操作的权限。在此示例 JSON 中，它允许用户调用 redshift:DescribeClusters 以获取集群信息，例如集群端点、AWS 区域和端口。它还允许用户调用 iam:ListRoles 以检查用户可以代入哪些角色。

第 2 步：设置 JDBC 或 ODBC，以便向 Azure 进行身份验证

JDBC

设置 JDBC 以在 Microsoft Azure AD 中进行身份验证

- 将数据库客户端配置为通过 JDBC 连接（使用 Azure AD 单点登录）到集群。

  您可以使用任何采用 JDBC 驱动程序的客户端通过 Azure AD 单点登录进行连接，也可以使用像 Java 这样的语言通过脚本进行连接。有关安装和配置信息，请参阅为 Amazon Redshift 配置 JDBC 驱动程序版本 2.x 连接。

  例如，您可以使用 SQLWorkbench/J 作为客户端。当您配置 SQLWorkbench/J 时，数据库的 URL 使用以下格式。

  ```
  jdbc:redshift:iam://cluster-identifier:us-west-1/dev
  ```

  如果您使用 SQLWorkbench/J 作为客户端，请执行以下步骤：

a. 启动 SQL Workbench/J。在选择连接配置文件页面上，添加一个称为 **AzureAuth** 的配置文件组。

b. 对于连接配置文件，请输入 **Azure**。

c. 选择管理驱动程序，然后选择 Amazon Redshift。选择库旁边的打开文件夹图标，然后选择适当的 JDBC .jar 文件。

d. 在选择连接配置文件页面上，向连接配置文件添加信息，如下所示：

- 对于用户，请输入 Microsoft Azure 用户名。这是您用于单点登录的 Microsoft Azure 账户的用户名，该账户有权限访问您在尝试进行身份验证时使用的集群。
- 对于密码，请输入您的 Microsoft Azure 密码。
- 对于驱动程序，请选择 Amazon Redshift (com.amazon.redshift.jdbc.Driver)。
- 对于 URL，请输入 **jdbc:redshift:iam://*your-cluster-identifier*:*your-cluster-region*/*your-database-name***。

e. 选择扩展属性以在连接属性中添加其他信息，如下所述。

对于 Azure AD 单点登录配置，请添加附加信息，如下所示：

- 对于 plugin_name，请输入 **com.amazon.redshift.plugin.AzureCredentialsProvider**。该值指定驱动程序将 Azure AD 单点登录作为身份验证方法。
- 对于 idp_tenant，请输入 ***your-idp-tenant***。仅用于 Microsoft Azure AD。这是在 Azure AD 上为您的公司配置的租户名称。此值可以是租户名或带连字符的租户唯一 ID。
- 对于 client_secret，请输入 ***your-azure-redshift-application-client-secret***。仅用于 Microsoft Azure AD。这是您在设置 Azure 单点登录配置时创建的 Amazon Redshift 应用程序的客户端密钥。这仅适用于 com.amazon.redshift.plugin.AzureCredentialsProvider 插件。
- 对于 client_id，请输入 ***your-azure-redshift-application-client-id***。仅用于 Microsoft Azure AD。这是您在设置 Azure 单点登录配置时创建的 Amazon Redshift 应用程序的客户端 ID（带连字符）。

对于具有 MFA 的 Azure AD 单点登录配置，请在连接属性中添加附加信息，如下所示：

- 对于 plugin_name，请输入 **com.amazon.redshift.plugin.BrowserAzureCredentialsProvider**。此值指定驱动程序将具有 MFA 的 Azure 单点登录作为身份验证方法。
- 对于 idp_tenant，请输入 ***your-idp-tenant***。仅用于 Microsoft Azure AD。这是在 Azure AD 上为您的公司配置的租户名称。此值可以是租户名或带连字符的租户唯一 ID。
- 对于 client_id，请输入 ***your-azure-redshift-application-client-id***。此选项仅用于 Microsoft Azure AD。这是您在设置具有 MFA 的 Azure AD 单点登录配置时创建的 Amazon Redshift 应用程序的客户端 ID（带连字符）。
- 对于侦听端口，请输入 ***your-listen-port***。这是本地服务器正在侦听的端口。默认值为 7890。
- 对于 idp_response_timeout，请输入 ***the-number-of-seconds***。这是 IdP 服务器发回响应时超时之前等待的秒数。最小秒数必须为 10。如果建立连接的用时长于此阈值，则连接将被中止。

ODBC

设置 ODBC 以在 Microsoft Azure AD 中进行身份验证

- 将数据库客户端配置为通过 ODBC 连接（使用 Azure AD 单点登录）到集群。

  Amazon Redshift 提供了适用于 Linux、Windows 和 macOS 操作系统的 ODBC 驱动程序。在安装 ODBC 驱动程序之前，请确定您的 SQL 客户端工具是 32 位还是 64 位。安装符合 SQL 客户端工具要求的 ODBC 驱动程序。

  在 Windows 上的 Amazon Redshift ODBC 驱动程序 DSN 安装页的连接设置下，输入以下信息：

  - 对于数据源名称，输入 ***your-DSN***。这将指定用作 ODBC 配置文件名称的数据源名称。
  - 对于 Azure AD 单点登录配置的身份验证类型，请选择 **Identity Provider: Azure AD**。这是 ODBC 驱动程序在通过 Azure 单点登录进行身份验证时使用的验证方法。
  - 对于具有 MFA 的 Azure AD 单点登录配置的身份验证类型，请选择 **Identity Provider: Browser Azure AD**。这是 ODBC 驱动程序在通过 Azure 单点登录（采用 MFA）进行身份验证时使用的验证方法。
  - 对于集群 ID，请输入 ***your-cluster-identifier***。

- 对于区域，请输入 ***your-cluster-region***。
- 对于数据库，请输入 ***your-database-name***。
- 对于用户，请输入 ***your-azure-username***。这是您用于单点登录的 Microsoft Azure 账户的用户名，该账户有权限访问您在尝试进行身份验证时使用的集群。请仅将其用于身份验证类型 为身份提供者: Azure AD 的情况。
- 对于密码，请输入 ***your-azure-password***。请仅将其用于 Auth Type (身份验证类型) 为 Identity Provider: Azure AD (身份提供者: Azure AD) 的情况。
- 对于 IdP 租户，请输入 ***your-idp-tenant***。这是在 IdP (Azure) 上配置的公司的租户名称。此值可以是租户名或带连字符的租户唯一 ID。
- 对于 Azure 客户端密钥，请输入 ***your-azure-redshift-application-client-secret***。这是您在设置 Azure 单点登录配置时创建的 Amazon Redshift 应用程序的客户端密钥。
- 对于 Azure 客户端 ID，请输入 ***your-azure-redshift-application-client-id***。这是您在设置 Azure 单点登录配置时创建的 Amazon Redshift 应用程序的客户端 ID（带连字符）。
- 对于侦听端口，请输入 ***your-listen-port***。这是本地服务器正在侦听的默认侦听端口。默认值为 7890。这仅适用于浏览器 Azure AD 插件。
- 对于响应超时，请输入 ***the-number-of-seconds***。这是 IdP 服务器发回响应时超时之前等待的秒数。最小秒数必须为 10。如果建立连接的用时长于此阈值，则连接将被中止。该选项仅适用于浏览器 Azure AD 插件。

在 macOS 和 Linux 上，按如下方式编辑 `odbc.ini` 文件：

> **Note**
>
> 所有条目不区分大小写。

- 对于 clusterid，请输入 ***your-cluster-identifier***。这是已创建的 Amazon Redshift 集群的名称。
- 对于区域，请输入 ***your-cluster-region***。这是已创建的 Amazon Redshift 集群的 AWS 区域。
- 对于数据库，请输入 ***your-database-name***。这是您尝试在 Amazon Redshift 集群上访问的数据库的名称。

- 对于区域设置，请输入 **en-us**。这是显示错误消息的语言。
- 对于 iam，请输入 **1**。此值指定要使用 IAM 凭证进行身份验证的驱动程序。
- 对于 Azure AD 单点登录配置的 plugin_name，请输入 **AzureAD**。这指定驱动程序使用 Azure 单点登录作为身份验证方法。
- 对于具有 MFA 的 Azure AD 单点登录配置的 plugin_name，请输入 **BrowserAzureAD**。这指定驱动程序将 Azure 单点登录（采用 MFA）作为身份验证方法。
- 对于 uid，请输入 ***your-azure-username***。这是您用于单点登录的 Microsoft Azure 账户的用户名，该账户有权限访问您在尝试进行身份验证时使用的集群。请仅将其用于 plugin_name 为 AzureAD 的情况。
- 对于密码，请输入 ***your-azure-password***。请仅将其用于 plugin_name 为 AzureAD 的情况。
- 对于 idp_tenant，请输入 ***your-idp-tenant***。这是在 IdP (Azure) 上配置的公司的租户名称。此值可以是租户名或带连字符的租户唯一 ID。
- 对于 client_secret，请输入 ***your-azure-redshift-application-client-secret***。这是您在设置 Azure 单点登录配置时创建的 Amazon Redshift 应用程序的客户端密钥。
- 对于 client_id，请输入 ***your-azure-redshift-application-client-id***。这是您在设置 Azure 单点登录配置时创建的 Amazon Redshift 应用程序的客户端 ID（带连字符）。
- 对于侦听端口，请输入 ***your-listen-port***。这是本地服务器正在侦听的端口。默认值为 7890。这适用于浏览器 Azure AD 插件。
- 对于 idp_response_timeout，请输入 ***the-number-of-seconds***。这是等待 Azure 响应的指定时间段（以秒为单位）。此选项适用于浏览器 Azure AD 插件。

在 macOS 和 Linux 上，还要编辑配置文件设置以添加以下导出。

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

### 问题排查

要解决浏览器 Azure AD 插件的问题，请考虑以下几点。

- 要使用浏览器 Azure AD 插件，在请求中指定的回复 URL 必须与为应用程序配置的回复 URL 一致。导航到 Microsoft Azure 门户上的使用 SAML 设置单点登录页面。然后检查回复 URL 是否已设置为 http://localhost/redshift/。
- 如果收到 IdP 租户错误，请验证 IdP 租户名称是否与最初在 Microsoft Azure 中设置 Active Directory 时所用的域名匹配。

  在 Windows 上，导航到 Amazon Redshift ODBC DSN 设置页面的连接设置部分。然后检查在 IdP (Azure) 上配置的公司租户名称是否与最初在 Microsoft Azure 中设置 Active Directory 时所用的域名匹配。

  在 macOS 和 Linux 上，找到 odbc.ini 文件。然后检查在 IdP (Azure) 上配置的公司租户名称是否与最初在 Microsoft Azure 中设置 Active Directory 时所用的域名匹配。
- 如果收到请求中指定的回复 URL 与为应用程序配置的回复 URL 不一致的错误，请验证重定向 URI 是否与回复 URL 相同。

  在 Microsoft Azure 门户上，导航到您的应用程序的应用程序注册页面。然后检查重定向 URI 是否与回复 URL 一致。
- 如果收到“未经授权错误”意外响应，请验证是否已完成移动和桌面应用程序配置。

  在 Microsoft Azure 门户上，导航到您的应用程序的应用程序注册页面。然后导航到身份验证，检查是否已将移动和桌面应用程序配置为使用 http://localhost/redshift/ 作为重定向 URI。

Ping Identity

您可以将 Ping Identity 作为身份提供者 (IdP) 以访问 Amazon Redshift 集群。本教程将向您展示如何将 Ping Identity 用作身份提供者（IdP）来访问 Amazon Redshift 集群。

第 1 步：设置 Ping Identity 和您的 AWS 账户相互信任

以下过程介绍如何使用 PingOne 门户建立信任关系。

将 Ping 身份和您的 AWS 账户设置为相互信任

1. 创建或使用现有的 Amazon Redshift 集群，以使 Ping 身份用户连接到该集群。要配置连接，需要此集群的某些属性，例如集群标识符。有关更多信息，请参阅创建集群。
2. 在 PingOne 门户上添加 Amazon Redshift 以作为新的 SAML 应用程序。有关详细步骤，请参阅 Ping Identity 文档。

   1. 转到我的应用程序。

2. 在添加应用程序下面，选择新建 SAML 应用程序。
3. 对于应用程序名称，请输入 **`Amazon Redshift`**。
4. 对于协议版本，请选择 SAML v2.0。
5. 对于类别，请选择 ***`your-application-category`***。
6. 对于断言消费者服务(ACS)，请键入 ***`your-redshift-local-host-url`***。这是 SAML 断言重定向到的本地主机和端口。
7. 对于实体 ID，请输入 `urn:amazon:webservices`。
8. 对于签名，请选择签名断言。
9. 在 SSO 属性映射部分中，创建声明，如下表中所示。

| 应用程序属性 | 文本值的身份关联属性 |
| --- | --- |
| https://aws.amazon.com/SAML/Attributes/Role | arn:aws:iam::*123456789012* :role/*Ping*,arn:aws:iam::*123456789012* :saml-provider/*PingProvider* |
| https://aws.amazon.com/SAML/Attributes/RoleSessionName | email |
| https://redshift.amazon.com/SAML/Attributes/AutoCreate | "true" |
| https://redshift.amazon.com/SAML/Attributes/DbUser | email |
| https://redshift.amazon.com/SAML/Attributes/DbGroups | “DbGroups”属性中的组包含 @directory 前缀。若要删除此项，请在身份关联中，输入 memberOf。在函数中，选择 ExtractByRegularExpression。在表达式中，输入 (.*)[\@](?:.*)。 |

3. 对于组访问权限，请设置以下组访问权限（如果需要）：

- https://aws.amazon.com/SAML/Attributes/Role
- https://aws.amazon.com/SAML/Attributes/RoleSessionName
- https://redshift.amazon.com/SAML/Attributes/AutoCreate

- https://redshift.amazon.com/SAML/Attributes/DbUser

4. 查看设置，并在必要时进行更改。
5. 将启动单点登录(SSO) URL 作为浏览器 SAML 插件的登录 URL。
6. 在 IAM 控制台上创建 IAM SAML 身份提供者。您提供的元数据文档是您在设置 Ping Identity 时保存的联合元数据 XML 文件。有关详细步骤，请参阅 IAM 用户指南中的创建和管理 IAM 身份提供者（控制台）。
7. 在 IAM 控制台上为 SAML 2.0 联合身份创建 IAM 角色。有关详细步骤，请参阅 IAM 用户指南中的创建用于 SAML 的角色。
8. 创建一个 IAM 策略，您可以将其附加到您在 IAM 控制台上为 SAML 2.0 联合身份验证创建的 IAM 角色。有关详细步骤，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。有关 Azure AD 示例，请参阅设置 JDBC 或 ODBC 单点登录身份验证。

第 2 步：设置 JDBC 或 ODBC，以便向 Ping Identity 进行身份验证

JDBC

设置 JDBC 以在 Ping Identity 中进行身份验证

- 配置数据库客户端以使用 Ping Identity 单点登录通过 JDBC 连接到集群。

  您可以使用任何采用 JDBC 驱动程序的客户端通过 Ping Identity 单点登录进行连接，也可以使用像 Java 这样的语言通过脚本进行连接。有关安装和配置信息，请参阅为 Amazon Redshift 配置 JDBC 驱动程序版本 2.x 连接。

  例如，您可以使用 SQLWorkbench/J 作为客户端。当您配置 SQLWorkbench/J 时，数据库的 URL 使用以下格式。

  ```
  jdbc:redshift:iam://cluster-identifier:us-west-1/dev
  ```

  如果您使用 SQLWorkbench/J 作为客户端，请执行以下步骤：

  a. 启动 SQL Workbench/J。在选择连接配置文件页面中，添加一个配置文件组，例如，**Ping**。
  b. 对于连接配置文件，请输入 ***your-connection-profile-name***，例如，**Ping**。
  c. 选择管理驱动程序，然后选择 Amazon Redshift。选择库旁边的打开文件夹图标，然后选择适当的 JDBC .jar 文件。

d. 在选择连接配置文件页面上，向连接配置文件添加信息，如下所示：

- 对于用户，请输入您的 PingOne 用户名。这是您用于单点登录的 PingOne 账户的用户名，该账户有权限访问您在尝试进行身份验证时使用的集群。
- 对于密码，请输入您的 PingOne 密码。
- 对于驱动程序，请选择 Amazon Redshift (com.amazon.redshift.jdbc.Driver)。
- 对于 URL，请输入 **jdbc:redshift:iam://*your-cluster-identifier*:*your-cluster-region*/*your-database-name***。

e. 选择扩展属性并执行以下操作之一：

- 对于 login_url，请输入 ***your-ping-sso-login-url***。该值指定 URL 将单点登录用作登录时的身份验证方法。
- 对于 Ping Identity，请为 plugin_name 输入 **com.amazon.redshift.plugin.PingCredentialsProvider**。此值指定驱动程序使用 Ping Identity 单点登录作为身份验证方法。
- 对于具有单点登录的 Ping Identity，请为 plugin_name 输入 **com.amazon.redshift.plugin.BrowserSamlCredentialsProvider**。该值指定驱动程序将具有单点登录的 Ping Identity PingOne 用作身份验证方法。

ODBC

设置 ODBC 以在 Ping Identity 中进行身份验证

- 配置数据库客户端以使用 Ping Identity PingOne 单点登录通过 ODBC 连接到集群。

  Amazon Redshift 提供了适用于 Linux、Windows 和 macOS 操作系统的 ODBC 驱动程序。在安装 ODBC 驱动程序之前，请确定您的 SQL 客户端工具是 32 位还是 64 位。安装符合 SQL 客户端工具要求的 ODBC 驱动程序。

  在 Windows 上的 Amazon Redshift ODBC 驱动程序 DSN 安装页的连接设置下，输入以下信息：

  - 对于数据源名称，输入 ***your-DSN***。这将指定用作 ODBC 配置文件名称的数据源名称。
  - 对于身份验证类型，请执行以下操作之一：
    - 对于 Ping Identity 配置，请选择身份提供者：Ping Federate。这是 ODBC 驱动程序在通过 Ping Identity 单点登录进行身份验证时使用的身份验证方法。

  - 对于具有单点登录的 Ping Identity 配置，请选择身份提供者：浏览器 SAML。这是 ODBC 驱动程序在使用具有单点登录的 Ping Identity 进行身份验证时使用的身份验证方法。
- 对于集群 ID，请输入 ***your-cluster-identifier***。
- 对于区域，请输入 ***your-cluster-region***。
- 对于数据库，请输入 ***your-database-name***。
- 对于用户，请输入 ***your-ping-username***。这是您用于单点登录的 PingOne 账户的用户名，该账户有权访问您在尝试进行身份验证时使用的集群。请仅将其用于身份验证类型为身份提供者: PingFederate的情况。
- 对于密码，请输入 ***your-ping-password***。请仅将其用于 Auth type (身份验证类型) 为 Identity Provider: PingFederate (身份提供者: PingFederate) 的情况。
- 对于侦听端口，请输入 ***your-listen-port***。这是本地服务器正在侦听的端口。默认值为 7890。这仅适用于浏览器 SAML 插件。
- 对于响应超时，请输入 ***the-number-of-seconds***。这是 IdP 服务器发回响应时超时之前等待的秒数。最小秒数必须为 10。如果建立连接的用时长于此阈值，则连接将被中止。这仅适用于浏览器 SAML 插件。
- 对于登录 URL，请输入 ***your-login-url***。这仅适用于浏览器 SAML 插件。

在 macOS 和 Linux 上，按如下方式编辑 `odbc.ini` 文件：

> **Note**
>
> 所有条目不区分大小写。

- 对于 clusterid，请输入 ***your-cluster-identifier***。这是已创建的 Amazon Redshift 集群的名称。
- 对于区域，请输入 ***your-cluster-region***。这是已创建的 Amazon Redshift 集群的 AWS 区域。
- 对于数据库，请输入 ***your-database-name***。这是您尝试在 Amazon Redshift 集群上访问的数据库的名称。
- 对于区域设置，请输入 **en-us**。这是显示错误消息的语言。
- 对于 iam，请输入 **1**。此值指定要使用 IAM 凭证进行身份验证的驱动程序。
- 对于 plugin_name，请执行以下操作之一：

  - 对于 Ping Identity 配置，请输入 **`BrowserSAML`**。这是 ODBC 驱动程序在使用 Ping Identity 进行身份验证时所采用的身份验证方法。
  - 对于具有单点登录的 Ping Identity 配置，请输入 **`Ping`**。这是 ODBC 驱动程序在使用具有单点登录的 Ping Identity 进行身份验证时使用的身份验证方法。
- 对于 uid，请输入 ***`your-ping-username`***。这是您用于单点登录的 Microsoft Azure 账户的用户名，该账户有权限访问您在尝试进行身份验证时使用的集群。请仅将其用于 plugin_name 为 Ping 的情况。
- 对于密码，请输入 ***`your-ping-password`***。请仅将其用于 plugin_name 为 Ping 的情况。
- 对于 login_url，请输入 ***`your-login-url`***。这是返回 SAML 响应的启动单点登录 URL。这仅适用于浏览器 SAML 插件。
- 对于 idp_response_timeout，请输入 ***`the-number-of-seconds`***。这是等待 PingOne Identity 响应的指定时间段（以秒为单位）。这仅适用于浏览器 SAML 插件。
- 对于侦听端口，请输入 ***`your-listen-port`***。这是本地服务器正在侦听的端口。默认值为 7890。这仅适用于浏览器 SAML 插件。

在 macOS 和 Linux 上，还要编辑配置文件设置以添加以下导出。

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

Okta

您可以将 Okta 作为身份提供者 (IdP) 以访问 Amazon Redshift 集群。本教程将向您展示如何将 Okta 用作身份提供者（IdP）来访问 Amazon Redshift 集群。

第 1 步：设置 Okta 和您的 AWS 账户相互信任

以下过程介绍如何建立信任关系。

将 Okta 和您的 AWS 账户设置为相互信任

1. 创建或使用现有的 Amazon Redshift 集群，以使 Okta 用户连接到该集群。要配置连接，需要此集群的某些属性，例如集群标识符。有关更多信息，请参阅创建集群。
2. 在 Okta 门户上添加 Amazon Redshift 以作为新应用程序。有关详细步骤，请参阅 Okta 文档。

- 选择添加应用程序。
- 在添加应用程序下面，选择创建新的应用程序。
- 在创建新添加应用程序集成页面上，为平台 选择 Web。
- 对于登录方法，请选择 SAML v2.0。
- 在常规设置页面上，为应用程序名称 输入 ***your-redshift-saml-sso-name***。这是您的应用程序的名称。
- 在 SAML 设置页面上，为单点登录 URL 输入 ***your-redshift-local-host-url***。这是 SAML 断言重定向到的本地主机和端口，例如 `http://localhost:7890/redshift/`。

3. 将单点登录 URL 作为收件人 URL 和目标 URL。
4. 对于签名，请选择签名断言。
5. 对于受众 URI (SP 实体 ID)，为声明输入 **`urn:amazon:webservices`**，如下表中所示。
6. 在高级设置部分中，为 SAML 颁发者 ID 输入 ***your-Identity-Provider-Issuer-ID***，您可以在查看设置说明部分中找到该 ID。
7. 在属性语句部分中，创建声明，如下表中所示。

| 声明名称 | 值 |
| --- | --- |
| https://aws.amazon.com/SAML/Attributes/Role | arn:aws:iam::*123456789012* :role/*Okta*,arn:aws:iam::*123456789012* :saml-provider/*Okta* |
| https://aws.amazon.com/SAML/Attributes/RoleSessionName | user.email |
| https://redshift.amazon.com/SAML/Attributes/AutoCreate | "true" |
| https://redshift.amazon.com/SAML/Attributes/DbUser | user.email |

8. 在应用程序嵌入式链接部分中，找到可用作浏览器 SAML 插件登录 URL 的 URL。
9. 在 IAM 控制台上创建 IAM SAML 身份提供者。您提供的元数据文档是您在设置 Okta 时保存的联合元数据 XML 文件。有关详细步骤，请参阅 IAM 用户指南中的创建和管理 IAM 身份提供者（控制台）。

10. 在 IAM 控制台上为 SAML 2.0 联合身份创建 IAM 角色。有关详细步骤，请参阅 IAM 用户指南中的创建用于 SAML 的角色。
11. 创建一个 IAM 策略，您可以将其附加到您在 IAM 控制台上为 SAML 2.0 联合身份验证创建的 IAM 角色。有关详细步骤，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。有关 Azure AD 示例，请参阅设置 JDBC 或 ODBC 单点登录身份验证。

第 2 步：设置 JDBC 或 ODBC，以便向 Okta 进行身份验证

JDBC

设置 JDBC 以在 Okta 中进行身份验证

- 将数据库客户端配置为通过 JDBC 并使用 Okta 单点登录连接到集群。

  您可以使用任何采用 JDBC 驱动程序的客户端通过 Okta 单点登录进行连接，也可以使用像 Java 这样的语言通过脚本进行连接。有关安装和配置信息，请参阅为 Amazon Redshift 配置 JDBC 驱动程序版本 2.x 连接。

  例如，您可以使用 SQLWorkbench/J 作为客户端。当您配置 SQLWorkbench/J 时，数据库的 URL 使用以下格式。

  ```
  jdbc:redshift:iam://cluster-identifier:us-west-1/dev
  ```

  如果您使用 SQLWorkbench/J 作为客户端，请执行以下步骤：

  a. 启动 SQL Workbench/J。在选择连接配置文件页面中，添加一个配置文件组，例如，**Okta**。
  b. 对于连接配置文件，请输入 ***your-connection-profile-name***，例如，**Okta**。
  c. 选择管理驱动程序，然后选择 Amazon Redshift。选择库旁边的打开文件夹图标，然后选择适当的 JDBC .jar 文件。
  d. 在选择连接配置文件页面上，向连接配置文件添加信息，如下所示：

     - 对于用户，请输入您的 Okta 用户名。这是您用于单点登录的 Okta 账户的用户名，该账户有权限访问您在尝试进行身份验证时使用的集群。
     - 对于密码，请输入您的 Okta 密码。
     - 对于驱动程序，请选择 Amazon Redshift (com.amazon.redshift.jdbc.Driver)。

- 对于 URL，请输入 **jdbc:redshift:iam://*your-cluster-identifier*:*your-cluster-region*/*your-database-name***。

e. 选择扩展属性并执行以下操作之一：

- 对于 login_url，请输入 ***your-okta-sso-login-url***。该值指定 URL 将单点登录作为身份验证方法以登录到 Okta。
- 对于 Okta 单点登录，请为 plugin_name 输入 **com.amazon.redshift.plugin.OktaCredentialsProvider**。此值指定驱动程序使用 Okta 单点登录作为身份验证方法。
- 对于具有 MFA 的 Okta 单点登录，请为 plugin_name 输入 **com.amazon.redshift.plugin.BrowserSamlCredentialsProvider**。此值指定驱动程序将具有 MFA 的 Okta 单点登录作为身份验证方法。

ODBC

设置 ODBC 以在 Okta 中进行身份验证

- 将数据库客户端配置为通过 ODBC 并使用 Okta 单点登录连接到集群。

  Amazon Redshift 提供了适用于 Linux、Windows 和 macOS 操作系统的 ODBC 驱动程序。在安装 ODBC 驱动程序之前，请确定您的 SQL 客户端工具是 32 位还是 64 位。安装符合 SQL 客户端工具要求的 ODBC 驱动程序。

  在 Windows 上的 Amazon Redshift ODBC 驱动程序 DSN 安装页的连接设置下，输入以下信息：

  - 对于数据源名称，输入 ***your-DSN***。这将指定用作 ODBC 配置文件名称的数据源名称。
  - 对于身份验证类型，请执行以下操作之一：
    - 对于 Okta 单点登录配置，请选择 **Identity Provider: Okta**。这是 ODBC 驱动程序在通过 Okta 单点登录进行身份验证时使用的身份验证方法。
    - 对于具有 MFA 的 Okta 单点登录配置，请选择 **Identity Provider: Browser SAML**。这是 ODBC 驱动程序在通过具有 MFA 的 Okta 单点登录进行身份验证时使用的身份验证方法。
  - 对于集群 ID，请输入 ***your-cluster-identifier***。
  - 对于区域，请输入 ***your-cluster-region***。
  - 对于数据库，请输入 ***your-database-name***。

- 对于用户，请输入 ***your-okta-username***。这是您用于单点登录的 Okta 账户的用户名，该账户有权访问您在尝试进行身份验证时使用的集群。请仅将其用于身份验证类型为身份提供者: Okta 的情况。
- 对于密码，请输入 ***your-okta-password***。请仅将其用于 Auth type (身份验证类型) 为 Identity Provider: Okta (身份提供者: Okta) 的情况。

在 macOS 和 Linux 上，按如下方式编辑 `odbc.ini` 文件：

> **Note**
>
> 所有条目不区分大小写。

- 对于 clusterid，请输入 ***your-cluster-identifier***。这是已创建的 Amazon Redshift 集群的名称。
- 对于区域，请输入 ***your-cluster-region***。这是已创建的 Amazon Redshift 集群的 AWS 区域。
- 对于数据库，请输入 ***your-database-name***。这是您尝试在 Amazon Redshift 集群上访问的数据库的名称。
- 对于区域设置，请输入 **`en-us`**。这是显示错误消息的语言。
- 对于 iam，请输入 **`1`**。此值指定要使用 IAM 凭证进行身份验证的驱动程序。
- 对于 plugin_name，请执行以下操作之一：
  - 对于具有 MFA 的 Okta 单点登录配置，请输入 **`BrowserSAML`**。这是 ODBC 驱动程序在通过具有 MFA 的 Okta 单点登录进行身份验证时使用的身份验证方法。
  - 对于 Okta 单点登录配置，请输入 **`Okta`**。这是 ODBC 驱动程序在通过 Okta 单点登录进行身份验证时使用的身份验证方法。
- 对于 uid，请输入 ***your-okta-username***。这是您用于单点登录的 Okta 账户的用户名，该账户有权访问您在尝试进行身份验证时使用的集群。请仅将其用于 plugin_name 为 Okta 的情况。
- 对于密码，请输入 ***your-okta-password***。请仅将其用于 plugin_name 为 Okta 的情况。
- 对于 login_url，请输入 ***your-login-url***。这是返回 SAML 响应的启动单点登录 URL。这仅适用于浏览器 SAML 插件。
- 对于 idp_response_timeout，请输入 ***the-number-of-seconds***。这是等待 PingOne 响应的指定时间段（以秒为单位）。这仅适用于浏览器 SAML 插件。

- 对于侦听端口，请输入 ***your-listen-port***。这是本地服务器正在侦听的端口。默认值为 7890。这仅适用于浏览器 SAML 插件。

在 macOS 和 Linux 上，还要编辑配置文件设置以添加以下导出。

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

## 授权 Amazon Redshift 代表您访问 AWS 服务

某些 Amazon Redshift 功能要求 Amazon Redshift 代表您访问其他 AWS 服务。例如，COPY 和 UNLOAD 命令可使用 Simple Storage Service（Amazon S3）存储桶将数据加载或卸载到您的 Amazon Redshift 集群中。CREATE EXTERNAL FUNCTION 命令可以使用标量 Lambda 用户定义的函数 (UDF) 调用 AWS Lambda 函数。Amazon Redshift Spectrum 可以在 Amazon Athena 或 AWS Glue 中使用数据目录。要让您的 Amazon Redshift 集群代表您执行操作，请为这些集群提供安全凭证。提供安全凭证的首选方法是指定一个 AWS Identity and Access Management (IAM) 角色。对于 COPY 和 UNLOAD，您可以提供临时凭证。

如果用户需要在 AWS 管理控制台之外与 AWS 交互，则需要编程式访问权限。授予编程式访问权限的方法取决于访问 AWS 的用户类型。

要向用户授予编程式访问权限，请选择以下选项之一。

| 哪个用户需要编程式访问权限？ | 目的 | 方式 |
| --- | --- | --- |
| IAM | （推荐）使用控制台凭证作为临时凭证来签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照您希望使用的界面的说明进行操作。<br>• 对于 AWS CLI，请参阅《AWS Command Line Interface用户指南》中的登录 AWS 以进行本地开发。<br>• 对于 AWS SDK，请参阅《AWS SDK 和工具参考指 |

| 哪个用户需要编程式访问权限？ | 目的 | 方式 |
| --- | --- | --- |
| | | 南》中的登录 AWS 以进行本地开发。 |
| 人力身份<br><br>（在 IAM Identity Center 中管理的用户） | 使用临时凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照您希望使用的界面的说明进行操作。<br><br>• 有关 AWS CLI 的更多信息，请参阅《AWS Command Line Interface 用户指南》中的配置 AWS CLI 以使用 AWS IAM Identity Center。<br>• 有关 AWS SDK、工具和 AWS API 的更多信息，请参阅《AWS SDK 和工具参考指南》中的 IAM Identity Center 身份验证。 |
| IAM | 使用临时凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照《IAM 用户指南》中将临时凭证用于 AWS 资源中的说明进行操作。 |

| 哪个用户需要编程式访问权限？ | 目的 | 方式 |
| --- | --- | --- |
| IAM | （不推荐使用）<br>使用长期凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照您希望使用的界面的说明进行操作。<br>• 有关 AWS CLI 的更多信息，请参阅《AWS Command Line Interface 用户指南》中的使用 IAM 用户凭证进行身份验证。<br>• 有关 AWS SDK 和工具的更多信息，请参阅《AWS SDK 和工具参考指南》中的使用长期凭证进行身份验证。<br>• 有关 AWS API 的更多信息，请参阅《IAM 用户指南》中的管理 IAM 用户的访问密钥。 |

接下来，了解如何创建具有访问其他 AWS 服务的适当权限的 IAM 角色。当您执行 Amazon Redshift 命令时，还需要将该角色与您的集群关联并指定角色的 Amazon Resource Name (ARN)。有关更多信息，请参阅 使用 IAM 角色授权 COPY、UNLOAD、CREATE EXTERNAL FUNCTION 和 CREATE EXTERNAL SCHEMA 操作。

此外，超级用户还可以向特定用户和组授予 ASSUMEROLE 权限，以便为 COPY 和 UNLOAD 操作提供对角色的访问权限。有关更多信息，请参阅 Amazon Redshift 数据库开发人员指南中的 GRANT。

## 创建 IAM 角色以允许 Amazon Redshift 集群访问 AWS 服务

## 创建 IAM 角色以允许 Amazon Redshift 集群访问 AWS 服务

要创建 IAM 角色以允许您的 Amazon Redshift 集群代表您与其他 AWS 服务通信，请执行以下步骤。本节中使用的值是示例，您可以根据需要选择值。

要创建 IAM 角色以允许 Amazon Redshift 访问 AWS 服务

1. 打开 IAM 控制台。

2. 在导航窗格中，选择角色。
3. 选择 Create role（创建角色）。
4. 选择 AWS 服务，然后选择 Redshift。
5. 在 Select your use case 下，选择 Redshift - Customizable，然后选择 Next: Permissions。此时显示 Attach permissions policy 页面。
6. 对于使用 COPY 访问 Simple Storage Service（Amazon S3），作为示例，您可以使用 **AmazonS3ReadOnlyAccess** 并附加。要使用 COPY 或 UNLOAD 访问 Simple Storage Service（Amazon S3），我们建议您创建托管式策略，以相应地限制对所需存储桶和前缀的访问。对于读取和写入操作，我们建议强制执行最低权限，并仅限于 Amazon Redshift 要求的 Simple Storage Service（Amazon S3）存储桶和键前缀。

   要想为 CREATE EXTERNAL FUNCTION 命令调用 Lambda 函数，请添加 **AWSLambdaRole**。

   对于 Redshift Spectrum，除 Simple Storage Service（Amazon S3）访问以外，添加 **AWSGlueConsoleFullAccess** 或 **AmazonAthenaFullAccess**。

   选择 下一步：标签。
7. 此时将显示添加标签页面。您可以选择性地添加标签。选择下一步：审核。
8. 对于角色名称，键入一个角色名称，例如 **RedshiftCopyUnload**。选择创建角色。
9. 使用新角色的集群中的所有用户都可使用该角色。如需限制访问，只允许特定集群中的特定用户、或特定区域中的集群访问，请编辑该角色的信任关系。有关更多信息，请参阅 限制对 IAM 角色的访问。
10. 将角色与您的集群关联。您可以在创建集群时关联 IAM 角色，或将角色添加到现有集群中。有关更多信息，请参阅 将 IAM 角色与集群相关联。

> **Note**
>
> 要限制对特定数据的访问，请使用授予所需最少权限的 IAM 角色。

## 限制对 IAM 角色的访问

预设情况下，对 Amazon Redshift 集群可用的 IAM 角色对集群上的所有用户都可用。您可选择将 IAM 角色限制为特定集群上的特定 Amazon Redshift 数据库用户，或限制为特定区域。

要仅允许特定数据库用户使用 IAM 角色，请执行以下步骤。

标识对 IAM 角色具有访问权限的特定数据库用户

1. 标识您的 Amazon Redshift 集群中的数据库用户的 Amazon Resource Name (ARN)。数据库用户的 ARN 采用以下格式：`arn:aws:redshift:region:account-id:dbuser:cluster-name/user-name`。

   对于 Amazon Redshift Serverless，请使用以下 ARN 格式：`arn:aws:redshift:region:account-id:dbuser:serverless-account-id-workgroup-id/user-name`。

2. 打开 IAM 控制台。
3. 在导航窗格中，选择角色。
4. 选择要限制到特定 Amazon Redshift 数据库用户的 IAM 角色。
5. 选择 Trust Relationships 选项卡，然后选择 Edit Trust Relationship。允许 Amazon Redshift 代表您访问其他 AWS 服务的新 IAM 角色具有以下信任关系：

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. 向信任关系的 `sts:AssumeRole` 操作部分添加一个条件以将 `sts:ExternalId` 字段限制为您指定的值。为您要授予对 IAM 角色的访问权限的每个数据库用户包含一个 ARN。外部 ID 可以是任何唯一的字符串。

   例如，以下信任关系指定只有区域 `user1` 中的集群 `user2` 上的数据库用户 `my-cluster` 和 `us-west-2` 有权使用此 IAM 角色。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "redshift.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "sts:ExternalId": [
          "arn:aws:redshift:us-west-2:123456789012:dbuser:my-cluster/user1",
          "arn:aws:redshift:us-west-2:123456789012:dbuser:my-cluster/user2"
        ]
      }
    }
  }]
}
```

7. 选择 Update Trust Policy。

## 将 IAM 角色限制为某个 AWS 区域

您可将 IAM 角色限制为仅在某个特定 AWS 区域中可访问。预设情况下，Amazon Redshift 的 IAM 角色不会限制到任何单一区域。

要按区域限制对 IAM 角色的使用，请执行以下步骤。

为 IAM 角色标识允许的区域

1. 通过以下网址打开 IAM 控制台：https://console.aws.amazon.com/。
2. 在导航窗格中，选择 Roles（角色）。
3. 选择要用特定区域修改的角色。
4. 选择 Trust Relationships (信任关系) 选项卡，然后选择 Edit Trust Relationship (编辑信任关系)。允许 Amazon Redshift 代表您访问其他 AWS 服务的新 IAM 角色具有以下信任关系：

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

5. 使用您要允许对其使用角色的特定区域的列表修改 Service 的 Principal 列表。Service 列表中的每个区域都必须采用以下格式：redshift.*region*.amazonaws.com。

   例如，以下编辑过的信任关系仅允许在 us-east-1 和 us-west-2 区域中使用 IAM 角色。

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "redshift.us-east-1.amazonaws.com",
          "redshift.us-west-2.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. 选择 Update Trust Policy

## 在 Amazon Redshift 中串联 IAM 角色

当您将角色附加到集群时，集群可以代入该角色以您的名义访问 Simple Storage Service（Amazon S3）、Amazon Athena、AWS Glue 和 AWS Lambda。如果附加到集群的角色无法访问必要的资源，则可以串联到另一个角色 (可能属于其他账户)。然后，您的集群临时代入串联的角色来访问数据。您还可以通过串联角色来授予跨账户访问权限。链中的每个角色都会代入链中的下一个角色，直到集群承担位于链尾的角色。您可以关联的最大 IAM 角色数量受配额限制。有关更多信息，请参阅 Amazon Redshift 对象的配额中的“Amazon Redshift 用于访问其他 AWS 服务的集群 IAM 角色”。

> **Note**
>
> 您必须指定 IAM 角色才能使链正常运行。

例如，假设公司 A 想要访问属于公司 B 的 Simple Storage Service（Amazon S3）存储桶中的数据。公司 A 为 Amazon Redshift 创建一个名为 `RoleA` 的 AWS 服务角色并将其附加到集群上。公司 B 创建一个名为 `RoleB` 的角色，该角色有权访问公司 B 存储桶中的数据。要访问公司 B 存储桶中的数据，公司 A 需要使用串联 `iam_role` 和 `RoleA` 的 `RoleB` 参数运行 COPY 命令。在 COPY 操作的持续时间内，`RoleA` 将临时代入 `RoleB` 以访问 Simple Storage Service（Amazon S3）存储桶。

要串联角色，您可以在角色之间建立信任关系。代入其他角色的角色 (例如，`RoleA`) 必须具有允许其代入下一个串联的角色 (例如，`RoleB`) 的权限策略。反过来，传递权限的角色 (`RoleB`) 必须具有允许其将权限传递给上一个串联的角色 (`RoleA`) 的信任策略。有关更多信息，请参阅 IAM 用户指南中的使用 IAM 角色。

链中的第一个角色必须是附加到集群的角色。第一个角色以及代入链中下一个角色的每个后续角色都必须具有包含特定语句的策略。该语句对 `Allow`操作以及 `sts:AssumeRole` 元素中的下一个角色的 Amazon Resource Name (ARN) 具有 `Resource` 效果。在我们的示例中，`RoleA` 具有允许其代入由 AWS 账户 `210987654321` 所有的 `RoleB` 的以下权限策略。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1487639602000",
            "Effect": "Allow",
            "Action": [
```

```
                "sts:AssumeRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/RoleB"
        }
    ]
}
```

传递给其他角色的角色必须与代入该角色的角色或拥有该角色的 AWS 账户建立信任关系。在我们的示例中，RoleB 具有与 RoleA 建立信任关系的以下信任策略。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/RoleA"
      }
    }
  ]
}
```

以下信任策略与 RoleA、AWS 账户 123456789012 的拥有者建立信任关系。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      }
    }
```

```
  ]
}
```

> **Note**
>
> 要将角色串联授权限制给特定用户，请定义条件。有关更多信息，请参阅 限制对 IAM 角色的访问。

当您运行 UNLOAD、COPY、CREATE EXTERNAL FUNCTION 或 CREATE EXTERNAL SCHEMA 命令时，可以通过在 iam_role 参数中包括一个逗号分隔的角色 ARN 列表来串联角色。下面显示了在 iam_role 参数中串联角色的语法。

```
unload ('select * from venue limit 10')
to 's3://acmedata/redshift/venue_pipe_'
IAM_ROLE 'arn:aws:iam::<aws-account-id-1>:role/<role-name-1>[,arn:aws:iam::<aws-account-id-2>:role/<role-name-2>][,...]';
```

> **Note**
>
> 整个角色链用单引号括起来，不能包含空格。

在以下示例中，RoleA 将附加到属于 AWS 账户 123456789012 的集群。属于账户 210987654321 的 RoleB 具有访问名为 s3://companyb/redshift/ 的存储桶的权限。以下示例将 RoleA 和 RoleB 进行串联以将数据卸载到 s3://companyb/redshift/ 存储桶。

```
unload ('select * from venue limit 10')
to 's3://companyb/redshift/venue_pipe_'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

以下示例使用 COPY 命令加载已在上一个示例中卸载的数据。

```
copy venue
from 's3://companyb/redshift/venue_pipe_'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

在以下示例中，CREATE EXTERNAL SCHEMA 使用串联的角色代入角色 RoleB。

```
create external schema spectrumexample from data catalog
database 'exampledb' region 'us-west-2'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

在以下示例中，CREATE EXTERNAL FUNCTION 使用串联的角色代入角色 RoleB。

```
create external function lambda_example(varchar)
returns varchar
volatile
lambda 'exampleLambdaFunction'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

## 使用 IAM 角色授权 COPY、UNLOAD、CREATE EXTERNAL FUNCTION 和 CREATE EXTERNAL SCHEMA 操作

您可使用 COPY 命令将数据加载（或导入）到 Amazon Redshift，使用 UNLOAD 命令从 Amazon Redshift 卸载（或导出）数据。您可以使用 CREATE EXTERNAL FUNCTION 命令创建用户定义的函数，这些函数从 AWS Lambda 调用函数。

在使用 Amazon Redshift Spectrum 时，您要使用 CREATE EXTERNAL SCHEMA 命令来指定包含您的数据的 Simple Storage Service（Amazon S3）存储桶的位置。当您运行 COPY、UNLOAD 或 CREATE EXTERNAL SCHEMA 命令时，会提供安全凭证。这些凭证授权您的 Amazon Redshift 集群在目标目的地 [如 Simple Storage Service（Amazon S3）存储桶] 中读取或写入数据。

运行 CREATE EXTERNAL FUNCTION 时，您可以使用 IAM 角色参数提供安全凭证。这些凭证授权您的 Amazon Redshift 集群从 AWS Lambda 调用 Lambda 函数。提供安全凭证的首选方法是指定一个 AWS Identity and Access Management (IAM) 角色。对于 COPY 和 UNLOAD，您可以提供临时凭证。有关创建 IAM 角色的信息，请参阅授权 Amazon Redshift 代表您访问 AWS 服务。

如果用户需要在 AWS 管理控制台之外与 AWS 交互，则需要编程式访问权限。授予编程式访问权限的方法取决于访问 AWS 的用户类型。

要向用户授予编程式访问权限，请选择以下选项之一。

| 哪个用户需要编程式访问权限？ | 目的 | 方式 |
| --- | --- | --- |
| IAM | （推荐）使用控制台凭证作为临时凭证来签署向 AWS | 按照您希望使用的界面的说明进行操作。 |

| 哪个用户需要编程式访问权限？ | 目的 | 方式 |
| --- | --- | --- |
| | CLI、AWS SDK 或 AWS API 发出的编程请求。 | • 对于 AWS CLI，请参阅《AWS Command Line Interface用户指南》中的登录 AWS 以进行本地开发。<br>• 对于 AWS SDK，请参阅《AWS SDK 和工具参考指南》中的登录 AWS 以进行本地开发。 |
| 人力身份<br><br>（在 IAM Identity Center 中管理的用户） | 使用临时凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照您希望使用的界面的说明进行操作。<br><br>• 有关 AWS CLI 的更多信息，请参阅《AWS Command Line Interface 用户指南》中的配置 AWS CLI 以使用 AWS IAM Identity Center。<br>• 有关 AWS SDK、工具和 AWS API 的更多信息，请参阅《AWS SDK 和工具参考指南》中的 IAM Identity Center 身份验证。 |
| IAM | 使用临时凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照《IAM 用户指南》中将临时凭证用于 AWS 资源中的说明进行操作。 |

| 哪个用户需要编程式访问权限？ | 目的 | 方式 |
| --- | --- | --- |
| IAM | （不推荐使用）<br>使用长期凭证签署向 AWS CLI、AWS SDK 或 AWS API 发出的编程请求。 | 按照您希望使用的界面的说明进行操作。<br>• 有关 AWS CLI 的更多信息，请参阅《AWS Command Line Interface 用户指南》中的使用 IAM 用户凭证进行身份验证。<br>• 有关 AWS SDK 和工具的更多信息，请参阅《AWS SDK 和工具参考指南》中的使用长期凭证进行身份验证。<br>• 有关 AWS API 的更多信息，请参阅《IAM 用户指南》中的管理 IAM 用户的访问密钥。 |

使用 IAM 角色的步骤如下所示：

- 创建要与您的 Amazon Redshift 集群结合使用的 IAM 角色。
- 将 IAM 角色与集群关联。
- 在调用 COPY、UNLOAD、CREATE EXTERNAL SCHEMA 或 CREATE EXTERNAL FUNCTION 命令时包含 IAM 角色的 ARN。

### 将 IAM 角色与集群相关联

在您创建一个 IAM 角色以授权 Amazon Redshift 为您访问其他 AWS 服务，必须将该角色与 Amazon Redshift 集群关联。在使用角色加载或卸载数据之前，您必须执行此操作。

### 将 IAM 角色与集群关联所需的权限

要将 IAM 角色与集群关联，用户必须具有对该 IAM 角色的 `iam:PassRole` 权限。此权限允许管理员限制用户可关联到 Amazon Redshift 集群的 IAM 角色。作为最佳实践，我们建议将权限策略附加到

IAM 角色，然后根据需要将其分配给用户和组。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

以下示例显示了一个 IAM policy，该策略可附加到用户以允许该用户执行以下操作：

- 获取用户的账户拥有的所有 Amazon Redshift 集群的详细信息。
- 将三个 IAM 角色中的任一角色与两个 Amazon Redshift 集群中的任一集群关联。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "redshift:DescribeClusters",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                 "redshift:ModifyClusterIamRoles",
                 "redshift:CreateCluster"
            ],
            "Resource": [
                 "arn:aws:redshift:us-east-1:123456789012:cluster:my-redshift-
cluster",
                 "arn:aws:redshift:us-east-1:123456789012:cluster:my-second-
redshift-cluster"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::123456789012:role/MyRedshiftRole",
                "arn:aws:iam::123456789012:role/SecondRedshiftRole",
                "arn:aws:iam::123456789012:role/ThirdRedshiftRole"
             ]
        }
    ]
```

```
}
```

当用户具有相应的权限后，该用户可以将 IAM 角色与 Amazon Redshift 集群关联。随后，IAM 角色可以与 COPY 或 UNLOAD 命令或其他 Amazon Redshift 命令一起使用。

有关 IAM 策略的更多信息，请参阅 IAM 用户指南中的 IAM 策略概览。

管理 IAM 角色与集群的关联

您可以在创建集群时将该 IAM 角色与 Amazon Redshift 集群关联。或者您可以修改现有集群并添加或删除一个或多个 IAM 角色关联。

请注意以下事项：

- 您可以关联的最大 IAM 角色数量受配额限制。
- 一个 IAM 角色可与多个 Amazon Redshift 集群关联。
- 仅当 IAM 角色和集群都归同一 AWS 账户拥有时，该角色才能与 Amazon Redshift 集群关联。

您可通过以下程序，借助控制台管理集群的 IAM 角色关联。

要管理 IAM 角色关联

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Clusters（集群）,然后选择要更新的集群。
3. 对于 Actions（操作），选择 Manage IAM roles（管理 IAM 角色）以显示与集群关联的当前列表 IAM 角色。
4. 在 Manage IAM roles（管理 IAM 角色）页面上，选择要添加的可用 IAM 角色，然后选择 Add IAM role（添加 IAM 角色）。
5. 选择 Done（完成）以保存您的更改。

您可以使用以下方法，通过 AWS CLI 管理集群的 IAM 角色关联。

要在创建集群后将 IAM 角色与其关联，请在 `--iam-role-arns` 命令的 `create-cluster` 参数中指定 IAM 角色的 Amazon Resource Name (ARN)。调用 `create-cluster` 命令时可以添加的最大 IAM 角色数量受配额限制。

将 IAM 角色与 Amazon Redshift 集群关联和解除两者之间的关联是一个异步过程。您可通过调用 describe-clusters 命令获取所有 IAM 角色集群关联的状态。

以下示例将两个 IAM 角色与新创建的名为 my-redshift-cluster 的集群关联。

```
aws redshift create-cluster \
    --cluster-identifier "my-redshift-cluster" \
    --node-type "ra3.4xlarge" \
    --number-of-nodes 16 \
    --iam-role-arns "arn:aws:iam::123456789012:role/RedshiftCopyUnload" \
                    "arn:aws:iam::123456789012:role/SecondRedshiftRole"
```

要将 IAM 角色与现有 Amazon Redshift 集群关联，请在 modify-cluster-iam-roles 命令的 --add-iam-roles 参数中指定 IAM 角色的 Amazon Resource Name (ARN)。调用 modify-cluster-iam-roles 命令时可以添加的最大 IAM 角色数量受配额限制。

以下示例将一个 IAM 角色与名为 my-redshift-cluster 的现有集群关联。

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier "my-redshift-cluster" \
    --add-iam-roles "arn:aws:iam::123456789012:role/RedshiftCopyUnload"
```

要取消 IAM 角色与集群的关联，请在 modify-cluster-iam-roles 命令的 --remove-iam-roles 参数中指定 IAM 角色的 ARN。modify-cluster-iam-roles调用 modify-cluster-iam-roles 命令时可以删除的最大 IAM 角色数量受配额限制。

以下示例从名为 123456789012 的集群中删除 my-redshift-cluster AWS 账户的 IAM 角色的关联。

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier "my-redshift-cluster" \
    --remove-iam-roles "arn:aws:iam::123456789012:role/RedshiftCopyUnload"
```

使用 AWS CLI 列出集群的 IAM 角色关联

要列出与 Amazon Redshift 集群关联的所有 IAM 角色以及 IAM 角色关联的状态，请调用 describe-clusters 命令。与集群关联的每个 IAM 角色的 ARN 将在 IamRoles 列表中返回，如以下示例输出所示。

已与集群关联的角色将显示状态 in-sync。正在与集群关联的角色将显示状态 adding。正与集群解除关联的角色将显示状态 removing。

```
{
    "Clusters": [
        {
            "ClusterIdentifier": "my-redshift-cluster",
            "NodeType": "ra3.4xlarge",
            "NumberOfNodes": 16,
            "IamRoles": [
                {
                    "IamRoleArn": "arn:aws:iam::123456789012:role/MyRedshiftRole",
                    "IamRoleApplyStatus": "in-sync"
                },
                {
                    "IamRoleArn": "arn:aws:iam::123456789012:role/SecondRedshiftRole",
                    "IamRoleApplyStatus": "in-sync"
                }
            ],
            ...
        },
        {
            "ClusterIdentifier": "my-second-redshift-cluster",
            "NodeType": "ra3.4xlarge",
            "NumberOfNodes": 10,
            "IamRoles": [
                {
                    "IamRoleArn": "arn:aws:iam::123456789012:role/MyRedshiftRole",
                    "IamRoleApplyStatus": "in-sync"
                },
                {
                    "IamRoleArn": "arn:aws:iam::123456789012:role/SecondRedshiftRole",
                    "IamRoleApplyStatus": "in-sync"
                },
                {
                    "IamRoleArn": "arn:aws:iam::123456789012:role/ThirdRedshiftRole",
                    "IamRoleApplyStatus": "in-sync"
                }
            ],
            ...
        }
    ]
}
```

有关使用 AWS CLI 的更多信息，请参阅 AWS CLI 用户指南。

## 创建一个 IAM 角色作为 Amazon Redshift 的默认角色

当您通过 Redshift 控制台创建 IAM 角色时，Amazon Redshift 以编程方式在您的 AWS 账户 中创建角色，并自动附上现有的 AWS 托管式策略。这种方法意味着您可以留在 Redshift 控制台上，而不必切换到 IAM 控制台创建角色。要更精细地控制在 Amazon Redshift 控制台上创建的现有 IAM 角色的权限，您可以将自定义托管式策略附加至 IAM 角色。

在控制台中创建的 IAM 角色

当您使用 Amazon Redshift 控制台创建 IAM 角色时，Amazon Redshift 会跟踪所有通过控制台创建的 IAM 角色。Amazon Redshift 预先选择最新的默认 IAM 角色来创建所有新集群，以及从快照中还原集群。

通过 Amazon Redshift 控制台创建一个 IAM 角色，该角色拥有的策略包括可运行 SQL 命令的权限。这些命令包括 COPY、UNLOAD、CREATE EXTERNAL FUNCTION、CREATE EXTERNAL TABLE、CREATE EXTERNAL SCHEMA、CREATE MODEL 或 CREATE LIBRARY。或者，您可以更精细地控制用户通过创建自定义策略并将其附加到 IAM 角色来进行对 AWS 资源的访问。

当您使用控制台创建 IAM 角色并将其设置为集群的默认角色时，您无需提供 IAM 角色的 Amazon Resource Name（ARN）即可执行身份验证和授权。

您通过控制台为集群创建的 IAM 角色具有自动附加的 `AmazonRedshiftAllCommandsFullAccess` 托管式策略。此 IAM 角色允许 Amazon Redshift 为 AWS IAM 账户中的资源复制、卸载、查询和分析数据。托管式策略提供对以下操作的访问权限：COPY、UNLOAD、CREATE EXTERNAL FUNCTION、CREATE EXTERNAL SCHEMA、CREATE MODEL 和 CREATE LIBRARY。此策略还授予权限，以便为相关 AWS 服务运行 SELECT 语句，例如 Amazon S3、Amazon CloudWatch Logs、Amazon SageMaker AI 和 AWS Glue。

CREATE EXTERNAL FUNCTION、CREATE EXTERNAL SCHEMA、CREATE MODEL 和CREATE LIBRARY 命令都有 `default` 关键字。对于这些命令的此关键字，Amazon Redshift 使用在命令运行时与集群关联的默认 IAM 角色。您可以运行 DEFAULT_IAM_ROLE 命令来检查附加至集群的当前默认 IAM 角色。

要控制已有 IAM 角色（设置为 Redshift 集群默认角色）的访问权限，请使用 AUSMEROLE 权限。当数据库用户和组运行前面列出的命令时，此访问控制适用于这些数据库用户和组。向用户或组授予 IAM 角色的 ASSUMEROLE 权限后，该用户或组可以在运行命令时代入该角色。ASSUMEROLE 权限允许您根据需要授予用户相应命令的访问权限。

您可以使用 Amazon Redshift 控制台执行以下操作：

- 默认创建 IAM 角色
- 从集群中删除 IAM 角色
- 将 IAM 角色与集群关联
- 将 IAM 角色设置为默认角色
- 将 IAM 角色设置为不再是集群的默认角色

AmazonRedshiftAllCommandsFullAccess 托管式策略的权限

以下示例显示了 AmazonRedshiftAllCommandsFullAccess 中的权限：托管式策略允许对集群默认的角色（IAM 角色）执行某些操作。附加了权限策略的 IAM 角色可以授权用户或组进行某些操作，以及不允许某些操作。使用这些权限，您可以从 Simple Storage Service（Amazon S3）运行 COPY 命令、运行 UNLOAD，然后使用 CREATE MODEL 命令。

```
{
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetBucketAcl",
                "s3:GetBucketCors",
                "s3:GetEncryptionConfiguration",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListAllMyBuckets",
                "s3:ListMultipartUploadParts",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject",
                "s3:PutBucketAcl",
                "s3:PutBucketCors",
                "s3:DeleteObject",
                "s3:AbortMultipartUpload",
                "s3:CreateBucket"
            ],
            "Resource": [
                "arn:aws:s3:::redshift-downloads",
                "arn:aws:s3:::redshift-downloads/*",
                "arn:aws:s3:::*redshift*",
                "arn:aws:s3:::*redshift*/*"
            ]
```

```
}
```

以下示例显示了 AmazonRedshiftAllCommandsFullAccess 中的权限：托管式策略允许对集群默认的角色（IAM 角色）执行某些操作。附加了权限策略的 IAM 角色可以授权用户或组进行某些操作，以及不允许某些操作。如果有以下权限，您可以运行 CREATE EXTERNAL FUNCTION 命令。

```
{
    "Action": [
        "lambda:InvokeFunction"
    ],
    "Resource": "arn:aws:lambda:*:*:function:*redshift*"
}
```

以下示例显示了 AmazonRedshiftAllCommandsFullAccess 中的权限：托管式策略允许对集群默认的角色（IAM 角色）执行某些操作。附加了权限策略的 IAM 角色可以授权用户或组进行某些操作，以及不允许某些操作。如果具有以下权限，您可以运行 Amazon Redshift Spectrum 所需的 CREATE EXTERNAL SCHEMA 和 CREATE EXTERNAL TABLE 命令。

```
{
            "Effect": "Allow",
            "Action": [
                "glue:CreateDatabase",
                "glue:DeleteDatabase",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:UpdateDatabase",
                "glue:CreateTable",
                "glue:DeleteTable",
                "glue:BatchDeleteTable",
                "glue:UpdateTable",
                "glue:GetTable",
                "glue:GetTables",
                "glue:BatchCreatePartition",
                "glue:CreatePartition",
                "glue:DeletePartition",
                "glue:BatchDeletePartition",
                "glue:UpdatePartition",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:BatchGetPartition"
            ],
            "Resource": [
```

```
                "arn:aws:glue:*:*:table/*redshift*/*",
                "arn:aws:glue:*:*:catalog",
                "arn:aws:glue:*:*:database/*redshift*"
            ]
}
```

以下示例显示了 AmazonRedshiftAllCommandsFullAccess 中的权限：托管式策略允许对集群默认的角色（IAM 角色）执行某些操作。附加了权限策略的 IAM 角色可以授权用户或组进行某些操作，以及不允许某些操作。如果具有以下权限，您可以使用联合查询运行 CREATE EXTERNAL SCHEMA 命令。

```
{
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": [
                "arn:aws:secretsmanager:*:*:secret:*Redshift*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetRandomPassword",
                "secretsmanager:ListSecrets"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "secretsmanager:ResourceTag/Redshift": "true"
                }
            }
},
```

使用控制台管理为集群创建的 IAM 角色

要创建、修改和删除从 Amazon Redshift 控制台创建的 IAM 角色，请使用控制台上的 Clusters（集群）部分。

### 默认创建 IAM 角色

在控制台上，您可以为集群创建一个 IAM 角色，并自动为该角色附上 AmazonRedshiftAllCommandsFullAccess 策略。这个新角色让 Amazon Redshift 能够在您的 IAM 角色中复制、加载、查询和分析 Amazon 资源的数据。

一个集群只能有一个默认的 IAM 角色。如果现有 IAM 角色已设置为默认角色，当您创建另一个 IAM 角色并设置为集群默认角色时，新的 IAM 角色将替换现有的 IAM 角色成为默认角色。

创建新集群和 IAM 角色，并设置 IAM 角色为新集群的默认角色。

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。列出您的账户在当前 AWS 区域 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。
3. 选择创建集群以创建集群。
4. 按照控制台页面上的说明进行操作，为集群配置输入属性。有关这一步骤的更多信息，请参阅创建集群。
5. （可选）选择 Load sample data（加载示例数据），将示例数据集加载到您的 Amazon Redshift 集群，以便开始使用查询编辑器查询数据。

   如果您在防火墙的后面，则数据库端口必须是接受入站连接的开放端口。
6. 按照控制台页面上的说明操作，为 Cluster configuration（集群配置）输入属性。
7. 在 Cluster permissions（集群权限）下，从 Manage IAM roles（管理 IAM 角色）选择 Create IAM role（创建 IAM 角色）。
8. 指定一个 Simple Storage Service（Amazon S3）存储桶，让 IAM 角色通过以下方法访问：

   - 选择 No additional Simple Storage Service（Amazon S3）bucket [没有额外 Simple Storage Service（Amazon S3）存储桶]，在不指定特定 Simple Storage Service（Amazon S3）存储桶的情况下创建 IAM 角色。
   - 选择 Any Simple Storage Service（Amazon S3）bucket [任何 Simple Storage Service（Amazon S3）存储桶]，允许有权限访问您的 Amazon Redshift 集群的用户，也可以访问在您的 AWS 账户 中的任何 Simple Storage Service（Amazon S3）存储桶及其内容。
   - 选择 Specific Simple Storage Service（Amazon S3）buckets [特定 Simple Storage Service（Amazon S3）存储桶]，为创建的 IAM 角色指定一个或多个 Simple Storage Service（Amazon S3）存储桶以供访问。然后从表中选择一个或多个 Simple Storage Service（Amazon S3）存储桶。

9. 选择创建 IAM 角色作为默认角色。Amazon Redshift 会自动创建 IAM 角色并将其设置为集群的默认角色。
10. 选择 Create cluster（创建集群）以创建集群。集群可能需要几分钟才可以使用。

### 从集群中删除 IAM 角色

您可以从集群中移除一个或多个 IAM 角色。

从集群中删除 IAM 角色

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。列出您的账户在当前 AWS 区域 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。
3. 选择要从中删除的 IAM 角色的集群的名称。
4. 在 Cluster permissions（集群权限）下，选择要从集群中删除的一个或多个 IAM 角色。
5. 从 Manage IAM roles（管理 IAM 角色）中选择 Remove IAM roles（删除 IAM 角色）。

### 将 IAM 角色与集群关联

您可以将一个或多个 IAM 角色与集群关联。

将 IAM 角色与您的集群关联

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。列出您的账户在当前 AWS 区域 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。
3. 选择 IAM 角色需要关联的集群。
4. 在 Cluster permissions（集群权限）下，选择要与集群关联的一个或多个 IAM 角色。
5. 从 Manage IAM roles（管理 IAM 角色）中选择 Associate IAM roles（关联 IAM 角色）。
6. 选择一个或多个 IAM 角色并与您的集群关联。
7. 选择 Associate IAM role（关联 IAM 角色）。

### 将 IAM 角色设置为默认角色

您可以将 IAM 角色设置为集群的默认角色。

#### 将 IAM 角色设置为集群的默认角色

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。列出您的账户在当前 AWS 区域 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。
3. 选择要为其设置默认 IAM 角色的集群。
4. 在 Cluster permissions（集群权限）下，从 Associated IAM roles（关联 IAM 角色）中选择要设置为集群默认角色的 IAM 角色。
5. 在 Set default（设置默认值）下选择 Make default（设为默认）。
6. 系统提示时，选择 Set default（设置默认值）以确认设置指定的 IAM 角色为默认角色。

### 将 IAM 角色设置为不再是集群的默认角色

您可以使 IAM 角色不再成为集群的默认角色。

#### 清除 IAM 角色作为集群默认角色的设置

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。列出您的账户在当前 AWS 区域 区域中的集群。列表中的各个列中显示了每个集群的一部分属性。
3. 选择 IAM 角色需要关联的集群。
4. 在 Cluster permissions（集群权限）下，从 Associated IAM roles（关联 IAM 角色）中选择要设置为集群默认角色的 IAM 角色。
5. 在 Set default（设置默认值）下，选择 Clear default（清除默认值）。
6. 系统提示时，选择 Clear default（清除默认值）以确指定的 IAM 角色不再是默认角色。

### 使用 AWS CLI 管理在集群上创建的 IAM 角色

您可以使用 AWS CLI 管理在集群上创建的 IAM 角色

创建默认设置为 IAM 角色的 Amazon Redshift 集群

要创建 Amazon Redshift 集群并将 IAM 角色设为其默认角色，请使用 `aws redshift create-cluster` AWS CLI 命令。

以下 AWS CLI 命令创建一个 Amazon Redshift 集群和名为 myrole1 的 IAM 角色。这个 AWS CLI 命令还将 myrole1 设置为集群的默认值。

```
aws redshift create-cluster \
    --node-type dc2.large \
    --number-of-nodes 2 \
    --master-username adminuser \
    --master-user-password TopSecret1 \
    --cluster-identifier mycluster \
    --iam-roles 'arn:aws:iam::012345678910:role/myrole1'
 'arn:aws:iam::012345678910:role/myrole2' \
    --default-iam-role-arn 'arn:aws:iam::012345678910:role/myrole1'
```

以下代码片段是一个反应示例。

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "MasterUsername": "adminuser",
        "DefaultIamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
        "IamRoles": [
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
                "ApplyStatus": "adding"
            },
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
                "ApplyStatus": "adding"
            }
        ]
        ...
    }
}
```

向 Amazon Redshift 集群添加一个或多个 IAM 角色

要添加与集群关联的一个或多个 IAM 角色，请使用 `aws redshift modify-cluster-iam-roles` AWS CLI 命令。

以下 AWS CLI 命令将 `myrole3` 和`myrole4` 添加到集群。

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier mycluster \
    --add-iam-roles 'arn:aws:iam::012345678910:role/myrole3'
 'arn:aws:iam::012345678910:role/myrole4'
```

以下代码片段是一个反应示例。

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "MasterUsername": "adminuser",
        "DefaultIamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
        "IamRoles": [
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
                "ApplyStatus": "in-sync"
            },
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
                "ApplyStatus": "in-sync"
            },
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole3",
                "ApplyStatus": "adding"
            },
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole4",
                "ApplyStatus": "adding"
            }
        ],
        ...
    }
}
```

从 Amazon Redshift 集群中删除一个或多个 IAM 角色

要删除与集群关联的一个或多个 IAM 角色，请使用 `aws redshift modify-cluster-iam-roles` AWS CLI 命令。

以下 AWS CLI 命令从集群中删除 `myrole3` 和 `myrole4`。

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier mycluster \
    --remove-iam-roles 'arn:aws:iam::012345678910:role/myrole3'
 'arn:aws:iam::012345678910:role/myrole4'
```

以下代码片段是一个反应示例。

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "MasterUsername": "adminuser",
        "DefaultIamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
        "IamRoles": [
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
                "ApplyStatus": "in-sync"
            },
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
                "ApplyStatus": "in-sync"
            },
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole3",
                "ApplyStatus": "removing"
            },
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole4",
                "ApplyStatus": "removing"
            }
        ],
        ...
    }
}
```

将关联的 IAM 角色设置为集群的默认角色

要将关联的 IAM 角色设置为集群的默认角色，请使用 `aws redshift modify-cluster-iam-roles` AWS CLI 命令。

以下 AWS CLI 命令集设置 `myrole2` 为集群默认角色。

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier mycluster \
    --default-iam-role-arn 'arn:aws:iam::012345678910:role/myrole2'
```

以下代码片段是一个反应示例。

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "MasterUsername": "adminuser",
        "DefaultIamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
        "IamRoles": [
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
                "ApplyStatus": "in-sync"
            },
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
                "ApplyStatus": "in-sync"
            }
        ],
        ...
    }
}
```

将未关联的 IAM 角色设置为集群的默认角色

要将未关联的 IAM 角色设置为集群的默认角色，请使用 `aws redshift modify-cluster-iam-roles` AWS CLI 命令。

以下 AWS CLI 命令把 `myrole2` 添加到 Amazon Redshift 集群，然后将其设置为集群的默认角色。

```
aws redshift modify-cluster-iam-roles \
```

```
    --cluster-identifier mycluster \
    --add-iam-roles 'arn:aws:iam::012345678910:role/myrole3' \
    --default-iam-role-arn 'arn:aws:iam::012345678910:role/myrole3'
```

以下代码片段是一个反应示例。

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster",
        "NodeType": "dc2.large",
        "MasterUsername": "adminuser",
        "DefaultIamRoleArn": "arn:aws:iam::012345678910:role/myrole3",
        "IamRoles": [
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
                "ApplyStatus": "in-sync"
            },
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
                "ApplyStatus": "in-sync"
            },
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole3",
                "ApplyStatus": "adding"
            }
        ],
        ...
    }
}
```

从快照还原集群，并将 IAM 角色设置为其默认角色

从快照还原集群时，您可以关联现有 IAM 角色，也可以创建一个新角色并将其设置为集群的默认角色。

要从快照还原 Amazon Redshift 集群，并将 IAM 角色设置为集群默认角色，请使用 `aws redshift restore-from-cluster-snapshot` AWS CLI 命令。

以下 AWS CLI 命令从快照还原集群，并将 `myrole2` 设置为集群默认角色。

```
aws redshift restore-from-cluster-snapshot \
    --cluster-identifier mycluster-clone \
```

```
    --snapshot-identifier my-snapshot-id
    --iam-roles 'arn:aws:iam::012345678910:role/myrole1'
 'arn:aws:iam::012345678910:role/myrole2' \
    --default-iam-role-arn 'arn:aws:iam::012345678910:role/myrole1'
```

以下代码片段是一个反应示例。

```
{
    "Cluster": {
        "ClusterIdentifier": "mycluster-clone",
        "NodeType": "dc2.large",
        "MasterUsername": "adminuser",
        "DefaultIamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
        "IamRoles": [
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole1",
                "ApplyStatus": "adding"
            },
            {
                "IamRoleArn": "arn:aws:iam::012345678910:role/myrole2",
                "ApplyStatus": "adding"
            }
        ],
        ...
    }
}
```

## 使用联合身份管理 Amazon Redshift 对本地资源和 Amazon Redshift Spectrum 外部表的访问权限

将 AWS 中的身份联合验证和 GetDatabaseCredentials 提供的凭证结合使用，可以简化对本地数据和外部数据的授权和访问。在本教程中，我们将说明如何使用 AWS 身份联合验证（而不是使用特定 IAM 角色）提供对资源的访问权限。

目前，要向用户提供对驻留在 Amazon S3 中的外部数据的访问权限，您需要创建一个具有权限策略中定义的权限的 IAM 角色。之后，附加了此角色的用户便能访问外部数据。虽然这行得通，但如果您想提供粒度规则（例如，使特定列对特定用户不可用），则可能需要在外部架构上进行其他配置。

利用由 GetDatabaseCredentials 提供的凭证，身份联合验证可以使用更易于指定和更改的粒度 IAM 规则提供对 AWS Glue 和 Redshift Spectrum 资源的访问权限。这更易于应用符合您的业务规则的访问权限。

使用联合凭证的好处如下：

- 您不必为 Redshift Spectrum 管理集群附加的 IAM 角色。
- 集群管理员可以创建一个可供具有不同 IAM 上下文的使用者访问的外部架构。例如，这对于对表执行列筛选会很有用，在此情况下，其他使用者将查询同一外部架构并在返回的记录中获得不同的字段。
- 您可以使用具有 IAM 权限的用户查询 Amazon Redshift，而不是仅使用角色。

### 准备身份以使用联合身份进行登录

在使用联合身份登录之前，您必须执行多个预备步骤。这些说明假定您目前有一个 Redshift Spectrum 外部架构，该架构引用了存储在 Amazon S3 存储桶中的数据文件，并且该存储桶与您的 Amazon Redshift 集群或 Amazon Redshift Serverless 数据仓库位于同一账户中。

1. 创建一个 IAM 身份。这可以是用户或 IAM 角色。使用 IAM 支持的任何名称。
2. 将权限策略附加到此身份。指定以下任一项：

   - `redshift:GetClusterCredentialsWithIAM`（适用于 Amazon Redshift 预调配集群）
   - `redshift-serverless:GetCredentials`（适用于 Amazon Redshift Serverless）

   您可以使用 IAM 控制台通过策略编辑器添加权限。

   IAM 身份还需要对外部数据的访问权限。通过直接添加以下 AWS 托管式策略，授予对 Amazon S3 的访问权限：

   - `AmazonS3ReadOnlyAccess`
   - `AWSGlueConsoleFullAccess`

   如果您使用 AWS Glue 准备外部数据，则需要最后一个托管式策略。有关授予对 Amazon Redshift Spectrum 的访问权限的步骤的更多信息，请参阅为 Amazon Redshift 创建 IAM 角色，它是 Amazon Redshift 和 Redshift Spectrum 入门指南的一部分。此部分介绍了添加 IAM 策略以访问 Redshift Spectrum 的步骤。
3. 设置 SQL 客户端以连接到 Amazon Redshift。使用 Amazon Redshift JDBC 驱动程序，并将用户的凭证添加到此工具的凭证属性中。像 SQL Workbench/J 这样的客户端非常适合执行此操作。设置以下客户端连接扩展属性：

   - AccessKeyID – 您的访问密钥标识符。

- SecretAccessKey – 您的秘密访问密钥。（请注意，如果您不使用加密，则传输私有密钥会带来安全风险。）
- SessionToken – IAM 角色的一组临时凭证。
- groupFederation – 如果要为预调配集群配置联合身份，则设置为 `true`。如果您使用的是 Amazon Redshift Serverless，请不要设置此参数。
- LogLevel – 整数日志级别值。该项为可选项。

4. 将 URL 设置为在 Amazon Redshift 或 Amazon Redshift Serverless 控制台中找到的 JDBC 端点。将 URL 架构替换为 jdbc:redshift:iam: 并使用此格式：

   - 适用于 Amazon Redshift 预调配集群的格式：`jdbc:redshift:iam://<cluster_id>.<unique_suffix>.<region>.redshift.amazonaws.com:<port>/<database_name>`

     示例：`jdbc:redshift:iam://test1.12345abcdefg.us-east-1.redshift.amazonaws.com:5439/dev`
   - 适用于 Amazon Redshift Serverless 的格式：`jdbc:redshift:iam://<workgroup-name>.<account-number>.<aws-region>.redshift-serverless.amazonaws.com:5439:<port>/<database_name>`

     示例：`jdbc:redshift:iam://default.123456789012.us-east-1.redshift-serverless.amazonaws.com:5439/dev`

   在您首次使用 IAM 身份连接到数据库后，Amazon Redshift 会自动创建一个同名 Amazon Redshift 身份，其前缀为 `IAM:`（对于用户）或 `IAMR:`（对于 IAM 角色）。本主题中的剩余步骤展示了适用于用户的示例。

   如果未自动创建 Redshift 用户，您可以通过以下方式创建一个此类用户：运行 `CREATE USER` 语句、使用管理员账户并以格式 `IAM:<user name>` 指定用户名。

5. 作为 Amazon Redshift 集群管理员，向 Redshift 用户授予访问外部架构所需的权限。

```
GRANT ALL ON SCHEMA my_schema to "IAM:my_user";
```

要使 Redshift 用户能够在外部架构中创建表，他们必须是架构所有者。例如：

```
ALTER SCHEMA my_schema owner to "IAM:my_user";
```

6. 要验证配置，请在授予权限后，使用 SQL 客户端以用户身份运行查询。此查询示例从外部表中检索数据。

```
SELECT * FROM my_schema.my_table;
```

开始将身份和授权传播到 Redshift Spectrum

要传递联合身份以查询外部表，请将 SESSION 设置为 CREATE EXTERNAL SCHEMA 的 IAM_ROLE 查询参数的值。以下步骤说明如何设置和利用 SESSION 授予对外部架构的查询权限。

1. 创建本地表和外部表。使用 AWS Glue 编写目录的外部表适合此操作。
2. 使用 IAM 身份连接到 Amazon Redshift。如上一节中所述，在身份连接到 Amazon Redshift 时，将创建一个 Redshift 数据库用户。如果用户以前不存在，则将创建该用户。如果用户是新用户，则管理员必须向其授予在 Amazon Redshift 中执行任务（例如查询和创建表）的权限。
3. 使用管理员账户连接到 Redshift。运行该命令以使用 SESSION 值创建外部架构。

```
create external schema spectrum_schema from data catalog
database '<my_external_database>'
region '<my_region>'
iam_role 'SESSION'
catalog_id '<my_catalog_id>';
```

请注意，在此示例中，将设置 catalog_id。这是随功能一起添加的新设置，因为 SESSION 替换了特定角色。

在此示例中，查询中的值将模仿实际值的显示方式。

```
create external schema spectrum_schema from data catalog
database 'spectrum_db'
region 'us-east-1'
iam_role 'SESSION'
catalog_id '123456789012'
```

在此情况下，catalog_id 值是您的 AWS 账户 ID。

4. 使用您在步骤 2 中连接的 IAM 身份运行查询以访问外部数据。例如：

```
select * from spectrum_schema.table1;
```

例如，在此情况下，table1 可以是 Amazon S3 存储桶中某个文件中的采用 JSON 格式的数据。

5. 如果您已有一个使用集群附加的 IAM 角色的外部架构并指向您的外部数据库或架构，则可以替换现有架构并使用联合身份（如这些步骤中所述），也可以创建一个新的架构。

SESSION 表示使用联合身份凭证用于查询外部架构。在使用 SESSION 查询参数时，请务必设置 catalog_id。由于它指向用于架构的数据目录，因此它是必需的。之前，catalog_id 是从分配给 iam_role 的值中检索的。当您通过此方式设置身份和授权传播时，例如，通过使用联合凭证查询外部架构来传播到 Redshift Spectrum 时，不需要通过 IAM 角色进行授权。

使用说明

常见的连接错误如下：检索临时凭证时出现 IAM 错误：无法使用提供的解组器对异常响应进行解组。此错误是由于使用旧版 JDBC 驱动程序造成的。联合身份所需的最低驱动程序版本为 2.1.0.9。您可以从下载 Amazon Redshift JDBC 驱动程序版本 2.x 获取 JDBC 驱动程序。

其他资源

这些链接为管理外部数据的访问提供了额外信息。

- 您仍可以使用 IAM 角色访问 Redshift Spectrum 数据。有关更多信息，请参阅 授权 Amazon Redshift 代表您访问 AWS 服务。
- 当您使用 AWS Lake Formation 管理对外部表的访问权限时，您可以使用具有联合 IAM 身份的 Redshift Spectrum 对其进行查询。您不再需要管理集群附加的 IAM 角色，Redshift Spectrum 可以查询向 AWS Lake Formation 注册的数据。有关更多信息，请参阅将 AWS Lake Formation 与 Amazon Redshift Spectrum 结合使用。

# 使用 AWS Secrets Manager 管理 Amazon Redshift 管理员密码

Amazon Redshift 可以与 AWS Secrets Manager 集成，以便使用加密密钥来生成和管理您的管理员凭证。使用 AWS Secrets Manager，您可以将管理员密码替换为 API 调用，以便在需要时以编程方式检索密钥。使用密钥而不是硬编码凭证可以降低这些凭证被公开或泄漏的风险。有关 AWS Secrets Manager 的更多信息，请参阅《AWS Secrets Manager 用户指南》。https://docs.aws.amazon.com/secretsmanager/latest/userguide/intro.html

在执行以下操作之一时，您可以使用 AWS Secrets Manager 来指定由 Amazon Redshift 管理您的管理员密码：

- 创建预置集群或无服务器命名空间

- 编辑、更新或修改预置集群或无服务器命名空间的管理员凭证
- 从快照还原集群或无服务器命名空间

当您在 AWS Secrets Manager 中指定由 Amazon Redshift 来管理管理员密码时，Amazon Redshift 会生成密码并将其存储在 Secrets Manager 中。您可以直接在 AWS Secrets Manager 中访问密钥以检索管理员用户的凭证。（可选）如果您需要从其他 AWS 账户访问密钥，您可以指定客户管理的密钥来加密密钥。您也可以使用 AWS Secrets Manager 提供的 KMS 密钥。

Amazon Redshift 管理密钥的设置，默认情况下每 30 天轮换一次密钥。您可以随时手动轮换密钥。如果您删除管理 AWS Secrets Manager 中密钥的预置集群或无服务器命名空间，则该密钥及其关联的元数据也会被删除。

要使用密钥管理的凭证连接到集群或无服务器命名空间，您可以使用 Secrets Manager 控制台或 `GetSecretValue` Secrets Manager API 调用，从 AWS Secrets Manager 检索密钥。有关更多信息，请参阅《AWS Secrets Manager 用户指南》中的从 AWS Secrets Manager 中检索密钥和使用 AWS Secrets Manager 密钥中的凭证连接到 SQL 数据库。

## AWS Secrets Manager 集成所需的权限

用户必须拥有所需的权限才能执行与 AWS Secrets Manager 集成相关的操作。创建 IAM 策略，以便授予权限对所需的指定资源执行特定的 API 操作。然后，将这些策略附加到需要这些权限的 IAM 权限集或角色。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

指定 Amazon Redshift 管理 AWS Secrets Manager 中管理员密码的用户必须具有执行以下操作的权限：

- `secretsmanager:CreateSecret`
- `secretsmanager:RotateSecret`
- `secretsmanager:DescribeSecret`
- `secretsmanager:UpdateSecret`
- `secretsmanager:DeleteSecret`
- `secretsmanager:GetRandomPassword`
- `secretsmanager:TagResource`

如果用户想要为预置集群在 `MasterPasswordSecretKmsKeyId` 参数中传递 KMS 密钥，或者想要为无服务器命名空间在 `AdminPasswordSecretKmsKeyId` 参数中传递 KMS 密钥，则在上面列出的权限之外，他们还需要以下权限。

- kms:Decrypt
- kms:GenerateDataKey
- kms:CreateGrant
- kms:RetireGrant

## 管理员密码密钥轮换

默认情况下，Amazon Redshift 每 30 天自动轮换一次密钥，以确保您的凭证不会长时间保持不变。当 Amazon Redshift 轮换管理员密码密钥时，AWS Secrets Manager 会更新现有密钥以包含新的管理员密码。Amazon Redshift 更改集群的管理员密码，使其与更新后的密钥中的密码相匹配。

您可以立即轮换密钥，而不必等待使用 AWS Secrets Manager 的计划轮换。有关轮换密钥的更多信息，请参阅《AWS Secrets Manager 用户指南》中的轮换 AWS Secrets Manager 密钥。

## 将 AWS Secrets Manager 与 Amazon Redshift 配合使用时的注意事项

使用 AWS Secrets Manager 管理您的预置集群或无服务器命名空间的管理员凭证时，请注意以下几点：

- 当您暂停某个集群而集群的管理员凭证由 AWS Secrets Manager 管理时，您集群的密钥不会被删除，并且系统将继续向您收取该密钥的费用。只有在您删除集群时才会删除密钥。
- 如果您的集群在 Amazon Redshift 尝试轮换其上附加的密钥时处于暂停状态，则轮换将失败。在这种情况下，Amazon Redshift 会停止自动轮换，即使在您恢复集群后也不会尝试再次轮换。您必须使用 secretsmanager:RotateSecret API 调用重新启动自动轮换计划，才能让 AWS Secrets Manager 继续自动轮换您的密钥。
- 如果在 Amazon Redshift 尝试轮换其上附加的密钥时，您的无服务器命名空间没有关联工作组，则轮换将失败并且不会尝试再次轮换，即使您关联了工作组也是如此。您必须使用 secretsmanager:RotateSecret API 调用重新启动自动轮换计划，才能让 AWS Secrets Manager 继续自动轮换您的密钥。

## 在 Amazon Redshift 中检索密钥的 Amazon 资源名称（ARN）

您可以使用 Amazon Redshift 控制台，查看由 AWS Secrets Manager 管理的所有密钥的 Amazon 资源名称（ARN）。在拥有密钥的 ARN 时，您可以使用 AWS Secrets Manager 查看有关密钥的详细信息以及密钥中的加密数据。有关使用 ARN 检索密钥更多信息，请参阅《AWS Secrets Manager 用户指南》中的检索密钥。

查看有关 Amazon Redshift 预置集群的密钥的详细信息

使用 Amazon Redshift 控制台查看集群的密钥的 Amazon 资源名称（ARN），步骤如下：

1. 登录 AWS 管理控制台 并打开 Amazon Redshift 控制台。
2. 在集群概览窗格中，选择要查看其密钥的集群。
3. 选择属性选项卡。
4. 查看管理员凭证 ARN 下密钥的 ARN。此 ARN 是密钥的标识符，您可以在 AWS Secrets Manager 中用它来查看密钥的详细信息。

查看有关 Amazon Redshift Serverless 命名空间的密钥的详细信息

使用 Amazon Redshift 控制台查看无服务器命名空间的密钥的 Amazon 资源名称（ARN），步骤如下：

1. 登录 AWS 管理控制台 并打开 Amazon Redshift 控制台。
2. 在预置集群控制面板中，选择页面右上角的转到无服务器。
3. 在无服务器控制面板中，滚动到命名空间/工作组窗格，然后选择要查看其密钥的命名空间。
4. 在一般信息窗格中，在管理员凭证 ARN 下查看密钥的 ARN。此 ARN 是密钥的标识符，您可以在 AWS Secrets Manager 中用它来查看密钥的详细信息。

## 为数据库连接凭证创建密钥

您可以创建 Secrets Manager 密钥来存储用于连接到 Amazon Redshift 预置集群或 Redshift Serverless 命名空间和工作组的凭证。在 Amazon Redshift 查询编辑器 v2 中计划查询时，也可以使用该密钥。

使用 Secrets Manager 控制台为 Amazon Redshift 预调配集群中的数据库创建密钥

1. 打开 Secrets Manager 控制台（https://console.aws.amazon.com/secretsmanager/）。
2. 导航到密钥列表，然后选择存储新密钥。
3. 选择 Amazon Redshift 数据仓库的凭证。在创建密钥的步骤中输入您的信息，如下所示：

   - 在用户名的凭证中，输入数据仓库管理用户的名称。
   - 在密码的凭证中，输入用户名的密码。
   - 对于加密密钥，请选择您的加密密钥。
   - 对于数据仓库，请选择包含您的数据的 Amazon Redshift 预调配集群。
   - 对于密钥名称，输入密钥的名称。

- 对于描述，请输入密钥的描述。
- 对于标签，输入带有字词 **Redshift** 的标签键。当您尝试使用 Amazon Redshift 查询编辑器 v2 连接到数据仓库时，需要使用此标签键来列出密钥。该密钥的标签键必须以字符串 **Redshift** 开头，才能在管理控制台上的 AWS Secrets Manager 下列出该密钥。

4. 继续通过多个步骤输入有关密钥的信息，直到您在审核步骤中存储您所做的更改。

   您的凭证、引擎、主机、端口和集群标识符的特定值存储在密钥中。此外，该密钥还使用标签键 `Redshift` 进行标记。

使用 Redshift Serverless 控制台为 Redshift Serverless 命名空间中的数据库创建密钥

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 选择 Redshift Serverless 并导航到命名空间配置。
3. 选择要为其创建密钥凭证的命名空间。
4. 打开操作、编辑管理员凭证。
5. 对于管理员密码，请选择在 AWS Secrets Manager 中管理管理员凭证。
6. 选择保存更改以保存您的更改。

确认出现一条消息，说明已成功更改密码。您也可以在 Secrets Manager 控制台中查看密钥。您可以使用该密钥，在 Redshift Serverless 控制台和 Amazon Redshift 查询编辑器 v2 中通过 AWS Secrets Manager 连接方法连接到工作组中的数据库。该密钥的标签键必须以字符串“Redshift”开头，才能在查询编辑器 v2 Web 应用程序中列出该密钥。该密钥的标签键必须以字符串 **Redshift** 开头，才能在管理控制台上的 AWS Secrets Manager 下列出该密钥。

使用 Secrets Manager 控制台为 Redshift Serverless 命名空间中的数据库创建密钥

1. 打开 Secrets Manager 控制台（https://console.aws.amazon.com/secretsmanager/）。
2. 导航到密钥列表，然后选择存储新密钥。
3. 选择 Amazon Redshift 数据仓库的凭证。在创建密钥的步骤中输入您的信息，如下所示：

   - 在用户名的凭证中，输入数据仓库管理用户的名称。
   - 在密码的凭证中，输入用户名的密码。
   - 对于加密密钥，请选择您的加密密钥。
   - 对于数据仓库，请选择包含您的数据的 Redshift Serverless 命名空间。

- 对于密钥名称，输入密钥的名称。
- 对于描述，请输入密钥的描述。
- 对于标签，输入带有字词 **Redshift** 的标签键。当您尝试使用 Amazon Redshift 查询编辑器 v2 连接到数据仓库时，需要使用此标签键来列出密钥。该密钥的标签键必须以字符串 **Redshift** 开头，才能在管理控制台上的 AWS Secrets Manager 下列出该密钥。

4. 继续通过多个步骤输入有关密钥的信息，直到您在审核步骤中存储您所做的更改。

   您的凭证、数据库名称、主机、端口、命名空间和引擎的特定值存储在密钥中。此外，该密钥还使用标签键 Redshift 进行标记。

使用 AWS CLI 为 Redshift Serverless 命名空间中的数据库创建密钥

您可以使用 AWS CLI 创建密钥。一种方法是使用 AWS CloudShell 来运行 Secrets Manager AWS CLI 命令，如下所示。您必须具有适当的权限才能运行 AWS CLI 命令，如以下过程所示。

1. 在 AWS 控制台上，打开 AWS CloudShell 命令提示符。有关 AWS CloudShell 的更多信息，请参阅《AWS CloudShell User Guide》中的What is AWS CloudShell。
2. 例如，对于密钥 MyTestSecret，输入一条 Secrets Manager 命令，来存储用于连接数据库或计划 Amazon Redshift 查询编辑器 v2 查询的密钥。将命令中的以下值替换为环境的值：

   - *admin* 是数据仓库的管理员用户名。
   - *passw0rd* 是管理员的密码。
   - *dev* 是数据仓库中的初始数据库名称。
   - *region* 是包含数据仓库的 AWS 区域。例如 us-east-1。
   - *123456789012* 为 AWS 账户。
   - *namespace-id* 是类似于 c3928f0e-c889-4d2b-97a5-5738324d5d3e 的命名空间标识符。您可以在 Amazon Redshift 控制台中的无服务器命名空间详细信息页面上找到该标识符。

```
aws secretsmanager create-secret \
--name MyTestSecret \
--description "My test secret created with the CLI." \
--secret-string "{\"username\":\"admin\",\"password\":\"passw0rd\",\"dbname\":
\"dev\",\"engine\":\"redshift\"}" \
--tags "[{\"Key\":\"redshift-serverless:namespaceArn\",\"Value\":
\"arn:aws:redshift-serverless:region:123456789012:namespace/namespace-id\"}]"
```

# Amazon Redshift 中的日志记录和监控

监控是保持 Amazon Redshift 和您的 AWS 解决方案的可靠性、可用性和性能的重要方面。您可以从 AWS 解决方案的各个部分收集监控数据，以便您可以更轻松地调试多点故障（如果发生）。AWS 提供了多种工具来监控您的 Amazon Redshift 资源并对潜在事件做出响应：

Amazon CloudWatch 警报

使用 Amazon CloudWatch 警报，您可以在指定时间段内监控某个指标。如果指标超过给定阈值，则会向 Amazon SNS 主题或 AWS Auto Scaling 策略发送通知。CloudWatch 警报将不会调用操作，因为这些操作处于特定状态。而是必须在状态已改变并在指定的若干个时间段内保持不变后才调用。有关更多信息，请参阅 创建警报。有关 指标的列表，请参阅 Amazon Redshift 中的性能数据。

AWS CloudTrail 日志

CloudTrail 会提供用户、IAM 角色或 AWS 服务在 Amazon Redshift 中所执行的 API 操作的记录。使用 CloudTrail 收集的信息，您可以确定向 Amazon Redshift 发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其他详细信息。有关更多信息，请参阅 使用 Cloudtrail 进行日志记录。

## 数据库审计日志记录

Amazon Redshift 记录您的数据库中的连接和用户活动相关信息。这些日志有助于您监控数据库以确保安全并进行故障排除，该流程称为数据库审计。可以存储日志的位置包括：

- Amazon S3 存储桶 - 这为负责监控数据库中活动的用户提供了访问权限以及数据安全功能。
- Amazon CloudWatch - 您可以使用 CloudWatch 中内置的功能（例如可视化功能和设置操作）查看审计日志记录数据。

> **Note**
>
> SYS_CONNECTION_LOG 收集 Amazon Redshift Serverless 的连接日志数据。请注意，当您收集 Amazon Redshift Serverless 的审计日志记录数据时，无法将其发送到日志文件，只能发送到 CloudWatch。

主题

- Amazon Redshift 日志
- 审计日志和 Amazon CloudWatch
- 启用审计日志记录
- 安全日志记录

## Amazon Redshift 日志

Amazon Redshift 在以下日志文件中记录信息：

- 连接日志 – 记录身份验证尝试以及连接与断开连接。
- 用户日志 – 记录与数据库用户定义更改相关的信息。
- 用户活动日志 – 记录在数据库中运行之前的每个查询。

连接日志和用户日志主要用于实现安全性。您可以使用连接日志来监控连接到数据库的用户信息以及相关的连接信息。这些信息可能是他们的 IP 地址、发出请求的时间以及使用的身份验证类型等。您可以使用用户日志来监控数据库用户定义更改。

用户活动日志主要用于进行故障排除。它会跟踪用户及系统在数据库中执行的查询类型的相关信息。

连接日志和用户日志均对应于数据库的系统表中存储的信息。您可以使用系统表获取相同的信息，但日志文件可提供更简单的检索和查看机制。日志文件依赖 Simple Storage Service（Amazon S3）权限（而非数据库权限）针对表执行查询。此外，通过查看日志文件中的信息（而非查询系统表），您可以减少与数据库互动产生的任何影响。

> **Note**
>
> 日志文件没有系统日志表（即 STL_USERLOG 和 STL_CONNECTION_LOG）那么新。早于 (但不包括) 最新记录的记录将被复制到日志文件。

> **Note**
>
> 对于 Amazon Redshift Serverless，SYS_CONNECTION_LOG 收集连接日志数据。当您收集 Amazon Redshift Serverless 的审计日志记录数据时，无法将其发送到日志文件，只能发送到 CloudWatch。

连接日志

记录身份验证尝试以及连接与断开连接。下表介绍了连接日志中的信息。有关这些字段的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 STL_CONNECTION_LOG。有关为 Amazon Redshift Serverless 收集的连接日志数据的更多信息，请参阅 SYS_CONCON_LOG。

| 列名称 | 描述 |
| --- | --- |
| event | 连接或身份验证事件。 |
| recordtime | 事件发生的时间。 |
| remotehost | 远程主机的名称或 IP 地址。 |
| remoteport | 远程主机的端口号。 |
| pid | 与语句关联的进程 ID。 |
| dbname | 数据库名称。 |
| username | 用户名。 |
| authmethod | 身份验证方法。 |
| duration | 连接的持续时间（单位为微秒）。 |
| sslversion | 安全套接字层 (SSL) 版本。 |
| sslcipher | SSL 密码。 |
| mtu | 最大传输单元 (MTU)。 |
| sslcompression | SSL 压缩类型。 |
| sslexpansion | SSL 扩展类型。 |
| iamauthguid | AWS CloudTrail 请求的 AWS Identity and Access Management（IAM）身份验证 ID。这是 GetClusterCredentials API 调用创建用于给定连接的凭证时使用的标识符。 |

| 列名称 | 描述 |
| --- | --- |
| application_name | 会话应用程序的初始名称或更新名称。 |
| os_version | 连接到 Amazon Redshift 集群的客户端计算机上的操作系统版本。 |
| driver_version | 从第三方 SQL 客户端工具连接到 Amazon Redshift 集群的 ODBC 或 JDBC 驱动程序版本。 |
| plugin_name | 用于连接到您的 Amazon Redshift 集群的插件名称。 |
| protocol_version | Amazon Redshift 驱动程序在与服务器建立连接时使用的内部协议版本。 |
| sessionid | 当前会话的全局唯一标识符。 |
| 压缩 | 连接正在使用的压缩算法。 |

用户日志

记录数据库用户的以下更改的详细信息。

- 创建用户
- 删除用户
- 更改用户（重命名）
- 更改用户（更改属性）

| 列名称 | 描述 |
| --- | --- |
| userid | 受更改影响的用户的 ID。 |
| username | 受更改影响的用户的用户名称。 |
| oldusername | 对于重命名操作，这指的是原始用户名称。对于任何其他操作，此字段为空。 |
| 操作 | 发生的操作。有效值： |

| 列名称 | 描述 |
| --- | --- |
| | • 更改<br>• 创建<br>• Drop<br>• 重命名 |
| usecreatedb | 如果为 true (1)，则表示用户具有创建数据库的权限。 |
| usesuper | 如果为 true (1)，则表示用户为超级用户。 |
| usecatupd | 如果为 true (1)，则表示用户可更新系统目录。 |
| valuntil | 密码到期日期。 |
| pid | 进程 ID。 |
| xid | 事务 ID。 |
| recordtime | 查询开始的时间（采用 UTC 表示）。 |

查询 SYS_USERLOG 系统视图，以查找有关用户更改的更多信息。此视图包括来自 Amazon Redshift Serverless 的日志数据。

用户活动日志

记录在数据库中运行之前的每个查询。

| 列名称 | 描述 |
| --- | --- |
| recordtime | 事件发生的时间。 |
| db | 数据库名称。 |
| user | 用户名。 |
| pid | 与语句关联的进程 ID。 |
| userid | 用户 ID。 |

| 列名称 | 描述 |
| --- | --- |
| xid | 事务 ID。 |
| query | 日志前缀：后跟查询的文字，包括换行符。 |

## 审计日志和 Amazon CloudWatch

在 Amazon Redshift 中，原定设置情况下审计日志记录处于未启用状态。当您针对集群打开日志记录时，Amazon Redshift 将日志导出到 Amazon CloudWatch 中，或创建日志并将其上载到 Simple Storage Service (Amazon S3) 中，以捕获从启用审计日志记录的时间到当前时间的数据。每个日志记录更新都是以前日志的延续。

在 CloudWatch 或 Simple Storage Service (Amazon S3) 中存储审计日志记录是一个可选流程。记录到数据表并非可选流程，而会自动发生。有关系统表日志记录的更多信息，请参阅 Amazon Redshift 数据库开发人员指南中的系统表参考。

您可以通过使用 AWS 管理控制台、Amazon Redshift API 参考或 AWS Command Line Interface (AWS CLI) 一起启用连接日志、用户日志以及用户活动日志。对于用户活动日志，您还必须启用 `enable_user_activity_logging` 数据库参数。如果您仅启用审计日志记录功能，但不启用相关参数，则数据库审计日志将仅为连接日志和用户日志记录信息，而不为用户活动日志记录信息。`enable_user_activity_logging` 参数预设情况下未启用 (`false`)。您可以将它设置为 `true` 以启用用户活动日志。有关更多信息，请参阅 Amazon Redshift 参数组。

在对 CloudWatch 启用日志记录后，Amazon Redshift 将集群连接、用户和用户活动日志数据导出到 Amazon CloudWatch Logs 日志组。就 架构 而言，日志数据不会改变。CloudWatch 专为监控应用程序而构建，您可以使用它来执行实时分析或将其设置为采取措施。还可以使用 Amazon CloudWatch Logs 在持久性存储中存储日志记录。

使用 CloudWatch 查看日志是在 Simple Storage Service (Amazon S3 )中存储日志文件的推荐替代方法。它不需要太多的配置，而且可能符合您的监控要求，尤其是如果您已经使用它来监控其他服务和应用程序。

### Amazon CloudWatch 中的日志组和日志事件

在选择要导出的 Amazon Redshift 日志后，您可以在 Amazon CloudWatch Logs 中监控日志事件。将使用以下前缀为 Amazon Redshift 无服务器自动创建新的日志组，其中 `log_type` 表示日志类型。

```
/aws/redshift/cluster/<cluster_name>/<log_type>
```

例如，如果您选择导出连接日志，则日志数据将存储在以下日志组中。

```
/aws/redshift/cluster/cluster1/connectionlog
```

使用日志流将日志事件导出到日志组。要在无服务器端点的日志事件中搜索信息，请使用 Amazon CloudWatch Logs 控制台、AWS CLI 或 Amazon CloudWatch Logs API。有关搜索和筛选日志数据的信息，请参阅使用筛选条件从日志事件创建指标。

在 CloudWatch 中，您可以使用所提供的旨在实现粒度和灵活性的查询语法搜索日志数据。有关更多信息，请参阅 CloudWatch Logs Insights 查询语法。

迁移到 Amazon CloudWatch 审计日志记录

在任何情况下，如果您要向 Simple Storage Service (Amazon S3) 发送日志并更改配置（例如向 CloudWatch 发送日志），保留在 Simple Storage Service (Amazon S3) 中的日志都不会受到影响。您仍可以在日志数据所在的 Simple Storage Service (Amazon S3) 存储桶中查询日志数据。

Amazon S3 中的日志文件

Simple Storage Service（Amazon S3）中的 Amazon Redshift 日志文件的数量和大小在很大程度上取决于您集群中的活动。如果您有一个活动的集群生成了大量日志，则 Amazon Redshift 可能会更频繁地生成日志文件。对于同一类活动，您可能有一系列日志文件，例如同一个小时有多个连接日志。

当 Amazon Redshift 使用 Simple Storage Service (Amazon S3) 存储日志时，您需要为在 Simple Storage Service (Amazon S3) 中使用的存储支付相应费用。在对 Simple Storage Service (Amazon S3) 配置日志记录之前，应计划需存储日志文件的时长。在此过程中，请确定何时可根据审计需求删除或归档日志文件。您制定的计划在很大程度上取决于您存储的数据类型，例如需满足合规性或法规要求的数据。有关 Simple Storage Service（Amazon S3）定价的更多信息，请转至 Amazon Simple Storage Service（S3）定价。

对 Amazon S3 启用日志记录时的限制

审计日志记录具有以下约束：

- 您只能使用 Amazon S3 托管式密钥（SSE-S3）加密（AES-256）。
- Amazon S3 桶必须关闭 S3 对象锁定功能。

### Amazon Redshift 审计日志记录的存储桶权限

当您对 Simple Storage Service (Amazon S3) 启用日志记录时，Amazon Redshift 会收集日志记录信息并将其上载到 Simple Storage Service (Amazon S3) 中存储的日志文件。您可以使用现有存储桶或新存储桶。Amazon Redshift 需要对存储桶具备以下 IAM 权限：

- `s3:GetBucketAcl` 该服务要求对 Simple Storage Service（Amazon S3）存储桶具备读取权限，以便可以识别存储桶拥有者。
- `s3:PutObject` 该服务要求具备放置对象权限，以便上载日志。此外，开启日志记录的用户或 IAM 角色必须具有对 Amazon S3 桶的 `s3:PutObject` 权限。每次上传日志时，该服务就会确定当前存储桶拥有者与启用日志记录时的存储桶拥有者是否一致。如果这些拥有者不匹配，您将收到一条错误。

如果启用审计日志记录时，您选择了创建新存储桶的选项，则会对其应用正确的权限。不过，如果您在 Simple Storage Service（Amazon S3）中创建自己的存储桶或使用现有存储桶，请确保添加包含存储桶名称的存储桶策略。日志使用服务主体凭证传送。对于大多数 AWS 区域，您可添加 Redshift 服务主体名称，*redshift t.amazonaws.com*。

该存储桶策略使用以下格式。*ServiceName* 和 *BucketName* 是您自己的值的占位符。还可在存储桶策略中指定关联的操作和资源。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Put bucket policy needed for audit logging",
            "Effect": "Allow",
            "Principal": {
                "Service": "ServiceName"
            },
            "Action": [
                "s3:PutObject",
                "s3:GetBucketAcl"
            ],
            "Resource": [
                "arn:aws:s3:::BucketName",
                "arn:aws:s3:::BucketName/*"
            ]
```

```
        }
    ]
}
```

以下示例就是针对美国东部（弗吉尼亚北部）区域的存储桶策略，该存储桶名为 AuditLogs。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Put bucket policy needed for audit logging",
            "Effect": "Allow",
            "Principal": {
                "Service": "redshift.amazonaws.com"
            },
            "Action": [
                "s3:PutObject",
                "s3:GetBucketAcl"
            ],
            "Resource": [
                "arn:aws:s3:::AuditLogs",
                "arn:aws:s3:::AuditLogs/*"
            ]
        }
    ]
}
```

原定设置情况下未启用的区域（也称为“选择加入”区域）需要特定于区域的服务主体名称。对于这些内容，服务主体名称包括区域，格式为 redshift.*region*.amazonaws.com。例如，*redshift.ap-east-1.amazonaws.com* 针对亚太地区（香港）区域。有关默认情况下未启用的区域列表，请参阅《AWS 一般参考》中的管理 AWS 区域。

> **Note**
>
> 特定于区域的服务主体名称与集群所在的区域对应。

日志文件的最佳实践

当 Redshift 将日志文件上载到 Simple Storage Service（Amazon S3）时，可以分段上载大型文件。如果分段上载不成功，则文件的一部分可能会保留在 Simple Storage Service（Amazon S3）存储桶中。这可能会导致额外的存储成本，因此了解什么情况下分段上载会失败非常重要。有关审计日志的分段上载的详细说明，请参阅使用分段上载来上载和复制对象和中止分段上载。

有关创建 S3 存储桶和添加存储桶策略的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的创建通用存储桶和 Amazon S3 的存储桶策略。

Amazon Redshift 审计日志记录的存储桶结构

预设情况下，Amazon Redshift 通过使用下列存储桶和对象结构在 Simple Storage Service（Amazon S3）存储桶中整理日志文件：

AWSLogs/*AccountID*/*ServiceName*/*Region*/*Year*/*Month*/*Day*/*AccountID_ServiceName_Region*

例如：AWSLogs/123456789012/redshift/us-east-1/2013/10/29/123456789012_redshift_us-east-1_mycluster_userlog_2013-10-29T18:01.gz

如果您提供 Simple Storage Service（Amazon S3）键前缀，则该前缀放在密钥开头。

例如，如果指定前缀 myprefix：myprefix/AWSLogs/123456789012/redshift/us-east-1/2013/10/29/123456789012_redshift_us-east-1_mycluster_userlog_2013-10-29T18:01.gz

Simple Storage Service（Amazon S3）键前缀不得超过 512 个字符。其中不得包含空格（ ）、双引号（“）、单引号（‘）、反斜杠（\）。此外还有很多不得使用的特殊字符和控制字符。这些字符的十六进制代码如下：

- x00 – x20
- x 22
- x 27
- x5c
- x7f 或更大

Amazon S3 中的审计日志记录注意事项

Amazon Redshift 审计日志记录可能会因如下原因而中断：

- Amazon Redshift 无权将日志上载到 Simple Storage Service（Amazon S3）存储桶中。验证相应存储桶是否配置了适当的 IAM 策略。有关更多信息，请参阅 Amazon Redshift 审计日志记录的存储桶权限。
- 存储桶拥有者发生变化。当 Amazon Redshift 上载日志时，它会验证存储桶拥有者与启用日志记录时的存储桶拥有者是否相同。如果存储桶拥有者发生变化，则在您配置其他存储桶以用于审计日志记录之前，Amazon Redshift 不会上载日志。
- 找不到存储桶。如果相应存储桶在 Simple Storage Service（Amazon S3）中已被删除，则 Amazon Redshift 无法上载日志。您必须重新创建该存储桶，或对 Amazon Redshift 进行配置以将日志上载到其他存储桶。

使用 AWS CloudTrail 的 API 调用

Amazon Redshift 与 AWS CloudTrail 集成，后者是在 Amazon Redshift 中提供用户、角色或 AWS 服务所采取操作的记录的服务。CloudTrail 将 Amazon Redshift 的所有 API 调用作为事件捕获。有关 Amazon Redshift 与 AWS CloudTrail 集成的更多信息，请参阅使用 CloudTrail 进行日志记录。

您可以将 CloudTrail 与 Amazon Redshift 数据库审计日志记录一起使用，也可以单独使用前者。

要了解有关 CloudTrail 的更多信息，请参阅 AWS CloudTrail 用户指南。

## 启用审计日志记录

配置 Amazon Redshift 以导出审计日志数据。日志可以导出到 CloudWatch，也可以作为文件导出到 Simple Storage Service (Amazon S3) 存储桶。

使用控制台启用审计日志记录

控制台步骤

要启用集群的审计日志记录

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Clusters（集群），然后选择要更新的集群。
3. 选择 Properties（属性）选项卡。在 Database configurations（数据库配置）面板中，依次选择 Edit（编辑）和 Edit audit logging（编辑审计日志记录）。
4. 在 Edit audit logging（编辑审计日志记录）页上，选择 Turn on（打开）并选择 S3 bucket（S3 存储桶）或 CloudWatch。我们建议使用 CloudWatch，因为管理很简单，而且它具有实现数据可视化的有用功能。

5. 选择要导出的日志。
6. 要保存您的选择，请选择 Save changes（保存更改）。

## 安全日志记录

如果 Amazon Redshift 记录引用一个或多个 AWS Glue Data Catalog 视图的查询，则 Amazon Redshift 在记录有关该查询的元数据时，会自动屏蔽某些系统表列和视图列中的字段。

安全日志屏蔽适用于 Amazon Redshift 在运行符合屏蔽条件的查询时生成的所有系统表和视图条目。下表列出了应用了安全日志记录、用 `******` 屏蔽文本和用 `-1` 屏蔽数字的系统视图和列。用于屏蔽文本的星号数量与原始文本中的字符数相匹配，最多 6 个字符。长度超过 6 个字符的字符串仍显示为 6 个星号。

| 系统表 | 敏感列 |
| --- | --- |
| SYS_EXTERNAL_QUERY_DETAIL | 列：source_type、total_partitions、qualified_partitions、scanned_files、returned_rows、returned_bytes、file_format、file_location、external_query_text、warning_message。 |
| SYS_EXTERNAL_QUERY_ERROR | 列：file_location、rowid、column_name、original_value、modified_value、trigger、action、action_value、error_code。 |
| SYS_QUERY_DETAIL | 列：step_id、step_name、table_id、table_name、input_bytes、input_rows、output_bytes、output_rows、blocks_read、blocks_write、local_read_IO、remote_read_IO、spilled_block_local_disk、spilled_block_remote_disk、step_attribute。 |
| SYS_QUERY_HISTORY | 列：returned_rows、returned_bytes。 |
| STL_AGGR | 列：rows、bytes、tbl、type。 |
| STL_BCAST | 列：rows、bytes、packets。 |
| STL_DDLTEXT | 列：label、text。 |
| STL_DELETE | 列：rows、tbl。 |
| STL_DIST | 列：rows、bytes、packets。 |

| 系统表 | 敏感列 |
| --- | --- |
| STL_ERROR | 列：file、linenum、context、error。 |
| STL_EXPLAIN | 列：plannode、info。 |
| STL_FILE_SCAN | 列：name、line、bytes。 |
| STL_HASH | 列：rows、bytes、tbl、est_rows。 |
| STL_HASHJOIN | 列：rows、tbl、num_parts、join_type。 |
| STL_INSERT | 列：rows、tbl。 |
| STL_LIMIT | 列：rows。 |
| STL_MERGE | 列：rows。 |
| STL_MERGEJOIN | 列：rows、tbl。 |
| STL_NESTLOOP | 列：rows、tbl。 |
| STL_PARSE | 列：rows。 |
| STL_PLAN_INFO | 列：startupcost、totalcost、rows、bytes。 |
| STL_PROJECT | 列：rows、tbl。 |
| STL_QUERY | 列：querytxt。 |
| STL_QUERY_METRICS | 列：max_rows、rows、max_blocks_read、blocks_read、max_blocks_to_disk、blocks_to_disk、max_query_scan_size、query_scan_size。 |
| STL_QUERYTEXT | 列：text。 |
| STL_RETURN | 列：rows、bytes。 |
| STL_S3CLIENT | 列：bucket、key、transfer_size、data_size。 |
| STL_S3CLIENT_ERROR | 列：bucket、key、error、transfer_size。 |

| 系统表 | 敏感列 |
| --- | --- |
| STL_SAVE | 列：rows、bytes、tbl。 |
| STL_SCAN | 列：rows、bytes、fetches、type、tbl、rows_pre_filter、rows_pre_user_filter、perm_table_name、scanned_mega_value。 |
| STL_SORT | 列：rows、bytes、tbl。 |
| STL_SSHCLIENT_ERROR | 列：ssh_username、endpoint、command、error。 |
| STL_TR_CONFLICT | 列：table_id。 |
| STL_UNDONE | 列：table_id。 |
| STL_UNIQUE | 列：rows、type、bytes。 |
| STL_UTILITYTEXT | 列：label、text。 |
| STL_WINDOW | 列：rows。 |
| STV_BLOCKLIST | 列：col、tbl、num_values、minvalue、maxvalue。 |
| STV_EXEC_STATE | 列：rows、bytes、label。 |
| STV_INFLIGHT | 列：label、text。 |
| STV_LOCKS | 列：table_id。 |
| STV_QUERY_METRICS | 列：rows、max_rows、blocks_read、max_blocks_read、max_blocks_to_disk、blocks_to_disk、max_query_scan_size、query_scan_size。 |
| STV_STARTUP_RECOVERY_STATE | 列：table_id、table_name。 |
| STV_TBL_PERM | 列：id、name、rows、sorted_rows、temp、block_count、query_scan_size。 |

| 系统表 | 敏感列 |
| --- | --- |
| STV_TBL_TRANS | 列：id、rows、size。 |
| SVCS_EXPLAIN | 列：plannode、info。 |
| SVCS_PLAN_INFO | 列：rows、bytes。 |
| SVCS_QUERY_SUMMARY | 列：step、rows、bytes、rate_row、rate_byte、label、rows_pre_filter。 |
| SVCS_S3LIST | 列：bucket、prefix、retrieved_files、max_file_size、avg_file_size。 |
| SVCS_S3LOG | 列：message。 |
| SVCS_S3PARTITION_SUMMARY | 列：total_partitions、qualified_partitions、min_assigned_partitions、max_assigned_partitions、avg_assigned_partitions。 |
| SVCS_S3QUERY_SUMMARY | 列：external_table_name、file_format、s3_scanned_rows、s3_scanned_bytes、s3query_returned_rows、s3query_returned_bytes。 |
| SVL_QUERY_METRICS | 列：step_label、scan_row_count、join_row_count、nested_loop_join_row_count、return_row_count、spectrum_scan_row_count、spectrum_scan_size_mb。 |
| SVL_QUERY_METRICS_SUMMARY | 列：step_label、scan_row_count、join_row_count、nested_loop_join_row_count、return_row_count、spectrum_scan_row_count、spectrum_scan_size_mb。 |
| SVL_QUERY_REPORT | 列：rows、bytes、label、rows_pre_filter。 |
| SVL_QUERY_SUMMARY | 列：rows、bytes、rows_pre_filter。 |
| SVL_S3LIST | 列：bucket、prefix、retrieved_files、max_file_size、avg_file_size。 |
| SVL_S3LOG | 列：message。 |
| SVL_S3PARTITION | 列：rows、bytes、label、rows_pre_filter。 |

| 系统表 | 敏感列 |
| --- | --- |
| SVL_S3PARTITION_SUMMARY | 列：total_partitions、qualified_partitions、min_assigned_partitions、max_assigned_partitions、avg_assigned_partitions。 |
| SVL_S3QUERY | 列：external_table_name、file_format、s3_scanned_rows、s3_scanned_bytes、s3query_returned_rows、s3query_returned_bytes、files。 |
| SVL_S3QUERY_SUMMARY | 列：external_table_name、file_format、s3_scanned_rows、s3_scanned_bytes、s3query_returned_rows、s3query_returned_bytes。 |
| SVL_S3RETRIES | 列：file_size、location、message。 |
| SVL_SPECTRUM_SCAN_ERROR | 列：location、rowid、colname、original_value、modified_value、trigger、action、action_value、error_code。 |
| SVL_STATEMENTTEXT | 列：type、text。 |
| SVL_STORED_PROC_CALL | 列：querytxt。 |
| SVL_STORED_PROC_MESSAGES | 列：message、linenum、querytext。 |
| SVL_UDF_LOG | 列：message、funcname。 |
| SVV_DISKUSAGE | 列：name、col、tbl、blocknum、num_values、minvalue、maxvalue。 |
| SVV_QUERY_STATE | 列：rows、bytes、label。 |
| SVV_TABLE_INFO | 列：table_id、table。 |
| SVV_TRANSACTIONS | 列：relation。 |

有关系统表和视图的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的系统表和视图参考。有关 Amazon Redshift 的动态屏蔽查询结果的功能的信息，请参阅《Amazon Redshift 数据库开

发人员指南》中的动态数据掩蔽。有关使用 Amazon Redshift 在 AWS Glue Data Catalog 中创建视图的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 AWS Glue Data Catalog 视图。

# 使用 Cloudtrail 进行日志记录

Amazon Redshift、数据共享、Amazon Redshift Serverless、Amazon Redshift 数据 API 和查询编辑器 v2 都与 AWS CloudTrail 集成。CloudTrail 服务提供用户、角色或 AWS 服务在 Amazon Redshift 中所执行的操作的记录。CloudTrail 将 Amazon Redshift 的所有 API 调用作为事件捕获。捕获的调用包括通过 Redshift 控制台进行的调用以及对 Redshift 操作的代码调用。

如果您创建 CloudTrail 跟踪记录，则可以让 CloudTrail 事件持续传送到 Amazon S3 桶（包括 Redshift 的事件）。如果您不配置跟踪记录，则仍可在 CloudTrail 控制台中的 Event history（事件历史记录）中查看最新事件。通过使用 CloudTrail 收集的信息，您可以确定某些事项。这些事项包括向 Redshift 发出的请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间，以及其他详细信息。

您可以将 CloudTrail 与 Amazon Redshift 数据库审计日志记录一起使用，也可以单独使用前者。

要了解有关 CloudTrail 的更多信息，请参阅 AWS CloudTrail 用户指南。

## CloudTrail 中的 信息

在您创建账户时，您的 AWS 账户中已启用 CloudTrail。发生活动时，该活动将记录在 CloudTrail 事件中，并与其他 AWS 服务事件一同保存在 Event history（事件历史记录）中。您可以在 AWS 账户中查看、搜索和下载最新事件。有关更多信息，请参阅《AWS CloudTrail 用户指南》中的使用 CloudTrail 事件历史记录查看事件。

要持续记录 AWS 账户中的事件（包括 Redshift 的事件），请创建跟踪记录。CloudTrail 使用跟踪记录 将日志文件传送至 Amazon S3 桶。预设情况下，在控制台中创建跟踪时，此跟踪应用于所有 AWS 区域。此跟踪记录在 AWS 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Amazon S3 存储桶。此外，您可以配置其他AWS服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅《AWS CloudTrail 用户指南》中的以下内容：

- 创建跟踪概览
- CloudTrail 支持的服务和集成
- 为 CloudTrail 配置 Amazon SNS 通知
- 从多个区域接收 CloudTrail 日志文件和从多个账户接收 CloudTrail 日志文件

所有 Amazon Redshift、Amazon Redshift Serverless、数据 API、数据共享和查询编辑器 v2 操作都由 CloudTrail 记录。例如，对 `AuthorizeDatashare`、`CreateNamespace`、`ExecuteStatement` 和 `CreateConnection` 操作的调用将在 CloudTrail 日志文件中生成条目。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根凭证还是用户凭证发出的。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他AWS服务发出。

有关更多信息，请参阅《AWS CloudTrail 用户指南》中的 CloudTrail userIdentity 元素。

## 日志文件条目

跟踪记录 是一种配置，允许将事件作为日志文件传送到您指定的 Amazon S3 桶。CloudTrail 日志文件包含一个或多个日记账条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

## Amazon Redshift 数据共享示例

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 `AuthorizeDataShare` 操作。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE:janedoe",
        "arn": "arn:aws:sts::111122223333:user/janedoe",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE:janedoe",
                "arn": "arn:aws:sts::111122223333:user/janedoe",
                "accountId": "111122223333",
                "userName": "janedoe"
            },
            "attributes": {
                "creationDate": "2021-08-02T23:40:45Z",
```

```
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2021-08-02T23:40:58Z",
    "eventSource": "redshift.amazonaws.com",
    "eventName": "AuthorizeDataShare",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "3.227.36.75",
    "userAgent":"aws-cli/1.18.118 Python/3.6.10
 Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 botocore/1.17.41",
    "requestParameters": {
        "dataShareArn": "arn:aws:redshift:us-
east-1:111122223333:datashare:4c64c6ec-73d5-42be-869b-b7f7c43c7a53/testshare",
        "consumerIdentifier": "555555555555"
    },
    "responseElements": {
        "dataShareArn": "arn:aws:redshift:us-
east-1:111122223333:datashare:4c64c6ec-73d5-42be-869b-b7f7c43c7a53/testshare",
        "producerNamespaceArn": "arn:aws:redshift:us-
east-1:123456789012:namespace:4c64c6ec-73d5-42be-869b-b7f7c43c7a53",
        "producerArn": "arn:aws:redshift:us-
east-1:111122223333:namespace:4c64c6ec-73d5-42be-869b-b7f7c43c7a53",
        "allowPubliclyAccessibleConsumers": true,
        "dataShareAssociations": [
            {
                "consumerIdentifier": "555555555555",
                "status": "AUTHORIZED",
                "createdDate": "Aug 2, 2021 11:40:56 PM",
                "statusChangeDate": "Aug 2, 2021 11:40:57 PM"
            }
        ]
    },
    "requestID": "87ee1c99-9e41-42be-a5c4-00495f928422",
    "eventID": "03a3d818-37c8-46a6-aad5-0151803bdb09",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

## Amazon Redshift Serverless 示例

Amazon Redshift Serverless 与 AWS CloudTrail 集成，以提供在 Amazon Redshift Serverless 中执行的操作的记录。CloudTrail 将 Amazon Redshift Serverless 的所有 API 调用作为事件捕获。有关 Amazon Redshift Serverless 功能的更多信息，请参阅 Amazon Redshift Serverless 功能概览。

下面的示例显示了一个 CloudTrail 日志条目，该条目演示了 `CreateNamespace` 操作。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AAKEOFPINEXAMPLE:admin",
        "arn": "arn:aws:sts::111111111111:assumed-role/admin/admin",
        "accountId": "111111111111",
        "accessKeyId": "AAKEOFPINEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AAKEOFPINEXAMPLE",
                "arn": "arn:aws:iam::111111111111:role/admin",
                "accountId": "111111111111",
                "userName": "admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-03-21T20:51:58Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-03-21T23:15:40Z",
    "eventSource": "redshift-serverless.amazonaws.com",
    "eventName": "CreateNamespace",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "56.23.155.33",
    "userAgent": "aws-cli/2.4.14 Python/3.8.8 Linux/5.4.181-109.354.amzn2int.x86_64
 exe/x86_64.amzn.2 prompt/off command/redshift-serverless.create-namespace",
    "requestParameters": {
        "adminUserPassword": "HIDDEN_DUE_TO_SECURITY_REASONS",
        "adminUsername": "HIDDEN_DUE_TO_SECURITY_REASONS",
        "dbName": "dev",
        "namespaceName": "testnamespace"
```

```
    },
    "responseElements": {
        "namespace": {
            "adminUsername": "HIDDEN_DUE_TO_SECURITY_REASONS",
            "creationDate": "Mar 21, 2022 11:15:40 PM",
            "defaultIamRoleArn": "",
            "iamRoles": [],
            "logExports": [],
            "namespaceArn": "arn:aws:redshift-serverless:us-
east-1:111111111111:namespace/befa5123-16c2-4449-afca-1d27cb40fc99",
            "namespaceId": "8b726a0c-16ca-4799-acca-1d27cb403599",
            "namespaceName": "testnamespace",
            "status": "AVAILABLE"
        }
    },
    "requestID": "ed4bb777-8127-4dae-aea3-bac009999163",
    "eventID": "1dbee944-f889-4beb-b228-7ad0f312464",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111111111111",
    "eventCategory": "Management",
}
```

## Amazon Redshift 数据 API 示例

下面的示例显示了一个 CloudTrail 日志条目，该条目演示了 `ExecuteStatement` 操作。

```
{
    "eventVersion":"1.05",
    "userIdentity":{
        "type":"IAMUser",
        "principalId":"AKIAIOSFODNN7EXAMPLE:janedoe",
        "arn":"arn:aws:sts::123456789012:user/janedoe",
        "accountId":"123456789012",
        "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
        "userName": "janedoe"
    },
    "eventTime":"2020-08-19T17:55:59Z",
    "eventSource":"redshift-data.amazonaws.com",
    "eventName":"ExecuteStatement",
    "awsRegion":"us-east-1",
    "sourceIPAddress":"192.0.2.0",
```

```
    "userAgent":"aws-cli/1.18.118 Python/3.6.10
 Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 botocore/1.17.41",
    "requestParameters":{
        "clusterIdentifier":"example-cluster-identifier",
        "database":"example-database-name",
        "dbUser":"example_db_user_name",
        "sql":"***OMITTED***"
    },
    "responseElements":{
        "clusterIdentifier":"example-cluster-identifier",
        "createdAt":"Aug 19, 2020 5:55:58 PM",
        "database":"example-database-name",
        "dbUser":"example_db_user_name",
        "id":"5c52b37b-9e07-40c1-98de-12ccd1419be7"
    },
    "requestID":"00c924d3-652e-4939-8a7a-cd0612eeb8ac",
    "eventID":"c1fb7076-102f-43e5-9ec9-40820bcc1175",
    "readOnly":false,
    "eventType":"AwsApiCall",
    "recipientAccountId":"123456789012"
}
```

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 ExecuteStatement 操作，显示了幂等性所用的 clientToken。

```
{
    "eventVersion":"1.05",
    "userIdentity":{
        "type":"IAMUser",
        "principalId":"AKIAIOSFODNN7EXAMPLE:janedoe",
        "arn":"arn:aws:sts::123456789012:user/janedoe",
        "accountId":"123456789012",
        "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
        "userName": "janedoe"
    },
    "eventTime":"2020-08-19T17:55:59Z",
    "eventSource":"redshift-data.amazonaws.com",
    "eventName":"ExecuteStatement",
    "awsRegion":"us-east-1",
    "sourceIPAddress":"192.0.2.0",
    "userAgent":"aws-cli/1.18.118 Python/3.6.10
 Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 botocore/1.17.41",
    "requestParameters":{
```

```
        "clusterIdentifier":"example-cluster-identifier",
        "database":"example-database-name",
        "dbUser":"example_db_user_name",
        "sql":"***OMITTED***",
        "clientToken":"32db2e10-69ac-4534-b3fc-a191052616ce"
    },
    "responseElements":{
        "clusterIdentifier":"example-cluster-identifier",
        "createdAt":"Aug 19, 2020 5:55:58 PM",
        "database":"example-database-name",
        "dbUser":"example_db_user_name",
        "id":"5c52b37b-9e07-40c1-98de-12ccd1419be7"
    },
    "requestID":"00c924d3-652e-4939-8a7a-cd0612eeb8ac",
    "eventID":"c1fb7076-102f-43e5-9ec9-40820bcc1175",
    "readOnly":false,
    "eventType":"AwsApiCall",
    "recipientAccountId":"123456789012"
}
```

## Amazon Redshift 查询编辑器 v2 示例

下面的示例显示了一个 CloudTrail 日志条目，该条目演示了 `CreateConnection` 操作。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AAKEOFPINEXAMPLE:session",
        "arn": "arn:aws:sts::123456789012:assumed-role/MyRole/session",
        "accountId": "123456789012",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AAKEOFPINEXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/MyRole",
                "accountId": "123456789012",
                "userName": "MyRole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-09-21T17:19:02Z",
```

```
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-09-21T22:22:05Z",
    "eventSource": "sqlworkbench.amazonaws.com",
    "eventName": "CreateConnection",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "192.2.0.2",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:102.0)
 Gecko/20100101 Firefox/102.0",
    "requestParameters": {
        "password": "***",
        "databaseName": "***",
        "isServerless": false,
        "name": "***",
        "host": "redshift-cluster-2.c8robpbxvbf9.ca-central-1.redshift.amazonaws.com",
        "authenticationType": "***",
        "clusterId": "redshift-cluster-2",
        "username": "***",
        "tags": {
            "sqlworkbench-resource-owner": "AAKEOFPINEXAMPLE:session"
        }
    },
    "responseElements": {
        "result": true,
        "code": "",
        "data": {
            "id": "arn:aws:sqlworkbench:ca-central-1:123456789012:connection/ce56b1be-
dd65-4bfb-8b17-12345123456",
            "name": "***",
            "authenticationType": "***",
            "databaseName": "***",
            "secretArn": "arn:aws:secretsmanager:ca-
central-1:123456789012:secret:sqlworkbench!7da333b4-9a07-4917-b1dc-12345123456-qTCoFm",
            "clusterId": "redshift-cluster-2",
            "dbUser": "***",
            "userSettings": "***",
            "recordDate": "2022-09-21 22:22:05",
            "updatedDate": "2022-09-21 22:22:05",
            "accountId": "123456789012",
            "tags": {
                "sqlworkbench-resource-owner": "AAKEOFPINEXAMPLE:session"
            },
```

```
            "isServerless": false
        }
    },
    "requestID": "9b82f483-9c03-4cdd-bb49-a7009e7da714",
    "eventID": "a7cdd442-e92f-46a2-bc82-2325588d41c3",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

## AWS CloudTrail 日志中的 Amazon Redshift 账户 ID

当 Amazon Redshift 为您调用其他 AWS 服务时，该次调用将以属于 Amazon Redshift 的账户 ID 记录下来。它未以您的账户 ID 记录。例如，假设 Amazon Redshift 调用 AWS Key Management Service（AWS KMS）操作（如 `CreateGrant`、`Decrypt`、`Encrypt` 和 `RetireGrant`）以管理集群上的加密。在这种情况下，调用由 AWS CloudTrail 使用 Amazon Redshift 账户 ID 记录。

当调用其他 AWS 服务时，Amazon Redshift 使用下表中的账户 ID。

| 区域 | 区域 | 账户 ID |
| --- | --- | --- |
| 美国东部（弗吉尼亚北部）区域 | us-east-1 | 368064434614 |
| 美国东部（俄亥俄）区域 | us-east-2 | 790247189693 |
| 美国西部（加利福尼亚北部）区域 | us-west-1 | 703715109447 |
| 美国西部（俄勒冈州）区域 | us-west-2 | 473191095985 |
| Africa (Cape Town) Region | af-south-1 | 420376844563 |
| 亚太地区（香港）区域 | ap-east-1 | 651179539253 |
| 亚太地区（海得拉巴）区域 | ap-south-2 | 297058826802 |
| 亚太地区（雅加达）区域 | ap-southeast-3 | 623197973179 |
| 亚太地区（马来西亚）区域 | ap-southeast-5 | 590184011157 |

| 区域 | 区域 | 账户 ID |
| --- | --- | --- |
| 亚太地区（墨尔本）区域 | ap-southeast-4 | 945512339897 |
| 亚太（孟买）区域 | ap-south-1 | 408097707231 |
| 亚太地区（新西兰） | ap-southeast-6 | 730335362089 |
| 亚太地区（大阪）区域 | ap-northeast-3 | 398671365691 |
| 亚太区域（首尔） | ap-northeast-2 | 713597048934 |
| 亚太区域（新加坡） | ap-southeast-1 | 960118270566 |
| 亚太区域（悉尼） | ap-southeast-2 | 485979073181 |
| 亚太地区（台北） | ap-east-2 | 211125526206 |
| 亚太地区（泰国）区域 | ap-southeast-7 | 767397930036 |
| 亚太区域（东京） | ap-northeast-1 | 615915377779 |
| 加拿大（中部）区域 | ca-central-1 | 764870610256 |
| 加拿大西部（卡尔加里）区域 | ca-west-1 | 830903446466 |
| 欧洲（法兰克福）区域 | eu-central-1 | 434091160558 |
| 欧洲（爱尔兰）区域 | eu-west-1 | 246478207311 |
| 欧洲（伦敦）区域 | eu-west-2 | 885798887673 |
| Europe (Milan) Region | eu-south-1 | 041313461515 |
| 欧洲（巴黎）区域 | eu-west-3 | 694668203235 |
| 欧洲地区（西班牙）区域 | eu-south-2 | 028811157404 |
| 欧洲地区（斯德哥尔摩）区域 | eu-north-1 | 553461782468 |
| 欧洲（苏黎世） | eu-central-2 | 668912161003 |

| 区域 | 区域 | 账户 ID |
| --- | --- | --- |
| 以色列（特拉维夫）区域 | il-central-1 | 901883065212 |
| 墨西哥（中部）区域 | mx-central-1 | 058264411980 |
| Middle East (Bahrain) Region | me-south-1 | 051362938876 |
| 中东（阿联酋）区域 | me-central-1 | 595013617770 |
| 南美洲（圣保罗）区域 | sa-east-1 | 392442076723 |

以下示例显示了 Amazon Redshift 所调用的 AWS KMS Decrypt 操作的 CloudTrail 日志条目。

```
{

    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAI5QPCMKLTL4VHFCYY:i-0f53e22dbe5df8a89",
        "arn": "arn:aws:sts::790247189693:assumed-role/prod-23264-role-wp/
i-0f53e22dbe5df8a89",
        "accountId": "790247189693",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2017-03-03T16:24:54Z"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAI5QPCMKLTL4VHFCYY",
                "arn": "arn:aws:iam::790247189693:role/prod-23264-role-wp",
                "accountId": "790247189693",
                "userName": "prod-23264-role-wp"
            }
        }
    },
    "eventTime": "2017-03-03T17:16:51Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-east-2",
```

```
    "sourceIPAddress": "52.14.143.61",
    "userAgent": "aws-internal/3",
    "requestParameters": {
        "encryptionContext": {
            "aws:redshift:createtime": "20170303T1710Z",
            "aws:redshift:arn": "arn:aws:redshift:us-east-2:123456789012:cluster:my-dw-
instance-2"
        }
    },
    "responseElements": null,
    "requestID": "30d2fe51-0035-11e7-ab67-17595a8411c8",
    "eventID": "619bad54-1764-4de4-a786-8898b0a7f40c",
    "readOnly": true,
    "resources": [
        {
            "ARN": "arn:aws:kms:us-east-2:123456789012:key/f8f4f94f-e588-4254-
b7e8-078b99270be7",
            "accountId": "123456789012",
            "type": "AWS::KMS::Key"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012",
    "sharedEventID": "c1daefea-a5c2-4fab-b6f4-d8eaa1e522dc"

}
```


## 系统表与 S3 表类数据存储服务集成

使用系统表与 Amazon S3 表类数据存储服务的集成，可将您的 Amazon Redshift 系统表数据保留超过 7 天的集群内保留期。这有助于您满足合规性、审计和可观测性要求。启用此功能后，Redshift 会将系统表日志以 Apache Iceberg 格式写入 S3 表类数据存储服务，并管理分区、压缩和快照维护。您可以根据分析和审计需要，将这些数据保留任意长的时间。由于数据以开放的 Iceberg 格式存储，您可以使用 Redshift、Amazon Athena 或任何其它与 Iceberg 兼容的引擎进行查询。您还可以使用人工智能代理技能进行自然语言查询（请参阅 GitHub 网站上的 System table skills for AWS Agent Toolkit）。

当您通过控制台、Redshift CLI 或 SDK 启用此功能后，Redshift 很快便会开始以固定频率将您所选系统表中的新行写入 S3 表类数据存储服务。您可以选择要发布哪些系统表；S3 表类数据存储服务将强制执行您所配置的保留期。

这支持以下使用案例：

- 满足合规性和审计要求。将查询、连接和更改历史记录保留数月或数年。
- 从一个位置监控您的实例集。整合来自您账户和区域中多个数据仓库的监控数据，并分析整个实例集的活动。
- 不再需要自定义提取、转换、加载（ETL）管道。停止构建和维护将系统表数据导出到 S3 以延长保留期的作业。
- 调查过往事件。查询超出 7 天集群内时段的历史数据，以进行根本原因分析、事件后审查，并监控工作负载随时间的演变及性能趋势。

此功能适用于 Amazon Redshift 预调配 RA3 和 RG 实例以及 Amazon Redshift Serverless，覆盖所有支持 Redshift 和 S3 表类数据存储服务的 AWS 商业区域。在本主题中，数据仓库指的是预调配集群或 Redshift Serverless 工作组。

主题


- 工作原理
- 启用系统表与 S3 表类数据存储服务的集成
- 支持的系统表
- 向每个表中添加的元数据列
- 配置保留期
- 数据治理
- 传输和生命周期行为
- 最佳实践
- 计费
- 注意事项和限制


## 工作原理

启用该功能并选择一个或多个受支持的系统表后，Redshift 会将这些系统表中新完成的记录写入 S3 表类数据存储服务。数据按固定频率分批传输。

系统表存在于您的 AWS 账户中，但由服务管理：AWS 拥有并运营数据传输和表管理，包括以下各项：

- 在您的账户中创建 S3 表存储桶、命名空间和 S3 表。
- 定义和演进表架构。

- 将数据以 Apache Iceberg 格式写入 S3 表中。
- 压缩数据并维护快照（由 S3 表类数据存储服务自动处理）。

无需构建或维护基础设施，也不会对您的工作负载产生任何影响。

您可以通过 S3 表类数据存储服务记录过期功能配置保留数据的时长。有关更多信息，请参阅 the section called “配置保留期”。

由于表由 AWS 管理，因此您无法修改或删除 Redshift 传输的行。传输的数据是不可变的。您只能控制对表的读取权限。有关更多信息，请参阅 the section called “数据治理”。

只有已完成的活动才会传输到 S3 表类数据存储服务。数据仓库中的系统表可以显示正在进行的活动，但只复制已达到最终状态的记录。例如，将传输已完成、已中止或已取消的查询，但仍在运行的查询要等到达到最终状态后才会传输。

## 启用系统表与 S3 表类数据存储服务的集成

要启用系统表与 S3 表类数据存储服务的集成，请将现有 Redshift API 与新的日志目标类型 `s3table` 结合使用。启用后，Redshift 会在您的账户中创建一个名为 `aws-redshift` 的 S3 表存储桶。
您可以使用与用于其它日志目标的相同操作来管理该功能：对于预调配集群，使用 `enable-logging`、`disable-logging` 和 `describe-logging-status`；对于 Redshift Serverless，使用 `update-namespace` 和 `get-namespace`。

权限

启用、修改或禁用该功能的主体必须具有以下权限：

- `redshift:EnableLogging`（适用于预调配集群）或 `redshift-serverless:UpdateNamespace`（适用于 Redshift Serverless）。
- 创建和配置 S3 表存储桶的权限。启用该功能的主体需要以下 S3 表类数据存储服务权限：
  - `s3tables:CreateTableBucket`
  - `s3tables:PutTableBucketEncryption`
  - `s3tables:PutTableBucketPolicy`

  启用该功能后，Redshift 使用触发操作的主体的身份创建和配置 S3 表存储桶。

创建存储桶后，Redshift 利用 Redshift 与 S3 表类数据存储服务之间的服务信任关系，在该存储桶内创建命名空间和表。启用该功能的主体无需具备创建命名空间或表的权限。

部署选项

您可以通过以下两种模式之一配置传输。给定的数据仓库一次使用一种模式。

| 部署选项 | 说明 |
| --- | --- |
| 按仓库 | Redshift 将每个数据仓库的系统表数据写入其各自的一组 S3 表中。表中的每一行均来自单个数据仓库。适用于需要在仓库之间实现物理隔离的场景。 |
| 已合并 | Redshift 将同一账户和区域内多个数据仓库的系统表数据写入一组共享的 S3 表中。来自不同仓库的行通过 `warehouse_namespace_arn` 和 `warehouse_name` 列加以区分。适用于集中式、跨仓库分析。 |

部署模型决定了数据在 S3 表类数据存储服务中的组织方式。在按仓库模型中，S3 表类数据存储服务命名空间名称包含数据仓库的唯一标识符，因此，每个仓库的数据在物理上隔离到其自己的命名空间中。在合并模型中，S3 表类数据存储服务命名空间名称包含 AWS 账号，而该账户和区域中所有仓库的数据统一存储在相同的命名空间和表中。

这两个选项均支持在单个 AWS 账户和单个 AWS 区域内使用。要分析跨区域或跨账户的数据，请在查询时合并每个区域或账户的表中的结果。要进行跨账户访问，您可以使用 AWS Glue 数据目录共享。有关更多信息，请参阅《AWS Glue 开发人员指南》中的授予跨账户访问权限。

> **Note**
>
> 如果您将数据仓库从一种部署模式切换到另一种部署模式，则先前传输的数据将保留在其现有表中，并根据其配置的保留期进行保留。Redshift 开始向新目标传输新数据，而不会回填历史数据。

使用 控制台

配置系统表与 S3 表类数据存储服务的集成

1. 打开 Redshift 控制台，在导航窗格中的 Amazon Redshift 或 Amazon Redshift Serverless 下，选择系统表集成。您也可以从集群或命名空间详细信息页面进行配置，方法是选择集成选项卡，或者依次选择操作、集成、配置系统表集成。
2. 选择创建系统表集成。

3. 选择您的预调配集群或 Redshift Serverless 工作组。
4. 选择要发布的系统表。选择单个 SYS_* 视图，或选择选择所有支持的系统表以发布所有当前及未来支持的视图。如果选择全部，则将来添加的新视图将自动包含在内，无需更改配置。
5. 选择部署模型：
   - 每个数据仓库、每个系统表对应单独的 S3 表，用于将该仓库的数据保存在其自己的一组表中。
   - 跨数据仓库的每个系统表的共享 S3 表，用于将账户中多个仓库的数据整合到共享的一组表中。
6. （可选）选择加密密钥。默认情况下，数据使用 Amazon S3 托管式密钥（SSE-S3）进行加密。要使用自己的 AWS KMS 密钥，请在此处选择。如果您使用自己的 AWS KMS 密钥，则密钥策略必须向 Redshift 和 S3 表类数据存储服务授予访问该密钥的权限。有关更多信息，请参阅 the section called “配置加密”。
7. 保存更改。Redshift 开始将选定的系统表发布到 S3 表类数据存储服务，并持续定期添加新记录。

要停止发布某个系统表，请返回设置并将其移除。已发布的数据将持续保留，直到您通过 S3 表类数据存储服务配置过期。

您可以从集群或命名空间详细信息页面查看集成的状态。展开查看 S3 表映射，以查看每个系统表与其对应的 S3 表名称及命名空间之间的映射关系。您也可以从 S3 表类数据存储服务控制台查看已发布的数据。

使用 AWS CLI

以下参数用于配置 S3 表类数据存储服务发布：

| 参数 | 适用于 | 说明 |
| --- | --- | --- |
| `--log-destination-type` | 二者 | 设置为 `s3table` 以发布到 S3 表类数据存储服务。其它值为 `s3` 和 `cloudwatch` 。 |
| `--log-exports` | 已预置 | 要发布的系统表，以 SYS_* 视图名称列表的形式指定，或为 `all`。 |
| `--s3-table-names` | Serverless | 要发布的系统表，以 SYS_* 视图名称列表的形式指定，或为 `all`。 |
| `--s3-table-action` | Serverless | `Enable` 或 `Disable` S3 表类数据存储服务发布。 |

| 参数 | 适用于 | 说明 |
| --- | --- | --- |
| --s3-table-granularity | 二者 | 表范围。预调配：cluster（默认）或 account。无服务器：namespace （默认）或 account。 |
| --s3-table-kms-key-id | 二者 | 用于加密的可选 AWS KMS 密钥 ARN 或 ID。默认为 Amazon S3 托管式密钥（SSE-S3）。 |

预置集群

启用传输。--log-exports 接受 all 或以空格分隔的受支持系统表列表，--s3-table-granularity 接受 cluster（按仓库）或 account（已合并）。

```
aws redshift enable-logging \
    --cluster-identifier my-redshift-cluster \
    --log-destination-type s3table \
    --log-exports all \
    --s3-table-granularity account
```

要仅为特定系统表启用传输：

```
aws redshift enable-logging \
    --cluster-identifier my-redshift-cluster \
    --log-destination-type s3table \
    --log-exports sys_query_history sys_query_text sys_userlog \
    --s3-table-granularity cluster
```

检查传输状态。

```
aws redshift describe-logging-status \
    --cluster-identifier my-redshift-cluster
```

禁用传输（适用于所有系统表），或对于特定表使用 --log-exports：

```
aws redshift disable-logging \
    --cluster-identifier my-redshift-cluster \
    --log-destination-type s3table \
    --log-exports sys_stream_scan_states
```

Redshift Serverless

在 Redshift Serverless 中，您可以使用 `update-namespace` 为每个命名空间配置该功能。系统表名称通过 `--s3-table-names`（而非 `--log-exports`）传递。

启用传输。

```
aws redshift-serverless update-namespace \
    --namespace-name my-namespace \
    --log-destination-type s3table \
    --s3-table-action Enable \
    --s3-table-names all \
    --s3-table-granularity namespace
```

对特定的系统表禁用传输（或 `all`）：

```
aws redshift-serverless update-namespace \
    --namespace-name my-namespace \
    --log-destination-type s3table \
    --s3-table-action Disable \
    --s3-table-names sys_stream_scan_states
```

检查传输状态。

```
aws redshift-serverless get-namespace \
    --namespace-name my-namespace
```

要在删除 S3 表存储桶或表后恢复传输，请再次运行 enable 命令。

向 AWS Glue Data Catalog 注册

在使用 Redshift、Athena 或其它分析服务查询保留的数据之前，服务托管的 S3 表存储桶必须先与 AWS Glue Data Catalog 集成。每个账户和区域只需执行此步骤一次。如果您已将 S3 表类数据存储服务与 AWS Glue Data Catalog 集成，则无需执行任何其它操作。

有关集成说明，请参阅《AWS Glue 开发人员指南》中的将 Amazon S3 表类数据存储服务与 AWS Glue Data Catalog 集成。

### 配置加密

默认情况下，传输到 S3 表类数据存储服务的数据使用 Amazon S3 托管式密钥（SSE-S3）进行加密。要改用 AWS KMS 客户自主管理型密钥，请在启用传输时使用 `--s3-table-kms-key-id` 指定其 ARN：

```
aws redshift enable-logging \
    --cluster-identifier my-redshift-cluster \
    --log-destination-type s3table \
    --log-exports all \
    --s3-table-granularity account \
    --s3-table-kms-key-id arn:aws:kms:us-west-2:111122223333:key/key-id
```

如果您使用客户自主管理型密钥，则其密钥策略必须支持 Redshift 系统表集成服务主体在写入您的系统表时生成数据密钥，并可让 S3 表类数据存储服务维护服务主体在表维护（例如压缩）期间使用该密钥。将以下语句添加到密钥策略中，用您自己的值替换 *REGION*、*ACCOUNT* 和 *KEY_ID*。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnableRedshiftSystemTableKeyUsage",
      "Effect": "Allow",
      "Principal": {
        "Service": "systemtables.redshift.amazonaws.com"
      },
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:REGION:ACCOUNT:key/KEY_ID",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "ACCOUNT"
        }
      }
    },
    {
      "Sid": "EnableS3TableMaintenanceKeyUsage",
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": "maintenance.s3tables.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:REGION:ACCOUNT:key/KEY_ID",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3tables:REGION:ACCOUNT:bucket/
aws-redshift/*"
        }
      }
    }
  ]
}
```

如果密钥策略未授予这些权限，则 Redshift 无法写入加密的 S3 表类数据存储服务，传输将失败。

## 支持的系统表

您可以选择以下任一 SYS_* 监控视图来与 S3 表类数据存储服务集成。要了解每个视图中各列的描述，请选择视图名称。

- SYS_ANALYZE_COMPRESSION_HISTORY
- SYS_ANALYZE_HISTORY
- SYS_AUTOMATIC_OPTIMIZATION
- SYS_AUTO_TABLE_OPTIMIZATION
- SYS_CHILD_QUERY_TEXT
- SYS_CONNECTION_LOG
- SYS_COPY_JOB_INFO
- SYS_COPY_REPLACEMENTS
- SYS_DATASHARE_CHANGE_LOG
- SYS_DATASHARE_USAGE_CONSUMER
- SYS_DATASHARE_USAGE_PRODUCER
- SYS_DATASHARE_WRITE_HISTORY
- SYS_EXTERNAL_QUERY_ERROR
- SYS_INTEGRATION_ACTIVITY

- SYS_MV_STATE
- SYS_PROCEDURE_MESSAGES
- SYS_QUERY_DETAIL
- SYS_QUERY_EXPLAIN
- SYS_QUERY_HISTORY
- SYS_QUERY_TEXT
- SYS_SCHEMA_QUOTA_VIOLATIONS
- SYS_SESSION_HISTORY
- SYS_SPATIAL_SIMPLIFY
- SYS_STREAM_SCAN_ERRORS
- SYS_STREAM_SCAN_STATES
- SYS_UNLOAD_DETAIL
- SYS_USERLOG
- SYS_VACUUM_HISTORY

> **Note**
>
> 以下视图需要补丁 P203 或更高版本：SYS_CHILD_QUERY_TEXT、SYS_COPY_REPLACEMENTS、SYS_EXTERNAL_QUERY_ERROR、和 SYS_UNLOAD_DETAIL。如果在运行早于 P203 的补丁的数据仓库上启用这些视图，则这些表将使用正确的架构创建，但在数据仓库更新到 P203 或更高版本之前，表中不包含任何数据。

## 向每个表中添加的元数据列

每个 S3 表都包含源 SYS_* 视图中的所有列，以及 Redshift 添加的以下五个元数据列，用于标识每行的来源及其传输时间。

| 列 | 类型 | 说明 |
| --- | --- | --- |
| `warehouse_account_id` | 字符串 | 拥有源数据仓库的 AWS 账户。 |

| 列 | 类型 | 说明 |
| --- | --- | --- |
| warehouse_region_name | 字符串 | 源数据仓库运行所在的 AWS 区域。 |
| warehouse_namespace_arn | 字符串 | 源数据仓库的命名空间的 ARN。这是一个稳定、唯一的标识符，在重命名和重新创建时不可变。 |
| warehouse_name | 字符串 | 源数据仓库的用户可读名称（集群名称或工作组名称）。 |
| s3_tables_ingestion_time | timestamp(6)，UTC | Redshift 将该行提交到 S3 表类数据存储服务的时间。这是传输时间，而非底层事件发生的时间。 |

## 配置保留期

Redshift 不会为其传输到 S3 表类数据存储服务的数据定义保留策略。您可以通过 S3 表类数据存储服务设置记录过期策略，来配置保留数据的时长。您可以指定天数（例如 5 天、100 天或 365 天），S3 表类数据存储服务将自动移除超过所配置的保留时长的数据。

如果您未配置过期策略，数据将无限期保留。

当您禁用集成或移除系统表时，Redshift 会停止写入新数据，但之前传输的数据仍保留在 S3 表类数据存储服务中，并且继续受您配置的任何过期策略约束。

有关更多信息，请参阅《Amazon S3 用户指南》中的为 S3 表存储桶配置记录过期。

## 数据治理

您可以控制对保留的系统表数据的访问权限；Redshift 不会代表您应用读取权限，也不会屏蔽数据。

- 访问控制。将 S3 表存储桶与 AWS Glue Data Catalog 集成后，您可以使用 AWS Identity and Access Management（IAM）或 AWS Lake Formation 来管理读取访问权限。在整合的部署中，您可以按每个仓库限定读取访问权限范围。
- 编目和发现。AWS Glue Data Catalog 集成提供了一个中心位置，用于发现保留的表并应用细粒度权限（例如，使用 Lake Formation 进行表级、列级或行级访问控制）。

## 传输和生命周期行为

- 传输频率。数据按固定频率分批传输。
- 仅传输一次。每条记录传输一次；重新启用或重新添加系统表不会创建重复记录。
- 启用、禁用、重新启用。当该功能（或特定的系统表）处于禁用状态时，不会捕获该时间段内的数据。禁用会停止写入，但不会移除 S3 表类数据存储服务或之前传输的数据。重新启用将恢复后续的传输，而不会回填空缺期间的数据。
- 从您的选择中移除系统表。Redshift 将停止为该系统表传输新数据。之前传输的数据仍保留在 S3 表类数据存储服务中，并受您所配置的记录过期策略约束。重新添加系统表可在不重复数据的情况下恢复传输。
- 删除包含动态数据的数据仓库。如果您在传输过程中删除了数据仓库，且该数据仓库的加密密钥与用于 S3 表类数据存储服务的密钥不同，则 Redshift 可能会在删除后的短时间内继续使用数据仓库密钥上的 AWS KMS 授权，以完成所有剩余数据的传输。

### 检查传输状态

您可以随时查看每个系统表的当前配置和最近的传输时间：

- 预调配集群。`describe-logging-status`（或 `describe-cluster`）返回每个系统表的活动系统表、S3 表类数据存储服务命名空间、粒度和上次摄取时间。
- Redshift Serverless。`get-namespace` 在命名空间的 S3 表类数据存储服务发布状态中返回相同的信息。

### Redshift Serverless

传输操作不会使工作组保持唤醒状态，也不会消耗您的计算资源。在工作组暂停之前，Redshift 会确保所有待处理的批次均已传输完毕。

## 最佳实践

1. 评测您的数据敏感度。了解您的数据仓库中存储了何种数据，以及系统表中是否包含敏感信息。某些系统表（例如 SYS_QUERY_TEXT 和 SYS_PROCEDURE_MESSAGES）可以捕获查询和存储过程中的文本值。
2. 选择部署模型。如果您的数据仓库存储了敏感信息，请考虑使用按仓库部署模型，以便对每个数据仓库的系统表数据保持物理隔离。如果您的数据不敏感，并且您希望在不合并多个表的结果的情况下运行跨仓库查询，请使用合并部署模型。

3. 选择所需的系统表。查看支持的系统表列表，然后选择符合合规性、审计或可观测性要求的表。您无需启用所有系统表。
4. 在启用该功能之前配置加密。如果要使用客户自主管理型 AWS KMS 密钥，请在首次启用该功能时指定该密钥。创建 S3 表类数据存储服务后，您将无法更改密钥。要更改密钥，请禁用该功能，使用 S3 表类数据存储服务 API 删除 S3 表类数据存储服务（这将永久删除保留的数据），然后使用新密钥重新启用该功能。
5. 根据业务需求设置保留期。根据您的合规性、审计或运营需求，直接在 S3 表类数据存储服务中在表级别配置记录过期策略。不同的表可能有不同的保留期。如果您未配置过期策略，数据将无限期保留。要监控系统表的存储使用情况，请参阅《Amazon S3 用户指南》中的 Amazon S3 表类数据存储服务 CloudWatch 指标。

### 计费

将系统表数据写入 S3 表类数据存储服务是免费的。您需要按照所用查询引擎的定价，为保留的数据支付标准 S3 表类数据存储服务的存储和维护费用，以及读取数据所用查询引擎的费用。有关更多信息，请参阅 Amazon S3 表类数据存储服务定价。

### 注意事项和限制

- 在单个 AWS 账户和单个 AWS 区域内支持。要分析跨账户或跨区域的数据，请在查询时合并结果。
- 数据仓库一次只能使用一种部署模式（按仓库部署或合并部署）。
- 切换部署模式、禁用后重新启用，或移除后重新添加系统表，均不会回填历史数据。
- 已传输的数据是不可变的。您无法通过 Redshift 修改或删除单个行。
- 您可以删除此功能创建的 S3 表类数据存储服务，但这样做会永久移除这些表中所有保留的数据并停止传输。Redshift 不会自动重新创建已删除的表。要恢复传输，您必须重新启用该功能，该功能将创建新表并开始继续传输新数据。之前传输的数据不会还原。有关更多信息，请参阅《Amazon S3 用户指南》中的删除 S3 表。
- 传输是基于批次的（数据以固定频率写入）。
- 从 Redshift 进行查询要求将 S3 表存储桶与 AWS Glue Data Catalog 集成。

## Amazon Redshift 的合规性验证

作为多个 AWS 合规性计划的一部分，第三方审计员将评估 Amazon Redshift 的安全性和合规性。其中包括 SOC、PCI、FedRAMP、HIPAA 及其它。

有关特定合规性计划范围内的 AWS 服务的列表，请参阅合规性计划范围内的 AWS 服务。有关一般信息，请参阅 AWS 合规性计划。

您可以使用 AWS Artifact 下载第三方审计报告。有关更多信息，请参阅下载 AWS Artifact 中的报告。

您在使用 Amazon Redshift 时的合规性责任由您数据的敏感性、您组织的合规性目标以及适用的法律法规决定。如果您对 Amazon Redshift 的使用需遵守 HIPAA、PCI 或 FedRAMP 等标准，AWS 提供了以下实用资源：

- 安全性和合规性快速入门指南，介绍了架构注意事项，以及在 AWS 上部署侧重于安全性和合规性的基准环境的步骤。
- 设计符合 HIPAA 安全性和合规性要求的架构白皮书，介绍公司如何使用 AWS 创建符合 HIPAA 标准的应用程序。
- AWS 合规性资源，这些业务手册和指南可能适用于您的行业和位置。
- AWS Config，一项 AWS 服务，可以评估您的资源配置对内部实践、行业指南和法规的遵循情况。
- AWS Security Hub CSPM 是一项 AWS 服务，提供了 AWS 中安全状态的全面视图，可帮助您检查是否符合安全行业标准和最佳实践。Security Hub CSPM 使用安全控件来评估资源配置和安全标准，以帮助您遵守各种合规框架。有关使用 Security Hub CSPM 评估 Amazon Redshift 资源的更多信息，请参阅《AWS Security Hub 用户指南》中的 Amazon Redshift 控件。

以下合规性和安全性文档涵盖 Amazon Redshift，并可通过 AWS Artifact 按需提供。有关更多信息，请参阅 AWS Artifact。

- 云计算合规性控制目录 (C5)
- ISO 27001:2013 适用性声明 (SoA)
- ISO 27001:2013 认证
- ISO 27017:2015 适用性声明 (SoA)
- ISO 27017:2015 认证
- ISO 27018:2015 适用性声明 (SoA)
- ISO 27018:2014 认证
- ISO 9001:2015 认证
- PCI DSS 合规证明 (AOC) 和责任摘要
- 服务组织控制 (SOC) 1 报告
- 服务组织控制 (SOC) 2 报告

- 服务组织控制 (SOC) 2 保密性报告

# Amazon Redshift 中的恢复能力

AWS 全球基础设施围绕 AWS 区域和可用区 (AZ) 构建。AWS区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个数据中心基础设施或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

几乎所有 AWS 区域都有多个可用区和数据中心。您可以跨一个区域中的多个可用区部署您的应用来获得容错性和低延迟。

要将集群移动到另一个可用区，而不丢失任何数据或对您的应用程序进行更改，您可以为您的集群设置重新定位。通过重新定位，您可以在集群上出现服务中断时继续操作，而产生的影响最小。开启集群重新定位后，Amazon Redshift 可能会在某些情况下选择重新定位集群。有关 Amazon Redshift 中的重新定位的更多信息，请参阅重新定位集群。

在可用区中发生意外事件的故障情况下，您可以设置多个可用区（多可用区）部署，以确保 Amazon Redshift 数据仓库能够继续运行。Amazon Redshift 在可通过单个端点访问的两个可用区中部署相等的计算资源。如果整个可用区出现故障，第二个可用区中的剩余计算资源将可用于继续处理工作负载。有关多可用区部署的更多信息，请参阅 多可用区部署。

有关 AWS 区域和可用区的更多信息，请参阅 AWS 全球基础设施。有关使用 Amazon Redshift 进行灾难恢复的更多信息，请参阅 Implement disaster recovery with Amazon Redshift。

.

# Amazon Redshift 中的基础设施安全性

作为一项托管式服务，Amazon Redshift 受 AWS 全球网络安全保护。有关 AWS 安全服务以及 AWS 如何保护基础设施的信息，请参阅 AWS 云安全性。要按照基础设施安全最佳实践设计您的 AWS 环境，请参阅《安全性支柱 AWS Well-Architected Framework》中的基础设施保护。

您可以使用 AWS 发布的 API 调用通过网络访问 Amazon Redshift。客户端必须支持以下内容：

- 传输层安全性协议（TLS）。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密（PFS）的密码套件，例如 DHE（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

# 网络隔离

基于 Amazon VPC 服务的 Virtual Private Cloud (VPC) 是您在AWS云中的逻辑上隔离的私有网络。可以通过执行以下步骤，在 VPC 中部署 Amazon Redshift 集群或 Redshift Serverless 组：

- 在 AWS 区域中创建 VPC。有关更多信息，请参阅 Amazon VPC 用户指南中的什么是 Amazon VPC？。
- 创建两个或更多私有 VPC 子网。有关更多信息，请参阅 Amazon VPC 用户指南中的 VPC 和子网。
- 部署 Amazon Redshift 集群或 Redshift Serverless 工作组。有关更多信息，请参阅 Redshift 资源的子网 或 工作组和命名空间。

预设情况下，Amazon Redshift 集群在预置时被锁定。要允许来自 Amazon Redshift 客户端的入站网络流量，请将 VPC 安全组与 Amazon Redshift 集群相关联。有关更多信息，请参阅 Redshift 资源的子网。

要仅允许来往于特定 IP 地址范围的流量，请使用 VPC 更新安全组。例如，仅允许来自或流入公司网络的流量。

在配置与您的 Amazon Redshift 集群标记的子网关联的网络访问控制列表时，请确保将相应的 AWS 区域的 S3 CIDR 范围添加到入口和出口规则的允许列表中。这样做可以让您在没有任何中断的情况下执行基于 S3 的操作，如 Redshift Spectrum、COPY 和 UNLOAD。

以下示例命令解析了 us-east-1 区域内的 Amazon S3 中使用的所有 IPv4 地址的 JSON 响应。

```
curl https://ip-ranges.amazonaws.com/ip-ranges.json | jq -r '.prefixes[] |
 select(.region=="us-east-1") | select(.service=="S3") | .ip_prefix'

54.231.0.0/17

52.92.16.0/20

52.216.0.0/15
```

有关如何获取特定区域的 S3 IP 范围的说明，请参阅 AWS IP 地址范围。

Amazon Redshift 支持将集群部署到专用租赁 VPC 中。有关更多信息，请参阅 Amazon EC2 用户指南中的专用实例。

# Amazon Redshift 安全组

在您预置 Amazon Redshift 集群时，它默认处于锁定状态，因此任何人都无法访问。要为其他用户授予针对 Amazon Redshift 集群的入站访问权限，您可以将该集群与安全组关联起来。如果您处于 EC2-VPC 平台上，您可以使用现有的 Amazon VPC 安全组或者定义一个新的安全组，然后将其与集群关联。有关在 EC2-VPC 平台上管理集群的更多信息，请参阅VPC 中的 Redshift 资源。

## 接口 VPC 端点

可以使用虚拟私有云（VPC）中的接口 VPC 端点（AWS PrivateLink）直接连接到 Amazon Redshift 和 Amazon Redshift Serverless API 服务，而不是通过互联网进行连接。有关 Amazon Redshift API 操作的信息，请参阅 Amazon Redshift API 参考中的操作。有关 Redshift Serverless API 操作的信息，请参阅《Amazon Redshift Serverless API 参考》中的 Actions。有关 AWS PrivateLink 的更多信息，请参阅 Amazon VPC 用户指南中的接口 VPC 终端节点 (AWS PrivateLink)。请注意，与集群或工作区的 JDBC/ODBC 连接不是 Amazon Redshift API 服务的一部分。

使用接口 VPC 端点时，VPC 和 Amazon Redshift 或 Redshift Serverless 之间的通信完全在 AWS 网络内进行，从而可以提供更好的安全性。每个 VPC 终端节点都由您的 VPC 子网中一个或多个使用私有 IP 地址的弹性网络接口代表。有关弹性网络接口的更多信息，请参阅《Amazon EC2 用户指南》中的弹性网络接口。

一个接口 VPC 终端节点将您的 VPC 直接连接到 Amazon Redshift。它不使用互联网网关、网络地址转换 (NAT) 设备、虚拟专用网络 (VPN) 连接或 Direct Connect 连接。VPC 中的实例不需要公有 IP 地址便可与 Amazon Redshift API 进行通信。

要在 VPC 中使用 Amazon Redshift 或 Redshift Serverless，您有两种选择。一种是从 VPC 内的实例进行连接。另一种方法是将您的私有网络连接到您的 VPC，方法是使用 Site-to-Site VPN 选项或 Direct Connect。有关 Site-to-Site VPN 选项的更多信息，请参阅 Amazon VPC 用户指南中的 VPN 连接。有关 Direct Connect 的信息,请参阅 Direct Connect 用户指南中的创建连接。

您可以创建接口 VPC 终端节点以使用 AWS 管理控制台 或 AWS Command Line Interface (AWS CLI) 命令连接到 Amazon Redshift。有关更多信息，请参阅创建接口端点。

在创建接口 VPC 终端节点后，您可以为端点启用私有 DNS 主机名。当您启用时，默认端点如下所示：

- Amazon Redshift 预置：`https://redshift.Region.amazonaws.com`
- Amazon Redshift Serverless：`https://redshift-serverless.Region.amazonaws.com`

如果您不启用私有 DNS 主机名，Amazon VPC 将提供一个您可以使用的 DNS 端点名称，格式如下。

- Amazon Redshift 预置：*VPC_endpoint_ID*.redshift.*Region*.vpce.amazonaws.com
- Amazon Redshift Serverless：*VPC_endpoint_ID*.redshift-serverless.*Region*.vpce.amazonaws.com

有关更多信息，请参阅《Amazon VPC 用户指南》中的接口 VPC 端点 (AWS PrivateLink)。

Amazon Redshift 和 Redshift Serverless 支持在 VPC 内调用所有 Amazon Redshift API operations 和 Redshift Serverless API operations。

您可以将 VPC 终端节点策略附加到 VPC 终端节点，以控制 AWS Identity and Access Management (IAM) 委托人的访问权限。您还可以将安全组与 VPC 终端节点关联，以便根据网络流量的源和目标控制入站和出站访问。示例为 IP 地址的范围。有关更多信息，请参阅《Amazon VPC 用户指南》中的使用 VPC 端点控制对服务的访问权限。

## Amazon Redshift 的 VPC 端点策略

您可以为 Amazon Redshift 的 VPC 终端节点创建一个策略，在该策略中指定以下内容：

- 可以或不能执行操作的委托人
- 可执行的操作
- 可对其执行操作的资源

有关更多信息，请参阅《Amazon VPC 用户指南》中的使用 VPC 端点控制对服务的访问权限。

接下来，您可以查找 VPC 终端节点策略示例。

### Amazon Redshift 预置端点策略示例

接下来，可以查找适用于 Amazon Redshift 预置的 VPC 端点策略示例。

#### 示例：用于拒绝来自指定 AWS 账户的所有访问的 VPC 终端节点策略

以下 VPC 终端节点策略会拒绝 AWS 账户 *123456789012* 使用此端点访问资源的所有权限。

```
{
```

```
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

示例：仅向指定的 IAM 角色授予 VPC 访问权限的 VPC 端点策略

以下 VPC 端点策略仅向 AWS 账户 *123456789012* 中的 IAM 角色 *redshiftrole* 授予完全访问权限。使用终端节点拒绝所有其它 IAM 委托人进行访问。

```
   {
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:role/redshiftrole"
                ]
            }
        }]
}
```

这只是一个示例。在大多数应用场景中，我们建议附加特定操作的权限，以缩小权限范围。

示例：仅允许 VPC 访问指定的 IAM 委托人（用户）的 VPC 终端节点策略

以下 VPC 终端节点策略仅允许对 AWS 账户 *123456789012* 中的 IAM 用户 *redshiftadmin* 进行完全访问。使用端点拒绝所有其他 IAM 委托人进行访问。

```
   {
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:user/redshiftadmin"
                ]
            }
        }]
}
```

这只是一个示例。在大多数应用场景中，我们建议在将权限分配给用户之前为角色附加权限。此外，我们建议使用特定操作来缩小权限范围。

示例：允许只读 Amazon Redshift 操作的 VPC 终端节点策略

以下 VPC 终端节点策略仅允许 AWS 账户 *123456789012* 执行指定的 Amazon Redshift 操作。

指定的操作为 Amazon Redshift 提供等效的只读访问权限。针对指定账户拒绝 VPC 上的所有其它操作。同样，所有其他帐户都被拒绝进行任何访问。有关 Amazon Redshift 操作的列表，请参阅 IAM 用户指南中的 Amazon Redshift 的操作、资源和条件键。

```
  {
    "Statement": [
        {
            "Action": [
                "redshift:DescribeAccountAttributes",
                "redshift:DescribeClusterParameterGroups",
                "redshift:DescribeClusterParameters",
                "redshift:DescribeClusterSecurityGroups",
                "redshift:DescribeClusterSnapshots",
                "redshift:DescribeClusterSubnetGroups",
                "redshift:DescribeClusterVersions",
```

```
                "redshift:DescribeDefaultClusterParameters",
                "redshift:DescribeEventCategories",
                "redshift:DescribeEventSubscriptions",
                "redshift:DescribeHsmClientCertificates",
                "redshift:DescribeHsmConfigurations",
                "redshift:DescribeLoggingStatus",
                "redshift:DescribeOrderableClusterOptions",
                "redshift:DescribeQuery",
                "redshift:DescribeReservedNodeOfferings",
                "redshift:DescribeReservedNodes",
                "redshift:DescribeResize",
                "redshift:DescribeSavedQueries",
                "redshift:DescribeScheduledActions",
                "redshift:DescribeSnapshotCopyGrants",
                "redshift:DescribeSnapshotSchedules",
                "redshift:DescribeStorage",
                "redshift:DescribeTable",
                "redshift:DescribeTableRestoreStatus",
                "redshift:DescribeTags",
                "redshift:FetchResults",
                "redshift:GetReservedNodeExchangeOfferings"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

示例：拒绝访问指定集群的 VPC 终端节点策略

以下 VPC 终端节点策略允许所有账户和委托人的完全访问权限。与此同时，它拒绝 AWS 账户 *123456789012* 对于在具有集群 ID *my-redshift-cluster* 的 Amazon Redshift 集群上执行的操作的任何访问。仍然允许其他不支持集群资源级权限的 Amazon Redshift 操作。有关 Amazon Redshift 操作及其相应资源类型的列表，请参阅 IAM 用户指南中的 Amazon Redshift 的操作、资源和条件键。

```
 {
```

```
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "arn:aws:redshift:us-east-1:123456789012:cluster:my-redshift-
cluster",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

Amazon Redshift Serverless 端点策略示例

接下来，可以查找适用于 Redshift Serverless 的 VPC 端点策略示例。

示例：支持只读 Redshift Serverless 操作的 VPC 端点策略

以下 VPC 端点策略仅支持 AWS 账户 *123456789012* 执行指定的 Redshift Serverless 操作。

指定的操作为 Redshift Serverless 提供等效的只读访问权限。针对指定账户拒绝 VPC 上的所有其它操作。同样，所有其他帐户都被拒绝进行任何访问。有关 Redshift Serverless 操作的列表，请参阅《IAM 用户指南》中的 Actions, Resources, and Condition Keys for Redshift Serverless。

```
{
  "Statement": [
      {
          "Action": [
              "redshift-serverless:DescribeOneTimeCredit",
              "redshift-serverless:GetCustomDomainAssociation",
              "redshift-serverless:GetEndpointAccess",
```

```
                "redshift-serverless:GetNamespace",
                "redshift-serverless:GetRecoveryPoint",
                "redshift-serverless:GetResourcePolicy",
                "redshift-serverless:GetScheduledAction",
                "redshift-serverless:GetSnapshot",
                "redshift-serverless:GetTableRestoreStatus",
                "redshift-serverless:GetUsageLimit",
                "redshift-serverless:GetWorkgroup"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

示例：拒绝访问指定工作组的 VPC 端点策略

以下 VPC 终端节点策略允许所有账户和委托人的完全访问权限。同时，它拒绝 AWS 账户 *123456789012* 对于在工作组 ID 为 *my-redshift-workgroup* 的 Amazon Redshift 工作组上执行的操作进行任何访问。仍然可以执行其它不支持工作组资源级权限的 Amazon Redshift 操作。有关 Redshift Serverless 操作及其相应资源类型的列表，请参阅《IAM 用户指南》中的 Actions, Resources, and Condition Keys for Redshift Serverless。

```
 {
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "arn:aws:redshift-serverless:us-
east-1:123456789012:workgroup:my-redshift-workgroup",
```

```
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

# Amazon Redshift 中的配置和漏洞分析

AWS 负责处理来宾操作系统 (OS) 和数据库补丁、防火墙配置和灾难恢复 (DR) 等基本安全任务。这些流程已经过认证的第三方审计。有关更多信息，请参阅Amazon Redshift 的合规性验证、责任共担模式和安全性、身份和合规性的最佳实践。

Amazon Redshift 会自动应用升级并修补您的数据仓库，以便您可以专注于您的应用程序，而不是应用程序的管理。补丁和升级在可配置的维护时段中应用。有关更多信息，请参阅 维护时段。

Amazon Redshift 查询编辑器 v2 是 AWS 托管式应用程序。所有补丁和更新均由 AWS 根据需要应用。

# 使用身份增强 IAM 角色会话连接到 Redshift

您可以使用 IAM Identity Center 提供对 Amazon Redshift 集群和无服务器工作组的联合访问权限。这种方法可让用户使用其 Identity Center 凭证进行身份验证。

Amazon Redshift 提供 `GetIdentityCenterAuthToken` API 操作来生成包含用户身份信息的授权令牌。这些 API 同时可用于预置集群和无服务器工作组。通过令牌，可以使用现有的 Identity Center 设置对 Amazon Redshift 数据库进行无缝单点登录访问。

## 先决条件

在将 Identity Center 身份验证与 Amazon Redshift 结合使用之前，请确保您具备以下各项：

- Identity Center 设置：您的账户必须使用用户身份和相应的应用程序分配配置了 IAM Identity Center。有关设置说明，请参阅设置 IAM Identity Center。

  > **Important**
  >
  > 如果要连接到 Redshift，则必须使用 redshift:connect 范围。

- 身份增强凭证：您的应用程序必须使用包含嵌入式用户身份信息的身份增强凭证。有关更多信息，请参阅使用身份增强 IAM 角色会话。
- IAM 权限：您的 IAM 角色或用户必须具有调用 `GetIdentityCenterAuthToken` API 和访问指定集群或工作组的权限。所需权限：
  - 对于预置集群：对集群 ARN（格式：`arn:aws:redshift:region:account:cluster:cluster-name`）具有 `redshift:GetIdentityCenterAuthToken`
  - 对于无服务器工作组：对工作组 ARN（格式：`arn:aws:redshift-serverless:region:account:workgroup/workgroup-name`）具有 `redshift-serverless:GetIdentityCenterAuthToken`
- 兼容的驱动程序：使用支持 Identity Center 授权令牌的 Amazon Redshift JDBC 或 ODBC 驱动程序：
  - JDBC 驱动程序：请参阅安装和配置 Amazon Redshift JDBC 驱动程序版本 2.x
  - ODBC 驱动程序：请参阅安装和配置 Amazon Redshift ODBC 驱动程序版本 2.0
- 增强型 VPC 路由的网络配置：如果开启了增强型 VPC 路由，则您的 VPC 必须能够访问 Identity Center 服务。接口 VPC 端点是提供这种连接的推荐方式。请在用户登录之前设置此连接。有关

Identity Center 的增强型 VPC 路由要求的更多信息，请参阅[将 AWS IAM Identity Center 身份验证与增强型 VPC 路由结合使用]。

# Identity Center 身份验证的工作原理

Amazon Redshift 的 Identity Center 身份验证使用以下工作流程：

1. 您的应用程序使用包含嵌入式用户身份信息的身份增强凭证调用 `GetIdentityCenterAuthToken` API。
2. Amazon Redshift 验证 Identity Center 身份，并生成范围限定于特定集群或工作组的加密授权令牌。请参阅示例 IAM 策略。
3. 您的应用程序使用此令牌连接到指定的 Amazon Redshift 集群或工作组。
4. Amazon Redshift 数据面板验证令牌，并根据 Identity Center 用户在 Identity Center 应用程序中的权限来授予访问权限。

> **Important**
>
> 此 API 需要身份增强凭证。有关更多信息，请参阅[使用身份增强 IAM 角色会话]。
> 如果您在没有身份增强凭证的情况下调用 API，则会收到 `UnsupportedOperationFault` 错误。

# GetIdentityCenterAuthToken API 操作

Amazon Redshift 提供了两个单独的 `GetIdentityCenterAuthToken` API 操作：一个用于预置集群，另一个用于无服务器工作组。这两个操作具有相同的名称，但接受的参数因目标资源类型而异。

## 适用于预置集群的 GetIdentityCenterAuthToken

对于预置的 Amazon Redshift 集群，请在 Amazon Redshift 服务中使用 `GetIdentityCenterAuthToken` API 来生成授权令牌。

### 请求语法

```
{
```

```
   "ClusterIds": [ "string" ]
}
```

### 请求参数

ClusterIds

令牌将有权访问的 Amazon Redshift 集群标识符的列表。该令牌只能用于对在此列表中指定的集群进行身份验证。

类型：字符串数组

长度约束：最少 1 个项目。最多 20 个项目。

是否必需：是

### CLI 示例

示例：获取单个集群的授权令牌

```
aws redshift get-identity-center-auth-token \
    --cluster-ids my-redshift-cluster
```

示例：获取多个集群的授权令牌

```
aws redshift get-identity-center-auth-token \
    --cluster-ids my-cluster-1 my-cluster-2
```

## 适用于无服务器工作组的 GetIdentityCenterAuthToken

对于 Amazon Redshift 无服务器工作组，请在 Amazon Redshift Serverless 服务中使用 `GetIdentityCenterAuthToken` API 来生成授权令牌。

### 请求语法

```
{
```

```
   "WorkgroupNames": [ "string" ]
}
```

## 请求参数

WorkgroupNames

令牌将有权访问的 Amazon Redshift Serverless 工作组名称的列表。该令牌只能用于对在此列表中指定的工作组进行身份验证。

类型：字符串数组

长度约束：最少 1 个项目。最多 20 个项目。

是否必需：是

## CLI 示例

示例：获取单个工作组的授权令牌

```
aws redshift-serverless get-identity-center-auth-token \
    --workgroup-names my-workgroup
```

示例：获取多个工作组的授权令牌

```
aws redshift-serverless get-identity-center-auth-token \
    --workgroup-names workgroup-1 workgroup-2
```

# 响应语法

这两个 API 都返回相同的响应结构：

```
{
   "AuthorizedToken": "string",
   "ExpirationTime": "timestamp"
}
```

### 响应参数

AuthorizedToken

一个加密的授权令牌，其中包含用户身份信息以及已授权集群或工作组的列表。此令牌应被视为敏感数据。

类型：字符串

ExpirationTime

令牌到期的日期和时间（UTC 时间）。令牌自生成之时起 1 小时内有效。

类型：时间戳

### 响应示例

```
{
    "AuthorizedToken":
 "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ1c2VyQGV4YW1wbGUuY29tIiwiaWF0IjoxNjQwOTk1MjAw
    "ExpirationTime": "2024-01-01T12:00:00Z"
}
```

## 驱动程序集成

Amazon Redshift 驱动程序通过直接使用令牌来支持 Identity Center 身份验证：

## 直接使用令牌

调用 `GetIdentityCenterAuthToken` API 以获取令牌后，请将 `IdpTokenAuthPlugin` 与 `SUBJECT_TOKEN` 令牌类型结合使用。

连接配置：

```
plugin_name = com.amazon.redshift.plugin.IdpTokenAuthPlugin
token_type = SUBJECT_TOKEN
token = {encrypted_token_from_api_response}
```

有关 Identity Center 身份验证插件和驱动程序配置的详细信息，请参阅连接到 Amazon Redshift 集群。

### Java 代码示例

使用 Identity Center 身份验证进行连接的 Java 代码示例：

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

// Get token from GetIdentityCenterAuthToken API
String token = "your_encrypted_token_from_api_response";

// Configure connection properties
Properties props = new Properties();
props.setProperty("user", "your_username");
props.setProperty("plugin_name", "com.amazon.redshift.plugin.IdpTokenAuthPlugin");
props.setProperty("token_type", "SUBJECT_TOKEN");
props.setProperty("token", token);

// Connect to Redshift
String url = "jdbc:redshift://your-cluster.region.redshift.amazonaws.com:5439/
your_database";
try (Connection conn = DriverManager.getConnection(url, props)) {
    // Use connection
    System.out.println("Connected successfully!");
} catch (SQLException e) {
    e.printStackTrace();
}
```

## IAM 策略要求

要将 Identity Center 身份验证与 Amazon Redshift 结合使用，除了连接到 Amazon Redshift 集群和工作组所需的标准权限外，还需要特定 IAM 权限。

### API 权限

对于预置集群，增强的 IAM 角色会话必须：

- 对集群 ARN（格式：arn:aws:redshift:region:account:cluster:cluster-name）具有 redshift:GetIdentityCenterAuthToken

对于无服务器工作组，增强的 IAM 角色会话必须：

- 对工作组 ARN（格式：arn:aws:redshift-serverless:region:account:workgroup/workgroup-name）具有 redshift-serverless:GetIdentityCenterAuthToken

## 示例 IAM policies

预置集群的策略示例：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "redshift:GetIdentityCenterAuthToken"
            ],
            "Resource": [
                "arn:aws:redshift:us-east-1:123456789012:cluster:my-cluster"
            ]
        }
    ]
}
```

无服务器工作组的策略示例：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "redshift-serverless:GetIdentityCenterAuthToken"
            ],
            "Resource": [
                "arn:aws:redshift-serverless:us-east-1:123456789012:workgroup/my-
workgroup"
            ]
```

```
        }
    ]
}
```

多个资源的策略示例：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "redshift:GetIdentityCenterAuthToken"
            ],
            "Resource": [
                "arn:aws:redshift:*:123456789012:cluster/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "redshift-serverless:GetIdentityCenterAuthToken"
            ],
            "Resource": [
                "arn:aws:redshift-serverless:*:123456789012:workgroup/*"
            ]
        }
    ]
}
```

# 区域可用性

Identity Center 身份验证在以下 AWS 区域中可用：

- 商业区域：所有支持的 Amazon Redshift 区域
- AWS GovCloud：在 us-gov-east-1 和 us-gov-west-1 中可用
- 中国区域：在 cn-north-1 和 cn-northwest-1 中可用

> **Note**
>
> 在初始推出期间，功能可用性可能会有所不同。

# Identity Center 身份验证问题排查

使用以下部分，协助排查您在 Identity Center 身份验证方面遇到的问题。

主题


- API 错误
- 连接错误


## API 错误

`GetIdentityCenterAuthToken` API 可能返回以下错误：

- UnsupportedOperationFault：请求是在没有身份增强凭证的情况下发出的，或者当前配置不支持 Identity Center 身份传播。
- ClusterNotFoundFault (redshift:GetIdentityCenterAuthToken)：一个或多个指定的集群标识符不存在，或者您无权访问它们。
- WorkgroupNotFoundFault (redshift-serverless:GetIdentityCenterAuthToken)：一个或多个指定的工作组名称不存在，或者您无权访问它们。
- InvalidParameterValueException：请求参数无效，例如集群或工作组列表为空，或者列表中的项目超过 20 个。

## 连接错误

使用 Identity Center 令牌连接到 Amazon Redshift 时，您可能会遇到以下错误：

- Token credentials expired or invalid：使用 `GetIdentityCenterAuthToken` API 生成新令牌，并更新您的连接配置。
- invalid token：验证令牌格式，并确保它是为正确的集群或工作组生成的。
- TYPE_INVALID_TOKEN：确保在连接参数中使用 `token_type = SUBJECT_TOKEN`。
- identity provider type "awsidc" is not supported：请验证您的 Amazon Redshift 集群或工作组是否支持 Identity Center 身份验证，并更新到兼容版本。

- AWS IdC identity provider does not exist：通过 AWS 控制台为您的 Amazon Redshift 资源配置 Identity Center 集成。
- Invalid scope. User's credentials are not authorized to connect to Amazon Redshift：验证用户是否在 Identity Center 中具有相应的权限，并且令牌是为正确的资源生成的。
- 增强型 VPC 路由的连接失败：开启增强型 VPC 路由后，Amazon Redshift 通过您的 VPC 访问 Identity Center 服务。验证您的 VPC 是否具有适用于这些服务的接口 VPC 端点。确认已为每个端点开启私有 DNS 名称。确认端点安全组支持 TCP 端口 443 上的入站流量。有关 Identity Center 的增强型 VPC 路由要求的更多信息，请参阅将 AWS IAM Identity Center 身份验证与增强型 VPC 路由结合使用。

# 联网任务

您可以执行联网任务，例如自定义与 Redshift 数据库的连接。这样可以控制流量，以便确保安全性或满足其他目的。您还可以执行与 DNS 相关的任务，例如为 Redshift 资源设置自定义域名。如果您拥有 Amazon Redshift 预置集群或 Amazon Redshift Serverless 工作组，则可以执行这些配置任务。

主题


- 用于客户端连接的自定义域名
- Redshift 托管的 VPC 端点
- VPC 中的 Redshift 资源
- 使用 Redshift 增强型 VPC 路由控制网络流量


# 用于客户端连接的自定义域名

您可以为 Amazon Redshift 集群和 Amazon Redshift Serverless 工作组创建自定义域名，也称为自定义 URL。这是易于阅读的 DNS 记录，可将 SQL 客户端连接路由到您的端点。您可以随时为现有集群或工作组进行配置。自定义域名有几个好处：

- 自定义域名是一个比默认 URL 更简单的字符串，后者通常包括集群名称或工作组名称和区域。它更易于记起和使用。
- 例如，在进行故障转移时，您可以将流量快速路由到新集群或工作组。这使得客户端在重新连接时不必更改配置。连接可以集中重新路由，将中断降至最低。
- 您可以避免在连接 URL 中分享私有信息，例如服务器名称。您可以将其隐藏在自定义 URL 中。

使用 CNAME 设置自定义域名时，Amazon Redshift 不会收取任何额外费用。如果您创建一个新域名，您的 DNS 提供商可能会向您收取域名的费用，但这笔费用通常很小。

## 注册域名

设置自定义域名由多项任务组成：包括向 DNS 提供商注册域名和创建证书。执行这些工作后，您可以在 Amazon Redshift 控制台中或 Amazon Redshift Serverless 控制台中配置自定义域名，或者使用 AWS CLI 命令对其进行配置。

您必须有一个已注册的互联网域名，才能在 Amazon Redshift 中配置自定义域名。您可以使用 Route 53 或第三方域注册提供商来注册互联网域。您在 Amazon Redshift 控制台之外完成这些任务。要想完成创建自定义域的剩余过程，有一个已注册的域是先决条件。

> **Note**
>
> 如果您使用预置集群，在执行步骤以配置自定义域名之前，必须启用重新定位。有关更多信息，请参阅 重新定位集群。Amazon Redshift Serverless 不需要此步骤。

自定义域名通常包括根域和子域，例如 `mycluster.example.com`。要对其进行配置，请执行以下步骤：

为您的自定义域名创建 DNS CNAME 条目

1. 注册一个根域，例如，`example.com`。您可以选择使用现有域。您的自定义名称可能会受到对特定字符的限制或其他命名验证的限制。有关向 Route 53 注册域的更多信息，请参阅 Registering a new domain。
2. 添加一个 DNS CNAME 记录，将您的自定义域名指向集群或工作组的 Redshift 端点。您可以在 Redshift 控制台或 Amazon Redshift Serverless 控制台中，在集群或工作组的属性中查找端点。在一般信息下，复制集群或工作组属性中提供的 JDBC URL。URL 类似于以下内容：

   - 对于Amazon Redshift 集群：`redshift-cluster-sample.abc123456.us-east-1.redshift.amazonaws.com`
   - 对于 Amazon Redshift Serverless 工作组：`endpoint-name.012345678901.us-east-1-dev.redshift-serverless-dev.amazonaws.com`

   如果 URL 有 JDBC 前缀，请将其删除。

   > **Note**
   >
   > DNS 记录视可用性而定，因为每个名称都必须是唯一的，并且可以在您的组织内使用。

限制

在为自定义域创建 CNAME 记录时，需要注意几个限制：

- 对于同一个预置集群或 Amazon Redshift Serverless 工作组，不支持创建多个自定义域名。您只能关联一个 CNAME 记录。
- 不支持将 CNAME 记录与多个集群或工作组关联。每个 Redshift 资源的 CNAME 必须唯一。

注册域并创建 CNAME 记录后，选择新的或现有的证书。您可以使用 AWS Certificate Manager 执行此步骤：

我们建议您创建符合托管续订资格的 DNS 经验证证书，这些证书在 AWS Certificate Manager 中提供。托管续订意味着 ACM 将自动续订您的证书，或者在接近过期时向您发送电子邮件通知。有关托管证书续订的更多信息，请参阅ACM 证书的托管续订。

## 为域名请求证书。

Amazon Redshift 或 Amazon Redshift Serverless 要求自定义端点具有经验证的安全套接字层 (SSL) 证书，以保持通信安全和验证域名的所有权。您可以将您的 AWS Certificate Manager 账户与 AWS KMS key 结合使用，以进行安全的证书管理。安全验证包括完整主机名验证（sslmode=verify-full）。

只有在您选择 DNS 验证而不是电子邮件验证时，证书续订才会由 Amazon Redshift 托管。如果您使用电子邮件验证，则可以使用证书，但您必须在证书到期之前自行进行续订。我们建议您为证书选择 DNS 验证。您可以在 AWS Certificate Manager 中监控导入证书的到期日期。

为域名请求来自 ACM 的证书

1. 登录到 AWS 管理控制台 并通过 https://console.aws.amazon.com/acm/ 打开 ACM 控制台。
2. 选择请求证书。
3. 在域名字段中，输入您的自定义域名。

   > **Note**
   >
   > 除了证书域之外，您还可以指定许多前缀，以便将单个证书用于多个自定义域记录。举例来说，您可以将其他记录（例如 `one.example.com`、`two.example.com`）或通配符 DNS 记录（例如 `*.example.com`）与同一个证书一起使用。

4. 选择审核并请求。
5. 选择确认并请求。
6. 如果请求有效，Internet 域中注册的拥有者必须同意请求，然后 ACM 才能颁发证书。完成这些步骤后，请确保在 ACM 控制台中，状态显示为已颁发。

# 配置自定义域

您可以使用 Amazon Redshift 或 Amazon Redshift Serverless 控制台创建自定义域 URL。如果您尚未进行配置，则自定义域名属性会在一般信息下显示为短划线（–）。创建 CNAME 记录和证书后，您可以关联集群或工作组的自定义域名。

创建自定义域关联需要以下 IAM 权限：

- `redshift:CreateCustomDomainAssociation` – 您可以通过添加特定集群的 ARN 来限制对该集群的权限。
- `redshiftServerless:CreateCustomDomainAssociation` – 您可以通过添加特定工作组的 ARN 来限制对该工作组的权限。
- `acm:DescribeCertificate`

作为最佳实践，我们建议将权限策略附加到 IAM 角色，然后根据需要将其分配给用户和组。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

您可以通过执行以下步骤来分配自定义域名。

1. 在 Redshift 控制台中选择集群，或者在 Amazon Redshift Serverless 控制台中选择工作组，然后在操作菜单下选择创建自定义域名。此时将显示对话框。
2. 输入自定义域名。
3. 为 ACM 证书选择来自 AWS Certificate Manager 的 ARN。确认您所做的更改。根据您创建证书的步骤中的指导，我们建议您通过 AWS Certificate Manager，选择符合托管续订资格的 DNS 经验证证书 。
4. 在集群属性中，验证自定义域名和自定义域证书 ARN 中填充了您的条目，并列出了自定义域证书的到期日期。

配置自定义域后，仅可为新的自定义域使用 `sslmode=verify-full`。它不适用于默认端点。但您仍然可以使用其他 ssl 模式（例如 `sslmode=verify-ca`）连接到默认端点。

> **Note**
>
> 提醒一下，集群重新定位不是配置其它 Redshift 联网特征的先决条件。您无需将其开启即可启用以下功能：

- 从跨账户或跨区域 VPC 连接到 Redshift – 您可以从一个 AWS Virtual Private Cloud (VPC) 连接到另一个包含 Redshift 数据库的虚拟私有云 (VPC)。这简化了管理，例如，对于来自不同账户或 VPC 的客户端访问，无需对连接到数据库的身份提供本地 VPC 访问权限。有关更多信息，请参阅从其他账户或区域中的 Redshift VPC 端点连接到 Amazon Redshift Serverless。
- 设置自定义域名 – 您可以创建自定义域名（如本主题中所述），以使端点名称更加相关和简单。

## 连接到 Amazon Redshift 预置集群或 Amazon Redshift Serverless 工作组

要使用自定义域名进行连接，预置集群需要以下 IAM 权限：`redshift:DescribeCustomDomainAssociations`。对于 Amazon Redshift Serverless，您不必添加权限。

作为最佳实践，我们建议将权限策略附加到 IAM 角色，然后根据需要将其分配给用户和组。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

在控制台中完成创建 CNAME 并将其分配给集群或工作组的步骤后，您可以在 SQL 客户端的连接属性中提供自定义 URL。请注意，在创建 CNAME 记录后，DNS 传播可能会立即导致延迟。

1. 打开 SQL 客户端。例如，您可以使用 SQL/Workbench J。打开连接的属性，然后为连接字符串添加自定义域名。例如 `jdbc:redshift://mycluster.example.com:5439/dev?sslmode=verify-full`。在此示例中，`dev` 指定默认数据库。
2. 为您的数据库用户添加用户名和密码。
3. 测试连接。根据授予数据库用户的权限或授予所分配的 Amazon Redshift 数据库角色的权限，您查询特定表等数据库资源的能力可能会有所不同。

   请注意，如果您的集群或工作组位于 VPC 中，则可能需要将集群或工作组设置为可公开访问才能连接到其上。您可以在网络属性中更改此设置。

> **Note**
>
> JDBC、ODBC 和 Python 驱动程序支持与自定义域名的连接。

# 重命名分配了自定义域的集群

> **Note**
>
> 这一系列步骤不适用于 Amazon Redshift Serverless 工作组。您无法更改工作组名称。

要重命名具有自定义域名的集群，需要 `acm:DescribeCertificate` IAM 权限。

1. 请转至 Amazon Redshift 控制台并选择要更改其名称的集群。选择编辑，编辑集群属性。
2. 编辑集群标识符。您也可以更改集群的其他属性。然后选择 Save changes（保存更改)。
3. 重命名集群后，你必须更新 DNS 记录，将自定义域的 CNAME 条目更改为指向更新后的 Amazon Redshift 端点。

## 描述自定义域关联

使用此部分中的命令，获取与特定预置集群或 Amazon Redshift Serverless 工作组关联的自定义域名列表。

您需要以下权限：

- 对于预置集群：`redshift:DescribeCustomDomainAssociations`
- 对于 Amazon Redshift Serverless 工作组：`redshiftServerless:ListCnameAssociations`

作为最佳实践，我们建议将权限策略附加到 IAM 角色，然后根据需要将其分配给用户和组。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

以下显示的示例命令用于列出给定 Amazon Redshift 集群的自定义域名：

```
aws redshift describe-custom-domain-associations --custom-domain-name customdomainname
```

启用自定义域名后，您可以运行此命令来确定与集群关联的自定义域名。有关用于描述自定义域关联的 CLI 命令的更多信息，请参阅 describe-custom-domain-associations。

与此类似，以下显示的示例命令用于列出给定 Amazon Redshift Serverless 工作组的自定义域名：有一些不同的方法可以做到这一点。您可以只提供自定义域名：

```
aws redshift-serverless list-custom-domain-associations --custom-domain-
name customdomainname
```

您也可以通过只提供证书 ARN 来获取关联：

```
aws redshift-serverless list-custom-domain-associations --custom-domain-certificate-
arn certificatearn
```

启用自定义域名后，您可以运行这些命令来确定与工作组关联的自定义域名。您也可以运行命令来获取自定义域关联的属性。为此，您必须将自定义域名和工作组名称作为参数提供。该命令返回证书 ARN、工作组名称和自定义域的证书到期时间：

```
aws redshift-serverless get-custom-domain-association --workgroup-name workgroupname --
custom-domain-name customdomainname
```

有关可用于 Amazon Redshift Serverless 的 CLI 参考命令更多信息，请参阅 redshift-serverless。

## 将自定义域与不同的证书关联

更改自定义域名的证书关联需要以下 IAM 权限：

- `redshift:ModifyCustomDomainAssociation`
- `acm:DescribeCertificate`

作为最佳实践，我们建议将权限策略附加到 IAM 角色，然后根据需要将其分配给用户和组。有关更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

使用以下命令将自定义域关联到不同的证书。`--custom-domain-name` 和 `custom-domain-certificate-arn` 参数为必需。新证书的 ARN 必须与现有 ARN 不同。

```
aws redshift modify-custom-domain-association --cluster-id redshiftcluster --custom-
domain-name customdomainname --custom-domain-certificate-arn certificatearn
```

以下示例显示了如何为 Amazon Redshift Serverless 将自定义域关联到不同的证书。

```
aws redshift-serverless modify-custom-domain-association --workgroup-
name redshiftworkgroup --custom-domain-name customdomainname --custom-domain-
certificate-arn certificatearn
```

最多会有 30 秒的延迟，然后您才能连接到集群。部分延迟发生在 Amazon Redshift 集群更新其属性时，而在 DNS 更新时也会造成一些额外的延迟。有关 API 和每个属性设置的更多信息，请参阅 modifyCustomDomainAssociation。

## 删除自定义域

要删除自定义域名，用户必须具有执行以下操作的权限。

- 对于预置集群：`redshift:DeleteCustomDomainAssociation`
- 对于 Amazon Redshift Serverless 工作组：`redshiftServerless:DeleteCustomDomainAssociation`

在控制台上

您可以选择操作按钮并选择删除自定义域名，从而删除自定义域名。完成此操作后，您仍然可以更新工具以使用控制台中列出的端点，从而连接到服务器。

使用 CLI 命令

以下示例显示了如何删除自定义域名。删除操作要求您提供集群的现有自定义域名。

```
aws redshift delete-custom-domain-association --cluster-id redshiftcluster --custom-domain-name customdomainname
```

以下示例显示了如何删除 Amazon Redshift Serverless 工作组的自定义域名。自定义域名是必填参数。

```
aws redshift-serverless delete-custom-domain-association --workgroup-name workgroupname --custom-domain-name customdomainname
```

有关更多信息，请参阅 DeleteCustomDomainAssociation。

# Redshift 托管的 VPC 端点

默认情况下，Amazon Redshift 集群或 Amazon Redshift Serverless 工作组在虚拟私有云（VPC）中进行预置。当您允许公共访问或设置互联网网关、NAT 设备或 AWS Direct Connect 连接将流量路由到集群时，可从另一个 VPC 或子网中访问该 VPC。您也可以通过设置 Redshift 托管的 VPC 端点（由 AWS PrivateLink 支持）来访问集群或工作组。

您可以将 Redshift 托管的 VPC 端点设置为包含集群或工作组的 VPC 与运行客户端工具的 VPC 之间的私有连接。如果集群或工作组位于另一个账户中，则账户拥有者（授予者）必须授予对连接账户（被授予者）的访问权限。通过这种方法，您可以访问数据仓库，而无需使用公有 IP 地址或通过互联网路由流量。

以下是允许使用 Redshift 托管的 VPC 端点进行访问的常见原因：

- AWS 账户 A 希望允许 AWS 账户 B 中的 VPC 对集群或工作组具有访问权限。
- AWS 账户 A 希望允许同样位于 AWS 账户 A 中的 VPC 对集群或工作组具有访问权限。
- AWS 账户 A 希望允许 AWS 账户 A 中的 VPC 中的另一个子网对集群或工作组具有访问权限。

设置 Redshift 托管的 VPC 端点以访问其他账户中的集群或工作组的工作流如下所示：

1. 拥有者账户向其他账户授予访问权限，并指定被授权者的 AWS 账户 ID 和 VPC 标识符（或所有 VPC）。
2. 被授权者账户收到通知，告知他们有权创建 Redshift 托管的 VPC 终端节点。
3. 被授权者账户创建一个 Redshift 托管的 VPC 终端节点。
4. 被授权者账户可以使用 Redshift 托管的 VPC 端点访问拥有者账户的集群或工作组。

可以使用 Amazon Redshift 控制台、AWS CLI 或 Amazon Redshift API 执行此操作。

## 使用 Redshift 托管的 VPC 终端节点时的注意事项

> **Note**
>
> 要创建或修改 Redshift 托管式 VPC 端点，除了在 AWS 托管式策略 `AmazonRedshiftFullAccess` 中指定的其它权限外，IAM 策略中还需要权限 `ec2:CreateVpcEndpoint` 或 `ec2:ModifyVpcEndpoint`。

使用 Redshift 托管的 VPC 终端节点时，请牢记以下内容：

- 如果您使用的是预置集群，集群必须为 RG 或 RA3 节点类型。Amazon Redshift Serverless 工作组也可用于设置 VPC 端点。
- 对于预调配集群，请确保对集群启用集群重新定位或多可用区。有关启用集群重新定位的要求的信息，请参阅重新定位集群。有关启用多可用区的信息，请参阅创建新集群时设置多可用区。

- 确保要通过安全组访问的集群或工作组在有效端口范围 5431-5455 和 8191-8215 内。默认值为 5439。
- 您可以修改与现有 Redshift 托管的 VPC 终端节点关联的 VPC 安全组。要修改其他设置，请删除当前 Redshift 托管 VPC 终端节点并创建一个新的端点。
- 您可以创建的 Redshift 托管 VPC 终端节点的数量限制为您的 VPC 终端节点配额。
- 无法从互联网访问 Redshift 托管的 VPC 终端节点。Redshift 托管的 VPC 端点只能在预置了端点的 VPC 中访问，或者在路由表和安全组允许的情况下与预置端点的 VPC 对等连接的任何 VPC 中访问。
- 您不能使用 Amazon VPC 控制台管理 Redshift 托管的 VPC 终端节点。
- 为预置集群创建 Redshift 托管的 VPC 端点时，您选择的 VPC 必须具有子网组。要创建子网组，请参阅创建集群子网组。
- 如果一个可用区关闭，Amazon Redshift 不会在另一个可用区中创建新的弹性网络接口。在这种情况下，您可能需要创建新端点。

有关配额和命名约束的信息，请参阅Amazon Redshift 资源中的配额和限制。

有关定价的信息，请参阅 AWS PrivateLink 定价。

## 授予对 VPC 的访问权限

如果您要访问集群或工作组的 VPC 位于另一个 AWS 账户中，请确保从拥有者（授予者）账户授权它。

允许另一个 AWS 账户中的 VPC 访问您的集群或工作组

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择集群。对于 Amazon Redshift Serverless，请选择无服务器控制面板。
3. 对于您要允许访问的集群，请通过选择集群名称查看详细信息。选择集群的属性选项卡。

   授予的账户部分显示有权访问集群的账户和相应的 VPC。对于 Amazon Redshift Serverless 工作组，请选择工作组。授权账户位于数据访问权限选项卡下。
4. 选择授予访问权限以显示要输入被授权者信息的表单以添加账户。
5. 对于 AWS 账户 ID，输入您授予访问权限的账户 ID。您可以授予对指定账户中特定 VPC 或所有 VPC 的访问权限。
6. 选择授予访问权限以授予访问权限。

## 创建 Redshift 托管的 VPC 终端节点

如果您拥有某个集群或工作组，或者您已被授予对集群或工作组的访问权限，则可以为该集群或工作组创建一个 Redshift 托管的 VPC 端点。

要创建 Redshift 托管的 VPC 终端节点

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择配置。

   配置页面显示已创建的 Redshift 托管 VPC 终端节点。要查看端点的详细信息，请选择其名称。对于 Amazon Redshift Serverless，当您选择工作组时，VPC 端点位于数据访问权限选项卡下。
3. 选择创建端点以显示一个表单，在其中输入有关要添加的端点的信息。
4. 输入端点名称、12 位 AWS 账户 ID、端点所在的虚拟私有云（VPC）、子网和 VPC 安全组的值。

   子网中的子网定义了 Amazon Redshift 部署端点所在的子网和 IP 地址。Amazon Redshift 选择具有可用于与端点关联的网络接口的 IP 地址的子网。

   VPC 安全组中的安全组规则定义了您为端点授权的入站流量的端口、协议和源。您允许通过工作负载运行所在的安全组或 CIDR 范围访问所选端口。
5. 选择创建端点以创建端点。

创建端点后，您可以通过 Redshift 托管 VPC 端点的配置设置的端点 URL 中所示的 URL 来访问集群或工作组。

# VPC 中的 Redshift 资源

您可以在基于 Amazon VPC 服务的 EC2-VPC 平台上，在 VPC 中启动 Amazon Redshift 集群或 Amazon Redshift Serverless 工作组。有关更多信息，请参阅 使用 EC2 创建集群。

> **Note**
>
> 不支持在专用租赁 VPC 中启动集群和 Serverless 工作组。有关更多信息，请参阅《Amazon VPC 用户指南》中的专用实例。

在 VPC 中预置资源时，您必须执行以下操作：

- 提供有关 VPC 的信息。

  在您的 VPC 中创建预置集群时，您必须通过创建集群子网组来提供 VPC 信息。此信息包括 VPC ID 和 VPC 中的子网列表，启动集群时，您需要提供子网组，以便 Redshift 能够在 VPC 的其中一个子网中预置您的集群。对于 Amazon Redshift Serverless，过程类似。您可以将子网直接分配给 Serverless 工作组。但是对于 Serverless，您无需创建子网组。有关在 Amazon Redshift 中创建子网组的更多信息，请参阅Redshift 资源的子网。有关设置 VPC 的更多信息，请参阅《Amazon VPC 入门指南》中的 Amazon VPC 入门。

- （可选）配置辅助功能选项。

  默认情况下，Amazon Redshift 中的预调配集群和无服务器工作组是私有的。如果您将预调配集群或无服务器工作组配置为可公开访问，Amazon Redshift 会为外部 IP 地址使用弹性 IP 地址。弹性 IP 地址是静态 IP 地址。借助弹性 IP 地址，您可以更改自己的基本配置，而不影响客户端用来连接的 IP 地址。在某些情况下（如在故障后进行恢复时），这种方法会很有用。是否创建弹性 IP 地址取决于您的可用区域重新定位设置。有两种方式：

  1. 如果您开启了可用区重新定位，并且想要启用公有访问，则不指定弹性 IP 地址。分配了一个由 Amazon Redshift 管理的弹性 IP 地址。它与您的 AWS 账户关联。
  2. 如果您已关闭可用区重新放置，且希望启用公开访问，则可以选择先在 Amazon EC2 中为 VPC 创建一个弹性 IP 地址，然后再启动 Amazon Redshift 集群或工作组。如果您未创建 IP 地址，Amazon Redshift 会提供已配置的弹性 IP 地址以用于 VPC。这个弹性 IP 地址由 Amazon Redshift 托管，并且不与您的 AWS 账户关联。

  有关更多信息，请参阅《Amazon EC2 用户指南》中的弹性 IP 地址。

  在某些情况下，您可能在 VPC 中有可公开访问的集群，并且希望使用该 VPC 内的私有 IP 地址连接到此集群。如果是这样的话，则必须将下列 VPC 参数设置为 `true`：

  - `DNS resolution`
  - `DNS hostnames`

  请注意，在 Amazon Redshift Serverless 中，您无法以这种方式进行连接。

  假设您在 VPC 中有可公开访问的预置集群，但没有在 VPC 中将这些参数设置为 `true`。在这些情况下，从 VPC 内部进行的连接将解析为资源的弹性 IP 地址，而不是私有 IP 地址。建议您将这些参数设置为 `true`，并使用私有 IP 地址从该 VPC 中连接到可公开访问的集群。有关更多信息，请参阅《Amazon VPC 用户指南》 中的在 VPC 中使用 DNS。

> **Note**
>
> 如果您在 VPC 中有可公开访问的集群且该集群为预置集群，则在您调整该集群的大小之前，从该 VPC 中发出的连接将继续使用弹性 IP 地址连接到集群。即使设置了前面的参数，也会发生此情况。从同一 VPC 中连接到可公开访问的集群时，任何新创建的集群都将遵循使用私有 IP 地址这一新行为。

弹性 IP 地址是用于访问 VPC 外部资源的外部 IP 地址。对于预置集群，它与公有 IP 地址和私有 IP 地址（显示在 Amazon Redshift 控制台中的节点 IP 地址下）不相关。无论集群是否可公开访问，公有和私有集群节点 IP 地址都会显示。它们仅在某些情况下用于配置远程主机上的入口规则。在使用 Secure Shell (SSH) 连接从 Amazon EC2 实例或其他远程主机加载数据时，会发生这些情况。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的步骤 1：检索集群公有密钥和集群节点 IP 地址。

> **Note**
>
> 节点 IP 地址不适用于 Redshift Serverless 工作组。

在您创建集群或从快照还原集群时，将预置集群与弹性 IP 地址相关联的选项可用。在某些情况下，您可能需要将集群与弹性 IP 地址关联，或更改与集群关联的弹性 IP 地址。要在创建集群后附加弹性 IP 地址，请首先更新集群，使其不可公开访问，然后在同一个操作中使其可公开访问并添加弹性 IP 地址。

有关如何使预置集群或 Amazon Redshift Serverless 工作组可公开访问并向该工作组分配弹性 IP 地址的更多信息，请参阅使用默认或自定义安全组配置实现公开访问功能。

- 关联 VPC 安全组。

  您可以使用 VPC 安全组授予入站访问权限。有关更多信息，请参阅为 Amazon Redshift 集群配置安全组通信设置，其中提供了有关在客户端和预调配集群或 Amazon Redshift Serverless 工作组之间配置入站和出站规则的指南。另一个有助于您了解安全组的资源是《Amazon VPC 用户指南》中的 VPC 中的安全性

还原 VPC 中的预置集群或 Serverless 工作组的快照

只能在 VPC 内部还原 VPC 中的集群或 Serverless 工作组的快照，而不能在 VPC 外部。您可以在同一 VPC 中或您账户中的其他 VPC 中还原快照。有关 快照的更多信息，请参阅 Amazon Redshift 快照和备份。

## 在 VPC 中创建 Redshift 预置集群或 Amazon Redshift Serverless 工作组

以下介绍了如何在虚拟私有云（VPC）中部署集群或工作组的一般步骤。

在 VPC 中创建集群或 Serverless 工作组

1. 配置 VPC – 您可以在账户的默认 VPC（如果账户已有）中或您创建的 VPC 中创建 Redshift 资源。有关更多信息，请参阅 使用 EC2 创建集群。要创建 VPC，请参阅《Amazon VPC 用户指南》中的 VPC 的子网。记下 VPC 标识符、子网和子网的可用区。您在启动集群或工作组时需要使用这些信息。

   > **Note**
   >
   > 您必须在自己的 VPC 中至少定义一个子网，以便您在下一步中将其添加到子网组。有关将子网添加到 VPC 的更多信息，请参阅《Amazon VPC 用户指南》中的将子网添加到 VPC 中。

2. 创建 Amazon Redshift 集群子网组，以指定 Amazon Redshift 集群可在 VPC 中使用的子网。对于 Redshift Serverless，您无需创建子网组，而是在创建工作组时为其分配一组子网。创建工作组时，您可以在 Serverless 控制面板中执行此操作。

   您可以使用 Amazon Redshift 控制台或以编程方式创建子网组。有关更多信息，请参阅 Redshift 资源的子网。

3. 在您要与集群或工作组相关联的 VPC 安全组中，授予针对入站连接的访问权限。您可以允许 VPC 外部（在公共 Internet 上）的客户端连接到集群。为此，请将集群与一个授予入站访问权限的 VPC 安全组相关联。有关更多信息，请参阅 为 Amazon Redshift 集群或 Amazon Redshift Serverless 工作组配置安全组通信设置。

4. 按照以下步骤，在 Redshift 预置控制台中创建集群，或者在 Amazon Redshift Serverless 控制台中创建工作组。在网络和安全性中，指定您设置的虚拟私有云（VPC）、集群子网组和 VPC 安全组。

   有关介绍创建预置数据仓库集群的更详细步骤的演练，请参阅《Amazon Redshift 入门指南》中的 Amazon Redshift 预置数据仓库入门。有关创建 Amazon Redshift Serverless 工作组的更多信息，请参阅《Amazon Redshift 入门指南》中的 Amazon Redshift Serverless 数据仓库入门。

您可以按照“入门”步骤操作，通过上传示例数据并尝试示例查询来测试集群或工作组。有关更多信息，请参阅《Amazon Redshift 入门指南》中的 Amazon Redshift Serverless 数据仓库入门。

## VPC 安全组

在您预置 Amazon Redshift 集群或 Amazon Redshift Serverless 工作组时，默认情况下限制访问，因此任何人都无法进行访问。要为其他用户授予入站访问权限，您可以将其与安全组关联起来。如果您处于 EC2-VPC 平台上，则可使用现有的 Amazon VPC 安全组或者定义一个新的安全组。然后，您可以将安全组关联到集群或工作组，如下所述。如果您在 EC2-Classic 平台上，则可以定义一个安全组，并将其与集群或工作组关联。有关在 EC2-Classic 平台上使用安全组的更多信息，请参阅 Amazon Redshift 安全组。

VPC 安全组中包含一组规则，用于控制对 VPC 上实例（如您的集群）的访问。各个规则根据 IP 地址范围或其他 VPC 安全组设置访问权限。当您将 VPC 安全组与集群或工作组关联后，在 VPC 安全组中定义的规则即可控制访问权限。

您在 EC2-VPC 平台上预置的每个集群均拥有一个或多个与其关联的 Amazon VPC 安全组。Amazon VPC 提供一个名为 default 的 VPC 安全组，该安全组是在您创建 VPC 时自动创建的。如果您在创建 Redshift 资源时未指定其他 VPC 安全组，则您在 VPC 内启动的每个集群都会自动与默认 VPC 安全组关联。您可以在创建集群时将 VPC 安全组与之关联，也可以稍后通过修改该集群再将 VPC 安全组与之关联。

以下屏幕截图介绍了默认 VPC 安全组的默认规则。

| **Inbound** | | | |
|---|---|---|---|
| **Source** | **Protocol** | **Port Range** | **Comments** |
| The security group ID (*sg-xxxxxxxx*) | All | All | Allow inbound traffic from instances assigned to the same security group |
| **Outbound** | | | |
| **Destination** | **Protocol** | **Port Range** | **Comments** |
| 0.0.0.0/0 | All | All | Allow all outbound traffic |

您可以视需要更改默认 VPC 安全组的规则。

如果默认 VPC 安全组足以满足您的需求，则无需创建更多 VPC 安全组。不过，您可以选择创建其他 VPC 安全组，以便更好地管理入站访问权限。例如，假设您正在 Amazon Redshift 集群或 Serverless 工作组上运行一项服务，并向客户提供多种不同的服务水平。如果您不希望在所有服务级别提供相同的访问权限，则可能需要创建单独的 VPC 安全组，每种服务级别一个。然后，您可以将这些 VPC 安全组与您的集群或工作组关联起来。

您可以为一个 VPC 创建最多 100 个 VPC 安全组，并将一个 VPC 安全组与多个集群和工作组关联。但请注意，可以与一个集群或工作组关联的 VPC 安全组的数量有限制。

Amazon Redshift 可将更改即时应用到 VPC 安全组。因此，如果您已将该 VPC 安全组与某个集群关联，则更新后的 VPC 安全组中的入站集群访问规则将立即应用。

您可以在 https://console.aws.amazon.com/vpc/ 上创建和修改 VPC 安全组。您还可以使用 AWS CLI、Amazon EC2 CLI 和 AWS Tools for Windows PowerShell 以编程方式管理 VPC 安全组。有关使用 VPC 安全组的更多信息，请参阅《Amazon VPC 用户指南》中的您的 VPC 的安全组。

# 为 Amazon Redshift 集群或 Amazon Redshift Serverless 工作组配置安全组通信设置

本主题可帮助您配置安全组，以适当地路由和接收网络流量。以下是一些常见应用场景：

- 您为 Amazon Redshift 集群或 Amazon Redshift Serverless 工作组开启了公开访问功能，但该集群并未接收流量。为此，您必须配置入站规则，以允许流量从互联网到达集群。
- 您的集群或工作组不可公开访问，您使用 Redshift 的预配置原定设置 VPC 安全组以允许入站流量。但是，您需要使用默认安全组以外的其他安全组，而这个自定义安全组不允许入站流量。必须将其配置为允许通信。

以下部分可帮助您为每个使用案例选择正确的响应，并说明您如何根据您的要求配置网络流量。您可以选择使用这些步骤来设置来自其他私有安全组的通信。

> **Note**
>
> 在大多数情况下，不会在 Amazon Redshift 中自动配置网络流量设置。这是因为它们可能在精细级别上有所不同，这取决于流量来源是互联网还是私有安全组，也因为安全要求各不相同。

## 使用默认或自定义安全组配置实现公开访问功能

如果您正在创建集群或工作组或已经拥有集群或工作组，请执行以下配置步骤，以使它可公开访问。当您选择默认安全组或自定义安全组时，以下这一点都适用：

1. 查找网络设置：
   - 对于预调配的 Amazon Redshift 集群，选择属性选项卡，然后在网络和安全设置下，为您的集群选择 VPC。
   - 对于 Amazon Redshift Serverless 工作组，请选择工作组配置。从列表中选择工作组。然后，在数据访问下的网络和安全面板中，选择编辑。

2. 为您的 VPC 配置互联网网关和路由表。您可以通过按名称选择 VPC 开始配置。此时会打开 VPC 控制面板。要从互联网连接到可公开访问的集群或工作组，必须将互联网网关附加到路由表。您可以通过在 VPC 控制面板中选择路由表来进行配置。确认使用源 0.0.0.0/0 或公共 IP CIDR 设置该互联网网关的目标。路由表必须与您的集群所在的 VPC 相关联。有关为 VPC 设置互联网访问的更多信息（如此处所述），请参阅 Amazon VPC 文档中的启用互联网访问。有关配置路由表的更多信息，请参阅配置路由表。
3. 配置互联网网关和路由表后，返回到 Redshift 的网络设置。通过选择安全组，然后选择入站规则，打开入站访问。选择编辑入站规则。
4. 根据您的要求，为一条或多条入站规则选择协议和端口，以允许来自客户端的流量。对于 RG 或 RA3 集群，请在 5431-5455 或 8191-8215 范围内选择一个端口。完成后，保存每条规则。
5. 编辑可公开访问设置以启用它。您可以从集群或工作组的操作菜单中执行此操作。

当您开启可公开访问的设置时，Redshift 会创建弹性 IP 地址。它是与您的 AWS 账户关联的静态 IP 地址。VPC 外部的客户端可以使用它进行连接。

有关配置安全组的更多信息，请参阅Amazon Redshift 安全组。

您可以通过使用客户端进行连接来测试您的规则，如果您要连接到 Amazon Redshift Serverless，请执行以下操作。完成网络配置后，使用客户端工具进行连接，例如 Amazon Redshift RSQL。使用您的 Amazon Redshift Serverless 域作为主机，输入以下内容：

```
rsql -h workgroup-name.account-id.region.amazonaws.com -U admin -d dev -p 5439
```

## 使用默认或自定义安全组配置实现私有访问功能

当您不通过互联网与集群或工作组通信时，它称为可私密访问。如果您在创建原定设置安全组时选择了该安全组，则该安全组将包含以下原定设置通信规则：

- 入站规则，允许来自分配给此安全组的所有资源的流量。
- 允许所有出站流量的出站规则。此规则的目标是 0.0.0.0/0。在无类别域间路由 (CIDR) 表示法中，它表示所有可能的 IP 地址。

您可以通过为集群或工作组选择安全组，在控制台中查看规则。

如果您的集群或工作组和客户端都使用原定设置安全组，则无需进行任何其它配置，即可允许网络流量。但是，如果您删除或更改了 Redshift 或客户端的原定设置安全组中的任何规则，则这不再适用。在这种情况下，您必须配置规则以允许入站和出站通信。常见的安全组配置如下：

- 对于客户端 Amazon EC2 实例：
  - 允许客户端的 IP 地址的入站规则。
  - 一条出站规则，允许提供给 Redshift 使用的所有子网的 IP 地址范围（CIDR 块）。或者您可以指定 0.0.0.0/0，即所有 IP 地址范围。
- 对于您的 Redshift 集群或工作组：
  - 允许客户端安全组的入站规则。
  - 允许流量达到 0.0.0.0/0 的出站规则。通常，出站规则允许所有出站流量。或者，您可以添加出站规则以允许流量到达客户端安全组。在这种可选情况下，并不总是需要出站规则，因为允许每个请求的响应流量到达实例。有关请求和响应行为的更多详细信息，请参阅《Amazon VPC 用户指南》中的安全组。

如果您更改了为供 Redshift 使用而指定的任何子网或安全组的配置，则可能需要相应地更改流量规则以保持通信畅通。有关创建入站和出站规则的更多信息，请参阅《Amazon VPC 用户指南》中的 VPC CIDR 块。有关从客户端连接到 Amazon Redshift 的更多信息，请参阅在 Amazon Redshift 中配置连接。

## Redshift 资源的子网

如果您要在虚拟私有云（VPC）中创建预置集群，则可以创建集群子网组。每个 VPC 可以有一个或多个子网，这是 VPC 中的一部分 IP 地址，使您能够根据安全和运营需求对资源进行分组。在创建预置集群时，您可以创建子网组以在 VPC 中指定一组子网。在预置集群控制面板中，您可以在配置下查找和编辑集群子网组。在预置集群的初始配置过程中，您可以指定子网组，Amazon Redshift 将在其中一个子网中创建集群。有关 VPC 服务的更多信息，请参阅 Amazon VPC 产品详细信息页面。

Amazon Redshift Serverless 工作组的子网配置与预置集群类似，不过步骤略有不同。创建和设置 Serverless 工作组时，您应为该工作组指定子网，这些子网将被添加到列表中。您可以在 Serverless 控制面板中选择工作组属性，来查看现有工作组的子网。子网显示在网络和安全属性中。有关更多信息，请参阅创建带有命名空间的工作组。

有关创建 VPC 的更多信息，请参阅 Amazon VPC 用户指南文档。

为预置集群创建子网组之后，或为 Serverless 工作组选择子网后，您可以删除之前添加的子网，也可以添加更多子网。您可以使用控制台或 API 操作来进行这些更改。有关预置集群的 API 操作的更多信息，请参阅 ModifyClusterSubnetGroup。有关适用于 Serverless 工作组的 API 操作，请参阅 UpdateWorkgroup。

您可以在子网组中的一个子网上预置集群。您可以通过集群子网组在 Virtual Private Cloud (VPC) 中指定一组子网。

> **Warning**
>
> 在集群维护操作期间，例如经典大小调整、暂停和恢复、多可用区故障失效或其他事件，您的预置计算节点可能会移至 Amazon Redshift 集群子网组内的另一个子网。请注意，子网组中的所有子网必须具有相同的网络 ACL 入站和出站规则，以及相同的路由表路由。这样可以确保与 Amazon Redshift 计算资源的双向连接，使得出现此类维护事件后，可以顺利地通信和运行。避免将具有不同网络 ACL 或路由表配置的子网添加到同一个 Amazon Redshift 集群子网组。有关配置子网的更多信息，请参阅《Amazon VPC 用户指南》中的 VPC 的子网。有关 Redshift 多可用区部署的更多信息，请参阅《Redshift 管理指南》中的多可用区部署。《Redshift 管理指南》中还介绍了调整集群大小。

## 创建集群子网组

以下过程将为您演练如何为预置集群创建子网组。您必须定义至少一个集群子网组，才能在 VPC 中配置集群。

为预置集群创建集群子网组

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Configurations（配置），然后选择 Subnet groups（子网组）。此时将显示子网组列表。
3. 选择 Create cluster subnet group（创建集群子网组）以显示创建页面。
4. 输入子网组的信息，包括要添加的子网。
5. 选择 Create cluster subnet group（创建集群子网组）创建包含您选择的子网的组。

> **Note**
>
> 有关如何创建包含一组子网的 Amazon Redshift Serverless 工作组的信息，请参阅《Amazon VPC 用户指南》中的创建带有命名空间的工作组或创建子网。

## 修改集群子网组

创建子网组后，可以在 Amazon Redshift 控制台上修改其信息。以下过程将为您演练如何修改预置集群的子网组。

为预置集群修改集群子网组

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Configurations（配置），然后选择 Subnet groups（子网组）。此时将显示子网组列表。
3. 选择要修改的子网组。
4. 对于 Actions（操作），选择 Modify（修改）以显示子网组的详细信息。
5. 更新子网组的信息。
6. 选择 Save（保存）以修改组。

在某些情况下，更改或删除子网需要额外的步骤。例如，这篇 AWS 知识中心文章如何将预调配的 Amazon Redshift 集群移入不同的子网？介绍了涵盖移动集群的使用案例。

## 删除预置集群的集群子网组

使用完集群子网组后，应通过删除该组进行清理。以下过程将为您演练为预置集群删除子网组的步骤。

删除集群子网组

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Configurations（配置），然后选择 Subnet groups（子网组）。此时将显示子网组列表。
3. 选择要删除的子网组，然后选择 Delete（删除）。

> **Note**
>
> 您无法删除集群正在使用的集群子网组。

# 屏蔽 VPC 和子网的公共访问权限

VPC 屏蔽公共访问权限（BPA）是一种集中式安全功能，可以使用它来阻止您在 AWS 区域中拥有的 VPC 和子网中的资源通过互联网网关和仅出口互联网网关访问互联网，或从互联网进行访问。如果您在 AWS 账户中开启此功能，则默认情况下，它将影响 Amazon Redshift 使用的任何 VPC 或子网。这意味着 Amazon Redshift 会阻止公众执行所有操作。

当您开启了 VPC BPA 并想通过公共互联网使用 Amazon Redshift API 时，您必须添加一个排除项，以便在您的 VPC 或子网中使用 Amazon EC2 API。排除可以采用以下任一模式：

- 双向：允许进出已排除 VPC 和子网的所有互联网流量。
- 仅出口：允许来自已排除 VPC 和子网的出站互联网流量。进入已排除 VPC 和子网的入站互联网流量已被阻止。这仅在 BPA 设置为双向时适用。

VPC BPA 排除项将整个 VPC 或 VPC 内的特定子网指定为可公开访问。该边界内的网络接口遵守常规的 VPC 网络控制，例如安全组、路由表和网络 ACL，以决定该接口是否具有访问公共互联网的路由和权限。有关添加排除项的更多信息，请参阅《Amazon VPC 用户》指南中的创建和删除排除项。

### 预置集群

子网组是同一 VPC 中子网的组合。如果预置集群的子网组位于开启了 VPC BPA 的账户中，则以下功能将被阻止：

- 创建公有集群
- 还原公有集群
- 将私有集群修改为公有集群
- 当子网组中至少有一个公有集群时，向该子网组添加已开启 VPC BPA 的子网

### 无服务器集群

Redshift Serverless 不使用子网组。相反，每个集群都有自己的子网集。如果工作组位于开启了 VPC BPA 的账户中，则以下功能将被阻止：

- 创建公有访问工作组
- 将私有工作组修改为公有工作组
- 当工作组为公有工作组时，向工作组添加已开启 VPC BPA 的子网

# 使用 Redshift 增强型 VPC 路由控制网络流量

在使用 Amazon Redshift 增强型 VPC 路由时，Amazon Redshift 会强制通过基于 Amazon VPC 服务的 Virtual Private Cloud (VPC) 路由集群和数据存储库之间的所有 COPY 和 UNLOAD 流量。通过使用增强型 VPC 路由，您可以使用标准 VPC 功能，例如 VPC 安全组、网络访问控制列表 (ACL)、VPC 终端节点、VPC 终端节点策略、互联网 网关和域名系统 (DNS) 服务器，如 Amazon VPC 用户指南中所述。您可以使用这些功能来扩展 Amazon Redshift 集群与其他资源之间的数据流。在使用增强型 VPC 路由以通过 VPC 路由流量时，您还可以使用 VPC 流日志监视 COPY 和 UNLOAD 流量。

Amazon Redshift 集群和 Amazon Redshift Serverless 工作组均支持增强型 VPC 路由。您不能将增强型 VPC 路由与 Redshift Spectrum 一起使用。有关更多信息，请参阅 通过增强型 VPC 路由查询数据湖表。

如果未开启增强型 VPC 路由，则 Amazon Redshift 会通过互联网路由流量，包括至 AWS 网络中的其他服务的流量。

> **Important**
>
> 由于增强型 VPC 路由影响 Amazon Redshift 访问其他资源的方式，除非您正确配置了 VPC，否则 COPY 和 UNLOAD 命令可能会失败。您必须专门在集群的 VPC 和数据资源之间创建网络路径，如下所述。

在对开启了增强型 VPC 路由的集群运行 COPY 或 UNLOAD 命令时，VPC 使用最严格或最具体的可用网络路径将流量路由到指定的资源。

例如，可以在您的 VPC 中配置以下路径：

- VPC 端点 – 对于传输到集群或工作组所在 AWS 区域中的 Amazon S3 存储桶的流量，您可以创建 VPC 端点来将流量直接传送到该存储桶。在使用 VPC 终端节点时，您可以附加端点策略来管理对 Amazon S3 的访问。有关将端点与 Redshift 结合使用的更多信息，请参阅使用 VPC 端点控制数据库流量。如果您使用的是 Lake Formation，则可以在 AWS Lake Formation 和接口 VPC 端点 (AWS PrivateLink) 中找到有关在 VPC 和 AWS Lake Formation 之间建立私有连接的更多信息。

  > **Note**
  >
  > 将 Redshift VPC 端点与 Amazon S3 VPC 网关端点一起使用时，您必须在 Redshift 中启用增强型 VPC 路由。有关更多信息，请参阅用于 Amazon S3 的网关端点。

- NAT 网关 – 您可以连接到另一个 AWS 区域中的 Amazon S3 存储桶，并且可以连接到 AWS 网络中的另一个服务。您还可以访问 AWS 网络外部的主机实例。为此，请配置网络地址转换 (NAT) 网关，如 Amazon VPC 用户指南中所述。
- 互联网网关 – 要连接到 VPC 外部的 AWS 服务，您可以将互联网网关附加到您的 VPC 子网，如 Amazon VPC 用户指南中所述。要使用互联网网关，您的集群或工作组必须可以公开访问，来允许其他服务与您的集群或工作组通信。

有关更多信息，请参阅 Amazon VPC 用户指南中的 VPC 终端节点。

> **Note**
>
> 增强型 VPC 路由还会影响当用户登录时 Redshift 向 AWS IAM Identity Center 发出的调用。如果您的集群或工作组使用 AWS IAM Identity Center 身份验证，则必须为这些服务创建接口 VPC 端点。否则，登录将失败。有关将增强型 VPC 路由与 AWS IAM Identity Center 结合使用的更多信息，请参阅将 AWS IAM Identity Center 身份验证与增强型 VPC 路由结合使用。

使用增强型 VPC 路由不会产生任何额外的费用。您可能需要为某些操作支付额外的数据传输费用。其中包括在不同 AWS 区域中 UNLOAD 到 Amazon S3 之类的操作。使用公有 IP 地址从 Amazon EMR 或安全外壳 (SSH) 执行 COPY。有关定价的更多信息，请参阅 Amazon EC2 定价。

主题


- 使用 VPC 端点控制数据库流量
- 启用增强型 VPC 路由
- 通过增强型 VPC 路由查询数据湖表


## 使用 VPC 端点控制数据库流量

您可以使用 VPC 端点，创建 VPC 中的 Amazon Redshift 集群或 Serverless 工作组与 Amazon Simple Storage Service（Amazon S3）之间的托管连接。在执行此操作时，您的数据库与 Amazon S3 数据之间的 COPY 和 UNLOAD 流量将保留在您的 Amazon VPC 中。可以将终端节点策略附加到您的终端节点，以便更严格地管理对数据的访问。例如，可以向 VPC 终端节点添加策略以仅允许将数据上载到您账户中的特定 Amazon S3 存储桶。

要使用 VPC 终端节点，请为数据仓库所在的 VPC 创建 VPC 端点，然后开启增强型 VPC 路由。可以在 VPC 中创建集群时开启增强型 VPC 路由，也可以修改 VPC 中的集群或工作组以使用增强型 VPC 路由。

VPC 端点使用路由表来控制 VPC 中的集群或工作组和 Amazon S3 之间的流量路由。与指定路由表关联的子网中的所有集群和工作组会自动使用该端点来访问服务。

您的 VPC 使用与流量匹配的最具体的或最严格的路由来决定路由流量的方式。例如，假设路由表中有一条路由用于所有指向互联网网关和 Amazon S3 端点的互联网流量 (0.0.0.0/0)。在这种情况下，对所有传送到 Amazon S3 的流量优先使用端点路由。这是因为 Amazon S3 服务的 IP 地址范围比 0.0.0.0/0 更具体。在此示例中，所有其他互联网流量（包括定位到其他 AWS 区域 区域内的 Amazon S3 存储桶的流量）将流向互联网网关。

有关创建端点的更多信息，请参阅《Amazon VPC User Guide》中的 Create a VPC endpoint。

您使用端点策略控制从集群或工作组到包含数据文件的 Amazon S3 存储桶的访问。要实现更具体的控制，您可以选择附加一个自定义终端节点策略。有关更多信息，请参阅《AWS PrivateLink 指南》中的使用端点策略控制对服务的访问权限。

> **Note**
>
> AWS Database Migration Service（AWS DMS）是一项云服务，可轻松迁移关系数据库、数据仓库及其他类型的数据存储。这项服务可以连接到任何 AWS 源数据库或目标数据库，包括启用了 VPC 的 Amazon Redshift 数据库，但有一些配置限制。支持 Amazon VPC 端点可使 AWS DMS 更轻松地维护复制任务的端到端网络安全。有关将 Redshift 与 AWS DMS 结合使用的详细信息，请参阅《AWS Database Migration Service 用户指南》中的将 VPC 端点配置为 AWS DMS 源端点和目标端点。

使用终端节点不收取任何额外费用。采用标准的数据传输和资源使用计费方式。有关定价的更多信息，请参阅 Amazon EC2 定价。

## 启用增强型 VPC 路由

您可以在创建或修改集群以及创建或修改 Amazon Redshift Serverless 工作组时开启增强型 VPC 路由。

要使用增强型 VPC 路由，集群或 Serverless 工作组，必须满足以下要求和约束：

- 集群必须在 VPC 中。

  如果您附加一个 Amazon S3 VPC 端点，则该 VPC 端点仅用于访问同一 AWS 区域中的 Amazon S3 存储桶。要访问其他 AWS 区域中的存储桶（不使用 VPC 端点）或访问其他 AWS 服务，请使您的

集群或 Serverless 工作组可公开访问，或者使用网络地址转换（NAT）网关。有关更多信息，请参阅 在 VPC 中创建 Redshift 预置集群或 Amazon Redshift Serverless 工作组。

- 您必须在 VPC 中启用域名服务 (DNS) 解析。或者，如果您使用自己的 DNS 服务器，请确保将针对 Amazon S3 的 DNS 请求正确解析为 AWS 维护的 IP 地址。有关更多信息，请参阅 Amazon VPC 用户指南中的在您的 VPC 中使用 DNS。
- 必须在 VPC 中启用 DNS 主机名。DNS 主机名默认处于启用状态。
- 您的 VPC 终端节点策略必须允许访问用于 Amazon Redshift 中的 COPY、UNLOAD 或 CREATE LIBRARY 调用的任何 Amazon S3 存储桶，包括访问涉及的任何清单文件。对于从远程主机执行的 COPY 操作，您的终端节点策略必须允许访问每台主机。有关更多信息，请参阅 Amazon Redshift 数据库开发人员指南中的 COPY、UNLOAD 和 CREATE LIBRARY 的 IAM 权限。

为预置集群开启增强型 VPC 路由

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Provisioned clusters dashboard（预置集群控制面板），然后选择 Create cluster（创建集群）并输入 Cluster details（集群详细信息）属性。
3. 要显示 Additional configurations (其他配置) 部分，请选择关闭 Use defaults (使用默认值)。
4. 导航到 Network and security（网络和安全）部分。
5. 要开启 Enhanced VPC routing（增强型 VPC 路由），请选择 Turn on（开启）以强制集群流量通过 VPC。
6. 选择创建集群以创建集群。集群可能需要几分钟才可以使用。

为 Amazon Redshift Serverless 开启增强型 VPC 路由

1. 登录到 AWS 管理控制台 并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Serverless dashboard（Serverless 控制面板），然后选择 Create workgroup（创建工作组）并输入工作组的属性。
3. 导航到 Network and security（网络和安全）部分。
4. 选择 Turn on enhanced VPC routing（开启增强型 VPC 路由），通过 VPC 路由网络流量。
5. 选择 Next（下一步），输入工作组属性，直到 Create（创建）工作组。

# 通过增强型 VPC 路由查询数据湖表

增强型 VPC 路由开启后，Amazon Redshift 通过 VPC 路由流量。数据湖表将数据存储在 Amazon S3 中，将元数据存储在 AWS Glue Data Catalog中，这两者都位于您的 VPC 之外。要使数据湖查询成功执行，您的集群或工作组必须能够访问 Amazon S3 和 AWS Glue。如果您的数据湖表由 AWS Lake Formation 管理，则它还必须能够访问 Lake Formation。由于增强型 VPC 路由影响 Amazon Redshift 访问这些外部资源的方式，除非您正确配置了 VPC，否则查询可能会失败。

在集群或工作组运行的 VPC 与子网中创建以下 VPC 端点，并将其与相应的路由表关联。集成式数据湖查询引擎与 Amazon Redshift Spectrum 均需要这些端点：

- Amazon S3 网关端点 - 为集群或工作组提供经由 AWS 网络访问 Amazon S3 的路径，用于读取数据湖数据文件。有关步骤，请参阅《Amazon VPC 用户指南》中的 Amazon S3 网关端点。
- AWS Glue 接口端点 - 为集群或工作组提供访问 AWS Glue Data Catalog的路由，用于解析数据湖的架构与表。有关步骤，请参阅《Amazon VPC 用户指南》中的创建接口端点。
- AWS Lake Formation 接口端点 - 仅当您的数据湖表由 AWS Lake Formation 管理时才需要。为集群或工作组提供访问 Lake Formation 的路由。有关步骤，请参阅《Amazon VPC 用户指南》中的创建接口端点。

无论您如何查询外部表，端点配置都是一样的，但是流量路由方式因查询引擎而异。以下各部分介绍了 RG 预调配集群和 Amazon Redshift Serverless 上的集成式数据湖查询引擎的行为，以及 RA3 和 DC2 预调配集群上的 Redshift Spectrum 的行为。

## RG 预调配集群和 Amazon Redshift Serverless 的集成式数据湖查询引擎

RG 预调配集群和 Amazon Redshift Serverless 包含集成式数据湖查询引擎。该引擎运行在集群或工作组自身的计算资源上，位于您的 VPC 内。当增强型 VPC 路由已开启且所需 VPC 端点就位时，对 Amazon S3 和 AWS Glue 的访问请求会从 VPC 内的计算资源流经这些 VPC 端点。该流量始终处于您的 VPC 边界之内和 AWS 网络之上。此访问权限基于附加到集群或工作组的 IAM 角色授予。要进一步管理该流量，您可以修改 IAM 角色以及附加到 Amazon S3 存储桶的策略。

## RA3 和 DC2 预调配集群的 Amazon Redshift Spectrum

Redshift Spectrum 在由 Amazon Redshift 拥有的 AWS 托管资源上运行，但这些资源位于您的 VPC 之外。因此，即使增强型 VPC 路由已开启，Redshift Spectrum 发往 Amazon S3 的流量也不会经过您的 VPC；不过集群仍然需要前文所述的 Amazon S3 和 AWS Glue VPC 端点来执行数据湖查询。

将通过 VPC 外部的 AWS 私有网络，安全地路由 Redshift Spectrum 和 Amazon S3 之间的流量。使用 Amazon 签名版本 4 协议 (SIGv4) 签署正在传输的流量并使用 HTTPS 对该流量加密。此流量基于附加

到 Amazon Redshift 集群的 IAM 角色进行授权。要进一步管理 Redshift Spectrum 流量，您可以修改集群的 IAM 角色以及附加到 Amazon S3 存储桶的策略。

## 数据湖查询的权限策略配置

查询 Amazon S3 中的数据湖表时，请考虑以下事项：

- Amazon S3 存储桶访问策略和 IAM 角色
- 代入 IAM 角色的权限
- 记录和审计 Amazon S3 访问
- 访问 AWS Glue 或 Amazon Athena

### Amazon S3 存储桶访问策略和 IAM 角色

您可以使用附加到存储桶的存储桶策略以及使用附加到预置集群的 IAM 角色，来控制对 Amazon S3 存储桶中数据的访问。

预置集群上的 Redshift Spectrum 无法访问存储在下面这样的 Amazon S3 存储桶中的数据：此类存储桶使用的存储桶策略限制仅访问指定的 VPC 端点。相反，应使用限制仅访问特定委托人（例如特定AWS账户或特定用户）的存储桶策略。增强型 VPC 路由开启后，RG 预调配集群和 Amazon Redshift Serverless 上的集成式数据湖查询引擎可以访问此类存储桶，因为其 Amazon S3 访问请求流经您的 VPC 端点。

对于被授予存储桶访问权限的 IAM 角色，请使用允许仅由 Amazon Redshift 服务委托人代入该角色的信任关系。该角色附加到集群时，只能在 Amazon Redshift 的上下文中使用，并且不能在集群外部共享。有关更多信息，请参阅 限制对 IAM 角色的访问。服务控制策略（SCP）也可用于进一步限制角色，请参阅《AWS Organizations 用户指南》中的阻止 IAM 用户和角色进行指定的更改，但指定的管理员角色除外。

> **Note**
>
> 要使用 Redshift Spectrum，不能制定任何阻止使用 Amazon S3 预签名 URL 的 IAM 策略。Amazon Redshift Spectrum 生成的预签名 URL 有效期为 1 小时，这样 Amazon Redshift 就有足够的时间从 Amazon S3 存储桶中加载所有文件。Redshift Spectrum 扫描的每个文件都会生成一个唯一的预签名 URL。对于包含 `s3:signatureAge` 操作的存储桶策略，请确保将该值至少设置为 3,600,000 毫秒。

以下示例存储桶策略允许访问由 AWS 账户 `123456789012` 拥有的指定存储桶。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "BucketPolicyForSpectrum",
            "Effect": "Allow",
            "Principal": {
                "AWS": ["arn:aws:iam::123456789012:role/redshift"]
            },
            "Action": [
                "s3:GetObject",
                "s3:ListBucketVersions",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

代入 IAM 角色的权限

附加到集群的角色应该具有信任关系，只允许 Amazon Redshift 服务代入它，如下所示。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
```

```
}
```

有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 Redshift Spectrum 的 IAM 策略。

记录和审计 Amazon S3 访问

使用 Amazon Redshift 增强型 VPC 路由的一个好处是，在 VPC 流日志中记录所有 COPY 和 UNLOAD 流量。源自 Redshift Spectrum 且传入 Amazon S3 的流量不会通过您的 VPC，因此它不会记录在 VPC 流日志中。当 Redshift Spectrum 访问 Amazon S3 中的数据时，它会在 AWS 账户和相应角色权限的上下文中执行这些操作。您可以使用 AWS CloudTrail 和 Amazon S3 中的服务器访问日志记录来记录和审计 Amazon S3 访问。

确保将 S3 IP 范围添加到您的允许列表中。要了解有关所需 S3 IP 范围的更多信息，请参阅 Network isolation（网络隔离）。

AWS CloudTrail 日志

要跟踪 Amazon S3 中对象的所有访问，包括 Redshift Spectrum 访问，请为 Amazon S3 对象启用 CloudTrail 日志记录。

您可以使用 CloudTrail 来查看、搜索、下载、归档、分析和响应您的 AWS 基础设施中的账户活动。有关更多信息，请参阅 CloudTrail 入门。

预设情况下，CloudTrail 仅跟踪存储桶级别的操作。要跟踪对象级别的操作（例如 `GetObject`），请为每个已记录的存储桶启用数据和管理事件。

Amazon S3 服务器访问日志记录

服务器访问日志记录详细地记录对存储桶提出的各种请求。访问日志信息可能在安全和访问审计方面十分有用。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的如何启用服务器访问日志记录。

有关更多信息，请参阅 AWS 安全博客文章如何使用存储桶策略并应用深度防御来帮助保护您的 Simple Storage Service（Amazon S3）数据。

访问 AWS Glue 或 Amazon Athena

Redshift Spectrum 和集成式数据湖查询引擎都可以访问 AWS Glue 或 Athena 中的数据目录。另一种选择是为您的数据目录使用专用的 Hive 元存储。

推荐的 AWS Glue Data Catalog 的访问方法是为 AWS Glue 创建一个接口 VPC 端点（AWS PrivateLink）。当您使用 VPC 接口端点时，您的 VPC 与 AWS Glue 之间的通信会在 AWS 网络内路由。有关更多信息，请参阅创建接口端点。

或者，要访问 AWS Glue 和 Athena 的公共端点，可为 VPC 配置互联网网关或 NAT 网关，并将 VPC 安全组配置为允许发往这些公共端点的出站流量。可以在您的 VPC 中配置以下通道：

- 互联网网关 – 要连接到 VPC 外部的 AWS 服务，您可以将互联网网关附加到您的 VPC 子网，如 Amazon VPC 用户指南中所述。要使用互联网网关，预置集群必须具有一个公有 IP 地址来允许其他服务与其通信。
- NAT 网关 – 要连接到另一个 AWS 区域中的 Simple Storage Service（Amazon S3）存储桶或 AWS 网络中的另一种服务，可以配置网络地址转换（NAT）网关，如 Amazon VPC 用户指南中所述。使用此配置还可以访问 AWS 网络外部的主机实例。

# Amazon Redshift 事件

Amazon Redshift 跟踪集群事件并在您的 AWS 账户中将事件的相关信息保留几周。对于每个事件，Amazon Redshift 会报告事件发生日期、描述、事件源（例如，集群、参数组或快照）和源 ID 等信息。

Amazon Redshift 为某些事件提前提供通知。这些事件的事件类别是 `pending`。例如，如果集群中的一个节点需要硬件更新，我们会发送预先通知。您可以订阅与其他 Amazon Redshift 事件相同的待处理事件。有关更多信息，请参阅 Amazon Redshift 集群事件通知订阅。

您可以使用 Amazon Redshift 管理控制台、Amazon Redshift API 或 AWS 开发工具包以获取事件信息。您可以获取所有事件的列表，也可以应用筛选条件（如事件持续时间或开始和结束日期）以获取特定时间段的事件信息。

您还可以获取由特定源类型生成的事件，如集群事件或参数组事件。Source（源）列显示触发给定操作的资源名称和资源类型。

您可以创建 Amazon Redshift 事件通知订阅以指定一组事件筛选器。当发生与筛选条件匹配的事件时，Amazon Redshift 将使用 Amazon Simple Notification Service 主动通知您发生了该事件。

有关按源类型和类别筛选的 Amazon Redshift 事件列表，请参阅the section called “预置集群事件通知”

# Amazon Redshift 集群事件通知订阅

Amazon Redshift 使用 Amazon Simple Notification Service (Amazon SNS) 传输 Amazon Redshift 事件的通知。创建 Amazon Redshift 事件订阅即可启用通知。当给定的集群、快照、安全组或参数组发生事件时，您可以收到通知。创建订阅最简单的方式是使用 Amazon SNS 控制台。有关创建和订阅 Amazon SNS 主题的信息，请参阅 Amazon SNS 入门。

在 Amazon Redshift 订阅中，您需要为 Amazon Redshift 事件和 Amazon SNS 主题指定一组筛选条件。每当发生与筛选条件匹配的事件时，Amazon Redshift 就会向 Amazon SNS 主题发布通知消息。

然后，Amazon SNS 会将消息传输给任何拥有 Amazon SNS 主题订阅的 Amazon SNS 使用者。发送给 Amazon SNS 使用者的消息可以采用 AWS 区域的 Amazon SNS 支持的任何形式，如电子邮件、文本消息或对 HTTP 端点的调用。例如，所有区域都支持电子邮件通知，但 SMS 通知只能在美国东部（弗吉尼亚州北部）区域中创建。

> **Note**
>
> 目前，您只能创建对 Amazon SNS 标准主题（而不是 Amazon SNS FIFO 主题）的事件订阅。有关更多信息，请参阅《Amazon Simple Notification Service 开发人员指南》中的 Amazon SNS 事件来源。

创建事件通知订阅时，您需要指定一个或多个事件筛选条件。每当发生与所有筛选条件匹配的事件时，Amazon Redshift 就会通过该订阅发送通知。筛选条件包含源类型（例如集群或快照）、源 ID（例如集群或快照的名称）、事件类别（例如监控或安全）和事件严重性（例如 INFO 或 ERROR）。

如果您使用 CLI 或 API 创建事件通知订阅，则必须创建 Amazon Simple Notification Service 主题，并通过 Amazon SNS 控制台或 Amazon SNS API 订阅该主题。您还必须保留该主题的 Amazon Resource Name (ARN)，因为在提交 CLI 命令或者 API 操作时会用到。

您可以将AWS 管理控制台中的 Enabled（已启用）单选按钮设置为 No，或者使用 Amazon Redshift CLI 或 API 将 `Enabled` 参数设置为 `false`，从而轻松地关闭通知而无需删除订阅。

Amazon Redshift 事件订阅可指定以下事件条件：

- 源类型，值是集群、快照、参数组和安全组。
- 资源的源 ID，例如 `my-cluster-1` 或 `my-snapshot-20130823`。该 ID 所属的资源必须位于与事件订阅相同的 AWS 区域中。
- “事件”类别的值为“配置”、“管理”、“监控”和“安全性”和“待处理”
- 事件严重性，值是 INFO 或 ERROR。

事件条件可以单独进行指定，只不过您必须先指定源类型，然后才能在控制台中指定源 ID。例如，您可以指定事件类别，而无需指定源类型、源 ID 或严重性。对于不属于在源类型中指定的类型的资源，您可以为其指定源 ID，但系统不会针对来自这些资源的事件发送任何通知。例如，如果您指定了集群源类型和某个安全组的 ID，则该安全组触发的所有事件与相应源类型筛选条件均不匹配，因此系统不会针对这些事件发送任何通知。

Amazon Redshift 会针对任何与在订阅中指定的所有条件匹配的事件发送通知。返回的一系列事件的部分示例：

- 订阅指定的是集群源类型、my-cluster-1 的源 ID、监控类别和 ERROR 严重性。该订阅将仅针对来自 my-cluster-1、严重性为 ERROR 的监控事件发送通知。

- 订阅指定的是集群源类型、配置类别和 INFO 严重性。该订阅将针对来自 AWS 账户中的任何 Amazon Redshift 集群、严重性为 INFO 的配置事件发送通知。
- 订阅指定的是配置类别和 INFO 严重性。该订阅将针对来自 AWS 账户中的任何 Amazon Redshift 资源、严重性为 INFO 的配置事件发送通知。
- 订阅指定的是 ERROR 严重性。该订阅将针对来自 AWS 账户中的任何 Amazon Redshift 资源、严重性为 ERROR 的所有事件发送通知。

如果您在某个现有订阅中删除或重命名其名称被引用为源 ID 的对象，那么该订阅将保持有效，但不会转发来自该对象的任何事件。如果您日后使用该订阅源 ID 中引用的同一名称创建新对象，那么该订阅将会开始针对来自新对象的事件发送通知。

Amazon Redshift 会向 Amazon SNS 主题发布事件通知，该主题由其 Amazon Resource Name (ARN) 标识。当您使用 Amazon Redshift 控制台创建事件订阅时，您可以指定现有 Amazon SNS 主题，也可以请求控制台在创建订阅时创建主题。

发送到 Amazon SNS 主题的所有 Amazon Redshift 事件通知均会转发给订阅了该主题的所有 Amazon SNS 使用者。您可以使用 Amazon SNS 控制台对 Amazon SNS 主题进行更改，例如向主题添加使用者订阅或删除主题的使用者订阅。

以下章节列出了您可以收取通知的所有类型和事件。此外，它还提供有关订阅和使用 Amazon Redshift 事件订阅的详细信息。

# 创建事件通知订阅

您可以创建 Amazon Simple Notification Service (Amazon SNS) 事件通知订阅，以便在特定 Amazon Redshift 集群、快照、安全组或参数组发生事件时发送通知。这些通知被发送到 SNS 主题，然后 SNS 主题将消息传输给订阅该主题的任何 SNS 使用者。

发送给使用者的 SNS 消息可以采用 AWS 区域的 Amazon SNS 支持的任何通知形式，如电子邮件、文本消息或对 HTTP 端点的调用。例如，所有区域都支持电子邮件通知，但 SMS 通知只能在美国东部（弗吉尼亚北部）区域中进行创建。有关更多信息，请参阅 Amazon Redshift 预置集群事件通知。

要创建事件订阅

1. 登录到 AWS 管理控制台并打开 Amazon Redshift 控制台，网址：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Events（事件）。

3. 选择 Event subscription (事件订阅) 选项卡，然后选择 Create event subscriptions (创建事件订阅)。
4. 输入事件订阅的属性，例如名称、源类型、类别和严重性。您还可以启用 Amazon SNS 主题以获取事件通知。
5. 选择 Create event subscriptions（创建事件订阅）以创建订阅。

# Amazon Redshift 预置集群事件通知

本页介绍每种 Amazon Redshift 源类型的事件 ID 和类别。

集群源类型的类别和事件

下表显示了集群为源类型时的事件类别和事件列表。

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 配置 | REDSHIFT-EVENT-1000 | 信息 | 参数组 [参数组名称] 已于 [时间] 更新。如果只更改了动态参数，则现在正在修改关联的集群。如果您更改了静态参数，则在重新启动关联集群时将应用所有更新（包括动态参数）。 |
| 配置 | REDSHIFT-EVENT-1001 | 信息 | 您的 Amazon Redshift 集群 [集群名称] 已于 [时间] 修改为使用参数组 [参数组名称]。 |
| 配置 | REDSHIFT-EVENT-1500 | 错误 | Amazon VPC [VPC 名称] 不存在。未应用您对集群 [集群名称] 进行的配置更改。请访问AWS管理控制台，以解决该问题。 |
| 配置 | REDSHIFT-EVENT-1501 | 错误 | 您为 Amazon VPC [VPC 名称] 指定的客户子网 [子网名称] 不存在或无效。未应用您对集群 [集群名称] 进行的配置更改。请访问AWS 管理控制台，以解决该问题。 |
| 配置 | REDSHIFT-EVENT-1502 | 错误 | 集群子网组 [子网组名称] 中的子网没有可用的 IP 地址。无法创建集群 [集群名称]。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 配置 | REDSHIFT-EVENT-1503 | 错误 | Amazon VPC [VPC 名称] 未连接任何互联网网关。未应用您对集群 [集群名称] 进行的配置更改。请访问AWS 管理控制台，以解决该问题。 |
| 配置 | REDSHIFT-EVENT-1504 | 错误 | 无法访问集群 [集群名称] 的 HSM。 |
| 配置 | REDSHIFT-EVENT-1505 | 错误 | 无法注册集群 [集群名称] 的 HSM。尝试采用其他配置。 |
| 配置 | REDSHIFT-EVENT-1506 | 错误 | Amazon Redshift 超出了您的账户的弹性网络接口限制。请最多删除 [最大弹性网络接口数] 个弹性网络接口，或请求使用 EC2 增加每个 AWS 区域的网络接口数限制。 |
| 配置 | REDSHIFT-EVENT-1509 | 错误 | 无法创建 Amazon Redshift 集群 [集群名称]，因为已达到您账户的 VPC 端点限制。删除未使用的 VPC 端点或请求增加 VPC 端点的限制。<br><br>有关更多信息，请参阅《Amazon VPC 用户指南》中的 VPC 端点。 |
| 配置 | REDSHIFT-EVENT-1510 | ERROR | 我们检测到在 Amazon Redshift 集群 [集群名称] 上加载示例数据的尝试未成功。要加载示例数据，请首先将 VPC 配置为有权访问 Amazon S3 桶，然后创建新集群并加载示例数据。<br><br>有关更多信息，请参阅《Amazon Redshift 管理指南》中的启用增强型 VPC 路由。 |
| 配置 | REDSHIFT-EVENT-1511 | ERROR | 无法创建 Amazon Redshift 集群 [集群名称]，因为您已超出您账户中的弹性 IP 地址限制。请删除未使用的弹性 IP 地址，或者使用 Amazon EC2 请求提高限制。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 管理 | REDSHIFT-EVENT-2000 | 信息 | 您的 Amazon Redshift 集群 [集群名称] 已创建，随时可用。 |
| 管理 | REDSHIFT-EVENT-2001 | 信息 | 您的 Amazon Redshift 集群 [集群名称] 已于 [时间] 被删除。最终快照 [ 已/未] 保存。 |
| 管理 | REDSHIFT-EVENT-2002 | INFO | 集群 [集群名称] 的 VPC 安全组已于 [UTC 时间] 更新。 |
| 管理 | REDSHIFT-EVENT-2003 | INFO | 集群 [集群名称] 已于 [UTC 时间] 开始维护。 |
| 管理 | REDSHIFT-EVENT-2004 | INFO | 集群 [集群名称] 已于 [UTC 时间] 完成维护。 |
| 管理 | REDSHIFT-EVENT-2006 | INFO | 已于 [UTC 时间] 开始集群 [集群名称] 调整大小操作。集群处于只读模式。 |
| 管理 | REDSHIFT-EVENT-2007 | INFO | 集群 [集群名称] 的大小调整请求已得到确认。 |
| 管理 | REDSHIFT-EVENT-2008 | 信息 | 已于 [时间] 开始执行还原操作来创建新的 Amazon Redshift 集群 [集群名称] 快照 [快照名称]。要监控还原进度，请访问AWS 管理控制台。 |
| 管理 | REDSHIFT-EVENT-2013 | 信息 | 您的 Amazon Redshift 集群 [集群名称] 已于 [时间] 被重命名。 |
| 管理 | REDSHIFT-EVENT-2014 | 信息 | 收到 Amazon Redshift 集群 [集群名称] 表还原请求。 |
| 管理 | REDSHIFT-EVENT-2015 | 信息 | Amazon Redshift 集群 [集群名称] 表还原已于 [时间] 取消。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 管理 | REDSHIFT-EVENT-2016 | 信息 | 已于 [时间] 开始替换您的 Amazon Redshift 集群 [集群名称]。 |
| 管理 | REDSHIFT-EVENT-2017 | 信息 | 客户启动了于 [时间] 在 Amazon Redshift 集群 [集群名称] 上开始的维护。该集群在维护期间不可用。 |
| 管理 | REDSHIFT-EVENT-2018 | 信息 | 客户启动了于 [时间] 在 Amazon Redshift 集群 [集群名称] 上完成的维护。 |
| 管理 | REDSHIFT-EVENT-2019 | 错误 | 客户启动了于 [时间] 在 Amazon Redshift 集群 [集群名称] 上失败的维护。将集群返回到其原始状态。 |
| 管理 | REDSHIFT-EVENT-2020 | 信息 | 您的 Amazon Redshift 集群 [集群名称] 的跟踪已从 [起点跟踪] 修改为 [目的地跟踪]。 |
| 管理 | REDSHIFT-EVENT-2021 | 错误 | 从我们的容量池获取容量时，Amazon Redshift 集群 [集群名称] 的 [操作] 未成功。我们正在努力获取容量，但目前我们已取消了您的请求。删除此集群并在稍后重试。 |
| 管理 | REDSHIFT-EVENT-2022 | 错误 | 从我们的容量池获取容量时，Amazon Redshift 集群 [集群名称] 的 [操作] 未成功。我们正在努力获取容量，但目前我们已取消了您的请求。容量是在 [备用可用区] 中提供的。请删除该集群，然后在备用可用区中重试。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 管理 | REDSHIFT-EVENT-2023 | 错误 | 我们在您的单节点 Amazon Redshift 集群 [集群名称] 上检测到硬件故障，这可能会导致查询失败或集群间歇性可用。从我们的容量池中获取容量时，更换集群不成功。您需要从快照还原新集群。删除此集群，选择最新的可用快照，然后从该快照还原新集群。这将自动为您预配置正常工作的硬件。 |
| 管理 | REDSHIFT-EVENT-2024 | 错误 | 我们在您的单节点 Amazon Redshift 集群 [集群名称] 上检测到硬件故障，这可能会导致查询失败或集群间歇性可用。从我们的容量池中获取容量时，更换集群不成功。容量是在可用区中提供的：[备用可用区]。删除此集群，选择最新的可用快照，然后从该快照还原新集群。这将自动为您预配置正常工作的硬件。 |
| 管理 | REDSHIFT-EVENT-3011 | 信息 | 已于 [时间] 开始 Amazon Redshift 集群“[集群名称]”的弹性调整大小操作。在调整大小期间，我们将保持数据库连接。一些查询和连接可能在此操作期间终止或超时。 |
| 管理 | REDSHIFT-EVENT-3012 | 信息 | 我们已收到于 [时间] 开始的集群“[集群名称]”的弹性调整大小请求。在调整大小操作开始时，我们将提供事件通知。 |
| 待处理 | REDSHIFT-EVENT-2025 | INFO | 集群 [集群名称] 的数据库将在 [开始时间] 和 [结束时间] 之间更新。您的集群将不可访问。相应地做好计划。 |
| 待处理 | REDSHIFT-EVENT-2026 | INFO | 集群 [集群名称] 将在 [开始时间] 和 [结束时间] 之间更新。您的集群将不可访问。相应地做好计划。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 监控 | REDSHIFT-EVENT-2050 | 信息 | Amazon Redshift 集群 [集群名称] 中检测到硬件问题。已于 [时间] 发起替换请求。 |
| 监控 | REDSHIFT-EVENT-3000 | 信息 | 您的 Amazon Redshift 集群 [集群名称] 已于 [时间] 重启。 |
| 监控 | REDSHIFT-EVENT-3001 | 信息 | 已于 [时间] 自动替换您的 Amazon Redshift 集群 [集群名称] 上的一个节点，您的集群运行正常。 |
| 监控 | REDSHIFT-EVENT-3002 | 信息 | 您的 Amazon Redshift 集群 [集群名称] 已完成调整，可供读取和写入。[时间] 开始调整，[小时数] 小时后完成调整。 |
| 监控 | REDSHIFT-EVENT-3003 | 信息 | 已从快照 [快照名称] 成功创建 Amazon Redshift 集群 [集群名称]，可供使用。 |
| 监控 | REDSHIFT-EVENT-3007 | 信息 | 已于 [时间] 成功将您的 Amazon Redshift 快照 [快照名称] 从 [源 AWS 区域] 复制到 [目标 AWS 区域]。 |
| 监控 | REDSHIFT-EVENT-3008 | 信息 | Amazon Redshift 集群 [集群名称] 表还原已于 [时间] 启动。 |
| 监控 | REDSHIFT-EVENT-3009 | 信息 | Amazon Redshift 集群 [集群名称] 表还原已于 [时间] 成功完成。 |
| 监控 | REDSHIFT-EVENT-3010 | 错误 | Amazon Redshift 集群 [集群名称] 表还原于 [时间] 失败。 |
| 监控 | REDSHIFT-EVENT-3013 | 错误 | 为 Amazon Redshift 集群 [集群名称] 请求的弹性调整大小操作在 [事件] 失败，原因是 [原因]。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 监控 | REDSHIFT-EVENT-3014 | 信息 | Amazon Redshift 已于 [时间] 重启集群 [集群名称]。 |
| 待处理 | REDSHIFT-EVENT-3015 | INFO | 我们已计划在 [UTC 开始时间] 和 [UTC 结束时间] 之间重启您的集群 [集群名称]，以便进行维护。在此期间，您的集群将无法访问。我们建议您规划好停机事宜，以避免任何挫折。 |
| 监控 | REDSHIFT-EVENT-3016 | INFO | 集群将重启。 |
| 监控 | REDSHIFT-EVENT-3500 | 错误 | 您的 Amazon Redshift 集群 [集群名称] 的大小调整失败。将在几分钟内自动重新尝试调整大小操作。 |
| 监控 | REDSHIFT-EVENT-3501 | 错误 | [时间] 执行还原操作来从快照 [快照名称] 创建 Amazon Redshift 集群 [集群名称] 失败。请重新尝试操作。 |
| 监控 | REDSHIFT-EVENT-3504 | 错误 | Amazon S3 桶 [桶名称] 无效，无法对集群 [集群名称] 进行日志记录。 |
| 监控 | REDSHIFT-EVENT-3505 | 错误 | Amazon S3 桶 [桶名称] 没有适用于集群 [集群名称] 的 IAM 策略。 |
| 监控 | REDSHIFT-EVENT-3506 | 错误 | Amazon S3 桶 [桶名称] 不存在。无法继续集群 [集群名称] 的日志记录。 |
| 监控 | REDSHIFT-EVENT-3507 | 错误 | 无法使用 EIP [IP 地址] 创建 Amazon Redshift 集群 [集群名称]。此 EIP 已在使用中。 |
| 监控 | REDSHIFT-EVENT-3508 | 错误 | 无法使用 EIP [IP 地址] 创建 Amazon Redshift 集群 [集群名称]。找不到 EIP。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 监控 | REDSHIFT-EVENT-3509 | 错误 | 没有为集群 [集群名称] 启用跨区域快照复制。 |
| 监控 | REDSHIFT-EVENT-3510 | 错误 | Amazon Redshift 集群 [集群名称] 表还原于 [时间] 启动失败。原因：[原因]。 |
| 监控 | REDSHIFT-EVENT-3511 | 错误 | Amazon Redshift 集群 [集群名称] 表还原于 [时间] 失败。 |
| 监控 | REDSHIFT-EVENT-3512 | 错误 | Amazon Redshift 集群 [集群名称] 因硬件问题失败。集群正在从 [时间] 创建的最新快照 [快照名称] 中自动恢复。 |
| 监控 | REDSHIFT-EVENT-3513 | 错误 | Amazon Redshift 集群 [集群名称] 因硬件问题失败。集群正在从 [时间] 创建的最新快照 [快照名称] 中自动恢复。在此时间之后所做的任何数据库更改都需要重新提交。 |
| 监控 | REDSHIFT-EVENT-3514 | 错误 | Amazon Redshift 集群 [集群名称] 因硬件问题失败。集群正处于硬件故障状态。请删除集群并从 [时间] 创建的最新快照 [快照名称] 中恢复集群。 |
| 监控 | REDSHIFT-EVENT-3515 | 错误 | Amazon Redshift 集群 [集群名称] 因硬件问题失败。集群正处于硬件故障状态。请删除集群并从 [时间] 创建的最新快照 [快照名称] 中恢复集群。在此时间之后所做的任何数据库更改都需要重新提交。 |
| 监控 | REDSHIFT-EVENT-3516 | 错误 | Amazon Redshift 集群 [集群名称] 因硬件问题失败，且该集群没有备份。集群正处于硬件故障状态并且可以删除。 |
| 监控 | REDSHIFT-EVENT-3519 | 信息 | 集群 [集群名称] 已于 [时间] 开始重新启动。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 监控 | REDSHIFT-EVENT-3520 | 信息 | 集群 [集群名称] 已于 [时间] 完成重启操作。 |
| 监控 | REDSHIFT-EVENT-3521 | 信息 | 我们检测到集群“[[集群名称]”的连接问题。已在 [时间] 启动自动诊断检查。 |
| 监控 | REDSHIFT-EVENT-3522 | 信息 | 集群“[集群名称]”的恢复操作于 [时间] 失败。Amazon Redshift 团队在寻求解决方案。 |
| 监控 | REDSHIFT-EVENT-3533 | 错误 | “[集群名称]”的集群调整大小操作已于 [时间] 取消。由于 [原因]，操作被取消。[需要采取的操作]。 |
| 监控 | REDSHIFT-EVENT-3534 | 信息 | 已于 [时间] 完成 Amazon Redshift 集群“[集群名称]”的弹性调整大小操作。在我们传输数据时，现在可以对集群执行读取和写入操作。在完成数据传输之前，某些查询可能需要更长的时间才能完成。 |
| 监控 | REDSHIFT-EVENT-3537 | 信息 | 集群“[集群名称]”数据传输已于 [时间，以 UTC 表示] 完成。 |
| 监控 | REDSHIFT-EVENT-3600 | 信息 | Amazon Redshift 集群“[集群名称]”所请求的调整大小操作已于过去取消。回滚已于 [时间] 完成。 |
| 待处理 | REDSHIFT-EVENT-3601 | INFO | 集群 [集群名称] 上的节点将在 [开始时间] 和 [结束时间] 之间替换。您无法推迟该维护。相应地做好计划。 |
| 待处理 | REDSHIFT-EVENT-3602 | INFO | 集群 [集群名称] 上的节点计划在 [开始时间] 和 [结束时间] 之间替换。您的集群将不可访问。相应地做好计划。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 管理 | REDSHIFT-EVENT-3603 | 信息 | 执行还原操作来从快照 [快照名称] 创建集群 [集群名称] 因内部错误而失败。集群正处于不兼容的还原状态并且可以删除。尝试将快照恢复到具有不同配置的集群。 |
| 管理 | REDSHIFT-EVENT-3614 | 信息 | 计划操作 [计划操作名称] 已在 [时间，以 UTC 表示] 创建。第一次调用计划在 [时间，以 UTC 表示] 进行。 |
| 管理 | REDSHIFT-EVENT-3615 | 信息 | 计划操作 [计划操作名称] 已在 [时间，以 UTC 表示] 计划。 |
| 监控 | REDSHIFT-EVENT-3616 | 信息 | [时间，以 UTC 表示] 计划的操作 [计划操作名称] 已完成，状态为“SUCCEEDED”。 |
| 监控 | REDSHIFT-EVENT-3617 | 错误 | 计划操作 [计划操作名称] 在 [时间，以 UTC 表示] 因延迟而被跳过。 |
| 监控 | REDSHIFT-EVENT-3618 | 信息 | 集群 [集群名称] 暂停操作已于 [时间，以 UTC 表示] 开始。暂停已开始 |
| 监控 | REDSHIFT-EVENT-3619 | INFO | Amazon Redshift 集群 [集群名称] 已于 [UTC 时间] 被成功暂停。 |
| 管理 | REDSHIFT-EVENT-3626 | 信息 | 计划操作 [计划操作名称] 已在 [时间，以 UTC 表示] 修改。第一次调用计划在 [时间，以 UTC 表示] 进行。 |
| 管理 | REDSHIFT-EVENT-3627 | 信息 | 计划操作 [计划操作名称] 已在 [时间，以 UTC 表示] 删除。 |
| 监控 | REDSHIFT-EVENT-3628 | 错误 | [时间，以 UTC 表示] 计划的操作 [计划操作名称] 已完成，状态为“FAILED”。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 监控 | REDSHIFT-EVENT-3629 | INFO | 为集群 [集群名称] 启动了内部失效转移请求。 |
| 监控 | REDSHIFT-EVENT-3630 | INFO | Amazon Redshift 已成功将 Amazon Redshift 集群 [集群名称] 从 [可用区] 重新放置到 [可用区]，以便进行恢复。 |
| 管理 | REDSHIFT-EVENT-3631 | INFO | Amazon Redshift [集群名称] 收到了您的重新放置请求。当可用区重新定位完成时，Amazon Redshift 会发送一个事件通知。 |
| 监控 | REDSHIFT-EVENT-3632 | INFO | Amazon Redshift 集群 [集群名称] 已成功地从 [可用区] 重新放置到 [可用区]。您现在可以使用集群。 |
| 监控 | REDSHIFT-EVENT-3658 | 错误 | 针对 Redshift 集群 [集群 ID]，EC2-Classic 到 EC2-VPC 的迁移失败。 |
| 监控 | REDSHIFT-EVENT-3659 | 信息 | 针对 Redshift 集群 [集群 ID]，EC2-Classic 到 EC2-VPC 的迁移成功。 |
| 监控 | REDSHIFT-EVENT-3660 | 信息 | 集群正处于硬件故障状态。请删除 EC2-Classic 集群并从在 [UTC 时间] 创建的最新快照 [快照名称] 还原到 EC2-VPC 集群。 |
| 管理 | REDSHIFT-EVENT-3666 | INFO | Amazon Redshift 多可用区集群 [集群名称] 已于 [UTC 时间] 检测到故障并触发了自动恢复。 |
| 管理 | REDSHIFT-EVENT-3667 | INFO | Amazon Redshift 多可用区集群 [集群名称] 已于 [UTC 时间] 成功恢复，可在 [第一个可用区] 中使用。另一个可用区的辅助计算不久也将可用。 |
| 监控 | REDSHIFT-EVENT-3668 | ERROR | Amazon Redshift 多可用区集群 [集群名称] 未能于 [UTC 时间] 恢复。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 管理 | REDSHIFT-EVENT-3669 | INFO | Amazon Redshift 多可用区集群 [集群名称] 已于 [UTC 时间] 成功恢复，可与 [第一个可用区] 和 [第二个可用区] 的计算资源一起使用。 |
| 管理 | REDSHIFT-EVENT-3670 | INFO | Amazon Redshift 集群 [集群名称] 已于 [UTC 时间] 完成维护，可与 [第一个可用区] 中的计算资源一起使用。另一个可用区的辅助计算不久也将可用。 |
| 管理 | REDSHIFT-EVENT-3671 | INFO | Amazon Redshift 集群 [集群名称] 已于 [UTC 时间] 完成大小调整，可在 [第一个可用区] 中使用。另一个可用区的辅助计算不久也将可用。 |
| 管理 | REDSHIFT-EVENT-3672 | INFO | Amazon Redshift 多可用区集群 [集群名称] 于 [UTC 时间] 在 [第二个可用区] 中检测到故障并触发了自动恢复。 |
| 管理 | REDSHIFT-EVENT-3673 | INFO | 为 Amazon Redshift 集群 [集群名称] 启用多可用区的操作已于 [UTC 时间] 开始。 |
| 管理 | REDSHIFT-EVENT-3674 | INFO | 为 Amazon Redshift 集群 [集群名称] 启用多可用区的操作已于 [UTC 时间] 成功完成。 |
| 监控 | REDSHIFT-EVENT-3675 | ERROR | 为 Amazon Redshift 集群 [集群名称] 启用多可用区的操作已于 [UTC 时间] 失败。 |
| 管理 | REDSHIFT-EVENT-3676 | INFO | 为 Amazon Redshift 多可用区集群 [集群名称] 禁用多可用区的操作已于 [UTC 时间] 开始。 |
| 管理 | REDSHIFT-EVENT-3677 | INFO | 为 Amazon Redshift 集群 [集群名称] 禁用多可用区的操作已于 [UTC 时间] 成功完成。 |
| 监控 | REDSHIFT-EVENT-3678 | ERROR | 为 Amazon Redshift 集群 [集群名称] 禁用多可用区的操作已于 [UTC 时间] 失败。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 配置 | REDSHIFT-EVENT-3679 | INFO | 已成功修改 Amazon Redshift 集群 [集群名称] 的端口。 |
| 配置 | REDSHIFT-EVENT-3680 | ERROR | Amazon Redshift 无法创建集群 [集群名称]，因为无法访问此操作所需的服务相关角色 (SLR)。从 Amazon Redshift 控制台重试创建此集群。Amazon Redshift 将自动创建 SLR。 |
| 监控 | REDSHIFT-EVENT-3684 | ERROR | 您的 Amazon S3 桶 [桶名称] 已使用未知或无法访问的 AWS KMS 密钥加密。修改您的 Amazon S3 桶的加密。 |
| 管理 | REDSHIFT-EVENT-3685 | ERROR | 集群 [集群名称] 的还原操作失败，因为它没有足够的可用磁盘空间。正在回滚操作。尝试恢复到具有不同配置的集群。 |
| 管理 | REDSHIFT-EVENT-3686 | ERROR | 集群 [集群名称] 的大小调整操作失败，因为它没有足够的可用磁盘空间。正在回滚操作。尝试恢复到具有不同配置的集群。 |
| 管理 | REDSHIFT-EVENT-3687 | INFO | 已于 [UTC 时间] 完成所请求的 Amazon Redshift 集群 [集群名称] 的调整大小操作。 |
| 安全性 | REDSHIFT-EVENT-3688 | ERROR | Amazon Redshift 集群 [集群名称] 的加密密钥轮换无法完成。 |
| 数据共享 | REDSHIFT-EVENT-3689 | INFO | 将 Amazon Redshift 命名空间 <集群名称> 注册到 Glue Data Catalog 账户 <账户 id> 的操作已于 <UTC 时间> 开始。 |
| 数据共享 | REDSHIFT-EVENT-3690 | INFO | 将 Amazon Redshift 命名空间 <集群名称> 注册到 Glue Data Catalog 账户 <账户 id> 的操作已于 <UTC 时间> 成功完成。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 数据共享 | REDSHIFT-EVENT-3691 | INFO | 将 Amazon Redshift 命名空间 <集群名称> 注册到 Glue Data Catalog 账户 <账户 id> 的操作已于 <UTC 时间> 失败。 |
| 数据共享 | REDSHIFT-EVENT-3692 | INFO | 从 Glue Data Catalog 账户 <账户 id> 注销 Amazon Redshift 命名空间 <集群名称> 的操作已于 <UTC 时间> 开始。 |
| 数据共享 | REDSHIFT-EVENT-3693 | INFO | 从 Glue Data Catalog 账户 <账户 id> 注销 Amazon Redshift 命名空间 <集群名称> 的操作已于 <UTC 时间> 成功完成。 |
| 数据共享 | REDSHIFT-EVENT-3694 | INFO | 从 Glue Data Catalog 账户 <账户 id> 注销 Amazon Redshift 命名空间 <集群名称> 的操作已于 <UTC 时间> 失败。 |
| 安全性 | REDSHIFT-EVENT-4000 | 信息 | 您的 Amazon Redshift 集群 [集群名称] 的管理员凭证已于 [时间] 更新。 |
| 安全性 | REDSHIFT-EVENT-4001 | 信息 | 已于 [时间] 修改安全组 [安全组名称]。这些更改将自动针对所有关联的集群进行。 |
| 安全性 | REDSHIFT-EVENT-4005 | ERROR | 集群 [集群名称] 具有无效的 AWS KMS key 权限。集群将进入“INACCESSIBLE_KMS_KEY”状态，您将无法再对其进行访问。 |
| 安全性 | REDSHIFT-EVENT-4006 | INFO | 由于 AWS KMS key权限已恢复，集群 [集群名称] 现已恢复到其之前的状态。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 安全性 | REDSHIFT-EVENT-4500 | ERROR | 找不到您为 Amazon Redshift 集群 [集群名称] 提供的一个或多个安全组。如果有任何有效的安全组，则会将其应用于集群。未应用任何无效的安全组。如果您正在修改现有集群，但没有任何安全组有效，则该集群会保留其原始安全组。如果您正在创建新集群，但没有任何安全组有效，则集群创建失败。无论哪种情况，都请使用有效的安全组重试请求。有关管理 Amazon Redshift 中的安全组的更多信息，请转至《Amazon Redshift 管理指南》中的 VPC 安全组。 |
| 安全性 | REDSHIFT-EVENT-4501 | 错误 | 找不到在集群安全组 [集群安全组名称] 中指定的安全组 [安全组名称]。无法完成授权。 |
| 安全性 | REDSHIFT-EVENT-4502 | 错误 | Amazon Redshift 集群 [集群名称] 的管理员凭证在 [时间] 因并发活动更新失败。允许当前工作负载完成或减少活动工作量，然后重试该操作。 |
| 安全性 | REDSHIFT-EVENT-4503 | ERROR | Amazon Redshift 无法访问您集群 [集群名称] 的密钥。 |
| 安全性 | REDSHIFT-EVENT-4504 | ERROR | Amazon Redshift 无法访问用于加密您集群 [集群名称] 的管理员凭证密钥的 KMS 密钥 [KMS 密钥]。 |
| 安全性 | REDSHIFT-EVENT-4505 | ERROR | Amazon Redshift 无法轮换您集群 [集群名称] 的密钥，因为该集群正在进行操作。 |
| 安全性 | REDSHIFT-EVENT-4506 | ERROR | 您的 Amazon Redshift 集群 [集群名称] 已暂停。Amazon Redshift 无法轮换已暂停集群的密钥。 |

参数组源类型的类别和事件

下表显示的是参数组为源类型时的事件类型和事件列表。

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 配置 | REDSHIFT-EVENT-1002 | 信息 | 参数 [参数名称] 已于 [时间] 从 [值] 更新为 [值]。 |
| 配置 | REDSHIFT-EVENT-1003 | 信息 | 已创建集群参数组 [组名称]。 |
| 配置 | REDSHIFT-EVENT-1004 | 信息 | 已删除集群参数组 [组名称]。 |
| 配置 | REDSHIFT-EVENT-1005 | 信息 | 已于 [时间] 更新集群参数组 [名称]。如果只更改了动态参数，则现在正在修改关联的集群。如果您更改了静态参数，则在重新启动关联集群时将应用所有更新（包括动态参数）。 |

安全组源类型的类别和事件

下表显示了安全组为源类型时的事件类别和事件列表。

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 安全性 | REDSHIFT-EVENT-4002 | 信息 | 已创建集群安全组 [组名称]。 |
| 安全性 | REDSHIFT-EVENT-4003 | 信息 | 已创建集群安全组 [组名称]。 |
| 安全性 | REDSHIFT-EVENT-4004 | 信息 | 集群安全组 [组名称] 已于 [时间] 更改。更改将自动应用于所有关联的集群。 |

## 快照源类型的类别和事件

下表显示了快照为源类型时的事件类别和事件列表。

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| 管理 | REDSHIFT-EVENT-2009 | 信息 | Amazon Redshift 集群 [集群名称] 的用户快照 [快照名称] 已于 [时间] 开始拍摄。要监控快照制作进度，请访问AWS 管理控制台。 |
| 管理 | REDSHIFT-EVENT-2010 | 信息 | 您的 Amazon Redshift 集群 [集群名称] 的用户快照 [快照名称] 已于 [时间] 取消拍摄。 |
| 管理 | REDSHIFT-EVENT-2011 | 信息 | 您的 Amazon Redshift 集群 [集群名称] 的用户快照 [快照名称] 已于 [时间] 被删除。 |
| 管理 | REDSHIFT-EVENT-2012 | 信息 | Amazon Redshift 集群 [集群名称] 的最终快照 [快照名称] 已于 [时间] 开始拍摄。 |
| 监控 | REDSHIFT-EVENT-3004 | 信息 | 您的 Amazon Redshift 集群 [集群名称] 的用户快照 [快照名称] 已于 [时间] 成功完成拍摄。 |
| 监控 | REDSHIFT-EVENT-3005 | 信息 | Amazon Redshift 集群 [名称] 的最终快照 [名称] 已于 [时间] 成功完成拍摄。 |
| 监控 | REDSHIFT-EVENT-3006 | 信息 | Amazon Redshift 集群 [集群名称] 的最终快照 [快照名称] 已于 [时间] 取消拍摄。 |
| 监控 | REDSHIFT-EVENT-3502 | 错误 | Amazon Redshift 集群 [集群名称] 的最终快照 [快照名称] 于 [时间] 拍摄失败。团队正在调查此问题。请访问AWS 管理控制台，以重试该操作。 |
| 监控 | REDSHIFT-EVENT-3503 | 错误 | 您的 Amazon Redshift 集群 [集群名称] 的用户快照 [快照名称] 于 [时间] 拍摄失败。 |

| Amazon Redshift 类别 | 事件 ID | 事件严重性 | 描述 |
| --- | --- | --- | --- |
| | | | 团队正在调查此问题。请访问AWS 管理控制台，以重试该操作。 |

# 使用 Amazon EventBridge 的 Amazon Redshift Serverless 事件通知

Amazon Redshift Serverless 使用 Amazon EventBridge 管理事件通知，以便随时了解数据仓库中的更改。Amazon EventBridge 是一种无服务器事件总线服务，让您可以轻松地将应用程序与来自各种源的数据相连接。在这种情况下，事件源是 Amazon Redshift。事件（在环境中监控到的变化）将从您的 Amazon Redshift 数据仓库自动发送到 EventBridge。事件将近乎实时地进行传输。

EventBridge 的功能包括为您编写事件规则提供一个环境，这些规则可以指定针对特定事件采取的操作。您还可以设置目标，这些目标是 EventBridge 可以向其发送事件的资源。目标可以包括 API 目标、Amazon CloudWatch 日志组等。有关规则的更多信息，请参阅 Amazon EventBridge 规则。有关目标的更多信息，请参阅 Amazon EventBridge 目标。

事件可以按严重性和类别分类。可使用以下筛选条件：

- 资源筛选：接收基于与事件关联的资源的消息。资源包括工作组、快照等。
- 时间窗口筛选：确定特定时间段内事件的范围。
- 类别筛选：接收指定类别中所有事件的事件通知。

下表包括 Amazon Redshift Serverless 事件以及其他元数据：

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| RateChange | REDSHIFT-SERVERLESS-EVENT-1001 | INFO | 工作组基础 RPU 更改已于 <time in UTC> 成功完成。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| RateChange | REDSHIFT-SERVERLESS-EVENT-1002 | ERROR | 工作组基础 RPU 更改已于 <time in UTC> 失败。 |
| 监控 | REDSHIFT-SERVERLESS-EVENT-1003 | INFO | 该软件已于 <time in UTC> 在您的 Amazon Redshift 数据仓库 <endpoint name> 上更新。 |
| 配置 | REDSHIFT-SERVERLESS-EVENT-1011 | ERROR | Amazon Redshift Serverless 无法创建工作组 [工作组名称]，因为无法访问此操作所需的服务相关角色 (SLR)。尝试在 Amazon Redshift 控制台中再次创建此工作组。Amazon Redshift 将自动创建 SLR。 |
| 监控 | REDSHIFT-SERVERLESS-EVENT-1029 | ERROR | 工作组基本 RPU 更改未能在 [时间 (UTC)] 完成，因为可用的磁盘空间不足。请采用其他配置重试。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 监控 | REDSHIFT-SERVERLESS-EVENT-1500 | ERROR | 无法创建或更新工作组 <workgroup name>，因为您已超出账户的弹性 IP 地址限制。请删除未使用的弹性 IP 地址，或者使用 Amazon EC2 请求提高限制。 |
| 监控 | REDSHIFT-SERVERLESS-EVENT-1501 | ERROR | 子网 <subnet id> 没有可用的 IP 地址。这将阻止以下查询类型在工作组 <workgroup name> 中成功运行：EMR、联合查询、来自 Amazon EC2 的 COPY/UNLOAD。要解决该问题，请通过删除 ENI 来释放您的子网中的 IP。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 监控 | REDSHIFT-SERVERLESS-EVENT-1502 | ERROR | 子网 <subnet id> 没有可用的 IP 地址。这将阻止 Amazon EMR、Redshift 联合查询、Redshift COPY/UNLOAD、Redshift ML 查询类型在工作组 <workgroup name> 中成功运行。要解决该问题，请通过删除未使用的弹性网络接口 (ENI) 来释放您的子网中的 IP。 |
| 管理 | REDSHIFT-SERVERLESS-EVENT-1008 | INFO | 您的 Amazon Redshift 工作组 <workgroup name> 已创建，并且随时可用。 |
| 管理 | REDSHIFT-SERVERLESS-EVENT-1009 | INFO | 您的 Amazon Redshift 工作组 <workgroup name> 已于 <time in UTC> 删除。 |
| 监控 | REDSHIFT-SERVERLESS-EVENT-1000 | INFO | 快照 <snapshot name> 已于 <time in UTC> 成功完成。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 管理 | REDSHIFT-SERVERLESS-EVENT-1004 | INFO | 从命名空间 <namespace name> 上的快照还原已于 <time in UTC> 成功完成。 |
| 管理 | REDSHIFT-SERVERLESS-EVENT-1005 | ERROR | 从命名空间 <namespace name> 上的快照还原已于 <time in UTC> 失败。 |
| 管理 | REDSHIFT-SERVERLESS-EVENT-1006 | INFO | 从命名空间 <namespace name> 上的恢复点还原已于 <time in UTC> 成功完成。 |
| 管理 | REDSHIFT-SERVERLESS-EVENT-1007 | INFO | 从命名空间 <namespace name> 上的恢复点还原已于 <time in UTC> 失败。 |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1012 | ERROR | Amazon Redshift 无法访问您的命名空间 <命名空间名称> 的密钥。 |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1013 | ERROR | Amazon Redshift 无法访问用于加密您命名空间 <命名空间名称> 的管理员凭证密钥的 KMS 密钥。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1014 | ERROR | Amazon Redshift 无法轮换您的命名空间 <命名空间名称> 的密钥，因为工作组正在进行操作。 |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1015 | ERROR | 您的命名空间 <命名空间名称> 没有附加工作组。Amazon Redshift 只能轮换附加了工作组的命名空间的密钥。 |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1016 | INFO | 您的命名空间 <命名空间名称> 已于 <UTC 时间> 更新了管理员凭证。 |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1030 | INFO | 将 Amazon Redshift 命名空间 <命名空间名称> 注册到 Glue Data Catalog 账户 <账户 id> 的操作已于 <UTC 时间> 开始。 |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1031 | INFO | 将 Amazon Redshift 命名空间 <命名空间名称> 注册到 Glue Data Catalog 账户 <账户 id> 的操作已于 <UTC 时间> 成功完成。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1032 | INFO | 将 Amazon Redshift 命名空间 <命名空间名称> 注册到 Glue Data Catalog 账户 <账户 id> 的操作已于 <UTC 时间> 失败。 |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1033 | INFO | 从 Glue Data Catalog 账户 <账户 id> 注销 Amazon Redshift 命名空间 <命名空间名称> 的操作已于 <UTC 时间> 开始。 |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1034 | INFO | 从 Glue Data Catalog 账户 <账户 id> 注销 Amazon Redshift 命名空间 <命名空间名称> 的操作已于 <UTC 时间> 成功完成。 |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1035 | INFO | 从 Glue Data Catalog 账户 <账户 id> 注销 Amazon Redshift 命名空间 <命名空间名称> 的操作已于 <UTC 时间> 失败。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 管理 | REDSHIFT-SERVERLESS-EVENT-1036 | ERROR | 客户发起的跟踪更新在您的 Redshift Serverless 工作组 <工作组名称> 上失败。将工作组返回到其原始跟踪。 |
| 管理 | REDSHIFT-SERVERLESS-EVENT-1037 | ERROR | 跟踪更新在您的 Redshift Serverless 工作组 <工作组名称> 上失败。您的 Amazon Redshift Serverless 工作组的跟踪更新失败，因为该工作组当时正忙。因此，工作组已回到其原始跟踪。要重试跟踪更新，请等到工作组上活动较低的时间，然后再次尝试更新跟踪。 |
| 管理 | REDSHIFT-SERVERLESS-EVENT-1038 | INFO | 您的 Amazon Redshift 工作组 <工作组名称> 跟踪已修改。跟踪更改已完成。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 命名空间 | REDSHIFT-SERVERLESS-EVENT-1039 | ERROR | 命名空间 <命名空间名称> 具有无效的 AWS KMS key权限。命名空间将进入“INACCESSIBLE_KMS_KEY”状态，您将无法再对其进行访问。 |
| 命名空间 | REDSHIFT-SERVERLESS-EVENT-1040 | INFO | 由于 AWS KMS key 权限已恢复，命名空间 <命名空间名称> 现已恢复到其之前的状态。 |
| 命名空间 | REDSHIFT-SERVERLESS-EVENT-1045 | INFO | 由于无服务器还原，Redshift Serverless 命名空间 <namespace name> 上的数据共享 <datashare name> 已被删除。 |
| 命名空间 | REDSHIFT-SERVERLESS-EVENT-1046 | INFO | 由于无服务器还原，Redshift Serverless 命名空间 <namespace name> 上数据共享 <datashare name> 的使用者 <Consumer principle id> 已被撤销。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 命名空间 | REDSHIFT-SERVERLESS-EVENT-1047 | INFO | 由于无服务器还原，Redshift Serverless 命名空间 <namespace name> 上的数据共享 <datashare name> 的公共可访问性发生了变化。 |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1048 | ERROR | 由于内部问题，您选择退出命名空间 <namespace name> 的 Redshift Secrets Manager 集成的请求失败。手动重试选择退出操作以解决此问题。 |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1049 | ERROR | 由于内部问题，您选择加入命名空间 <namespace name> 的 Redshift Secrets Manager 集成的请求无法成功完成。手动轮换与 Redshift Serverless 命名空间关联的密钥以解决此问题。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 安全性 | REDSHIFT-SERVERLESS-EVENT-1050 | ERROR | 由于内部问题，您在命名空间 <namespace name> 上重置管理员凭证的请求已失败。重试该操作以重置管理员用户凭证。 |

# 使用 Amazon EventBridge 发送零 ETL 集成事件通知

零 ETL 集成使用 Amazon EventBridge 管理事件通知，以便及时了解集成中的更改。Amazon EventBridge 是一种无服务器事件总线服务，让您可以轻松地将应用程序与来自各种源的数据相连接。在这种情况下，事件源是 Amazon Redshift。事件（在环境中监控到的变化）将从您的 Amazon Redshift 数据仓库自动发送到 EventBridge。事件将近乎实时地进行传输。

EventBridge 提供了一个环境，供您编写事件规则，这些规则可以指定针对特定事件采取的操作。您还可以设置目标，这些目标是 EventBridge 可以向其发送事件的资源。目标可以包括 API 目标、Amazon CloudWatch 日志组等。有关规则的更多信息，请参阅 Amazon EventBridge 规则。有关目标的更多信息，请参阅 Amazon EventBridge 目标。

事件可以按严重性和类别分类。可使用以下筛选条件：

- 资源筛选 – 接收基于与事件关联的资源的消息。资源包括工作组或快照。
- 时间窗口筛选：确定特定时间段内事件的范围。
- 类别筛选：接收指定类别中所有事件的事件通知。

下表包括零 ETL 集成事件以及其他元数据：

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0000 | INFO | 零 ETL 集成 <集成名称> 已于 <UTC 时间> 创建并变为 ACTIVE 状态。 |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0001 | INFO | 零 ETL 集成 <集成名称> 已于 <UTC 时间> 删除。 |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0002 | INFO | 零 ETL 集成 <集成名称> 已于 <UTC 时间> 启动删除。 |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0003 | INFO | 零 ETL 集成 <集成名称> 正在将事务数据同步到 Amazon Redshift 数据仓库。 |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0004 | WARNING | Amazon Redshift 无法复制该表，因为该表缺少主键。向源表添加主键，该表将自动重新同步。 |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0005 | WARNING | Amazon Redshift 无法复制该表，因为一个或多个列使用了不受支持的数据类型。修改集成以在筛选条件中排除该表，或者从源表中删除列，然后运行“ALTER DATABASE <Redshift 数据库名 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| | | | 称> INTEGRATION REFRESH TABLE <架构名称>.<表名>”来同步该表。 |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0006 | ERROR | 无法创建集成。删除并重新创建集成。如果错误仍然存在，请联系 AWS Support。 |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0007 | ERROR | 由于内部错误，无法加载数据。删除并重新创建集成。如果错误仍然存在，请联系 AWS Support。 |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0008 | ERROR | 授权失败，因为已从源数据库集群撤销权限。如果您使用客户自主管理型密钥（CMK）来加密集成，请检查 KMS 密钥权限。然后，删除并重新创建集成。 |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0009 | ERROR | 无法向 Amazon Redshift 发送数据，因为表和架构的数量超过了 Amazon Redshift 的限制。删除并重新创建集成。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0012 | ERROR | 在目标无服务器命名空间上调用了从恢复点执行还原的操作。删除并重新创建集成。 |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0013 | INFO | 零 ETL 集成 <集成名称> 现处于 ACTIVE 状态。 |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0014 | ERROR | 集成 <集成名称> 处于 FAILED 状态，因存在内部错误，我们无法修改集成。删除并重新创建集成。如果错误仍然存在，请联系 AWS Support。 |
| 操作 | REDSHIFT-INTEGRATION-EVENT-0016 | INFO | 零 ETL 集成 <集成名称> 正在处理修改请求。 |
| 操作 | REDSHIFT-INTEGRATION-EVENT-0017 | INFO | 您对零 ETL 集成 <集成名称> 的修改已应用。 |
| 操作 | REDSHIFT-INTEGRATION-EVENT-0018 | WARNING | 正在暂停目标 Amazon Redshift 集群。等待集群暂停，然后恢复集群以继续流式传输数据。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 操作 | REDSHIFT-INTEGRATION-EVENT-0019 | WARNING | 暂停目标 Amazon Redshift 集群。必须恢复集群才能继续流式传输数据。 |
| 操作 | REDSHIFT-INTEGRATION-EVENT-0020 | WARNING | 正在恢复目标 Amazon Redshift 集群。等待集群处于活动状态，以继续流式传输数据。 |
| 监控 | REDSHIFT-INTEGRATION-EVENT-0025 | ERROR | 已在目标无服务器命名空间上调用了无服务器还原，并且您的集成已停用，因为它在还原的快照中不可用。还原在集成可用时创建的快照，或者删除该集成。 |
| 配置 | REDSHIFT-INTEGRATION-EVENT-1000 | ERROR | 源 Aurora 数据库集群中的一个或多个参数配置错误。修复参数组并重启集群以应用更改，然后重新创建集成。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 配置 | REDSHIFT-INTEGRATION-EVENT-1001 | ERROR | 集成失败，因为 enable_case_sensitive_identifier 参数的值不正确。对于源 Aurora 数据库集群，将该值设置为 true，然后删除并重新创建该集成。 |
| 配置 | REDSHIFT-INTEGRATION-EVENT-1002 | ERROR | 集成失败，因为 cdc_insert_enabled 参数的值不正确。为源 Aurora 数据库集群将该值设置为 true，然后删除并重新创建集成。 |
| 配置 | REDSHIFT-INTEGRATION-EVENT-1003 | ERROR | 源数据库集群参数组中的 binlog_format 参数必须设置为 ROW。修复参数组并重启集群以应用更改，然后重新创建集成。 |
| 配置 | REDSHIFT-INTEGRATION-EVENT-1004 | ERROR | 无法加载数据，因为已启用 binlog_transaction_compression 集群参数。将该参数值设置为 OFF 并重启写入器实例以应用更改，然后重新创建集成。 |

| Amazon Redshift 类别 | 外部事件 ID | 事件严重性 | 消息描述 |
| --- | --- | --- | --- |
| 配置 | REDSHIFT-INTEGRATION-EVENT-1005 | ERROR | 无法加载数据，因为 binlog_row_value_options 集群参数设置为 PARTIAL_JSON，不支持此参数值。修复参数组并重启写入器实例以应用更改，然后重新创建集成。 |
| 配置 | REDSHIFT-INTEGRATION-EVENT-1006 | WARNING | 无法解析集成筛选条件。修复筛选条件语法。 |

# Amazon Redshift 资源中的配额和限制

Amazon Redshift 具有的配额限制了每个 AWS 区域 AWS 账户中的多种资源的使用。每个配额都有一个默认值，并且部分配额可以调整。

## Amazon Redshift 对象的配额

Amazon Redshift 具有的配额限制了多种对象类型的使用。每个配额都有一个默认值。

| 配额名称 | AWS 默认值 | 可调整 | 描述 |
| --- | --- | --- | --- |
| AWS您可以向其授予为每个快照还原快照的权限的 账户 | 20 | 否 | 您可以向其授予为每个快照还原快照的权限的 AWS 账户的最大数量。 |
| AWS您可以向其授予为每个 密钥还原快照的权限的账户AWS KMS key | 100 | 否 | 您可以向其授予为每个 KMS 密钥还原快照的权限的 AWS 账户的最大数量。例如，如果您有 10 个快照，它们使用了一个 KMS 密钥加密，那么您可以授权 10 个 AWS 账户来还原每个快照，或者是其他组合：总共 100 账户以及每个快照不超过 20 个账户。 |
| 适用于 Amazon Redshift 的集群 IAM 角色，可用于访问其他 AWS 服务 | 50[1] | 否 | IAM 角色的最大数量，可将这些 IAM 角色与集群关联以授权 Amazon Redshift 访问拥有集群和 IAM 角色的用户的其他 AWS 服务。<br>[1] 在以下 AWS 区域中，配额为 10：us-iso-east-1、us-iso-west-1、us-isob-east-1。 |
| 所有用户定义的手动 | 50 | 否 | 由手动工作负载管理定义的所有用户定义的队列的最大查询槽数。 |

| 配额名称 | AWS 默认值 | 可调整 | 描述 |
| --- | --- | --- | --- |
| WLM 队列的并发级别（查询槽） | | | |
| 并发扩展集群 | 10 | 是 | 并发扩展集群的最大数量。 |
| 集群中的 DC2 节点 | 128 | 是 | 可以分配给集群的 DC2 节点的最大数量。有关每个节点类型的节点限制的更多信息，请参阅Amazon Redshift 中的集群和节点。 |
| 事件订阅 | 20 | 是 | 此账户在当前 AWS 区域中的事件订阅的最大数量。 |
| 节点 | 200 | 是 | 此账户在当前 AWS 区域所有数据库实例中的节点的最大数量。 |
| 参数组 | 20 | 否 | 此账户在当前 AWS 区域中的参数组的最大数量。 |
| 集群中的 RG 节点 | 128 | 是 | 可以分配给集群的 RG 节点的最大数量。有关每个节点类型的节点限制的更多信息，请参阅Amazon Redshift 中的集群和节点。 |
| 集群中的 RA3 节点 | 128 | 是 | 可以分配给集群的 RA3 节点的最大数量。有关每个节点类型的节点限制的更多信息，请参阅Amazon Redshift 中的集群和节点。 |
| 连接到集群的 Redshift 托管 VPC 端点 | 30 | 是 | 可以连接到集群的 Redshift 托管 VPC 端点的最大数量。有关 Redshift 托管 VPC 端点的更多信息，请参阅Redshift 托管的 VPC 端点。 |

| 配额名称 | AWS 默认值 | 可调整 | 描述 |
| --- | --- | --- | --- |
| 通过 RedShift 托管的 VPC 端点访问集群的被授权者 | 10 | 是 | 集群拥有者可以授权为集群创建 RedShift 托管 VPC 端点的最大被授权者数量。有关 Redshift 托管 VPC 端点的更多信息，请参阅Redshift 托管的 VPC 端点。 |
| 每个授权的 Redshift 托管 VPC 端点 | 10 | 是 | 可为每个授权创建的 Redshift 托管 VPC 端点的最大数量。有关 Redshift 托管 VPC 端点的更多信息，请参阅Redshift 托管的 VPC 端点。 |
| 预留节点 | 200 | 是 | 此账户在当前 AWS 区域中的预留节点的最大数量。 |
| 每个集群的每个数据库中的架构 | 9900 | 否 | 可以在每个集群的每个数据库中创建的 schema 的最大数量。但是，`pg_temp_*` schema 不计入此配额。 |
| 安全组 | 20 | 是 | 此账户在当前 AWS 区域中的安全组的最大数量。 |
| 通过 COPY 加载时的单一行大小 | 64 | 否 | 使用 COPY 命令加载时的单一行的最大大小（以 MB 为单位）。这是 COPY 命令可以扫描的单个未解析记录的最大大小；行中的单个 SUPER 对象独立限制为 16 MB。 |
| 快照 | 700 | 是 | 此账户在当前 AWS 区域中的用户快照的最大数量。 |
| 子网组 | 50 | 可以 | 此账户在当前 AWS 区域中的子网组的最大数量。 |
| 子网组中的子网 | 20 | 是 | 子网组的最大子网数。 |

| 配额名称 | AWS 默认值 | 可调整 | 描述 |
| --- | --- | --- | --- |
| 具有单节点集群的 rg.large 集群节点类型的表 | 9900 | 否 | 具有单节点集群的 rg.large 集群节点类型的最大表数量。此限制包括永久表、临时表、数据共享表和实体化视图。外部表被计为临时表。临时表包括用户定义的临时表以及查询处理或系统维护期间由 Amazon Redshift 创建的临时表。此限制中并不包括视图和系统表。 |
| 具有多节点集群的 rg.large 集群节点类型的表。 | 20000 | 否 | 具有多节点集群的 rg.large 节点类型的表的最大数量。此限制包括永久表、临时表、数据共享表和实体化视图。外部表被计为临时表。临时表包括用户定义的临时表以及查询处理或系统维护期间由 Amazon Redshift 创建的临时表。此限制中并不包括视图和系统表。 |
| rg.xlarge 集群节点类型的表。 | 20000 | 否 | rg.xlarge 集群节点类型的表的最大数量。此限制包括永久表、临时表、数据共享表和实体化视图。外部表被计为临时表。临时表包括用户定义的临时表以及查询处理或系统维护期间由 Amazon Redshift 创建的临时表。此限制中并不包括视图和系统表。 |
| rg.4xlarge 集群节点类型的表 | 200,000 | 否 | rg.4xlarge 集群节点类型的表的最大数量。此限制包括永久表、临时表、数据共享表和实体化视图。外部表被计为临时表。临时表包括用户定义的临时表以及查询处理或系统维护期间由 Amazon Redshift 创建的临时表。此限制中并不包括视图和系统表。 |
| rg.12xlarge 集群节点类型的表 | 200,000 | 否 | rg.12xlarge 集群节点类型的表的最大数量。此限制包括永久表、临时表、数据共享表和实体化视图。外部表被计为临时表。临时表包括用户定义的临时表以及查询处理或系统维护期间由 Amazon Redshift 创建的临时表。此限制中并不包括视图和系统表。 |
| ra3.large 集群节点类型的表 | 9900 | 否 | ra3.large 集群节点类型的表的最大数量。此限制包括永久表、临时表、数据共享表和实体化视图。外部表被计为临时表。临时表包括用户定义的临时表以及查询处理或系统维护期间由 Amazon Redshift 创建的临时表。此限制中并不包括视图和系统表。 |

| 配额名称 | AWS 默认值 | 可调整 | 描述 |
| --- | --- | --- | --- |
| 具有单节点集群的 ra3.xlplus 集群节点类型的表。 | 9900 | 否 | 具有单节点集群的 ra3.xlplus 集群节点类型的最大表数量。此限制包括永久表、临时表、数据共享表和实体化视图。外部表被计为临时表。临时表包括用户定义的临时表以及查询处理或系统维护期间由 Amazon Redshift 创建的临时表。此限制中并不包括视图和系统表。 |
| 具有多节点集群的 ra3.xlplus 集群节点类型的表。 | 20000 | 否 | 具有多节点集群的 ra3.xlplus 集群节点类型的最大表数量。此限制包括永久表、临时表、数据共享表和实体化视图。外部表被计为临时表。临时表包括用户定义的临时表以及查询处理或系统维护期间由 Amazon Redshift 创建的临时表。此限制中并不包括视图和系统表。 |
| ra3.4xlarge 集群节点类型的表 | 200,000 | 否 | ra3.4xlarge 集群节点类型的表的最大数量。此限制包括永久表、临时表、数据共享表和实体化视图。外部表被计为临时表。临时表包括用户定义的临时表以及查询处理或系统维护期间由 Amazon Redshift 创建的临时表。此限制中并不包括视图和系统表。 |
| ra3.16xlarge 集群节点类型的表 | 200,000 | 否 | ra3.16xlarge 集群节点类型的表的最大数量。此限制包括永久表、临时表、数据共享表和实体化视图。外部表被计为临时表。临时表包括用户定义的临时表以及查询处理或系统维护期间由 Amazon Redshift 创建的临时表。此限制中并不包括视图和系统表。 |
| dc2.large 集群节点类型的表 | 9900 | 否 | dc2.large 集群节点类型的表的最大数量。此限制包括永久表、临时表、数据共享表和实体化视图。外部表被计为临时表。临时表包括用户定义的临时表以及查询处理或系统维护期间由 Amazon Redshift 创建的临时表。此限制中并不包括视图和系统表。 |

| 配额名称 | AWS 默认值 | 可调整 | 描述 |
| --- | --- | --- | --- |
| dc2.8xlarge 集群节点类型的表 | 200,000 | 否 | dc2.8xlarge 集群节点类型的表的最大数量。此限制包括永久表、临时表、数据共享表和实体化视图。外部表被计为临时表。临时表包括用户定义的临时表以及查询处理或系统维护期间由 Amazon Redshift 创建的临时表。此限制中并不包括视图和系统表。 |
| 数据库数 | 60 | 否 | Amazon Redshift 集群中允许的最大数据库计数。这不包括从数据共享创建的数据库。 |
| 空闲或非活动会话超时 | 4 小时 | 否 | 此设置适用于集群。有关为用户设置空闲会话超时值的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的更改用户。用户设置优先于集群设置。 |
| 空闲事务的超时 | 6 小时 | 否 | 在 Amazon Redshift 结束与事务关联的会话之前，未结事务的最长不活动时间。此设置优先于任何用户定义的空闲超时设置。它适用于集群。 |
| 数据库中存储的程序 | 10000 | 否 | 存储程序的最大数量。请参阅存储程序支持的限制和区别了解更多限制。 |
| RG 节点的最大连接数 | 2000 | 否 | 与 RG 集群的最大连接数。这是一项硬性限制，适用于所有 RG 节点类型，无法提高。 |
| RA3 节点的最大连接数 | 2000 | 否 | 与 RA3 集群的最大连接数。这是一项硬性限制，适用于所有 RA3 节点类型，无法提高。 |
| DS2 节点的最大连接数 | 变化 | 否 | 与 dc2.large 集群的最大连接数为 500。与 dc2.8xlarge 集群的最大连接数为 2000。 |
| 集群中 Amazon Redshift 角色的数量 | 1000 | 是 | 您可以在每个集群中创建的 Amazon Redshift 角色的最大数量。有关基于角色的访问控制 (RBAC) 角色的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 基于角色的访问控制 (RBAC)。 |

# Amazon Redshift Serverless 对象的配额

Amazon Redshift 具有的配额限制了在 Amazon Redshift Serverless 命名空间中使用多种对象类型。每个配额都有一个默认值。

| 配额名称 | AWS 默认值 | 可调整 | 说明 |
| --- | --- | --- | --- |
| 数据库数 | 100 | 否 | Amazon Redshift Serverless 命名空间中允许的最大数据库计数。这不包括从数据共享创建的数据库。 |
| 架构的数量 | 9900 | 否 | Amazon Redshift Serverless 命名空间中支持的最大架构计数。 |
| 表的数量 | 200,000 | 否 | Amazon Redshift Serverless 命名空间中支持的最大表计数。 |
| 使用 4 个 Redshift 处理器（RPU）为数据仓库提供 Redshift 托管式存储单元 | 最大为 32 TB | 否 | 具有 4 个 RPU 的数据仓库支持的最大 RMS 存储空间。在 32 TB 限制内，Amazon Redshift 在内部使用几百 MB 的存储空间。 |
| 使用 8、16 或 24 个 Redshift 处理器（RPU）为数据仓库提供 Redshift 托管式存储单元 | 最大为 128 TB | 否 | 具有 8、16 或 24 个 RPU 的数据仓库支持的最大 RMS 存储空间。在 128 TB 限制内，Amazon Redshift 在内部使用几百 MB 的存储空间。 |

| 配额名称 | AWS 默认值 | 可调整 | 说明 |
| --- | --- | --- | --- |
| 空闲或非活动会话超时 | 1 小时 | 否 | 有关为用户设置空闲会话超时值的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的更改用户。用户设置优先。 |
| 运行查询的超时 | 86,399 秒（24 小时） | 否 | Amazon Redshift 结束正在运行的查询之前等待的最长时间。 |
| 空闲事务的超时 | 6 小时 | 否 | 在 Amazon Redshift Serverless 结束与事务关联的会话之前，未结事务的最长不活动时间。此设置优先于任何用户定义的空闲超时设置。 |
| 最大连接数 | 2000 | 否 | 允许连接到工作组的连接的最大数目。 |
| 工作组数 | 25 | 是 | 支持的工作组数。 |
| 命名空间数 | 25 | 是 | 支持的命名空间数。 |
| 工作组中 Amazon Redshift 角色的数量 | 1000 | 是 | 您可以在每个工作组中创建的 Amazon Redshift 角色的最大数量。有关基于角色的访问控制 (RBAC) 角色的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 基于角色的访问控制 (RBAC)。 |
| 每个 AWS 账户的基本 Redshift 处理器（RPU）数量 | 3200 个 RPU 或前六个月最大累计基本 RPU 的 1.5 倍（以较大值为准）。 | 是 | Redshift Serverless 基本 RPU 的 AWS 账户配额为 3200 个 RPU 或前六个月最大累计基本 RPU 的 1.5 倍（以较大值为准）。<br><br>. |

有关 Amazon Redshift Serverless 账单如何受超时配置影响的更多信息，请参阅Amazon Redshift Serverless 的计费。

# Amazon Redshift 数据 API 的配额

Amazon Redshift 具有配额，用于限制 Redshift 数据 API 用量。每个配额都有一个默认值。有关 Amazon Redshift 数据 API 的更多信息，请参阅使用 Amazon Redshift 数据 API。

| 配额名称 | AWS 默认值 | 可调整 | 说明 |
| --- | --- | --- | --- |
| BatchExecuteStatement API 的每秒事务数 (TPS) | 20 | 否 | 在不受限制的情况下，每秒可发出的操作请求的最大数目。 |
| CancelStatement API 的每秒事务数 (TPS) | 3 | 否 | 在不受限制的情况下，每秒可发出的操作请求的最大数目。 |
| DescribeStatement API 的每秒事务数 (TPS) | 100 | 否 | 在不受限制的情况下，每秒可发出的操作请求的最大数目。 |
| DescribeTable API 的每秒事务数 (TPS) | 3 | 否 | 在不受限制的情况下，每秒可发出的操作请求的最大数目。 |
| ExecuteStatement API 的每秒事务数 (TPS) | 30 | 否 | 在不受限制的情况下，每秒可发出的操作请求的最大数目。 |

| 配额名称 | AWS 默认值 | 可调整 | 说明 |
| --- | --- | --- | --- |
| GetStatementResult API 的每秒事务数 (TPS) | 20 | 否 | 在不受限制的情况下，每秒可发出的操作请求的最大数目。 |
| ListDatabases API 的每秒事务数 (TPS) | 3 | 否 | 在不受限制的情况下，每秒可发出的操作请求的最大数目。 |
| ListSchemas API 的每秒事务数 (TPS) | 3 | 否 | 在不受限制的情况下，每秒可发出的操作请求的最大数目。 |
| ListStatements API 的每秒事务数 (TPS) | 3 | 否 | 在不受限制的情况下，每秒可发出的操作请求的最大数目。 |
| ListTables API 的每秒事务数 (TPS) | 3 | 否 | 在不受限制的情况下，每秒可发出的操作请求的最大数目。 |

# 查询编辑器 v2 对象的配额

Amazon Redshift 具有的配额限制了 Amazon Redshift 查询编辑器 v2 中多种对象类型的使用。每个配额都有一个默认值。

| 配额名称 | AWS 默认值 | 可调整 | 描述 |
| --- | --- | --- | --- |
| 连接 | 500 | 是 | 您可以在当前区域的此账户中使用查询编辑器 v2 创建的最大连接数。 |
| 每个账户的活动主体数 | 50 | 可以 | 允许同时存在的最多主体数，这些主体可以在当前区域中在此账户中使用查询编辑器 v2。 |
| 保存的查询 | 2,500 | 是 | 您可以在当前区域的此账户中使用查询编辑器 v2 创建的最大已保存查询数。 |
| 查询版本 | 20 | 是 | 您可以在当前区域的此账户中使用查询编辑器 v2 创建的每个查询的最大版本数。 |
| 已保存图表 | 500 | 是 | 您可以在当前区域的此账户中使用查询编辑器 v2 创建的最大已保存图表数。 |
| 每个查询获取的数据大小 | 100 | 否 | 当前区域中此账户中查询编辑器 v2 每次查询获取的数据的最大大小（以兆字节为单位）。 |
| 每 100 行提取的数据大小 | 5 | 否 | 在此账户和当前区域中，查询编辑器 v2 在每个查询结果页（100 行）中提取的数据的最大大小（MB）。 |
| 最大并发连接数 | 3 | 否 | 每个用户的最大数据库连接数（包括隔离会话）。查询编辑器 v2 管理员可以在 Account settings（账户设置）中将此值设置为 1–10。如果您达到管理员设置的限制，请考虑在运行 SQL 时使用共享会话而不是隔离会话。有关 连接的更多信息，请参阅 打开查询编辑器 v2。有关设置限制的更多信息，请参阅账户设置。 |

# Amazon Redshift Spectrum 对象的配额和限制

Amazon Redshift Spectrum 具有以下配额和限制：

- 使用 AWS Glue Data Catalog 时每个AWS账户的最大数据库数。有关此值，请参阅《Amazon Web Services 一般参考》中的 AWS Glue 服务限额。
- 使用 AWS Glue Data Catalog 时每个数据库的最大表数。有关此值，请参阅《Amazon Web Services 一般参考》中的 AWS Glue 服务限额。
- 使用 AWS Glue Data Catalog 时每个表的最大分区数。有关此值，请参阅《Amazon Web Services 一般参考》中的 AWS Glue 服务限额。
- 使用 AWS Glue Data Catalog 时每个AWS账户的最大分区数。有关此值，请参阅《Amazon Web Services 一般参考》中的 AWS Glue 服务限额。
- 使用 AWS Glue Data Catalog 时外部表的最大列数（启用伪列时为 1,597，未启用伪列时为 1600）。
- 使用 AWS Glue Data Catalog 时 ION 或 JSON 文件中字符串值的最大大小为 16 KB。如果达到此限制，会截断字符串。
- 您可以使用单个 ALTER TABLE 语句添加最多 100 个分区。
- 所有 S3 数据必须与 Amazon Redshift 集群位于同一 AWS 区域。
- ION 和 JSON 中的时间戳必须使用 ISO8601 格式。
- 不支持外部压缩 ORC 文件。
- Text、OpenCSV 和 Regex SERDEs 不支持大于 '\177' 的八进制分隔符。
- 您必须在分区列上指定谓词以避免从所有分区读取。

  例如，以下谓词在列 `ship_dtm` 上进行筛选，但不会将筛选条件应用于分区列 `ship_yyyymm`：

  ```
  WHERE ship_dtm > '2018-04-01'.
  ```

  要跳过不需要的分区，您需要添加谓词 `WHERE ship_yyyymm = '201804'`。此谓词仅允许对分区 `\ship_yyyymm=201804\` 进行读操作。

这些限制不适用于 Apache Hive 元存储。

# 命名约束

下表介绍 Amazon Redshift 中的命名约束。

| | |
|---|---|
| 集群标识符 | • 集群标识符必须仅包含小写字符。<br>• |

| | |
|---|---|
| | 它必须包含 1–63 个字母数字字符或连字符。<br>• 它的第一个字符必须是字母。<br>• 它不能以连字符结束或包含两个连续连字符。<br>• 一个 AWS 账户内的所有集群必须拥有唯一的标识符。 |
| 数据库名称 | • 数据库名称必须包含 1-64 个字母数字字符。<br>• 必须仅由小写字母组成。<br>• 不能使用保留字。要查看保留关键字的列表，请参阅《Amazon Redshift 数据库开发人员指南》中的保留关键字。 |
| Redshift 托管的 VPC 端点的端点名称 | • 端点名必须包含 1—30 个字符。<br>• 有效字符为 A-Z、a-z、0-9 和连字符 (-)。<br>• 第一个字符必须是字母。<br>• 名称不能包含两个连续的连字符，也不能以连字符结束。 |
| 管理员用户名 | • 管理员用户名必须仅包含小写字符。<br>• 必须包含 1–128 个字母数字字符。<br>• 它的第一个字符必须是字母。<br>• 不能使用保留字。要查看保留关键字的列表，请参阅《Amazon Redshift 数据库开发人员指南》中的保留关键字。 |

| | |
|---|---|
| 管理员密码 | • 管理员密码必须包含 8–64 个字符。<br>• 至少必须包含一个大写字母。<br>• 至少必须包含一个小写字母。<br>• 它必须包含一个数字。<br>• 它可以使用带有 ASCII 代码 33–126 的任何 ASCII 字符，但 ' ( 单引号 ) 、 " ( 双引号 ) 、 \、 / 或 @ 除外。 |
| 参数组名称 | • 参数组名称必须包含 1–255 个字母数字字符或连字符。<br>• 它必须只由小写字母组成。<br>• 它的第一个字符必须是字母。<br>• 它不能以连字符结尾，也不能包含两个连续连字符。 |
| 集群安全组名称 | • 集群安全组名称必须包含不超过 255 个字母数字字符或连字符。<br>• 它必须只由小写字母组成。<br>• 它不得是 **`Default`**。<br>• 它在您的 AWS 账户创建的所有安全组中必须具有唯一性。 |

| | |
|---|---|
| 子网组名称 | • 子网组名称必须包含不超过 255 个字母数字字符或连字符。<br>• 它必须只由小写字母组成。<br>• 它不得是 **Default**。<br>• 它在您的 AWS 账户创建的所有子网组中必须具有唯一性。 |
| 集群快照标识符 | • 集群快照标识符必须包含不超过 255 个字母数字字符或连字符。<br>• 它必须只由小写字母组成。<br>• 它不得是 **Default**。<br>• 它在您的 AWS 账户创建的所有快照标识符中必须具有唯一性。 |

# 在 Amazon Redshift 中为资源添加标签

在 AWS 中，标签是用户定义的标记，由键-值对组成。Amazon Redshift 支持通过添加标签来提供资源的元数据概述，以及根据成本分配对您的账单报告进行分类。要将标签用于成本分配，您必须先在 AWS 账单与成本管理 服务中激活这些标签。有关设置标签并将其用于计费的更多信息，请参阅在自定义账单报告中使用成本分配标签和设置月度成本分配报告。

虽然标签对于 Amazon Redshift 中的资源来说并非必不可少，但它们却有助于提供背景信息。您可能想使用与成本中心、项目名称及该资源的其他相关信息有关的元数据来为资源添加标签。例如，假设您要跟踪属于测试环境和生产环境的资源。您可以创建名为 `environment` 的键并提供值 `test` 或 `production`，以确定在各环境中使用的资源。如果您在其他 AWS 服务中使用了标签或者具有针对您业务的标准类别，那么，我们建议您为 Amazon Redshift 中的资源创建相同的键值对以保持一致性。

在您调整集群以及将集群快照还原到同一区域后，资源的标签将得以保留。但是，如果您将快照复制到其他区域，则不会保留标签，因此您必须在新的区域重新创建标签。如果删除资源，则会删除所有关联的标签。

每个资源都有一个标签集，它是分配给该资源的一个或多个标签的集合。每个资源的每个标签集内最多有 50 个标签。您可以在创建资源时以及资源创建完成后添加标签。您可以向 Amazon Redshift 中的以下资源类型添加标签：

- CIDR/IP
- 集群
- 集群安全组
- 集群安全组传入规则
- Amazon EC2 安全组
- 硬件安全模块 (HSM) 连接
- HSM 客户端证书
- 参数组
- 快照
- 子网组
- 集成（零 ETL 集成）

要使用 Amazon Redshift 控制台中的标记，您的用户可以附加 AWS 托管式策略 `AmazonRedshiftFullAccess`。有关具有可附加到 Amazon Redshift 控制台用户的有限标记权限的

示例 IAM 策略，请参阅示例 7：允许用户使用 Amazon Redshift 控制台标记资源。有关标记的更多信息，请参阅什么是 AWS Resource Groups？。

## 标记要求

标签具有以下要求：

- 键不得以 aws: 作为前缀。
- 每个标签集中的各个键必须是独一无二的。
- 键的长度必须介于 1 到 128 个允许的字符之间。
- 值的长度必须介于 0 到 256 个允许的字符之间。
- 每个标签集中的值不需要是唯一的。
- 可以用作键和值的字符包括 Unicode 字符、数字、空格及以下符号：_ . : / = + - @。
- 键和值区分大小写。

## 管理资源标签

以下过程将为您演练如何使用资源的标签。

要管理您的 Amazon Redshift 资源上的标签

1. 登录 AWS 管理控制台，然后通过以下网址打开 Amazon Redshift 控制台：https://console.aws.amazon.com/redshiftv2/。
2. 在导航菜单上，选择 Configurations（配置），然后选择 Manage tags（管理标签）。
3. 输入您的资源选择，并选择要添加、修改或删除哪些标签。然后选择 Manage tags of the resources that you chose（管理您选择的资源的标签）。

   您可以标记的资源包括集群、参数组、子网组、HSM 客户端证书、HSM 连接和快照。
4. 在 Manage tags（管理标签）导航页面中，选择 Review and apply tag changes（查看并应用标签更改），然后选择 Apply（应用）以保存您的更改。

# AWS Backup 与 Amazon Redshift 集成

AWS Backup 是一项完全托管式服务，帮助您在云中以及在本地集中管理和自动执行各种 AWS 服务中的数据保护。

使用 AWS Backup for Amazon Redshift，您可以配置数据保护策略并在一个位置监控不同 Amazon Redshift 资源的活动。还可以在 Amazon Redshift 预置集群和无服务器命名空间上创建和存储快照。这使您可以自动执行和整合以前必须单独执行的备份任务，而无需任何手动流程。

> **Note**
>
> RG 或 RA3 预置集群和 Amazon Redshift Serverless 工作组不支持无备份表。在 RG 或 RA3 集群和无服务器工作组中标记为无备份的表将被视为永久表，在拍摄快照时将始终对其进行备份，并在从快照还原时还原该表。要避免无备份表产生快照成本，请在拍摄快照之前将其截断。

备份也称为恢复点，表示资源（例如，Amazon Redshift 集群）在指定时间的内容。AWS Backup 将备份保存在备份保管库中，您可根据业务需求进行这些备份进行整理。术语恢复点和备份可以互换使用。有关 AWS Backup 的更多信息，请参阅《AWS Backup Developer Guide》中的 Backup creation, maintenance, and restore。

Amazon Redshift 与 AWS Backup 原生集成。这样，您就可以定义备份计划并为备份计划分配 Amazon Redshift 资源。AWS Backup 自动创建 Amazon Redshift 手动快照，并将这些快照安全存储在您在备份计划中指定的备份保管库中。有关保管库的更多信息，请参阅《AWS Backup Developer Guide》中的 Backup vaults。在备份计划中，您可以定义备份频率、备份时段、生命周期或备份文件库。有关备份计划的更多信息，请参阅《AWS Backup Developer Guide》中的 Backup plans。

有关在不使用 AWS Backup 的情况下创建和还原 Amazon Redshift Serverless 快照的信息，请参阅快照和恢复点。有关在不使用 AWS Backup 的情况下创建和还原 Amazon Redshift 预置集群快照的信息，请参阅 Amazon Redshift 快照和备份。

主题

- 将 AWS Backup 与 Amazon Redshift 配合使用时的注意事项
- 将 AWS Backup 与 Amazon Redshift 结合使用的限制
- 使用 Amazon Redshift 管理 AWS Backup

# 将 AWS Backup 与 Amazon Redshift 配合使用时的注意事项

以下是将 AWS Backup 与 Amazon Redshift 配合使用时的注意事项：

- 如果在同一个 AWS 区域 中同时提供了 AWS Backup 和 Amazon Redshift，则可使用 AWS Backup for Amazon Redshift。有关 AWS Backup 可用的区域的信息，请参阅《AWS 一般参考》中的 Amazon Redshift endpoints and quotas。
- 要开始使用 AWS Backup，请确认您已满足所有先决条件。有关更多信息，请参阅《AWS Backup Developer Guide》中的 Prerequisites。
- 积极地选择加入 AWS Backup 服务。选择加入选项适用于特定账户和 AWS 区域。如果您想通过给定账户在多个区域中使用备份，则必须使用该账户选择加入每个单独的区域。有关更多信息，请参阅《AWS Backup Developer Guide》中的 Opt in to managing services with AWS Backup。
- Amazon Redshift 的 AWS Backup 集成仅支持预置集群和无服务器命名空间的手动快照。
- 使用 AWS Backup 管理快照设置后，您无法继续使用 Amazon Redshift 管理手动快照设置。相反，您可以继续使用 AWS Backup 计划管理设置。有关更多信息，请参阅《AWS Backup Developer Guide》中的 Backup plans。
- 将整个数据仓库快照还原到无服务器命名空间是一种破坏性更改。将数据仓库快照还原到目标命名空间时，该命名空间中以前存在的所有数据都将丢失。这仅适用于还原数据仓库快照。将单表快照还原到命名空间不会删除现有数据。
- 要将快照还原到预置集群，您需要具有拥有 `RestoreFromClusterSnapshot` 权限的 IAM 策略。要将快照还原到无服务器命名空间，您需要具有拥有 `RestoreFromSnapshot` 权限的 IAM 策略。这些权限适用于目标数据仓库类型，而不适用于源快照类型。例如，要将集群快照还原到命名空间，您需要 `RestoreFromSnapshot` 权限，而不是 `RestoreFromClusterSnapshot`。有关管理 IAM 策略的更多信息，请参阅 Amazon Redshift 中的 Identity and Access Management。

# 将 AWS Backup 与 Amazon Redshift 结合使用的限制

以下是将 AWS Backup 与 Amazon Redshift 结合使用的限制：

- 您不能使用 AWS Backup 来管理 Amazon Redshift 自动快照。要管理自动快照，请使用标签。有关标记资源的信息，请参阅在 Amazon Redshift 中标记资源。
- 从快照还原单个表时，您无法从预置集群快照还原到无服务器命名空间，反之亦然。您可以在任何配置中还原整个快照。例如，您可以将预置集群快照中的所有数据库还原到无服务器命名空间，但不能将该同一个快照中的单个表还原到同一个命名空间。

# 使用 Amazon Redshift 管理 AWS Backup

要保护 Amazon Redshift 数据仓库上的资源，您可以使用 AWS Backup 控制台，也可以通过编程方式使用 AWS Backup API 或 AWS Command Line Interface（AWS CLI）。需要恢复某个资源时，您可以使用 AWS Backup 控制台或 AWS CLI 来查找和恢复所需的资源。有关更多信息，请参阅 AWS Command Line Interface。

使用 AWS Backup for Amazon Redshift 时，您可以执行以下操作：

- 创建定期备份，自动启动 Amazon Redshift 快照。为了满足长期数据留存需求，定期备份很有用。有关更多信息，请参阅《AWS Backup Developer Guide》中的 Amazon Redshift backups。
- 通过集中配置备份策略，自动执行备份计划和保留。
- 将预置集群或无服务器命名空间还原到您选择的已保存备份。您可以选择还原快照中的所有数据，也可以选择从快照还原单个表。您可以设置备份资源的频率。有关还原预置集群快照的信息，请参阅《AWS Backup Developer Guide》中的 Restore an Amazon Redshift cluster。有关还原无服务器命名空间快照的信息，请参阅《AWS Backup Developer Guide》中的 Amazon Redshift Serverless restore。

# Amazon Redshift 的集群版本

Amazon Redshift 定期发布集群版本。您的 Amazon Redshift 集群将在系统维护时段进行修补。补丁的时间取决于 AWS 区域和维护时段设置。您可以从 Amazon Redshift 控制台中查看或更改维护时段设置。有关维护的更多信息，请参阅集群维护。

您可以在 Amazon Redshift 控制台上集群详细信息的维护选项卡中查看集群的集群版本。或者，您可以在 SQL 命令的输出中看到集群版本：

```
SELECT version();
```

> **Note**
>
> 随着 Amazon Redshift 的发展，我们引入了会影响 Amazon Redshift 行为的关键更新。要及时了解这些更改、采取措施并避免可能对工作负载造成中断，请参阅 Amazon Redshift 中的行为更改。

主题


- Amazon Redshift 补丁 205
- Amazon Redshift 补丁 204
- Amazon Redshift 补丁 203
- Amazon Redshift 补丁 202
- Amazon Redshift 补丁 201
- Amazon Redshift 补丁 200
- Amazon Redshift 补丁 199
- Amazon Redshift 补丁 198
- Amazon Redshift 补丁 197
- Amazon Redshift 补丁 196
- Amazon Redshift 补丁 195
- Amazon Redshift 补丁 194
- Amazon Redshift 补丁 193

- Amazon Redshift 补丁 192
- Amazon Redshift 补丁 191
- Amazon Redshift 补丁 190
- Amazon Redshift 补丁 189
- Amazon Redshift 补丁 188
- Amazon Redshift 补丁 187
- Amazon Redshift 补丁 186
- Amazon Redshift 补丁 185
- Amazon Redshift 补丁 184
- Amazon Redshift 补丁 183
- Amazon Redshift 补丁 182
- Amazon Redshift 补丁 181
- Amazon Redshift 补丁 180
- Amazon Redshift 补丁 179
- Amazon Redshift 补丁 178
- Amazon Redshift 补丁 177
- Amazon Redshift 补丁 176
- Amazon Redshift 补丁 175
- Amazon Redshift 补丁 174
- Amazon Redshift 补丁 173
- Amazon Redshift 补丁 172
- Amazon Redshift 补丁 171
- Amazon Redshift 补丁 170
- Amazon Redshift 补丁 169
- Amazon Redshift 补丁 168

# Amazon Redshift 补丁 205

**此修补程序中的集群版本：**

- 1.0.477953：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 10 月 6 日发布
- 1.0.434008：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 9 月 16 日发布

## 此补丁中的新功能和改进功能

- Iceberg MV：存储在 Iceberg 中的实体化视图，提供增量刷新和完全重新计算。
- 系统表现在包含一个 `external_user_id` 列，用于记录与每条查询关联的持久 IAM 或 IAM Identity Center 身份，从而支持跨集群生命周期事件按用户筛选数据。
- 针对外部表的 Amazon Redshift Spectrum 查询现已支持读取采用 gzip、bzip2 压缩的文本/CSV 文件。
- COPY 与 UNLOAD 命令新增对 IAM_ROLE 'SESSION' 关键字的支持。客户可使用会话绑定的 FAS 凭证访问 Amazon S3，无需显式指定 IAM 角色 ARN。
- 新增 `ALTER DATABASE ... INTEGRATION REFRESH REMEDIABLE TABLES` 语句，可按需重新同步标有重复日期警告的零 ETL 表，还可选择限定至某个架构范围。
- 为零 ETL 集成新增 `IntegrationLatestSourceCommit` Amazon CloudWatch 指标，用于报告观测到的最新源提交时间，以便您区分源闲置与管道停滞。
- Amazon Redshift 现在允许管理员使用 `ALTER SYSTEM SET passwd_min_length = N`（8-32）来提高数据库用户密码所需的最小长度要求。
- 修复了一个问题：在配置为自动创建角色的 Azure 身份提供商上执行 `ALTER IDENTITY PROVIDER ... ENABLE` 时，会抛出参数不正确错误。
- Amazon Redshift 现在支持 TLS 连接的后量子混合密钥交换 (`X25519MLKEM768`)。
- 在联合身份验证权限目录数据库上，Amazon Redshift 现在支持 `DEBUG` 权限，允许全局身份、IAM Identity Center（IdC）组或使用者账户管理员查看受精细访问权限控制（FGAC）保护的数据的未脱敏安全日志记录。
- Amazon Redshift 现在通过 Metadata Security 支持限定范围的权限。
- Iceberg 表现已支持在建表时定义 NOT NULL 列约束。
- 由 Amazon Redshift 发起 API 调用产生的 Amazon CloudTrail 审计事件现在包含一个 `queryUuid` 字段，可用于与系统表关联。
- 为了保持一致性，系统表与视图中的 `user_query_uuid` 列重命名为 `query_uuid`。
- 提升了高查询并发场景下，带有零 ETL 集成的集群的查询性能与可用性。

- 现在，对于列数较多、列级权限较少的表，权限检查速度得到优化，从而加快了元数据查询以及用于列出表权限的客户端工具的执行速度。
- 增强了在聚合中使用 Lambda UDF 的查询的稳定性。
- 现在，自动去除膨胀功能可以让超级块收缩到适配其实时标题集，无需再通过 Superblock Vacuum（SBV）回收膨胀。
- 改进了自动表排序，以减少包含大量列的表的存储增长。
- 修复了一个问题，该问题会导致自动 vacuum 操作停止在集群上执行，必须重启集群才能恢复。
- 修复了一个问题，在使用自动工作负载管理并开启并发扩展的集群上，该问题在 vacuum 操作执行期间可能引发集群重启。
- 修复了一个错误，该错误可能导致包含字符串字面量的递归 CTE 出现类型推断错误。
- 修复了一个问题：联合身份验证 PostgreSQL 查询在远端数据库无响应时会发生挂起，并且忽略取消请求。
- 修复了一个罕见问题：内存压力场景下，COPY 加载状态管理发生内存二次释放，进而导致集群重启。
- 修复了一个问题：当 `LIKE` 子句传入长模式时，`SHOW` 发现命令无法找到结果。
- 修复了一个问题：当集群数据库角色数量已达上限时，创建或发布数据共享的操作可能失败。
- 现在，取消 `DROP DATABASE` 命令时，该命令将以报错终止，不再引发意外的集群重启。
- 提升了并发扩展启用场景下存储过程的可靠性：存储过程多次查询 Iceberg 表后再查询本地表时，稳定性得到改善。
- 提升了并发扩展启用场景下会话的可靠性：会话中混用 Iceberg 游标与非 Iceberg 查询时，稳定性得到改善。
- 新增了功能，支持对 fdisk 提取块的 IPC 等待操作做可取消处理，当 ACK 丢失时，可直接终止查询，避免查询僵死无法杀掉。
- 优化了在线弹性调整大小期间内部进程的可靠性，使该进程更具确定性/稳健性。

# Amazon Redshift 补丁 204

此修补程序中的集群版本：

- 1.0.436211：后续发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 9 月 22 日发布
- 1.0.436211：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 9 月 10 日发布

- 1.0.416217：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 8 月 25 日发布
- 1.0.394035：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 8 月 11 日发布

## 此补丁中的新功能和改进功能

- 添加了对单节点 rg.large 集群的支持。这是创建单节点 rg.large 集群所需的最低集群版本。
- 向系统表和视图中添加了 query_uuid 列，提供了在集群暂停/恢复周期中持续存在的全局唯一查询标识符。
- 向 stl_udf_log 表中添加了 user_query_id，以消除 sys_ 视图定义中对 stl_user_query_map 联接的依赖。
- 为零 ETL 集成添加了两个新的 Amazon CloudWatch 指标，即 IntegrationLatestDetectedChange 和 IntegrationLatestAppliedChange，以便监控复制运行状况。
- 在零 ETL 集成数据库上为 CREATE DATABASE 和 ALTER DATABASE 添加了 AUTO_REMEDIATION 选项。设置 AUTO_REMEDIATION 后，Amazon Redshift 会自动重新同步存在复制问题的表。详细信息显示在 svv_integration_table_state.reason 中。
- 增加了对使用 Amazon Redshift 原生引擎读取、创建和写入 Apache Iceberg v3 表的支持。这包括跨以前支持的所有数据类型对默认值、行谱系和删除向量的支持。此版本不包括对 Variant、Geography 和 Geometry 数据类型的支持，这一支持将在未来的补丁中公布。
- 将 Iceberg 表升级到 v3 后，Iceberg `timestamptz` 类型映射到 Amazon Redshift `TIMESTAMPTZ` 类型。在 Iceberg v2 表中，`timestamptz` 映射到 Amazon Redshift `TIMESTAMP` 类型。在 v3 中，这意味着您的查询会根据其时区输出时间戳。
- 添加了对超过实例类型的最大结果集大小的 `CURSOR` 的支持。当游标的结果集超过可用的磁盘容量时，Amazon Redshift 会自动将溢出数据转移到 Amazon S3。可以通过向 AWS Support 发出请求来为每个集群启用此功能。
- 在 `stl_fasma_data_cache_stats` 系统表中添加了新的装箱指标列，用于监控缓存活动。
- 现在，对引用后期绑定视图（LBV）的查询进行解析和语义分析是在使用者集群上进行的。这消除了 LBV 处理中的生产者方瓶颈。
- 现在，Amazon Redshift ML 在向 Amazon SageMaker 发出 `CreateAutoMLJob` 和 `CreateProcessingJob` 请求时启用容器间流量加密选项。这与已应用于 `BuildTrainingRequest` 的设置保持一致，可以防止出现安全扫描器警报。
- 提高了高频工作负载的零 ETL 集成性能，以减少具有大量并发更新的表的复制滞后。

- 零 ETL 集成现在可以正确地将超过目标列长度的源数据报告为不可重试的错误。
- 提高了集群重启后的零 ETL 集成稳定性。
- 现在，当零 ETL 集成上的表筛选条件更改从架构中移除所有表时，会自动清理空的架构。
- 改进了在具有有效零 ETL 集成的数据库上尝试执行 DROP DATABASE 时的错误消息。
- DynamoDB 零 ETL 集成现在在初始做种和持续复制期间都支持 ACCEPTINVCHARS 和 TRUNCATECOLUMNS 选项。当源包含无效的 UTF-8 字符时，表可继续复制。
- 提高了使用零 ETL 集成的集群的查询性能，特别是对于复制许多数据库或表的工作负载。
- 提高了使用临时表的工作负载的性能，减少了目录访问开销并提高了总体吞吐量。
- 诸如 SHOW TABLES、SHOW SCHEMAS、SHOW COLUMNS 和 SHOW GRANTS 等 SHOW 发现命令现在最多可返回 133500 个对象，因此，对象数量非常大的架构和数据库的元数据列出将获得成功，而不是因行数限制为 10000 错误而失败。
- 当 DROP USER 命令因用户仍拥有对象或持有权限而遭拒绝时，错误消息现在会指明具体的阻止对象，并报告还剩余多少其它依赖项。
- Amazon Redshift 现在可在领导节点上缓存 IAM 角色凭证。这减少了 COPY、UNLOAD、Amazon Redshift Spectrum 和 Lambda UDF 查询的检索开销。在查询中重用常用的凭证，从而缩短了查询启动时间。
- AWS IAM Identity Center 身份验证现在适用于启用了增强型 VPC 路由的集群和工作组。
- 改进了 RG 节点类型的磁盘运行状况检测，以防止在正常磁盘 I/O 下运行的集群上发生不必要的自动硬件修复。
- 通过跳过后端进程中的额外副本，提升了数据共享性能。
- 扩展了对多仓库写入查询的使用者规模支持。
- 增强了在联接中没有参数的 Lambda 用户定义函数的稳定性。
- 修复了谓词包含 IDF 时 MDDL 排序键计算中的一个问题。
- 修复了在 BEGIN READ ONLY 事务中执行对外部表、数据共享对象和跨数据库引用的查询时失败并显示“事务为只读”的问题。
- 修复了在同一个表上同时运行 COPY 或 INSERT 与进行自动表优化时可能导致集群重启的罕见情况。
- 修复了一个罕见的编码不匹配错误，该错误可能导致对通过数据共享访问的、在较旧的集群版本上创建的表进行自动表优化时失败。
- 修复了一种罕见的情况，即对数据分布存在偏差的表进行自动 vacuum 可能会导致集群重启。
- 修复了 Auto COPY 作业可能停滞不前并阻止同一集群上的其它 Auto COPY 操作的问题。
- 修复了在查询清理期间由于争用条件而导致 COPY 命令可能引发集群重启的罕见情况。

- 修复了 svl_s3query_summary.s3query_returned_rows 可能会为外部表上带有 LIMIT 子句的查询额外报告一行的问题。
- 修复了在具有零 ETL 集成且未配置 Auto COPY 作业的集群上引导期间可能发生的罕见集群重启问题。
- 修复了在将多字符分隔符与 ESCAPE 选项结合使用时 COPY 字段解析的问题。
- 修复了通过 IAM Identity Center 连接的用户使用 CREDENTIALS 参数提供显式临时凭证时，COPY 和 UNLOAD 命令失败的问题。
- 修复了 SHOW CONSTRAINTS 中可能导致集群重启的问题。
- 修复了向角色授予的 CONNECT 权限不支持映射组中的 AWS IAM Identity Center 用户在启用了 Amazon Redshift 联合权限的数据库上进行连接的问题。
- 修复了在 SUPER 数组上使用 UNNEST 的查询被错误地阻止在并发扩展集群上运行的问题。
- 修复了 ARRAY_SORT 函数在应用于嵌套的 SUPER 结构时可能因路径导航错误而失败的问题。
- 修复了深度嵌套的 PartiQL 或 SUPER 表达式可能导致集群重启的问题。
- 修复了格式字符串指定的精度大于 38 时 TO_NUMBER 函数可能失败的问题。
- 修复了加密密钥轮换可能损坏嫁接表的密钥，从而可能导致数据访问失败的问题。
- 修复了存储操作中可能导致查询挂起或节点无响应的死锁情况。
- 修复了在磁盘块管理期间可能导致节点重启的罕见崩溃。
- 修复了在块分配期间内存不足导致的崩溃，该问题可能导致节点重启。
- 修复了在块复制分配期间磁盘已满条件导致的崩溃，该问题可能导致节点重启。

# Amazon Redshift 补丁 203

此修补程序中的集群版本：

- 1.0.417950：后续发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 8 月 26 日发布
- 1.0.417950：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 8 月 25 日发布
- 1.0.400034：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 8 月 17 日发布
- 1.0.384821：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 8 月 4 日发布

- 1.0.377293：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 7 月 25 日发布
- 1.0.365190：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 7 月 21 日发布
- 1.0.347559：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 7 月 6 日发布

## 此补丁中的新功能和改进功能

- 向 stl_alter_dist_sort 中添加了一个新列，用于指示 ALTER DISTKEY 和 ALTER SORTKEY 操作是在主集群、数据共享使用者还是在并发扩展集群上执行的。
- 添加了 SHOW GRANTS FOR GROUP 语法，用于发现授予特定用户组的权限。
- 通过为检查点添加精细的成功状态，增强了零 ETL 集成的可观测性，从而可以更精细地监控系统表中的复制进度。
- 引入了对零 ETL 集成内的源表架构和 DDL 元数据中不支持的 UTF-8 字符的妥善处理，从而防止集成失败。
- Amazon Redshift 现在支持并发扩展，用于刷新与 Amazon Kinesis Data Streams（KDS）连接的流式实体化视图（MV）。
- 从补丁 203 开始，Amazon Redshift 开始强制终止对现有 Python UDF 的支持。在补丁 198 中弃用了创建新的 Python UDF，所有 Python UDF 支持都将在 2026 年 6 月 30 日之后终止。将现有 Python UDF 迁移到 Lambda UDF，以确保持续运行。有关迁移指南，请参阅 AWS 大数据博客上的 Amazon Redshift Python user-defined functions will reach end of support after June 30, 2026 和 Best practices for Amazon Redshift Lambda user-defined functions。有关行为变更的详细信息，请参阅标量 Python UDF 将在 2026 年 6 月 30 日之后终止支持。
- 从 Amazon Redshift Spectrum 中使用 COPY 加载的表、具有显式列编码的表或加载到非空表中的表，现在在表具有主键时都有资格自动分配 DISTKEY，从而提高主键联接的查询性能。
- 更新了数据共享表上的 ALTER TABLE … ALTER DISTKEY 错误消息，以显示面向客户的表名。
- 减少了对表数量非常多的集群进行自动 vacuum 时的后台活动，从而提高了以前维护活动过多的集群的稳定性。
- 在某些情况下，通过将子查询与不匹配的筛选条件、聚合和窗口函数相结合，优化了查询性能。
- 提高了在表重新同步后使用 Amazon DynamoDB 历史记录模式对零 ETL 表进行时间旅行查询的时间戳准确性。
- 在具有某些主键配置的表上处理 CDC 检查点时，零 ETL 集成不再进入重新同步循环。

- 改进了针对无效通道筛选表达式的零 ETL 集成错误报告。
- 原始查询文本不再使用安全日志记录进行屏蔽（适用于使用联合 Amazon Redshift 权限对外部 AWS Lake Formation 视图或数据库进行的查询）。
- 将针对大型表的 ANALYZE 的性能提高了多达 30%。
- 增强了将 AWS Lambda 用户定义函数与 ROLLUP 聚合结合使用的查询的性能。
- 通过跳过对哈希联接的外部表进行不必要的数据扫描，增强了带 LIMIT 子句的 ORDER BY 的查询性能。
- 减少了后台树扫描操作期间的内存消耗，消除了实例集范围的节点崩溃。通过实施幸存者计数的批量耗尽，限制了计算节点树扫描的内存占用，防止在高元数据流失情况下内存无限增长。现在，树扫描路径以后台树重播完成为门控条件。
- 改进了 Amazon S3 读取失败后的故障恢复，消除了节点崩溃。现在，在 Amazon S3 读取失败后，涓流树迭代器可以正确地重置，从而确保后续读取正确重试，而不是静默地返回陈旧或不完整的数据。
- 修复了在自适应集群调整大小后，使用 ALTER TABLE … ALTER DISTSTYLE AUTO 修改的表可能会在 sys_restore_state 中无限期地保持为待定状态的问题。
- 修复了 SYS_QUERY_TEXT 系统视图可能显示缺少空格的查询文本或在分块边界处错误地拆分单词的问题。
- 修复了由于 Amazon Redshift 目录元数据不一致性，导致在 2023 年 8 月至 2024 年 3 月期间创建的集群因零 ETL 集成而启动失败的问题。
- 修复了 SHOW GRANTS 错误地显示向 identity_type 为“role”的用户组授予的权限的问题。
- 修复了在处理会话暂停期间查询可能无限期挂起的问题。
- 修复了以下问题：拥有 SYSLOG ACCESS UNRESTRICTED 的非超级用户只能在 sys_session_history 中查看自己的会话数据，而不能像预期的那样查看所有用户的会话。
- 修复了数据库或用户连接限制设置为 1000 或更高时被截断的问题，该问题会导致连接数不正确。
- 修复了在内存压力下查询可能无限期挂起的问题。
- 修复了 Amazon Redshift Serverless 端点在暂停操作期间可能出现短暂不可用的问题。
- 修复了在查询编译过程中具有某些参数组合的 COPY 命令可能意外失败的问题。
- 修复了将 APPROXIMATE PERCENTILE_DISC 与 GROUP BY 结合使用的查询可能因编译错误而失败的问题。
- 修复了使用 COLLATE 表达式的查询可能返回不正确列值的错误结果问题。
- 修复了具有 SUPER 类型值的 LEAST 和 GREATEST 函数可能导致查询失败的问题。

- 修复了 REGEXP_REPLACE 中的一个错误结果问题，即使用捕获组引用的替换字符串可能会生成截断的输出。
- 修复了在中止的重新复制读取期间可能出现的数据损坏问题。解决了重新复制读取在中途中止时可能出现的双重释放情况，从而防止可能的内存损坏或崩溃。

# Amazon Redshift 补丁 202

此修补程序中的集群版本：

- 1.0.369760：后续发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 7 月 29 日发布
- 1.0.377292：后续发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 7 月 25 日发布
- 1.0.377292：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 7 月 25 日发布
- 1.0.358853：后续发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 7 月 21 日发布
- 1.0.358853：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 7 月 14 日发布
- 1.0.343555：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 6 月 29 日发布
- 1.0.340728：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 6 月 22 日发布
- 1.0.331549：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 6 月 16 日发布
- 1.0.323213：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 6 月 3 日发布
- 1.0.314775：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 5 月 28 日发布

## 此补丁中的新功能和改进功能

- 增加了对 RG 节点类型（rg.large 和 rg.12xlarge 多节点）的支持。这是迁移到 rg.large 和 rg.12xlarge 所需的最低集群版本。

- 添加了对超级用户的支持，使其可删除 Lakehouse 类型处于禁用状态的 IAM Identity Center 身份提供商
- 在 STL_QUERY 和 STL_QUERYTEXT 系统表中出现了标有“JIT ANALYZE”的内部维护查询。这些查询由 Amazon Redshift 自动执行，以收集外部表统计数据并使查询计划保持优化。客户无需执行任何操作。
- 我们已优化了 Kinesis 流身份验证，以减少“DescribeStreamSummary”API 调用。对于拥有许多 Kinesis 流 MV 的 Redshift 客户，这将避免达到 Kinesis API 配额。有关详细信息，请参阅 Redshift Streaming 文档部分。
- 通过对带有 GROUP BY、ORDER BY 和 LIMIT 子句的查询使用 Top K 聚合，减少了聚合期间处理的数据量，从而增强了查询执行性能。
- 通过将复合筛选条件谓词（OR、IN、NOT IN）推送到 Glue GetPartitions API，减少检索到的分区数量，增强了对已分区的 AWS Glue 数据目录表的查询性能。
- 增强了结合使用 Lambda 用户定义函数与窗口函数的查询的性能。
- 通过在哈希联接的联接筛选条件中使用 Lambda 用户定义函数，增强了特定查询的性能。
- 增强了在联接中使用 Lambda UDF 的查询的稳定性。
- 在创建 Lambda 用户定义函数时，支持使用 ARN 作为 AWS Lambda 的名称。
- 通过优化 COALESCE 和 CASE WHEN 表达式的内存使用情况，提高了查询性能。
- 对包含子计划的查询应用高级筛选，这些子计划只评估一次，其结果将在剩下的执行中作为常量重用。
- 使用 ST_Transform 提高了首次运行查询的性能
- 通过接受包含重复行索引的位置删除文件（这些文件可以通过 AWS Glue 压缩生成），改进了 Apache Iceberg 兼容性。
- 通过在删除文件引用中接受 s3a:// 路径，改进了 Apache Iceberg 兼容性。
- 改进了在处理配置不正确的表时对 Spectrum 分区级别提取错误的处理
- 提高了通过 USE 跨数据库调用嵌套 SQL UDF 的存储过程的稳定性
- 改进了存储过程不支持的操作（例如，在数据共享中添加或移除存储过程）的错误消息。
- 修复了在使用 EVR 时无法检测到客户的 VPC 信息的错误。此类信息仅在训练本地 Redshift 机器学习模型时才是必需的，不会对任何其它 Redshift 功能产生影响。这不会阻止使用经过训练的现有 Redshift 机器学习模型。
- 修复了当内部查询 ID 计数器接近其最大值时而出现集群暂停并恢复后，系统表中可能会出现重复的查询 ID 的问题。
- 修复了由于内部缓存不一致而导致在多可用区失效转移后对 STL 系统表的查询可能返回不完整结果的问题

- 修复了扩展行描述中的“isAutoIncremented”字段可能会为 IDENTITY 列返回错误值的问题
- 修复了在给定 NULL 输入时，ARRAY_CONTAINS 和 ARRAY_POSITION 函数未按预期返回 NULL 的问题。
- 修复了一个罕见的问题，即在向量化扫描期间使用多个组合筛选条件谓词的查询可能会意外失败。
- 修复了一个罕见的问题，即并发 COPY 操作可能会导致集群在数据加载过程中出现不稳定性
- 修复了在进程关闭期间自动维护操作（例如 vacuum 和分析）可能导致集群不稳定的问题
- 修复了在动态关闭工作负载管理（WLM）短查询加速并重新开启后查询可能失败的问题
- 修复了在历史记录模式下对零 ETL 表进行时间旅行查询的一个问题。
- 修复了在并发客户端关闭期间零 ETL 集成可能变得无响应的问题
- 修复了共享源分片的零 ETL 表可能触发不必要的重新同步的问题
- 修复了当内部原因字段超过了支持的长度时，零 ETL 集成表状态更新可能失败的问题
- 修复了在 Aurora PostgreSQL 分区表的列更改过程中零 ETL 集成遇到错误的问题

# Amazon Redshift 补丁 201

此修补程序中的集群版本：

- 1.0.331388：后续发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 6 月 16 日发布
- 1.0.331388：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 6 月 15 日发布
- 1.0.326225：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 6 月 8 日发布
- 1.0.319797：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 6 月 3 日发布
- 1.0.313685：当前发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 5 月 27 日发布
- 1.0.300094：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 5 月 12 日发布
- 1.0.298697：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 5 月 7 日发布
- 1.0.292316：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 5 月 4 日发布

- 1.0.280974：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 4 月 22 日发布

## 此补丁中的新功能和改进功能

- 增加了对 RG 节点类型（rg.xlarge 和 rg.4xlarge 多节点）的支持。这是迁移到 rg.xlarge 和 rg.4xlarge 所需的最低集群版本。
- 增加了对 Iceberg 表上 ALTER 的支持。
- 增加了对挂载在 AwsDataCatalog 下的 Iceberg 表的写入支持。
- 增加了在使用 USE 命令连接到数据库时，对 SHOW GRANTS 命令的支持。
- 增加了对具有联合权限的身份的 SHOW GRANTS 和 SHOW COLUMN GRANTS 命令的支持。
- 增加了在 Amazon Redshift Serverless 上，用户工作负载出现容量爆增时对实体化视图自动刷新的支持。
- 允许 AutoWLM 为写入宽表的查询分配更多集群内存。
- Amazon Redshift 提高了 SVV_ATTACHED_MASKING_POLICY 系统视图的查询性能，缩短了在具有大量权限和动态数据掩蔽策略附件的实例上的执行时间。
- 将 librdkafka 库从版本 2.8.x 升级到 2.13.x，以提供最新的错误修复和性能增强。
- 增强了在聚合的 HAVING 子句中，使用 Lambda 用户定义函数的查询的性能。
- 增强了结合使用 Lambda 用户定义函数与窗口函数的查询的性能。
- 增强了没有参数的 Lambda 用户定义函数的稳定性。
- 通过消除不必要的数据扫描，提高了使用 ORDER BY 和 LIMIT 子句时的数据湖表查询执行性能。
- 改进了零 ETL 集成的错误消息，为不支持的约束、不支持的 DDL 语句、日期精度丢失、多字节字符长度不匹配和跳过源的表提供了更清晰的详细信息。
- 通过在复制状态检查期间妥善处理凭证获取失败，提高了零 ETL 集成的韧性。
- 提高了运行并发复制任务时零 ETL 集成的性能。
- 改进了具有频繁更新行的零 ETL 表的时间旅行查询结果，以确保每个历史行版本都反映其各自的删除时间戳。
- 修复了一些人工智能驱动的扩展无法正确调整主集群大小的边缘情况。
- 修复了在工作负载管理（WLM）驱逐和重试查询时，Amazon CloudWatch 指标可能会报告重复的失败查询计数的问题。
- 修复了使用 Redshift Table Copy 的 COPY 操作可能导致分布方式不匹配的问题。

- 修复了不正确的分区排序结束行计算问题，该问题可能会影响限制范围的扫描结果。
- 修复了在删除表的同时自动 vacuum 操作可能失败的问题。
- 修复了 VACUUM 在获取关系时会使用陈旧表版本而可能导致错误的问题。
- 确保在突发集群上正确清理 vacuum 事务映射。
- 修复了通过数据共享访问时，对零 ETL 复制表的查询可能失败的问题。
- 修复了零 ETL 集成数据复制中可能导致集群重启的问题。
- 修复了在更新合并操作期间与 DynamoDB 历史模式的零 ETL 集成可能失败的问题。
- 修复了零 ETL 集成表可能会在 Amazon Redshift 控制台中显示错误行数的问题。
- 修复了在零 ETL 集成操作期间阻止 Amazon Redshift Serverless 工作组暂停的问题。
- 修复了空 Avro 文件导致 COPY 摄取失败的问题。现在系统可以跳过空 Avro 文件并继续摄取而不会出现错误。
- 修复了使用与位置无关的超级块时，在 epsilon 树重播期间出现过时块标头校验和的问题。

# Amazon Redshift 补丁 200

此修补程序中的集群版本：

- 1.0.314426：后续发布通道 Amazon Redshift 预调配集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 5 月 27 日发布
- 1.0.293472：尾随 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 5 月 7 日发布
- 1.0.293472：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 4 月 30 日发布
- 1.0.266483：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 4 月 9 日发布
- 1.0.257586：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 3 月 27 日发布

## 此补丁中的新功能和改进功能

- 修复了使用多维数据布局排序键对表执行 INSERT 操作可能失败的问题。
- 针对可以受益于基于工作负载的排序的表，改进了 Amazon Redshift Advisor 关于多维数据布局（MDDL）排序键的建议。

- 增强了 Redshift Serverless 人工智能驱动的扩展和优化，改进了磁盘使用量信号，从而实现更准确、更灵敏的扩展决策。
- 增强了 Redshift Serverless 人工智能驱动的扩展和优化，可在您将预置集群快照迁移到无服务器命名空间时智能地确定初始容量。
- 修复了使用方缓存的身份令牌过期时，使用 IAM Identity Center 身份传播的数据共享查询可能失败的问题。
- 修复了 IAM Identity Center 组同步可能导致后续后台维护操作失败的问题。
- 修复了使用不同应用程序 ARN 更新 IAM Identity Center 身份提供商配置可能导致集群重启的问题。
- 修复了在某些 AWS 区域使用 IAM Identity Center 身份传播进行数据共享查询可能失败的问题。
- 通过确保身份提供商配置更改的一致传播，提高了 IAM Identity Center 集成的可靠性。
- 通过将相关的 NOT EXISTS 子查询去关联化为反联接，优化了此类子查询的查询计划。
- 修复了当目标表不存在时，Apache Iceberg 表上的 DROP TABLE IF EXISTS 会错误地报告查询失败而不是成功的问题。
- 增加了用户尝试对目录表运行 ANALYZE 时正确的报错消息。
- Spectrum COPY 现已正式上线，消除了基于 Spectrum 的 COPY 操作对临时表的依赖。
- 通过在长时间运行的持久性任务无法跟上进度时对 S3 通知队列读取进行节流，减少了 S3 列出操作的需求来协调 COPY JOB 状态，从而提高了 Auto COPY 的可靠性。
- 并发扩展现已正式上线，可用于零 ETL 表上的读取和写入操作。
- 修复了零 ETL CDC 复制中的争用条件，即在并发磁盘提交干扰到未提交的表状态跟踪时，可能会导致集群重新启动。
- 通过对表和客户端状态使用最新摄取位置来筛选已应用的数据，提高了零 ETL CDC 摄取的可靠性。
- 修复了零 ETL 表重新同步中的数据可靠性问题，通过在重新同步生命周期中正确跟踪检查点下限和重新同步状态，可以在表失败后重新摄取已应用的旧检查点。
- 改进了零 ETL 错误消息，提供更清晰的失败原因，包括超出列长度和无效 UTF-8 字符错误的详细信息。
- 增强了 SYS_LUDF_DETAIL 系统视图，以微秒而不是毫秒为单位报告持续时间，从而改进了可观测性。
- 修复了在零 ETL 数据库中错误地允许 Auto COPY 作业创建的问题。
- 通过将其他内置函数归类为行级别安全性（RLS）安全，提高了受 RLS 保护的表的查询性能，从而为以前会遇到性能下降的常见查询模式启用谓词下推。
- 增加了对 Apache Iceberg 表上 DELETE、UPDATE 和 MERGE 操作的支持。

- 增加了对与位置无关的超级块的支持，从而无论数据量如何，都可实现可预测的集群重启时间。
- 启用基于 GUID 的磁盘缓存，消除了磁盘缓存对超级块位置的依赖，从而无论数据量如何，都可以实现可预测的集群重启时间。
- 改进了自动表分布优化：具有单个主键的表现在可以自动从 ALL 分布转换为 KEY 分布，从而提高查询性能并减少存储开销。
- 修复了在 vacuum 和并发 DML 操作访问相同存储对象时，VACUUM 操作可能由于冲突而失败或生成错误结果的问题。
- 修复了处理特殊分区链时由于越界内存访问而导致某些表的 VACUUM FULL 会失败的问题。
- 修复了 ALTER TABLE OWNER 操作可能错误地允许将对象所有权更改为内部引导用户的问题。
- 修复了并发 VACUUM 操作和暂存表重用可能导致集群不稳定的争用条件。
- 修复了当处理的分区数量超过内部限制时，大型表上的 VACUUM FULL 或 VACUUM SORT 可能会失败并出现内部错误的问题。
- 修复了某个表已删除但系统仍在从远程集群聚合统计信息时，后台分区统计信息收集会失败的问题。
- 通过避免不必要的数据刷新，减少对已排序的表格执行 vacuum 所需的时间和资源，提高了 VACUUM SORT 操作的性能。
- 修复了在 vacuum 操作期间不正确的内部分区数据访问可能导致意外错误或集群不稳定的问题。
- 修复了 ALTER TABLE 操作可能错误地将表所有权更改为非特权用户的问题。在 ALTER 操作期间进行的所有者更改现在会正确地仅限于授权用户。
- 修复了一个罕见的问题，即对于具有大量分区的表，在 vacuum 操作期间进行的内部限制检查可能会导致不必要的查询失败。
- 修复了后台统计信息收集在遇到已删除表时可能失败或生成错误的问题，提高了集群的稳定性。
- 修复了 vacuum 操作可能使用过时的表元数据的问题，这可能会在对具有并发架构更改的表执行 vacuum 操作期间导致错误的行为。
- 发布了新的查询编译增强功能，可提高当前维护跟踪上 Amazon Redshift 补丁 P198 中新查询（查询的首次运行）的性能。现在，从尾随跟踪上的 Amazon Redshift 补丁 P200 开始的所有版本也都启用了此功能。

# Amazon Redshift 补丁 199

此修补程序中的集群版本：

- 1.0.266409：尾随 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 4 月 20 日发布

- 1.0.252420：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 3 月 25 日发布
- 1.0.232773：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 3 月 5 日发布

## 此补丁中的新功能和改进功能

- 发布了新的查询编译增强功能，可提高 Redshift 当前维护跟踪中新查询（查询的首次运行）的性能。尾随版本跟踪将随后在即将发布的补丁版本中发布。
- 支持使用者使用 Redshift 联合权限向生产者上的新被授权者提供的列级别授权。
- Amazon Redshift 联合权限支持通过后期绑定视图（LBV）查询外部表。
- ALTER EXTERNAL SCHEMA 现在可以与 KINESIS 外部架构类型一起使用，以更改现有外部架构的 IAM_ROLE 和 REGION。
- 在 Redshift RA3 和无服务器数据仓库上添加了对 NO-BACKUP 表的还原支持。由于这些实例类型不支持 NO-BACKUP 功能，因此，标记为 NO-BACKUP 的表始终包含在快照中，现在可以正确地还原为永久表
- Amazon Redshift 现在支持在使用 IAM 角色或 IAM 用户进行连接时写入外部表。以前，基于 IAM 角色的身份验证仅支持对外部表执行读取操作。通过此更新，Redshift 在向 Amazon S3 中写入外部表的数据时，可以正确使用您的 IAM 会话凭证，从而实现完全读写访问权限。
- 对于自动刷新 Kafka 流 MV，我们现在使用更高效的启发式方法来确定是否有新记录可供摄取。这将减少从 Kafka 代理传输的用于此检查的数据量，并减少启动后续 MV 刷新的延迟。
- 升级了备份和还原操作以使用符合 FIPS 标准的 SHA384 加密，取代了传统的 MD5 哈希算法
- 结合了 Lambda 用户定义函数和 UNNEST 语句的查询性能得到了增强
- 使用 Lambda 用户定义函数对系统表和视图进行查询的性能得到了增强
- 通过消除不必要的数据扫描，提高了使用 ORDER BY 和 LIMIT 子句时的查询执行性能
- 增强了 Redshift Serverless 人工智能驱动的扩展和优化，支持的更大范围的 RPU，现在至少支持 8 个 RPU
- Amazon Redshift 实体化视图（MV）的自动 REFRESH 查询现在被视为用户查询，而不是后台自主进程。现在，自动 REFRESH 查询的运行优先级与其它用户查询相同，并且不再被自主后台进程延迟。MV 自动 REFRESH 行为更改功能仅适用于补丁版本 P198 及更高版本的当前跟踪版本上的 Amazon Redshift 预置集群。它目前在无服务器上处于禁用状态。
- 改进了自动任务的调度，以更好地确定用户关键任务的优先级，例如流式摄取、自动复制（COPY JOB）和实体化视图自动刷新。

- 修复了 pg_last_query_id() 错误地返回重写的查询 ID 而不是多可用区集群的用户查询 ID 的问题
- 修复了 SHOW GRANTS 无法处理由单部分组成的对象引用的问题。
- 通过在 DROP TABLE 期间跳过删除并支持时间戳工作线程在其下次运行时进行清理，修复了由于 pg_class_extended 上的同时更新而导致 DROP TABLE 与目录重建工作线程的时间戳自动工作线程冲突的问题。
- 修复了当数据库、架构或表名称包括包含/排除关键字时的零 ETL 通道筛选条件解析问题
- 修复了 vacuum 操作与零 ETL 表操作并行运行时可能发生的争用情况
- 修复了当 KMS 加密上下文更改时零 ETL 集成错误地尝试重新同步操作而不是转换到错误状态的问题
- 修复了当数据复制处于活动状态时，在集群重启期间零 ETL 集成中可能发生的争用情况
- 修复了删除表的所有者后 SHOW TABLES 命令返回错误的问题

# Amazon Redshift 补丁 198

此修补程序中的集群版本：

- 1.0.239402：尾随 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 3 月 18 日发布
- 1.0.227967：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 2 月 25 日发布
- 1.0.224252：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 2 月 19 日发布
- 1.0.219674：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 2 月 11 日发布
- 1.0.212832：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 2 月 10 日发布
- 1.0.204436：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 2 月 4 日发布

## 此补丁中的新功能和改进功能

- 发布了新的查询编译增强功能，可提高 Redshift 当前维护跟踪中新查询（查询的首次运行）的性能。尾随版本跟踪将随后在即将发布的补丁版本中发布。

- 通过在将列与常量值进行比较时应用正确的排序规则，改进了 INTERSECT 和 EXCEPT 查询。
- 支持自动刷新手动创建的实体化视图，以便在并发扩展集群上运行。
- 通过在主集群和并发扩展集群之间实现直接连接，减少延迟并消除连接吞吐量限制，从而提高了连接性能。
- 改进了零 ETL 集成的错误分类，使识别根本原因和调试问题变得更加容易。
- 从 SYS_QUERY_HISTORY 中移除成功执行的查询的错误消息，从而提供更清晰的查询历史记录。
- 提高了 SYS_QUERY_HISTORY 中突增查询的编译时日志记录准确性，从而提供更好的性能见解。
- 添加了对数组函数的支持：ARRAY_CONTAINS、ARRAY_POSITION、ARRAYS_OVERLAP、ARRAY_INTERSECTION、ARRAY_S和 ARRAY_POSITIONS。
- 添加了对 GET_NUMBER_ATTRIBUTES 函数的支持，该函数返回 SUPER 对象内的属性数量。
- 为失败的事务启用空闲事务超时，以改进资源管理。
- 添加了对 TEMPLATE 对象类型的支持，此对象类型使您能够定义可重用的格式化参数，而无需为每个操作手动指定它们。
- 提高了在公用表表达式（CTE）上调用 Lambda 用户定义函数（LUDF）的查询的性能。
- 提高了在 CASE 表达式的 THEN 子句中调用 Lambda 用户定义函数（LUDF）的查询的性能。
- 弃用了创建新的 Python 用户定义函数。现有的 Python UDF 将继续得到支持，直至 2026 年 6 月 30 日。迁移到 Lambda 用户定义函数（LUDF）以获得持续支持。
- 使用新的系统表 SYS_LUDF_DETAIL 提高了 Lambda 用户定义函数（LUDF）的可观测性。此表记录查询中使用的 LUDF 的信息和指标。
- Amazon Redshift 实体化视图（MV）的自动 REFRESH 查询现在被视为用户查询，而不是后台自主进程。现在，自动 REFRESH 查询的运行优先级与其它用户查询相同，并且不再被自主后台进程延迟。MV 自动 REFRESH 行为更改功能仅适用于补丁版本 P198 及更高版本的当前跟踪版本上的 Amazon Redshift 预置集群。它目前在无服务器上处于禁用状态。

# Amazon Redshift 补丁 197

此修补程序中的集群版本：

- 1.0.237947：尾随 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 3 月 10 日发布
- 1.0.198462：尾随 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 1 月 28 日发布

- 1.0.198462：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 1 月 21 日发布
- 1.0.194394：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 1 月 14 日发布
- 1.0.187928：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 1 月 9 日发布
- 1.0.179517：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 12 月 16 日发布
- 1.0.176066：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 12 月 9 日发布
- 1.0.166219：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 11 月 21 日发布

## 此补丁中的新功能和改进功能

- 网格中集群的全局自主处理，自主处理现在会考虑整个网格的工作负载，以优化 Vacuum、分析和 ATO。
- 从多个 Amazon Redshift 数据仓库中创建和刷新实体化视图。
- 增加了对在 SUPER 数据类型中存储最多 1600 万个字节的字符串文本的支持
- 增加了 GET_NUMBER_ATTRIBUTES(super_object) 函数，该函数可返回 SUPER 对象中所含属性的总数。
- 对表还原操作强制实施版本补丁兼容性限制。尝试还原表时，备份只能用于运行的补丁版本与创建备份时相同的集群、运行的补丁版本比备份版本低一个补丁版本的集群，或运行的补丁版本比备份版本更高的集群。
- 增强了 LIKE 运算符，以便在使用 CHAR 数据类型时正确处理模式中的尾随空格。
- 新增了 4 个可在单元格层次结构上操作的空间型 H3 函数：H3_Resolution、H3_ToParent、H3_ToChildren 和 H3_IsValid。
- 启用 Amazon Redshift 联合身份验证权限，这可让您一次性定义数据权限并跨 AWS 账户中的所有仓库自动实施这些权限，从而简化跨多个 Redshift 数据仓库的权限管理工作。
- 支持在 Amazon Redshift 联合身份验证权限目录对象上执行对象级别授权、精细列级别访问控制、限定范围的权限授予以及数据库级别授权。
- 支持在 Amazon Redshift 联合身份验证权限目录上执行 CREATE、ALTER、ATTACH、DETACH 和 DROP RLS 策略操作。

- 支持为 Amazon Redshift 联合身份验证权限目录中的关系启用或禁用行级别安全性。
- 支持在 Amazon Redshift 联合身份验证权限目录上执行 CREATE、ALTER、ATTACH、DETACH 和 DROP 掩蔽策略操作。
- 支持通过 SHOW POLICIES 命令显示已连接数据库中的 RLS 策略和掩蔽策略，以及 Amazon Redshift 联合身份验证权限目录对象中的关系连接。
- 允许数据库用户代入 IAM 角色来查询 Amazon Redshift 联合身份验证权限目录对象。
- 增强了 SHOW GRANTS 命令，可在同一 Redshift 集群中实现跨数据库授权可见性。
- 增加了对 SHOW COLUMN GRANTS 命令的支持，可简化列级别访问控制发现操作。
- 增加了对 SHOW PROCEDURES 命令的支持，可简化存储过程元数据发现操作。
- 增加了对 SHOW FUNCTIONS 命令的支持，可简化用户定义的函数元数据发现操作。
- 引入了 SHOW PARAMETERS 命令，可显示函数和存储过程的参数元数据。
- 引入了 SHOW CONSTRAINTS 命令，可显示函数和存储过程的约束元数据。
- 增加了对在 Amazon Redshift 联合身份验证权限目录上和同一 Redshift 集群中的数据库间执行 CREATE/CREATE OR REPLACE/DROP VIEW 操作的支持。
- 改进了部分 INTERSECT 或 EXCEPT 查询中的常量字符串排序规则处理。
- 增强了 SHOW TABLES，可在访问 AWS Glue Data Catalog 所需的 FAS 凭证缺失时显示警告消息而不是指示失败。
- 增强了 SHOW COLUMNS 命令，增加了用于排序键、分布方式、编码和排序规则信息的元数据列。
- 向 SHOW GRANTS 输出添加了 grantor_name 列，并实现了所有 SHOW GRANTS 结果集中 database_name 包含的标准化。
- 增强了 SHOW TABLES 命令，增加了以下附加列：table_owner、last_altered_time、last_modified_time、dist_style 和 table_sub_type。
- Identity Center 身份验证支持：Amazon Redshift 现在支持通过新的 `GetIdentityCenterAuthToken` API 操作进行 IAM Identity Center 身份验证。借助此功能，可以使用现有 Identity Center 身份对 Amazon Redshift 集群和无服务器工作组进行联合访问。主要优势包括使用 Identity Center 凭证进行单点登录访问、使用加密身份验证令牌增强安全性、全面的合规性审计日志记录，以及与现有 Identity Center 设置的无缝集成。API 操作生成范围限定于特定集群标识符或工作组名称的身份验证令牌。要求包括带有嵌入式 Identity Center 身份信息的身份增强凭证以及用于 API 访问的相应 IAM 权限。

# Amazon Redshift 补丁 196

此修补程序中的集群版本：

- 1.0.179872：尾随 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2026 年 1 月 8 日发布
- 1.0.163480：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 11 月 20 日发布
- 1.0.160807：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 11 月 14 日发布
- 1.0.155905：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 11 月 11 日发布

### 此补丁中的新功能和改进功能

- 已通过将相关的 IN 和 EXISTS 子查询去关联化为半联接，优化此类子查询的计划。
- 扩展了 SHOW TABLE 命令以包含每个列的排序规则信息。

## Amazon Redshift 补丁 195

此修补程序中的集群版本：

- 1.0.172541：尾随 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 12 月 8 日发布
- 1.0.162991：尾随 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 11 月 20 日发布
- 1.0.151179：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 11 月 4 日发布
- 1.0.148180：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 10 月 31 日发布
- 1.0.142459：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 10 月 24 日发布

### 此补丁中的新功能和改进功能

- 增加了对使用 CREATE TABLE、CREATE TABLE AS SELECT、INSERT INTO、SQL 命令向 Iceberg 表写入内容的支持。
- 增加了对具有不区分大小写的排序规则的数据库中的 SUPER 数据类型的支持。

- 将 IANA 时区数据库的数据库版本更新到 2025b。
- 增加了对 TZDB_VERSION 函数的支持。此函数显示当前正在使用的 IANA 时区数据库版本。
- Amazon Redshift 现已实施补丁兼容性限制。对于相比当前集群低一个以上补丁版本的补丁版本，不再允许对在此类补丁版本上创建的备份执行表还原操作。
- 扩展 UNNEST 功能以支持 JOIN 表达式。以前，UNNEST 仅限于单表引用。
- 增加了对 JIT（即时）ANALYZE 功能的支持。JIT ANALYZE 功能可自动、智能且高效地收集 Amazon Redshift 的查询优化器生成高效查询执行计划所需的最小统计数据集，让客户能够在 Apache Iceberg 开箱即用（OOB）查询和工作负载中实现最佳性价比。

# Amazon Redshift 补丁 194

此修补程序中的集群版本：

- 1.0.148169：尾随 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 11 月 5 日发布
- 1.0.138443：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 10 月 18 日发布
- 1.0.129791：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 10 月 8 日发布

## 此补丁中的新功能和改进功能

- Amazon Redshift 将不再判定以下查询执行失败：该查询访问由 AWS Lake Formation 治理的关系且该关系的 `AuthorizedColumns` 字段包含反斜杠。
- 增加了对 SUPER 列在 COPY 命令中使用 NULL AS 选项的支持，可让您指定一个自定义字符串来代表 NULL 值。
- 增加了对在 EXTRACT 和 DATE_PART 函数中对日期和时间戳使用 ISOWEEK 和 ISOYEAR 日期部分的支持。
- 修复了分组变量的 PartiQL 导航功能，确保在 GROUP BY 查询中正确解析嵌套字段访问并进行求值。
- 增加了对使用 SHOW CONSTRAINTS 命令查看主键和外键约束的支持。
- 增加了对通过数据共享实体化视图创建的本地实体化视图的支持。
- 提升了从各个表检索数据并在联接条件中调用 Lambda 用户定义的函数的查询的性能。

- 增加了对数据共享 MV 上的 MV 的支持
- DROP DATABASE 命令现在使用后台工作进程在提交后删除领导节点目录。使用工作进程可避免出现与目录清理相关的罕见错误。

# Amazon Redshift 补丁 193

此修补程序中的集群版本：

- 1.0.136890：尾随 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 10 月 17 日发布
- 1.0.127211：当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本，已于 2025 年 9 月 30 日发布

## 此补丁中的新功能和改进功能

- 增加了对使用 CREATE TABLE LIKE 创建表的支持，以便从远程数据库（包括同一个集群中以及不同集群之间）克隆一个父表。
- 增加了 ST_GeomFromGeoJSON 重载，用于从 GeoJSON 格式的 SUPER 数据类型创建几何体，支持最大 1 MB 的几何体。
- 当自动工作负载管理（WLM）中出现配置错误时，更改为默认回退选项。目前，默认回退选项是自动 WLM，而不是手动 WLM。
- 通过采用最新的 IANA 时区数据库补丁更新了时区计算。此更改会更改某些时区和时段的日期和时间转换的运作方式。有关更多信息，请参阅 Amazon Redshift 使用最新的 IANA 时区数据库。
- 增加了对多维数据布局（MDDL）排序的支持。MDDL 排序键根据实际查询筛选条件对数据进行动态排序，从而提升查询性能。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的多维数据布局存储。

# Amazon Redshift 补丁 192

此修补程序中的集群版本：

- 1.0.124422 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 9 月 22 日发布
- 1.0.122914 – 早先 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 9 月 16 日发布

- 1.0.121035 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 8 月 22 日发布
- 1.0.119650 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 8 月 7 日发布
- 1.0.118447 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 7 月 24 日发布

### 此补丁中的新功能和改进功能

- 修复了当整理后的不区分大小写的列和常量字符串列上的 JOIN 选择不匹配的哈希函数时出现的问题。

## Amazon Redshift 补丁 191

此修补程序中的集群版本：

- 1.0.117891 – 早先 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 7 月 29 日发布
- 1.0.117891 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 7 月 22 日发布
- 1.0.117458 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 7 月 15 日发布
- 1.0.116415 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 6 月 26 日发布
- 1.0.116263 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 6 月 25 日发布
- 1.0.115865 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 6 月 18 日发布

### 此补丁中的新功能和改进功能

- 在 SELECT 语句的 FROM 子句中添加了对 UNNEST 操作的支持，使客户能够将数组元素转换为单独的行。

- 改进了在公用表表达式（CTE）和递归 CTE 中使用 SUPER 数据类型时对 PartiQL 取消嵌套语法的错误处理，尤其是在解析具有相同名称的表别名时。
- 通过添加对大型表的增量自动分析的支持，提高了自动分析的性能。
- Amazon Redshift 现在支持嵌套实体化视图（MV）的级联刷新。
- Amazon Redshift 现在支持自动刷新在 Apache Iceberg 表上定义的实体化视图（MV）。
- 添加了对 Amazon RDS for Oracle 与 Amazon Redshift 的零 ETL 集成的支持。
- 添加了对 Amazon RDS for PostgreSQL 与 Amazon Redshift 的零 ETL 集成的支持。

# Amazon Redshift 补丁 190

此修补程序中的集群版本：

- 1.0.115140 – 早先 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 6 月 16 日发布
- 1.0.115564 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 6 月 14 日发布
- 1.0.113436 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 5 月 15 日发布
- 1.0.112895 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 5 月 7 日发布

## 此补丁中的新功能和改进功能

- 为 Amazon Redshift Serverless 人工智能驱动的扩展和优化添加了改进的模型。
- 在启用增强型 VPC 路由的情况下使用 Amazon Redshift 时，添加了对自动复制的支持。
- 添加了对多个自动表优化操作的支持，以便跨不同的表并发运行。
- 修复了流式摄取中的一种罕见情况，即排队的自动刷新查询会触发也在排队的 UNDO 查询，从而导致锁争用。
- 添加了对 Oracle Database@AWS 与 Amazon Redshift 的零 ETL 集成的支持。

# Amazon Redshift 补丁 189

此修补程序中的集群版本：

- 1.0.112790 – 早先 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 5 月 14 日发布
- 1.0.112790 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 5 月 6 日发布
- 1.0.112086 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 4 月 29 日发布
- 1.0.111040 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 4 月 16 日发布
- 1.0.109768 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 3 月 22 日发布
- 1.0.109284 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 3 月 13 日发布
- 1.0.108971 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 3 月 7 日发布

### 此补丁中的新功能和改进功能

- 为使用 SUPER、GEOMETRY 和 GEOGRAPHY 数据类型的写入查询添加了对并发扩展的支持。
- 修复了在使用 CREATE TABLE AS 创建表期间支持将 SUPER、GEOMETRY 或 GEOGRAPHY 列设置为 DISTKEY 或 SORTKEY 列的问题。
- 添加了对 TRY_CAST 函数的支持。
- 支持在不同的表中并发运行多个 vacuum 命令。

## Amazon Redshift 补丁 188

此修补程序中的集群版本：

- 1.0.109616 – 早先 Amazon Redshift Serverless 工作组版本 – 于 2025 年 3 月 27 日发布
- 1.0.109616 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 3 月 24 日发布
- 1.0.108470 – 早先 Amazon Redshift Serverless 工作组版本 – 于 2025 年 3 月 13 日发布
- 1.0.108470 – 当前 Amazon Redshift Serverless 工作组版本 – 于 2025 年 3 月 11 日发布
- 1.0.108950 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 3 月 10 日发布

- 1.0.108790 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 3 月 9 日发布
- 1.0.108470 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 3 月 4 日发布
- 1.0.107910 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 2 月 20 日发布
- 1.0.107360 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 2 月 13 日发布
- 1.0.106767 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 2 月 5 日发布

### 此补丁中的新功能和改进功能

- 增加了对自动装载 Amazon S3 表类数据存储服务目录的支持，使您可以更轻松地对在 Amazon S3 表类数据存储服务中管理的 Apache Iceberg 表运行查询。有关所需权限，请参阅《AWS Lake Formation Developer Guide》中的 Prerequisites for managing Amazon Redshift namespaces in the AWS Glue Data Catalog。
- 现在，您可以绕过数据掩蔽检查，并将带有动态数据掩蔽（DDM）附件的表添加到数据共享。
- 不支持使用跳过查询块的关联引用，也称为“跳过级别相关性引用”。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的不支持的关联子查询模式。
- 通过数据共享和 3 部分表示法添加了对关联 SUPER 数据类型查询的支持。
- 添加了对 EXCLUDE SQL 关键字的支持。
- 添加了对新的 GROUP BY ALL SQL 关键字的支持。
- 提高了配置了角色的工作负载管理（WLM）队列的性能，从而减少了锁争用。
- 添加了对开启历史记录模式的零 ETL 表上自动刷新的实体化视图的支持。
- 添加了对零 ETL 复制期间在字符串中嵌入的 null 字符的支持。
- 改进了适用于 AI 驱动的扩展和优化的模型。
- 将 `username` 列添加到 `sys_query_history` 视图中。

## Amazon Redshift 补丁 187

此修补程序中的集群版本：

- 1.0.107351 – Amazon Redshift 预置集群早先版本跟踪版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 2 月 13 日发布
- 1.0.106452 – Amazon Redshift 预置集群尾随跟踪版本 – 于 2025 年 2 月 5 日发布
- 1.0.106980 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 2 月 3 日发布
- 1.0.106452 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 1 月 24 日发布
- 1.0.106073 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 1 月 21 日发布
- 1.0.105722 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 1 月 10 日发布
- 1.0.105373 – 当前 Amazon Redshift 预置集群版本和 Amazon Redshift Serverless 工作组版本 – 于 2025 年 1 月 8 日发布
- 1.0.104930 – Amazon Redshift Serverless 工作组版本 – 于 2024 年 12 月 20 日发布

## 此补丁中的新功能和改进功能

- 引入两个新的空间 H3 函数（H3_Center 和 H3_Boundary）。
- 在并发扩展集群上启用动态磁盘缓存。
- 修复了在摄取已排序表期间内存使用效率低下的问题，此问题可能会影响性能。
- 修复了一个问题，该问题导致在涉及单个 SUPER 值的特定场景中，SELECT 子句中某些包含子查询的查询（例如 `SELECT ARRAY_FLATTEN((SELECT FROM ...))`）会错误地触发 `size >= min_partiql_size` XCHECK 错误。
- 修复了一个问题，该问题导致在不可为 Null 的 SUPER 表达式（`json_serialize()`、`json_size()`、`json_typeof()`、`is_object()` 等）与参数表达式（例如 `CASE ... END` 或 `COALESCE()`）组合使用时，有时会生成错误结果。
- 修复了一个问题，即在视图上启用行级安全性后，从后期绑定视图中执行 SELECT 会引发 ERROR。
- 修复了零 ETL 集成上的 CDC 会导致领导节点 CPU 消耗较高的问题。
- 向零 ETL 集成添加了查询所有状态的表的功能，包括在更新期间。
- 向零 ETL 集成添加了截断大型文本/字符串的功能，以便映射到 Amazon Redshift 上的最大 varchar 大小。

- 向零 ETL 集成添加了将无效的 UTF-8 字符替换为您选择的指定字符的功能。
- 修复了一个与 Amazon RDS for MySQL 的零 ETL 集成的“刷新间隔”相关的问题。
- 添加了对 JSON/JSONB 数据类型的支持，以便与 Aurora PostgreSQL 进行零 ETL 集成。
- 修复了一个问题，该问题导致包含带 `IS [NOT] {TRUE|FALSE|UNKNOWN}` 的表达式的查询会遇到断言错误。
- 修复了一个问题，该问题导致在诸如访问不存在的数组元素或属性、JSON `'null'` 值或空 JSON 字符串这样的情况下，`json_extract_path_text()` 和 `json_extract_array_element_text()` 函数会生成空字符串 `''` 而不是 NULL，反之亦然。
- 通过让写入同一个表的并发事务中的 INSERT/COPY 语句共享锁定并取得进展，直到它们需要向表写入数据，来提高这些语句在此类并发事务中的性能。

# Amazon Redshift 补丁 186

此修补程序中的集群版本：

- 1.0.82096 – 尾随版本跟踪版本 – 于 2025 年 1 月 10 日发布
- 1.0.82000 – Amazon Redshift Serverless 版本 – 于 2025 年 1 月 10 日发布
- 1.0.81981 – 当前跟踪版本 – 于 2025 年 1 月 10 日发布
- 1.0.81475 – 尾随版本跟踪版本 – 于 2025 年 1 月 6 日发布
- 1.0.81473 – Amazon Redshift Serverless 版本 – 于 2025 年 1 月 6 日发布
- 1.0.81462 – 当前跟踪版本 – 于 2025 年 1 月 6 日发布
- 1.0.80643 - 尾随跟踪版本 – 于 2024 年 12 月 17 日发布
- 1.0.80583 – Amazon Redshift Serverless 版本 – 于 2024 年 12 月 17 日发布
- 1.0.80560 – 当前跟踪版本 – 于 2024 年 12 月 17 日发布
- 1.0.80498 – Amazon Redshift Serverless 版本 – 于 2024 年 12 月 13 日发布
- 1.0.80491 – 当前跟踪版本 – 于 2024 年 12 月 13 日发布
- 1.0.80036 – Amazon Redshift Serverless 版本 – 于 2024 年 12 月 6 日发布
- 1.0.80009 – 当前跟踪版本 – 于 2024 年 12 月 6 日发布
- 1.0.79372 – Amazon Redshift Serverless 版本 – 于 2024 年 11 月 26 日发布
- 1.0.79237 – Amazon Redshift Serverless 版本 – 于 2024 年 11 月 24 日发布
- 1.0.79229 – 当前跟踪版本 – 于 2024 年 11 月 24 日发布

- 1.0.79003 – Amazon Redshift Serverless 版本 – 于 2024 年 11 月 19 日发布
- 1.0.78987 - 当前跟踪版本 – 于 2024 年 11 月 19 日发布
- 1.0.78890 - Amazon Redshift Serverless 版本 – 于 2024 年 11 月 18 日发布
- 1.0.78881 - 当前跟踪版本 – 于 2024 年 11 月 18 日发布
- 1.0.78646 – Amazon Redshift Serverless 版本 – 于 2024 年 11 月 14 日发布
- 1.0.78641 – 当前跟踪版本 – 于 2024 年 11 月 14 日发布
- 1.0.78178 - Amazon Redshift Serverless 版本 – 于 2024 年 11 月 12 日发布
- 1.0.78160 - 当前跟踪版本 – 于 2024 年 11 月 12 日发布
- 1.0.77809 – Amazon Redshift Serverless 版本 – 于 2024 年 10 月 31 日发布
- 1.0.77777 – 当前跟踪版本 – 于 2024 年 10 月 31 日发布
- 1.0.77292 – Amazon Redshift Serverless 版本 – 于 2024 年 10 月 24 日发布
- 1.0.77272 – 当前跟踪版本 – 于 2024 年 10 月 24 日发布
- 1.0.77040 – Amazon Redshift Serverless 版本 – 于 2024 年 10 月 22 日发布
- 1.0.77028 – 当前跟踪版本 – 于 2024 年 10 月 22 日发布

## 此补丁中的新功能和改进功能

- 增加了对表的实体化视图自动和增量刷新的支持，这些表来自 Amazon Aurora MySQL、Amazon Aurora PostgreSQL、Amazon RDS for MySQL 和 Amazon DynamoDB 的零 ETL 集成。
- 改进 Amazon Redshift 使用相关的单行子查询（例如 `SELECT … WHERE key = (SELECT … correlated subquery)`）重写查询的方式。请注意，只有当子查询生成单行结果时，此类查询才有效。由于改进了重写，某些违反此条件的查询现在可能会失败，并显示错误消息 ERROR: single-row subquery returns more than one row，而在以前允许这种情况。为避免出现这种情况，可能需要纠正此类子查询，以便保证它们仅返回单行，例如，通过添加 MIN() 或 MAX() 聚合函数来保证。
- 在 Amazon Redshift 中添加 SQL 标识符“KAFKA”，以支持通过直接流式摄取从外部 Kafka 源流式传输到 Amazon Redshift。这些源包括外部 Kafka 源，例如 Confluent Managed Cloud 和 Apache Kafka。
- 使用数据共享添加对多仓库写入的支持，以便可以根据工作负载需求，使用不同类型和大小的不同仓库，扩展写入工作负载并提高提取、转换、加载（ETL）工作负载的性能。
- 现在，您可以使用 Amazon Redshift 对在 Amazon S3 表类数据存储服务中管理的 Apache Iceberg 表运行 `SELECT` 查询。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用 Amazon Redshift 访问 Amazon S3 表类数据存储服务。

- 在 Amazon Redshift 集群中添加了对跨数据库写入查询的支持，以便您可以在 Amazon Redshift 集群中跨数据库进行写入。有关更多信息，请参阅跨数据库查询。

# Amazon Redshift 补丁 185

此修补程序中的集群版本：

- 1.0.79878 – 尾随跟踪版本 – 于 2024 年 12 月 3 日发布
- 1.0.79868 – Amazon Redshift Serverless 版本 – 于 2024 年 12 月 3 日发布
- 1.0.79845 – 当前跟踪版本 – 于 2024 年 12 月 3 日发布
- 1.0.78946 – 尾随跟踪版本 – 于 2024 年 11 月 19 日发布
- 1.0.78354 – 尾随跟踪版本 – 于 2024 年 11 月 12 日发布
- 1.0.78130 – Amazon Redshift Serverless 版本 – 于 2024 年 11 月 12 日发布
- 1.0.78125 – 当前跟踪版本 – 于 2024 年 11 月 12 日发布
- 1.0.78016 – Amazon Redshift Serverless 版本 – 于 2024 年 11 月 5 日发布
- 1.0.78014 – 当前跟踪版本 – 于 2024 年 11 月 5 日发布
- 1.0.77707 – Amazon Redshift Serverless 版本 – 于 2024 年 11 月 4 日发布
- 1.0.77687 – 当前跟踪版本 – 于 2024 年 11 月 4 日发布
- 1.0.77467 – Amazon Redshift Serverless 版本 – 于 2024 年 11 月 4 日发布
- 1.0.77433 – 当前跟踪版本 – 于 2024 年 11 月 4 日发布
- 1.0.77467 – Amazon Redshift Serverless 版本 – 于 2024 年 10 月 29 日发布
- 1.0.77433 – 当前跟踪版本 – 于 2024 年 10 月 29 日发布
- 1.0.76991 – Amazon Redshift Serverless 版本 – 于 2024 年 10 月 21 日发布
- 1.0.76913 – 当前跟踪版本 – 于 2024 年 10 月 21 日发布
- 1.0.76645 – Amazon Redshift Serverless 版本 – 于 2024 年 10 月 14 日发布
- 1.0.76642 – 当前跟踪版本 – 于 2024 年 10 月 14 日发布
- 1.0.76242 – Amazon Redshift Serverless 版本 – 于 2024 年 10 月 10 日发布
- 1.0.76230 – 当前跟踪版本 – 于 2024 年 10 月 10 日发布

## 此补丁中的新功能和改进功能

- 增加了对在数据湖表上创建的实体化视图（MV）的增量刷新支持。

- 引入了对更改实体化视图的分配键和排序键的支持。
- 增加了对 Amazon Redshift 机器学习（ML）与 Amazon Bedrock 集成的支持，以便通过简单的 SQL 命令和 Amazon Redshift 中的数据利用大型语言模型（LLM）。
- 修复了一个 Bug，该 Bug 允许在处理 Amazon Redshift 空间数据中的空 shapefile 时不出错。
- 修复了一个 Bug，该 Bug 可导致 varbit 和 varbinary 数据类型在零 ETL 集成中将空字符串值摄取为 ""，而不是 NULL。
- 修复了结果缓存的争用条件，在零 ETL 集成中，该争用条件会导致跨数据库查询返回陈旧的结果。
- 优化了零 ETL 集成重新同步过程，缩短重新同步时间。
- 缩短了恢复和系统维护操作后的零 ETL 引导时间。现在，系统可以在最后一次查询操作后恢复 CDC，而不是恢复到最近可用的 CDC 点。
- 通过一个新的系统表 `sys_integration_table_activity`，提高了零 ETL 集成的可观测性。此表跟踪零 ETL 集成表的插入、更新和删除操作。
- 改善了自动创建联合角色的体验。Amazon Redshift 管理员现在可以在联合用户登录期间对联合角色的自动创建进行更多控制。现在，您可以为每个身份提供商启用、禁用、应用筛选器和配置自动创建设置。
- IAM Identity Center 用户可使用 S3 访问授权，以 IAM Identity Center 凭证运行 COPY、UNLOAD 和 CREATE LIBRARY。
- 自动复制（COPY JOB），支持从 Amazon S3 持续摄取文件，现已正式发布。

# Amazon Redshift 补丁 184

此修补程序中的集群版本：

- 1.0.77706 – 尾随跟踪版本 – 于 2024 年 11 月 12 日发布
- 1.0.76832 - 早先版本跟踪版本 – 于 2024 年 10 月 17 日发布
- 1.0.76169 – Amazon Redshift Serverless 版本 – 于 2024 年 10 月 10 日发布
- 1.0.76142 – 当前跟踪版本 – 于 2024 年 10 月 10 日发布
- 1.0.75677 – Amazon Redshift Serverless 版本 – 于 2024 年 9 月 27 日发布
- 1.0.75672 – 当前跟踪版本 – 于 2024 年 9 月 27 日发布
- 1.0.75504 – Amazon Redshift Serverless 版本 – 于 2024 年 9 月 23 日发布
- 1.0.75449 – 当前跟踪版本 – 于 2024 年 9 月 24 日发布
- 1.0.74765 – Amazon Redshift Serverless 版本 – 于 2024 年 9 月 12 日发布

- 1.0.74754 – 当前跟踪版本 – 于 2024 年 9 月 12 日发布

## 此补丁中的新功能和改进功能

- 用于流式数据摄取的 Amazon Redshift 流式实体化视图（MV）已将 VARBYTE 列的大小从 1024000 字节增加到更大。Amazon Redshift 现在可以从 Amazon Kinesis Data Streams 中摄取高达 1048576 字节（即 1MiB）的记录。Redshift 可以从 Amazon Managed Streaming for Apache Kafka 中摄取高达 16777216 字节（即 16 MiB）的记录。如果您使用的是 ATA SQL 命令 `ALTER TABLE <target_tbl> APPEND FROM <streaming_mv>`，请删除并重新创建 <target_tbl> 以获得相应的较大 VARBYTE 列大小。
- Amazon Redshift 数据共享现在可以包含引用 AWS Glue Data Catalog 的 Amazon S3 数据湖表和视图，包括由 Lake Formation 管理的表。
- 增加了对来自与 DynamoDB 的零 ETL 集成表的实体化视图自动和增量刷新的支持。
- 增加了对配置零 ETL 集成表上的“刷新间隔”的支持，以指定 Amazon Redshift 上的复制刷新率。可以在为新集成创建数据库时设置，也可以更改现有集成的数据库。
- 在 Amazon Redshift 流式摄取中为 Amazon Managed Streaming for Apache Kafka 增加了对双向传输层安全（mTLS）身份验证的支持。
- 增加了对查询哈希的支持，这是基于查询的文本表示及其参数值的 SQL 查询的唯一标识符。它可用于识别、分组和分析类似查询。现在可以在 SYS_QUERY_HISTORY 视图中找到查询哈希，并增加了两个新列：
  - `user_query_hash` – 用户提交的哈希值，包括查询文本。
  - `generic_query_hash` – 用户提交的哈希值，不含任何查询文本。
- 修复了零 ETL 在运行 CDC 复制和扫描时，利用结果缓存执行表查询的罕见情况下的系统死锁。
- 解决了工作负载管理（WLM）中的一个问题，即当用于 Python 用户定义函数（UDF）查询的 WLM 资源不可用时，Python UDF 查询会优先于其它查询。
- 解决了工作负载管理（WLM）无法将查询路由到映射到新创建用户角色的队列的问题。
- 提高了查询大型生产者表的小型数据共享使用者的磁盘利用率。
- 提高了预置集群的 INSERT 语句性能，可弹性调整到更大的大小。

# Amazon Redshift 补丁 183

此修补程序中的集群版本：

- 1.0.75655 - 早先版本跟踪版本 – 于 2024 年 9 月 30 日发布

- 1.0.75388 – Amazon Redshift Serverless 版本 – 于 2024 年 9 月 25 日发布
- 1.0.75379 – 当前跟踪版本 – 于 2024 年 9 月 25 日发布
- 1.0.74967 – Amazon Redshift Serverless 版本 – 于 2024 年 9 月 17 日发布
- 1.0.74927 – 当前跟踪版本 – 于 2024 年 9 月 17 日发布
- 1.0.74518 – Amazon Redshift Serverless 版本 – 于 2024 年 9 月 11 日发布
- 1.0.74503 – 当前跟踪版本 – 于 2024 年 9 月 11 日发布
- 1.0.74223 – Amazon Redshift Serverless 版本 – 于 2024 年 9 月 5 日发布
- 1.0.74159 – 当前跟踪版本 – 于 2024 年 9 月 5 日发布
- 1.0.74126 – Amazon Redshift Serverless 版本 – 于 2024 年 8 月 30 日发布
- 1.0.74097 – 当前跟踪版本 – 于 2024 年 8 月 30 日发布
- 1.0.73016 - Amazon Redshift Serverless 版本 – 于 2024 年 8 月 8 日发布
- 1.0.72982 - 当前跟踪版本 – 于 2024 年 8 月 8 日发布

## 此补丁中的新功能和改进功能

- 通过 SVV_DATABASE_PRIVILEGES 和 SVV_SCHEMA_PRIVILEGES 增加了对发现限定范围权限的支持。此外还将列 privilege_scope 引入到 SVV_DATABASE_PRIVILEGES 和 SVV_SCHEMA_PRIVILEGES。
- 提高了在分组列具有的独特值（NDV，Number of Distinct Value）较少时，执行不同聚合操作的查询的性能。
- 提高了预置数据仓库的 INSERT/COPY 语句的性能，数据仓库的大小可弹性调整为 2 倍或更大。
- 支持动态数据掩蔽政策内的会话上下文变量。
- 添加将子查询和视图作为 MERGE 语句源的支持。
- 对于预置并发扩展和无服务器自动扩缩计算，支持包含 MERGE 语句的存储过程。
- 对于 COPY 命令以及在经历调整大小的数据仓库，在工作负载管理中可更好地进行资源预测，从而提高查询性能。
- 在可用内存有限的集群中，改进了对内存不足错误的恢复能力
- 添加了对 COPY 命令中非 ASCII 字符作为字段分隔符的支持。
- 添加了对使用 COPY 命令摄取以 ISO-8859-1 字符集编码的数据的支持。
- 在已指定 URI 时，移除了在 MSK 外部架构定义中指定 CLUSTER_ARN 的要求。
- 支持在对零 ETL 集成表进行扫描期间应用范围扫描筛选条件。

- 支持向零 ETL 集成表添加排序键。
- 在创建零 ETL 集成数据库时，支持要使用 CREATE DATABASE 语句指定的诸如 `serializable` 和 `collation` 等数据库选项。
- 修复了在零 ETL 集成中数据筛选条件超过 2 KB 时导致集群重新启动的问题。

# Amazon Redshift 补丁 182

此修补程序中的集群版本：

- 1.0.73589 – 尾随跟踪版本 – 于 2024 年 8 月 22 日发布
- 1.0.73359 – Amazon Redshift Serverless 版本 – 于 2024 年 8 月 15 日发布
- 1.0.73348 – 当前跟踪版本 – 于 2024 年 8 月 15 日发布
- 1.0.72917 – Amazon Redshift Serverless 版本 – 于 2024 年 8 月 12 日发布
- 1.0.72899 – 当前跟踪版本 – 于 2024 年 8 月 12 日发布
- 1.0.72528 - Amazon Redshift Serverless 版本 – 于 2024 年 8 月 7 日发布
- 1.0.72503 - 当前跟踪版本 – 于 2024 年 8 月 8 日发布
- 1.0.72239 - Amazon Redshift Serverless 版本 – 于 2024 年 8 月 1 日发布
- 1.0.71714 – Amazon Redshift Serverless 版本 – 于 2024 年 7 月 24 日发布
- 1.0.71629 – 当前跟踪版本 – 于 2024 年 7 月 24 日发布
- 1.0.70953 – Amazon Redshift Serverless 版本 – 于 2024 年 7 月 11 日发布
- 1.0.70890 – 当前跟踪版本 – 于 2024 年 7 月 11 日发布
- 1.0.70716 – Amazon Redshift Serverless 版本 – 于 2024 年 7 月 8 日发布
- 1.0.70695 – 当前跟踪版本 – 于 2024 年 7 月 8 日发布
- 1.0.69945 – Amazon Redshift Serverless 版本 – 于 2024 年 6 月 27 日发布
- 1.0.69938 – 当前跟踪版本 – 于 2024 年 6 月 27 日发布

## 此补丁中的新功能和改进功能

- 将 LISTAGG、MEDIAN、PERCENTILE_CONT 和 PERCENTILE_DISC 更新为不再需要用户定义的表。引用目录表或未引用任何表的查询也可以使用这些函数。
- 针对高并发工作负载，通过在单个会话中整合多个查询的临时表，缩短数据共享读取查询的查询计划时间。

- 全面提供 Redshift 机器学习（ML）与 Amazon SageMaker AI Jumpstart 的集成，以使用您自己的大型语言模型。
- 在 Redshift ML 中引入了对 SUPER 输入和输出数据类型的支持。
- 当目标表受动态数据掩蔽政策保护并在 JOIN 子句中引用目标表时，支持带有 JOIN 子句的 UPDATE 语句。
- 支持在 Redshift 上查询零 ETL 数据库，即使已从源中删除此集成。
- 修复了可能导致零 ETL 集成失败的复制错误。这使得集成更具韧性。
- 支持除创建零 ETL 集成的用户以外的用户在通过 GRANT 获得权限后查询数据。
- 修复了一个内存不足问题，该问题可能导致在启用零 ETL 集成的 Amazon Redshift 预置集群中重启集群。
- 支持从源 RDS Multi-AZ 数据库集群中创建 RDS for MySQL 与 Redshift 的零 ETL 集成。多可用区数据库集群是 Amazon RDS 的半同步、高可用性部署模式，具有两个可读副本数据库实例。
- 通过在将 Amazon Redshift 流式实体化视图与 Amazon MSK 主题关联所需的外部架构定义中指定 Amazon MSK 集群的代理 URI，使用户能够从 Amazon Redshift 流式使用者客户端连接到 Amazon MSK 集群。此功能不再需要在 Amazon MSK 集群上通过互联网网关调用 GetBootStrapBroker API，来获取 Amazon MSK 引导代理节点名称。
- 解决了恢复 Amazon Redshift Serverless 实例的问题，支持现有数据库用户使用 IAM Identity Center 身份验证方法在 Amazon Redshift 查询编辑器 v2 中连接数据库时恢复无服务器实例。
- 通过转向基于表的分片，优化 CDC 复制并降低了 Redshift 计算的资源利用率。
- 通过改善工作负载管理（WLM）中的资源预测，提高查询性能。
- 修复了查询在并发扩展集群上失败并显示消息 WLM 队列已用完，无法重启的问题。
- 修复了一个工作负载管理（WLM）问题：当客户尝试应用无效的 WLM 配置时，Amazon Redshift 回退到手动 WLM。
- 支持数据共享使用者运行读取查询，即使生产者因计划内维护或计划外停机而停机也是如此。
- 修复极少出现的集群重启问题，当聚合数据（例如，COUNT(DISTINCT) 聚合函数）的查询中使用了 ANY_VALUE 函数时会出现该问题。

# Amazon Redshift 补丁 181

此修补程序中的集群版本：

- 1.0.72031 - 当前跟踪版本 – 于 2024 年 8 月 1 日发布

- 1.0.71912 – 尾随跟踪版本 – 于 2024 年 8 月 1 日发布
- 1.0.70665 – Amazon Redshift Serverless 版本 – 于 2024 年 7 月 8 日发布
- 1.0.70634 – 当前跟踪版本 – 于 2024 年 7 月 8 日发布
- 1.0.69954 – Amazon Redshift Serverless 版本 – 于 2024 年 6 月 26 日发布
- 1.0.69952 – 当前跟踪版本 – 于 2024 年 6 月 26 日发布
- 1.0.69497 – Amazon Redshift Serverless 版本 – 于 2024 年 6 月 18 日发布
- 1.0.69451 – 当前跟踪版本 – 于 2024 年 6 月 18 日发布
- 1.0.69076 – Amazon Redshift Serverless 版本 – 于 2024 年 6 月 14 日发布
- 1.0.69065 – 当前跟踪版本 – 于 2024 年 6 月 14 日发布
- 1.0.68555 – Amazon Redshift Serverless 版本 – 于 2024 年 5 月 31 日发布
- 1.0.68540 – 当前跟踪版本 – 于 2024 年 5 月 31 日发布
- 1.0.68328 – Amazon Redshift Serverless 版本 – 于 2024 年 5 月 23 日发布
- 1.0.68205 – 当前跟踪版本 – 于 2024 年 5 月 23 日发布
- 1.0.67796 – Amazon Redshift Serverless 版本 – 于 2024 年 5 月 15 日发布
- 1.0.67788 – 当前跟踪版本 – 于 2024 年 5 月 15 日发布
- 1.0.67308 – Amazon Redshift Serverless 版本 – 于 2024 年 5 月 1 日发布
- 1.0.67305 – 当前跟踪版本 – 于 2024 年 5 月 1 日发布

## 此补丁中的新功能和改进功能

- 引入了对“lower_attribute_names()”和“upper_attribute_names()”函数的支持，这些函数修改 SUPER 对象值的属性名称的大小写。
- 修复了使用身份列时 CREATE TABLE LIKE 中的一个问题。以前，新表将继承源表的标识符。如果稍后删除源表，这会导致问题，因为标识符在新表中会失效。
- 修复了导致某些外部表无法在 SVV_ALL_TABLES 中显示的问题。
- 缩短集群引导时间，并加快高并发工作负载的查询初始化速度。
- 修复了在将 split_part() 函数传递到联合源直至 RDS 和 Aurora MySQL 时会导致错误的联合查询问题
- 对于预调配并发扩展集群和无服务器自动扩缩计算，支持用户通过 ALTER TABLE...ALTER DISTSTYLE KEY DISTKEY 命令发起对分配键的更改。
- 对于预调配并发扩展和无服务器自动扩缩计算，支持手动刷新且涉及聚合的实体化视图。

- 增加了对零 ETL 的支持，来处理大小不超过 16 MB 的记录，并支持最大为 16 MB 的 SUPER 值。
- 通过提供诸如架构和表名称等其它详细信息，增强在 Aurora MySQL 中使用零 ETL 进行初始同步期间的错误消息。
- 引入了对使用 Amazon Redshift ML CREATE MODEL 进行标记的支持。通过这项改进，您现在可以标记 Amazon Redshift ML 使用的 Amazon SageMaker AI 资源。标记可帮助您管理、识别、组织、搜索和筛选资源。
- 通过使用 AWS Lambda 优化数据处理，提高涉及 Lambda 用户定义函数（UDF）的查询性能。
- 降低在经弹性调整大小的无服务器集群的排序表中摄取数据期间的内存利用率。
- 添加对视图 SYS_QUERY_HISTORY 的列 `query_text` 中换行符（\n）和视图 SYS_QUERY_TEXT 中列 `text` 的支持。

# Amazon Redshift 补丁 180

此修补程序中的集群版本：

- 1.0.68520 – 尾随跟踪版本 – 于 2024 年 5 月 28 日发布
- 1.0.67699 – 尾随跟踪版本 – 于 2024 年 5 月 15 日发布
- 1.0.66960 – 早先版本跟踪版本 – 于 2024 年 4 月 21 日发布
- 1.0.66954 - 当前跟踪版本 – 于 2024 年 4 月 21 日发布
- 1.0.66276 - 当前跟踪版本 – 于 2024 年 4 月 12 日发布
- 1.0.66290 - Amazon Redshift Serverless 版本 – 于 2024 年 4 月 10 日发布
- 1.0.63590 – 当前跟踪版本 – 于 2024 年 2 月 19 日发布
- 1.0.63567 – Amazon Redshift Serverless 版本 – 于 2024 年 2 月 16 日发布
- 1.0.63282 – Amazon Redshift Serverless 版本 – 于 2024 年 2 月 13 日发布
- 1.0.63269 – 当前跟踪版本 – 于 2024 年 2 月 13 日发布
- 1.0.63215 – Amazon Redshift Serverless 版本 – 于 2024 年 2 月 12 日发布
- 1.0.63205 – 当前跟踪版本 – 于 2024 年 2 月 12 日发布
- 1.0.63030 – Amazon Redshift Serverless 版本 – 于 2024 年 2 月 7 日发布
- 1.0.62913 – 当前跟踪版本 – 于 2024 年 2 月 7 日发布
- 1.0.62922 – Amazon Redshift Serverless 版本 – 于 2024 年 2 月 5 日发布
- 1.0.62878 – 当前跟踪版本 – 于 2024 年 2 月 5 日发布

- 1.0.62698 – Amazon Redshift Serverless 版本 – 于 2024 年 1 月 31 日发布
- 1.0.62614 – 当前跟踪版本 – 于 2024 年 1 月 31 日发布
- 1.0.61687 – Amazon Redshift Serverless 版本 – 于 2024 年 1 月 5 日发布
- 1.0.61678 – 当前跟踪版本 – 于 2024 年 1 月 5 日发布
- 1.0.61567 – Amazon Redshift Serverless 版本 – 于 2023 年 12 月 31 日发布
- 1.0.61559 – 当前跟踪版本 – 于 2023 年 12 月 31 日发布
- 1.0.61430 – Amazon Redshift Serverless 版本 – 于 2023 年 12 月 29 日发布
- 1.0.61395 – 当前跟踪版本 – 于 2023 年 12 月 29 日发布

## 此补丁中的新功能和改进功能

- 对 CURRENT_USER 进行了更改，不再将返回的用户名截断为 64 个字符。
- 增加了对标准视图和后期绑定视图应用数据掩蔽策略的功能。
- 添加了将动态数据掩蔽 (DDM) 应用于 SUPER 数据类型列中的标量属性的功能。
- 添加 OBJECT_TRANSFORM SQL 函数。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 OBJECT_TRANSFORM 函数。
- 添加了对嵌套数据应用 AWS Lake Formation 精细访问控制以及使用 Amazon Redshift 数据湖分析进行查询的功能。
- 添加 INTERVAL 数据类型。
- 添加 CONTINUE_HANDLER，这是一种控制存储过程流程的异常处理程序。使用此处理程序，您可以在不结束现有语句块的情况下捕获和处理异常。
- 添加了在单个对象之外，对范围（架构或数据库）定义权限的功能。这样便可以向用户和角色授予对范围内所有当前和未来对象的权限。
- 添加了从数据共享创建数据库的功能，且其权限允许使用者端管理员将共享数据库上对象的单独权限，授予使用者端用户和角色。
- 添加了对来自远程 BYOM 模型的 SUPER 返回数据类型的支持。这扩大了可接受的 SageMaker AI 模型的范围，以包括那些具有更复杂返回格式的模型。
- 对外部函数进行了更改，现在可将带或不带小数部分的数字隐式转换为列的数值数据类型。对于 int2、int4 和 int8 列，现在可通过截断小数来接受带有小数位数的数字，除非数字超出范围。对于 float4 和 float8 列，接受不带小数位数的数字。
- 添加三个与 H3 分层地理空间索引网格系统配合使用的空间函数：H3_FromLongLat、H3_FromPoint 和 H3_Polyfill。

# Amazon Redshift 补丁 179

此修补程序中的集群版本：

- 1.0.62317 – Amazon Redshift Serverless 版本 – 于 2024 年 1 月 29 日发布
- 1.0.62312 – 尾随版本跟踪版本 – 于 2024 年 1 月 29 日发布
- 1.0.61631 – Amazon Redshift Serverless 版本 – 于 2024 年 1 月 5 日发布
- 1.0.61626 – 当前跟踪版本 – 于 2024 年 1 月 5 日发布
- 1.0.61191 – 当前跟踪版本 – 于 2023 年 12 月 16 日发布
- 1.0.61150 – Amazon Redshift Serverless 版本 – 于 2023 年 12 月 16 日发布
- 1.0.60982 – Amazon Redshift Serverless 版本 – 于 2023 年 12 月 13 日发布
- 1.0.60854 – 当前跟踪版本 – 于 2023 年 12 月 10 日发布
- 1.0.60354 - Amazon Redshift Serverless 版本 – 于 2023 年 11 月 22 日发布
- 1.0.60353 - 当前跟踪版本 – 于 2023 年 11 月 21 日发布
- 1.0.60293 - Amazon Redshift Serverless 版本 – 于 2023 年 11 月 21 日发布
- 1.0.60292 - 当前跟踪版本 – 于 2023 年 11 月 22 日发布
- 1.0.60161 - Amazon Redshift Serverless 版本 – 于 2023 年 11 月 18 日发布
- 1.0.60140 - 当前跟踪版本 – 于 2023 年 11 月 18 日发布
- 1.0.60139 - Amazon Redshift Serverless 版本 – 于 2023 年 11 月 18 日发布
- 1.0.59947 - Amazon Redshift Serverless 版本 – 于 2023 年 11 月 16 日发布
- 1.0.59945 - 当前跟踪版本 – 于 2023 年 11 月 16 日发布
- 1.0.59118 - Amazon Redshift Serverless 版本 – 于 2023 年 11 月 9 日发布
- 1.0.59117 - 当前跟踪版本 – 于 2023 年 11 月 9 日发布

## 此补丁中的新功能和改进功能

- 添加了支持，使具有适当权限的联合用户可以查看行级安全视图和动态数据掩蔽系统视图，包括：
  - SVV_ATTACHED_MASKING_POLICY
  - SVV_MASKING_POLICY
  - SVV_RLS_ATTACHED_POLICY
  - SVV_RLS_POLICY

  - SVV_RLS_RELATION
- 添加了功能，使在 FROM 子句中仅包含标量函数的查询现在会导致错误。
- 在并发扩展集群中添加了带有永久目标表功能的 CREATE TABLE AS (CTAS) 语句。并发扩展集群现在支持更多查询。
- 添加了以下系统表，用于在 RA3 集群上运行经典大小调整后跟踪表的重新分配状态：
  - SYS_RESTORE_STATE 系统表显示了表级别的重新分配进度。
  - SYS_RESTORE_LOG 系统表显示了数据重新分配的历史吞吐量。
- 改进了在 RA3 节点类型上运行经典大小调整后，EVEN 表上的切片偏斜最小化。这也适用于运行经典大小调整的补丁 178 集群。
- 在并发扩展集群上添加了对带 EXTENSION 的 UNLOAD 的支持。
- 提高了在 HashJoins 和 NestLoop 联接中包含 Λ UDF 的查询的性能。
- 提高了 RA3 节点类型禁用弹性调整大小的性能。
- 提高了数据共享查询的性能。
- 在弹性调整大小的预置集群和无服务器工作组中，提高了手动启动的分析查询的性能。
- 通过改善工作负载管理中的资源预测，提高自动 WLM 查询性能。
- 删除在专用租赁 VPC 中启动集群的功能。此更改不会影响 VPC 中任何 EC2 实例的租赁。您可以使用 `modify-vpc-tenancy` AWS CLI 命令将 VPC 的租赁修改为默认值。
- 预置并发扩展集群和无服务器自动扩缩计算现在支持实体化视图手动刷新。
- 在 EXTRACT 函数中添加了对 INTERVAL 文本的支持。例如，`EXTRACT('hours' from Interval '50 hours')` 之所以返回 2，是因为将 50 小时解释为 2 天零 2 小时，并且提取了小时部分 2。

# Amazon Redshift 补丁 178

此修补程序中的集群版本：

- 1.0.63327 - 当前跟踪版本 – 于 2024 年 2 月 9 日发布
- 1.0.63313 – 尾随跟踪版本 – 于 2024 年 2 月 9 日发布
- 1.0.60977 - 尾随跟踪版本 – 于 2023 年 12 月 15 日发布
- 1.0.59596 - 当前跟踪版本 – 于 2023 年 11 月 9 日发布
- 1.0.58593 - Amazon Redshift Serverless 版本 – 于 2023 年 10 月 23 日发布
- 1.0.58558 - 当前跟踪版本 – 于 2023 年 10 月 23 日发布

- 1.0.57864 - 当前跟踪版本 – 于 2023 年 10 月 12 日发布
- 1.0.57850 - Amazon Redshift Serverless 版本 – 于 2023 年 10 月 12 日发布
- 1.0.56952 - 当前跟踪版本 – 于 2023 年 9 月 25 日发布
- 1.0.56970 - Amazon Redshift Serverless 版本 – 于 2023 年 9 月 25 日发布

## 此补丁中的新功能和改进功能

- 现在，Amazon Redshift 通过加快使用者实例上的元数据刷新速度来提高数据共享查询性能，同时在生产者实例上发生并发数据更改。
- 添加了对以下功能的支持：当实体化视图的基表引用共享数据时，对 Amazon Redshift 数据共享使用者实例上的实体化视图进行自动增量刷新。
- 添加了对以 SUPER 数据类型存储大小不超过 16MB 的大型对象的支持。从 JSON、PARQUET、TEXT 和 CSV 源文件中摄取时，您可以将半结构化数据或文档加载为 SUPER 数据类型的值，最大为 16MB。
- 添加了对弹性调整大小的支持，可在单节点 Amazon Redshift RA3 集群中进行扩展。
- 单节点 Amazon Redshift RA3 集群现在可以受益于加密增强功能，从而缩短了加密过程中的总体加密时间，并提高了加密期间数据仓库的可用性。
- 改进了在取消嵌套和取消透视以 SUPER 数据类型存储的数据时对查询的支持。
- 提高了刷新具有 SUPER 数据类型的实体化视图的性能。
- 添加了对使用 ANY_VALUE 函数聚合 INTERVAL 文本的支持。
- 流式摄取现在支持以下新的 SQL 命令来清流式数据：`DELETE FROM streaming_materialized_views WHERE <where filter clause>`。
- DECODE 函数将一个特定值替换为另一个特定值或默认值，具体取决于等式条件的结果。DECODE 现在需要以下三个参数：
  - expression
  - 搜索
  - 结果
- 向存储过程添加了功能，以允许捕获数据溢出数据类型转换错误，并在异常处理块内进行处理。
- 现在，如果您将 enable_case_sensitive_identifier 更改为与会话默认设置不同，则在查询行级别安全性或受动态数据掩蔽保护的关系时会收到错误消息。此外，当在预调配集群或无服务器命名空间中应用行级别安全性或动态数据掩蔽策略时，将阻止以下配置：

  `ALTER USER <current_user> SET case-sensitive identifier.`

- MERGE 命令现在支持一种简化的语法，该语法只需要目标表和源表。有关更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的 MERGE。
- 添加了对将相同的动态数据掩蔽策略附加到具有相同优先级或未指定优先级的多个用户或角色的支持。
- 现在，您可以通过 ALTER TABLE ADD COLUMN 在添加新列时指定 COLLATION。
- 修复了在并发扩展集群和 Amazon Redshift Serverless 上延迟执行 QMR 规则的问题。
- Amazon Redshift 联合查询在 Amazon RDS for PostgreSQL 和 Amazon Aurora PostgreSQL 上扩展了对带有时间戳的时区的下推支持。
- 现在，可以将以数字开头的 Amazon RDS for MySQL 和 Aurora MySQL 数据库名称与联合身份查询结合使用。
- 添加了 SYS_ANALYZE_HISTORY 视图，其中包含 ANALYZE 操作的记录详细信息。
- 添加了 SYS_ANALYZE_COMPRESSION_HISTORY 视图，其中包含 COPY 或 ANALYZE COMPRESSION 命令期间压缩分析操作的记录详细信息。
- 添加了 SYS_SESSION_HISTORY 视图，其中包含与活动的会话、历史会话和重启会话相关的记录详细信息。
- 添加了 SYS_TRANSACTION_HISTORY 视图，其中包含与事务级别分析相关的记录详细信息，这一分析提供提交所花费的时间、提交的数据块数和隔离级别。
- 添加了 SVV_REDSHIFT_SCHEMA_QUOTA 视图，其中包含与限额相关的记录，以及数据库中每个模式的当前磁盘使用情况。
- 添加了 SYS_PROCEDURE_CALL 视图，该视图包含与存储过程调用相关的记录，包括开始时间、结束时间、存储过程调用的状态以及嵌套存储过程调用的调用层次结构。
- 添加了 SYS_CROSS_REGION_DATASHARING_USAGE 视图，其中包含与跟踪跨区域数据共享使用情况相关的记录。
- 添加了 SYS_PROCEDURE_MESSAGES 视图，其中包含与记录的存储过程消息的跟踪信息相关的记录。
- 添加 SYS_UDF_LOG 视图，其中包含与跟踪来自用户定义函数调用、错误、警告或跟踪（如果适用）的系统日志消息相关的记录。
- 向 SYS_EXTERNAL_QUERY_DETAIL 添加了新列 IS_RECURSIVE、IS_NESTED、S3LIST_TIME 和 GET_PARTITION_TIME。
- 添加了 MaxRPU，这是面向 Redshift Serverless 的新计算成本控制设置。使用 MaxRPU，您可以选择指定计算资源阈值上限，以此来控制不同时间点的数据仓库成本，其做法是选择 Redshift Serverless 可以为每个工作组扩展的最大计算资源水平。

- 将 INTERVAL 文本的输出更改为数字式的间隔字符串。例如，指定为 INTERVAL '1' YEAR 的间隔现在将返回 1 YEAR 而不是 "00:00:00。此外，INTERVAL 文本的输出会被截断为指定的最小 INTERVAL 分量。例如，INTERVAL '1 day 1 hour 1 minute 1.123 seconds' HOUR TO MINUTE 将被截断为 1 day 01:01:00。

# Amazon Redshift 补丁 177

此修补程序中的集群版本：

- 1.0.57922 - 早先版本跟踪版本 – 于 2023 年 10 月 12 日发布
- 1.0.57799 - Amazon Redshift Serverless 版本 – 于 2023 年 10 月 10 日发布
- 1.0.57798 - 当前跟踪版本 – 于 2023 年 10 月 10 日发布
- 1.0.57085 - 早先版本跟踪版本 – 于 2023 年 9 月 26 日发布
- 1.0.56899 - Amazon Redshift Serverless 版本 – 于 2023 年 9 月 21 日发布
- 1.0.56754 - 当前跟踪版本 – 于 2023 年 9 月 21 日发布
- 1.0.56242 - 当前跟踪版本 – 于 2023 年 9 月 11 日发布
- 1.0.55539 - Amazon Redshift Serverless 版本 – 于 2023 年 8 月 28 日发布
- 1.0.55524 - 当前跟踪版本 – 于 2023 年 8 月 28 日发布
- 1.0.54899 - 当前跟踪版本 – 于 2023 年 8 月 15 日发布
- 1.0.54899 - 当前跟踪版本 – 于 2023 年 8 月 14 日发布
- 1.0.54899 - 当前跟踪版本 – 于 2023 年 8 月 15 日发布
- 1.0.54239 - 当前跟踪版本 – 于 2023 年 8 月 3 日发布
- 1.0.54321 - Amazon Redshift Serverless 版本 – 于 2023 年 8 月 3 日发布

## 此补丁中的新功能和改进功能

- 添加了 SYS_MV_STATE 视图，该视图对于实体化视图的每次状态转换包含一行。SYS_MV_STATE 可用于 Amazon Redshift Serverless 和 Amazon Redshift 预置实例的 MV 刷新监控。
- 添加了 SYS_USERLOG 视图，该视图记录为创建用户、删除用户、更改用户（重命名）、更改用户（更改属性）而对数据库用户所做更改的详细信息。
- 添加了 SYS_COPY_REPLACEMENTS 视图，该视图显示当无效的 UTF-8 字符由带有 ACCEPTINVCHARS 选项的 COPY 命令替换时所记录的日志。

- 添加了 SYS_SPATIAL_SIMPLIFY 视图，该视图包含有关使用 COPY 命令简化空间几何对象的信息。
- 添加了 SYS_VACUUM_HISTORY 视图，您可以使用该视图查看 VACUUM 操作的详细信息和结果。
- 添加了 SYS_SCHEMA_QUOTA_VIOLATIONS 视图，该视图记录超出架构限额时的匹配项、时间戳、XID 和其他有用信息。
- 添加了 SYS_RESTORE_STATE 视图，在异步经典大小调整期间，您可以使用该视图监控集群中每个表的再分配进度。
- 添加了 SYS_EXTERNAL_QUERY_ERROR 视图，该视图返回有关 Redshift Spectrum 扫描错误的信息。
- 在 CREATE MODEL 命令中添加了标签参数，因此，您现在可以在自动驾驶训练任务中跟踪训练成本。
- 为 Amazon Redshift 集群添加了自定义域名 (CNAME)。
- 添加了对 Apache Iceberg 的预览支持，使客户能够对 Amazon Redshift 中的 Apache Iceberg 表运行分析查询。
- 添加了对在工作负载管理 (WLM) 中使用用户角色以及参数组的支持。
- 添加了对自动安装 AWS Glue Data Catalog 的支持，使客户可以更轻松地在其数据湖中运行查询。
- 添加了相关功能，以便使用不带 GROUP BY 子句的分组函数或在 WHERE 子句中使用分组操作会导致错误。
- 向存储过程添加了功能，以允许捕获除以零错误并在异常处理块内进行处理。
- 修复了一个错误：当源表是数据共享表时，查询无法使用并发扩展将数据写入表。
- 修复了 enable_case_sensitive_identifier 中记录的区分大小写的标识符，现在可以用于 MERGE 语句。
- 修复了一个错误：对函数 pg_get_late_binding_view_cols() 的查询偶尔会被忽略。现在，您始终可以取消此类查询。
- 提高了在生产者上运行 vacuum 任务时在使用者上运行的数据共享查询的性能。
- 提高了数据共享和并发扩展查询的性能，尤其是在生产者上进行并行数据更改或分载到与使用者相连的并发扩展实例时。

# Amazon Redshift 补丁 176

此修补程序中的集群版本：

- 1.0.56738 - 早先版本跟踪版本 – 于 2023 年 9 月 21 日发布
- 1.0.55837 - 早先版本跟踪版本 – 于 2023 年 9 月 11 日发布
- 1.0.54776 - 当前跟踪版本 – 于 2023 年 8 月 15 日发布
- 1.0.54052 - 当前跟踪版本 – 于 2023 年 7 月 26 日发布
- 1.0.53642 - Amazon Redshift Serverless 版本 – 于 2023 年 7 月 20 日发布
- 1.0.53301 - 当前跟踪版本 – 于 2023 年 7 月 20 日发布
- 1.0.52943 - Amazon Redshift Serverless 版本 – 于 2023 年 7 月 7 日发布
- 1.0.52931 - 当前跟踪版本 – 于 2023 年 7 月 7 日发布
- 1.0.52194 - Amazon Redshift Serverless 版本 – 于 2023 年 6 月 21 日发布
- 1.0.51986 - 当前跟踪版本 – 于 2023 年 6 月 16 日发布
- 1.0.51594 - 当前跟踪版本 – 于 2023 年 6 月 9 日发布

## 此补丁中的新功能和改进功能

- 改进了为空的分组集编写 GROUP BY () 时的错误处理。此前曾被忽略，现在返回解析器错误。
- 增强了使用 SUPER 列以增量方式刷新实体化视图的性能。
- ALTER TABLE <target_tbl> APPEND FROM <streaming_mv> – (ATA) SQL 命令现在支持将所有记录从作为源的流式实体化视图 (MV) 以及作为源的表移至目标表。在流式 MV 上支持 ATA，可允许用户通过将流式 MV 中的所有记录移到另一个表来快速清除这些记录，以此来管理数据增长。
- TRUNCATE <streaming_mv> – SQL 命令现在支持截断流式实体化视图 (MV) 以及表中的所有记录。TRUNCATE 会删除流式 MV 中的所有记录，同时保持流式 MV 结构不变。在流式 MV 上运行 TRUNCATE 可让客户快速清除流式 MV 中的所有记录，以管理数据增长。
- 在 SELECT 命令中添加了 QUALITY 子句的功能。
- 通过与 Amazon Forecast 集成，Redshift 机器学习支持时间序列预测。
- AWS Glue Data Catalog支持 自动挂载来简化数据湖的查询，无需额外步骤来创建外部架构引用。
- 现在支持更改 RLS 策略。有关更多详细信息，请参阅 ALTER RLS POLICY 中的文档。
- Lambda UDF 现在支持 CREATE FUNCTION 语句中的 STABLE 函数不稳定性参数。如果在 CREATE FUNCTION 语句中使用了 STABLE 参数，并使用相同的参数多次调用 Lambda UDF，则 Lambda UDF 函数的预期调用次数会减少。CREATE FUNCTION 参数中更详细地说明了 STABLE 函数不稳定性类别。
- 多项 Lambda UDF 性能改进。具体而言，改进了查询受行级别安全性 (RLS) 策略保护的表时的记录批处理支持。

- 缩短了 Amazon Redshift RA3 集群的总体加密时间，并提高了加密期间数据仓库的可用性。有关更多信息，请参阅 Amazon Redshift 数据库加密。
- Redshift 中添加了新的系统视图 SYS_MV_REFRESH_HISTORY。SYS_MV_REFRESH_HISTORY 视图包含与实体化视图的刷新活动相对应的一行。使用 SYS_MV_REFRESH_HISTORY，您可以检查实体化视图的刷新历史记录。SYS_MV_REFRESH_HISTORY 对所有用户均可见。超级用户可以查看所有行；普通用户只能查看其自己的数据。

  系统视图 SYS_QUERY_DETAIL 中添加了一个新列 SPILLED_BLOCK_LOCAL_DISK。新列 SPILLED_BLOCK_LOCAL_DISK 有助于客户确定溢出到本地磁盘的块。可以使用 SYS_QUERY_DETAIL 在步骤级别查看查询的详细信息。SYS_QUERY_DETAIL 对所有用户可见。超级用户可以查看所有行；普通用户只能查看其具有访问权限的元数据。
- Amazon Redshift Serverless 中添加了新的系统视图 SYS_QUERY_TEXT，且预调配了 Amazon Redshift。SYS_QUERY_TEXT 视图类似于预调配集群的 SVL_STATEMENTTEXT。使用 SYS_QUERY_TEXT 视图中的 sequence 列获取完整的 SQL 语句文本。

# Amazon Redshift 补丁 175

此修补程序中的集群版本：

- 1.0.53064 - 当前跟踪版本 – 于 2023 年 7 月 7 日发布
- 1.0.51973 - 当前跟踪版本 – 于 2023 年 6 月 16 日发布
- 1.0.51781 - 当前跟踪版本 – 于 2023 年 6 月 10 日发布
- 1.0.51314 - Amazon Redshift Serverless 版本 – 于 2023 年 6 月 3 日发布
- 1.0.51304 - 当前跟踪版本 – 于 2023 年 6 月 2 日发布
- 1.0.50708 - 当前跟踪版本 – 于 2023 年 5 月 19 日发布
- 1.0.50300 - 当前跟踪版本 – 于 2023 年 5 月 8 日发布
- 1.0.49710 - Amazon Redshift Serverless 版本 – 于 2023 年 4 月 28 日发布
- 1.0.49676 - 当前跟踪版本 – 于 2023 年 4 月 28 日发布

## 此补丁中的新功能和改进功能

- 次要错误修复。
- Amazon Redshift 流式摄取现在支持跨区域流式摄取，其中，您的来源 Amazon Kinesis Data Streams (KDS) 或 Amazon Managed Streaming for Apache Kafka (MSK) 主题可以位于与您的

Amazon Redshift 数据仓库所在 AWS 区域不同的 AWS 区域。开始使用 Amazon Kinesis Data Streams 流式摄取中的文档已经过修订，解释了如何使用 REGION 关键字。

- 埃及夏令时调整。
- 缩短了 RA3 集群的总体加密时间。

# Amazon Redshift 补丁 174

## 1.0.51296 – 于 2023 年 6 月 2 日发布

发布到早先版本跟踪。无版本注释。

## 1.0.50468 – 于 2023 年 5 月 12 日发布

维护版本。无版本注释。

## 1.0.49780、1.0.49868 和 1.0.49997 – 于 2023 年 4 月 28 日发布

此版本的版本注释：

- 改进了对 Lambda UDF 的批处理支持。
- Lambda UDF 的增量批处理。
- 新的 MERGE SQL 命令可将源数据更改应用于 Amazon Redshift 表。
- 新的动态数据掩蔽功能可简化在 Amazon Redshift 数据仓库中保护敏感数据的过程。
- 用于与 Lake Formation 共享数据的新集中式访问控制，允许使用 Lake Formation API 和 AWS 控制台对 Amazon Redshift 数据共享中的表和视图进行权限授予、查看访问控制和审核权限等管理。
- 埃及夏令时调整。

## 1.0.49087 – 于 2023 年 4 月 12 日发布

维护版本。无版本注释。

## 1.0.48805 – 于 2023 年 4 月 5 日发布

此版本的版本注释：

- Amazon Redshift 使用 BYTEDICT，为字符串密集型查询引入了额外的性能增强，BYTEDICT 是 Amazon Redshift 中的一种新的压缩编码，与 LZO 或 ZSTD 等其他压缩编码相比，可将基于字符串

的数据处理速度提高 5 到 63 倍。有关此功能的更多信息，请参阅《Amazon Redshift 数据库开发人员指南》中的压缩编码。

### 1.0.48004 – 于 2023 年 3 月 17 日发布

维护版本。无版本注释。

### 1.0.47470 – 于 2023 年 3 月 11 日发布

此版本的版本注释：

- 提高了 pg_catalog.svv_table_info 上的查询性能。还添加了新列 create_time。创建表时，此列以 UTC 格式存储日期/时间戳。
- 增加了对在联合查询上指定会话级别超时的支持。

## Amazon Redshift 补丁 173

### 1.0.49788 – 于 2023 年 4 月 28 日发布

此版本的版本注释：

- 埃及夏令时调整。

### 1.0.49074 – 于 2023 年 4 月 12 日发布

此版本的版本注释：

- 时区配置已更新为 IANA 库版本 2022g。

### 1.0.48766 – 于 2023 年 4 月 5 日发布

维护版本。无版本注释。

### 1.0.48714 – 于 2023 年 4 月 5 日发布

维护版本。无版本注释。

## 1.0.48022 – 于 2023 年 3 月 17 日发布

维护版本。无版本注释。

## 1.0.47357 – 于 2023 年 3 月 7 日发布

维护版本。无版本注释。

## 1.0.46987 – 于 2023 年 2 月 24 日发布

维护版本。无版本注释。

## 1.0.46806 – 于 2023 年 2 月 18 日发布

维护版本。无版本注释。

## 1.0.46607 – 于 2023 年 2 月 13 日发布

此版本的版本注释：

- 现在，如果具有手动设置的交错排序键的表的分配方式已设置为 DISTSTYLE KEY，我们会自动将这些表转换为复合排序键，以提高它们的性能。这是在将快照还原到 Amazon Redshift Serverless 时完成的。

## 1.0.45698 – 于 2023 年 1 月 20 日发布

此版本的版本注释：

- 向 UNLOAD 命令添加文件扩展名参数，这样，文件扩展名就会自动添加到文件名中。
- 默认情况下，在将受 RLS 保护的对象添加到数据共享或者它们已经是数据共享的一部分时，支持对其进行保护。管理员现在可以对数据共享关闭 RLS，以允许使用者访问受保护的对象。
- 增加了新的系统表以监控：SVV_ML_MODEL_INFO、SVV_MV_DEPENDENCY 和 SYS_LOAD_DETAIL。还将 data_skewness 和 time_skewness 列添加到系统表 SYS_QUERY_DETAIL 中。

# Amazon Redshift 补丁 172

此修补程序中的集群版本：

- 1.0.46534 – 于 2023 年 2 月 18 日发布
- 1.0.46523 – 于 2023 年 2 月 13 日发布
- 1.0.46206 – 于 2023 年 2 月 1 日发布
- 1.0.45603 – 于 2023 年 1 月 20 日发布
- 1.0.44924 – 于 2022 年 12 月 19 日发布
- 1.0.44903 – 于 2022 年 12 月 18 日发布
- 1.0.44540 – 于 2022 年 12 月 13 日发布
- 1.0.44126 – 于 2022 年 11 月 23 日发布
- 1.0.43980 – 于 2022 年 11 月 17 日发布

### 此补丁中的新功能和改进功能

- 默认情况下，由 CTAS 创建的表为 AUTO。
- 在实体化视图中增加了行级别安全性 (RLS) 支持。
- 增加了 S3 超时以改善跨区域数据共享。
- 增加了新的空间函数 ST_GeomFromGeohash。
- 改进了从复合主键中自动选择分配键的功能，以提高即时可用的性能。
- 将自动主键添加到具有复合主键的表的分配键，以提高即时可用的性能。
- 改进了并发扩展，即使在数据发生变化时也能扩展更多查询。
- 提高了数据共享查询性能。
- 为分类模型增加了机器学习概率指标。
- 增加了新的系统表以进行监控：SVV_USER_INFO、SVV_MV_INFO、SYS_CONNECTION_LOG、SYS_DATASHARE_USAGE_PRODUCER 和 SYS_DATASHARE_CHANGE_LOG。
- 添加了对在外部表中查询 Parquet 和 ORC 文件类型的 VARBYTE 列的支持。

## Amazon Redshift 补丁 171

此修补程序中的集群版本：

- 1.0.43931 – 于 2022 年 11 月 16 日发布
- 1.0.43551 – 于 2022 年 11 月 5 日发布

- 1.0.43331 – 于 2022 年 9 月 29 日发布
- 1.0.43029 – 于 2022 年 9 月 26 日发布

### 此补丁中的新功能和改进功能

- CONNECT BY 支持：增加了对 CONNECT BY SQL 构造的支持，使您可以基于数据集内的父子关系递归查询数据仓库中的分层数据。

## Amazon Redshift 补丁 170

此修补程序中的集群版本：

- 1.0.43922 – 于 2022 年 11 月 21 日发布
- 1.0.43573 – 于 2022 年 11 月 7 日发布
- 1.0.41881 – 于 2022 年 9 月 20 日发布
- 1.0.41465 – 已于 2022 年 9 月 7 日发布
- 1.0.40325 – 于 2022 年 7 月 27 日发布

### 此补丁中的新功能和改进功能

- ST_GeomfromGeoJSON：从采用 GeoJSON 表示形式的 VARCHAR 构造 Amazon Redshift 空间几何体对象。

## Amazon Redshift 补丁 169

此修补程序中的集群版本：

- 1.0.41050 – 于 2022 年 9 月 7 日发布
- 1.0.40083：于 2022 年 7 月 16 日发布
- 1.0.39734：于 2022 年 7 月 7 日发布
- 1.0.39380：于 2022 年 6 月 23 日发布
- 1.0.39251：于 2022 年 6 月 15 日发布
- 1.0.39009：于 2022 年 6 月 8 日发布

## 此补丁中的新功能和改进功能

- 添加角色作为 Alter Default Privileges 命令的参数，以支持基于角色的访问控制。
- 添加 ACCEPTINVCHARS 参数，以支持在从 Parquet 和 ORC 文件复制时替换无效的 UTF-8 字符。
- 添加 OBJECT(k,v) 函数，以从键值对构建 SUPER 对象。

# Amazon Redshift 补丁 168

此修补程序中的集群版本：

- 1.0.38698 – 于 2022 年 5 月 25 日发布
- 1.0.38551 – 于 2022 年 5 月 20 日发布
- 1.0.38463 – 于 2022 年 5 月 18 日发布
- 1.0.38361 – 于 2022 年 5 月 13 日发布
- 1.0.38199：于 2022 年 5 月 9 日发布
- 1.0.38112 — 于 2022 年 5 月 6 日发布
- 1.0.37684 — 于 2022 年 4 月 20 日发布

## 此补丁中的新功能和改进功能

- 在 Amazon Redshift ML 中添加对线性学习器模型类型的支持。
- 为 SQL 事务隔离级别添加了 SNAPSHOT 选项。
- 添加 `farmhashFingerprint64` 作为 `VARBYTE` 和 `VARCHAR` 数据的新哈希算法。
- 在实体化视图的增量刷新中支持 AVG 函数。
- 支持 Redshift Spectrum 中对外部表的相关子查询。
- 为了提高开箱即用的查询性能，Amazon Redshift 会自动为特定表选择单列主键作为分发键。

# Amazon Redshift 中的行为更改

随着 Amazon Redshift 不断发展和改进，为了增强性能、提高安全性和改善用户体验，我们引入了一些行为更改。此页面包含全面的资源，让您可以随时了解这些关键更新，采取必要措施，并避免对工作负载造成中断。

## 即将发生的行为更改

以下描述了即将发生的行为更改。

主题


- 从 2026 年 9 月 30 日起，Amazon Redshift 会为暂停的数据共享创建者集群自动拍摄快照
- 从补丁 206 开始，SUPER 数据类型支持更大的单个字符串
- 2026 年 8 月 17 日之后，强制执行 Amazon Redshift Serverless API 的 AWS KMS 密钥权限
- 从补丁 208 开始，Amazon Redshift 会在多次登录尝试失败后强制执行用户锁定
- Amazon Redshift Serverless 和 Amazon Redshift RG 实例上手动快照的增强计费模式于 2026 年 6 月 8 日生效
- 从补丁 202 开始，Lake Formation 表上的 Iceberg DELETE 需要 DELETE 权限
- 从补丁 202 开始，Amazon Redshift Serverless 在快照还原时保留零 ETL 和 S3 事件集成
- 2026 年 12 月 31 日终止对 Amazon Redshift ODBC 1.x 驱动程序的支持
- 标量 Python UDF 将在 2026 年 6 月 30 日之后终止支持
- 2026 年 2 月 27 日之后的实体化视图（MV）自动 REFRESH 行为更改
- 2026 年 2 月 16 日之后，Amazon Redshift 将不支持通过数据共享访问使用者信息的函数
- 传输层安全性协议（TLS）最低版本变更自 2026 年 10 月 31 日起生效
- 2025 年 10 月 30 日之后，Amazon Redshift 将不支持创建新的标量 Python UDF


### 从 2026 年 9 月 30 日起，Amazon Redshift 会为暂停的数据共享创建者集群自动拍摄快照

从 2026 年 9 月 30 日起，Amazon Redshift 会为暂停的数据共享创建者集群自动拍摄快照。由于使用者可向暂停的创建者集群写入数据，这类快照能够捕获数据共享写入，并提供恢复点。

如果您会暂停数据共享创建者集群，则会受到此变更影响。

在此之前，Amazon Redshift 不会为暂停状态下的此类集群自动拍摄快照，因此可用于恢复的最新自动快照仅为集群暂停那一刻拍摄的快照。现在，集群暂停期间将持续创建自动快照。

这些快照遵循集群现有的自动快照计划和保留期，您无需执行任何操作。对于并非数据共享创建者的暂停集群，Amazon Redshift 在集群暂停期间不会自动拍摄快照。

有关更多信息，请参阅 自动快照。

## 从补丁 206 开始，SUPER 数据类型支持更大的单个字符串

从补丁 206 开始，Amazon Redshift 支持在 SUPER 数据类型（用于存储半结构化数据）中存储多达 16000000 字节的单个字符串。此限制也适用于内存中字符串操作。要使用此功能，请创建数据库参数组并将 enable_large_strings_opt_in 参数设置为 Yes。然后，将参数组附加到您的数据仓库。对于预调配集群，您必须重启集群才能应用参数更改。

如果您使用 SUPER 数据类型来存储大型 JSON 文档，则此更改可能会影响您。如果您的工作负载在处理长度超过 65535 字节的字符串的视图、用户定义函数（UDF）或存储过程中依赖于 VARCHAR(MAX)，也可能会受到影响。

如果将 enable_large_strings_opt_in 设置为 Yes，则内存中操作会将 VARCHAR(MAX) 解析为 VARCHAR(16000000)。这些操作包括 CAST、视图、UDF 和存储过程。然后，它们就可以处理长度超过 65535 个字节的字符串。此更改不会影响 VARCHAR 列的持久存储。采用 VARCHAR(MAX) 的 CREATE TABLE 会继续创建 VARCHAR(65535) 列。当您将参数设置为 No 时，Amazon Redshift 会保留数据仓库的现有行为。要检查数据仓库的当前状态，请运行 SELECT is_large_string_enabled();。

当您将 enable_large_strings_opt_in 设置为 Yes 时：

- 通过 COPY 和 INSERT 操作写入到 SUPER 列时，支持超过之前 65535 字节的大小限制。诸如 JSON_PARSE、JSON_SERIALIZE 和 LISTAGG 之类的字符串函数也支持这些较大的大小。
- 现有视图、UDF 和存储过程在重新创建之前，其大小上限仍为 65535 字节。
- 如果现有视图使用 VARCHAR(MAX)，而其它对象（例如其它视图）依赖于该视图，则 CREATE OR REPLACE VIEW 将失败。错误为“cannot change data type of view column”。要更新视图，请删除依赖对象，删除并重新创建视图，然后重新创建依赖对象。对于使用 VARCHAR(MAX) 且没有依赖对象的视图，无需这些额外步骤即可替换它们。
- 在数据共享生产者和使用者数据仓库之间使此设置保持一致。

当您将 enable_large_strings_opt_in 设置为 No 时：

- 您无法将长度超过 65535 字节的字符串通过 COPY 或 INSERT 写入 SUPER 列。您仍然可以通过 SUPER 读取现有字符串。但是，强制转换为 VARCHAR(MAX) 会将该值截断为 65535 字节。
- 诸如 JSON_PARSE、JSON_SERIALIZE 和 LISTAGG 之类的字符串函数无法处理长度超过 65535 字节的 VARCHAR 值。
- 对于处理长度超过 65535 字节的字符串文本的视图、存储过程和 UDF，您无法查询它们。
- 如果您在参数设置为 Yes 时使用 VARCHAR(MAX) 创建了视图，而其它对象（例如其它视图）依赖于该视图，则 CREATE OR REPLACE VIEW 将失败，并显示相同的“cannot change data type of view column”错误。按照上述列表中描述的相同补救步骤进行操作。
- 跨实例快照还原可能无法摄取长的字符串。当快照的 SUPER 列中包含长度超过 65535 字节的字符串时，就会发生此故障。当您将该快照还原到此参数设置为 No 的集群时，就会发生这种情况。

有关补丁版本的更多信息，请参阅 Amazon Redshift 的集群版本。

## 2026 年 8 月 17 日之后，强制执行 Amazon Redshift Serverless API 的 AWS KMS 密钥权限

Amazon Redshift Serverless 现在会执行一项新的授权检查，要求调用特定 Amazon Redshift Serverless API 的 IAM 主体，必须对用于加密目标命名空间的客户自主管理型密钥拥有显式 AWS KMS 权限。以前，Amazon Redshift 使用 Amazon Redshift 服务角色代表您执行 AWS KMS 操作，而不验证发出调用的 IAM 主体本身是否有权使用该密钥。

启用新的授权检查后，缺乏所需 AWS KMS 权限的主体所发出的 API 调用将失败，并显示“ValidationException: "Insufficient KMS permissions."”。如果您使用客户自主管理型密钥加密 Amazon Redshift Serverless 命名空间，并使用 IAM 主体来调用下面列出的受影响 API，则您可能会受到此问题影响。

为避免中断，请检查您的账户中哪些 IAM 主体针对使用客户自主管理型密钥加密的命名空间调用了下面列出的 API，并确保这些主体对密钥拥有所需的 AWS KMS 权限。您可以通过 AWS KMS 密钥策略、基于 IAM 身份的策略或 AWS KMS 授权来授予这些权限。

此更改不会影响使用默认的 AWS 拥有的密钥加密的命名空间。如果您的 IAM 主体已经拥有下面列出的所需 AWS KMS 权限，则无需执行任何操作。

下面列出了受影响的 API 和针对命名空间的客户自主管理型密钥所需的权限：

- CreateWorkgroup、RestoreFromSnapshot、RestoreFromRecoveryPoint：需要 kms:Encrypt、kms:GenerateDataKey、kms:CreateGrant 和 kms:Decrypt。

- `RestoreTableFromSnapshot`、`RestoreTableFromRecoveryPoint`：需要 `kms:Decrypt`。

有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的控制对 AWS KMS 密钥的访问。有关 Amazon Redshift 数据库加密的更多信息，请参阅 Amazon Redshift 数据库加密。

## 从补丁 208 开始，Amazon Redshift 会在多次登录尝试失败后强制执行用户锁定

从 Amazon Redshift 补丁 208 开始，为了增强数据仓库的安全态势，Amazon Redshift 会在数据库用户超过配置的连续登录尝试失败次数后自动锁定该用户。默认情况下，该功能处于启用状态，失败尝试次数的阈值为 5 次（可配置为 2 到 50 次），并且同时适用于预调配集群和无服务器工作组。

如果您使用数据库（基于密码的）连接进行连接，则可能会受到这一问题影响。一旦锁定，即使密码正确，登录尝试也会遭拒绝，直到管理员或超级用户将其解锁。通过 IAM 或 AWS IAM Identity Center 进行身份验证的联合身份不会受到影响，因为它们使用的是短期令牌而非静态密码。

要管理此行为：

- 使用 `ALTER SYSTEM SET max_failed_login_attempts TO value;` 来设置阈值（2–50，默认为 5）。
- 使用 `SHOW USER LOCKOUT;` 来检查所有用户的锁定状态。
- 使用 `ALTER USER username LOGIN PASSWORD 'password';` 来解锁用户，这也会重置失败尝试计数器。

有关更多信息，请参阅 SHOW USER LOCKOUT、ALTER USER 和 max_failed_login_attempts。

## Amazon Redshift Serverless 和 Amazon Redshift RG 实例上手动快照的增强计费模式于 2026 年 6 月 8 日生效

从 2026 年 6 月 8 日起，Amazon Redshift 宣布对 Amazon Redshift Serverless 和 Amazon Redshift RG 实例上的手动快照采用增强计费模式。通过此增强功能，现在可以根据您账户中所有活跃手动快照中存储的唯一数据块数，来对手动快照进行计量。出现在多个快照中的共享数据块仅计数一次，按您所在区域的现有手动快照费率计费。

如果您在 Amazon Redshift Serverless 工作组或 Amazon Redshift RG 集群上使用手动快照，则可能会受到此影响。这样可以降低维护多个快照的客户的手动快照成本。

无需采取行动。增强计费模式同时自动适用于现有和新的手动快照。

有关快照定价的更多信息，请参阅 Amazon Redshift 定价。

## 从补丁 202 开始，Lake Formation 表上的 Iceberg DELETE 需要 DELETE 权限

从 Amazon Redshift 补丁 202 开始，对 Lake Formation（LF）托管表执行 Iceberg DELETE 操作需要 DELETE Lake Formation 权限。UPDATE 和 MERGE 操作要求同时具有 INSERT 和 DELETE 权限。所有 Iceberg DML 操作都需要 ALTER 权限。

如果您对由 AWS Lake Formation 管理的 Iceberg 表执行 DELETE、UPDATE 或 MERGE 操作，则可能会受到此影响。

以前，仅具有 INSERT 权限（而没有 DELETE 权限）的主体可以对 LF 管理的 Iceberg 表执行删除操作。从补丁 202 开始，DELETE 操作需要 DELETE Lake Formation 权限。在显式授予 DELETE 权限之前，仅依赖 INSERT 权限来执行删除操作的主体将收到权限错误。S3 表类数据存储服务不受此更改的影响。

要在此更改后继续执行 DELETE 操作，请查看您对 Iceberg 表授予的 Lake Formation 权限，并确保执行删除操作的主体已获得 DELETE Lake Formation 权限。您可以使用 AWS Lake Formation 控制台或 `aws lakeformation list-permissions` AWS CLI 命令验证现有授权。

如果您需要暂时恢复到之前的行为，请联系 AWS Support 以在集群上禁用此更改，而无需部署代码。

有关补丁版本的更多信息，请参阅 Amazon Redshift 的集群版本。

## 从补丁 202 开始，Amazon Redshift Serverless 在快照还原时保留零 ETL 和 S3 事件集成

从 Amazon Redshift 补丁 202 开始，当您将 Amazon Redshift Serverless 命名空间从快照或恢复点还原到相同的无服务器命名空间时，将自动保留与该命名空间和快照/恢复点关联的零 ETL 和 S3 事件集成。无需执行其它操作。

如果您将快照或恢复点还原到配置了零 ETL 或 S3 事件集成的 Amazon Redshift Serverless 命名空间，则您可能会受到此影响。

以前，将快照或恢复点还原到无服务器命名空间，会将关联的零 ETL 和 S3 事件集成标记为失败，需要您在还原完成后手动重新创建它们。从补丁 202 开始，默认情况下会保留这些集成，并在还原完成后恢复运行。

仅当还原到相同的无服务器命名空间时，此功能才适用于 Amazon Redshift Serverless。将快照还原到其它命名空间不会保留集成。预调配集群上的快照还原不会保留零 ETL 或 S3 事件集成。

要选择退出在还原期间保留集成，请在 AWS 管理控制台中取消选中还原页面上的保留集成框。如果您使用的是 AWS CLI，请在调用 restore-from-snapshot 或 restore-from-recovery-point API 操作时设置 `--no-maintain-integration` 参数。当您选择退出时，集成将在还原后进入 `FAILED` 状态。然后，您可以删除失败的集成并重新创建它们。

有关补丁版本的更多信息，请参阅 Amazon Redshift 的集群版本。

## 2026 年 12 月 31 日终止对 Amazon Redshift ODBC 1.x 驱动程序的支持

从 2026 年 12 月 31 日起，Amazon Redshift 将停止支持 ODBC 1.x 驱动程序。根据客户反馈，我们已将最初的支持终止日期从 2026 年 9 月 30 日延长至 2026 年 12 月 31 日，以便为迁移提供更多时间。这同时适用于 Amazon Redshift 预置集群和无服务器工作组。

如果您使用任何版本的 ODBC 1.x 驱动程序连接到 Amazon Redshift，则可能会受到影响。要验证您使用的是否为 ODBC 1.x 驱动程序，请运行以下查询：

```
SELECT *
FROM SYS_CONNECTION_LOG
WHERE
(driver_version ilike 'Amazon Redshift ODBC Driver 1%'
OR driver_version ilike 'Redshift ODBC Driver 01%'
OR driver_version ilike 'Redshift ODBC Driver 1,%'
)
OR
(application_name ilike 'Amazon Redshift ODBC Driver 1%');
```

要继续获得 Amazon Redshift ODBC 驱动程序连接的技术支持，请在 2026 年 12 月 30 日之前迁移到最新的 Amazon Redshift ODBC 2.x 驱动程序。

在生产环境中迁移到 ODBC 2.x 驱动程序之前，我们建议进行全面的概念验证，以验证新驱动程序满足您的所有功能要求。

我们建议使用最新版本的 Amazon Redshift ODBC 2.x 驱动程序，并设置 ApplicationName 属性，以便在连接到 Amazon Redshift 时标识您的应用程序。

## 标量 Python UDF 将在 2026 年 6 月 30 日之后终止支持

Amazon Redshift 将在 2026 年 6 月 30 日之后终止对 Python UDF 的支持。作为替代方案，我们建议您使用 Lambda UDF。

与 Python UDF 相比，Lambda UDF 具有以下优点：

- Lambda UDF 可以从 UDF 逻辑内部连接到外部服务和 API。
- Lambda UDF 使用 Lambda 计算资源。计算密集型或内存密集型 Lambda UDF 不会影响 Amazon Redshift 的查询性能或资源并发性。
- Lambda UDF 支持运行 Python 代码。Lambda UDF 支持多个 Python 运行时，具体取决于特定的使用案例。有关更多信息，请参阅《AWS Lambda 开发人员指南》中的使用 Python 构建。
- 您可以将自定义代码执行隔离在单独的服务边界中。这可简化维护、监控、预算制定和权限管理。

有关创建和使用 Lambda UDF 的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的标量 Lambda UDF。有关将现有 Python UDF 转换为 Lambda UDF 的信息，请参阅博客文章。

## 2026 年 2 月 27 日之后的实体化视图（MV）自动 REFRESH 行为更改

从 2026 年 2 月 27 日起，Amazon Redshift 实体化视图的自动 REFRESH 查询将作为用户查询而不是后台自主进程执行。因此，自动 REFRESH 查询现在的运行优先级与其它用户查询相同。

与之前的行为相比，此更改提高了启用自动 REFRESH 功能的实体化视图的新鲜度，有助于它们更及时地了解对基表的最新更改。

注意：MV 自动 REFRESH 行为更改功能仅适用于补丁版本 P198 及更高版本的当前跟踪版本上的 Amazon Redshift 预置集群。它目前在无服务器上处于禁用状态。

## 2026 年 2 月 16 日之后，Amazon Redshift 将不支持通过数据共享访问使用者信息的函数

从 2026 年 2 月 16 日起，Amazon Redshift 将不再支持使用 `user_is_member_of` 以及通过数据共享访问使用者用户、角色或组信息的相关函数。

## 传输层安全性协议（TLS）最低版本变更自 2026 年 10 月 31 日起生效

从 2026 年 10 月 31 日起，Amazon Redshift 将强制使用最低的传输层安全性协议（TLS）版本 1.2。使用 TLS 版本 1.0 或 1.1 的传入连接将遭到拒绝。这同时适用于 Amazon Redshift 预置集群和无服务器工作组。未使用 TLS 的 Amazon Redshift 数据仓库将不受此变更的影响。

如果您使用 TLS 版本 1.0 或 1.1 连接到 Amazon Redshift，此更新可能会影响您。

要验证您当前使用的 TLS 版本，您可以：

对于 Amazon Redshift 预置：检查 STL_CONNECTION_LOG 系统表中的 sslversion 列 [1]。

对于 Amazon Redshift Serverless 工作组：检查 SYS_CONNECTION_LOG 系统表中的 ssl_version 列 [2]。

要在此更改后保持对 Amazon Redshift 数据仓库的不间断访问，请按照以下所列步骤操作：

1. 更新您的客户端以支持 TLS 1.2 或更高版本
2. 安装支持 TLS 1.2+ 的最新驱动程序版本

如果可能，我们建议使用最新版本的 Amazon Redshift 驱动程序 [3]。

[1] https://docs.aws.amazon.com/redshift/latest/dg/r_STL_CONNECTION_LOG.html

[2] https://docs.aws.amazon.com/redshift/latest/dg/SYS_CONNECTION_LOG.html

[3] https://docs.aws.amazon.com/redshift/latest/mgmt/configuring-connections.html

## 2025 年 10 月 30 日之后，Amazon Redshift 将不支持创建新的标量 Python UDF

2025 年 10 月 30 日之后，Amazon Redshift 将不再支持创建新的 Python UDF。现有的 Python UDF 将继续正常运行。我们强烈建议您在该日期之前将现有的 Python UDF 迁移至 Lambda UDF。

与 Python UDF 相比，Lambda UDF 具有以下优点：

- Lambda UDF 可以从 UDF 逻辑内部连接到外部服务和 API。
- Lambda UDF 使用 Lambda 计算资源。计算密集型或内存密集型 Lambda UDF 不会影响 Amazon Redshift 的查询性能或资源并发性。
- Lambda UDF 支持运行 Python 代码。Lambda UDF 支持多个 Python 运行时，具体取决于特定的使用案例。有关更多信息，请参阅《AWS Lambda 开发人员指南》中的使用 Python 构建。
- 您可以将自定义代码执行隔离在单独的服务边界中。这可简化维护、监控、预算制定和权限管理。

有关创建和使用 Lambda UDF 的信息，请参阅《Amazon Redshift 数据库开发人员指南》中的标量 Lambda UDF。有关将现有 Python UDF 转换为 Lambda UDF 的信息，请参阅博客文章。

# 最近的行为更改

主题

- 2025 年 8 月 26 日之后，Amazon Redshift 使用最新的 IANA 时区数据库
- Amazon Redshift Serverless RPU 更改将于 2025 年 8 月 15 日之后生效
- 数据库审计日志记录更改在 2025 年 8 月 10 日之后生效
- 无服务器工作组的虚拟私有云端点更改将于 2025 年 6 月 27 日之后生效
- 查询监控更改于 2025 年 5 月 2 日之后生效
- 在 2025 年 1 月 10 日之后生效的安全更改

## 2025 年 8 月 26 日之后，Amazon Redshift 使用最新的 IANA 时区数据库

从 2025 年 8 月 26 日起，Amazon Redshift 通过采用最新的 IANA 时区数据库补丁来计算时区。此更改会更改某些时区和时段的日期和时间转换的运作方式。此更新会影响显式时区转换，例如使用 CONVERT_TIMEZONE 函数或 TIMEZONE 和 AT TIME ZONE 命令执行的转换，以及在类型强制转换操作期间发生的隐式转换，特别是 TIMESTAMP 和 TIMESTAMPTZ 格式之间的转换。

以下是时区和时段组合的更新列表：

- 现在，在 2038 年后，时区可以正确地遵守夏令时（DST）。以前，在 2038 年后，没有时区遵循夏令时。
- 对于 `America/Toronto` 时区以及与之关联的时区，在 1947-1950 年，夏令时在当地时间凌晨 2 点而不是午夜进行切换。
- Amazon Redshift 现在可以正确反映在所有时区标准化之前时段的本地平均时间（LMT）。这个时段是特定于时区的，大多数时区在 19 世纪中叶之前都转向了标准化。
- `EET`、`CET`、`WET` 和 `MET` 现在被视为正常时区而不是缩写。
- Amazon Redshift 中不再存在以下时区名称：
  - `Asia/Riyadh87`
  - `Asia/Riyadh88`
  - `Asia/Riyadh89`
  - `Mideast/Riyadh87`
  - `Mideast/Riyadh88`
  - `Mideast/Riyadh89`
  - `US/Pacific-New`

有关 IANA 时区数据库的更多信息，请参阅 IANA 时区数据库网站上的 Time Zone Database。

## Amazon Redshift Serverless RPU 更改将于 2025 年 8 月 15 日之后生效

从 2025 年 8 月 15 日起，Amazon Redshift Serverless 基本 Redshift 处理器（RPU）的 AWS 账户配额为 3200 个 RPU 或前六个月最大累计基本 RPU 的 1.5 倍（以较大值为准）。

## 数据库审计日志记录更改在 2025 年 8 月 10 日之后生效

从 2025 年 8 月 10 日起，Amazon Redshift 将对数据库审计日志记录进行更改，这需要您采取行动。Amazon Redshift 会将有关数据库中连接和用户活动的信息记录到 Amazon S3 存储桶和 CloudWatch。2025 年 8 月 10 日之后，对于具有用于指定 Redshift IAM USER 的存储桶策略的 Amazon S3 存储桶，Amazon Redshift 将停止数据库审计日志记录。我们建议更新您的策略，以便在 S3 存储桶策略中改用 Redshift SERVICE-PRINCIPAL 来实现审计日志记录。有关审计日志记录的信息，请参阅 Amazon Redshift 审计日志记录的存储桶权限。

为避免任何日志记录中断，请在 2025 年 8 月 10 日之前查看并更新您的 S3 存储桶策略，以便向关联区域中的 Redshift 服务主体授予访问权限。有关数据库审计日志记录的信息，请参阅 Amazon S3 中的日志文件。

如有疑问或疑虑，请通过以下链接联系 AWS 支持人员：AWS Support。

## 无服务器工作组的虚拟私有云端点更改将于 2025 年 6 月 27 日之后生效

从 2025 年 6 月 27 日起，Amazon Redshift 更改虚拟私有云端点（VPCE）对无服务器工作组的支持。在此日期之前，Amazon Redshift 在工作组创建期间将端点部署到单个可用区（AZ），并随着时间推移将 VPCE 支持扩展到多达三个可用区。在此日期之后，Amazon Redshift 将在工作组创建期间指定的多达三个可用区中部署 VPCE。

有关更多信息，请参阅 使用 Amazon Redshift Serverless 时的注意事项。

如有疑问或疑虑，请通过以下链接联系 AWS 支持人员：AWS Support。

主题

- 查询监控更改于 2025 年 5 月 2 日之后生效
- 在 2025 年 1 月 10 日之后生效的安全更改

## 查询监控更改于 2025 年 5 月 2 日之后生效

自 2025 年 5 月 2 日起，我们将不再为现有和新创建的 Redshift Serverless 工作组提供查询限制选项卡中的查询 CPU 时间 (`max_query_cpu_time`) 和查询 CPU 使用率

(max_query_cpu_percentage) 指标。在此日期之后，我们将自动移除所有 Redshift Serverless 工作组中基于这些指标的所有查询限制。

查询限制旨在捕获失控的查询。但是，在查询的生命周期中，查询 CPU 时间 (max_query_cpu_time) 和查询 CPU 使用率 (max_query_cpu_percentage) 可能会有所不同，因此不是用于捕获失控查询的持续有效的方法。要捕获失控的查询，我们建议您利用可提供一致且切实可行的信息的查询监控指标。一些示例包括：

- 查询执行时间 (max_query_execution_time)：确保查询在预期的时间范围内完成。
- 返回行数 (max_scan_row_count)：监控正在处理的数据的规模。
- 查询队列时间 (max_query_queue_time)：识别需要花费时间排队的查询。

有关支持的指标的完整列表，请参阅查询 Amazon Redshift Serverless 的监控指标。

## 在 2025 年 1 月 10 日之后生效的安全更改

安全性一直是 Amazon Web Services（AWS）的重中之重。为此，我们通过引入增强的安全默认值来进一步加强 Amazon Redshift 环境的安全状况，这些默认值有助于您在无需额外设置的情况下遵守数据安全的最佳实践，并降低潜在错误配置的风险。为避免任何潜在的中断，请在生效日期之前查看预置集群和无服务器工作组的创建配置、脚本和工具，以进行必要的更改，使其与新的默认设置保持一致。

### 默认情况下禁用公共访问权限

2025 年 1 月 10 日之后，默认情况下，对于所有新创建的预置集群以及从快照还原的集群禁用公共可访问性。在此版本中，默认情况下，只允许从同一虚拟私有云（VPC）内的客户端应用程序连接到集群。要从其它 VPC 中的应用程序访问数据仓库，请配置跨 VPC 访问。此更改将反映在 CreateCluster 和 RestoreFromClusterSnapshot API 操作以及相应的 SDK 和 AWS CLI 命令中。如果您通过 Amazon Redshift 控制台创建预置集群，则默认情况下，该集群已禁用公共访问权限。

如果您仍然需要公共访问权限，将需要在运行 CreateCluster 或 RestoreFromClusterSnapshot API 操作时覆盖默认值并将 PubliclyAccessible 参数设置为 true。对于可公共访问的集群，建议您必须使用安全组或网络访问控制列表（ACL）来限制访问权限。有关更多信息，请参阅VPC 安全组和为 Amazon Redshift 集群或 Amazon Redshift Serverless 工作组配置安全组通信设置。

## 默认加密

2025 年 1 月 10 日之后，Amazon Redshift 将通过启用加密来作为所有新创建的 Amazon Redshift 预置集群的默认设置，进一步增强数据和集群的安全性。这不适用于从快照还原的集群。

进行此更改后，当使用 AWS 管理控制台、AWS CLI 或 API 创建预置集群而不指定 KMS 密钥时，解密集群的功能将不再可用。集群将自动使用 AWS 拥有的密钥进行加密。

如果您使用自动脚本创建未加密的集群，或利用与未加密集群的数据共享，则此更新可能会对您产生影响。为确保无缝过渡，请更新用于创建未加密集群的脚本。此外，如果您定期创建新的未加密使用者集群并将其用于数据共享，请检查您的配置以确保生产者和使用者集群都经过加密，从而防止中断数据共享活动。有关更多信息，请参阅 Amazon Redshift 数据库加密。

## 强制执行 SSL 连接

2025 年 1 月 10 日之后，Amazon Redshift 将默认对连接到新创建的预置和还原集群的客户端强制执行 SSL 连接。此默认更改也将适用于无服务器工作组。

通过此更改，将为所有新创建或还原的集群引入一个名为 `default.redshift-2.0` 的新默认参数组，而 `require_ssl` 参数默认设置为 `true`。在没有指定的参数组的情况下创建的任何新集群都将自动利用 `default.redshift-2.0` 参数组。通过 Amazon Redshift 控制台创建集群时，将自动选择新的 `default.redshift-2.0` 参数组。此更改还将反映在 `CreateCluster` 和 `RestoreFromClusterSnapshot` API 操作以及相应的 SDK 和 AWS CLI 命令中。如果您使用现有或自定义参数组，Amazon Redshift 将继续使用在参数组中指定的 `require_ssl` 值。您可以根据需要，继续选择更改自定义参数组中的 `require_ssl` 值。

对于 Amazon Redshift Serverless 用户，`config-parameters` 中的默认值 `require_ssl` 将更改为 `true`。任何创建将 `require_ssl` 设置为 `false` 的新工作组的请求都将被拒绝。创建工作组后，您可以将 `require_ssl` 值更改为 `false`。有关更多信息，请参阅 配置连接的安全选项。

请注意，如果您的特定用例需要，您仍然可以修改集群或工作组设置以更改默认行为。

# 使用 AWS SDK 的 Amazon Redshift 代码示例

以下代码示例演示了如何将 Amazon Redshift 与 AWS 软件开发工具包（SDK）结合使用。

基本功能是向您展示如何在服务中执行基本操作的代码示例。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

场景是向您展示如何通过在一个服务中调用多个函数或与其他 AWS 服务 服务结合来完成特定任务的代码示例。

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

代码示例

- 使用 AWS SDK 的 Amazon Redshift 的基本功能示例
  - 开始使用 Amazon Redshift
  - 了解使用 AWS SDK 的 Amazon Redshift 的基本功能
  - 使用 AWS SDK 对 Amazon Redshift 执行的操作
    - 将 CreateCluster 与 AWS SDK 或 CLI 配合使用
    - 将 DeleteCluster 与 AWS SDK 或 CLI 配合使用
    - 将 DescribeClusters 与 AWS SDK 或 CLI 配合使用
    - 将 DescribeStatement 和 AWS SDK 搭配使用
    - 将 ExecuteStatement 和 AWS SDK 搭配使用
    - 将 GetStatementResult 和 AWS SDK 搭配使用
    - 将 ListDatabases 和 AWS SDK 搭配使用
    - 将 ModifyCluster 与 AWS SDK 或 CLI 配合使用
- 使用 AWS SDK 的 Amazon Redshift 场景
  - 创建 Amazon Redshift 项目追踪器
  - 通过 CLI 开始使用无服务器数据仓库
  - 预调配数据仓库集群入门

# 使用 AWS SDK 的 Amazon Redshift 的基本功能示例

以下代码示例展示了如何将 Amazon Redshift 的基本功能与 AWS SDK 结合使用。

示例

- 开始使用 Amazon Redshift
- 了解使用 AWS SDK 的 Amazon Redshift 的基本功能
- 使用 AWS SDK 对 Amazon Redshift 执行的操作
  - 将 CreateCluster 与 AWS SDK 或 CLI 配合使用
  - 将 DeleteCluster 与 AWS SDK 或 CLI 配合使用
  - 将 DescribeClusters 与 AWS SDK 或 CLI 配合使用
  - 将 DescribeStatement 和 AWS SDK 搭配使用
  - 将 ExecuteStatement 和 AWS SDK 搭配使用
  - 将 GetStatementResult 和 AWS SDK 搭配使用
  - 将 ListDatabases 和 AWS SDK 搭配使用
  - 将 ModifyCluster 与 AWS SDK 或 CLI 配合使用

## 开始使用 Amazon Redshift

以下代码示例展示了如何开始使用 Amazon Redshift。

.NET

适用于 .NET 的 SDK (v4)

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
/// <summary>
/// Main method to run the Hello Amazon Redshift example.
/// </summary>
/// <param name="args">Command line arguments (not used).</param>
```

```
   public static async Task Main(string[] args)
   {
       var redshiftClient = new AmazonRedshiftClient();

       Console.WriteLine("Hello, Amazon Redshift! Let's list available
clusters:");

       var clusters = new List<Cluster>();

       try
       {
           // Use pagination to retrieve all clusters.
           var clustersPaginator =
redshiftClient.Paginators.DescribeClusters(new DescribeClustersRequest());

           await foreach (var response in clustersPaginator.Responses)
           {
               if (response.Clusters != null)
                   clusters.AddRange(response.Clusters);
           }

           Console.WriteLine($"{clusters.Count} cluster(s) retrieved.");

           foreach (var cluster in clusters)
           {
               Console.WriteLine($"\t{cluster.ClusterIdentifier} (Status:
{cluster.ClusterStatus})");
           }
       }
       catch (AmazonRedshiftException ex)
       {
           Console.WriteLine($"Couldn't list clusters. Here's why:
{ex.Message}");
       }
       catch (Exception ex)
       {
           Console.WriteLine($"An error occurred: {ex.Message}");
       }
   }
```

- 有关 API 详细信息，请参阅《适用于 .NET 的 AWS SDK API Reference》中的 DescribeClusters。

Go

适用于 Go 的 SDK V2

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
package main

import (
 "context"
 "fmt"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/redshift"
)

// main uses the AWS SDK for Go V2 to create a Redshift client
// and list up to 10 clusters in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
 ctx := context.Background()
 sdkConfig, err := config.LoadDefaultConfig(ctx)
 if err != nil {
  fmt.Println("Couldn't load default configuration. Have you set up your AWS
 account?")
  fmt.Println(err)
  return
 }
 redshiftClient := redshift.NewFromConfig(sdkConfig)
 count := 20
 fmt.Printf("Let's list up to %v clusters for your account.\n", count)
 result, err := redshiftClient.DescribeClusters(ctx,
 &redshift.DescribeClustersInput{
  MaxRecords: aws.Int32(int32(count)),
 })
```

```
 if err != nil {
  fmt.Printf("Couldn't list clusters for your account. Here's why: %v\n", err)
  return
 }
 if len(result.Clusters) == 0 {
  fmt.Println("You don't have any clusters!")
  return
 }
 for _, cluster := range result.Clusters {
  fmt.Printf("\t%v : %v\n", *cluster.ClusterIdentifier, *cluster.ClusterStatus)
 }
}
```

- 有关 API 详细信息，请参阅《适用于 Go 的 AWS SDK API Reference》中的 DescribeClusters。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.redshift.RedshiftClient;
import
 software.amazon.awssdk.services.redshift.paginators.DescribeClustersIterable;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
```

```
 */
public class HelloRedshift {
    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        RedshiftClient redshiftClient = RedshiftClient.builder()
            .region(region)
            .build();

        listClustersPaginator(redshiftClient);
    }

    public static void listClustersPaginator(RedshiftClient redshiftClient) {
        DescribeClustersIterable clustersIterable =
 redshiftClient.describeClustersPaginator();
        clustersIterable.stream()
            .flatMap(r -> r.clusters().stream())
            .forEach(cluster -> System.out
                .println(" Cluster identifier: " + cluster.clusterIdentifier() +
 " status = " + cluster.clusterStatus()));
    }
}
```

- 有关 API 详细信息，请参阅《AWS SDK for Java 2.x API Reference》中的 DescribeClusters。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
import boto3


def hello_redshift(redshift_client):
    """
```

```
    Use the AWS SDK for Python (Boto3) to create an Amazon Redshift client and
 list
    the clusters in your account. This list might be empty if you haven't created
    any clusters.
    This example uses the default settings specified in your shared credentials
    and config files.

    :param redshift_client: A Boto3 Redshift Client object.
    """
    print("Hello, Redshift! Let's list your clusters:")
    paginator = redshift_client.get_paginator("describe_clusters")
    clusters = []
    for page in paginator.paginate():
        clusters.extend(page["Clusters"])

    print(f"{len(clusters)} cluster(s) were found.")

    for cluster in clusters:
        print(f"  {cluster['ClusterIdentifier']}")


if __name__ == "__main__":
    hello_redshift(boto3.client("redshift"))
```

- 有关 API 详细信息，请参阅《AWS SDK for Python（Boto3）API Reference》中的 DescribeClusters。

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

## 了解使用 AWS SDK 的 Amazon Redshift 的基本功能

以下代码示例演示了如何：

- 创建 Redshift 集群。
- 列出集群中的数据库。
- 创建名为 Movies 的文件。
- 填充 Movies 表。
- 按年份查询 Movies 表。

- 修改 Redshift 集群。
- 删除 Amazon Redshift 集群。

.NET

适用于 .NET 的 SDK (v4)

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

创建一个 Redshift 包装器类来管理操作。

```
/// <summary>
/// Wrapper class for Amazon Redshift operations.
/// </summary>
public class RedshiftWrapper
{
    private readonly IAmazonRedshift _redshiftClient;
    private readonly IAmazonRedshiftDataAPIService _redshiftDataClient;

    /// <summary>
    /// Constructor for RedshiftWrapper.
    /// </summary>
    /// <param name="redshiftClient">Amazon Redshift client.</param>
    /// <param name="redshiftDataClient">Amazon Redshift Data API client.</param>
    public RedshiftWrapper(IAmazonRedshift redshiftClient,
 IAmazonRedshiftDataAPIService redshiftDataClient)
    {
        _redshiftClient = redshiftClient;
        _redshiftDataClient = redshiftDataClient;
    }

    /// <summary>
    /// Create a new Amazon Redshift cluster.
    /// </summary>
    /// <param name="clusterIdentifier">The identifier for the cluster.</param>
    /// <param name="databaseName">The name of the database.</param>
    /// <param name="masterUsername">The master username.</param>
```

```
   /// <param name="masterUserPassword">The master user password.</param>
   /// <param name="nodeType">The node type for the cluster.</param>
   /// <returns>The cluster that was created.</returns>
   public async Task<Cluster> CreateClusterAsync(string clusterIdentifier,
string databaseName,
       string masterUsername, string masterUserPassword, string nodeType =
"ra3.large")
   {
       try
       {
           var request = new CreateClusterRequest
           {
               ClusterIdentifier = clusterIdentifier,
               DBName = databaseName,
               MasterUsername = masterUsername,
               MasterUserPassword = masterUserPassword,
               NodeType = nodeType,
               NumberOfNodes = 1,
               ClusterType = "single-node"
           };

           var response = await _redshiftClient.CreateClusterAsync(request);
           Console.WriteLine($"Created cluster {clusterIdentifier}");
           return response.Cluster;
       }
       catch (ClusterAlreadyExistsException ex)
       {
           Console.WriteLine($"Cluster already exists: {ex.Message}");
           throw;
       }
       catch (Exception ex)
       {
           Console.WriteLine($"Couldn't create cluster. Here's why:
{ex.Message}");
           throw;
       }
   }

   /// <summary>
   /// Describe Amazon Redshift clusters.
   /// </summary>
   /// <param name="clusterIdentifier">Optional cluster identifier to describe a
specific cluster.</param>
   /// <returns>A list of clusters.</returns>
```

```
   public async Task<List<Cluster>> DescribeClustersAsync(string?
clusterIdentifier = null)
   {
       try
       {
           var clusters = new List<Cluster>();
           var request = new DescribeClustersRequest();
           if (!string.IsNullOrEmpty(clusterIdentifier))
           {
               request.ClusterIdentifier = clusterIdentifier;
           }

           var clustersPaginator =
_redshiftClient.Paginators.DescribeClusters(request);
           await foreach (var response in clustersPaginator.Responses)
           {
               if (response.Clusters != null)
                   clusters.AddRange(response.Clusters);
           }

           Console.WriteLine($"{clusters.Count} cluster(s) retrieved.");
           foreach (var cluster in clusters)
           {
               Console.WriteLine($"\t{cluster.ClusterIdentifier} (Status:
{cluster.ClusterStatus})");
           }

           return clusters;
       }
       catch (ClusterNotFoundException ex)
       {
           Console.WriteLine($"Cluster {clusterIdentifier} not found:
{ex.Message}");
           throw;
       }
       catch (Exception ex)
       {
           Console.WriteLine($"Couldn't describe clusters. Here's why:
{ex.Message}");
           throw;
       }
   }

   /// <summary>
```

```
   /// Modify an Amazon Redshift cluster.
   /// </summary>
   /// <param name="clusterIdentifier">The identifier for the cluster.</param>
   /// <param name="preferredMaintenanceWindow">The preferred maintenance
window.</param>
   /// <returns>True if successful.</returns>
   public async Task<bool> ModifyClusterAsync(string clusterIdentifier, string
preferredMaintenanceWindow)
   {
       try
       {
           var request = new ModifyClusterRequest
           {
               ClusterIdentifier = clusterIdentifier,
               PreferredMaintenanceWindow = preferredMaintenanceWindow
           };

           var response = await _redshiftClient.ModifyClusterAsync(request);
           Console.WriteLine($"The modified cluster was successfully modified
and has {response.Cluster.PreferredMaintenanceWindow} as the maintenance
window");
           return true;
       }
       catch (ClusterNotFoundException ex)
       {
           Console.WriteLine($"Cluster {clusterIdentifier} not found:
{ex.Message}");
           return false;
       }
       catch (Exception ex)
       {
           Console.WriteLine($"Couldn't modify cluster. Here's why:
{ex.Message}");
           return false;
       }
   }

   /// <summary>
   /// Delete an Amazon Redshift cluster without a final snapshot.
   /// </summary>
   /// <param name="clusterIdentifier">The identifier for the cluster.</param>
   /// <returns>True if successful.</returns>
   public async Task<bool> DeleteClusterWithoutSnapshotAsync(string
clusterIdentifier)
```

```
    {
        try
        {
            var request = new DeleteClusterRequest
            {
                ClusterIdentifier = clusterIdentifier,
                SkipFinalClusterSnapshot = true
            };

            var response = await _redshiftClient.DeleteClusterAsync(request);
            Console.WriteLine($"The {clusterIdentifier} was deleted");
            return true;
        }
        catch (ClusterNotFoundException ex)
        {
            Console.WriteLine($"Cluster not found: {ex.Message}");
            return false;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't delete cluster. Here's why:
{ex.Message}");
            return false;
        }
    }

    /// <summary>
    /// List databases in a Redshift cluster.
    /// </summary>
    /// <param name="clusterIdentifier">The cluster identifier.</param>
    /// <param name="dbUser">The database user.</param>
    /// <param name="dbUser">The database name for authentication.</param>
    /// <returns>A list of database names.</returns>
    public async Task<List<string>> ListDatabasesAsync(string clusterIdentifier,
string dbUser, string databaseName)
    {
        try
        {
            var request = new ListDatabasesRequest
            {
                ClusterIdentifier = clusterIdentifier,
                DbUser = dbUser,
                Database = databaseName
            };
```

```
            var response = await _redshiftDataClient.ListDatabasesAsync(request);
            var databases = new List<string>();

            foreach (var database in response.Databases)
            {
                Console.WriteLine($"The database name is : {database}");
                databases.Add(database);
            }

            return databases;
        }
        catch (Amazon.RedshiftDataAPIService.Model.ValidationException ex)
        {
            Console.WriteLine($"Validation error: {ex.Message}");
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't list databases. Here's why:
{ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Create a table in the Redshift database.
    /// </summary>
    /// <param name="clusterIdentifier">The cluster identifier.</param>
    /// <param name="database">The database name.</param>
    /// <param name="dbUser">The database user.</param>
    /// <returns>The statement ID.</returns>
    public async Task<string> CreateTableAsync(string clusterIdentifier, string
database, string dbUser)
    {
        try
        {
            var sqlStatement = @"
                CREATE TABLE Movies (
                    id INTEGER PRIMARY KEY,
                    title VARCHAR(250) NOT NULL,
                    year INTEGER NOT NULL
                )";
```

```
            var request = new ExecuteStatementRequest
            {
                ClusterIdentifier = clusterIdentifier,
                Database = database,
                DbUser = dbUser,
                Sql = sqlStatement
            };

            var response = await
_redshiftDataClient.ExecuteStatementAsync(request);
            await WaitForStatementToCompleteAsync(response.Id);
            Console.WriteLine("Table created: Movies");
            return response.Id;
        }
        catch (Amazon.RedshiftDataAPIService.Model.ValidationException ex)
        {
            Console.WriteLine($"Validation error: {ex.Message}");
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't create table. Here's why:
{ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Insert a record into the Movies table using parameterized query.
    /// </summary>
    /// <param name="clusterIdentifier">The cluster identifier.</param>
    /// <param name="database">The database name.</param>
    /// <param name="dbUser">The database user.</param>
    /// <param name="id">The movie ID.</param>
    /// <param name="title">The movie title.</param>
    /// <param name="year">The movie year.</param>
    /// <returns>The statement ID.</returns>
    public async Task<string> InsertMovieAsync(string clusterIdentifier, string
database, string dbUser,
        int id, string title, int year)
    {
        try
        {
```

```
            var sqlStatement = "INSERT INTO Movies (id, title, year) VALUES
(:id, :title, :year)";

            var request = new ExecuteStatementRequest
            {
                ClusterIdentifier = clusterIdentifier,
                Database = database,
                DbUser = dbUser,
                Sql = sqlStatement,
                Parameters = new List<SqlParameter>
                {
                    new SqlParameter { Name = "id", Value = id.ToString() },
                    new SqlParameter { Name = "title", Value = title },
                    new SqlParameter { Name = "year", Value = year.ToString() }
                }
            };

            var response = await
_redshiftDataClient.ExecuteStatementAsync(request);
            await WaitForStatementToCompleteAsync(response.Id);
            Console.WriteLine($"Inserted: {title} ({year})");
            return response.Id;
        }
        catch (Amazon.RedshiftDataAPIService.Model.ValidationException ex)
        {
            Console.WriteLine($"Validation error: {ex.Message}");
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't insert movie. Here's why:
{ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Query movies by year using parameterized query.
    /// </summary>
    /// <param name="clusterIdentifier">The cluster identifier.</param>
    /// <param name="database">The database name.</param>
    /// <param name="dbUser">The database user.</param>
    /// <param name="year">The year to query.</param>
    /// <returns>A list of movie titles.</returns>
```

```
   public async Task<List<string>> QueryMoviesByYearAsync(string
clusterIdentifier, string database,
       string dbUser, int year)
   {
       try
       {
           var sqlStatement = "SELECT title FROM Movies WHERE year = :year";

           var request = new ExecuteStatementRequest
           {
               ClusterIdentifier = clusterIdentifier,
               Database = database,
               DbUser = dbUser,
               Sql = sqlStatement,
               Parameters = new List<SqlParameter>
               {
                   new SqlParameter { Name = "year", Value = year.ToString() }
               }
           };

           var response = await
_redshiftDataClient.ExecuteStatementAsync(request);
           Console.WriteLine($"The identifier of the statement is
{response.Id}");

           await WaitForStatementToCompleteAsync(response.Id);

           var results = await GetStatementResultAsync(response.Id);
           var movieTitles = new List<string>();

           foreach (var row in results)
           {
               if (row.Count > 0)
               {
                   var title = row[0].StringValue;
                   Console.WriteLine($"The Movie title field is {title}");
                   movieTitles.Add(title);
               }
           }

           return movieTitles;
       }
       catch (Amazon.RedshiftDataAPIService.Model.ValidationException ex)
       {
```

```
            Console.WriteLine($"Validation error: {ex.Message}");
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't query movies. Here's why:
{ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Describe a statement execution.
    /// </summary>
    /// <param name="statementId">The statement ID.</param>
    /// <returns>The statement description.</returns>
    public async Task<DescribeStatementResponse> DescribeStatementAsync(string
statementId)
    {
        try
        {
            var request = new DescribeStatementRequest
            {
                Id = statementId
            };

            var response = await
_redshiftDataClient.DescribeStatementAsync(request);
            return response;
        }
        catch (Amazon.RedshiftDataAPIService.Model.ResourceNotFoundException ex)
        {
            Console.WriteLine($"Statement not found: {ex.Message}");
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't describe statement. Here's why:
{ex.Message}");
            throw;
        }
    }

    /// <summary>
```

```
   /// Get the results of a statement execution.
   /// </summary>
   /// <param name="statementId">The statement ID.</param>
   /// <returns>A list of result rows.</returns>
   public async Task<List<List<Field>>> GetStatementResultAsync(string
statementId)
   {
       try
       {
           var request = new GetStatementResultRequest
           {
               Id = statementId
           };

           var response = await
_redshiftDataClient.GetStatementResultAsync(request);
           return response.Records;
       }
       catch (Amazon.RedshiftDataAPIService.Model.ResourceNotFoundException ex)
       {
           Console.WriteLine($"Statement not found: {ex.Message}");
           throw;
       }
       catch (Exception ex)
       {
           Console.WriteLine($"Couldn't get statement result. Here's why:
{ex.Message}");
           throw;
       }
   }

   /// <summary>
   /// Wait for a statement to complete execution.
   /// </summary>
   /// <param name="statementId">The statement ID.</param>
   /// <returns>A task representing the asynchronous operation.</returns>
   private async Task WaitForStatementToCompleteAsync(string statementId)
   {
       var status = StatusString.SUBMITTED;
       DescribeStatementResponse? response = null;

       while (status == StatusString.SUBMITTED || status == StatusString.PICKED
|| status == StatusString.STARTED)
       {
```

```
            await Task.Delay(1000); // Wait 1 second
            response = await DescribeStatementAsync(statementId);
            status = response.Status;
            Console.WriteLine($"...{status}");
        }

        if (status == StatusString.FINISHED)
        {
            Console.WriteLine("The statement is finished!");
        }
        else
        {
            var errorMessage = response?.Error ?? "Unknown error";
            Console.WriteLine($"The statement failed with status: {status}");
            Console.WriteLine($"Error message: {errorMessage}");
        }
    }

    /// <summary>
    /// Wait for a cluster to become available.
    /// </summary>
    /// <param name="clusterIdentifier">The cluster identifier.</param>
    /// <returns>A task representing the asynchronous operation.</returns>
    public async Task WaitForClusterAvailableAsync(string clusterIdentifier)
    {
        Console.WriteLine($"Wait until {clusterIdentifier} is available. This may
take a few minutes.");

        var startTime = DateTime.Now;
        var clusters = await DescribeClustersAsync(clusterIdentifier);

        while (clusters[0].ClusterStatus != "available")
        {
            var elapsed = DateTime.Now - startTime;
            Console.WriteLine($"Elapsed Time: {elapsed:mm\\:ss} - Waiting for
cluster...");

            await Task.Delay(5000); // Wait 5 seconds
            clusters = await DescribeClustersAsync(clusterIdentifier);
        }

        var totalElapsed = DateTime.Now - startTime;
        Console.WriteLine($"Cluster is available! Total Elapsed Time:
{totalElapsed:mm\\:ss}");
```

```
    }
}
```

运行一个交互式场景，演示 Redshift 的基础功能。

```
/// <summary>
/// Amazon Redshift Getting Started Scenario.
/// </summary>
public class RedshiftBasics
{
    public static bool IsInteractive = true;
    public static RedshiftWrapper? Wrapper = null;
    public static ILogger logger = null!;
    private static readonly string _moviesFilePath = "../../../../../../
resources/sample_files/movies.json";

    /// <summary>
    /// Main method for the Amazon Redshift Getting Started scenario.
    /// </summary>
    /// <param name="args">Command line arguments.</param>
    public static async Task Main(string[] args)
    {
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonRedshift>()
                    .AddAWSService<IAmazonRedshiftDataAPIService>()
                    .AddTransient<RedshiftWrapper>()
            )
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<RedshiftBasics>();

        Wrapper = host.Services.GetRequiredService<RedshiftWrapper>();

        await RunScenarioAsync();
    }

    /// <summary>
    /// Run the complete Amazon Redshift scenario.
    /// </summary>
    public static async Task RunScenarioAsync()
```

```
    {
        // Set all variables to default values
        string userName = "awsuser";
        string userPassword = "AwsUser1000";
        string clusterIdentifier = "redshift-cluster-movies";
        var databaseName = "dev";
        int recordCount = 50;
        int year = 2013;
        try
        {
            Console.WriteLine(

"================================================================================");
            Console.WriteLine("Welcome to the Amazon Redshift SDK Getting Started
scenario.");
            Console.WriteLine(
                "This .NET program demonstrates how to interact with Amazon
Redshift by using the AWS SDK for .NET.");
            Console.WriteLine("Let's get started...");
            Console.WriteLine(

"================================================================================");

            // Step 1: Get user credentials (if interactive)
            if (IsInteractive)
            {
                Console.WriteLine("Please enter a user name for the cluster
(default is awsuser):");
                var userInput = Console.ReadLine();
                if (!string.IsNullOrEmpty(userInput))
                    userName = userInput;


Console.WriteLine("================================================================================
                Console.WriteLine("Please enter a user password for the cluster
(default is AwsUser1000):");
                var passwordInput = Console.ReadLine();
                if (!string.IsNullOrEmpty(passwordInput))
                    userPassword = passwordInput;


Console.WriteLine("================================================================================

                // Step 2: Get cluster identifier
```

```
                Console.WriteLine("Enter a cluster id value (default is redshift-
cluster-movies):");
                var clusterInput = Console.ReadLine();
                if (!string.IsNullOrEmpty(clusterInput))
                    clusterIdentifier = clusterInput;
            }
            else
            {
                Console.WriteLine($"Using default values: userName={userName},
 clusterIdentifier={clusterIdentifier}");
            }

            // Step 3: Create Redshift cluster
            await Wrapper!.CreateClusterAsync(clusterIdentifier, databaseName,
 userName, userPassword);

 Console.WriteLine("=============================================================================

            // Step 4: Wait for cluster to become available

 Console.WriteLine("=============================================================================
            await Wrapper.WaitForClusterAvailableAsync(clusterIdentifier);

 Console.WriteLine("=============================================================================

            // Step 5: List databases

 Console.WriteLine("=============================================================================
            Console.WriteLine($" When you created {clusterIdentifier}, the dev
 database is created by default and used in this scenario.");
            Console.WriteLine(" To create a custom database, you need to have a
 CREATEDB privilege.");
            Console.WriteLine(" For more information, see the documentation here:
 https://docs.aws.amazon.com/redshift/latest/dg/r_CREATE_DATABASE.html.");
            if (IsInteractive)
            {
                Console.WriteLine("Press Enter to continue...");
                Console.ReadLine();
            }

 Console.WriteLine("=============================================================================


 Console.WriteLine("=============================================================================
```

```
            Console.WriteLine($"List databases in {clusterIdentifier}");
            if (IsInteractive)
            {
                Console.WriteLine("Press Enter to continue...");
                Console.ReadLine();
            }
            await Wrapper.ListDatabasesAsync(clusterIdentifier, userName,
databaseName);

Console.WriteLine("=========================================================================

            // Step 6: Create Movies table

Console.WriteLine("=========================================================================
            Console.WriteLine("Now you will create a table named Movies.");
            if (IsInteractive)
            {
                Console.WriteLine("Press Enter to continue...");
                Console.ReadLine();
            }
            await Wrapper.CreateTableAsync(clusterIdentifier, databaseName,
userName);

Console.WriteLine("=========================================================================

            // Step 7: Populate the Movies table

Console.WriteLine("=========================================================================
            Console.WriteLine("Populate the Movies table using the Movies.json
file.");

            if (IsInteractive)
            {
                Console.WriteLine("Specify the number of records you would like
to add to the Movies Table.");
                Console.WriteLine("Please enter a value between 50 and 200.");
                Console.Write("Enter a value: ");

                var recordCountInput = Console.ReadLine();
                if (int.TryParse(recordCountInput, out var inputCount) &&
inputCount is >= 50 and <= 200)
                {
                    recordCount = inputCount;
                }
```

```
                else
                {
                    Console.WriteLine($"Invalid input. Using default value of
{recordCount}.");
                }
            }
            else
            {
                Console.WriteLine($"Using default record count: {recordCount}");
            }

            await PopulateMoviesTableAsync(clusterIdentifier, databaseName,
userName, recordCount);
            Console.WriteLine($"{recordCount} records were added to the Movies
table.");

Console.WriteLine("=============================================================================

            // Step 8 & 9: Query movies by year

Console.WriteLine("=============================================================================
            Console.WriteLine("Query the Movies table by year. Enter a value
between 2012-2014.");

            if (IsInteractive)
            {
                Console.Write("Enter a year: ");
                var yearInput = Console.ReadLine();
                if (int.TryParse(yearInput, out var inputYear) && inputYear is >=
2012 and <= 2014)
                {
                    year = inputYear;
                }
                else
                {
                    Console.WriteLine($"Invalid input. Using default value of
{year}.");
                }
            }
            else
            {
                Console.WriteLine($"Using default year: {year}");
            }
```

```
            await Wrapper.QueryMoviesByYearAsync(clusterIdentifier, databaseName,
 userName, year);

 Console.WriteLine("=======================================================================

            // Step 10: Modify the cluster

 Console.WriteLine("=======================================================================
            Console.WriteLine("Now you will modify the Redshift cluster.");
            if (IsInteractive)
            {
                Console.WriteLine("Press Enter to continue...");
                Console.ReadLine();
            }
            await Wrapper.ModifyClusterAsync(clusterIdentifier, "wed:07:30-
wed:08:00");

 Console.WriteLine("=======================================================================

            // Step 11 & 12: Delete cluster confirmation

 Console.WriteLine("=======================================================================
            if (IsInteractive)
            {
                Console.WriteLine("Would you like to delete the Amazon Redshift
 cluster? (y/n)");
                var deleteResponse = Console.ReadLine();
                if (deleteResponse?.ToLower() == "y")
                {
                    await
 Wrapper.DeleteClusterWithoutSnapshotAsync(clusterIdentifier);
                }
            }
            else
            {
                Console.WriteLine("Deleting the Amazon Redshift cluster...");
                await
 Wrapper.DeleteClusterWithoutSnapshotAsync(clusterIdentifier);
            }

 Console.WriteLine("=======================================================================


 Console.WriteLine("=======================================================================
```

```
            Console.WriteLine("This concludes the Amazon Redshift SDK Getting
Started scenario.");

Console.WriteLine("========================================================================
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred during the scenario:
{ex.Message}");
            Console.WriteLine("Deleting the Amazon Redshift cluster...");
            await Wrapper!.DeleteClusterWithoutSnapshotAsync(clusterIdentifier);
            throw;
        }
    }

    /// <summary>
    /// Populate the Movies table with data from the JSON file.
    /// </summary>
    /// <param name="clusterIdentifier">The cluster identifier.</param>
    /// <param name="database">The database name.</param>
    /// <param name="dbUser">The database user.</param>
    /// <param name="recordCount">Number of records to insert.</param>
    private static async Task PopulateMoviesTableAsync(string clusterIdentifier,
string database, string dbUser, int recordCount)
    {
        if (!File.Exists(_moviesFilePath))
        {
            throw new FileNotFoundException($"Required movies data file not found
at: {_moviesFilePath}");
        }

        var jsonContent = await File.ReadAllTextAsync(_moviesFilePath);
        var options = new JsonSerializerOptions
        {
            PropertyNameCaseInsensitive = true
        };
        var movies = JsonSerializer.Deserialize<List<Movie>>(jsonContent,
options);

        if (movies == null || movies.Count == 0)
        {
            throw new InvalidOperationException("Failed to parse movies JSON file
or file is empty.");
        }
```

```
        var insertCount = Math.Min(recordCount, movies.Count);

        for (int i = 0; i < insertCount; i++)
        {
            var movie = movies[i];
            await Wrapper!.InsertMovieAsync(clusterIdentifier, database, dbUser,
 i, movie.Title, movie.Year);
        }
    }

    /// <summary>
    /// Movie data model.
    /// </summary>
    private class Movie
    {
        public string Title { get; set; } = string.Empty;
        public int Year { get; set; }
    }
}
```

- 有关 API 详细信息，请参阅《适用于 .NET 的 AWS SDK API Reference》中的以下主题。
  - CreateCluster：
  - DescribeClusters
  - DescribeStatement
  - ExecuteStatement
  - GetStatementResult
  - ListDatabasesPaginator
  - ModifyCluster

Go

适用于 Go 的 SDK V2

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库 中查找完整示例，了解如何进行设置和运行。

```
package scenarios

import (
 "context"
 "encoding/json"
 "errors"
 "fmt"
 "log"
 "math/rand"
 "strings"
 "time"

 "github.com/aws/aws-sdk-go-v2/aws"
 redshift_types "github.com/aws/aws-sdk-go-v2/service/redshift/types"
 redshiftdata_types "github.com/aws/aws-sdk-go-v2/service/redshiftdata/types"
 "github.com/aws/aws-sdk-go-v2/service/secretsmanager"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/redshift/actions"

 "github.com/aws/aws-sdk-go-v2/service/redshift"
 "github.com/aws/aws-sdk-go-v2/service/redshiftdata"
)

// IScenarioHelper abstracts input and wait functions from a scenario so that
 they
// can be mocked for unit testing.
type IScenarioHelper interface {
 GetName() string
}

const rMax = 100000
```

```
type ScenarioHelper struct {
 Prefix string
 Random *rand.Rand
}

// GetName returns a unique name formed of a prefix and a random number.
func (helper ScenarioHelper) GetName() string {
 return fmt.Sprintf("%v%v", helper.Prefix, helper.Random.Intn(rMax))
}

// RedshiftBasicsScenario separates the steps of this scenario into individual
 functions so that
// they are simpler to read and understand.
type RedshiftBasicsScenario struct {
 sdkConfig         aws.Config
 helper            IScenarioHelper
 questioner        demotools.IQuestioner
 pauser            demotools.IPausable
 filesystem        demotools.IFileSystem
 redshiftActor     *actions.RedshiftActions
 redshiftDataActor *actions.RedshiftDataActions
 secretsmanager    *SecretsManager
}

// SecretsManager is used to retrieve username and password information from a
 secure service.
type SecretsManager struct {
 SecretsManagerClient *secretsmanager.Client
}

// RedshiftBasics constructs a new Redshift Basics runner.
func RedshiftBasics(sdkConfig aws.Config, questioner demotools.IQuestioner,
 pauser demotools.IPausable, filesystem demotools.IFileSystem, helper
 IScenarioHelper) RedshiftBasicsScenario {
 scenario := RedshiftBasicsScenario{
  sdkConfig:         sdkConfig,
  helper:            helper,
  questioner:        questioner,
  pauser:            pauser,
  filesystem:        filesystem,
  secretsmanager:    &SecretsManager{SecretsManagerClient:
 secretsmanager.NewFromConfig(sdkConfig)},
```

```
  redshiftActor:     &actions.RedshiftActions{RedshiftClient:
 redshift.NewFromConfig(sdkConfig)},
  redshiftDataActor: &actions.RedshiftDataActions{RedshiftDataClient:
 redshiftdata.NewFromConfig(sdkConfig)},
 }
 return scenario
}


// Movie makes it easier to use Movie objects given in json format.
type Movie struct {
 ID    int    `json:"id"`
 Title string `json:"title"`
 Year  int    `json:"year"`
}


// User makes it easier to get the User data back from SecretsManager and use it
 later.
type User struct {
 Username string `json:"userName"`
 Password string `json:"userPassword"`
}

// Run runs the RedshiftBasics interactive example that shows you how to use
 Amazon
// Redshift and how to interact with its common endpoints.
//
// 0. Retrieve username and password information to access Redshift.
// 1. Create a cluster.
// 2. Wait for the cluster to become available.
// 3. List the available databases in the region.
// 4. Create a table named "Movies" in the "dev" database.
// 5. Populate the movies table from the "movies.json" file.
// 6. Query the movies table by year.
// 7. Modify the cluster's maintenance window.
// 8. Optionally clean up all resources created during this demo.
//
// This example creates an Amazon Redshift service client from the specified
 sdkConfig so that
// you can replace it with a mocked or stubbed config for unit testing.
//
// It uses a questioner from the `demotools` package to get input during the
 example.
```

```
// This package can be found in the ..\..\demotools folder of this repo.
func (runner *RedshiftBasicsScenario) Run(ctx context.Context) {

 user := User{}
 secretId := "s3express/basics/secrets"
 clusterId := "demo-cluster-1"
 maintenanceWindow := "wed:07:30-wed:08:00"
 databaseName := "dev"
 tableName := "Movies"
 fileName := "Movies.json"
 nodeType := "ra3.xlplus"
 clusterType := "single-node"

 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
   _, isMock := runner.questioner.(*demotools.MockQuestioner)
   if isMock || runner.questioner.AskBool("Do you want to see the full error
 message (y/n)?", "y") {
    log.Println(r)
   }
   runner.cleanUpResources(ctx, clusterId, databaseName, tableName,
 user.Username, runner.questioner)
  }
 }()

 // Retrieve the userName and userPassword from SecretsManager
 output, err := runner.secretsmanager.SecretsManagerClient.GetSecretValue(ctx,
 &secretsmanager.GetSecretValueInput{
  SecretId: aws.String(secretId),
 })
 if err != nil {
  log.Printf("There was a problem getting the secret value: %s", err)
  log.Printf("Please make sure to create a secret named 's3express/basics/
secrets' with keys of 'userName' and 'userPassword'.")
  panic(err)
 }

 err = json.Unmarshal([]byte(*output.SecretString), &user)
 if err != nil {
  log.Printf("There was a problem parsing the secret value from JSON: %s", err)
  panic(err)
 }
```

```
// Create the Redshift cluster
_, err = runner.redshiftActor.CreateCluster(ctx, clusterId, user.Username,
user.Password, nodeType, clusterType, true)
if err != nil {
 var clusterAlreadyExistsFault *redshift_types.ClusterAlreadyExistsFault
 if errors.As(err, &clusterAlreadyExistsFault) {
  log.Println("Cluster already exists. Continuing.")
 } else {
  log.Println("Error creating cluster.")
  panic(err)
 }
}

// Wait for the cluster to become available
waiter :=
redshift.NewClusterAvailableWaiter(runner.redshiftActor.RedshiftClient)
err = waiter.Wait(ctx, &redshift.DescribeClustersInput{
 ClusterIdentifier: aws.String(clusterId),
}, 5*time.Minute)
if err != nil {
 log.Println("An error occurred waiting for the cluster.")
 panic(err)
}

// Get some info about the cluster
describeOutput, err := runner.redshiftActor.DescribeClusters(ctx, clusterId)
if err != nil {
 log.Println("Something went wrong trying to get information about the
cluster.")
 panic(err)
}
log.Println("Here's some information about the cluster.")
log.Printf("The cluster's status is %s",
*describeOutput.Clusters[0].ClusterStatus)
log.Printf("The cluster was created at %s",
*describeOutput.Clusters[0].ClusterCreateTime)

// List databases
log.Println("List databases in", clusterId)
runner.questioner.Ask("Press Enter to continue...")
err = runner.redshiftDataActor.ListDatabases(ctx, clusterId, databaseName,
user.Username)
if err != nil {
 log.Printf("Failed to list databases: %v\n", err)
```

```
 panic(err)
}

// Create the "Movies" table
log.Println("Now you will create a table named " + tableName + ".")
runner.questioner.Ask("Press Enter to continue...")
err = nil
result, err := runner.redshiftDataActor.CreateTable(ctx, clusterId,
databaseName, tableName, user.Username, runner.pauser, []string{"title
VARCHAR(256)", "year INT"})
if err != nil {
 log.Printf("Failed to create table: %v\n", err)
 panic(err)
}

describeInput := redshiftdata.DescribeStatementInput{
 Id: result.Id,
}
query := actions.RedshiftQuery{
 Context: ctx,
 Input:   describeInput,
 Result:  result,
}
err = runner.redshiftDataActor.WaitForQueryStatus(query, runner.pauser, true)
if err != nil {
 log.Printf("Failed to execute query: %v\n", err)
 panic(err)
}
log.Printf("Successfully executed query\n")

// Populate the "Movies" table
runner.PopulateMoviesTable(ctx, clusterId, databaseName, tableName,
user.Username, fileName)

// Query the "Movies" table by year
log.Println("Query the Movies table by year.")
year := runner.questioner.AskInt(
 fmt.Sprintf("Enter a value between %v and %v:", 2012, 2014),
 demotools.InIntRange{Lower: 2012, Upper: 2014})
runner.QueryMoviesByYear(ctx, clusterId, databaseName, tableName, user.Username,
year)

// Modify the cluster's maintenance window
runner.redshiftActor.ModifyCluster(ctx, clusterId, maintenanceWindow)
```

```
 // Delete the Redshift cluster if confirmed
 runner.cleanUpResources(ctx, clusterId, databaseName, tableName, user.Username,
 runner.questioner)

 log.Println("Thanks for watching!")
}

// cleanUpResources asks the user if they would like to delete each resource
 created during the scenario, from most
// impactful to least impactful. If any choice to delete is made, further
 deletion attempts are skipped.
func (runner *RedshiftBasicsScenario) cleanUpResources(ctx context.Context,
 clusterId string, databaseName string, tableName string, userName string,
 questioner demotools.IQuestioner) {
 deleted := false
 var err error = nil
 if questioner.AskBool("Do you want to delete the entire cluster? This will clean
 up all resources. (y/n)", "y") {
  deleted, err = runner.redshiftActor.DeleteCluster(ctx, clusterId)
  if err != nil {
   log.Printf("Error deleting cluster: %v", err)
  }
 }
 if !deleted && questioner.AskBool("Do you want to delete the dev table? This
 will clean up all inserted records but keep your cluster intact. (y/n)", "y") {
  deleted, err = runner.redshiftDataActor.DeleteTable(ctx, clusterId,
 databaseName, tableName, userName)
  if err != nil {
   log.Printf("Error deleting movies table: %v", err)
  }
 }
 if !deleted && questioner.AskBool("Do you want to delete all rows in the Movies
 table? This will clean up all inserted records but keep your cluster and table
 intact. (y/n)", "y") {
  deleted, err = runner.redshiftDataActor.DeleteDataRows(ctx, clusterId,
 databaseName, tableName, userName, runner.pauser)
  if err != nil {
   log.Printf("Error deleting data rows: %v", err)
  }
 }
 if !deleted {
  log.Print("Please manually delete any unwanted resources.")
 }
```

```
}


// loadMoviesFromJSON takes the <fileName> file and populates a slice of Movie
 objects.
func (runner *RedshiftBasicsScenario) loadMoviesFromJSON(fileName string,
 filesystem demotools.IFileSystem) ([]Movie, error) {
 file, err := filesystem.OpenFile("../../resources/sample_files/" + fileName)
 if err != nil {
  return nil, err
 }
 defer filesystem.CloseFile(file)

 var movies []Movie
 err = json.NewDecoder(file).Decode(&movies)
 if err != nil {
  return nil, err
 }

 return movies, nil
}



// PopulateMoviesTable reads data from the <fileName> file and inserts records
 into the "Movies" table.
func (runner *RedshiftBasicsScenario) PopulateMoviesTable(ctx context.Context,
 clusterId string, databaseName string, tableName string, userName string,
 fileName string) {
 log.Println("Populate the " + tableName + " table using the " + fileName + "
 file.")
 numRecords := runner.questioner.AskInt(
  fmt.Sprintf("Enter a value between %v and %v:", 10, 100),
  demotools.InIntRange{Lower: 10, Upper: 100})

 movies, err := runner.loadMoviesFromJSON(fileName, runner.filesystem)
 if err != nil {
  log.Printf("Failed to load movies from JSON: %v\n", err)
  panic(err)
 }

 var sqlStatements []string

 for i, movie := range movies {
```

```
  if i >= numRecords {
   break
  }

  sqlStatement := fmt.Sprintf(`INSERT INTO %s (title, year) VALUES ('%s', %d);`,
   tableName,
   strings.Replace(movie.Title, "'", "''", -1), // Double any single quotes to
 escape them
   movie.Year)

  sqlStatements = append(sqlStatements, sqlStatement)
 }

 input := &redshiftdata.BatchExecuteStatementInput{
  ClusterIdentifier: aws.String(clusterId),
  Database:          aws.String(databaseName),
  DbUser:            aws.String(userName),
  Sqls:              sqlStatements,
 }

 result, err := runner.redshiftDataActor.ExecuteBatchStatement(ctx, *input)
 if err != nil {
  log.Printf("Failed to execute batch statement: %v\n", err)
  panic(err)
 }

 describeInput := redshiftdata.DescribeStatementInput{
  Id: result.Id,
 }

 query := actions.RedshiftQuery{
  Context: ctx,
  Result:  result,
  Input:   describeInput,
 }
 err = runner.redshiftDataActor.WaitForQueryStatus(query, runner.pauser, true)
 if err != nil {
  log.Printf("Failed to execute batch insert query: %v\n", err)
  return
 }
 log.Printf("Successfully executed batch statement\n")

 log.Printf("%d records were added to the Movies table.\n", numRecords)
}
```

```
// QueryMoviesByYear retrieves only movies from the "Movies" table which match
 the given year.
func (runner *RedshiftBasicsScenario) QueryMoviesByYear(ctx context.Context,
 clusterId string, databaseName string, tableName string, userName string, year
 int) {

 sqlStatement := fmt.Sprintf(`SELECT title FROM %s WHERE year = %d;`, tableName,
 year)

 input := &redshiftdata.ExecuteStatementInput{
  ClusterIdentifier: aws.String(clusterId),
  Database:          aws.String(databaseName),
  DbUser:            aws.String(userName),
  Sql:               aws.String(sqlStatement),
 }

 result, err := runner.redshiftDataActor.ExecuteStatement(ctx, *input)
 if err != nil {
  log.Printf("Failed to query movies: %v\n", err)
  panic(err)
 }

 log.Println("The identifier of the statement is ", *result.Id)

 describeInput := redshiftdata.DescribeStatementInput{
  Id: result.Id,
 }

 query := actions.RedshiftQuery{
  Context: ctx,
  Input:   describeInput,
  Result:  result,
 }
 err = runner.redshiftDataActor.WaitForQueryStatus(query, runner.pauser, true)
 if err != nil {
  log.Printf("Failed to execute query: %v\n", err)
  panic(err)
 }
 log.Printf("Successfully executed query\n")
```

```
 getResultOutput, err := runner.redshiftDataActor.GetStatementResult(ctx,
 *result.Id)
 if err != nil {
  log.Printf("Failed to query movies: %v\n", err)
  panic(err)
 }
 for _, row := range getResultOutput.Records {
  for _, col := range row {
   title, ok := col.(*redshiftdata_types.FieldMemberStringValue)
   if !ok {
    log.Println("Failed to parse the field")
   } else {
    log.Printf("The Movie title field is %s\n", title.Value)
   }
  }
 }
}
```

- 有关 API 详细信息，请参阅《适用于 Go 的 AWS SDK API Reference》中的以下主题。
  - CreateCluster：
  - DescribeClusters
  - DescribeStatement
  - ExecuteStatement
  - GetStatementResult
  - ListDatabasesPaginator
  - ModifyCluster

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

运行一个交互式场景，演示 Amazon Redshift 的功能。

```
import com.example.redshift.User;
import com.google.gson.Gson;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.redshift.model.ClusterAlreadyExistsException;
import software.amazon.awssdk.services.redshift.model.CreateClusterResponse;
import software.amazon.awssdk.services.redshift.model.DeleteClusterResponse;
import software.amazon.awssdk.services.redshift.model.ModifyClusterResponse;
import software.amazon.awssdk.services.redshift.model.RedshiftException;
import
 software.amazon.awssdk.services.redshiftdata.model.ExecuteStatementResponse;
import software.amazon.awssdk.services.redshiftdata.model.RedshiftDataException;
import java.util.Scanner;
import java.util.concurrent.CompletableFuture;
import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 *
 *  This example requires an AWS Secrets Manager secret that contains the
 *  database credentials. If you do not create a
 *  secret that specifies user name and password, this example will not work. For
 details, see:
 *
 *  https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_how-
services-use-secrets_RS.html
 *
 This Java example performs these tasks:
 *
```

```
 * 1. Prompts the user for a unique cluster ID or use the default value.
 * 2. Creates a Redshift cluster with the specified or default cluster Id value.
 * 3. Waits until the Redshift cluster is available for use.
 * 4. Lists all databases using a pagination API call.
 * 5. Creates a table named "Movies" with fields ID, title, and year.
 * 6. Inserts a specified number of records into the "Movies" table by reading
 the Movies JSON file.
 * 7. Prompts the user for a movie release year.
 * 8. Runs a SQL query to retrieve movies released in the specified year.
 * 9. Modifies the Redshift cluster.
 * 10. Prompts the user for confirmation to delete the Redshift cluster.
 * 11. If confirmed, deletes the specified Redshift cluster.
 */

public class RedshiftScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");
    private static final Logger logger =
 LoggerFactory.getLogger(RedshiftScenario.class);

    static RedshiftActions redshiftActions = new RedshiftActions();
    public static void main(String[] args) throws Exception {
        final String usage = """

            Usage:
                <jsonFilePath> <secretName>\s

            Where:
                jsonFilePath - The path to the Movies JSON file (you can locate
 that file in ../../../resources/sample_files/movies.json)
                secretName - The name of the secret that belongs to Secret
 Manager that stores the user name and password used in this scenario.
            """;

        if (args.length != 2) {
            logger.info(usage);
            return;
        }

        String jsonFilePath = args[0];
        String secretName = args[1];
        Scanner scanner = new Scanner(System.in);
        logger.info(DASHES);
        logger.info("Welcome to the Amazon Redshift SDK Basics scenario.");
```

```
        logger.info("""
            This Java program demonstrates how to interact with Amazon Redshift
 by using the AWS SDK for Java (v2).\s
            Amazon Redshift is a fully managed, petabyte-scale data warehouse
 service hosted in the cloud.

            The program's primary functionalities include cluster creation,
 verification of cluster readiness,\s
            list databases, table creation, data population within the table, and
 execution of SQL statements.
            Furthermore, it demonstrates the process of querying data from the
 Movie table.\s

            Upon completion of the program, all AWS resources are cleaned up.
            """);

        logger.info("Lets get started...");
        logger.info("""
            First, we will retrieve the user name and password from Secrets
 Manager.

            Using Amazon Secrets Manager to store Redshift credentials provides
 several security benefits.
            It allows you to securely store and manage sensitive information,
 such as passwords, API keys, and
            database credentials, without embedding them directly in your
 application code.

            More information can be found here:

            https://docs.aws.amazon.com/secretsmanager/latest/userguide/
integrating_how-services-use-secrets_RS.html
            """);
        Gson gson = new Gson();
        User user = gson.fromJson(String.valueOf(getSecretValues(secretName)),
 User.class);
        waitForInputToContinue(scanner);
        logger.info(DASHES);

        try {
            runScenario(user, scanner, jsonFilePath);
        } catch (RuntimeException e) {
            e.printStackTrace();
        } catch (Throwable e) {
```

```
            throw new RuntimeException(e);
        }
    }

    private static void runScenario(User user, Scanner scanner,  String
jsonFilePath) throws Throwable {
        String databaseName = "dev";
        System.out.println(DASHES);
        logger.info("Create a Redshift Cluster");
        logger.info("A Redshift cluster refers to the collection of computing
resources and storage that work together to process and analyze large volumes of
data.");
        logger.info("Enter a cluster id value or accept the default by hitting
Enter (default is redshift-cluster-movies): ");
        String userClusterId = scanner.nextLine();
        String clusterId = userClusterId.isEmpty() ? "redshift-cluster-movies" :
userClusterId;
        try {
            CompletableFuture<CreateClusterResponse> future =
redshiftActions.createClusterAsync(clusterId, user.getUserName(),
user.getUserPassword());
            CreateClusterResponse response = future.join();
            logger.info("Cluster successfully created. Cluster Identifier {} ",
response.cluster().clusterIdentifier());

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof ClusterAlreadyExistsException) {
                logger.info("The Cluster {} already exists. Moving on...",
clusterId);
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
        }
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("Wait until {} is available.", clusterId);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Void> future =
redshiftActions.waitForClusterReadyAsync(clusterId);
            future.join();
            logger.info("Cluster is ready!");
```

```
        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof RedshiftException redshiftEx) {
                logger.info("Redshift error occurred: Error message: {}, Error
 code {}", redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            throw cause;
        }
        logger.info(DASHES);

        logger.info(DASHES);
        String databaseInfo = """
            When you created $clusteridD, the dev database is created by default
 and used in this scenario.\s

            To create a custom database, you need to have a CREATEDB privilege.\s
            For more information, see the documentation here: https://
docs.aws.amazon.com/redshift/latest/dg/r_CREATE_DATABASE.html.
           """.replace("$clusteridD", clusterId);

        logger.info(databaseInfo);
        waitForInputToContinue(scanner);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("List databases in {} ",clusterId);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Void> future =
 redshiftActions.listAllDatabasesAsync(clusterId, user.getUserName(), "dev");
            future.join();
            logger.info("Databases listed successfully.");

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof RedshiftDataException redshiftEx) {
                logger.error("Redshift Data error occurred: {} Error code: {}",
 redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
            } else {
                logger.error("An unexpected error occurred: {}",
 rt.getMessage());
```

```
            }
            throw cause;
        }
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("Now you will create a table named Movies.");
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<ExecuteStatementResponse> future =
redshiftActions.createTableAsync(clusterId, databaseName, user.getUserName());
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof RedshiftDataException redshiftEx) {
                logger.info("Redshift Data error occurred: {} Error code: {}",
redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: {}", rt.getMessage());
            }
            throw cause;
        }
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("Populate the Movies table using the Movies.json file.");
        logger.info("Specify the number of records you would like to add to the
Movies Table.");
        logger.info("Please enter a value between 50 and 200.");
        int numRecords;
        do {
            logger.info("Enter a value: ");
            while (!scanner.hasNextInt()) {
                logger.info("Invalid input. Please enter a value between 50 and
200.");
                logger.info("Enter a year: ");
                scanner.next();
            }
            numRecords = scanner.nextInt();
        } while (numRecords < 50 || numRecords > 200);
        try {
```

```
            redshiftActions.popTableAsync(clusterId, databaseName,
user.getUserName(), jsonFilePath, numRecords).join();  // Wait for the operation
to complete
        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof RedshiftDataException redshiftEx) {
                logger.info("Redshift Data error occurred: {} Error code: {}",
redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: {}", rt.getMessage());
            }
            throw cause;
        }
        waitForInputToContinue(scanner);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("Query the Movies table by year. Enter a value between
2012-2014.");
        int movieYear;
        do {
            logger.info("Enter a year: ");
            while (!scanner.hasNextInt()) {
                logger.info("Invalid input. Please enter a valid year between
2012 and 2014.");
                logger.info("Enter a year: ");
                scanner.next();
            }
            movieYear = scanner.nextInt();
            scanner.nextLine();
        } while (movieYear < 2012 || movieYear > 2014);

        String id;
        try {
            CompletableFuture<String> future =
redshiftActions.queryMoviesByYearAsync(databaseName, user.getUserName(),
movieYear, clusterId);
            id = future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof RedshiftDataException redshiftEx) {
                logger.info("Redshift Data error occurred: {} Error code: {}",
redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
```

```
            } else {
                logger.info("An unexpected error occurred: {}", rt.getMessage());
            }
            throw cause;
        }

        logger.info("The identifier of the statement is " + id);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Void> future =
redshiftActions.checkStatementAsync(id);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof RedshiftDataException redshiftEx) {
                logger.info("Redshift Data error occurred: {} Error code: {}",
redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: {}", rt.getMessage());
            }
            throw cause;
        }
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Void> future = redshiftActions.getResultsAsync(id);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof RedshiftDataException redshiftEx) {
                logger.info("Redshift Data error occurred: {} Error code: {}",
redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: {}", rt.getMessage());
            }
            throw cause;
        }
        waitForInputToContinue(scanner);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("Now you will modify the Redshift cluster.");
        waitForInputToContinue(scanner);
```

```
        try {
            CompletableFuture<ModifyClusterResponse> future =
 redshiftActions.modifyClusterAsync(clusterId);;
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof RedshiftDataException redshiftEx) {
                logger.info("Redshift Data error occurred: {} Error code: {}",
 redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: {}", rt.getMessage());
            }
            throw cause;
        }
        waitForInputToContinue(scanner);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("Would you like to delete the Amazon Redshift cluster? (y/
n)");
        String delAns = scanner.nextLine().trim();
        if (delAns.equalsIgnoreCase("y")) {
            logger.info("You selected to delete {} ", clusterId);
            waitForInputToContinue(scanner);
            try {
                CompletableFuture<DeleteClusterResponse> future =
 redshiftActions.deleteRedshiftClusterAsync(clusterId);;
                future.join();

            } catch (RuntimeException rt) {
                Throwable cause = rt.getCause();
                if (cause instanceof RedshiftDataException redshiftEx) {
                    logger.info("Redshift Data error occurred: {} Error code:
 {}", redshiftEx.getMessage(), redshiftEx.awsErrorDetails().errorCode());
                } else {
                    logger.info("An unexpected error occurred: {}",
 rt.getMessage());
                }
                throw cause;
            }
        } else {
            logger.info("The {}  was not deleted", clusterId);
        }
```

```
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("This concludes the Amazon Redshift SDK Basics scenario.");
        logger.info(DASHES);
    }

    private static SecretsManagerClient getSecretClient() {
        Region region = Region.US_EAST_1;
        return SecretsManagerClient.builder()
            .region(region)
            .build();
    }

    private static void waitForInputToContinue(Scanner scanner) {
        while (true) {
            System.out.println("");
            System.out.println("Enter 'c' followed by <ENTER> to continue:");
            String input = scanner.nextLine();

            if (input.trim().equalsIgnoreCase("c")) {
                System.out.println("Continuing with the program...");
                System.out.println("");
                break;
            } else {
                // Handle invalid input.
                System.out.println("Invalid input. Please try again.");
            }
        }
    }

    // Get the Amazon Redshift credentials from AWS Secrets Manager.
    private static String getSecretValues(String secretName) {
        SecretsManagerClient secretClient = getSecretClient();
        GetSecretValueRequest valueRequest = GetSecretValueRequest.builder()
            .secretId(secretName)
            .build();

        GetSecretValueResponse valueResponse =
 secretClient.getSecretValue(valueRequest);
        return valueResponse.secretString();
    }
}
```

Amazon Redshift SDK 方法的封装类。

```
public class RedshiftActions {

    private static final Logger logger =
 LoggerFactory.getLogger(RedshiftActions.class);
    private static RedshiftDataAsyncClient redshiftDataAsyncClient;

    private static RedshiftAsyncClient redshiftAsyncClient;

    private static RedshiftAsyncClient getAsyncClient() {
        if (redshiftAsyncClient == null) {
            SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
                .maxConcurrency(100)
                .connectionTimeout(Duration.ofSeconds(60))
                .readTimeout(Duration.ofSeconds(60))
                .writeTimeout(Duration.ofSeconds(60))
                .build();

            ClientOverrideConfiguration overrideConfig =
 ClientOverrideConfiguration.builder()
                .apiCallTimeout(Duration.ofMinutes(2))
                .apiCallAttemptTimeout(Duration.ofSeconds(90))
                .retryStrategy(RetryMode.STANDARD)
                .build();

            redshiftAsyncClient = RedshiftAsyncClient.builder()
                .httpClient(httpClient)
                .overrideConfiguration(overrideConfig)
                .build();
        }
        return redshiftAsyncClient;
    }

    private static RedshiftDataAsyncClient getAsyncDataClient() {
        if (redshiftDataAsyncClient == null) {
            SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
                .maxConcurrency(100)
                .connectionTimeout(Duration.ofSeconds(60))
                .readTimeout(Duration.ofSeconds(60))
                .writeTimeout(Duration.ofSeconds(60))
```

```
                .build();

            ClientOverrideConfiguration overrideConfig =
ClientOverrideConfiguration.builder()
                .apiCallTimeout(Duration.ofMinutes(2))
                .apiCallAttemptTimeout(Duration.ofSeconds(90))
                .retryStrategy(RetryMode.STANDARD)
                .build();

            redshiftDataAsyncClient = RedshiftDataAsyncClient.builder()
                .httpClient(httpClient)
                .overrideConfiguration(overrideConfig)
                .build();
        }
        return redshiftDataAsyncClient;
    }

    /**
     * Creates a new Amazon Redshift cluster asynchronously.
     * @param clusterId     the unique identifier for the cluster
     * @param username      the username for the administrative user
     * @param userPassword  the password for the administrative user
     * @return a CompletableFuture that represents the asynchronous operation of
creating the cluster
     * @throws RuntimeException if the cluster creation fails
     */
    public CompletableFuture<CreateClusterResponse> createClusterAsync(String
clusterId, String username, String userPassword) {
        CreateClusterRequest clusterRequest = CreateClusterRequest.builder()
            .clusterIdentifier(clusterId)
            .masterUsername(username)
            .masterUserPassword(userPassword)
            .nodeType("ra3.4xlarge")
            .publiclyAccessible(true)
            .numberOfNodes(2)
            .build();

        return getAsyncClient().createCluster(clusterRequest)
            .whenComplete((response, exception) -> {
                if (response != null) {
                    logger.info("Created cluster ");
                } else {
                    throw new RuntimeException("Failed to create cluster: " +
exception.getMessage(), exception);
```

```
                }
            });
    }

    /**
     * Waits asynchronously for the specified cluster to become available.
     * @param clusterId the identifier of the cluster to wait for
     * @return a {@link CompletableFuture} that completes when the cluster is
ready
     */
    public CompletableFuture<Void> waitForClusterReadyAsync(String clusterId) {
        DescribeClustersRequest clustersRequest =
DescribeClustersRequest.builder()
            .clusterIdentifier(clusterId)
            .build();

        logger.info("Waiting for cluster to become available. This may take a few
minutes.");
        long startTime = System.currentTimeMillis();

        // Recursive method to poll the cluster status.
        return checkClusterStatusAsync(clustersRequest, startTime);
    }

    private CompletableFuture<Void>
checkClusterStatusAsync(DescribeClustersRequest clustersRequest, long startTime)
{
        return getAsyncClient().describeClusters(clustersRequest)
            .thenCompose(clusterResponse -> {
                List<Cluster> clusterList = clusterResponse.clusters();
                boolean clusterReady = false;
                for (Cluster cluster : clusterList) {
                    if ("available".equals(cluster.clusterStatus())) {
                        clusterReady = true;
                        break;
                    }
                }

                if (clusterReady) {
                    logger.info(String.format("Cluster is available!"));
                    return CompletableFuture.completedFuture(null);
                } else {
                    long elapsedTimeMillis = System.currentTimeMillis() -
startTime;
```

```
                    long elapsedSeconds = elapsedTimeMillis / 1000;
                    long minutes = elapsedSeconds / 60;
                    long seconds = elapsedSeconds % 60;
                    System.out.printf("\rElapsed Time: %02d:%02d - Waiting for
cluster...", minutes, seconds);
                    System.out.flush();

                    // Wait 1 second before the next status check
                    return CompletableFuture.runAsync(() -> {
                        try {
                            TimeUnit.SECONDS.sleep(1);
                        } catch (InterruptedException e) {
                            throw new RuntimeException("Error during sleep: " +
e.getMessage(), e);
                        }
                    }).thenCompose(ignored ->
checkClusterStatusAsync(clustersRequest, startTime));
                }
            }).exceptionally(exception -> {
                throw new RuntimeException("Failed to get cluster status: " +
exception.getMessage(), exception);
            });
    }

    /**
     * Lists all databases asynchronously for the specified cluster, database
user, and database.
     * @param clusterId the identifier of the cluster to list databases for
     * @param dbUser the database user to use for the list databases request
     * @param database the database to list databases for
     * @return a {@link CompletableFuture} that completes when the database
listing is complete, or throws a {@link RuntimeException} if there was an error
     */
    public CompletableFuture<Void> listAllDatabasesAsync(String clusterId, String
dbUser, String database) {
        ListDatabasesRequest databasesRequest = ListDatabasesRequest.builder()
            .clusterIdentifier(clusterId)
            .dbUser(dbUser)
            .database(database)
            .build();

        // Asynchronous paginator for listing databases.
        ListDatabasesPublisher databasesPaginator =
getAsyncDataClient().listDatabasesPaginator(databasesRequest);
```

```
        CompletableFuture<Void> future = databasesPaginator.subscribe(response ->
{
            response.databases().forEach(db -> {
                logger.info("The database name is {} ", db);
            });
        });

        // Return the future for asynchronous handling.
        return future.exceptionally(exception -> {
            throw new RuntimeException("Failed to list databases: " +
exception.getMessage(), exception);
        });
    }

    /**
     * Creates an asynchronous task to execute a SQL statement for creating a new
table.
     *
     * @param clusterId    the identifier of the Amazon Redshift cluster
     * @param databaseName the name of the database to create the table in
     * @param userName     the username to use for the database connection
     * @return a {@link CompletableFuture} that completes with the result of the
SQL statement execution
     * @throws RuntimeException if there is an error creating the table
     */
    public CompletableFuture<ExecuteStatementResponse> createTableAsync(String
clusterId, String databaseName, String userName) {
        ExecuteStatementRequest createTableRequest =
ExecuteStatementRequest.builder()
            .clusterIdentifier(clusterId)
            .dbUser(userName)
            .database(databaseName)
            .sql("CREATE TABLE Movies (" +
                "id INT PRIMARY KEY, " +
                "title VARCHAR(100), " +
                "year INT)")
            .build();

        return getAsyncDataClient().executeStatement(createTableRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    throw new RuntimeException("Error creating table: " +
exception.getMessage(), exception);
                } else {
```

```
                    logger.info("Table created: Movies");
                }
            });
    }

    /**
     * Asynchronously pops a table from a JSON file.
     *
     * @param clusterId   the ID of the cluster
     * @param databaseName the name of the database
     * @param userName    the username
     * @param fileName    the name of the JSON file
     * @param number      the number of records to process
     * @return a CompletableFuture that completes with the number of records
added to the Movies table
     */
    public CompletableFuture<Integer> popTableAsync(String clusterId, String
databaseName, String userName, String fileName, int number) {
        return CompletableFuture.supplyAsync(() -> {
                try {
                    JsonParser parser = new JsonFactory().createParser(new
File(fileName));
                    JsonNode rootNode = new ObjectMapper().readTree(parser);
                    Iterator<JsonNode> iter = rootNode.iterator();
                    return iter;
                } catch (IOException e) {
                    throw new RuntimeException("Failed to read or parse JSON
file: " + e.getMessage(), e);
                }
            }).thenCompose(iter -> processNodesAsync(clusterId, databaseName,
userName, iter, number))
            .whenComplete((result, exception) -> {
                if (exception != null) {
                    logger.info("Error {} ", exception.getMessage());
                } else {
                    logger.info("{} records were added to the Movies table." ,
result);
                }
            });
    }

    private CompletableFuture<Integer> processNodesAsync(String clusterId, String
databaseName, String userName, Iterator<JsonNode> iter, int number) {
        return CompletableFuture.supplyAsync(() -> {
```

```
            int t = 0;
            try {
                while (iter.hasNext()) {
                    if (t == number)
                        break;
                    JsonNode currentNode = iter.next();
                    int year = currentNode.get("year").asInt();
                    String title = currentNode.get("title").asText();

                    // Use SqlParameter to avoid SQL injection.
                    List<SqlParameter> parameterList = new ArrayList<>();
                    String sqlStatement = "INSERT INTO Movies
VALUES( :id , :title, :year);";
                    SqlParameter idParam = SqlParameter.builder()
                        .name("id")
                        .value(String.valueOf(t))
                        .build();

                    SqlParameter titleParam = SqlParameter.builder()
                        .name("title")
                        .value(title)
                        .build();

                    SqlParameter yearParam = SqlParameter.builder()
                        .name("year")
                        .value(String.valueOf(year))
                        .build();
                    parameterList.add(idParam);
                    parameterList.add(titleParam);
                    parameterList.add(yearParam);

                    ExecuteStatementRequest insertStatementRequest =
ExecuteStatementRequest.builder()
                        .clusterIdentifier(clusterId)
                        .sql(sqlStatement)
                        .database(databaseName)
                        .dbUser(userName)
                        .parameters(parameterList)
                        .build();


getAsyncDataClient().executeStatement(insertStatementRequest);
                    logger.info("Inserted: " + title + " (" + year + ")");
                    t++;
```

```
                }
            } catch (RedshiftDataException e) {
                throw new RuntimeException("Error inserting data: " +
e.getMessage(), e);
            }
            return t;
        });
    }

    /**
     * Checks the status of an SQL statement asynchronously and handles the
completion of the statement.
     *
     * @param sqlId the ID of the SQL statement to check
     * @return a {@link CompletableFuture} that completes when the SQL
statement's status is either "FINISHED" or "FAILED"
     */
    public CompletableFuture<Void> checkStatementAsync(String sqlId) {
        DescribeStatementRequest statementRequest =
DescribeStatementRequest.builder()
            .id(sqlId)
            .build();

        return getAsyncDataClient().describeStatement(statementRequest)
            .thenCompose(response -> {
                String status = response.statusAsString();
                logger.info("... Status: {} ", status);

                if ("FAILED".equals(status)) {
                    throw new RuntimeException("The Query Failed. Ending
program");
                } else if ("FINISHED".equals(status)) {
                    return CompletableFuture.completedFuture(null);
                } else {
                    // Sleep for 1 second and recheck status
                    return CompletableFuture.runAsync(() -> {
                        try {
                            TimeUnit.SECONDS.sleep(1);
                        } catch (InterruptedException e) {
                            throw new RuntimeException("Error during sleep: " +
e.getMessage(), e);
                        }
                    }).thenCompose(ignore -> checkStatementAsync(sqlId)); //
Recursively call until status is FINISHED or FAILED
```

```
                }
            }).whenComplete((result, exception) -> {
                if (exception != null) {
                    // Handle exceptions
                    logger.info("Error: {} ", exception.getMessage());
                } else {
                    logger.info("The statement is finished!");
                }
            });
    }

    /**
     * Asynchronously retrieves the results of a statement execution.
     *
     * @param statementId the ID of the statement for which to retrieve the
results
     * @return a {@link CompletableFuture} that completes when the statement
result has been processed
     */
    public CompletableFuture<Void> getResultsAsync(String statementId) {
        GetStatementResultRequest resultRequest =
GetStatementResultRequest.builder()
            .id(statementId)
            .build();

        return getAsyncDataClient().getStatementResult(resultRequest)
            .handle((response, exception) -> {
                if (exception != null) {
                    logger.info("Error getting statement result {} ",
exception.getMessage());
                    throw new RuntimeException("Error getting statement result: "
+ exception.getMessage(), exception);
                }

                // Extract and print the field values using streams if the
response is valid.
                response.records().stream()
                    .flatMap(List::stream)
                    .map(Field::stringValue)
                    .filter(value -> value != null)
                    .forEach(value -> System.out.println("The Movie title field
is " + value));

                return response;
```

```
            }).thenAccept(response -> {
                // Optionally add more logic here if needed after handling the
response
            });
    }


    /**
     * Asynchronously queries movies by a given year from a Redshift database.
     *
     * @param database    the name of the database to query
     * @param dbUser      the user to connect to the database with
     * @param year        the year to filter the movies by
     * @param clusterId   the identifier of the Redshift cluster to connect to
     * @return a {@link CompletableFuture} containing the response ID of the
executed SQL statement
     */
    public CompletableFuture<String> queryMoviesByYearAsync(String database,
                                                            String dbUser,
                                                            int year,
                                                            String
clusterId) {

        String sqlStatement = "SELECT * FROM Movies WHERE year = :year";
        SqlParameter yearParam = SqlParameter.builder()
            .name("year")
            .value(String.valueOf(year))
            .build();

        ExecuteStatementRequest statementRequest =
ExecuteStatementRequest.builder()
            .clusterIdentifier(clusterId)
            .database(database)
            .dbUser(dbUser)
            .parameters(yearParam)
            .sql(sqlStatement)
            .build();

        return CompletableFuture.supplyAsync(() -> {
            try {
                ExecuteStatementResponse response =
getAsyncDataClient().executeStatement(statementRequest).join(); // Use join() to
wait for the result
                return response.id();
```

```
            } catch (RedshiftDataException e) {
                throw new RuntimeException("Error executing statement: " +
e.getMessage(), e);
            }
        }).exceptionally(exception -> {
            logger.info("Error: {}", exception.getMessage());
            return "";
        });
    }

    /**
     * Modifies an Amazon Redshift cluster asynchronously.
     *
     * @param clusterId the identifier of the cluster to be modified
     * @return a {@link CompletableFuture} that completes when the cluster
modification is complete
     */
    public CompletableFuture<ModifyClusterResponse> modifyClusterAsync(String
clusterId) {
        ModifyClusterRequest modifyClusterRequest =
ModifyClusterRequest.builder()
            .clusterIdentifier(clusterId)
            .preferredMaintenanceWindow("wed:07:30-wed:08:00")
            .build();

        return getAsyncClient().modifyCluster(modifyClusterRequest)
            .whenComplete((clusterResponse, exception) -> {
                if (exception != null) {
                    if (exception.getCause() instanceof RedshiftException) {
                        logger.info("Error: {} ", exception.getMessage());
                    } else {
                        logger.info("Unexpected error: {} ",
exception.getMessage());
                    }
                } else {
                    logger.info("The modified cluster was successfully modified
and has "
                        + clusterResponse.cluster().preferredMaintenanceWindow()
+ " as the maintenance window");
                }
            });
    }

    /**
```

```
     * Deletes a Redshift cluster asynchronously.
     *
     * @param clusterId the identifier of the Redshift cluster to be deleted
     * @return a {@link CompletableFuture} that represents the asynchronous
 operation of deleting the Redshift cluster
     */
    public CompletableFuture<DeleteClusterResponse>
 deleteRedshiftClusterAsync(String clusterId) {
        DeleteClusterRequest deleteClusterRequest =
 DeleteClusterRequest.builder()
            .clusterIdentifier(clusterId)
            .skipFinalClusterSnapshot(true)
            .build();

        return getAsyncClient().deleteCluster(deleteClusterRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    // Handle exceptions
                    if (exception.getCause() instanceof RedshiftException) {
                        logger.info("Error: {}", exception.getMessage());
                    } else {
                        logger.info("Unexpected error: {}",
 exception.getMessage());
                    }
                } else {
                    // Handle successful response
                    logger.info("The status is {}",
 response.cluster().clusterStatus());
                }
            });
    }
}
```

- 有关 API 详细信息，请参阅《AWS SDK for Java 2.x API Reference》中的以下主题。
  - CreateCluster：
  - DescribeClusters
  - DescribeStatement
  - ExecuteStatement
  - GetStatementResult
  - ListDatabasesPaginator

- ModifyCluster

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class RedshiftScenario:
    """Runs an interactive scenario that shows how to get started with
 Redshift."""

    def __init__(self, redshift_wrapper, redshift_data_wrapper):
        self.redshift_wrapper = redshift_wrapper
        self.redshift_data_wrapper = redshift_data_wrapper

    def redhift_scenario(self, json_file_path):
        database_name = "dev"

        print(DASHES)
        print("Welcome to the Amazon Redshift SDK Getting Started example.")
        print(
            """
      This Python program demonstrates how to interact with Amazon Redshift
      using the AWS SDK for Python (Boto3).

      Amazon Redshift is a fully managed, petabyte-scale data warehouse
      service hosted in the cloud.

      The program's primary functionalities include cluster creation,
      verification of cluster readiness, listing databases, table creation,
      populating data within the table, and executing SQL statements.

      It also demonstrates querying data from the Movies table.

      Upon completion, all AWS resources are cleaned up.
    """
```

```
        )
        if not os.path.isfile(json_file_path):
            logging.error(f"The file {json_file_path} does not exist.")
            return

        print("Let's get started...")
        user_name = q.ask("Please enter your user name (default is awsuser):")
        user_name = user_name if user_name else "awsuser"

        print(DASHES)
        user_password = q.ask(
            "Please enter your user password (default is AwsUser1000):"
        )
        user_password = user_password if user_password else "AwsUser1000"

        print(DASHES)
        print(
            """A Redshift cluster refers to the collection of computing resources
 and storage that work
            together to process and analyze large volumes of data."""
        )
        cluster_id = q.ask(
            "Enter a cluster identifier value (default is redshift-cluster-
movies): "
        )
        cluster_id = cluster_id if cluster_id else "redshift-cluster-movies"

        self.redshift_wrapper.create_cluster(
            cluster_id, "ra3.4xlarge", user_name, user_password, True, 2
        )

        print(DASHES)
        print(f"Wait until {cluster_id} is available. This may take a few
 minutes...")
        q.ask("Press Enter to continue...")

        self.wait_cluster_available(cluster_id)

        print(DASHES)

        print(
            f"""
       When you created {cluster_id}, the dev database is created by default and
 used in this scenario.
```

```
        To create a custom database, you need to have a CREATEDB privilege.
        For more information, see the documentation here:
        https://docs.aws.amazon.com/redshift/latest/dg/r_CREATE_DATABASE.html.
       """
        )
        q.ask("Press Enter to continue...")
        print(DASHES)

        print(DASHES)
        print(f"List databases in {cluster_id}")
        q.ask("Press Enter to continue...")
        databases = self.redshift_data_wrapper.list_databases(
            cluster_id, database_name, user_name
        )
        print(f"The cluster contains {len(databases)} database(s).")
        for database in databases:
            print(f"    Database: {database}")
        print(DASHES)

        print(DASHES)
        print("Now you will create a table named Movies.")
        q.ask("Press Enter to continue...")

        self.create_table(cluster_id, database_name, user_name)

        print(DASHES)

        print("Populate the Movies table using the Movies.json file.")
        print(
            "Specify the number of records you would like to add to the Movies
Table."
        )
        print("Please enter a value between 50 and 200.")

        while True:
            try:
                num_records = int(q.ask("Enter a value: ", q.is_int))
                if 50 <= num_records <= 200:
                    break
                else:
                    print("Invalid input. Please enter a value between 50 and
200.")
            except ValueError:
```

```
                print("Invalid input. Please enter a value between 50 and 200.")

        self.populate_table(
            cluster_id, database_name, user_name, json_file_path, num_records
        )

        print(DASHES)
        print("Query the Movies table by year. Enter a value between 2012-2014.")

        while True:
            movie_year = int(q.ask("Enter a year: ", q.is_int))
            if 2012 <= movie_year <= 2014:
                break
            else:
                print("Invalid input. Please enter a valid year between 2012 and
2014.")

        # Function to query database
        sql_id = self.query_movies_by_year(
            database_name, user_name, movie_year, cluster_id
        )

        print(f"The identifier of the statement is {sql_id}")

        print("Checking statement status...")
        self.wait_statement_finished(sql_id)
        result = self.redshift_data_wrapper.get_statement_result(sql_id)

        self.display_movies(result)

        print(DASHES)

        print(DASHES)
        print("Now you will modify the Redshift cluster.")
        q.ask("Press Enter to continue...")

        preferred_maintenance_window = "wed:07:30-wed:08:00"
        self.redshift_wrapper.modify_cluster(cluster_id,
preferred_maintenance_window)

        print(DASHES)

        print(DASHES)
        delete = q.ask("Do you want to delete the cluster? (y/n) ", q.is_yesno)
```

```
        if delete:
            print(f"You selected to delete {cluster_id}")
            q.ask("Press Enter to continue...")
            self.redshift_wrapper.delete_cluster(cluster_id)
        else:
            print(f"Cluster {cluster_id}cluster_id was not deleted")

        print(DASHES)
        print("This concludes the Amazon Redshift SDK Getting Started scenario.")
        print(DASHES)

    def create_table(self, cluster_id, database, username):
        self.redshift_data_wrapper.execute_statement(
            cluster_identifier=cluster_id,
            database_name=database,
            user_name=username,
            sql="CREATE TABLE Movies (statement_id INT PRIMARY KEY, title
VARCHAR(100), year INT)",
        )

        print("Table created: Movies")


    def populate_table(self, cluster_id, database, username, file_name, number):
        with open(file_name) as f:
            data = json.load(f)

        i = 0
        for record in data:
            if i == number:
                break

            statement_id = i
            title = record["title"]
            year = record["year"]
            i = i + 1
            parameters = [
                {"name": "statement_id", "value": str(statement_id)},
                {"name": "title", "value": title},
                {"name": "year", "value": str(year)},
            ]

            self.redshift_data_wrapper.execute_statement(
```

```
                cluster_identifier=cluster_id,
                database_name=database,
                user_name=username,
                sql="INSERT INTO Movies VALUES(:statement_id, :title, :year)",
                parameter_list=parameters,
            )

        print(f"{i} records inserted into Movies table")

    def wait_cluster_available(self, cluster_id):
        """
        Waits for a cluster to be available.

        :param cluster_id: The cluster identifier.

        Note: The cluster_available waiter can also be used.
        It is not used in this case to allow an elapsed time message.
        """
        cluster_ready = False
        start_time = time.time()

        while not cluster_ready:
            time.sleep(30)
            cluster = self.redshift_wrapper.describe_clusters(cluster_id)
            status = cluster[0]["ClusterStatus"]
            if status == "available":
                cluster_ready = True
            elif status != "creating":
                raise Exception(
                    f"Cluster {cluster_id} creation failed with status {status}."
                )

            elapsed_seconds = int(round(time.time() - start_time))
            minutes = int(elapsed_seconds // 60)
            seconds = int(elapsed_seconds % 60)

            print(f"Elapsed Time: {minutes}:{seconds:02d} - status {status}...")

            if minutes > 30:
                raise Exception(
                    f"Cluster {cluster_id} is not available after 30 minutes."
                )

    def query_movies_by_year(self, database, username, year, cluster_id):
```

```
        sql = "SELECT * FROM Movies WHERE year = :year"

        params = [{"name": "year", "value": str(year)}]

        response = self.redshift_data_wrapper.execute_statement(
            cluster_identifier=cluster_id,
            database_name=database,
            user_name=username,
            sql=sql,
            parameter_list=params,
        )

        return response["Id"]

    @staticmethod
    def display_movies(response):
        metadata = response["ColumnMetadata"]
        records = response["Records"]

        title_column_index = None
        for i in range(len(metadata)):
            if metadata[i]["name"] == "title":
                title_column_index = i
                break

        if title_column_index is None:
            print("No title column found.")
            return

        print(f"Found {len(records)} movie(s).")
        for record in records:
            print(f"   {record[title_column_index]['stringValue']}")

    def wait_statement_finished(self, sql_id):
        while True:
            time.sleep(1)
            response = self.redshift_data_wrapper.describe_statement(sql_id)
            status = response["Status"]
            print(f"Statement status is {status}.")

            if status == "FAILED":
                print(f"The query failed because {response['Error']}. Ending
program")
                raise Exception("The Query Failed. Ending program")
```

```
            elif status == "FINISHED":
                break
```

用于显示场景实施的主函数。

```
def main():
    redshift_client = boto3.client("redshift")
    redshift_data_client = boto3.client("redshift-data")
    redshift_wrapper = RedshiftWrapper(redshift_client)
    redshift_data_wrapper = RedshiftDataWrapper(redshift_data_client)
    redshift_scenario = RedshiftScenario(redshift_wrapper, redshift_data_wrapper)
    redshift_scenario.redhift_scenario(
        f"{os.path.dirname(__file__)}/../../../resources/sample_files/
movies.json"
    )
```

场景中使用的包装器函数。

```
    def create_cluster(
        self,
        cluster_identifier,
        node_type,
        master_username,
        master_user_password,
        publicly_accessible,
        number_of_nodes,
    ):
        """
        Creates a cluster.

        :param cluster_identifier: The name of the cluster.
        :param node_type: The type of node in the cluster.
        :param master_username: The master username.
        :param master_user_password: The master user password.
        :param publicly_accessible: Whether the cluster is publicly accessible.
        :param number_of_nodes: The number of nodes in the cluster.
        :return: The cluster.
```

```
        """

        try:
            cluster = self.client.create_cluster(
                ClusterIdentifier=cluster_identifier,
                NodeType=node_type,
                MasterUsername=master_username,
                MasterUserPassword=master_user_password,
                PubliclyAccessible=publicly_accessible,
                NumberOfNodes=number_of_nodes,
            )
            return cluster
        except ClientError as err:
            logging.error(
                "Couldn't create a cluster. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def describe_clusters(self, cluster_identifier):
        """
        Describes a cluster.

        :param cluster_identifier: The cluster identifier.
        :return: A list of clusters.
        """
        try:
            kwargs = {}
            if cluster_identifier:
                kwargs["ClusterIdentifier"] = cluster_identifier

            paginator = self.client.get_paginator("describe_clusters")
            clusters = []
            for page in paginator.paginate(**kwargs):
                clusters.extend(page["Clusters"])

            return clusters

        except ClientError as err:
            logging.error(
                "Couldn't describe a cluster. Here's why: %s: %s",
                err.response["Error"]["Code"],
```

```
                err.response["Error"]["Message"],
            )
            raise


    def execute_statement(
        self, cluster_identifier, database_name, user_name, sql,
parameter_list=None
    ):
        """
        Executes a SQL statement.

        :param cluster_identifier: The cluster identifier.
        :param database_name: The database name.
        :param user_name: The user's name.
        :param sql: The SQL statement.
        :param parameter_list: The optional SQL statement parameters.
        :return: The SQL statement result.
        """

        try:
            kwargs = {
                "ClusterIdentifier": cluster_identifier,
                "Database": database_name,
                "DbUser": user_name,
                "Sql": sql,
            }
            if parameter_list:
                kwargs["Parameters"] = parameter_list
            response = self.client.execute_statement(**kwargs)
            return response
        except ClientError as err:
            logging.error(
                "Couldn't execute statement. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def describe_statement(self, statement_id):
        """
        Describes a SQL statement.
```

```
        :param statement_id: The SQL statement identifier.
        :return: The SQL statement result.
        """
        try:
            response = self.client.describe_statement(Id=statement_id)
            return response
        except ClientError as err:
            logging.error(
                "Couldn't describe statement. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def get_statement_result(self, statement_id):
        """
        Gets the result of a SQL statement.

        :param statement_id: The SQL statement identifier.
        :return: The SQL statement result.
        """
        try:
            result = {
                "Records": [],
            }
            paginator = self.client.get_paginator("get_statement_result")
            for page in paginator.paginate(Id=statement_id):
                if "ColumnMetadata" not in result:
                    result["ColumnMetadata"] = page["ColumnMetadata"]
                result["Records"].extend(page["Records"])
            return result
        except ClientError as err:
            logging.error(
                "Couldn't get statement result. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def modify_cluster(self, cluster_identifier, preferred_maintenance_window):
        """
        Modifies a cluster.
```

```
        :param cluster_identifier: The cluster identifier.
        :param preferred_maintenance_window: The preferred maintenance window.
        """
        try:
            self.client.modify_cluster(
                ClusterIdentifier=cluster_identifier,
                PreferredMaintenanceWindow=preferred_maintenance_window,
            )
        except ClientError as err:
            logging.error(
                "Couldn't modify a cluster. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def list_databases(self, cluster_identifier, database_name, database_user):
        """
        Lists databases in a cluster.

        :param cluster_identifier: The cluster identifier.
        :param database_name: The database name.
        :param database_user: The database user.
        :return: The list of databases.
        """
        try:
            paginator = self.client.get_paginator("list_databases")
            databases = []
            for page in paginator.paginate(
                ClusterIdentifier=cluster_identifier,
                Database=database_name,
                DbUser=database_user,
            ):
                databases.extend(page["Databases"])

            return databases
        except ClientError as err:
            logging.error(
                "Couldn't list databases. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
```

```
            raise


    def delete_cluster(self, cluster_identifier):
        """
        Deletes a cluster.

        :param cluster_identifier: The cluster identifier.
        """
        try:
            self.client.delete_cluster(
                ClusterIdentifier=cluster_identifier,
SkipFinalClusterSnapshot=True
            )
        except ClientError as err:
            logging.error(
                "Couldn't delete a cluster. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 详细信息，请参阅《AWS SDK for Python (Boto3) API Reference》中的以下主题。
  - CreateCluster：
  - DescribeClusters
  - DescribeStatement
  - ExecuteStatement
  - GetStatementResult
  - ListDatabasesPaginator
  - ModifyCluster

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

# 使用 AWS SDK 对 Amazon Redshift 执行的操作

以下代码示例演示了如何使用 AWS SDK 来执行各个 Amazon Redshift 操作。每个示例都包含一个指向 GitHub 的链接，您可以在其中找到有关设置和运行代码的说明。

这些代码节选调用了 Amazon Redshift API，是必须在上下文中运行的较大型程序的代码节选。您可以在使用 AWS SDK 的 Amazon Redshift 场景中结合上下文查看操作。

以下示例仅包括最常用的操作。有关完整列表，请参阅 Amazon Redshift API 参考。

示例

- 将 CreateCluster 与 AWS SDK 或 CLI 配合使用
- 将 DeleteCluster 与 AWS SDK 或 CLI 配合使用
- 将 DescribeClusters 与 AWS SDK 或 CLI 配合使用
- 将 DescribeStatement 和 AWS SDK 搭配使用
- 将 ExecuteStatement 和 AWS SDK 搭配使用
- 将 GetStatementResult 和 AWS SDK 搭配使用
- 将 ListDatabases 和 AWS SDK 搭配使用
- 将 ModifyCluster 与 AWS SDK 或 CLI 配合使用

## 将 `CreateCluster` 与 AWS SDK 或 CLI 配合使用

以下代码示例演示如何使用 `CreateCluster`。

操作示例是大型程序的代码摘录，必须在上下文中运行。您可以在以下代码示例中查看此操作的上下文：

- 了解基本功能
- 预调配数据仓库集群入门

.NET

适用于 .NET 的 SDK (v4)

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /// <summary>
    /// Create a new Amazon Redshift cluster.
    /// </summary>
    /// <param name="clusterIdentifier">The identifier for the cluster.</param>
    /// <param name="databaseName">The name of the database.</param>
    /// <param name="masterUsername">The master username.</param>
    /// <param name="masterUserPassword">The master user password.</param>
    /// <param name="nodeType">The node type for the cluster.</param>
    /// <returns>The cluster that was created.</returns>
    public async Task<Cluster> CreateClusterAsync(string clusterIdentifier,
string databaseName,
        string masterUsername, string masterUserPassword, string nodeType =
"ra3.large")
    {
        try
        {
            var request = new CreateClusterRequest
            {
                ClusterIdentifier = clusterIdentifier,
                DBName = databaseName,
                MasterUsername = masterUsername,
                MasterUserPassword = masterUserPassword,
                NodeType = nodeType,
                NumberOfNodes = 1,
                ClusterType = "single-node"
            };

            var response = await _redshiftClient.CreateClusterAsync(request);
            Console.WriteLine($"Created cluster {clusterIdentifier}");
            return response.Cluster;
        }
        catch (ClusterAlreadyExistsException ex)
```

```
        {
            Console.WriteLine($"Cluster already exists: {ex.Message}");
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't create cluster. Here's why:
 {ex.Message}");
            throw;
        }
    }
```

- 有关 API 详细信息，请参阅《适用于 .NET 的 AWS SDK API Reference》中的 CreateCluster。

## CLI

### AWS CLI

使用最少参数创建集群 此示例使用最少参数集创建集群。默认情况下，输出采用 JSON 格式。
命令：

```
aws redshift create-cluster --node-type dw.hs1.xlarge --number-of-nodes 2 --
master-username adminuser --master-user-password TopSecret1 --cluster-identifier
 mycluster
```

结果：

```
{
   "Cluster": {
      "NodeType": "dw.hs1.xlarge",
      "ClusterVersion": "1.0",
      "PubliclyAccessible": "true",
      "MasterUsername": "adminuser",
      "ClusterParameterGroups": [
         {
            "ParameterApplyStatus": "in-sync",
            "ParameterGroupName": "default.redshift-1.0"
         } ],
      "ClusterSecurityGroups": [
         {
```

```
            "Status": "active",
            "ClusterSecurityGroupName": "default"
         } ],
      "AllowVersionUpgrade": true,
      "VpcSecurityGroups": \[],
      "PreferredMaintenanceWindow": "sat:03:30-sat:04:00",
      "AutomatedSnapshotRetentionPeriod": 1,
      "ClusterStatus": "creating",
      "ClusterIdentifier": "mycluster",
      "DBName": "dev",
      "NumberOfNodes": 2,
      "PendingModifiedValues": {
         "MasterUserPassword": "\****"
      }
   },
   "ResponseMetadata": {
      "RequestId": "7cf4bcfc-64dd-11e2-bea9-49e0ce183f07"
   }
}
```

- 有关 API 详细信息，请参阅《AWS CLI 命令参考》中的 CreateCluster。

Go

适用于 Go 的 SDK V2

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
import (
 "context"
 "errors"
 "log"
 "time"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/redshift"
 "github.com/aws/aws-sdk-go-v2/service/redshift/types"
```

```
)



// RedshiftActions wraps Redshift service actions.
type RedshiftActions struct {
 RedshiftClient *redshift.Client
}



// CreateCluster sends a request to create a cluster with the given clusterId
 using the provided credentials.
func (actor RedshiftActions) CreateCluster(ctx context.Context, clusterId string,
 userName string, userPassword string, nodeType string, clusterType string,
 publiclyAccessible bool) (*redshift.CreateClusterOutput, error) {
 // Create a new Redshift cluster
 input := &redshift.CreateClusterInput{
  ClusterIdentifier:  aws.String(clusterId),
  MasterUserPassword: aws.String(userPassword),
  MasterUsername:     aws.String(userName),
  NodeType:           aws.String(nodeType),
  ClusterType:        aws.String(clusterType),
  PubliclyAccessible: aws.Bool(publiclyAccessible),
 }
 var opErr *types.ClusterAlreadyExistsFault
 output, err := actor.RedshiftClient.CreateCluster(ctx, input)
 if err != nil && errors.As(err, &opErr) {
  log.Println("Cluster already exists")
  return nil, nil
 } else if err != nil {
  log.Printf("Failed to create Redshift cluster: %v\n", err)
  return nil, err
 }

 log.Printf("Created cluster %s\n", *output.Cluster.ClusterIdentifier)
 return output, nil
}
```

- 有关 API 详细信息，请参阅《适用于 Go 的 AWS SDK API Reference》中的 CreateCluster。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

创建集群。

```
    /**
     * Creates a new Amazon Redshift cluster asynchronously.
     * @param clusterId     the unique identifier for the cluster
     * @param username      the username for the administrative user
     * @param userPassword  the password for the administrative user
     * @return a CompletableFuture that represents the asynchronous operation of
creating the cluster
     * @throws RuntimeException if the cluster creation fails
     */
    public CompletableFuture<CreateClusterResponse> createClusterAsync(String
clusterId, String username, String userPassword) {
        CreateClusterRequest clusterRequest = CreateClusterRequest.builder()
            .clusterIdentifier(clusterId)
            .masterUsername(username)
            .masterUserPassword(userPassword)
            .nodeType("ra3.4xlarge")
            .publiclyAccessible(true)
            .numberOfNodes(2)
            .build();

        return getAsyncClient().createCluster(clusterRequest)
            .whenComplete((response, exception) -> {
                if (response != null) {
                    logger.info("Created cluster ");
                } else {
                    throw new RuntimeException("Failed to create cluster: " +
exception.getMessage(), exception);
                }
            });
    }
```

- 有关 API 详细信息，请参阅《AWS SDK for Java 2.x API Reference》中的 CreateCluster。

JavaScript

SDK for JavaScript (v3)

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

创建客户端。

```
import { RedshiftClient } from "@aws-sdk/client-redshift";
// Set the AWS Region.
const REGION = "REGION";
//Set the Redshift Service Object
const redshiftClient = new RedshiftClient({ region: REGION });
export { redshiftClient };
```

创建集群。

```
// Import required AWS SDK clients and commands for Node.js
import { CreateClusterCommand } from "@aws-sdk/client-redshift";
import { redshiftClient } from "./libs/redshiftClient.js";

const params = {
  ClusterIdentifier: "CLUSTER_NAME", // Required
  NodeType: "NODE_TYPE", //Required
  MasterUsername: "MASTER_USER_NAME", // Required - must be lowercase
  MasterUserPassword: "MASTER_USER_PASSWORD", // Required - must contain at least
 one uppercase letter, and one number
  ClusterType: "CLUSTER_TYPE", // Required
  IAMRoleARN: "IAM_ROLE_ARN", // Optional - the ARN of an IAM role with
 permissions your cluster needs to access other AWS services on your behalf, such
 as Amazon S3.
```

```
  ClusterSubnetGroupName: "CLUSTER_SUBNET_GROUPNAME", //Optional - the name of a
 cluster subnet group to be associated with this cluster. Defaults to 'default'
 if not specified.
  DBName: "DATABASE_NAME", // Optional - defaults to 'dev' if not specified
  Port: "PORT_NUMBER", // Optional - defaults to '5439' if not specified
};

const run = async () => {
  try {
    const data = await redshiftClient.send(new CreateClusterCommand(params));
    console.log(
      `Cluster ${data.Cluster.ClusterIdentifier} successfully created`,
    );
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- 有关 API 详细信息，请参阅《适用于 JavaScript 的 AWS SDK API Reference》中的 CreateCluster。

Kotlin

SDK for Kotlin

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

创建集群。

```
suspend fun createCluster(
    clusterId: String?,
    masterUsernameVal: String?,
    masterUserPasswordVal: String?,
) {
```

```
    val clusterRequest =
        CreateClusterRequest {
            clusterIdentifier = clusterId
            availabilityZone = "us-east-1a"
            masterUsername = masterUsernameVal
            masterUserPassword = masterUserPasswordVal
            nodeType = "ra3.4xlarge"
            publiclyAccessible = true
            numberOfNodes = 2
        }

    RedshiftClient.fromEnvironment { region = "us-east-1" }.use { redshiftClient
 ->
        val clusterResponse = redshiftClient.createCluster(clusterRequest)
        println("Created cluster ${clusterResponse.cluster?.clusterIdentifier}")
    }
}
```

- 有关 API 详细信息，请参阅《AWS SDK for Kotlin API Reference》中的 CreateCluster。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class RedshiftWrapper:
    """
    Encapsulates Amazon Redshift cluster operations.
    """

    def __init__(self, redshift_client):
        """
        :param redshift_client: A Boto3 Redshift client.
        """
        self.client = redshift_client
```

```
    def create_cluster(
        self,
        cluster_identifier,
        node_type,
        master_username,
        master_user_password,
        publicly_accessible,
        number_of_nodes,
    ):
        """
        Creates a cluster.

        :param cluster_identifier: The name of the cluster.
        :param node_type: The type of node in the cluster.
        :param master_username: The master username.
        :param master_user_password: The master user password.
        :param publicly_accessible: Whether the cluster is publicly accessible.
        :param number_of_nodes: The number of nodes in the cluster.
        :return: The cluster.
        """

        try:
            cluster = self.client.create_cluster(
                ClusterIdentifier=cluster_identifier,
                NodeType=node_type,
                MasterUsername=master_username,
                MasterUserPassword=master_user_password,
                PubliclyAccessible=publicly_accessible,
                NumberOfNodes=number_of_nodes,
            )
            return cluster
        except ClientError as err:
            logging.error(
                "Couldn't create a cluster. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

以下代码会实例化 RedshiftWrapper 对象。

```
    client = boto3.client("redshift")
    redhift_wrapper = RedshiftWrapper(client)
```

- 有关 API 详细信息，请参阅《AWS SDK for Python（Boto3）API Reference》中的 CreateCluster。

## SAP ABAP

### 适用于 SAP ABAP 的 SDK

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

创建集群。

```
    TRY.
        " Example values: iv_cluster_identifier = 'my-redshift-cluster'
        " Example values: iv_node_type = 'ra3.4xlarge'
        " Example values: iv_master_username = 'awsuser'
        " Example values: iv_master_password = 'AwsUser1000'
        " Example values: iv_publicly_accessible = abap_true
        " Example values: iv_number_of_nodes = 2
        oo_result = lo_rsh->createcluster(
          iv_clusteridentifier = iv_cluster_identifier
          iv_nodetype = iv_node_type
          iv_masterusername = iv_master_username
          iv_masteruserpassword = iv_master_password
          iv_publiclyaccessible = iv_publicly_accessible
          iv_numberofnodes = iv_number_of_nodes
        ).
        MESSAGE 'Redshift cluster created successfully.' TYPE 'I'.
      CATCH /aws1/cx_rshclustalrdyexfault.
        MESSAGE 'Cluster already exists.' TYPE 'I'.
      CATCH /aws1/cx_rshclstquotaexcdfault.
        MESSAGE 'Cluster quota exceeded.' TYPE 'I'.
    ENDTRY.
```

- 有关 API 详细信息，请参阅《AWS SDK for SAP ABAP API Reference》中的 CreateCluster。

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

## 将 **DeleteCluster** 与 AWS SDK 或 CLI 配合使用

以下代码示例演示如何使用 DeleteCluster。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 预调配数据仓库集群入门

.NET

适用于 .NET 的 SDK (v4)

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /// <summary>
    /// Delete an Amazon Redshift cluster without a final snapshot.
    /// </summary>
    /// <param name="clusterIdentifier">The identifier for the cluster.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteClusterWithoutSnapshotAsync(string
clusterIdentifier)
    {
        try
        {
            var request = new DeleteClusterRequest
            {
                ClusterIdentifier = clusterIdentifier,
                SkipFinalClusterSnapshot = true
            };
```

```
            var response = await _redshiftClient.DeleteClusterAsync(request);
            Console.WriteLine($"The {clusterIdentifier} was deleted");
            return true;
        }
        catch (ClusterNotFoundException ex)
        {
            Console.WriteLine($"Cluster not found: {ex.Message}");
            return false;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't delete cluster. Here's why:
{ex.Message}");
            return false;
        }
    }
```

- 有关 API 详细信息，请参阅《适用于 .NET 的 AWS SDK API Reference》中的 DeleteCluster。

CLI

AWS CLI

删除没有最终集群快照的集群 此示例删除集群，强制删除数据，因此不会创建最终集群快照。命令：

```
aws redshift delete-cluster --cluster-identifier mycluster --skip-final-cluster-
snapshot
```

删除集群，支持使用最终集群快照 此示例删除集群，但指定最终集群快照。命令：

```
aws redshift delete-cluster --cluster-identifier mycluster --final-cluster-
snapshot-identifier myfinalsnapshot
```

- 有关 API 详细信息，请参阅《AWS CLI 命令参考》中的 DeleteCluster。

Go

适用于 Go 的 SDK V2

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
import (
 "context"
 "errors"
 "log"
 "time"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/redshift"
 "github.com/aws/aws-sdk-go-v2/service/redshift/types"
)



// RedshiftActions wraps Redshift service actions.
type RedshiftActions struct {
 RedshiftClient *redshift.Client
}



// DeleteCluster deletes the given cluster.
func (actor RedshiftActions) DeleteCluster(ctx context.Context, clusterId string)
 (bool, error) {
 input := redshift.DeleteClusterInput{
  ClusterIdentifier:        aws.String(clusterId),
  SkipFinalClusterSnapshot: aws.Bool(true),
 }
 _, err := actor.RedshiftClient.DeleteCluster(ctx, &input)
 var opErr *types.ClusterNotFoundFault
 if err != nil && errors.As(err, &opErr) {
  log.Println("Cluster was not found. Where could it be?")
```

```
  return false, err
 } else if err != nil {
  log.Printf("Failed to delete Redshift cluster: %v\n", err)
  return false, err
 }
 waiter := redshift.NewClusterDeletedWaiter(actor.RedshiftClient)
 err = waiter.Wait(ctx, &redshift.DescribeClustersInput{
  ClusterIdentifier: aws.String(clusterId),
 }, 5*time.Minute)
 if err != nil {
  log.Printf("Wait time exceeded for deleting cluster, continuing: %v\n", err)
 }
 log.Printf("The cluster %s was deleted\n", clusterId)
 return true, nil
}
```

- 有关 API 详细信息，请参阅《适用于 Go 的 AWS SDK API Reference》中的 DeleteCluster。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

请删除集群。

```
    /**
     * Deletes a Redshift cluster asynchronously.
     *
     * @param clusterId the identifier of the Redshift cluster to be deleted
     * @return a {@link CompletableFuture} that represents the asynchronous
operation of deleting the Redshift cluster
     */
    public CompletableFuture<DeleteClusterResponse>
deleteRedshiftClusterAsync(String clusterId) {
```

```
        DeleteClusterRequest deleteClusterRequest =
DeleteClusterRequest.builder()
            .clusterIdentifier(clusterId)
            .skipFinalClusterSnapshot(true)
            .build();

        return getAsyncClient().deleteCluster(deleteClusterRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    // Handle exceptions
                    if (exception.getCause() instanceof RedshiftException) {
                        logger.info("Error: {}", exception.getMessage());
                    } else {
                        logger.info("Unexpected error: {}",
exception.getMessage());
                    }
                } else {
                    // Handle successful response
                    logger.info("The status is {}",
response.cluster().clusterStatus());
                }
            });
    }
```

- 有关 API 详细信息，请参阅《AWS SDK for Java 2.x API Reference》中的 DeleteCluster。

JavaScript

SDK for JavaScript (v3)

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

创建客户端。

```
import { RedshiftClient } from "@aws-sdk/client-redshift";
// Set the AWS Region.
const REGION = "REGION";
```

```
//Set the Redshift Service Object
const redshiftClient = new RedshiftClient({ region: REGION });
export { redshiftClient };
```

创建集群。

```
// Import required AWS SDK clients and commands for Node.js
import { DeleteClusterCommand } from "@aws-sdk/client-redshift";
import { redshiftClient } from "./libs/redshiftClient.js";

const params = {
  ClusterIdentifier: "CLUSTER_NAME",
  SkipFinalClusterSnapshot: false,
  FinalClusterSnapshotIdentifier: "CLUSTER_SNAPSHOT_ID",
};

const run = async () => {
  try {
    const data = await redshiftClient.send(new DeleteClusterCommand(params));
    console.log("Success, cluster deleted. ", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- 有关 API 详细信息，请参阅《适用于 JavaScript 的 AWS SDK API Reference》中的 DeleteCluster。

Kotlin

SDK for Kotlin

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

请删除集群。

```
suspend fun deleteRedshiftCluster(clusterId: String?) {
    val request =
        DeleteClusterRequest {
            clusterIdentifier = clusterId
            skipFinalClusterSnapshot = true
        }

    RedshiftClient.fromEnvironment { region = "us-west-2" }.use { redshiftClient
 ->
        val response = redshiftClient.deleteCluster(request)
        println("The status is ${response.cluster?.clusterStatus}")
    }
}
```

- 有关 API 详细信息，请参阅《AWS SDK for Kotlin API Reference》中的 DeleteCluster。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class RedshiftWrapper:
    """
    Encapsulates Amazon Redshift cluster operations.
    """

    def __init__(self, redshift_client):
        """
        :param redshift_client: A Boto3 Redshift client.
        """
        self.client = redshift_client
```

```
    def delete_cluster(self, cluster_identifier):
        """
        Deletes a cluster.

        :param cluster_identifier: The cluster identifier.
        """
        try:
            self.client.delete_cluster(
                ClusterIdentifier=cluster_identifier,
SkipFinalClusterSnapshot=True
            )
        except ClientError as err:
            logging.error(
                "Couldn't delete a cluster. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

以下代码会实例化 RedshiftWrapper 对象。

```
    client = boto3.client("redshift")
    redhift_wrapper = RedshiftWrapper(client)
```

- 有关 API 详细信息，请参阅《AWS SDK for Python（Boto3）API Reference》中的 DeleteCluster。

SAP ABAP

适用于 SAP ABAP 的 SDK

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

请删除集群。

```
    TRY.
        " Example values: iv_cluster_identifier = 'my-redshift-cluster'
        lo_rsh->deletecluster(
          iv_clusteridentifier = iv_cluster_identifier
          iv_skipfinalclustersnapshot = abap_true
        ).
        MESSAGE 'Redshift cluster deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_rshclustnotfoundfault.
        MESSAGE 'Cluster not found.' TYPE 'I'.
      CATCH /aws1/cx_rshinvcluststatefault.
        MESSAGE 'Invalid cluster state for deletion.' TYPE 'I'.
    ENDTRY.
```

- 有关 API 详细信息，请参阅《AWS SDK for SAP ABAP API Reference》中的 DeleteCluster。

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

## 将 **DescribeClusters** 与 AWS SDK 或 CLI 配合使用

以下代码示例演示如何使用 DescribeClusters。

操作示例是大型程序的代码摘录，必须在上下文中运行。您可以在以下代码示例中查看此操作的上下文：

- 了解基本功能
- 预调配数据仓库集群入门

.NET

适用于 .NET 的 SDK (v4)

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   /// <summary>
   /// Describe Amazon Redshift clusters.
   /// </summary>
   /// <param name="clusterIdentifier">Optional cluster identifier to describe a
specific cluster.</param>
   /// <returns>A list of clusters.</returns>
   public async Task<List<Cluster>> DescribeClustersAsync(string?
clusterIdentifier = null)
   {
       try
       {
           var clusters = new List<Cluster>();
           var request = new DescribeClustersRequest();
           if (!string.IsNullOrEmpty(clusterIdentifier))
           {
               request.ClusterIdentifier = clusterIdentifier;
           }

           var clustersPaginator =
_redshiftClient.Paginators.DescribeClusters(request);
           await foreach (var response in clustersPaginator.Responses)
           {
               if (response.Clusters != null)
                   clusters.AddRange(response.Clusters);
           }

           Console.WriteLine($"{clusters.Count} cluster(s) retrieved.");
           foreach (var cluster in clusters)
           {
               Console.WriteLine($"\t{cluster.ClusterIdentifier} (Status:
{cluster.ClusterStatus})");
           }

           return clusters;
       }
       catch (ClusterNotFoundException ex)
       {
           Console.WriteLine($"Cluster {clusterIdentifier} not found:
{ex.Message}");
           throw;
       }
       catch (Exception ex)
       {
```

```
            Console.WriteLine($"Couldn't describe clusters. Here's why:
 {ex.Message}");
            throw;
        }
    }
```

- 有关 API 详细信息，请参阅《适用于 .NET 的 AWS SDK API Reference》中的 DescribeClusters。

## CLI

### AWS CLI

**获取所有集群的描述** 此示例返回账户中所有集群的描述。默认情况下，输出采用 JSON 格式。
命令：

```
aws redshift describe-clusters
```

结果：

```
{
   "Clusters": [
   {
      "NodeType": "dw.hs1.xlarge",
      "Endpoint": {
         "Port": 5439,
         "Address": "mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com"
      },
      "ClusterVersion": "1.0",
      "PubliclyAccessible": "true",
      "MasterUsername": "adminuser",
      "ClusterParameterGroups": [
         {
            "ParameterApplyStatus": "in-sync",
            "ParameterGroupName": "default.redshift-1.0"
         } ],
      "ClusterSecurityGroups": [
         {
            "Status": "active",
            "ClusterSecurityGroupName": "default"
         } ],
```

```
      "AllowVersionUpgrade": true,
      "VpcSecurityGroups": \[],
      "AvailabilityZone": "us-east-1a",
      "ClusterCreateTime": "2013-01-22T21:59:29.559Z",
      "PreferredMaintenanceWindow": "sat:03:30-sat:04:00",
      "AutomatedSnapshotRetentionPeriod": 1,
      "ClusterStatus": "available",
      "ClusterIdentifier": "mycluster",
      "DBName": "dev",
      "NumberOfNodes": 2,
      "PendingModifiedValues": {}
   } ],
   "ResponseMetadata": {
      "RequestId": "65b71cac-64df-11e2-8f5b-e90bd6c77476"
   }
}
```

您也可以使用 --output text 选项以文本格式获取相同的信息。命令：

--output text 选项。命令：

选项。命令：

```
aws redshift describe-clusters --output text
```

结果：

```
dw.hs1.xlarge       1.0     true    adminuser       True    us-east-1a
 2013-01-22T21:59:29.559Z        sat:03:30-sat:04:00     1       available
 mycluster       dev     2
ENDPOINT    5439    mycluster.coqoarplqhsn.us-east-1.redshift.amazonaws.com
in-sync     default.redshift-1.0
active      default
PENDINGMODIFIEDVALUES
RESPONSEMETADATA    934281a8-64df-11e2-b07c-f7fbdd006c67
```

- 有关 API 详细信息，请参阅《AWS CLI 命令参考》中的 DescribeClusters。

Go

适用于 Go 的 SDK V2

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
import (
 "context"
 "errors"
 "log"
 "time"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/redshift"
 "github.com/aws/aws-sdk-go-v2/service/redshift/types"
)



// RedshiftActions wraps Redshift service actions.
type RedshiftActions struct {
 RedshiftClient *redshift.Client
}



// DescribeClusters returns information about the given cluster.
func (actor RedshiftActions) DescribeClusters(ctx context.Context, clusterId
 string) (*redshift.DescribeClustersOutput, error) {
 input, err := actor.RedshiftClient.DescribeClusters(ctx,
 &redshift.DescribeClustersInput{
  ClusterIdentifier: aws.String(clusterId),
 })
 var opErr *types.AccessToClusterDeniedFault
 if errors.As(err, &opErr) {
  println("Access to cluster denied.")
  panic(err)
```

```
 } else if err != nil {
  println("Failed to describe Redshift clusters.")
  return nil, err
 }
 return input, nil
}
```

- 有关 API 详细信息，请参阅《适用于 Go 的 AWS SDK API Reference》中的 DescribeClusters。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

描述集群。

```
    /**
     * Waits asynchronously for the specified cluster to become available.
     * @param clusterId the identifier of the cluster to wait for
     * @return a {@link CompletableFuture} that completes when the cluster is
ready
     */
    public CompletableFuture<Void> waitForClusterReadyAsync(String clusterId) {
        DescribeClustersRequest clustersRequest =
DescribeClustersRequest.builder()
            .clusterIdentifier(clusterId)
            .build();

        logger.info("Waiting for cluster to become available. This may take a few
minutes.");
        long startTime = System.currentTimeMillis();

        // Recursive method to poll the cluster status.
        return checkClusterStatusAsync(clustersRequest, startTime);
```

```
    }

    private CompletableFuture<Void>
checkClusterStatusAsync(DescribeClustersRequest clustersRequest, long startTime)
{
        return getAsyncClient().describeClusters(clustersRequest)
            .thenCompose(clusterResponse -> {
                List<Cluster> clusterList = clusterResponse.clusters();
                boolean clusterReady = false;
                for (Cluster cluster : clusterList) {
                    if ("available".equals(cluster.clusterStatus())) {
                        clusterReady = true;
                        break;
                    }
                }

                if (clusterReady) {
                    logger.info(String.format("Cluster is available!"));
                    return CompletableFuture.completedFuture(null);
                } else {
                    long elapsedTimeMillis = System.currentTimeMillis() -
startTime;
                    long elapsedSeconds = elapsedTimeMillis / 1000;
                    long minutes = elapsedSeconds / 60;
                    long seconds = elapsedSeconds % 60;
                    System.out.printf("\rElapsed Time: %02d:%02d - Waiting for
cluster...", minutes, seconds);
                    System.out.flush();

                    // Wait 1 second before the next status check
                    return CompletableFuture.runAsync(() -> {
                        try {
                            TimeUnit.SECONDS.sleep(1);
                        } catch (InterruptedException e) {
                            throw new RuntimeException("Error during sleep: " +
e.getMessage(), e);
                        }
                    }).thenCompose(ignored ->
checkClusterStatusAsync(clustersRequest, startTime));
                }
            }).exceptionally(exception -> {
                throw new RuntimeException("Failed to get cluster status: " +
exception.getMessage(), exception);
            });
```

```
    }
```

- 有关 API 详细信息，请参阅《AWS SDK for Java 2.x API Reference》中的 DescribeClusters。

JavaScript

SDK for JavaScript (v3)

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

创建客户端。

```
import { RedshiftClient } from "@aws-sdk/client-redshift";
// Set the AWS Region.
const REGION = "REGION";
//Set the Redshift Service Object
const redshiftClient = new RedshiftClient({ region: REGION });
export { redshiftClient };
```

描述集群。

```
// Import required AWS SDK clients and commands for Node.js
import { DescribeClustersCommand } from "@aws-sdk/client-redshift";
import { redshiftClient } from "./libs/redshiftClient.js";

const params = {
  ClusterIdentifier: "CLUSTER_NAME",
};

const run = async () => {
  try {
    const data = await redshiftClient.send(new DescribeClustersCommand(params));
    console.log("Success", data);
```

```
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- 有关 API 详细信息，请参阅《适用于 JavaScript 的 AWS SDK API Reference》中的 DescribeClusters。

Kotlin

SDK for Kotlin

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

描述集群。

```
suspend fun describeRedshiftClusters() {
    RedshiftClient.fromEnvironment { region = "us-west-2" }.use { redshiftClient
 ->
        val clusterResponse =
 redshiftClient.describeClusters(DescribeClustersRequest {})
        val clusterList = clusterResponse.clusters

        if (clusterList != null) {
            for (cluster in clusterList) {
                println("Cluster database name is ${cluster.dbName}")
                println("Cluster status is ${cluster.clusterStatus}")
            }
        }
    }
}
```

- 有关 API 详细信息，请参阅《AWS SDK for Kotlin API Reference》中的 DescribeClusters。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class RedshiftWrapper:
    """
    Encapsulates Amazon Redshift cluster operations.
    """

    def __init__(self, redshift_client):
        """
        :param redshift_client: A Boto3 Redshift client.
        """
        self.client = redshift_client


    def describe_clusters(self, cluster_identifier):
        """
        Describes a cluster.

        :param cluster_identifier: The cluster identifier.
        :return: A list of clusters.
        """
        try:
            kwargs = {}
            if cluster_identifier:
                kwargs["ClusterIdentifier"] = cluster_identifier

            paginator = self.client.get_paginator("describe_clusters")
            clusters = []
            for page in paginator.paginate(**kwargs):
                clusters.extend(page["Clusters"])

            return clusters

        except ClientError as err:
```

```
            logging.error(
                "Couldn't describe a cluster. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

以下代码会实例化 RedshiftWrapper 对象。

```
    client = boto3.client("redshift")
    redhift_wrapper = RedshiftWrapper(client)
```

- 有关 API 的详细信息，请参阅《AWS SDK for Python（Boto3）API 参考》中的 DescribeClusters。

SAP ABAP

适用于 SAP ABAP 的 SDK

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

描述集群。

```
    TRY.
        " Example values: iv_cluster_identifier = 'my-redshift-
cluster' (optional)
        oo_result = lo_rsh->describeclusters(
          iv_clusteridentifier = iv_cluster_identifier
        ).
        lt_clusters = oo_result->get_clusters( ).
        lv_cluster_count = lines( lt_clusters ).
        MESSAGE |Retrieved { lv_cluster_count } cluster(s).| TYPE 'I'.
      CATCH /aws1/cx_rshclustnotfoundfault.
        MESSAGE 'Cluster not found.' TYPE 'I'.
```

```
    ENDTRY.
```

- 有关 API 详细信息，请参阅《AWS SDK for SAP ABAP API Reference》中的 DescribeClusters。

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

## 将 `DescribeStatement` 和 AWS SDK 搭配使用

以下代码示例演示如何使用 `DescribeStatement`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

.NET

适用于 .NET 的 SDK (v4)

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /// <summary>
    /// Describe a statement execution.
    /// </summary>
    /// <param name="statementId">The statement ID.</param>
    /// <returns>The statement description.</returns>
    public async Task<DescribeStatementResponse> DescribeStatementAsync(string
statementId)
    {
        try
        {
            var request = new DescribeStatementRequest
            {
```

```
                Id = statementId
            };

            var response = await
_redshiftDataClient.DescribeStatementAsync(request);
            return response;
        }
        catch (Amazon.RedshiftDataAPIService.Model.ResourceNotFoundException ex)
        {
            Console.WriteLine($"Statement not found: {ex.Message}");
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't describe statement. Here's why:
{ex.Message}");
            throw;
        }
    }
```

- 有关 API 详细信息，请参阅《适用于 .NET 的 AWS SDK API Reference》中的 DescribeStatement。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * Checks the status of an SQL statement asynchronously and handles the
completion of the statement.
     *
     * @param sqlId the ID of the SQL statement to check
     * @return a {@link CompletableFuture} that completes when the SQL
statement's status is either "FINISHED" or "FAILED"
```

```
     */
    public CompletableFuture<Void> checkStatementAsync(String sqlId) {
        DescribeStatementRequest statementRequest =
DescribeStatementRequest.builder()
            .id(sqlId)
            .build();

        return getAsyncDataClient().describeStatement(statementRequest)
            .thenCompose(response -> {
                String status = response.statusAsString();
                logger.info("... Status: {} ", status);

                if ("FAILED".equals(status)) {
                    throw new RuntimeException("The Query Failed. Ending
program");
                } else if ("FINISHED".equals(status)) {
                    return CompletableFuture.completedFuture(null);
                } else {
                    // Sleep for 1 second and recheck status
                    return CompletableFuture.runAsync(() -> {
                        try {
                            TimeUnit.SECONDS.sleep(1);
                        } catch (InterruptedException e) {
                            throw new RuntimeException("Error during sleep: " +
e.getMessage(), e);
                        }
                    }).thenCompose(ignore -> checkStatementAsync(sqlId)); //
Recursively call until status is FINISHED or FAILED
                }
            }).whenComplete((result, exception) -> {
                if (exception != null) {
                    // Handle exceptions
                    logger.info("Error: {} ", exception.getMessage());
                } else {
                    logger.info("The statement is finished!");
                }
            });
    }
```

- 有关 API 详细信息，请参阅《AWS SDK for Java 2.x API Reference》中的 DescribeStatement。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class RedshiftDataWrapper:
    """Encapsulates Amazon Redshift data."""

    def __init__(self, client):
        """
        :param client: A Boto3 RedshiftDataWrapper client.
        """
        self.client = client


    def describe_statement(self, statement_id):
        """
        Describes a SQL statement.

        :param statement_id: The SQL statement identifier.
        :return: The SQL statement result.
        """
        try:
            response = self.client.describe_statement(Id=statement_id)
            return response
        except ClientError as err:
            logging.error(
                "Couldn't describe statement. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

以下代码会实例化 RedshiftDataWrapper 对象。

```
    client = boto3.client("redshift-data")
    redshift_data_wrapper = RedshiftDataWrapper(client)
```

- 有关 API 详细信息，请参阅《AWS SDK for Python（Boto3）API Reference》中的 DescribeStatement。

SAP ABAP

适用于 SAP ABAP 的 SDK

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    TRY.
        " Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx'
        oo_result = lo_rsd->describestatement(
          iv_id = iv_statement_id
        ).
        lv_status = oo_result->get_status( ).
        MESSAGE |Statement status: { lv_status }| TYPE 'I'.
      CATCH /aws1/cx_rsdresourcenotfoundex.
        MESSAGE 'Statement not found.' TYPE 'I'.
      CATCH /aws1/cx_rsdinternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
    ENDTRY.
```

- 有关 API 详细信息，请参阅《AWS SDK for SAP ABAP API Reference》中的 DescribeStatement。

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

## 将 **ExecuteStatement** 和 AWS SDK 搭配使用

以下代码示例演示如何使用 ExecuteStatement。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

执行 SQL 语句来创建数据库表。

```
   /**
    * Creates an asynchronous task to execute a SQL statement for creating a new
table.
    *
    * @param clusterId    the identifier of the Amazon Redshift cluster
    * @param databaseName the name of the database to create the table in
    * @param userName     the username to use for the database connection
    * @return a {@link CompletableFuture} that completes with the result of the
SQL statement execution
    * @throws RuntimeException if there is an error creating the table
    */
   public CompletableFuture<ExecuteStatementResponse> createTableAsync(String
clusterId, String databaseName, String userName) {
       ExecuteStatementRequest createTableRequest =
ExecuteStatementRequest.builder()
           .clusterIdentifier(clusterId)
           .dbUser(userName)
           .database(databaseName)
           .sql("CREATE TABLE Movies (" +
               "id INT PRIMARY KEY, " +
               "title VARCHAR(100), " +
```

```
                "year INT)")
            .build();

        return getAsyncDataClient().executeStatement(createTableRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    throw new RuntimeException("Error creating table: " +
exception.getMessage(), exception);
                } else {
                    logger.info("Table created: Movies");
                }
            });
    }
```

执行 SQL 语句将数据插入数据库表。

```
    /**
     * Asynchronously pops a table from a JSON file.
     *
     * @param clusterId   the ID of the cluster
     * @param databaseName the name of the database
     * @param userName    the username
     * @param fileName    the name of the JSON file
     * @param number      the number of records to process
     * @return a CompletableFuture that completes with the number of records
added to the Movies table
     */
    public CompletableFuture<Integer> popTableAsync(String clusterId, String
databaseName, String userName, String fileName, int number) {
        return CompletableFuture.supplyAsync(() -> {
                try {
                    JsonParser parser = new JsonFactory().createParser(new
File(fileName));
                    JsonNode rootNode = new ObjectMapper().readTree(parser);
                    Iterator<JsonNode> iter = rootNode.iterator();
                    return iter;
                } catch (IOException e) {
                    throw new RuntimeException("Failed to read or parse JSON
file: " + e.getMessage(), e);
                }
            }).thenCompose(iter -> processNodesAsync(clusterId, databaseName,
userName, iter, number))
```

```
            .whenComplete((result, exception) -> {
                if (exception != null) {
                    logger.info("Error {} ", exception.getMessage());
                } else {
                    logger.info("{} records were added to the Movies table." ,
result);
                }
            });
    }

    private CompletableFuture<Integer> processNodesAsync(String clusterId, String
databaseName, String userName, Iterator<JsonNode> iter, int number) {
        return CompletableFuture.supplyAsync(() -> {
            int t = 0;
            try {
                while (iter.hasNext()) {
                    if (t == number)
                        break;
                    JsonNode currentNode = iter.next();
                    int year = currentNode.get("year").asInt();
                    String title = currentNode.get("title").asText();

                    // Use SqlParameter to avoid SQL injection.
                    List<SqlParameter> parameterList = new ArrayList<>();
                    String sqlStatement = "INSERT INTO Movies
VALUES( :id , :title, :year);";
                    SqlParameter idParam = SqlParameter.builder()
                        .name("id")
                        .value(String.valueOf(t))
                        .build();

                    SqlParameter titleParam = SqlParameter.builder()
                        .name("title")
                        .value(title)
                        .build();

                    SqlParameter yearParam = SqlParameter.builder()
                        .name("year")
                        .value(String.valueOf(year))
                        .build();
                    parameterList.add(idParam);
                    parameterList.add(titleParam);
                    parameterList.add(yearParam);
```

```
                    ExecuteStatementRequest insertStatementRequest =
ExecuteStatementRequest.builder()
                        .clusterIdentifier(clusterId)
                        .sql(sqlStatement)
                        .database(databaseName)
                        .dbUser(userName)
                        .parameters(parameterList)
                        .build();


getAsyncDataClient().executeStatement(insertStatementRequest);
                    logger.info("Inserted: " + title + " (" + year + ")");
                    t++;
                }
            } catch (RedshiftDataException e) {
                throw new RuntimeException("Error inserting data: " +
e.getMessage(), e);
            }
            return t;
        });
    }
```

执行 SQL 语句来查询数据库表。

```
    /**
     * Asynchronously queries movies by a given year from a Redshift database.
     *
     * @param database    the name of the database to query
     * @param dbUser      the user to connect to the database with
     * @param year        the year to filter the movies by
     * @param clusterId   the identifier of the Redshift cluster to connect to
     * @return a {@link CompletableFuture} containing the response ID of the
executed SQL statement
     */
    public CompletableFuture<String> queryMoviesByYearAsync(String database,
                                                            String dbUser,
                                                            int year,
                                                            String
clusterId) {

        String sqlStatement = "SELECT * FROM Movies WHERE year = :year";
        SqlParameter yearParam = SqlParameter.builder()
```

```
            .name("year")
            .value(String.valueOf(year))
            .build();

        ExecuteStatementRequest statementRequest =
ExecuteStatementRequest.builder()
            .clusterIdentifier(clusterId)
            .database(database)
            .dbUser(dbUser)
            .parameters(yearParam)
            .sql(sqlStatement)
            .build();

        return CompletableFuture.supplyAsync(() -> {
            try {
                ExecuteStatementResponse response =
getAsyncDataClient().executeStatement(statementRequest).join(); // Use join() to
wait for the result
                return response.id();
            } catch (RedshiftDataException e) {
                throw new RuntimeException("Error executing statement: " +
e.getMessage(), e);
            }
        }).exceptionally(exception -> {
            logger.info("Error: {}", exception.getMessage());
            return "";
        });
    }
```

- 有关 API 详细信息，请参阅《AWS SDK for Java 2.x API Reference》中的 ExecuteStatement。

SAP ABAP

适用于 SAP ABAP 的 SDK

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    TRY.
        " Example values: iv_cluster_identifier = 'redshift-cluster-movies'
        " Example values: iv_database_name = 'dev'
        " Example values: iv_user_name = 'awsuser'
        " Example values: iv_sql = 'SELECT * FROM movies WHERE year = :year'
        " Example values: it_parameter_list - SQL parameters for parameterized
queries

        " Only pass parameters if the list is not empty
        IF it_parameter_list IS NOT INITIAL.
          oo_result = lo_rsd->executestatement(
            iv_clusteridentifier = iv_cluster_identifier
            iv_database = iv_database_name
            iv_dbuser = iv_user_name
            iv_sql = iv_sql
            it_parameters = it_parameter_list
          ).
        ELSE.
          oo_result = lo_rsd->executestatement(
            iv_clusteridentifier = iv_cluster_identifier
            iv_database = iv_database_name
            iv_dbuser = iv_user_name
            iv_sql = iv_sql
          ).
        ENDIF.

        lv_statement_id = oo_result->get_id( ).
        MESSAGE |Statement executed. ID: { lv_statement_id }| TYPE 'I'.
      CATCH /aws1/cx_rsdexecutestatementex.
        MESSAGE 'Statement execution error.' TYPE 'I'.
      CATCH /aws1/cx_rsdresourcenotfoundex.
        MESSAGE 'Resource not found.' TYPE 'I'.
    ENDTRY.
```

- 有关 API 详细信息，请参阅《AWS SDK for SAP ABAP API Reference》中的 ExecuteStatement。

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

## 将 **GetStatementResult** 和 AWS SDK 搭配使用

以下代码示例演示如何使用 GetStatementResult。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

.NET

适用于 .NET 的 SDK (v4)

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /// <summary>
    /// Get the results of a statement execution.
    /// </summary>
    /// <param name="statementId">The statement ID.</param>
    /// <returns>A list of result rows.</returns>
    public async Task<List<List<Field>>> GetStatementResultAsync(string
statementId)
    {
        try
        {
            var request = new GetStatementResultRequest
            {
                Id = statementId
            };

            var response = await
_redshiftDataClient.GetStatementResultAsync(request);
            return response.Records;
        }
        catch (Amazon.RedshiftDataAPIService.Model.ResourceNotFoundException ex)
        {
            Console.WriteLine($"Statement not found: {ex.Message}");
            throw;
```

```
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't get statement result. Here's why:
{ex.Message}");
            throw;
        }
    }
```

- 有关 API 详细信息，请参阅《适用于 .NET 的 AWS SDK API Reference》中的 GetStatementResult。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

检查语句结果。

```
    /**
     * Asynchronously retrieves the results of a statement execution.
     *
     * @param statementId the ID of the statement for which to retrieve the
results
     * @return a {@link CompletableFuture} that completes when the statement
result has been processed
     */
    public CompletableFuture<Void> getResultsAsync(String statementId) {
        GetStatementResultRequest resultRequest =
GetStatementResultRequest.builder()
            .id(statementId)
            .build();

        return getAsyncDataClient().getStatementResult(resultRequest)
            .handle((response, exception) -> {
                if (exception != null) {
```

```
                    logger.info("Error getting statement result {} ",
exception.getMessage());
                    throw new RuntimeException("Error getting statement result: "
+ exception.getMessage(), exception);
                }

                // Extract and print the field values using streams if the
response is valid.
                response.records().stream()
                    .flatMap(List::stream)
                    .map(Field::stringValue)
                    .filter(value -> value != null)
                    .forEach(value -> System.out.println("The Movie title field
is " + value));

                return response;
            }).thenAccept(response -> {
                // Optionally add more logic here if needed after handling the
response
            });
    }
```

- 有关 API 详细信息，请参阅《AWS SDK for Java 2.x API Reference》中的 GetStatementResult。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class RedshiftDataWrapper:
    """Encapsulates Amazon Redshift data."""

    def __init__(self, client):
        """
```

```
        :param client: A Boto3 RedshiftDataWrapper client.
        """
        self.client = client


    def get_statement_result(self, statement_id):
        """
        Gets the result of a SQL statement.

        :param statement_id: The SQL statement identifier.
        :return: The SQL statement result.
        """
        try:
            result = {
                "Records": [],
            }
            paginator = self.client.get_paginator("get_statement_result")
            for page in paginator.paginate(Id=statement_id):
                if "ColumnMetadata" not in result:
                    result["ColumnMetadata"] = page["ColumnMetadata"]
                result["Records"].extend(page["Records"])
            return result
        except ClientError as err:
            logging.error(
                "Couldn't get statement result. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

以下代码会实例化 RedshiftDataWrapper 对象。

```
    client = boto3.client("redshift-data")
    redshift_data_wrapper = RedshiftDataWrapper(client)
```

- 有关 API 详细信息，请参阅《AWS SDK for Python（Boto3）API Reference》中的 GetStatementResult。

SAP ABAP

适用于 SAP ABAP 的 SDK

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

检查语句结果。

```
    TRY.
        " Example values: iv_statement_id = 'xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx'
        " Handle pagination for large result sets

        DO.
          lo_result_page = lo_rsd->getstatementresult(
            iv_id = iv_statement_id
            iv_nexttoken = lv_next_token
          ).

          " Collect records from this page
          lt_page_records = lo_result_page->get_records( ).
          APPEND LINES OF lt_page_records TO lt_all_records.

          " Check if there are more pages
          lv_next_token = lo_result_page->get_nexttoken( ).
          IF lv_next_token IS INITIAL.
            EXIT. " No more pages
          ENDIF.
        ENDDO.

        " For the last call, set oo_result for return value
        oo_result = lo_result_page.
        lv_record_count = lines( lt_all_records ).
        MESSAGE |Retrieved { lv_record_count } record(s).| TYPE 'I'.
      CATCH /aws1/cx_rsdresourcenotfoundex.
        MESSAGE 'Statement not found or results not available.' TYPE 'I'.
      CATCH /aws1/cx_rsdinternalserverex.
        MESSAGE 'Internal server error.' TYPE 'I'.
    ENDTRY.
```

- 有关 API 详细信息，请参阅《AWS SDK for SAP ABAP API Reference》中的 GetStatementResult。

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

## 将 **ListDatabases** 和 AWS SDK 搭配使用

以下代码示例演示如何使用 ListDatabases。

.NET

适用于 .NET 的 SDK (v4)

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /// <summary>
    /// List databases in a Redshift cluster.
    /// </summary>
    /// <param name="clusterIdentifier">The cluster identifier.</param>
    /// <param name="dbUser">The database user.</param>
    /// <param name="dbUser">The database name for authentication.</param>
    /// <returns>A list of database names.</returns>
    public async Task<List<string>> ListDatabasesAsync(string clusterIdentifier,
string dbUser, string databaseName)
    {
        try
        {
            var request = new ListDatabasesRequest
            {
                ClusterIdentifier = clusterIdentifier,
                DbUser = dbUser,
                Database = databaseName
            };
```

```
            var response = await _redshiftDataClient.ListDatabasesAsync(request);
            var databases = new List<string>();

            foreach (var database in response.Databases)
            {
                Console.WriteLine($"The database name is : {database}");
                databases.Add(database);
            }

            return databases;
        }
        catch (Amazon.RedshiftDataAPIService.Model.ValidationException ex)
        {
            Console.WriteLine($"Validation error: {ex.Message}");
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Couldn't list databases. Here's why:
{ex.Message}");
            throw;
        }
    }
```

- 有关 API 详细信息，请参阅《适用于 .NET 的 AWS SDK API Reference》中的 ListDatabases。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * Lists all databases asynchronously for the specified cluster, database
user, and database.
```

```
     * @param clusterId the identifier of the cluster to list databases for
     * @param dbUser the database user to use for the list databases request
     * @param database the database to list databases for
     * @return a {@link CompletableFuture} that completes when the database
listing is complete, or throws a {@link RuntimeException} if there was an error
     */
    public CompletableFuture<Void> listAllDatabasesAsync(String clusterId, String
dbUser, String database) {
        ListDatabasesRequest databasesRequest = ListDatabasesRequest.builder()
            .clusterIdentifier(clusterId)
            .dbUser(dbUser)
            .database(database)
            .build();

        // Asynchronous paginator for listing databases.
        ListDatabasesPublisher databasesPaginator =
getAsyncDataClient().listDatabasesPaginator(databasesRequest);
        CompletableFuture<Void> future = databasesPaginator.subscribe(response ->
{
            response.databases().forEach(db -> {
                logger.info("The database name is {} ", db);
            });
        });

        // Return the future for asynchronous handling.
        return future.exceptionally(exception -> {
            throw new RuntimeException("Failed to list databases: " +
exception.getMessage(), exception);
        });
    }
```

- 有关 API 详细信息，请参阅《AWS SDK for Java 2.x API Reference》中的 ListDatabases。

## SAP ABAP

### 适用于 SAP ABAP 的 SDK

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    TRY.
        " Example values: iv_cluster_identifier = 'redshift-cluster-movies'
        " Example values: iv_database_name = 'dev'
        " Example values: iv_database_user = 'awsuser'
        oo_result = lo_rsd->listdatabases(
          iv_clusteridentifier = iv_cluster_identifier
          iv_database = iv_database_name
          iv_dbuser = iv_database_user
        ).
        lt_databases = oo_result->get_databases( ).
        lv_db_count = lines( lt_databases ).
        MESSAGE |Retrieved { lv_db_count } database(s).| TYPE 'I'.
      CATCH /aws1/cx_rsddatabaseconnex.
        MESSAGE 'Database connection error.' TYPE 'I'.
      CATCH /aws1/cx_rsdresourcenotfoundex.
        MESSAGE 'Cluster not found.' TYPE 'I'.
    ENDTRY.
```

- 有关 API 详细信息，请参阅《AWS SDK for SAP ABAP API Reference》中的 ListDatabases。

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

## 将 **ModifyCluster** 与 AWS SDK 或 CLI 配合使用

以下代码示例演示如何使用 ModifyCluster。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

.NET

适用于 .NET 的 SDK (v4)

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /// <summary>
    /// Modify an Amazon Redshift cluster.
    /// </summary>
    /// <param name="clusterIdentifier">The identifier for the cluster.</param>
    /// <param name="preferredMaintenanceWindow">The preferred maintenance
window.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> ModifyClusterAsync(string clusterIdentifier, string
preferredMaintenanceWindow)
    {
        try
        {
            var request = new ModifyClusterRequest
            {
                ClusterIdentifier = clusterIdentifier,
                PreferredMaintenanceWindow = preferredMaintenanceWindow
            };

            var response = await _redshiftClient.ModifyClusterAsync(request);
            Console.WriteLine($"The modified cluster was successfully modified
and has {response.Cluster.PreferredMaintenanceWindow} as the maintenance
window");
            return true;
        }
        catch (ClusterNotFoundException ex)
        {
            Console.WriteLine($"Cluster {clusterIdentifier} not found:
{ex.Message}");
            return false;
        }
        catch (Exception ex)
        {
```

```
            Console.WriteLine($"Couldn't modify cluster. Here's why:
{ex.Message}");
            return false;
        }
    }
```

- 有关 API 详细信息，请参阅《适用于 .NET 的 AWS SDK API Reference》中的 ModifyCluster。

CLI

AWS CLI

将安全组与集群关联 此示例演示了如何将集群安全组与指定的集群相关联。命令：

```
aws redshift modify-cluster --cluster-identifier mycluster --cluster-security-
groups mysecuritygroup
```

修改集群的维护时段 此示例演示了如何将集群的每周首选维护时段更改为最少四小时的时段，从周日晚上 11:15 开始，到周一凌晨 3:15 结束。命令：

```
aws redshift modify-cluster --cluster-identifier mycluster --preferred-
maintenance-window Sun:23:15-Mon:03:15
```

更改集群的主密码 此示例演示了如何更改集群的主密码。命令：

```
aws redshift modify-cluster --cluster-identifier mycluster --master-user-password
 A1b2c3d4
```

- 有关 API 详细信息，请参阅《AWS CLI 命令参考》中的 ModifyCluster。

Go

适用于 Go 的 SDK V2

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
import (
 "context"
 "errors"
 "log"
 "time"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/redshift"
 "github.com/aws/aws-sdk-go-v2/service/redshift/types"
)



// RedshiftActions wraps Redshift service actions.
type RedshiftActions struct {
 RedshiftClient *redshift.Client
}



// ModifyCluster sets the preferred maintenance window for the given cluster.
func (actor RedshiftActions) ModifyCluster(ctx context.Context, clusterId string,
 maintenanceWindow string) *redshift.ModifyClusterOutput {
 // Modify the cluster's maintenance window
 input := &redshift.ModifyClusterInput{
  ClusterIdentifier:          aws.String(clusterId),
  PreferredMaintenanceWindow: aws.String(maintenanceWindow),
 }

 var opErr *types.InvalidClusterStateFault
 output, err := actor.RedshiftClient.ModifyCluster(ctx, input)
```

```
 if err != nil && errors.As(err, &opErr) {
  log.Println("Cluster is in an invalid state.")
  panic(err)
 } else if err != nil {
  log.Printf("Failed to modify Redshift cluster: %v\n", err)
  panic(err)
 }

 log.Printf("The cluster was successfully modified and now has %s as the
 maintenance window\n", *output.Cluster.PreferredMaintenanceWindow)
 return output
}
```

- 有关 API 详细信息，请参阅《适用于 Go 的 AWS SDK API Reference》中的 ModifyCluster。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

修改集群。

```
    /**
     * Modifies an Amazon Redshift cluster asynchronously.
     *
     * @param clusterId the identifier of the cluster to be modified
     * @return a {@link CompletableFuture} that completes when the cluster
modification is complete
     */
    public CompletableFuture<ModifyClusterResponse> modifyClusterAsync(String
clusterId) {
        ModifyClusterRequest modifyClusterRequest =
ModifyClusterRequest.builder()
```

```
            .clusterIdentifier(clusterId)
            .preferredMaintenanceWindow("wed:07:30-wed:08:00")
            .build();

        return getAsyncClient().modifyCluster(modifyClusterRequest)
            .whenComplete((clusterResponse, exception) -> {
                if (exception != null) {
                    if (exception.getCause() instanceof RedshiftException) {
                        logger.info("Error: {} ", exception.getMessage());
                    } else {
                        logger.info("Unexpected error: {} ",
exception.getMessage());
                    }
                } else {
                    logger.info("The modified cluster was successfully modified
and has "
                        + clusterResponse.cluster().preferredMaintenanceWindow()
+ " as the maintenance window");
                }
            });
    }
```

- 有关 API 详细信息，请参阅《AWS SDK for Java 2.x API Reference》中的 ModifyCluster。

JavaScript

SDK for JavaScript (v3)

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

创建客户端。

```
import { RedshiftClient } from "@aws-sdk/client-redshift";
// Set the AWS Region.
const REGION = "REGION";
//Set the Redshift Service Object
const redshiftClient = new RedshiftClient({ region: REGION });
```

```
export { redshiftClient };
```

修改集群。

```
// Import required AWS SDK clients and commands for Node.js
import { ModifyClusterCommand } from "@aws-sdk/client-redshift";
import { redshiftClient } from "./libs/redshiftClient.js";

// Set the parameters
const params = {
  ClusterIdentifier: "CLUSTER_NAME",
  MasterUserPassword: "NEW_MASTER_USER_PASSWORD",
};

const run = async () => {
  try {
    const data = await redshiftClient.send(new ModifyClusterCommand(params));
    console.log("Success was modified.", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

- 有关 API 详细信息，请参阅《适用于 JavaScript 的 AWS SDK API Reference》中的 ModifyCluster。

Kotlin

SDK for Kotlin

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

修改集群。

```
suspend fun modifyCluster(clusterId: String?) {
    val modifyClusterRequest =
        ModifyClusterRequest {
            clusterIdentifier = clusterId
            preferredMaintenanceWindow = "wed:07:30-wed:08:00"
        }

    RedshiftClient { region = "us-west-2" }.use { redshiftClient ->
        val clusterResponse = redshiftClient.modifyCluster(modifyClusterRequest)
        println(
            "The modified cluster was successfully modified and has
 ${clusterResponse.cluster?.preferredMaintenanceWindow} as the maintenance
 window",
        )
    }
}
```

- 有关 API 详细信息，请参阅《AWS SDK for Kotlin API Reference》中的 ModifyCluster。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class RedshiftWrapper:
    """
    Encapsulates Amazon Redshift cluster operations.
    """

    def __init__(self, redshift_client):
        """
        :param redshift_client: A Boto3 Redshift client.
        """
        self.client = redshift_client
```

```
    def modify_cluster(self, cluster_identifier, preferred_maintenance_window):
        """
        Modifies a cluster.

        :param cluster_identifier: The cluster identifier.
        :param preferred_maintenance_window: The preferred maintenance window.
        """
        try:
            self.client.modify_cluster(
                ClusterIdentifier=cluster_identifier,
                PreferredMaintenanceWindow=preferred_maintenance_window,
            )
        except ClientError as err:
            logging.error(
                "Couldn't modify a cluster. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

以下代码会实例化 RedshiftWrapper 对象。

```
client = boto3.client("redshift")
redhift_wrapper = RedshiftWrapper(client)
```

- 有关 API 详细信息，请参阅《AWS SDK for Python（Boto3）API Reference》中的 ModifyCluster。

## SAP ABAP

### 适用于 SAP ABAP 的 SDK

> **Note**
>
> 查看 GitHub，了解更多信息。在 AWS 代码示例存储库中查找完整示例，了解如何进行设置和运行。

修改集群。

```
TRY.
    " Example values: iv_cluster_identifier = 'my-redshift-cluster'
    " Example values: iv_pref_maintenance_wn = 'wed:07:30-wed:08:00'
    lo_rsh->modifycluster(
      iv_clusteridentifier = iv_cluster_identifier
      iv_preferredmaintenancewin00 = iv_pref_maintenance_wn
    ).
    MESSAGE 'Redshift cluster modified successfully.' TYPE 'I'.
  CATCH /aws1/cx_rshclustnotfoundfault.
    MESSAGE 'Cluster not found.' TYPE 'I'.
  CATCH /aws1/cx_rshinvcluststatefault.
    MESSAGE 'Invalid cluster state for modification.' TYPE 'I'.
ENDTRY.
```

- 有关 API 详细信息，请参阅《AWS SDK for SAP ABAP API Reference》中的 ModifyCluster。

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

# 使用 AWS SDK 的 Amazon Redshift 场景

以下代码示例展示了如何通过 AWS SDK 实施 Amazon Redshift 中的常见场景。这些场景向您展示了如何通过调用 Amazon Redshift 中的多个函数或与其它 AWS 服务结合来完成特定任务。每个场景都包含完整源代码的链接，您可以在其中找到有关如何设置和运行代码的说明。

场景以中等水平的经验为目标，可帮助您结合具体环境了解服务操作。

示例

- 创建 Amazon Redshift 项目追踪器
- 通过 CLI 开始使用无服务器数据仓库
- 预调配数据仓库集群入门

## 创建 Amazon Redshift 项目追踪器

以下代码示例展示如何使用 Amazon Redshift 数据库创建用于跟踪和报告工作项的 Web 应用程序。

Java

适用于 Java 的 SDK 2.x

展示如何创建 Web 应用程序来跟踪与报告存储与 Amazon Redshift 数据库的工作项。

有关完整源代码以及如何设置查询 Amazon Redshift 数据以及供 React 应用程序使用的 Spring REST API 的说明，请参阅 GitHub 上的完整示例。

本示例中使用的服务

- Amazon Redshift
- Amazon SES

Kotlin

SDK for Kotlin

展示如何创建 Web 应用程序来跟踪与报告存储与 Amazon Redshift 数据库的工作项。

有关完整源代码以及如何设置查询 Amazon Redshift 数据以及供 React 应用程序使用的 Spring REST API 的说明，请参阅 GitHub 上的完整示例。

本示例中使用的服务

- Amazon Redshift
- Amazon SES

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

## 通过 CLI 开始使用无服务器数据仓库

以下代码示例展示了如何：

- 使用 secrets-manager CreateSecret
- 使用 secrets-manager DeleteSecret
- 使用 secrets-manager GetSecretValue
- 使用 redshift CreateNamespace
- 使用 redshift CreateWorkgroup

- 使用 redshift DeleteNamespace
- 使用 iam CreateRole

Bash

AWS CLI 及 Bash 脚本

> **Note**
>
> 查看 GitHub，了解更多信息。在 Sample developer tutorials 存储库中查找完整示例，了解如何进行设置和运行。

```
#!/bin/bash

# Amazon Redshift Serverless Tutorial Script with Secrets Manager (No jq
 dependency)
# This script creates a Redshift Serverless environment, loads sample data, and
 runs queries
# Uses AWS Secrets Manager for secure password management without requiring jq

# Set up logging
LOG_FILE="redshift-serverless-tutorial-v4.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon Redshift Serverless tutorial script at $(date)"
echo "All commands and outputs will be logged to $LOG_FILE"

# Function to check for errors in command output
check_error() {
  local output=$1
  local cmd=$2

  if echo "$output" | grep -i "error\|exception\|fail" > /dev/null; then
    echo "ERROR: Command failed: $cmd"
    echo "Output: $output"
    cleanup_resources
    exit 1
  fi
}
```

```
# Function to generate a secure password that meets Redshift requirements
generate_secure_password() {
  # Redshift password requirements:
  # - 8-64 characters
  # - At least one uppercase letter
  # - At least one lowercase letter
  # - At least one decimal digit
  # - Can contain printable ASCII characters except /, ", ', \, @, space

  local password=""
  local valid=false
  local attempts=0
  local max_attempts=10

  while [[ "$valid" == false && $attempts -lt $max_attempts ]]; do
    # Generate base password with safe characters
    local base=$(openssl rand -base64 12 | tr -d '/+=' | head -c 12)

    # Ensure we have at least one of each required character type
    local upper=$(echo "ABCDEFGHIJKLMNOPQRSTUVWXYZ" | fold -w1 | shuf -n1)
    local lower=$(echo "abcdefghijklmnopqrstuvwxyz" | fold -w1 | shuf -n1)
    local digit=$(echo "0123456789" | fold -w1 | shuf -n1)
    local special=$(echo "!#$%&*()_+-=[]{}|;:,.<>?" | fold -w1 | shuf -n1)

    # Combine and shuffle
    password="${base}${upper}${lower}${digit}${special}"
    password=$(echo "$password" | fold -w1 | shuf | tr -d '\n')

    # Validate password meets requirements
    if [[ ${#password} -ge 8 && ${#password} -le 64 ]] && \
       [[ "$password" =~ [A-Z] ]] && \
       [[ "$password" =~ [a-z] ]] && \
       [[ "$password" =~ [0-9] ]] && \
       [[ ! "$password" =~ [/\"\'\\@[:space:]] ]]; then
      valid=true
    fi

    ((attempts++))
  done

  if [[ "$valid" == false ]]; then
    echo "ERROR: Failed to generate valid password after $max_attempts attempts"
    exit 1
  fi
```

```
  echo "$password"
}

# Function to create secret in AWS Secrets Manager
create_secret() {
  local secret_name=$1
  local username=$2
  local password=$3
  local description=$4

  echo "Creating secret in AWS Secrets Manager: $secret_name"

  # Create the secret using AWS CLI without jq
  local secret_output=$(aws secretsmanager create-secret \
    --name "$secret_name" \
    --description "$description" \
    --secret-string "{\"username\":\"$username\",\"password\":\"$password\"}" \
    --tags Key=project,Value=doc-smith Key=tutorial,Value=redshift-serverless
 2>&1)

  if echo "$secret_output" | grep -i "error\|exception\|fail" > /dev/null; then
    echo "ERROR: Failed to create secret: $secret_output"
    return 1
  fi

  echo "Secret created successfully: $secret_name"
  return 0
}

# Function to retrieve password from AWS Secrets Manager
get_password_from_secret() {
  local secret_name=$1

  # Get the secret value and extract password using sed/grep instead of jq
  local secret_value=$(aws secretsmanager get-secret-value \
    --secret-id "$secret_name" \
    --query 'SecretString' \
    --output text 2>/dev/null)

  if [[ $? -eq 0 ]]; then
    # Extract password from JSON using sed
    echo "$secret_value" | sed -n 's/.*"password":"\([^"]*\)".*/\1/p'
  else
```

```
    echo ""
  fi
}

# Function to wait for a resource to be available
wait_for_resource() {
  local resource_type=$1
  local resource_name=$2
  local max_attempts=$3
  local wait_seconds=$4
  local check_cmd=$5

  echo "Waiting for $resource_type $resource_name to be available..."

  for ((i=1; i<=$max_attempts; i++)); do
    local output=$($check_cmd 2>/dev/null)
    local status=$(echo "$output" | grep -o '"Status": "[^"]*' | cut -d'"' -f4 ||
 echo "")

    if [[ "$status" == "AVAILABLE" ]]; then
      echo "$resource_type $resource_name is now available"
      return 0
    fi

    echo "Attempt $i/$max_attempts: $resource_type $resource_name status:
 $status. Waiting $wait_seconds seconds..."
    sleep $wait_seconds
  done

  echo "ERROR: Timed out waiting for $resource_type $resource_name to be
 available"
  return 1
}

# Function to wait for a resource to be deleted
wait_for_resource_deletion() {
  local resource_type=$1
  local resource_name=$2
  local max_attempts=$3
  local wait_seconds=$4
  local check_cmd=$5

  echo "Waiting for $resource_type $resource_name to be deleted..."
```

```
  for ((i=1; i<=$max_attempts; i++)); do
    local output=$($check_cmd 2>&1)

    if echo "$output" | grep -i "not found\|does not exist" > /dev/null; then
      echo "$resource_type $resource_name has been deleted"
      return 0
    fi

    echo "Attempt $i/$max_attempts: $resource_type $resource_name is still being
 deleted. Waiting $wait_seconds seconds..."
    sleep $wait_seconds
  done

  echo "ERROR: Timed out waiting for $resource_type $resource_name to be deleted"
  return 1
}

# Function to clean up resources
cleanup_resources() {
  echo ""
  echo "========================================="
  echo "CLEANUP CONFIRMATION"
  echo "========================================="
  echo "The following resources were created:"
  echo "- Redshift Serverless Workgroup: $WORKGROUP_NAME"
  echo "- Redshift Serverless Namespace: $NAMESPACE_NAME"
  echo "- IAM Role: $ROLE_NAME"
  echo "- Secrets Manager Secret: $SECRET_NAME"
  echo ""
  echo "Do you want to clean up all created resources? (y/n): "
  read -r CLEANUP_CHOICE

  if [[ "${CLEANUP_CHOICE,,}" == "y" ]]; then
    echo "Cleaning up resources..."

    # Delete the workgroup
    echo "Deleting Redshift Serverless workgroup $WORKGROUP_NAME..."
    WORKGROUP_DELETE_OUTPUT=$(aws redshift-serverless delete-workgroup --
workgroup-name "$WORKGROUP_NAME" 2>&1)
    echo "$WORKGROUP_DELETE_OUTPUT"

    # Wait for workgroup to be deleted before deleting namespace
    wait_for_resource_deletion "workgroup" "$WORKGROUP_NAME" 20 30 "aws redshift-
serverless get-workgroup --workgroup-name $WORKGROUP_NAME"
```

```
    # Delete the namespace
    echo "Deleting Redshift Serverless namespace $NAMESPACE_NAME..."
    NAMESPACE_DELETE_OUTPUT=$(aws redshift-serverless delete-namespace --
namespace-name "$NAMESPACE_NAME" 2>&1)
    echo "$NAMESPACE_DELETE_OUTPUT"

    # Wait for namespace to be deleted
    wait_for_resource_deletion "namespace" "$NAMESPACE_NAME" 20 30 "aws redshift-
serverless get-namespace --namespace-name $NAMESPACE_NAME"

    # Delete the IAM role policy
    echo "Deleting IAM role policy..."
    POLICY_DELETE_OUTPUT=$(aws iam delete-role-policy --role-name "$ROLE_NAME" --
policy-name S3Access 2>&1)
    echo "$POLICY_DELETE_OUTPUT"

    # Delete the IAM role
    echo "Deleting IAM role $ROLE_NAME..."
    ROLE_DELETE_OUTPUT=$(aws iam delete-role --role-name "$ROLE_NAME" 2>&1)
    echo "$ROLE_DELETE_OUTPUT"

    # Delete the secret
    echo "Deleting Secrets Manager secret $SECRET_NAME..."
    SECRET_DELETE_OUTPUT=$(aws secretsmanager delete-secret --secret-id
 "$SECRET_NAME" --force-delete-without-recovery 2>&1)
    echo "$SECRET_DELETE_OUTPUT"

    echo "Cleanup completed."
  else
    echo "Cleanup skipped. Resources will remain in your AWS account."
  fi
}

# Check if required tools are available
if ! command -v openssl &> /dev/null; then
    echo "ERROR: openssl is required but not installed. Please install openssl to
 continue."
    exit 1
fi

# Generate unique names for resources
RANDOM_SUFFIX=$(cat /dev/urandom | tr -dc 'a-z0-9' | head -c 6)
NAMESPACE_NAME="rs-namespace-${RANDOM_SUFFIX}"
```

```
WORKGROUP_NAME="rs-workgroup-${RANDOM_SUFFIX}"
ROLE_NAME="RedshiftServerlessS3Role-${RANDOM_SUFFIX}"
SECRET_NAME="redshift-serverless-admin-${RANDOM_SUFFIX}"
DB_NAME="dev"
ADMIN_USERNAME="admin"

# Generate secure password
echo "Generating secure password..."
ADMIN_PASSWORD=$(generate_secure_password)

# Create secret in AWS Secrets Manager
create_secret "$SECRET_NAME" "$ADMIN_USERNAME" "$ADMIN_PASSWORD" "Admin
 credentials for Redshift Serverless namespace $NAMESPACE_NAME"
if [[ $? -ne 0 ]]; then
  echo "ERROR: Failed to create secret in AWS Secrets Manager"
  exit 1
fi

# Track created resources
CREATED_RESOURCES=()

echo "Using the following resource names:"
echo "- Namespace: $NAMESPACE_NAME"
echo "- Workgroup: $WORKGROUP_NAME"
echo "- IAM Role: $ROLE_NAME"
echo "- Secret: $SECRET_NAME"
echo "- Database: $DB_NAME"
echo "- Admin Username: $ADMIN_USERNAME"
echo "- Admin Password: [STORED IN SECRETS MANAGER]"

# Step 1: Create IAM role for S3 access
echo "Creating IAM role for Redshift Serverless S3 access..."

# Create trust policy document
cat > redshift-trust-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift-serverless.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
```

```
    }
  ]
}
EOF

# Create S3 access policy document
cat > redshift-s3-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::redshift-downloads",
        "arn:aws:s3:::redshift-downloads/*"
      ]
    }
  ]
}
EOF

# Create IAM role
echo "Creating IAM role $ROLE_NAME..."
ROLE_OUTPUT=$(aws iam create-role --role-name "$ROLE_NAME" --assume-role-policy-
document file://redshift-trust-policy.json 2>&1)
echo "$ROLE_OUTPUT"
check_error "$ROLE_OUTPUT" "aws iam create-role"
aws iam tag-role --role-name "$ROLE_NAME" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=redshift-serverless
CREATED_RESOURCES+=("IAM Role: $ROLE_NAME")

# Attach S3 policy to the role
echo "Attaching S3 access policy to role $ROLE_NAME..."
POLICY_OUTPUT=$(aws iam put-role-policy --role-name "$ROLE_NAME" --policy-name
 S3Access --policy-document file://redshift-s3-policy.json 2>&1)
echo "$POLICY_OUTPUT"
check_error "$POLICY_OUTPUT" "aws iam put-role-policy"

# Get the role ARN
```

```
ROLE_ARN=$(aws iam get-role --role-name "$ROLE_NAME" --query 'Role.Arn' --output
 text)
echo "Role ARN: $ROLE_ARN"

# Step 2: Create a namespace
echo "Creating Redshift Serverless namespace $NAMESPACE_NAME..."
NAMESPACE_OUTPUT=$(aws redshift-serverless create-namespace \
  --namespace-name "$NAMESPACE_NAME" \
  --admin-username "$ADMIN_USERNAME" \
  --admin-user-password "$ADMIN_PASSWORD" \
  --db-name "$DB_NAME" \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=redshift-serverless 2>&1)
echo "$NAMESPACE_OUTPUT"
check_error "$NAMESPACE_OUTPUT" "aws redshift-serverless create-namespace"
CREATED_RESOURCES+=("Redshift Serverless Namespace: $NAMESPACE_NAME")

# Wait for namespace to be available
wait_for_resource "namespace" "$NAMESPACE_NAME" 10 30 "aws redshift-serverless
 get-namespace --namespace-name $NAMESPACE_NAME"

# Associate IAM role with namespace
echo "Associating IAM role with namespace..."
UPDATE_NAMESPACE_OUTPUT=$(aws redshift-serverless update-namespace \
  --namespace-name "$NAMESPACE_NAME" \
  --iam-roles "$ROLE_ARN" 2>&1)
echo "$UPDATE_NAMESPACE_OUTPUT"
check_error "$UPDATE_NAMESPACE_OUTPUT" "aws redshift-serverless update-namespace"

# Step 3: Create a workgroup
echo "Creating Redshift Serverless workgroup $WORKGROUP_NAME..."
WORKGROUP_OUTPUT=$(aws redshift-serverless create-workgroup \
  --workgroup-name "$WORKGROUP_NAME" \
  --namespace-name "$NAMESPACE_NAME" \
  --base-capacity 8 \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=redshift-serverless 2>&1)
echo "$WORKGROUP_OUTPUT"
check_error "$WORKGROUP_OUTPUT" "aws redshift-serverless create-workgroup"
CREATED_RESOURCES+=("Redshift Serverless Workgroup: $WORKGROUP_NAME")

# Wait for workgroup to be available
wait_for_resource "workgroup" "$WORKGROUP_NAME" 20 30 "aws redshift-serverless
 get-workgroup --workgroup-name $WORKGROUP_NAME"

# Get workgroup endpoint
```

```
WORKGROUP_ENDPOINT=$(aws redshift-serverless get-workgroup \
  --workgroup-name "$WORKGROUP_NAME" \
  --query 'workgroup.endpoint.address' \
  --output text)
echo "Workgroup endpoint: $WORKGROUP_ENDPOINT"

# Wait additional time for the endpoint to be fully operational
echo "Waiting for endpoint to be fully operational..."
sleep 60

# Step 4: Create tables for sample data
echo "Creating tables for sample data..."

# Create users table
echo "Creating users table..."
USERS_TABLE_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "CREATE TABLE users(
    userid INTEGER NOT NULL DISTKEY SORTKEY,
    username CHAR(8),
    firstname VARCHAR(30),
    lastname VARCHAR(30),
    city VARCHAR(30),
    state CHAR(2),
    email VARCHAR(100),
    phone CHAR(14),
    likesports BOOLEAN,
    liketheatre BOOLEAN,
    likeconcerts BOOLEAN,
    likejazz BOOLEAN,
    likeclassical BOOLEAN,
    likeopera BOOLEAN,
    likerock BOOLEAN,
    likevegas BOOLEAN,
    likebroadway BOOLEAN,
    likemusicals BOOLEAN
  );" 2>&1)
echo "$USERS_TABLE_OUTPUT"
check_error "$USERS_TABLE_OUTPUT" "aws redshift-data execute-statement (users
 table)"
USERS_QUERY_ID=$(echo "$USERS_TABLE_OUTPUT" | grep -o '"Id": "[^"]*' | cut -d'"'
 -f4)
```

```
# Wait for query to complete
echo "Waiting for users table creation to complete..."
sleep 5

# Create event table
echo "Creating event table..."
EVENT_TABLE_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "CREATE TABLE event(
    eventid INTEGER NOT NULL DISTKEY,
    venueid SMALLINT NOT NULL,
    catid SMALLINT NOT NULL,
    dateid SMALLINT NOT NULL SORTKEY,
    eventname VARCHAR(200),
    starttime TIMESTAMP
  );" 2>&1)
echo "$EVENT_TABLE_OUTPUT"
check_error "$EVENT_TABLE_OUTPUT" "aws redshift-data execute-statement (event
 table)"
EVENT_QUERY_ID=$(echo "$EVENT_TABLE_OUTPUT" | grep -o '"Id": "[^"]*' | cut -d'"'
 -f4)

# Wait for query to complete
echo "Waiting for event table creation to complete..."
sleep 5

# Create sales table
echo "Creating sales table..."
SALES_TABLE_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "CREATE TABLE sales(
    salesid INTEGER NOT NULL,
    listid INTEGER NOT NULL DISTKEY,
    sellerid INTEGER NOT NULL,
    buyerid INTEGER NOT NULL,
    eventid INTEGER NOT NULL,
    dateid SMALLINT NOT NULL SORTKEY,
    qtysold SMALLINT NOT NULL,
    pricepaid DECIMAL(8,2),
    commission DECIMAL(8,2),
    saletime TIMESTAMP
  );" 2>&1)
```

```
echo "$SALES_TABLE_OUTPUT"
check_error "$SALES_TABLE_OUTPUT" "aws redshift-data execute-statement (sales
 table)"
SALES_QUERY_ID=$(echo "$SALES_TABLE_OUTPUT" | grep -o '"Id": "[^"]*' | cut -d'"'
 -f4)

# Wait for tables to be created
echo "Waiting for tables to be created..."
sleep 10

# Step 5: Load sample data from Amazon S3
echo "Loading sample data from Amazon S3..."

# Load data into users table
echo "Loading data into users table..."
USERS_LOAD_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "COPY users
    FROM 's3://redshift-downloads/tickit/allusers_pipe.txt'
    DELIMITER '|'
    TIMEFORMAT 'YYYY-MM-DD HH:MI:SS'
    IGNOREHEADER 1
    IAM_ROLE '$ROLE_ARN';" 2>&1)
echo "$USERS_LOAD_OUTPUT"
check_error "$USERS_LOAD_OUTPUT" "aws redshift-data execute-statement (load
 users)"
USERS_LOAD_QUERY_ID=$(echo "$USERS_LOAD_OUTPUT" | grep -o '"Id": "[^"]*' | cut -
d'"' -f4)

# Wait for data loading to complete
echo "Waiting for users data loading to complete..."
sleep 10

# Load data into event table
echo "Loading data into event table..."
EVENT_LOAD_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "COPY event
    FROM 's3://redshift-downloads/tickit/allevents_pipe.txt'
    DELIMITER '|'
    TIMEFORMAT 'YYYY-MM-DD HH:MI:SS'
    IGNOREHEADER 1
```

```
    IAM_ROLE '$ROLE_ARN';" 2>&1)
echo "$EVENT_LOAD_OUTPUT"
check_error "$EVENT_LOAD_OUTPUT" "aws redshift-data execute-statement (load
 event)"
EVENT_LOAD_QUERY_ID=$(echo "$EVENT_LOAD_OUTPUT" | grep -o '"Id": "[^"]*' | cut -
d'"' -f4)

# Wait for data loading to complete
echo "Waiting for event data loading to complete..."
sleep 10

# Load data into sales table
echo "Loading data into sales table..."
SALES_LOAD_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "COPY sales
    FROM 's3://redshift-downloads/tickit/sales_tab.txt'
    DELIMITER '\t'
    TIMEFORMAT 'MM/DD/YYYY HH:MI:SS'
    IGNOREHEADER 1
    IAM_ROLE '$ROLE_ARN';" 2>&1)
echo "$SALES_LOAD_OUTPUT"
check_error "$SALES_LOAD_OUTPUT" "aws redshift-data execute-statement (load
 sales)"
SALES_LOAD_QUERY_ID=$(echo "$SALES_LOAD_OUTPUT" | grep -o '"Id": "[^"]*' | cut -
d'"' -f4)

# Wait for data loading to complete
echo "Waiting for sales data loading to complete..."
sleep 30

# Step 6: Run sample queries
echo "Running sample queries..."

# Query 1: Find top 10 buyers by quantity
echo "Running query: Find top 10 buyers by quantity..."
QUERY1_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "SELECT firstname, lastname, total_quantity
    FROM (SELECT buyerid, sum(qtysold) total_quantity
          FROM sales
          GROUP BY buyerid
```

```
          ORDER BY total_quantity desc limit 10) Q, users
    WHERE Q.buyerid = userid
    ORDER BY Q.total_quantity desc;" 2>&1)
echo "$QUERY1_OUTPUT"
check_error "$QUERY1_OUTPUT" "aws redshift-data execute-statement (query 1)"
QUERY1_ID=$(echo "$QUERY1_OUTPUT" | grep -o '"Id": "[^"]*' | cut -d'"' -f4)

# Wait for query to complete
echo "Waiting for query 1 to complete..."
sleep 10

# Get query 1 results
echo "Getting results for query 1..."
QUERY1_STATUS_OUTPUT=$(aws redshift-data describe-statement --id "$QUERY1_ID"
 2>&1)
echo "$QUERY1_STATUS_OUTPUT"
check_error "$QUERY1_STATUS_OUTPUT" "aws redshift-data describe-statement (query
 1)"

QUERY1_STATUS=$(echo "$QUERY1_STATUS_OUTPUT" | grep -o '"Status": "[^"]*' | cut -
d'"' -f4)
if [ "$QUERY1_STATUS" == "FINISHED" ]; then
  QUERY1_RESULTS=$(aws redshift-data get-statement-result --id "$QUERY1_ID" 2>&1)
  echo "Query 1 Results:"
  echo "$QUERY1_RESULTS"
else
  echo "Query 1 is not yet complete. Status: $QUERY1_STATUS"
  echo "Waiting additional time for query to complete..."
  sleep 20

  # Check again
  QUERY1_STATUS_OUTPUT=$(aws redshift-data describe-statement --id "$QUERY1_ID"
 2>&1)
  QUERY1_STATUS=$(echo "$QUERY1_STATUS_OUTPUT" | grep -o '"Status": "[^"]*' | cut
 -d'"' -f4)

  if [ "$QUERY1_STATUS" == "FINISHED" ]; then
    QUERY1_RESULTS=$(aws redshift-data get-statement-result --id "$QUERY1_ID"
 2>&1)
    echo "Query 1 Results:"
    echo "$QUERY1_RESULTS"
  else
    echo "Query 1 is still not complete. Status: $QUERY1_STATUS"
  fi
```

```
fi

# Query 2: Find events in the 99.9 percentile in terms of all time total sales
echo "Running query: Find events in the 99.9 percentile in terms of all time
 total sales..."
QUERY2_OUTPUT=$(aws redshift-data execute-statement \
  --database "$DB_NAME" \
  --workgroup-name "$WORKGROUP_NAME" \
  --sql "SELECT eventname, total_price
    FROM (SELECT eventid, total_price, ntile(1000) over(order by total_price
 desc) as percentile
          FROM (SELECT eventid, sum(pricepaid) total_price
                FROM sales
                GROUP BY eventid)) Q, event E
    WHERE Q.eventid = E.eventid
    AND percentile = 1
    ORDER BY total_price desc;" 2>&1)
echo "$QUERY2_OUTPUT"
check_error "$QUERY2_OUTPUT" "aws redshift-data execute-statement (query 2)"
QUERY2_ID=$(echo "$QUERY2_OUTPUT" | grep -o '"Id": "[^"]*' | cut -d'"' -f4)

# Wait for query to complete
echo "Waiting for query 2 to complete..."
sleep 10

# Get query 2 results
echo "Getting results for query 2..."
QUERY2_STATUS_OUTPUT=$(aws redshift-data describe-statement --id "$QUERY2_ID"
 2>&1)
echo "$QUERY2_STATUS_OUTPUT"
check_error "$QUERY2_STATUS_OUTPUT" "aws redshift-data describe-statement (query
 2)"

QUERY2_STATUS=$(echo "$QUERY2_STATUS_OUTPUT" | grep -o '"Status": "[^"]*' | cut -
d'"' -f4)
if [ "$QUERY2_STATUS" == "FINISHED" ]; then
  QUERY2_RESULTS=$(aws redshift-data get-statement-result --id "$QUERY2_ID" 2>&1)
  echo "Query 2 Results:"
  echo "$QUERY2_RESULTS"
else
  echo "Query 2 is not yet complete. Status: $QUERY2_STATUS"
  echo "Waiting additional time for query to complete..."
  sleep 20
```

```
  # Check again
  QUERY2_STATUS_OUTPUT=$(aws redshift-data describe-statement --id "$QUERY2_ID"
 2>&1)
  QUERY2_STATUS=$(echo "$QUERY2_STATUS_OUTPUT" | grep -o '"Status": "[^"]*' | cut
 -d'"' -f4)

  if [ "$QUERY2_STATUS" == "FINISHED" ]; then
    QUERY2_RESULTS=$(aws redshift-data get-statement-result --id "$QUERY2_ID"
 2>&1)
    echo "Query 2 Results:"
    echo "$QUERY2_RESULTS"
  else
    echo "Query 2 is still not complete. Status: $QUERY2_STATUS"
  fi
fi

# Summary
echo ""
echo "========================================="
echo "TUTORIAL SUMMARY"
echo "========================================="
echo "You have successfully:"
echo "1. Created a Redshift Serverless namespace and workgroup"
echo "2. Created an IAM role with S3 access permissions"
echo "3. Stored admin credentials securely in AWS Secrets Manager"
echo "4. Created tables for sample data"
echo "5. Loaded sample data from Amazon S3"
echo "6. Run sample queries on the data"
echo ""
echo "Redshift Serverless Resources:"
echo "- Namespace: $NAMESPACE_NAME"
echo "- Workgroup: $WORKGROUP_NAME"
echo "- Database: $DB_NAME"
echo "- Endpoint: $WORKGROUP_ENDPOINT"
echo "- Credentials Secret: $SECRET_NAME"
echo ""
echo "To connect to your Redshift Serverless database using SQL tools:"
echo "- Host: $WORKGROUP_ENDPOINT"
echo "- Database: $DB_NAME"
echo "- Username: $ADMIN_USERNAME"
echo "- Password: Retrieve from AWS Secrets Manager secret '$SECRET_NAME'"
echo ""
echo "To retrieve the password from Secrets Manager (without jq):"
```

```
echo "aws secretsmanager get-secret-value --secret-id $SECRET_NAME --query
 'SecretString' --output text | sed -n 's/.*\"password\":\"\([^\"]*\)\".*/\1/p'"
echo ""

# Clean up temporary files
rm -f redshift-trust-policy.json redshift-s3-policy.json

# Clean up resources
cleanup_resources

echo "Tutorial completed at $(date)"
```

- 有关 API 详细信息，请参阅《AWS CLI 命令参考》中的以下主题。
  - CreateNamespace
  - CreateRole
  - CreateSecret
  - CreateWorkgroup
  - DeleteNamespace
  - DeleteRole
  - DeleteRolePolicy
  - DeleteSecret
  - DeleteWorkgroup
  - GetNamespace
  - GetRole
  - GetSecretValue
  - GetWorkgroup
  - PutRolePolicy
  - UpdateNamespace

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

## 预调配数据仓库集群入门

以下代码示例展示了如何：

- 创建 Redshift 集群
- 为 S3 访问创建 IAM 角色
- 创建表并加载数据
- 运行示例查询
- 清理资源

Bash

AWS CLI 及 Bash 脚本

> **Note**
>
> 查看 GitHub，了解更多信息。在 Sample developer tutorials 存储库中查找完整示例，了解如何进行设置和运行。

```
#!/bin/bash

# Amazon Redshift Provisioned Cluster Tutorial Script
# This script creates a Redshift cluster, loads sample data, runs queries, and
 cleans up resources
# Version 4: Security improvements and best practices

set -euo pipefail

# Set up logging
LOG_FILE="redshift_tutorial.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting Amazon Redshift tutorial script at $(date)"
echo "All commands and outputs will be logged to $LOG_FILE"

# Function to handle errors
handle_error() {
    echo "ERROR: $1" >&2
    echo "Resources created so far:"
    if [ -n "${CLUSTER_ID:-}" ]; then echo "- Redshift Cluster: $CLUSTER_ID"; fi
    if [ -n "${ROLE_NAME:-}" ]; then echo "- IAM Role: $ROLE_NAME"; fi

    echo "Attempting to clean up resources..."
```

```
    cleanup_resources
    exit 1
}

# Function to clean up resources
cleanup_resources() {
    echo "Cleaning up resources..."

    # Delete the cluster if it exists
    if [ -n "${CLUSTER_ID:-}" ]; then
        echo "Deleting Redshift cluster: $CLUSTER_ID"
        aws redshift delete-cluster --cluster-identifier "$CLUSTER_ID" --skip-
final-cluster-snapshot 2>/dev/null || true
        echo "Waiting for cluster deletion to complete..."
        aws redshift wait cluster-deleted --cluster-identifier "$CLUSTER_ID" 2>/
dev/null || true
        echo "Cluster deleted successfully."
    fi

    # Delete the IAM role if it exists
    if [ -n "${ROLE_NAME:-}" ]; then
        echo "Removing IAM role policy..."
        aws iam delete-role-policy --role-name "$ROLE_NAME" --policy-name
 RedshiftS3Access 2>/dev/null || true

        echo "Deleting IAM role: $ROLE_NAME"
        aws iam delete-role --role-name "$ROLE_NAME" 2>/dev/null || true
    fi

    # Clean up temporary files
    rm -f redshift-trust-policy.json redshift-s3-policy.json

    echo "Cleanup completed."
}

# Trap errors and cleanup
trap 'handle_error "Script interrupted"' INT TERM

# Function to wait for SQL statement to complete
wait_for_statement() {
    local statement_id=$1
    local max_attempts=30
    local attempt=1
    local status=""
```

```
    echo "Waiting for statement $statement_id to complete..."

    while [ $attempt -le $max_attempts ]; do
        status=$(aws redshift-data describe-statement --id "$statement_id" --
query 'Status' --output text 2>/dev/null || echo "")

        if [ "$status" == "FINISHED" ]; then
            echo "Statement completed successfully."
            return 0
        elif [ "$status" == "FAILED" ]; then
            local error=$(aws redshift-data describe-statement --id
 "$statement_id" --query 'Error' --output text 2>/dev/null || echo "Unknown
 error")
            echo "Statement failed with error: $error" >&2
            return 1
        elif [ "$status" == "ABORTED" ]; then
            echo "Statement was aborted." >&2
            return 1
        fi

        echo "Statement status: $status. Waiting... (Attempt $attempt/
$max_attempts)"
        sleep 10
        ((attempt++))
    done

    echo "Timed out waiting for statement to complete." >&2
    return 1
}

# Function to check if IAM role is attached to cluster
check_role_attached() {
    local role_arn=$1
    local max_attempts=10
    local attempt=1

    echo "Checking if IAM role is attached to the cluster..."

    while [ $attempt -le $max_attempts ]; do
        local status=$(aws redshift describe-clusters \
            --cluster-identifier "$CLUSTER_ID" \
            --query "Clusters[0].IamRoles[?IamRoleArn=='$role_arn'].ApplyStatus"
 \
```

```
            --output text 2>/dev/null || echo "")

        if [ "$status" == "in-sync" ]; then
            echo "IAM role is successfully attached to the cluster."
            return 0
        fi

        echo "IAM role status: $status. Waiting... (Attempt $attempt/
$max_attempts)"
        sleep 30
        ((attempt++))
    done

    echo "Timed out waiting for IAM role to be attached." >&2
    return 1
}

# Validate required commands
for cmd in aws jq; do
    if ! command -v "$cmd" &> /dev/null; then
        handle_error "Required command '$cmd' not found. Please install it and
 try again."
    fi
done

# Validate AWS credentials
if ! aws sts get-caller-identity &>/dev/null; then
    handle_error "AWS credentials not configured or invalid"
fi

# Variables to track created resources
CLUSTER_ID="examplecluster"
ROLE_NAME="RedshiftS3Role-$(date +%s)"
DB_NAME="dev"
DB_USER="awsuser"

# Generate secure password using AWS Secrets Manager or random string
if command -v openssl &> /dev/null; then
    DB_PASSWORD=$(openssl rand -base64 32 | tr -d "=+/" | cut -c1-20)
else
    DB_PASSWORD="TempPass$(date +%s | md5sum | cut -c1-20)"
fi

# Validate password meets requirements
```

```
if [ ${#DB_PASSWORD} -lt 8 ]; then
    handle_error "Generated password does not meet minimum length requirement"
fi

# Store password securely (optional: use AWS Secrets Manager in production)
echo "Generated database password (store securely): $DB_PASSWORD"

echo "=== Step 1: Creating Amazon Redshift Cluster ==="

# Create the Redshift cluster with encryption and audit logging enabled
echo "Creating Redshift cluster: $CLUSTER_ID"
CLUSTER_RESULT=$(aws redshift create-cluster \
  --cluster-identifier "$CLUSTER_ID" \
  --node-type ra3.xlplus \
  --number-of-nodes 2 \
  --master-username "$DB_USER" \
  --master-user-password "$DB_PASSWORD" \
  --db-name "$DB_NAME" \
  --port 5439 \
  --encrypted \
  --tags Key=project,Value=doc-smith Key=tutorial,Value=redshift-provisioned \
  2>&1) || handle_error "Failed to create Redshift cluster"

echo "$CLUSTER_RESULT"
echo "Waiting for cluster to become available..."

# Wait for the cluster to be available
aws redshift wait cluster-available --cluster-identifier "$CLUSTER_ID" ||
 handle_error "Timeout waiting for cluster to become available"

# Get cluster status to confirm
CLUSTER_STATUS=$(aws redshift describe-clusters \
  --cluster-identifier "$CLUSTER_ID" \
  --query 'Clusters[0].ClusterStatus' \
  --output text)

echo "Cluster status: $CLUSTER_STATUS"

echo "=== Step 2: Creating IAM Role for S3 Access ==="

# Create trust policy file with restricted permissions
echo "Creating trust policy for Redshift"
cat > redshift-trust-policy.json << 'EOF'
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

chmod 600 redshift-trust-policy.json

# Create IAM role
echo "Creating IAM role: $ROLE_NAME"
ROLE_RESULT=$(aws iam create-role \
  --role-name "$ROLE_NAME" \
  --assume-role-policy-document file://redshift-trust-policy.json 2>&1) ||
 handle_error "Failed to create IAM role"

echo "$ROLE_RESULT"

# Get the role ARN
ROLE_ARN=$(aws iam get-role --role-name "$ROLE_NAME" --query 'Role.Arn' --output
 text)
echo "Role ARN: $ROLE_ARN"

# Tag the IAM role
echo "Tagging IAM role: $ROLE_NAME"
aws iam tag-role --role-name "$ROLE_NAME" --tags Key=project,Value=doc-smith
 Key=tutorial,Value=redshift-provisioned

# Create policy document for S3 access with principle of least privilege
echo "Creating S3 access policy"
cat > redshift-s3-policy.json << 'EOF'
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
```

```
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::redshift-downloads",
        "arn:aws:s3:::redshift-downloads/*"
      ]
    }
  ]
}
EOF

chmod 600 redshift-s3-policy.json

# Attach policy to role
echo "Attaching S3 access policy to role"
POLICY_RESULT=$(aws iam put-role-policy \
  --role-name "$ROLE_NAME" \
  --policy-name RedshiftS3Access \
  --policy-document file://redshift-s3-policy.json 2>&1) || handle_error "Failed
 to attach policy to role"

echo "$POLICY_RESULT"

# Attach role to cluster
echo "Attaching IAM role to Redshift cluster"
ATTACH_ROLE_RESULT=$(aws redshift modify-cluster-iam-roles \
  --cluster-identifier "$CLUSTER_ID" \
  --add-iam-roles "$ROLE_ARN" 2>&1) || handle_error "Failed to attach role to
 cluster"

echo "$ATTACH_ROLE_RESULT"

# Wait for the role to be attached
echo "Waiting for IAM role to be attached to the cluster..."
if ! check_role_attached "$ROLE_ARN"; then
    handle_error "Failed to attach IAM role to cluster"
fi

echo "=== Step 3: Getting Cluster Connection Information ==="

# Get cluster endpoint
CLUSTER_INFO=$(aws redshift describe-clusters \
  --cluster-identifier "$CLUSTER_ID" \
  --query 'Clusters[0].Endpoint.{Address:Address,Port:Port}' \
```

```
  --output json)

echo "Cluster endpoint information:"
echo "$CLUSTER_INFO"

echo "=== Step 4: Creating Tables and Loading Data ==="

echo "Creating sales table"
SALES_TABLE_ID=$(aws redshift-data execute-statement \
  --cluster-identifier "$CLUSTER_ID" \
  --database "$DB_NAME" \
  --db-user "$DB_USER" \
  --sql "DROP TABLE IF EXISTS sales; CREATE TABLE sales(salesid integer not null,
 listid integer not null distkey, sellerid integer not null, buyerid integer
 not null, eventid integer not null, dateid smallint not null sortkey, qtysold
 smallint not null, pricepaid decimal(8,2), commission decimal(8,2), saletime
 timestamp);" \
  --query 'Id' --output text)

echo "Sales table creation statement ID: $SALES_TABLE_ID"

# Wait for statement to complete
if ! wait_for_statement "$SALES_TABLE_ID"; then
    handle_error "Failed to create sales table"
fi

echo "Creating date table"
DATE_TABLE_ID=$(aws redshift-data execute-statement \
  --cluster-identifier "$CLUSTER_ID" \
  --database "$DB_NAME" \
  --db-user "$DB_USER" \
  --sql "DROP TABLE IF EXISTS date; CREATE TABLE date(dateid smallint not null
 distkey sortkey, caldate date not null, day character(3) not null, week smallint
 not null, month character(5) not null, qtr character(5) not null, year smallint
 not null, holiday boolean default('N'));" \
  --query 'Id' --output text)

echo "Date table creation statement ID: $DATE_TABLE_ID"

# Wait for statement to complete
if ! wait_for_statement "$DATE_TABLE_ID"; then
    handle_error "Failed to create date table"
fi
```

```
echo "Loading data into sales table"
SALES_LOAD_ID=$(aws redshift-data execute-statement \
  --cluster-identifier "$CLUSTER_ID" \
  --database "$DB_NAME" \
  --db-user "$DB_USER" \
  --sql "COPY sales FROM 's3://redshift-downloads/tickit/sales_tab.txt' DELIMITER
 '\t' TIMEFORMAT 'MM/DD/YYYY HH:MI:SS' REGION 'us-east-1' IAM_ROLE '$ROLE_ARN';"
 \
  --query 'Id' --output text)

echo "Sales data load statement ID: $SALES_LOAD_ID"

# Wait for statement to complete
if ! wait_for_statement "$SALES_LOAD_ID"; then
    handle_error "Failed to load data into sales table"
fi

echo "Loading data into date table"
DATE_LOAD_ID=$(aws redshift-data execute-statement \
  --cluster-identifier "$CLUSTER_ID" \
  --database "$DB_NAME" \
  --db-user "$DB_USER" \
  --sql "COPY date FROM 's3://redshift-downloads/tickit/date2008_pipe.txt'
 DELIMITER '|' REGION 'us-east-1' IAM_ROLE '$ROLE_ARN';" \
  --query 'Id' --output text)

echo "Date data load statement ID: $DATE_LOAD_ID"

# Wait for statement to complete
if ! wait_for_statement "$DATE_LOAD_ID"; then
    handle_error "Failed to load data into date table"
fi

echo "=== Step 5: Running Example Queries ==="

echo "Running query: Get definition for the sales table"
QUERY1_ID=$(aws redshift-data execute-statement \
  --cluster-identifier "$CLUSTER_ID" \
  --database "$DB_NAME" \
  --db-user "$DB_USER" \
  --sql "SELECT * FROM pg_table_def WHERE tablename = 'sales';" \
  --query 'Id' --output text)

echo "Query 1 statement ID: $QUERY1_ID"
```

```
# Wait for statement to complete
if ! wait_for_statement "$QUERY1_ID"; then
    handle_error "Query 1 failed"
fi

# Get and display results
echo "Query 1 results (first 10 rows):"
aws redshift-data get-statement-result --id "$QUERY1_ID" --max-items 10

echo "Running query: Find total sales on a given calendar date"
QUERY2_ID=$(aws redshift-data execute-statement \
  --cluster-identifier "$CLUSTER_ID" \
  --database "$DB_NAME" \
  --db-user "$DB_USER" \
  --sql "SELECT sum(qtysold) FROM sales, date WHERE sales.dateid = date.dateid
 AND caldate = '2008-01-05';" \
  --query 'Id' --output text)

echo "Query 2 statement ID: $QUERY2_ID"

# Wait for statement to complete
if ! wait_for_statement "$QUERY2_ID"; then
    handle_error "Query 2 failed"
fi

# Get and display results
echo "Query 2 results:"
aws redshift-data get-statement-result --id "$QUERY2_ID"

echo "=== Tutorial Complete ==="
echo "The following resources were created:"
echo "- Redshift Cluster: $CLUSTER_ID"
echo "- IAM Role: $ROLE_NAME"

echo ""
echo "========================================="
echo "CLEANUP CONFIRMATION"
echo "========================================="
echo "Cleaning up all created resources..."
cleanup_resources
echo "All resources have been cleaned up."

# Securely clear password from memory
```

```
DB_PASSWORD=""

echo "Script completed at $(date)"
```

- 有关 API 详细信息，请参阅《AWS CLI 命令参考》中的以下主题。
  - CreateCluster：
  - CreateRole
  - DeleteCluster
  - DeleteRole
  - DeleteRolePolicy
  - DescribeClusters
  - GetRole
  - ModifyClusterIamRoles
  - PutRolePolicy
  - Wait

有关 AWS SDK 开发人员指南和代码示例的完整列表，请参阅 将此服务与 AWS SDK 结合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

# 文档历史记录

> **Note**
>
> 有关 Amazon Redshift 中的新特征的描述，请参阅新增特征。

下表介绍 2018 年 6 月之后对《Amazon Redshift 管理指南》的重要文档更改。如需对此文档更新的通知，您可以订阅 RSS 源。

API 版本：2012-12-01

有关对《Amazon Redshift 数据库开发人员指南》的更改的列表，请参阅 Amazon Redshift 数据库开发人员指南文档历史记录。

| 变更 | 说明 | 日期 |
| --- | --- | --- |
| Amazon Redshift 补丁 205 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 205。 | 2026 年 9 月 16 日 |
| Amazon Redshift 补丁 204 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 204。 | 2026 年 8 月 11 日 |
| Amazon Redshift 补丁 203 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域 | 2026 年 7 月 6 日 |

| | | |
|---|---|---|
| | 中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 203。 | |
| Amazon Redshift 补丁 202 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 202。 | 2026 年 5 月 28 日 |
| Amazon Redshift 补丁 201 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 201。 | 2026 年 4 月 22 日 |
| Amazon Redshift 补丁 200 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 200。 | 2026 年 3 月 26 日 |
| Amazon Redshift 补丁 199 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 199。 | 2026 年 3 月 4 日 |

| | | |
|---|---|---|
| Amazon Redshift 补丁 198 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 198。 | 2026 年 2 月 3 日 |
| 更新 Amazon Redshift 托管式策略 | 新增了 `AmazonRedshiftFederatedAuthorization` 托管式策略。 | 2025 年 11 月 21 日 |
| Amazon Redshift 补丁 197 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 197。 | 2025 年 11 月 20 日 |
| Amazon Redshift 补丁 196 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 196。 | 2025 年 11 月 10 日 |
| Amazon Redshift 补丁 195 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 195。 | 2025 年 10 月 24 日 |

| | | |
|---|---|---|
| Amazon Redshift 补丁 194 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 194。 | 2025 年 10 月 1 日 |
| Amazon Redshift 补丁 193 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 193。 | 2025 年 8 月 26 日 |
| Amazon Redshift 补丁 192 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 192。 | 2025 年 7 月 30 日 |
| Amazon Redshift 补丁 191 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 191。 | 2025 年 6 月 24 日 |

| | | |
|---|---|---|
| Amazon Redshift 补丁 190 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 190。 | 2025 年 5 月 7 日 |
| 更新 Amazon Redshift 托管式策略 | 使用权限 `lakeformation:GetDataAccess` 更新了 `AmazonRedshiftServiceLinkedRolePolicy` 托管式策略。 | 2025 年 3 月 13 日 |
| Amazon Redshift 补丁 189 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 189。 | 2025 年 3 月 7 日 |
| Amazon Redshift 补丁 188 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 188。 | 2025 年 2 月 5 日 |
| Amazon Redshift 补丁 187 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 187。 | 2024 年 12 月 20 日 |

| | | |
|---|---|---|
| 更新 Amazon Redshift 托管式策略 | 使用权限 `glue:GetCatalog` 和 `glue:GetCatalogs` 更新了 `AmazonRedshiftServiceLinkedRolePolicy` 托管式策略。 | 2024 年 12 月 3 日 |
| Amazon Redshift 补丁 186 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 186。 | 2024 年 10 月 22 日 |
| Amazon Redshift 补丁 185 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 185。 | 2024 年 10 月 9 日 |
| Amazon Redshift 补丁 184 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 184。 | 2024 年 9 月 12 日 |

| | | |
|---|---|---|
| Amazon Redshift 补丁 183 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 183。 | 2024 年 8 月 8 日 |
| Amazon Redshift 补丁 182 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 182。 | 2024 年 6 月 26 日 |
| Amazon Redshift 补丁 181 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 181。 | 2024 年 5 月 1 日 |
| 更新 Amazon Redshift 托管式策略 | 更新到具有权限 `servicequotas:GetServiceQuota` 来访问 AWS 配额或限制的 `AmazonRedshiftServiceLinkedRolePolicy` 托管式策略。 | 2024 年 3 月 8 日 |

| | | |
|---|---|---|
| 更新查询编辑器 v2 托管式策略 | 使用权限 `redshift-serverless:ListNamespaces` 和 `redshift-serverless:ListWorkgroups` 更新 `AmazonRedshiftQueryEditorV2FullAccess` 、`AmazonRedshiftQueryEditorV2NoSharing` 、`AmazonRedshiftQueryEditorV2ReadSharing` 和 `AmazonRedshiftQueryEditorV2ReadWriteSharing` 托管策略。 | 2024 年 2 月 21 日 |
| 更新 Amazon Redshift 只读访问托管式策略 | 使用权限 `redshift:ListRecommendations` 更新了 `AmazonRedshiftReadOnlyAccess` 托管式策略，以列出 Amazon Redshift Advisor 建议。 | 2024 年 2 月 7 日 |
| Amazon Redshift 补丁 180 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 180。 | 2023 年 12 月 29 日 |
| Amazon Redshift 补丁 179 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 179。 | 2023 年 11 月 9 日 |

| | | |
|---|---|---|
| 更新 Amazon Redshift 托管式策略 | 使用权限 `ec2:AssignIpv6Addresses` 和 `ec2:UnassignIpv6Addresses` 更新了 `AmazonRedshiftServiceLinkedRolePolicy` 托管式策略。 | 2023 年 10 月 31 日 |
| Amazon Redshift 补丁 178 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 178。 | 2023 年 9 月 25 日 |
| 更新查询编辑器 v2 托管式策略 | 使用权限 `sqlworkbench:GetAutocompletionMetadata` 和 `sqlworkbench:GetAutocompletionResource` 更新 `AmazonRedshiftQueryEditorV2NoSharing` 、`AmazonRedshiftQueryEditorV2ReadSharing` 和 `AmazonRedshiftQueryEditorV2ReadWriteSharing` 托管式策略。 | 2023 年 8 月 16 日 |
| 更新 Amazon Redshift 托管式策略 | 更新了 `AmazonRedshiftServiceLinkedRolePolicy` 托管式策略，以授予在 AWS Secrets Manager 上创建和管理管理员凭证密钥的权限。 | 2023 年 8 月 14 日 |

| | | |
|---|---|---|
| Amazon Redshift 补丁 177 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 177。 | 2023 年 8 月 3 日 |
| Amazon Redshift 补丁 176 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 176。 | 2023 年 6 月 8 日 |
| Amazon Redshift 补丁 175 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 175。 | 2023 年 4 月 28 日 |
| 更新 Amazon Redshift 托管式策略 | 更新 `AmazonRedshiftServiceLinkedRolePolicy` 托管式策略以移除 ec2 网络相关操作的权限。它们特别与 Purpose:RedshiftMigrateToVpc 资源标签有关联。 | 2023 年 4 月 27 日 |
| 更新 Amazon Redshift Data API 管理策略 | 使用权限 `redshift:GetClusterCredentialsWithIAM` 更新了 `AmazonRedshiftDataFullAccess` 托管式策略。 | 2023 年 4 月 7 日 |

| | | |
|---|---|---|
| 更新查询编辑器 v2 托管式策略 | 使用权限 sqlworkbench:GetSchemaInference 更新 AmazonRedshiftQueryEditorV2NoSharing 、AmazonRedshiftQueryEditorV2ReadSharing 和 AmazonRedshiftQueryEditorV2ReadWriteSharing 托管式策略。 | 2023 年 3 月 21 日 |
| Amazon Redshift 补丁 174 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 174。 | 2023 年 3 月 11 日 |
| 更新查询编辑器 v2 托管式策略 | 使用权限 sqlworkbench:AssociateNotebookWithTab 更新 AmazonRedshiftQueryEditorV2NoSharing 、AmazonRedshiftQueryEditorV2ReadSharing 和 AmazonRedshiftQueryEditorV2ReadWriteSharing 托管式策略。 | 2023 年 2 月 2 日 |

| | | |
|---|---|---|
| Amazon Redshift 补丁 173 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 173。 | 2023 年 1 月 20 日 |
| Amazon Redshift 补丁 172 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 172。 | 2022 年 11 月 17 日 |
| Amazon Redshift 补丁 171 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 171。 | 2022 年 11 月 9 日 |
| Amazon Redshift 补丁 170 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 170。 | 2022 年 7 月 20 日 |

| | | |
|---|---|---|
| Amazon Redshift 补丁 169 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 169。 | 2022 年 6 月 8 日 |
| Amazon Redshift 补丁 168 已发布。 | 正在部署一个新的 Amazon Redshift 补丁。新版本需要几周时间才能在所有支持 Amazon Redshift 的 AWS 区域中发布。有关此版本的更多信息，请参阅 Amazon Redshift 补丁 168。 | 2022 年 4 月 19 日 |
| 支持使用 Amazon Redshift 驱动程序的身份验证配置文件 | 您现在可以使用身份验证配置文件连接到 Amazon Redshift。 | 2021 年 8 月 2 日 |
| 支持由 AWS PrivateLink 提供支持的 Amazon Redshift 的跨 VPC 端点 | 您现在可以将 Redshift 托管的 VPC 端点与 Amazon Redshift 结合使用。 | 2021 年 4 月 1 日 |
| 支持 Amazon Redshift 查询编辑器增强功能 | 您现在可以使用具有增强型 VPC 路由、更长查询运行时间和更多集群节点类型的查询编辑器。 | 2021 年 2 月 17 日 |
| 支持与合作伙伴的控制台集成 | 您可以使用 Amazon Redshift 控制台与合作伙伴集成。 | 2020 年 12 月 9 日 |
| 支持在可用区之间移动集群的能力 | 现在，您可以在可用区之间移动 RA3 集群。 | 2020 年 12 月 9 日 |
| 支持 ra3.xlplus 节点类型 | 您现在可以创建 ra3.xlplus 节点类型。 | 2020 年 12 月 9 日 |

| | | |
|---|---|---|
| 支持 JDBC 驱动程序版本 2.0 | 您现在可以配置 JDBC 驱动程序版本 2.0。 | 2020 年 11 月 5 日 |
| 支持 Lambda UDF 和令牌化 | 您现在可以编写 Lambda UDF 来启用数据的外部令牌化。 | 2020 年 10 月 26 日 |
| 支持安排 SQL 语句的运行 | 您现在可以在 Amazon Redshift 控制台上安排查询。 | 2020 年 10 月 22 日 |
| 支持用于 Amazon Redshift 的数据 API | Amazon Redshift 现在可以使用内置数据 API 访问。文档更新包含 Amazon Redshift 数据 API 参考。 | 2020 年 9 月 10 日 |
| 支持 Amazon Redshift 控制台查询监控 | 更新了指南以描述新的查询监控图表。 | 2020 年 5 月 7 日 |
| 对使用限制的支持 | 更新了指南以描述使用限制。 | 2020 年 4 月 23 日 |
| 多重身份验证 | 更新了指南以描述多重验证支持。 | 2020 年 4 月 20 日 |
| 弹性调整大小现在支持节点类型更改 | 更新了弹性调整大小说明。 | 2020 年 4 月 6 日 |
| 支持具有托管存储的 ra3.4xlarge 节点类型 | 更新了指南以包含 ra3.4xlarge 节点类型。 | 2020 年 4 月 2 日 |
| 支持暂停和恢复 | 更新了指南以描述暂停和恢复集群操作。 | 2020 年 3 月 11 日 |
| 支持 Microsoft Azure AD 作为身份提供者 | 更新了指南，以描述将 Microsoft Azure AD 用作身份提供者的步骤。 | 2020 年 2 月 10 日 |
| 支持 RA3 节点类型 | 更新了指南以描述新的 RA3 节点类型。 | 2019 年 12 月 3 日 |
| 对新控制台的支持 | 更新了指南以描述新的 Amazon Redshift 控制台。 | 2019 年 11 月 11 日 |

| | | |
|---|---|---|
| 安全信息更新 | 对安全信息文档的更新。 | 2019 年 6 月 24 日 |
| 快照增强功能 | Amazon Redshift 现在支持多种管理和计划快照的增强功能。 | 2019 年 4 月 4 日 |
| 并发扩展 | 您可以配置工作负载管理 (WLM) 来启用并发扩展模式。有关更多信息，请参阅配置工作负载管理。 | 2019 年 3 月 21 日 |
| 更新了 JDBC 和 ODBC 驱动程序 | Amazon Redshift 现在支持新版本的 JDBC 和 ODBC 驱动程序。有关更多信息，请参阅配置 JDBC 连接。 | 2019 年 2 月 4 日 |
| 推迟的维护 | 如果需要重新计划集群的维护时段，您可以选择将维护最多延迟 14 天。如果我们需要在您推迟期间更新硬件或进行其他强制更新，我们会通知您并进行必要的更改。在这些更新期间，您的集群不可用。有关更多信息，请参阅推迟维护。 | 2018 年 11 月 20 日 |
| 预先通知 | Amazon Redshift 为某些事件提前提供通知。这些事件的事件类别是 pending。例如，如果集群中的一个节点需要硬件更新，我们会发送预先通知。您可以订阅与其他 Amazon Redshift 事件相同的待处理事件。有关更多信息，请参阅订阅 Amazon Redshift 事件通知。 | 2018 年 11 月 20 日 |

| | | |
|---|---|---|
| 弹性调整大小 | 弹性调整大小是调整集群大小的最快方法。弹性调整大小会在现有集群上添加或删除节点，然后自动将数据重新分配到新节点。因为它不会创建新集群，所以弹性调整大小操作通常会在几分钟内快速完成。有关更多信息，请参阅调整集群大小。 | 2018 年 11 月 15 日 |
| 取消调整大小操作 | 您现在可以在调整大小操作正在进行时取消它。有关更多信息，请参阅调整大小操作概述。 | 2018 年 11 月 2 日 |
| 修改集群以更改加密 | 您可以使用 AWS 托管式密钥或客户托管式密钥修改未加密的集群以使用 AWS Key Management Service（AWS KMS）加密。当您修改集群以启用 KMS 加密时，Amazon Redshift 会自动将您的数据迁移到新加密的集群。您还可以通过修改集群将未加密的集群迁移到加密的集群。 | 2018 年 10 月 16 日 |
| Amazon Redshift Spectrum 支持增强型 VPC 路由 | 现在，您可以在集群中使用启用了增强型 VPC 路由的 Redshift Spectrum。您可能需要执行其他配置步骤。有关更多信息，请参阅将 Amazon Redshift Spectrum 与增强型 VPC 路由结合使用。 | 2018 年 10 月 10 日 |
| 查询编辑器 | 您现在可以从 Amazon Redshift 管理控制台运行 SQL 查询。 | 2018 年 10 月 4 日 |

| | | |
|---|---|---|
| 工作负载执行细分图表 | 您现在可以通过查看控制台中的工作负载执行细分图表，详细查看工作负载的性能。有关更多信息，请参阅分析工作负载性能。 | 2018 年 7 月 30 日 |
| 维护跟踪 | 您现在可以通过选择维护跟踪来确定集群是否始终会更新到最新版的 Amazon Redshift 或先前版本。有关更多信息，请参阅选择集群维护跟踪。 | 2018 年 7 月 26 日 |

下表介绍了 2018 年 7 月前对《Amazon Redshift 管理指南》所作的一些重要更改。

| 更改 | 描述 | 发行日期 |
|---|---|---|
| 新的 CloudWatch 指标 | 针对监控查询性能添加了新 CloudWatch 指标。有关更多信息，请参阅Amazon Redshift 中的性能数据。 | 2018 年 5 月 17 日 |
| HSM 加密 | Amazon Redshift 仅支持适用于硬件安全模块 (HSM) 密钥管理的 AWS CloudHSM。有关更多信息，请参阅 Amazon Redshift 数据库加密。 | 2018 年 3 月 6 日 |
| IAM 角色创建 | 如果附加到集群的 IAM 角色无法访问必要的资源，则您可以串联另一个角色（可能属于其他账户）。然后，您的集群临时代入串联的角色来访问数据。您还可以通过串联角色来授予跨账户访问权限。链中的每个角色都会代入链中的下一个角色，直到集群承担位于链尾的角色。您可以串联最多 10 个角色。有关更多信息，请参阅 在 Amazon Redshift 中串联 IAM 角色。 | 2018 年 2 月 23 日 |
| 新的 DC2 节点类型 | 新一代密集计算 (DC) 节点类型以与 DC1 相同的价格提供更好的性能。为了利用性能改进，您可以将 DC1 集群迁移到较新的 DC2 节点类型。有关更多信息，请参阅 Amazon Redshift 中的集群和节点。 | 2017 年 10 月 17 日 |

| 更改 | 描述 | 发行日期 |
| --- | --- | --- |
| ACM 证书 | Amazon Redshift 正将您的集群中的 SSL 证书替换为 AWS Certificate Manager (ACM) 颁发的证书。ACM 是一个可信的公有证书颁发机构 (CA)，受当前大多数系统信任。您可能需要更新当前的信任根 CA 证书，才能继续使用 SSL 连接集群。有关更多信息，请参阅 将 SSL 连接过渡到 ACM 证书。 | 2017 年 9 月 18 日 |
| 服务相关角色 | 服务相关角色是一种独特类型的 IAM 角色，它与 Amazon Redshift 直接相关。服务相关角色由 Amazon Redshift 预定义，具有服务代表您的 Amazon Redshift 集群调用 AWS 服务所需的所有权限。有关更多信息，请参阅 对 Amazon Redshift 使用服务相关角色。 | 2017 年 9 月 18 日 |
| IAM 数据库用户身份验证 | 您可以将系统配置为允许用户创建用户凭证，并基于其 IAM 凭证登录数据库。也可以通过符合 SAML 2.0 标准的身份提供者配置系统，让用户使用联合单点登录进行登录。有关更多信息，请参阅 使用 IAM 身份验证生成数据库用户凭证。 | 2017 年 8 月 11 日 |
| 表级还原支持增强型 VPC 路由 | 使用使用增强型 VPC 路由控制网络流量的集群现在支持表级还原。有关更多信息，请参阅 从快照中还原表。 | 2017 年 7 月 19 日 |
| 查询监控规则 | 使用 WLM 查询监控规则，您可以为 WLM 查询定义基于指标的性能边界，并指定查询超出这些边界时需要采取的操作—log、hop 或 abort。您将在工作负载管理 (WLM) 配置中定义查询监控规则。有关更多信息，请参阅 工作负载管理。 | 2017 年 4 月 21 日 |
| 增强型 VPC 路由 | 在使用 Amazon Redshift 增强型 VPC 路由时，Amazon Redshift 会强制通过 Amazon VPC 路由集群和数据存储库之间的所有 COPY 和 UNLOAD 流量。有关更多信息，请参阅 使用 Redshift 增强型 VPC 路由控制网络流量。 | 2016 年 9 月 15 日 |

| 更改 | 描述 | 发行日期 |
| --- | --- | --- |
| 新的连接日志字段 | 连接日志 审计日志有两个用于跟踪 SSL 连接的新字段。如果您定期向 Amazon Redshift 表加载审计日志，则需要向目标表添加以下新列：sslcompression 和 sslexpansion。 | 2016 年 5 月 5 日 |
| 用于 COPY 和 UNLOAD 的 IAM 角色 | 您现在可指定您的集群可用于对其他 AWS 服务的访问的身份验证的一个或多个 AWS Identity and Access Management (IAM) 角色。IAM 角色提供了为 COPY、UNLOAD 或 CREATE LIBRARY 命令提供身份验证的更安全的替代方法。有关更多信息，请参阅授权 Amazon Redshift 代表您访问 AWS 服务和使用 IAM 角色授权 COPY、UNLOAD、CREATE EXTERNAL FUNCTION 和 CREATE EXTERNAL SCHEMA 操作。 | 2016 年 3 月 29 日 |
| 从表中还原 | 您可以将集群快照中的表还原为活动集群中的新表。有关更多信息，请参阅 从快照中还原表。 | 2016 年 3 月 10 日 |
| 在策略中使用 IAM 条件 | 您可使用 IAM 策略中的 Condition 元素进一步限制对资源的访问。有关更多信息，请参阅 使用 IAM 策略条件进行精细访问控制。 | 2015 年 12 月 10 日 |
| 修改公开访问性 | 可以修改 VPC 中的现有集群以更改它是否可公开访问。有关更多信息，请参阅 修改集群。 | 2015 年 11 月 20 日 |
| 文档修复 | 发布了各种文档修复。 | 2015 年 8 月 28 日 |
| 文档更新 | 更新了与配置网络设置有关的故障排除指南，以确保具有不同的最大传输单位 (MTU) 大小的主机能决定连接的包大小。有关更多信息，请参阅 查询似乎挂起，有时无法连接到集群。 | 2015 年 8 月 25 日 |
| 文档更新 | 修订了有关参数组的整个部分，以使其变得更清楚且更有条理。有关更多信息，请参阅 Amazon Redshift 参数组。 | 2015 年 8 月 17 日 |

| 更改 | 描述 | 发行日期 |
| --- | --- | --- |
| WLM 动态属性 | WLM 配置参数现在支持动态应用一些属性。其他属性保持静态更改，并需要重启关联的集群以便能应用配置更改。有关更多信息，请参阅WLM 动态和静态属性和Amazon Redshift 参数组。 | 2015 年 8 月 3 日 |
| 将 KMS 加密的集群复制到另一个 AWS 区域 | 添加了有关配置快照复制授权以支持将 AWS KMS 加密的集群复制到另一个 AWS 区域的内容。有关更多信息，请参阅 将 AWS KMS 加密的快照复制到另一个 AWS 区域。 | 2015 年 7 月 28 日 |
| 文档更新 | 更新了数据库加密部分，以更好地说明 Amazon Redshift 如何使用 AWS KMS 或 HSM 来管理密钥，以及加密过程如何使用所有这些选项。有关更多信息，请参阅 Amazon Redshift 数据库加密。 | 2015 年 7 月 28 日 |
| 新的节点类型 | Amazon Redshift 现在提供了一种新的节点类型 DS2。更新了现有节点类型的文档参考，以使用本版本中引入的新的名称。此外，还修订了该部分以便更好地说明节点类型组合和阐述默认配额限制。有关更多信息，请参阅 Amazon Redshift 中的集群和节点。 | 2015 年 6 月 9 日 |
| 预留节点产品 | 添加了有关新的预留节点产品的内容。此外，修订了该部分以便更好地说明和比较可用的产品，并提供了演示按需和保留节点定价如何影响计费的示例。有关更多信息，请参阅 预留节点。 | 2015 年 6 月 9 日 |
| 文档修复 | 发布了各种文档修复。 | 2015 年 4 月 30 日 |
| 新功能 | 此版本的 Amazon Redshift 引入了新的 ODBC 和 JDBC 驱动程序，这些驱动程序已经过优化以便用于 Amazon Redshift。有关更多信息，请参阅 使用 SQL 客户端工具连接到 Amazon Redshift 数据仓库。 | 2015 年 2 月 26 日 |

| 更改 | 描述 | 发行日期 |
| --- | --- | --- |
| 新功能 | 此版本的 Amazon Redshift 引入了集群性能指标，使您能够查看和分析查询执行详细信息。有关更多信息，请参阅 查看查询和加载。 | 2015 年 2 月 26 日 |
| 文档更新 | 添加了新的示例策略，该策略说明如何授予针对常见 AWS 服务操作和 Amazon Redshift 依赖的资源的权限。有关更多信息，请参阅 客户管理型策略示例。 | 2015 年 1 月 16 日 |
| 文档更新 | 更新了有关设置最大传输单位 (MTU) 以禁用 TCP/IP 极大帧的指南。有关更多信息，请参阅使用 EC2 创建集群和查询似乎挂起，有时无法连接到集群。 | 2015 年 1 月 16 日 |
| 文档更新 | 修订了有关 `wlm_json_configuration` 参数的内容，提供了示例语法以便在 Linux、macOS 和 Microsoft Windows 操作系统上使用 AWS CLI 来配置此参数。有关更多信息，请参阅 工作负载管理。 | 2015 年 1 月 13 日 |
| 文档更新 | 添加了缺少的事件通知和说明。有关更多信息，请参阅 Amazon Redshift 预置集群事件通知。 | 2015 年 1 月 8 日 |
| 文档更新 | 更新了有关针对 Amazon Redshift 操作和资源的 IAM 策略的指南。修订了该部分以使其变得更清楚且更有条理。有关更多信息，请参阅 Amazon Redshift 中的安全性。 | 2014 年 11 月 21 日 |
| 新功能 | 此版本的 Amazon Redshift 引入了用于通过来自 AWS Key Management Service (AWS KMS) 的加密密钥对集群进行加密的功能。AWS KMS 将安全、高度可用的硬件和软件结合起来，提供可扩展到云的密钥管理系统。有关 AWS KMS 的更多信息和 Amazon Redshift 的加密选项，请参阅Amazon Redshift 数据库加密和集群操作。 | 2014 年 11 月 12 日 |

| 更改 | 描述 | 发行日期 |
| --- | --- | --- |
| 新功能 | 此版本的 Amazon Redshift 引入了用于为资源（例如，集群和快照）添加标签的功能。利用标签，您可以提供用户定义的元数据来根据成本分配对账单报告进行分类，并帮助您更好地标识资源。有关更多信息，请参阅在 Amazon Redshift 中为资源添加标签。 | 2014 年 11 月 4 日 |
| 新功能 | 将 dw1.8xlarge 和 dw2.8xlarge 节点大小的最大节点限制增至 128 个节点。有关更多信息，请参阅 Amazon Redshift 中的集群和节点。 | 2014 年 10 月 30 日 |
| 新功能 | 添加了用于终止来自 Amazon Redshift 控制台的查询和加载的功能。有关更多信息，请参阅查看查询和加载和在加载操作期间查看集群指标。 | 2014 年 10 月 28 日 |
| 文档修复 | 发布了各种文档修复。 | 2014 年 10 月 17 日 |
| 新增内容 | 添加了有关关闭集群和删除集群的内容。有关更多信息，请参阅 关闭和删除集群。 | 2014 年 8 月 14 日 |
| 文档更新 | 阐明了集群的 Allow Version Upgrade 设置的行为。有关更多信息，请参阅 Amazon Redshift 预置集群。 | 2014 年 8 月 14 日 |
| 文档更新 | 修订了有关在 Amazon Redshift 控制台中使用集群的主题的过程、快照和组织。有关更多信息，请参阅 集群操作。 | 2014 年 7 月 11 日 |
| 新增内容 | 添加了有关调整 Amazon Redshift 集群大小的新教程，包括如何在调整集群大小的同时最大程度地减少集群处于只读模式的时间量。有关更多信息，请参阅 调整集群大小。 | 2014 年 6 月 27 日 |
| 新功能 | 添加了用于重命名集群的功能。有关更多信息，请参阅重命名集群和修改集群。 | 2014 年 6 月 2 日 |
| 新功能 | 添加了用于在从快照还原集群时选择其他参数组和安全组的选项。有关更多信息，请参阅 从快照还原集群。 | 2014 年 5 月 12 日 |

| 更改 | 描述 | 发行日期 |
| --- | --- | --- |
| 新功能 | 添加了新的部分以说明如何配置默认 Amazon CloudWatch 告警以监控 Amazon Redshift 集群中使用的磁盘空间百分比。此警报是集群创建过程中的新选项。有关更多信息，请参阅 默认磁盘空间警报。 | 2014 年 4 月 28 日 |
| 文档更新 | 阐明了有关 Amazon Redshift 中的 Elliptic curve Diffie—Hellman Exchange (ECDHE) 支持的信息。有关更多信息，请参阅 SSL。 | 2014 年 4 月 22 日 |
| 新功能 | 添加了有关 Amazon Redshift 支持 Elliptic curve Diffie—Hellman (ECDH) 密钥协商协议的陈述。有关更多信息，请参阅 SSL。 | 2014 年 4 月 18 日 |
| 文档更新 | 修订并重新整理了使用 SQL 客户端工具连接到 Amazon Redshift 数据仓库部分中的主题。添加了有关 JDBC 和 ODBC 连接的更多信息和一个关于连接问题的新的疑难解答部分。 | 2014 年 4 月 15 日 |
| 文档更新 | 添加了整个指南中的 IAM 策略示例中的版本。 | 2014 年 4 月 3 日 |
| 文档更新 | 添加了有关定价在调整集群大小时的工作方式的信息。有关更多信息，请参阅 预留节点。 | 2014 年 4 月 2 日 |
| 新功能 | 添加了有关新参数 `max_cursor_result_set_size` 的部分，此部分设置可为单个光标存储的最大结果集大小（以兆字节为单位）。此参数值还影响集群的并行活动光标数。有关更多信息，请参阅 Amazon Redshift 参数组。 | 2014 年 3 月 28 日 |
| 新功能 | 添加了有关 Cluster Version 字段的说明，现在包含集群引擎版本和数据库修订号。有关更多信息，请参阅 Amazon Redshift 预置集群。 | 2014 年 3 月 21 日 |
| 新功能 | 更新了大小调整过程以在集群的 Status 选项卡上显示新的大小调整进度信息。有关更多信息，请参阅 调整集群大小。 | 2014 年 3 月 21 日 |

| 更改 | 描述 | 发行日期 |
| --- | --- | --- |
| 文档更新 | 重新整理和更新了什么是 Amazon Redshift？，并修订了 Amazon Redshift 预置集群概览。发布了各种文档修复。 | 2014 年 2 月 21 日 |
| 新功能 | 添加了 Amazon Redshift 集群的新节点类型和大小，并且根据反馈重新编写了相关的集群概览主题，以使其变得更清楚且更有条理。有关更多信息，请参阅 Amazon Redshift 预置集群。 | 2014 年 1 月 23 日 |
| 新功能 | 添加了有关使用虚拟私有云中可公开访问的 Amazon Redshift 集群的弹性 IP (EIP) 地址的信息。有关 Amazon Redshift 中的 EIP 的更多信息，请参阅VPC 中的 Redshift 资源 和在 VPC 中创建 Redshift 预置集群或 Amazon Redshift Serverless 工作组。 | 2013 年 12 月 20 日 |
| 新功能 | 添加了有关 Amazon Redshift 的 AWS CloudTrail 日志的信息。有关 Amazon Redshift 支持 CloudTrail 的更多信息，请参阅使用 Cloudtrail 进行日志记录。 | 2013 年 12 月 13 日 |
| 新功能 | 添加了有关 Amazon Redshift 中的数据库审计日志记录功能的新用户活动日志和 `enable_user_activity_logging` 数据库参数的信息。有关数据库审计日志记录的更多信息，请参阅数据库审计日志记录。有关数据库参数的更多信息，请参阅 Amazon Redshift 参数组。 | 2013 年 12 月 6 日 |
| 新功能 | 已更新以描述如何配置 Amazon Redshift 以自动将自动化快照和手动快照复制到辅助 AWS 区域。有关配置跨区域快照复制的更多信息，请参阅将快照复制到其它 AWS 区域。 | 2013 年 11 月 14 日 |
| 新功能 | 添加了一个部分来描述连接和用户活动的 Amazon Redshift 审计日志记录，以及如何将这些日志存储在 Amazon S3 中。有关数据库审计日志记录的更多信息，请参阅数据库审计日志记录。 | 2013 年 11 月 11 日 |

| 更改 | 描述 | 发行日期 |
| --- | --- | --- |
| 新功能 | 添加了一个部分来描述 Amazon Redshift 加密与用于管理硬件安全模块 (HSM) 中的加密密钥并轮换加密密钥的新功能。有关加密、HSM 和密钥轮换的更多信息，请参阅 Amazon Redshift 数据库加密、使用硬件安全模块的加密和加密密钥轮换。 | 2013 年 11 月 11 日 |
| 新功能 | 已更新以描述如何使用 Amazon SNS 发布 Amazon Redshift 事件通知。有关 Amazon Redshift 事件通知的更多信息，请参阅Amazon Redshift 预置集群事件通知。 | 2013 年 11 月 11 日 |
| 新功能 | 已更新以描述 IAM 资源级权限。有关 Amazon Redshift IAM 权限的信息，请参阅Amazon Redshift 中的安全性。 | 2013 年 8 月 9 日 |
| 新功能 | 已更新以描述如何还原进度指标。有关更多信息，请参阅 从快照还原集群。 | 2013 年 8 月 9 日 |
| 新功能 | 已更新以描述集群快照共享和创建快照进度指标。有关更多信息，请参阅 共享快照。 | 2013 年 7 月 17 日 |
| 文档修复 | 发布了各种文档修复。 | 2013 年 7 月 8 日 |
| 新控制台屏幕 | 更新了《Amazon Redshift 管理指南》，使其与 Amazon Redshift 控制台中的变更保持一致。 | 2013 年 4 月 22 日 |
| 新指南 | 这是《Amazon Redshift 管理指南》的第一个版本。 | 2013 年 2 月 14 日 |