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 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

..... xvi
什麼是 Amazon Redshift？ 1
您第一次使用 Amazon Redshift 嗎？ 1
Amazon Redshift Serverless 功能概觀 2
Amazon Redshift 佈建叢集概觀 4
叢集管理 4
叢集存取和安全 5
監控叢集 6
資料庫 7
將 Amazon Redshift Serverless 與 Amazon Redshift 佈建資料倉儲進行比較 7
使用佈建叢集的 Amazon Redshift 管理介面 27
使用 AWS SDKs 28
簽署 HTTP 請求 29
設定 Amazon Redshift CLI 34
Amazon Redshift Serverless 35
什麼是 Amazon Redshift Serverless？ 35
Amazon Redshift Serverless 主控台 35
使用 Amazon Redshift Serverless 時的考量 39
Amazon Redshift Serverless 的運算容量 42
無伺服器端點容量的考量和限制 44
AI 驅動的擴展和最佳化 (預覽版) 45
Amazon Redshift Serverless 的帳單 50
運算容量的帳單 50
隨需運算容量的帳單 51
無伺服器保留的帳單 54
儲存的帳單 60
使用 Amazon Redshift Serverless 免費試用 61
帳單用量注意事項 61
具有連線集區的 Amazon Redshift Serverless 帳單 62
採用零 ETL 的 Amazon Redshift Serverless 成本最佳化 63
連線至 Amazon Redshift Serverless 63
連線至 Amazon Redshift Serverless 64
透過 JDBC 驅動程式連線至 Amazon Redshift Serverless 64
使用資料 API 連線至 Amazon Redshift Serverless 66

使用 SSL 連線到 Amazon Redshift Serverless 66
從 Amazon Redshift 受管 VPC 端點連線至 Amazon Redshift Serverless 68
從 VPC 端點介面 (AWS PrivateLink) 連線至 Amazon Redshift Serverless 68
從其他帳戶中的 Redshift VPC 端點連線至 Amazon Redshift Serverless 68
其他資源 73
定義聯合身分使用者的資料庫角色 73
定義資料庫角色 74
定義要向聯合身分使用者授予之資料庫角色的使用案例 74
其他資源 76
Amazon Redshift Serverless 中的身分和存取管理 76
授予許可 76
IAM 憑證入門 77
使用資料庫角色許可存取資料庫物件 78
將佈建的叢集遷移至 Amazon Redshift Serverless 79
建立已佈建叢集的快照 80
使用驅動程式連線至 Amazon Redshift Serverless 81
使用 Amazon Redshift Serverless SDK 87
工作群組和命名空間 87
使用主控台的工作群組和命名空間 87
使用 和 Amazon Redshift Serverless API 的工作群組 AWS Command Line Interface 和命名空間 88
工作群組 89
命名空間 93
監控查詢和工作負載 96
新增查詢監控政策 98
對使用者授予查詢監控許可 99
對角色授予查詢監控許可 99
設定用量限制 100
設定查詢限制 101
設定查詢佇列 101
使用儀表板查看摘要資料 105
稽核記錄 106
在 CloudWatch 中記錄事件 106
CloudWatch 指標 107
快照和復原點 113
AWS Backup 整合 114

建立快照 ... 115
建立最終快照 ... 116
共用快照或移除快照許可 ... 116
排程快照 ... 116
更新快照保留期間 ... 119
刪除快照 ... 119
還原快照 ... 120
轉換復原點 ... 121
還原復原點 ... 121
將備份複製到另一個備份 AWS 區域 ... 121
還原資料表 ... 123
資料共用 ... 124
考量事項 ... 124
授予檢視資料共用的存取權 ... 124
將命名空間註冊到 AWS Glue Data Catalog ... 125
標記資源 ... 126
Amazon Redshift 佈建叢集 ... 128
叢集和節點 ... 128
節點類型詳細資訊 ... 130
決定節點數目 ... 133
使用 EC2 建立叢集 ... 134
Amazon Virtual Private Cloud (Amazon VPC) ... 134
預設磁碟空間警示 ... 135
叢集狀態 ... 135
使用佈建叢集的考量事項 ... 137
區域與可用區域的考量 ... 137
叢集維護 ... 138
叢集操作 ... 151
建立叢集 ... 151
建立磁碟空間警示 ... 154
檢視叢集 ... 154
修改叢集 ... 155
調整叢集大小 ... 156
重新命名叢集 ... 170
升級叢集的發行版本 ... 170
暫停和繼續叢集 ... 171

重新啟動叢集 ... 173
重新定位叢集 ... 173
設定用量限制 ... 177
關閉及刪除叢集 ... 178
快照和備份 ... 179
將叢集註冊到 AWS Glue Data Catalog ... 200
多可用區域部署 ... 201
設定多可用區部署 ... 201
在建立新叢集時設定多可用區 ... 204
為從快照還原的資料倉儲設定多可用區 ... 206
將單一可用區資料倉儲轉換為多可用區資料倉儲 ... 207
將多可用區資料倉儲轉換為單一可用區資料倉儲 ... 209
調整多可用區資料倉儲的大小 ... 210
容錯移轉多可用區部署 ... 210
檢視多可用區資料倉儲的查詢和載入 ... 212
監控多可用區部署中的查詢 ... 213
終止叢集的查詢 ... 214
監控叢集效能 ... 214
效能資料 ... 215
檢視效能資料 ... 228
分析查詢執行 ... 251
建立 警示 ... 253
終止執行中的查詢 ... 253
CloudWatch 主控台中的效能指標 ... 254
查詢分析工具 ... 255
查詢和資料庫監控 ... 264
Sys 檢視型查詢和資料庫監控 ... 269
追蹤 ... 272
在追蹤之間切換 ... 272
追蹤和還原 ... 273
更新無伺服器工作群組 ... 273
管理版本 ... 274
判斷工作群組或叢集版本 ... 274
零 ETL 整合 ... 276
考量事項 ... 278
在目標上使用歷史記錄模式時的考量事項 ... 279

零 ETL 整合來源為 Aurora 或 Amazon RDS 時的考量事項 280
零 ETL 整合來源為 DynamoDB 時的考量事項 281
零 ETL 整合來源為 Salesforce、SAP、ServiceNow 和 Zendesk 等應用程式時的考量事項 .. 282
還原零 ETL 整合目標時的考量事項 282
調整零 ETL 整合目標大小時的考量事項 283
開始使用零 ETL 整合 284
建立和設定目標 Amazon Redshift 資料倉儲 284
開啟區分大小寫功能 285
在 Amazon Redshift 中設定授權 286
建立零 ETL 整合 291
建立目的地資料庫 301
查詢複寫的資料 303
透過具體化視觀表查詢複寫的資料 304
查詢從 DynamoDB 複寫的資料 306
檢視零 ETL 整合 307
歷史記錄模式 310
共用您的資料 312
監控零 ETL 整合 312
監控與 Amazon Redshift 系統檢視的零 ETL 整合 312
監控與 Amazon EventBridge 的零 ETL 整合 313
零 ETL 整合的指標 313
修改 DynamoDB 的零 ETL 整合 316
刪除 DynamoDB 的零 ETL 整合 317
支援的區域 318
Aurora MySQL 318
Aurora PostgreSQL 320
Amazon DynamoDB 322
Amazon RDS for MySQL 324
企業應用程式 326
對零 ETL 整合進行疑難排解 328
查詢資料庫 343
連線至 Amazon Redshift 343
使用 Amazon Redshift 查詢編輯器 v2 來查詢資料庫 344
設定您的 AWS 帳戶 345
開啟查詢編輯器 v2 351
連線至 Amazon Redshift 資料庫 355

瀏覽 Amazon Redshift 資料庫 356
建立資料庫物件 358
檢視查詢和標籤歷史記錄 365
與 Amazon Q 生成式 SQL 互動 365
將資料載入到資料庫 377
編寫查詢 385
Notebooks 390
查詢 AWS Glue Data Catalog 393
查詢資料湖 396
資料共用 398
排程查詢 401
將結果視覺化 411
以團隊的形式協作和共用 416
使用查詢編輯器 v1 查詢資料庫 417
考量事項 418
使用 SQL 用戶端工具連線至資料倉儲 418
透過用戶端工具連線的建議 419
在 Amazon Redshift 中設定連線 419
探索中繼資料的最佳實務 609
設定連線的安全選項 616
從用戶端工具和程式碼連線 623
使用身分驗證設定檔連線至 Amazon Redshift 669
針對 Amazon Redshift 中的連線問題進行疑難排解 672
使用資料 API 679
使用資料 API 679
呼叫資料 API 時的考量 680
選擇資料庫身分驗證憑證 682
映射 JDBC 資料類型 683
執行含有參數的 SQL 陳述式 683
執行含有等冪性字符的 SQL 陳述式 685
執行含有 session reuse 的 SQL 陳述式 686
擷取結果 688
授權存取 690
信任的身分傳播 699
呼叫資料 API 701
針對資料 API 問題進行故障診斷 736

使用 Amazon EventBridge 來排程資料 API 操作 736
監控資料 API 740
AWS KMS 搭配資料 API 使用 742
使用 Amazon Sagemaker Unified Studio 745
參數群組 746
預設參數值 747
工作負載管理 748
WLM 動態和靜態屬性 749
WLM 組態參數的屬性 749
使用 設定 WLM 參數 AWS CLI 755
建立參數群組 762
修改參數群組 763
建立查詢監控規則 767
刪除參數群組 768
與 AWS 合作夥伴整合 769
使用 AWS 合作夥伴載入資料 770
預留節點 772
預留節點優惠 772
比較預留節點方案間的定價 773
預留節點運作的方式 774
預留節點和合併帳單 775
在使用預留執行個體的叢集上變更節點類型之前需要考量的事項 775
預留節點範例 775
範例 1 775
範例 2 776
範例 3 776
範例 4 776
範例 5 776
範例 6 776
購買預留節點 776
安全 779
資料保護 780
資料加密 781
資料字符化 797
路由網際網路流量 797
身分與存取管理 798

使用身分進行身分驗證 ... 798
預防跨服務混淆代理人 ... 799
存取控制 ... 801
管理存取概觀 ... 801
使用以身分為基礎的政策 (IAM 政策) ... 807
原生身分提供者 (IdP) 聯合 ... 863
使用服務連結角色 ... 893
使用 IAM 身分驗證產生資料庫使用者憑證 ... 901
授權存取 AWS 服務 ... 952
管理管理員密碼 ... 984
AWS Secrets Manager 整合所需的許可 ... 985
管理員密碼機密輪換 ... 985
搭配使用 AWS Secrets Manager 與 Amazon Redshift 的考量事項 ... 986
擷取機密的 ARN ... 986
建立資料庫連線憑證的機密 ... 987
日誌記錄和監控 ... 989
資料庫稽核記錄日誌 ... 990
使用 CloudTrail 進行記錄 ... 1005
系統資料表與 S3 資料表整合 ... 1017
法規遵循驗證 ... 1029
恢復能力 ... 1030
基礎設施安全性 ... 1030
網路隔離 ... 797
Security groups (安全群組) ... 1032
介面 VPC 端點 ... 1032
組態與漏洞分析 ... 1039
使用身分中心身分驗證 ... 1040
先決條件 ... 1040
Identity Center 身分驗證的運作方式 ... 1041
GetIdentityCenterAuthToken API 操作 ... 1041
已佈建叢集的 GetIdentityCenterAuthToken ... 1041
適用於無伺服器工作群組的 GetIdentityCenterAuthToken ... 1042
回應語法 ... 1043
驅動程式整合 ... 1044
直接字符用量 ... 1044
Java 程式碼範例 ... 1045

IAM 政策要求 1045
API 許可 1045
範例 IAM 政策 1046
區域可用性 1047
疑難排解 1048
API 錯誤 1048
連線錯誤 1048
聯網任務 1050
自訂網域名稱來進行用戶端連線 1050
註冊網域名稱 1050
請求網域名稱的憑證 1052
設定自訂網域 1053
連線至佈建叢集或工作群組 1054
重新命名已指派自訂網域的叢集 1055
說明自訂網域關聯 1055
將自訂網域與不同的憑證建立關聯 1056
刪除自訂網域 1057
Redshift 受管 VPC 端點 1057
考量事項 1058
授予 VPC 的存取權 1059
建立 Redshift 管理的 VPC 端點 1060
VPC 中的 Redshift 資源 1060
在 VPC 中建立叢集或工作群組 1063
VPC security groups (VPC 安全群組) 1064
設定叢集或工作群組的安全設定 1065
Redshift 資源的子網路 1067
封鎖對 VPC 和子網路的公開存取 1070
使用增強型 VPC 路由控制網路流量 1071
使用 VPC 端點控制資料庫流量 1072
開啟增強型 VPC 路由 1073
使用增強型 VPC 路由查詢資料湖資料表 1075
事件 1080
叢集事件通知訂閱 1080
建立事件通知訂閱 1082
佈建叢集事件通知 1083
Amazon Redshift Serverless 事件通知 1100

零 ETL 整合事件通知 1109
配額和限制 1116
Amazon Redshift 物件的配額 1116
Amazon Redshift Serverless 物件的配額 1122
Amazon Redshift Data API 的配額 1124
查詢編輯器 v2 物件的配額 1126
Amazon Redshift Spectrum 物件的配額和限制 1127
命名限制條件 1128
標籤資源 1131
標記需求 1132
管理資源標籤 1132
AWS Backup 整合 1133
考量事項 1134
限制 1134
AWS Backup 搭配 Amazon Redshift 使用 1135
叢集版本 1136
修補程式 205 1137
新功能 1138
修補程式 204 1139
新功能 1139
修補程式 203 1142
新功能 1143
修補程式 202 1144
新功能 1145
修補程式 201 1147
新功能 1147
修補程式 200 1149
新功能 1149
修補程式 199 1151
新功能 1151
修補程式 198 1153
新功能 1153
修補程式 197 1154
新功能 1155
修補程式 196 1156
新功能 1157

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

修補程式 177 1179
新功能 1180
修補程式 176 1181
新功能 1182
修補程式 175 1183
新功能 1183
修補程式 174 1183
此版本的新功能 1183
此版本的新功能 1183
此版本的新功能 1183
此版本的新功能 1183
此版本的新功能 1183
此版本的新功能 1183
此版本的新功能 1183
修補程式 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
修補程式 172 1186
新功能 1187
修補程式 171 1187
新功能 1187
修補程式 170 1187
新功能 1188
修補程式 169 1188
新功能 1188
修補程式 168 1188
新功能 1189
行為變更 1190

即將發生的行為變更 1190
Amazon Redshift 從 2026 年 9 月 30 日開始拍攝暫停資料共用生產者叢集的自動快照 1190
SUPER 資料類型支援從修補程式 206 開始的較大個別字串 1191
AWS KMS 2026 年 8 月 17 日之後 Amazon Redshift Serverless APIs 的金鑰許可強制執行 1192
從修補程式 208 開始，Amazon Redshift 會在多次嘗試登入失敗後強制執行使用者鎖定 1193
自 2026 年 6 月 8 日起，Amazon Redshift Serverless 和 Amazon Redshift RG 執行個體上手動快照的增強型計費模型 1193
Lake Formation 資料表上的 Iceberg DELETE 需要從修補程式 202 開始的 DELETE 許可 .. 1194
從修補程式 202 開始，Amazon Redshift Serverless 會在快照還原時保留零 ETL 和 S3 事件整合 1194
2026 年 12 月 31 日終止對 Amazon Redshift ODBC 1.x 驅動程式的支援 1195
純量 Python UDF 將於 2026 年 6 月 30 日之後終止支援 1195
2026 年 2 月 27 日之後的具體化視觀表 (MV) Auto-REFRESH 行為變更 1196
Amazon Redshift 在 2026 年 2 月 16 日之後，即不支援透過資料共用存取取用者資訊的函式 1196
最低 Transport Layer Security (TLS) 版本變更自 2026 年 10 月 31 日開始生效 1196
Amazon Redshift 在 2025 年 10 月 30 日之後，即不支援建立新的純量 Python UDF 1197
近期行為變更 1198
Amazon Redshift 在 2025 年 8 月 26 日之後使用最新的 IANA 時區資料庫 1198
Amazon Redshift Serverless RPU 變更於 2025 年 8 月 15 日之後生效 1199
資料庫稽核記錄變更於 2025 年 8 月 10 日之後生效 1199
無伺服器工作群組的虛擬私有雲端點變更於 2025 年 6 月 27 日之後生效 1199
查詢監控變更在 2025 年 5 月 2 日之後生效 1200
安全變更在 2025 年 1 月 10 日之後生效 1200
程式碼範例 1202
基本概念 1203
Hello Amazon Redshift 1203
了解基本概念 1208
動作 1274
案例 1332
建立用於追蹤 Amazon Redshift 資料的 Web 應用程式 1332
開始使用無伺服器資料倉儲 1333
佈建資料倉儲叢集入門 1350
文件歷史紀錄 1362

Amazon Redshift 將不再支援在 2026 年 6 月 30 日之後使用 Python UDFs。我們將開始分階段強制執行。如需 Python 生命週期結束和遷移選項的詳細資訊，請參閱 2025 年 6 月 30 日發佈的部落格文章。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。

什麼是 Amazon Redshift？

歡迎使用《Amazon Redshift 管理指南》。Amazon Redshift 是一種在雲端中完全受管的 PB 級資料倉儲服務。Amazon Redshift Serverless 可讓您在沒有所佈建資料倉儲的所有組態的情況下，存取和分析資料。系統會自動佈建資源，並有智慧地擴展資料倉儲容量，即使是最嚴苛且無法預測的工作負載，也能為其提供快速的效能。資料倉儲閒置時不會產生費用，因此只需按實際用量支付費用。您可以在 Amazon Redshift 查詢編輯器 v2 或您最愛的商業智慧 (BI) 工具中立即載入資料並開始查詢。請在易於使用、無須管理的環境中，享受最佳的性價比和熟悉的 SQL 功能。

無論資料集大小為何，Amazon Redshift 都能使用您目前所用的相同 SQL 工具和商業智慧應用程式，提供快速的查詢效能。

您第一次使用 Amazon Redshift 嗎？

如果您是第一次使用 Amazon Redshift，建議您從閱讀下列章節開始：

- 服務特色和定價 — 此產品詳細資訊頁面提供 Amazon Redshift 的價值主張、服務特色和定價。
- 開始使用 Amazon Redshift Serverless 資料倉儲 - 本主題會引導您完成設定無伺服器資料倉儲、建立資源和查詢範例資料的程序。
- Amazon Redshift 資料庫開發人員指南 — 若您是資料庫開發人員，此指南會說明如何設計、建置、查詢和維護組成資料倉儲的資料庫。

如果您偏好手動管理 Amazon Redshift 資源，則可以針對資料查詢需求建立佈建叢集。如需詳細資訊，請參閱 Amazon Redshift 叢集。

身為應用程式開發人員，您可以使用 Amazon Redshift API 或 AWS 軟體開發套件 (SDK) 程式庫，以程式設計方式管理叢集。如果您使用 Amazon Redshift API，您必須透過簽署它來驗證對 API 的每個 HTTP 或 HTTPS 請求。如需簽署請求的相關資訊，請前往簽署 HTTP 請求。

如需 API、CLI 和 SDK 的資訊，請前往下列連結：

- Amazon Redshift Serverless API 參考
- Amazon Redshift API 參考
- Amazon Redshift 資料 API API 參考
- AWS CLI 命令參考
- Amazon Web Services 的工具中的軟體開發套件參考。

Amazon Redshift Serverless 功能概觀

Amazon Redshift 佈建資料倉儲所支援的大部分功能也受到 Amazon Redshift Serverless 支援。以下是它的一些重要功能。

功能	說明
快照	您可以將 Amazon Redshift Serverless 或佈建資料倉儲的快照還原到 Amazon Redshift Serverless。如需詳細資訊，請參閱快照和復原點。
復原點	Amazon Redshift Serverless 會每 30 分鐘自動建立一個復原點。這些復原點會保留 24 小時。您可以在發生意外寫入或刪除後使用它們來進行還原。當您從復原點還原時，Amazon Redshift Serverless 資料庫中的所有資料都會還原到較早的時間點。如果您需要延長復原點的保留期間，則也可以從復原點建立快照。如需詳細資訊，請參閱快照和復原點。
基本 RPU 容量	您能以 Redshift 處理單元 (RPU) 為單位來設定基本容量。一個 RPU 可提供 16 GB 的記憶體。此設定可讓您控制使用中資源與工作負載成本之間的平衡。您可以增加此值以增加可用資源並改善查詢效能，或降低此值以限制支出。預設值是 128 個 RPU。您也可以設定用量限制 (例如每天使用的 RPU) 以控制成本。如需詳細資訊，請參閱Amazon Redshift Serverless 的帳單。
資料共用的用量限制	您可以使用主控台或 API 來限制從生產者區域傳輸到取用者區域的資料量。這些資料傳輸成本會因 而異 AWS 區域，並以 TB 為單位。如需資料共用的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的開始使用主控台來進行資料共用。
使用者定義的函數 (UDF)	您可以在 Amazon Redshift Serverless 中執行使用者定義的函數 (UDF)。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的建立使用者定義的函數。
預存程序	您可以在 Amazon Redshift Serverless 中執行預存程序。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的建立預存程序。
具體化視觀表	您可以在 Amazon Redshift Serverless 中建立具體化視觀表。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的建立具體化視觀表。

功能	說明
空間函數	您可以在 Amazon Redshift Serverless 中執行空間函數。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的查詢空間資料。
聯合查詢	您可以從 Amazon Redshift Serverless 執行查詢，將資料與 Aurora 資料庫叢集和 Amazon RDS 資料庫聯結。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的使用聯合查詢來查詢資料。
資料湖查詢	您可以執行查詢，將來自 Amazon S3 資料湖的資料與 Amazon Redshift Serverless 聯結。如需詳細資訊，請參閱《Amazon Redshift 管理指南》中的查詢資料湖。
HyperLogLog	您可以在 Amazon Redshift Serverless 中執行 HyperLogLog 函數。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的使用 HyperLogLog 草圖。
跨資料庫查詢資料	您可以使用 Amazon Redshift Serverless 來跨資料庫查詢資料。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的跨資料庫查詢資料。
資料共用	您可以使用 Amazon Redshift Serverless 存取所佈建資料倉儲上的資料共用。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的跨叢集共用資料。
半結構化資料查詢	您可以使用 Amazon Redshift Serverless 擷取和儲存 SUPER 資料類型的半結構化資料。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的擷取和查詢半結構化資料。
標記資源	您可以使用 AWS CLI 或 Amazon Redshift Serverless API，以與資源相關的中繼資料標記資源。如需詳細資訊，請參閱標記資源。
機器學習	您可以將 Amazon Redshift 機器學習與 Amazon Redshift Serverless 搭配使用。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的使用機器學習。
SQL 命令和函數	除了一些例外狀況 (例如 `REBOOT_CLUSTER`)，您可以將 Amazon Redshift SQL 命令和函數與 Amazon Redshift Serverless 搭配使用。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SQL 參考。

功能	說明
CloudFormation 資源	您可以使用 CloudFormation 範本來部署和更新 Amazon Redshift Serverless 資源。這項整合意味著您可以減少管理資源的時間，並專注於應用程式。如需 Amazon Redshift Serverless 中 CloudFormation 資源的相關資訊，請參閱 Amazon Redshift Serverless 資源類型參考。
CloudTrail 資源	Amazon Redshift Serverless 已與 AWS CloudTrail 整合，以提供在 Amazon Redshift Serverless 中採取之動作的記錄。CloudTrail 會將 Amazon Redshift Serverless 的所有 API 呼叫擷取為事件。如需詳細資訊，請參閱 Amazon Redshift Serverless 的 CloudTrail。

Amazon Redshift 佈建叢集概觀

Amazon Redshift 服務會管理設定、操作和擴展資料倉儲的所有工作。這些工作包括佈建容量、監控和備份叢集，以及對 Amazon Redshift 引擎套用修補程式和升級。

以下影片說明如何使用 Amazon Redshift 查詢編輯器 v2 建立叢集和查詢資料。

叢集管理

Amazon Redshift 叢集是一組節點，其中包含一個領導者節點和一或多個運算節點。您需要的運算節點的類型和數量取決於您的資料的大小、將執行的查詢數量，以及您需要的查詢執行時間效能。

建立和管理叢集

取決於您的資料資料倉儲需求，您可以從小型的單一節點叢集開始，並隨著您的需求變更，輕鬆向上擴展至更大型的多節點叢集。您可以對叢集新增或移除運算節點而不會中斷服務。如需詳細資訊，請參閱Amazon Redshift 佈建叢集。

預留運算節點

如果想要讓叢集執行一年以上，可以透過預留一年或三年期間的運算節點來節省金錢。相較於隨需佈建運算節點時的每小時費率，預留運算節點可大幅節省成本。如需詳細資訊，請參閱預留節點。

建立叢集快照

快照是叢集的時間點備份。有兩種類型的快照：自動和手動。Amazon Redshift 會透過使用加密的 Secure Sockets Layer (SSL) 連線，將這些快照儲存在 Amazon Simple Storage Service (Amazon S3)

內部。如果您必須從快照還原，Amazon Redshift 會建立新的叢集並從您指定的快照匯入資料。如需快照的相關資訊，請參閱 Amazon Redshift 快照和備份。

叢集存取和安全

有數個功能與 Amazon Redshift 中的叢集存取和安全性相關。這些功能有助於控制對您的叢集的存取、定義連線規則，以及加密資料與連線。這些是 Amazon Redshift 中的資料庫存取和安全性相關功能以外的功能。如需資料庫安全的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的管理資料庫安全。

AWS 帳戶和 IAM 登入資料

根據預設，Amazon Redshift 叢集只能由建立叢集 AWS 的帳戶存取。叢集會遭鎖定以防止他人存取。在您的 AWS 帳戶中，您可以使用 AWS Identity and Access Management (IAM) 服務來建立使用者帳戶，並管理這些帳戶的許可，以控制叢集操作。如需詳細資訊，請參閱Amazon Redshift 中的安全。如需如何管理 IAM 身分的相關資訊 (包括 IAM 角色的指引和最佳實務)，請參閱 Amazon Redshift 中的身分和存取管理。

Security groups (安全群組)

根據預設，您建立的叢集會對所有人關閉。IAM 憑證只能控制對 Amazon Redshift API 相關資源的存取：Amazon Redshift 主控台、命令列介面 (CLI)、API 和 SDK。若要啟用透過 JDBC 或 ODBC 從 SQL 用戶端工具對叢集的存取，您可以使用安全群組：

- 如果對您的 Amazon Redshift 叢集使用 EC2-VPC 平台，您必須使用 VPC 安全群組。建議您在 EC2-VPC 平台中啟動叢集。

  叢集若已搭配 EC2-Classic 完成啟動，則您無法將該叢集移動至 VPC。不過，您可以使用 Amazon Redshift 主控台將 EC2-Classic 快照還原至 EC2-VPC 叢集。如需詳細資訊，請參閱從快照還原叢集。
- 如果您對您的 Amazon Redshift 叢集使用 EC2-Classic 平台，則必須使用 Amazon Redshift 安全群組。

在任一情況下，如果您的 SQL 用戶端在 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上執行，則可以新增規則至安全群組，以授予明確的輸入存取給特定範圍的 CIDR/IP 地址或 Amazon EC2 安全群組。如需詳細資訊，請參閱Amazon Redshift 安全群組。

除了輸入存取規則，您可以建立資料庫使用者來提供登入資料，向叢集本身的資料庫進行驗證。如需詳細資訊，請參閱此主題中的資料庫。

加密

佈建叢集時，您可以選擇性地加密叢集來獲得額外的安全性。啟用加密時，Amazon Redshift 會將所有資料以加密格式儲存在使用者建立的資料表中。您可以使用 AWS Key Management Service (AWS KMS) 來管理您的 Amazon Redshift 加密金鑰。

加密是叢集不可變的屬性。從加密叢集切換至未加密叢集的唯一方式，是卸載資料，再將資料重新載入至新的叢集。加密會套用至叢集和任何備份。從加密的快照還原叢集時，新的叢集也會加密。

如需加密、金鑰和硬體安全性模組的相關資訊，請參閱Amazon Redshift 資料庫加密。

SSL 連線

您可以使用 Secure Sockets Layer (SSL) 加密來加密您的 SQL 用戶端和您的叢集之間的連線。如需詳細資訊，請參閱設定連線的安全選項。

監控叢集

有數個功能與 Amazon Redshift 中的監控相關。您可使用資料庫稽核記錄產生活動紀錄、設定事件和通知訂閱，以追蹤感興趣的資訊。使用 Amazon Redshift 和 Amazon CloudWatch 中的指標，了解叢集和資料庫的運作狀態和效能。

資料庫稽核記錄日誌

您可以使用資料庫稽核記錄功能來追蹤身分驗證嘗試、連線、中斷連線、對資料庫使用者定義的變更，以及資料庫中所執行查詢的相關資訊。此資訊對於 Amazon Redshift 中的安全性和故障診斷目的很有幫助。日誌會儲存在 Amazon S3 儲存貯體中。如需詳細資訊，請參閱資料庫稽核記錄日誌。

事件和通知

Amazon Redshift 會追蹤事件，並在 AWS 您的帳戶中保留相關資訊數週。對於每個事件，Amazon Redshift 會報告相關的資訊，如事件發生的日期、描述、事件來源 (例如，叢集、參數群組或快照)，以及來源 ID。您可以建立指定一組事件篩選條件的 Amazon Redshift 事件通知訂閱。發生符合篩選條件的事件時，Amazon Redshift 便會使用 Amazon Simple Notification Service 來主動通知您已發生事件。如需事件和通知的相關資訊，請參閱Amazon Redshift 事件。

效能

Amazon Redshift 提供您可以追蹤叢集和資料庫之運作狀態和效能的效能指標和資料。Amazon Redshift 會使用 Amazon CloudWatch 指標來監控叢集的實體層面，例如 CPU 利用率、延遲和輸送

量。Amazon Redshift 也會提供查詢和載入的效能資料，以協助您監控叢集中的資料庫活動。如需效能指標和監控的相關資訊，請參閱監控 Amazon Redshift 叢集效能。

資料庫

Amazon Redshift 會在您佈建叢集時建立一個資料庫。這是您用來載入資料並對資料執行查詢的資料庫。您可以視需要執行 SQL 命令來建立額外的資料庫。如需建立額外資料庫的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的步驟 1：建立資料庫。

佈建叢集時，您會指定管理員使用者，其會具有叢集內所建立一切資料庫的存取權。這個管理員使用者是超級使用者，他是最初可存取資料庫的唯一使用者，不過這個使用者可以建立其他的超級使用者和使用者。如需詳細資訊，請前往《Amazon Redshift 資料庫開發人員指南》中的超級使用者和使用者。

Amazon Redshift 使用參數群組來定義叢集中所有資料庫的行為，例如日期呈現樣式和浮點精確度。如果佈建叢集時不指定參數群組，Amazon Redshift 會將叢集與預設的參數群組建立關聯。如需詳細資訊，請參閱Amazon Redshift 參數群組。

如需 Amazon Redshift 中資料庫的相關資訊，請前往《Amazon Redshift 資料庫開發人員指南》https://docs.aws.amazon.com/redshift/latest/dg/。

將 Amazon Redshift Serverless 與 Amazon Redshift 佈建資料倉儲進行比較

Amazon Redshift Serverless 的某些概念和功能，和其所對應的 Amazon Redshift 佈建資料倉儲功能不同。例如，一個差異極大的比較是 Amazon Redshift Serverless 沒有叢集或節點的概念。下表描述 Amazon Redshift Serverless 中的功能和行為，並說明它們與佈建資料倉儲中的同等功能有何不同。

功能	說明	無伺服器	佈建
工作群組和命名空間	若要在 Amazon Redshift Serverless 中隔離工作負載並管理不同的資源，您可以建立命名空	命名空間是資料庫物件和使用者的集合。工作群組是運算資源的集合。如需詳細資訊，請參	佈建叢集是您直接管理的運算節點和領導節點的集合。如需詳細資訊，請參閱Amazon Redshift 佈建叢集。

功能	說明	無伺服器	佈建
	間和工作群組，以便將儲存和運算資源分開管理。	閱 Amazon Redshift Serverless 以了解 Amazon Redshift Serverless 的設計。	

功能	說明	無伺服器	佈建
節點類型	當您使用 Amazon Redshift Serverless 時，不會像使用 Amazon Redshift 佈建叢集那樣地選擇節點類型或指定節點計數。	Amazon Redshift Serverless 會自動為您佈建和管理容量。您可以選擇性地指定基礎資料倉儲容量，以便為您的工作負載選擇合適的性價比。您也可以指定最大 RPU 時數來設定成本控制，以確保成本是可預測的。如需詳細資訊，請參閱Amazon Redshift Serverless 的運算容量。	您可以使用符合成本和效能規格的節點類型來建置叢集。如需詳細資訊，請參閱Amazon Redshift 佈建叢集。

功能	說明	無伺服器	佈建
工作負載管理和並行擴展	Amazon Redshift 可以在負載繁重時進行擴展。Amazon Redshift Serverless 也可以進行擴展以滿足間歇性高負載時段。	Amazon Redshift Serverless 會在成本控制的閾值內，自動根據工作負載有效率地管理資源並進行擴展。如需詳細資訊，請參閱運算容量的帳單。	使用佈建資料倉儲時，您可以在叢集上啟用並行擴展，以應付負載繁重的期間。如需詳細資訊，請參閱並行擴展。
連接埠	您用來連線的連接埠號碼。	使用 Amazon Redshift Serverless 時，您可以變更為連接埠範圍 5431 至 5455 或 8191 至 8215 內的另一個連接埠。如需詳細資訊，請參閱連線至 Amazon Redshift Serverless。	使用佈建資料倉儲時，您可以選擇任何連接埠來連線。

功能	說明	無伺服器	佈建
調整大小	新增或移除運算資源，以便有良好的工作負載執行效能。	調整大小不適用於 Amazon Redshift Serverless。不過，您可以根據您的價格和效能要求，變更基礎的資料倉儲 RPU 容量。如需詳細資訊，請參閱Amazon Redshift Serverless 的運算容量。	使用佈建叢集時，您可以執行叢集調整大小以新增節點或移除節點。如需詳細資訊，請參閱在 Amazon Redshift 中管理叢集的概觀。
暫停和繼續	您可以在沒有要執行的工作負載時暫停佈建叢集，以節省成本。	使用 Amazon Redshift Serverless 時，您只需就執行的查詢付費，因此無需暫停或繼續。如需詳細資訊，請參閱運算容量的帳單。	您根據不同時間的工作負載評估，手動暫停和繼續叢集。如需詳細資訊，請參閱在 Amazon Redshift 中管理叢集的概觀。

功能	說明	無伺服器	佈建
查詢外部資料	您可以在 Amazon S3 儲存貯體中查詢各種格式的資料，例如 JSON、Parquet 或 Iceberg。	Amazon Redshift Serverless 包含整合式資料湖查詢引擎，可在工作群組的運算資源上執行，並支援在 AWS 區域內或不同 AWS 區域之間查詢 Amazon S3 資料。若要調整資料湖查詢效能，請調整工作群組的基本 RPU 容量。	使用 RA3 和 DC2 佈建叢集時，資料湖查詢會使用獨立於叢集的專用伺服器在 Redshift Spectrum 上執行。Redshift Spectrum 要求 Amazon S3 儲存貯體與叢集位於相同的 AWS 區域。如需詳細資訊，請參閱使用 Amazon Redshift Spectrum 查詢外部資料。 RG 佈建叢集包含整合式資料湖查詢引擎。資料湖查詢直接在叢集自己的運算資源上執行，類似於 Redshift Serverless，並支援在相同 AWS 區域或不同 AWS 區域中查詢 Amazon S3 資料。如需詳細資訊，請參閱查詢您的資料湖。

功能	說明	無伺服器	佈建
運算資源計費	Amazon Redshift 與 Amazon Redshift Serverless 的計費方式。	使用 Amazon Redshift Serverless 時，您只需就所執行的工作負載支付費用，計費基礎為每秒 RPU 小時，最低費用為 60 秒。這包括在 Amazon S3 中存取開放檔案格式資料的查詢。如需詳細資訊，請參閱運算容量的帳單。	使用佈建叢集時，如果叢集未暫停，則會按秒計費。對於 RG 叢集，資料湖查詢包含在節點小時運算定價中，無需另外支付 Spectrum 費用。對於 RA3 和 DC2 叢集，資料湖查詢會產生額外的 Redshift Spectrum 費用。

功能	說明	無伺服器	佈建
Maintenance window (維護時段)	伺服器維護的運作方式。	使用 Amazon Redshift Serverless 時，不會有維護時段。系統會以無縫方式處理更新。如需詳細資訊，請參閱什麼是 Amazon Redshift Serverless？	使用佈建叢集時，則會在進行修補時指定維護時段。(一般來說，您可以選擇低使用量的週期性時間。)
加密	您可以啟用資料庫加密。	Amazon Redshift Serverless 一律使用 AWS KMS 受 AWS 管金鑰或客戶受管金鑰加密。	佈建資料倉儲中的資料可以使用 AWS KMS （使用 AWS 受管金鑰或客戶受管金鑰） 加密或未加密。請參閱 Amazon Redshift 資料庫加密。

功能	說明	無伺服器	佈建
儲存計費	儲存的計費方式。	若為 Amazon Redshift Serverless。會根據每月 GB 數來計算費率。請參閱運算容量的帳單。	除了具有 RG 或 RA3 節點的佈建叢集的運算資源之外，還會對儲存體計費。

功能	說明	無伺服器	佈建
使用者管理	管理使用者的方式。	對於 Amazon Redshift Serverless，使用者是 IAM 或 Redshift 使用者。如需詳細資訊，請參閱Amazon Redshift Serverless 中的身分和存取管理。 如需如何管理 IAM 身分的相關資訊 (包括 IAM 角色的最佳實務)，請參閱 Amazon Redshift 中的身分和存取管理。	對於佈建資料倉儲，使用者是 IAM 或 Redshift 使用者。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的管理資料庫安全。 如需如何管理 IAM 身分的相關資訊 (包括 IAM 角色的最佳實務)，請參閱 Amazon Redshift 中的身分和存取管理。

功能	說明	無伺服器	佈建
JDBC 和 ODBC 工具與相容性	用戶端連線的運作方式。	Amazon Redshift Serverless 與任何 JDBC 或 ODBC 合規工具或用戶端應用程式相容。如需驅動程式的相關資訊，請參閱《Amazon Redshift 管理指南》中的設定連線。如需有關連線至 Amazon Redshift Serverless 的資訊，請參閱連線至 Redshift Serverless。	Amazon Redshift 佈建與任何 JDBC 或 ODBC 合規工具或用戶端應用程式相容。如需驅動程式的相關資訊，請參閱《Amazon Redshift 管理指南》中的設定連線。如需有關連線至叢集的資訊，請參閱使用 SQL 用戶端工具連線至 Amazon Redshift 資料倉儲。

功能	說明	無伺服器	佈建
登入時的憑證要求	憑證的處理方式。	若為 Amazon Redshift Serverless，您不必在每個執行個體中輸入憑證。如需詳細資訊，請參閱連線至 Amazon Redshift Serverless。	想要存取 Amazon Redshift 就需要與 IAM 角色相關聯的使用者所提供的登入憑證。IAM 角色會針對佈建資料倉儲附加特定許可。經過驗證之後，使用者就可以直接連線到資料庫、Redshift 主控台，以及查詢編輯器 v2。

功能	說明	無伺服器	佈建
Data API (資料 API)	您可以從 Web 服務和其他應用程式存取資料。	Amazon Redshift Serverless 支援 Amazon Redshift 資料 API。使用 Amazon Redshift Serverless 時，您會使用 `workgroup-name` 參數而不是 `cluster-identity` 參數。如需呼叫資料 API 的相關資訊，請參閱使用 Amazon Redshift 資料 API。	Amazon Redshift 佈建支援 Amazon Redshift 資料 API。使用 Amazon Redshift 叢集時，您會使用 `cluster-identity` 參數而不是 `workgroup-name` 參數。如需呼叫資料 API 的相關資訊，請參閱使用 Amazon Redshift 資料 API。

功能	說明	無伺服器	佈建
快照	提供時間點復原。	Amazon Redshift Serverless 支援快照和復原點。如需命名空間的快照和復原點的相關資訊，請參閱快照和復原點。	佈建叢集支援快照。如需詳細資訊，請參閱使用主控台管理快照。
資料共用	提供在相同帳戶或不同帳戶中的資料庫之間共用資料的能力。	Amazon Redshift Serverless 支援佈建資料倉儲所具備的所有資料共用功能。其也可在 Amazon Redshift Serverless 和佈建資料倉儲、工具或用戶端應用程式之間支援資料共用。	佈建的叢集支援跨資料庫、跨帳戶、跨區域和 AWS Data Exchange 資料共用。如需詳細資訊，請參閱在 Amazon Redshift 中跨叢集共用資料。

功能	說明	無伺服器	佈建
追蹤	提供軟體更新的排程。	Amazon Redshift Serverless 沒有追蹤的概念。版本和更新會由服務處理。如需 Amazon Redshift Serverless 設計的相關資訊，請參閱快照和復原點。	佈建叢集支援在目前的追蹤和隨後的追蹤之間進行切換。

功能	說明	無伺服器	佈建
系統資料表和檢視	提供監控資源和系統中繼資料的方法。	Amazon Redshift Serverless 支援新的系統資料表和檢視。如需系統資料表的相關資訊，請參閱使用 Amazon Redshift Serverless 監控查詢和工作負載。如需如何將查詢從使用舊版的佈建系統資料表和檢視遷移至新檢視的相關資訊，請參閱遷移至 SYS 監控檢視。	佈建資料倉儲支援一組現有的系統資料表和檢視，可用於監控以及其他需要系統中繼資料的任務。

功能	說明	無伺服器	佈建
參數群組	這是套用至叢集中所建立所有資料庫的一組參數。這些參數用於設定查詢逾時和日期樣式等資料庫設定。	Amazon Redshift Serverless 沒有參數群組的概念。	佈建資料倉儲支援參數群組。如需佈建叢集之參數群組的相關資訊，請參閱 Amazon Redshift 參數群組。

功能	說明	無伺服器	佈建
查詢監控	提供以時間為基礎的查詢執行檢視。	Amazon Redshift Serverless 中的查詢監控需要使用者連線到資料庫才能使用系統資料表。如此一來，查詢監控和系統資料表才會同步。Amazon Redshift Serverless 中的系統資料表查詢會使用映射至 IAM 使用者的資料庫使用者來使用查詢監控。如需監控查詢的相關資訊，請參閱使用 Amazon Redshift Serverless 監控查詢和工作負載。	佈建叢集中的查詢監控不會顯示系統資料表中的所有資料。

功能	說明	無伺服器	佈建
稽核記錄	提供資料庫中連線和使用者活動的相關資訊。	使用 Amazon Redshift Serverless 時，CloudWatch 是稽核日誌的目的地。Amazon Redshift Serverless 不支援以 Amazon S3 為基礎的稽核日誌交付。如需詳細資訊，請參閱 Amazon Redshift Serverless 的稽核記錄。	若為佈建叢集，以 Amazon S3 為基礎的稽核日誌交付已成為常態。現在，將稽核記錄傳遞至 CloudWatch 的行為已擴大為涵蓋佈建資料倉儲。

功能	說明	無伺服器	佈建
事件通知	Amazon EventBridge 是無伺服器事件匯流排服務，可讓您用於將應用程式與來自各種來源的事件資料互相連線。	Amazon Redshift Serverless 會使用 Amazon EventBridge 來管理事件通知，讓您隨時掌握關於資料倉儲變化的最新資訊。如需詳細資訊，請參閱使用 Amazon EventBridge 的 Amazon Redshift Serverless 事件通知。	若為佈建叢集，您會使用 Amazon Redshift 主控台建立事件訂閱，以管理事件通知。如需詳細資訊，請參閱建立事件通知訂閱。
游標限制條件	Amazon Redshift 會對所有游標結果集的大小強制執行限制條件。	Amazon Redshift Serverless 的資料指標結果集大小總計上限為 150,000 MB。	對於佈建叢集，游標結果集大小總計上限取決於叢集類型。如需詳細資訊，請參閱游標限制條件。

使用佈建叢集的 Amazon Redshift 管理介面

> **Note**
>
> 本主題主要說明佈建叢集的 Amazon Redshift 管理介面。Amazon Redshift Serverless 和 Amazon Redshift Data API 有類似的管理介面。

Amazon Redshift 支援數個管理介面，可用來建立、管理和刪除 Amazon Redshift 叢集： AWS SDKs、 AWS Command Line Interface (AWS CLI) 和 Amazon Redshift 管理 API。

Amazon Redshift API — 您可以透過提交請求來呼叫此 Amazon Redshift 管理 API。請求是使用 HTTP 動詞 `GET` 或 `POST` 及 `Action` 參數的 HTTP 或 HTTPS 請求。呼叫 Amazon Redshift API 是存取 Amazon Redshift 服務最直接的方式。不過，這需要您的應用程式處理低階詳細資訊，例如錯誤處理和產生雜湊以簽署請求。

- 如需有關建置與簽署 Amazon Redshift API 請求的詳細資訊，請參閱簽署 HTTP 請求。
- 如需有關 Amazon Redshift API 動作和 Amazon Redshift 資料類型的資訊，請參閱 Amazon Redshift API 參考。

AWS SDKs – 您可以使用 AWS SDKs來執行 Amazon Redshift 叢集相關操作。多種軟體開發套件程式庫包裝了底層的 Amazon Redshift API。它們整合了 API 功能到特定的程式設計語言，並處理許多低階詳細資訊，例如計算簽章、處理請求重試和錯誤處理。在軟體開發套件程式庫中呼叫包裝器函式，可以大幅簡化撰寫 Amazon Redshift 叢集管理應用程式的流程。

- Java、.NET、PHP、Python、Ruby 和 Node.js AWS SDKs 支援 Amazon Redshift。Amazon Redshift 包裝器函式記錄在每個軟體開發套件的參考手冊中。如需 AWS SDKs 的清單及其文件的連結，請參閱適用於 Amazon Web Services 的工具。
- 本指南提供透過 Java 開發套件使用 Amazon Redshift 的範例。如需更一般的 AWS SDK 程式碼範例，請參閱 Amazon Redshift AWS SDKs的程式碼範例。

AWS CLI – CLI 提供一組命令列工具，可用來管理 Windows、Mac 和 Linux 電腦 AWS 的服務。 AWS CLI 包含以 Amazon Redshift API 動作為基礎的命令。

- 如需有關安裝和設定 Amazon Redshift CLI 的資訊，請參閱設定 Amazon Redshift CLI。
- 如需有關 Amazon Redshift CLI 命令的參考資料，請參閱《AWS CLI 參考》中的 Amazon Redshift。

搭配 AWS SDK 使用此服務

AWS 軟體開發套件 (SDKs) 適用於許多熱門的程式設計語言。每個 SDK 都提供 API、程式碼範例和說明文件，讓開發人員能夠更輕鬆地以偏好的語言建置應用程式。

SDK 文件	代碼範例
適用於 C++ 的 AWS SDK	適用於 C++ 的 AWS SDK 程式碼範例
AWS CLI	AWS CLI 程式碼範例
適用於 Go 的 AWS SDK	適用於 Go 的 AWS SDK 程式碼範例
適用於 Java 的 AWS SDK	適用於 Java 的 AWS SDK 程式碼範例
適用於 JavaScript 的 AWS SDK	適用於 JavaScript 的 AWS SDK 程式碼範例
適用於 Kotlin 的 AWS SDK	適用於 Kotlin 的 AWS SDK 程式碼範例
適用於 .NET 的 AWS SDK	適用於 .NET 的 AWS SDK 程式碼範例
適用於 PHP 的 AWS SDK	適用於 PHP 的 AWS SDK 程式碼範例
AWS Tools for PowerShell	AWS Tools for PowerShell 程式碼範例
AWS 適用於 Python v4 的 SDK	AWS 適用於 Python v4 的 SDK 程式碼範例
適用於 Python (Boto3) 的 AWS SDK	適用於 Python (Boto3) 的 AWS SDK 程式碼範例
適用於 Ruby 的 AWS SDK	適用於 Ruby 的 AWS SDK 程式碼範例
適用於 Rust 的 AWS SDK	適用於 Rust 的 AWS SDK 程式碼範例
適用於 SAP ABAP 的 AWS SDK	適用於 SAP ABAP 的 AWS SDK 程式碼範例
適用於 Swift 的 AWS SDK	適用於 Swift 的 AWS SDK 程式碼範例

可用性範例

找不到所需的內容嗎？ 請使用本頁面底部的提供意見回饋連結申請程式碼範例。

簽署 HTTP 請求

Amazon Redshift 要求您傳送至管理 API 的每個請求都必須經過簽章驗證。本主題將說明如何簽署請求。

如果您使用的是其中一個 AWS 軟體開發套件 (SDKs) 或 AWS Command Line Interface，系統會自動處理請求簽署，您可以略過本節。如需使用 AWS SDKs的詳細資訊，請參閱 使用佈建叢集的 Amazon Redshift 管理介面。如需使用 Amazon Redshift 命令列界面的相關資訊，請前往 Amazon Redshift 命令列參考。

若要簽署請求，請使用加密雜湊函數來計算數位簽章。加密雜湊是一個函數，其根據輸入傳回一個唯一的雜湊值。此雜湊函數的輸入包含請求和私密存取金鑰 (可從暫時憑證取得) 的文字。雜湊函數會傳回一個雜湊值，您將此值包含在請求中做為簽章。該簽章是請求 `Authorization` 標頭中的一部分。

Note

如果使用者想要與 AWS 外部互動，則需要程式設計存取 AWS 管理主控台。授予程式設計存取權的方式取決於正在存取的使用者類型 AWS。
若要授予使用者程式設計存取權，請選擇下列其中一個選項。

哪個使用者需要程式設計存取權？	到	根據
IAM	（建議）使用主控台登入資料做為臨時登入資料，以簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《使用者指南》中的登入以進行 AWS 本機開發。 • AWS SDKs，請參閱 AWS SDKs 和工具參考指

哪個使用者需要程式設計存取權？	到	根據
		南中的登入以進行 AWS 本機開發。
人力資源身分 (IAM Identity Center 中管理的使用者)	使用暫時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的設定 AWS CLI 要使用 AWS IAM Identity Center的 。 • AWS SDKs、工具和 AWS APIs，請參閱 AWS SDK 和工具參考指南中的 SDKsIAM Identity Center 身分驗證。
IAM	使用暫時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	遵循《IAM 使用者指南》中將臨時登入資料與 AWS 資源搭配使用的指示。

哪個使用者需要程式設計存取權？	到	根據
IAM	(不建議使用) 使用長期登入資料來簽署對 AWS CLI、 AWS SDKs 或 AWS APIs程式設計請求。	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的使用 IAM 使用者憑證進行身分驗證。 • AWS SDKs和工具，請參閱 AWS SDKs和工具參考指南中的使用長期憑證進行身分驗證。 • 對於 AWS APIs，請參閱《IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。

Amazon Redshift 收到請求後，其會使用您原先用來簽署請求的相同雜湊函數與輸入來重新計算簽章。若產生的簽章與請求中的簽章相符，Amazon Redshift 會處理該請求，否則請求會遭到拒絕。

Amazon Redshift 支援使用 AWS Signature 第 4 版進行身分驗證。計算簽章的程序由三個任務組成。這些任務如下圖說明。

- 任務 1：建立標準請求

  將 HTTP 請求重新編排為正式格式。必須使用正式格式，因為 Amazon Redshift 在比較您傳送的簽章來建立簽章時會使用相同的正式格式。

- 任務 2：建立登入字串

  建立一個字串，您會使用此字串做為密碼編譯雜湊函數的其中一個輸入值。此字串，稱為登入字串，是雜湊演算法的名稱、請求日期、登入資料範圍字串和前一個任務的正式請求的串連。登入資料範圍字串本身是日期、區域和服務資訊的串連。

- 任務 3：計算簽章

使用接受兩個輸入字串的密碼編譯雜湊函數來計算請求的簽章：您的登入字串和衍生金鑰。藉由從您的私密存取金鑰開始來計算此衍生金鑰和使用登入資料範圍字串來建立一系列雜湊型訊息身分驗證代碼 (HMAC-SHA256)。

範例簽章計算

下列範例會逐步解說如何建立 CreateCluster 請求之簽章的詳細資訊。您可以使用此範例做為檢查自身簽章計算方法的參考。其他參考計算包含在《IAM 使用者指南》的請求簽章範例章節中。

您可以使用 GET 或 POST 請求，將請求傳送至 Amazon Redshift。此兩者的差異在於，若是 GET 要求，會以查詢字串參數的形式來傳送您的參數。若是 POST 請求，它們會包含在請求的內文中。下列範例說明 POST 請求。

該範例假設如下：

- 請求的時間戳記為 `Fri, 07 Dec 2012 00:00:00 GMT`。
- 端點是美國東部 (維吉尼亞北部) 區域 (`us-east-1`)。

一般的請求語法是：

```
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
```

對於任務1：建立正式請求計算的請求的正式形式是：

```
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
```

正式請求的最後一行是請求內文的雜湊值。正式請求中的第三行是空的，因為此 API 沒有查詢參數。

「任務 2：建立登入字串」的登入字串是：

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

登入字串的第一行是演算法，第二行是時間戳記，第三行是登入資料範圍，最後一行是來自任務 1：建立正式請求的正式請求的雜湊。在憑證範圍內使用的服務名稱是 `redshift`。

對於任務 3：計算簽章，衍生金鑰可以表示為：

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

衍生金鑰是以湊雜函數的方式計算。從上述公式中的內部 HMAC 陳述式開始，您可以串連片語 **AWS4** 與私密存取金鑰並使用此做為金鑰來對資料 "us-east-1" 進行雜湊。此雜湊的結果都會變成下一個雜湊函數的金鑰。

在計算衍生金鑰後，您會在雜湊函數中使用它，該函數會接受兩個輸入字串：您的登入字串和衍生金鑰。例如，如果您使用私密存取金鑰 `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY` 和先前提供的登入字串，則已計算的簽章如下：

```
9a6b557aa9f38dea83d9215d8f0eae54100877f3e0735d38498d7ae489117920
```

最後步驟是建立 `Authorization` 標頭。對於示範存取金鑰 `AKIAIOSFODNN7EXAMPLE`，標頭 (為了可讀性而新增了換行) 是：

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20121207/us-east-1/
redshift/aws4_request,
SignedHeaders=content-type;host;x-amz-date,
Signature=9a6b557aa9f38dea83d9215d8f0eae54100877f3e0735d38498d7ae489117920
```

設定 Amazon Redshift CLI

本節說明如何設定和執行 AWS CLI 命令列工具，以用於管理 Amazon Redshift。Amazon Redshift 命令列工具會在 AWS Command Line Interface (AWS CLI) 上執行，而後者又使用 Python (https://www.python.org/：//)。您可以在支援 Python 的任何作業系統 AWS CLI 上執行 。

安裝 AWS Command Line Interface

若要開始使用 Amazon Redshift 命令列工具，請先設定 AWS CLI，然後新增定義 Amazon Redshift CLI 選項的組態檔案。

如果您已經 AWS CLI 為其他服務安裝並設定 AWS ，您可以略過此程序。

安裝 AWS Command Line Interface

1. 前往安裝或更新至最新版本的 AWS CLI，然後遵循安裝 的指示 AWS CLI。

   對於 CLI 存取，您需要存取金鑰 ID 和私密存取金鑰。盡可能使用臨時憑證，而不是長期存取金鑰。臨時憑證包含存取金鑰 ID、私密存取金鑰，以及指出憑證何時到期的安全符記。如需詳細資訊，請參閱《IAM 使用者指南》中的將臨時登入資料與 AWS 資源搭配使用。
2. 建立包含組態資訊 (例如您的存取金鑰、預設區域和命令輸出格式) 的檔案。然後設定 `AWS_CONFIG_FILE` 環境變數來參考該檔案。如需詳細說明，請參閱 AWS Command Line Interface 《 使用者指南》中的設定 AWS 命令列界面。
3. 執行測試命令以確認 AWS CLI 介面是否正常運作。例如，下列命令應會顯示 AWS CLI協助資訊：

   ```
   aws help
   ```

   下列命令應會顯示 Amazon Redshift 協助資訊：

   ```
   aws redshift help
   ```

如需有關 Amazon Redshift CLI 命令的參考資料，請前往《 AWS CLI 參考》中的 Amazon Redshift。

Amazon Redshift Serverless

Amazon Redshift Serverless 方便您執行和擴展分析，而不必佈建和管理內部部署資料倉儲。透過 Amazon Redshift Serverless，資料分析師、開發人員和資料科學家現在可以使用 Amazon Redshift 將資料載入到雲端中的資料倉儲以及從當中查詢記錄，在幾秒鐘內從資料中獲得深入分析。Amazon Redshift 會自動佈建和擴展資料倉儲容量，為嚴苛且無法預測的工作負載提供快速的效能。您只需支付實際使用容量的費用。不必變更現有的分析和商業智慧應用程式，就能透過這項簡易性獲得好處。

如需 Amazon Redshift Serverless SLA 的相關資訊，請參閱 Amazon Redshift 服務水準協議。

什麼是 Amazon Redshift Serverless？

Amazon Redshift Serverless 會自動佈建資料倉儲容量，並有智慧地擴展基礎資源。Amazon Redshift Serverless 可在幾秒鐘內調整容量，即使面對最苛刻、最不穩定的工作負載，也能持續提供高效能和簡化的操作機制。

透過 Amazon Redshift Serverless，您可以從下列功能獲得好處：

- 存取和分析資料，而不必設定、微調和管理 Amazon Redshift 佈建的叢集。
- 使用優異的 Amazon Redshift SQL 功能、領先業界的效能和資料湖整合，以在資料倉儲、資料湖和操作資料來源之間無縫地進行查詢。
- 透過智慧型自動擴展功能，為最嚴苛、最不穩定的工作負載持續提供高效能和簡化的操作機制。
- 使用工作群組和命名空間，透過精細的成本控制來組織運算資源和資料。
- 僅在資料倉儲處於使用狀態時才需要付費。

透過 Amazon Redshift Serverless，您可以使用主控台介面連線至無伺服器資料倉儲或 API 以建置應用程式。透過資料倉儲，您可以存取 Amazon Redshift 受管儲存和 Amazon S3 資料湖。

此影片會向您展示 Amazon Redshift Serverless 如何讓您輕鬆地執行和擴展分析，而不必管理資料倉儲基礎設施：

Amazon Redshift Serverless 主控台

若要開始使用 Amazon Redshift Serverless 主控台，請觀看以下影片：開始使用 Amazon Redshift Serverless。

Serverless 儀表板

在 Serverless 儀表板頁面上，您可以檢視資源摘要和用量圖表。

- 命名空間概觀 — 此區段會顯示命名空間內的快照和資料共用數量。
- 工作群組 — 此區段會顯示 Amazon Redshift Serverless 內的所有工作群組。
- 查詢指標 — 此區段會顯示過去 1 小時的查詢活動。
- 使用的 RPU 容量 — 此區段會顯示過去 1 小時使用的容量。
- 免費試用 – 本節顯示您 AWS 帳戶中剩餘的免費試用點數。這涵蓋了同一帳戶下 Amazon Redshift Serverless 資源和操作 (包括快照、儲存、工作群組等) 的所有用量。
- 警示 — 此區段會顯示您在 Amazon Redshift Serverless 中所設定的警示。

資料備份

在資料備份索引標籤上，您可以使用以下項目：

- 快照 — 您可以建立、刪除和管理 Amazon Redshift Serverless 資料的快照。預設保留期間為 `indefinitely`，但您可以將保留期間設定為 1 到 3653 天之間的任何值。您可以授權 從快照 AWS 帳戶 還原命名空間。
- 復原點 — 顯示系統所自動建立、可讓您從過去 24 小時內的意外寫入或刪除情況中復原的復原點。若要復原資料，您可以將復原點還原至任何可用的命名空間。如果您想要延長復原點的保留期間，則可以從復原點建立快照。預設保留期間為 `indefinitely`，但您可以將保留期間設定為 1 到 3653 天之間的任何值。

資料存取

在資料存取索引標籤上，您可以使用下列項目：

- 網路和安全設定 — 您可以檢視 VPC 相關的值、 AWS KMS 加密值和稽核記錄值。您只能更新稽核記錄。
- AWS KMS key — 用來加密 Amazon Redshift Serverless 中資源的 AWS KMS key 。
- 許可 — 您可以管理 Amazon Redshift Serverless 所能擔任以代表您使用資源的 IAM 角色。如需詳細資訊，請參閱Amazon Redshift Serverless 中的身分和存取管理。
- Redshift 受管 VPC 端點 — 您可以從其他 VPC 或子網路存取您的 Amazon Redshift Serverless 執行個體。如需詳細資訊，請參閱從其他 VPC 端點連線至 Amazon Redshift Serverless。

限制

在限制索引標籤上，您可以使用以下項目：

- 以 Redshift 處理單元 (RPU) 為單位的基本容量設定 — 您可以設定用來處理工作負載的基本容量。若要改善查詢效能，請增加您的 RPU 值。
- 用量限制 — Amazon Redshift Serverless 執行個體可在動作啟動前的期間內使用的運算資源上限。您可以限制 Amazon Redshift Serverless 用來執行工作負載的資源數量。用量的測量單位是 Redshift 處理單元 (RPU) 小時。RPU 小時是一小時內所使用的 RPU 數量。您可以決定在達到所設定的限制值時會發生的動作，如下所示：
  - 傳送提醒。
  - 在系統資料表中記錄項目。
  - 關閉使用者查詢。

  您最多可以設定四個限制。
- 查詢限制 — 您可以新增限制以監控效能和限制。如需查詢監控限制的相關資訊，請參閱 WLM 查詢監控規則。

如需詳細資訊，請參閱Amazon Redshift Serverless 的運算容量。

資料共用

在資料共用索引標籤上，您可以使用以下項目：

- 我的命名空間中建立的資料共用設定 — 您可以建立資料共用，並與其他命名空間和 AWS 帳戶共用。
- 其他命名空間和 AWS 帳戶的資料共用 — 您可以從其他命名空間和 AWS 帳戶的資料共用建立資料庫。

如需資料共用的相關資訊，請參閱 Amazon Redshift Serverless 中的資料共用。

查詢和資料庫監控

在查詢和資料庫監控頁面上，您可以檢視查詢歷史記錄和資料庫效能的圖表。

在查詢歷史記錄索引標籤上，您會看到下列圖表 (您可以在查詢清單和資源指標之間進行選擇)：

- 查詢執行期 — 此圖表會顯示在相同時間範圍內執行的查詢。選擇圖表中的長條可檢視更多的查詢執行詳細資訊。

- 查詢與載入 — 此區段會依查詢 ID 列出查詢和載入。
- 已使用的 RPU 容量 — 此圖表會顯示以 Redshift 處理單元 (RPU) 為單位的整體容量。
- 資料庫連線數 — 此圖表會顯示作用中資料庫連線的數目。

資料庫效能

在資料庫效能索引標籤上，您會看到下列圖表：

- 每秒完成的查詢數 — 此圖表會顯示每秒完成的平均查詢數目。
- 查詢持續時間 — 此圖表會顯示完成查詢的平均時間長度。
- 資料庫連線數 — 此圖表會顯示作用中資料庫連線的數目。
- 執行中的查詢數 — 此圖表會顯示給定時間的執行中查詢總數。
- 排入佇列的查詢數 — 此圖表會顯示給定時間的已排入佇列查詢總數。
- 查詢執行時間明細 — 此圖表會顯示依查詢類型的查詢所花執行時間總數。

資源監控

在資源監控頁面上，您可以檢視已取用資源的圖表。您可以根據數個面向來篩選資料。

- 指標篩選條件 — 您可以使用指標篩選條件來選取特定工作群組的篩選條件，以及選擇時間範圍和時間間隔。
- 已使用的 RPU 容量 — 此圖表會顯示以 Redshift 處理單元 (RPU) 為單位的整體容量。
- 運算用量 — 此圖表會顯示所選時間範圍內，依期間分類的 RPU 時數用量。對於少於 6 小時的時間範圍，RPU 時數會以精確的時間顯示。對於 6 小時或以上的時間範圍，RPU 時數會顯示為平均值。
- 自動最佳化收費秒數的額外運算 – 此圖表顯示針對所選時間範圍的自動資料庫最佳化收取的 RPU 秒數。當您的 Amazon Redshift 使用額外的運算資源來執行自動最佳化時，會向您收取費用。如需詳細資訊，請參閱為自動資料庫最佳化配置額外的運算資源。

資料共用

在資料共用頁面上，您可以管理在我的帳戶中和從其他帳戶資料共用。如需資料共用的相關資訊，請參閱 Amazon Redshift Serverless 中的資料共用。

AWS Glue Data Catalog

在 AWS Glue Data Catalog 索引標籤中，您可以檢視命名空間在 AWS Glue Data Catalog中的註冊狀態。此索引標籤在您開始註冊程序後才會顯示。如需向 註冊命名空間的詳細資訊 AWS Glue Data Catalog，請參閱《Amazon Redshift 資料庫開發人員指南》中的 Amazon Redshift 的 Apache Iceberg 相容性。

使用 Amazon Redshift Serverless 時的考量

如需 Amazon Redshift Serverless 可用 AWS 區域 位置的清單，請參閱《》中針對 Redshift Serverless API 列出的端點Amazon Web Services 一般參考。

Amazon Redshift Serverless 所使用的某些資源會受到配額限制。如需詳細資訊，請參閱Amazon Redshift Serverless 物件的配額。

當您 DECLARE (宣告) 游標時，系統會在 DECLARE 中指定 Amazon Redshift Serverless 的結果集大小規格。Amazon Redshift Serverless 的資料指標結果集大小總計上限為 150,000 MB。

線上修補 – Amazon Redshift Serverless 提供自動軟體更新，而不需要傳統的維護時段。有新的更新可用時，系統會在閒置期間於發行後 14 天內套用。更新程序通常最多需要 15 分鐘。如果 14 天內沒有閒置 15 分鐘，您的 Serverless 端點可能會短暫無法使用。在此期間，與端點的應用程式連線可能會失敗。您可以在「Amazon Redshift 叢集版本」文件中監控 Redshift 修補程式版本。如需 Amazon Redshift Serverless SLA 的相關資訊，請參閱 Amazon Redshift 服務水準協議。

追蹤 - 當 Amazon Redshift 發行新的工作群組版本時，您的工作群組會自動更新。您可控制是否要將工作群組更新至最新版或前一版。如需有關追蹤的資訊，請參閱 Amazon Redshift 佈建叢集和無伺服器工作群組的追蹤。

可用區域 ID - 在設定 Amazon Redshift Serverless 執行個體時，請開啟其他考量，並確保子網路中提供的子網路 ID 至少包含兩個支援的可用區域 ID。

- 對於未使用增強型 VPC 路由 (EVR) 的工作群組，您需要兩個可用區域 (AZ)。
- 對於使用 EVR 的工作群組，您需要三個 AZ。

若要查看子網路與可用區域 ID 的對應，請移至 VPC 主控台，然後選擇子網路以查看子網路 ID 及其可用區域 ID 的清單。請確認您的子網路有對應至支援的可用區域 ID。若要建立子網路，請參閱《Amazon VPC 使用者指南》中的在 VPC 中建立子網路。

兩個子網路 - 您必須至少有兩個子網路，而且這些子網路必須跨越兩個可用區域。

三個子網路 (具有 EVR ONLY) - 您使用 EVR 時至少須有三個子網路，而且這些子網路必須跨越三個或更多可用區域。

可用 IP 位址需求 - 在未啟用增強型 VPC 路由 (EVR) 的情況下使用 Redshift Serverless 時，每個子網路中至少須有三個可用的 IP 位址。這是服務正常運作的需求。

更新 Redshift Serverless 部署的 RPU 時，每個子網路中至少須有三個可用的 IP 位址，以因應服務的操作需求。

如需有關在 Amazon VPC 中配置 IP 位址和了解 IP 定址的詳細資訊，請參閱《Amazon VPC 使用者指南》中的 VPC 和子網路的 IP 定址。

Without EVR

如果您未使用增強型 VPC 路由，則無論基本 RPU 的大小 (4 到 1024 個 RPU) 或工作群組的 RPU 用量為何，每個子網路都至少須有三個可用的 IP 位址。3 個 IP 位址的需求也適用於已啟用 AI 驅動擴展和最佳化功能的工作群組。

With Enhanced VPC Routing (EVR)

如果您使用增強型 VPC 路由搭配 Redshift Serverless，則建立工作群組時所需的最少 IP 位址數量如下：

Redshift 處理單元 (RPU)	所需的可用 IP 地址	最小的 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

若使用 EVR，您將工作群組更新為使用更多 RPU 時，也需要可用的 IP 位址。更新工作群組的子網路時所需的可用 IP 位址數量如下：

Redshift 處理單元 (RPU)	更新後的 Redshift 處理單元 (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**
>
> 最大基本 RPU 容量上限 1024 僅適用於下列 AWS 區域：
>
> - 美國東部 (維吉尼亞北部)
> - 美國東部 (俄亥俄)
> - 美國西部 (奧勒岡)
> - 歐洲 (愛爾蘭)
> - 歐洲 (倫敦)

如需配置 IP 地址的相關資訊，請參閱《Amazon VPC 使用者指南》中的 IP 定址。

遷移後的儲存空間 — 將 Amazon Redshift 佈建的小型叢集遷移到 Amazon Redshift Serverless 時，您可能會發現遷移後的儲存空間配置有所增加。這是儲存空間配置經過最佳化，進而預先配置儲存空間的結果。隨著 Amazon Redshift Serverless 中的資料增加，此空間可使用一段時間。

Amazon Redshift Serverless 和 Amazon Redshift 佈建的叢集之間的資料共用 — 當有 Amazon Redshift Serverless 為生產者且已佈建的叢集是取用者的資料共用時，已佈建叢集的版本必須是 1.0.38214 之後的叢集版本。如果您使用此版本之前的叢集版本，則會在執行查詢時發生錯誤。您可以在 Amazon Redshift 主控台上的維護索引標籤上檢視叢集版本。您也可以執行 `SELECT version();`。

查詢執行時間上限 — 查詢所經歷的執行時間 (以秒為單位)。執行時間不包括在佇列中等待所花的時間。如果查詢超過設定的執行時間，Amazon Redshift Serverless 會停止查詢。有效值為 0 至 86,399。

具有交錯排序索引鍵的資料表遷移 — 將 Amazon Redshift 佈建的叢集遷移至 Amazon Redshift Serverless 時，Redshift 會將具有交錯排序索引鍵和 DISTSTYLE KEY 的資料表轉換為複合排序索引鍵。DiSTSTYLE 則不會變更。如需分佈樣式的相關資訊，請參閱《Amazon Redshift 開發人員指南》中的使用資料分佈樣式。如需排序索引鍵的相關資訊，請參閱使用排序索引鍵。

VPC 共用 — 您可以在共用的 VPC 中建立 Amazon Redshift Serverless 工作群組。如果您這麼做，建議您不要刪除資源共用，因為這會導致工作群組變得無法使用。

IPv6 支援：

Amazon Redshift Serverless 支援在 AWS 虛擬私有雲端 (VPCs) 中使用 IPv4 和 IPv6 地址 （雙堆疊）或IPv4-only 組態來設定 Amazon Redshift 工作群組。您可以在建立新的 Amazon Redshift Serverless 工作群組時啟用 IPv6 支援，或修改現有的工作群組以支援 IPv6 定址。透過此功能，您可以在IPv6-enabled VPC 子網路中部署 Amazon Redshift Serverless 倉儲，並設定網路設定以支援應用程式的擴充地址空間需求。您的應用程式現在可以使用 IPv4 或 IPv6 通訊協定與 Amazon Redshift Serverless 倉儲通訊，以確保與現有和未來網路架構的相容性。

Amazon Redshift Serverless 的運算容量

Amazon Redshift Serverless 運算容量會自動擴展和縮減，以符合您的工作負載需求。運算容量是指分配給 Amazon Redshift Serverless 工作負載的處理能力和記憶體。常見的使用案例包括處理高峰流量期間、執行複雜的分析，或有效率地處理大量資料。下列術語提供 Amazon Redshift 如何管理運算容量的詳細資訊。

RPU

Amazon Redshift Serverless 會以 Redshift 處理單元 (RPU) 為單位來測量資料倉儲的容量。RPU 是用來處理工作負載的資源。一個 RPU 可提供 16 GB 的記憶體。

基本容量

此設定會指定 Amazon Redshift 用來為查詢提供服務的基本資料倉儲容量。基本容量會以 RPU 為單位來指定。您能以 Redshift 處理單元 (RPU) 為單位來設定基本容量。設定較高的基本容量可改善查詢效能，對於需要大量資源的資料處理任務來說更是如此。Amazon Redshift Serverless 的預設基本容量為 128 個 RPU。您可以將基本容量設定從 4 個 RPU 調整為 512 個 RPU。您可以將此值設定為 4 個 RPU，或以 8 為單位設定為等於或大於 8 個 RPU (8、16、24...512)。您可以使用 AWS 主控台、`UpdateWorkgroup`API 操作或 中的 `update-workgroup`操作來設定此值 AWS CLI。

最小基本容量為 4 個 RPU 時，您可以根據資料倉儲需求和效能需求，彈性地執行較簡單到較複雜的工作負載。4 個基本 RPU 容量的目標是包含的資料少於 32TB 的倉儲，而 8、16 和 24 個 RPU 基本 RPU 容量的目標是需要的資料少於 128TB 的工作負載。如果您的資料需求大於 128 TB，則至少須使用 32 個基本 RPU。此外，如果工作負載的資料表有大量的欄且並行數量較高，建議您使用 32 個以上的基本 RPU。

可用的基本 RPU 上限為 1024 個，此值會將最高層級的運算資源新增至您的工作負載。這樣就可提供更大的彈性來支援高度複雜的工作負載，並加速載入和查詢資料。

> **Note**
>
> 擴充的基礎 RPU 容量上限為 1024，可在下列內容中使用 AWS 區域。在其他區域中，基本容量上限為 512 個 RPU。
>
> - 美國東部 (維吉尼亞北部)
> - 美國東部 (俄亥俄)
> - 美國西部 (奧勒岡)
> - 歐洲 (愛爾蘭)
> - 歐洲 (法蘭克福)
>
> 設定 512 到 1024 個之間的基本容量時，可採用 32 為單位增量或減量 RPU。

如果您管理更大且更複雜的工作負載，請考慮增加 Redshift Serverless 資料倉儲的大小。倉儲越大，能夠存取的運算資源越多，如此就能更有效率地處理查詢。

以下是設定較高基本容量較為有利的一些範例：

- 您的查詢較複雜，執行時間較長

- 您的資料表擁有大量的欄。
- 您的查詢有大量 JOIN。
- 您的查詢會彙總或掃描外部來源 (例如資料湖) 的大量資料。

如需 Amazon Redshift Serverless 配額和限制的詳細資訊，請前往 Amazon Redshift Serverless 物件的配額。

Amazon Redshift Serverless 容量的考量和限制

以下是 Amazon Redshift Serverless 容量的考量和限制。如需一般 Redshift Serverless 考量事項，請參閱 使用 Amazon Redshift Serverless 時的考量。

- 4 個基本 RPU 的組態最多可支援 32 TB 的受管儲存容量。如果您要使用超過 32 TB 的受管儲存，則無法將基本 RPU 設定為少於 8 個 RPU。
- 8 個或 16 個基本 RPU 的組態最多可支援 128 TB 的 Redshift 受管儲存容量。如果您要使用超過 128 TB 的受管儲存，則無法將基本設定為少於 32 個 RPU。
- 編輯工作群組的基本容量，如此可能會取消工作群組上執行的某些查詢。
- Redshift Serverless 使用以下增量擴展資料倉儲的 RPU：
  - 4 到 8 個 RPU：以 4 個 RPU 為單位增加。
  - 8 到 512 個 RPU：以 8 個 RPU 為單位增加。
  - 512 到 1024 個 RPU：以 32 個 RPU 為單位增加。
- 只有 8 個以上的 RPU 才支援清空提升。對於 8 個以下的 RPU，請改用下列命令：

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

具有 4 個 Redshift 處理單元 (RPU) 容量的 Redshift Serverless

具有 4 個基本 RPU 容量的 Redshift Serverless 適合較小或需求較低的工作負載。此起點提供既靈活又符合成本效益的解決方案。此入門層級組態支援最多擁有下列資源數的資料倉儲：

- 最多 32 TB 的 Redshift 受管儲存。
- 每個資料表最多 100 個欄
- 64 GB 記憶體

如果您的需求超過上述限制，則必須手動增加基本容量，而不是依賴自動擴展。一旦您將資料倉儲擴展到超過 4 個 RPU，您的資料倉儲將繼續使用更多 RPU，而 Amazon Redshift 不會將您的資料倉儲縮減至 4 個 RPU。

> **Note**
>
> 您可以在使用 4 個基本 RPU 時建立包含超過 100 個欄的資料表，不過建議您將資料表限制為 100 個欄。超過此限制可能會導致您的資料倉儲在執行查詢時耗盡其記憶體，而造成效能降低。

您可以在下列內容中建立使用 4 個 RPUs的資料倉儲 AWS 區域：

- 美國東部 (俄亥俄)
- 美國東部 (維吉尼亞北部)
- 美國西部 (加利佛尼亞北部)
- 美國西部 (奧勒岡)
- 亞太地區 (孟買)
- 亞太地區 (新加坡)
- 亞太地區 (雪梨)
- 亞太地區 (東京)
- 歐洲 (愛爾蘭)
- 歐洲 (斯德哥爾摩)

AI 驅動的擴展和最佳化 (預覽版)

> **Note**
>
> AI 驅動的擴展和最佳化可在可使用 Amazon Redshift Serverless 的所有 AWS 區域中使用。

Amazon Redshift Serverless 提供進階 AI 驅動的擴展和最佳化功能，以因應各種不同的工作負載需求。資料倉儲可能發生下列佈建問題：

- 資料倉儲可能過度佈建，以改善資源密集型查詢的效能
- 資料倉儲可能佈建不足，以節省成本。

在資料倉儲工作負載的效能和成本之間取得適當的平衡具有挑戰性，尤其是在臨機查詢和資料量不斷增加的情況下。執行包含低和高資源密集型查詢的混合工作負載時，需要智慧型擴展。AI 驅動的擴展和最佳化功能會自動擴展無伺服器運算或 RPU，以應付資料成長。此功能也有助於將查詢效能維持在目標性價比目標內。AI 驅動的擴展和最佳化會在資料量增加時動態配置運算資源，確保查詢持續符合效能目標。AI 驅動的擴展和最佳化可讓服務順利適應不斷變化的工作負載需求，而不需要手動介入或複雜的容量規劃。

Amazon Redshift Serverless 根據查詢複雜性和資料量等因素，提供更全方位且即時回應的擴展解決方案。此功能允許在最佳化工作負載性價比的同時保持彈性，以有效率地處理不同的工作負載和不斷增長的資料集。Amazon Redshift Serverless 可以自動將 AI 驅動的擴展和最佳化套用至 Amazon Redshift Serverless 端點，以滿足 Serverless 工作群組指定的價格效能目標。如果您不知道要為工作負載設定哪些基本容量，或是工作負載的某些部分可能會受益於更多配置的資源，則此自動定價效能優化特別有用。

範例

若您的組織通常執行的工作負載只需要 32 個 RPU，但突然導入了更複雜的查詢，您可能不知道應該要多少基本容量才算適當。設定較高的基本容量可產生更高的效能，但也會產生較高的成本，因此成本可能不符合您的預期。使用 AI 驅動的擴展和最佳化，Amazon Redshift Serverless 會自動調整 RPUs 以符合您的價格效能目標，同時保持組織的成本最佳化。無論工作負載大小是多少，此自動最佳化都很有效用。如果您有任何數量的複雜查詢，自動最佳化可協助您達成組織的價格績效目標。

> **Note**
>
> 價格效能目標是特定於工作群組的設定。不同的工作群組可以有不同的價格績效目標。

若要設定價格效能目標，請使用 AWS 主控台。根據預設，所有新的 Serverless 工作群組都會啟用價格效能目標，並將 設定為平衡。您可以在建立 Serverless 工作群組後修改價格效能目標。

若要編輯工作群組的性價比目標：

1. 在 Amazon Redshift Serverless 主控台中，選擇工作群組組態。
2. 選擇您要為其編輯價格效能目標的工作群組。選擇效能索引標籤，然後選擇編輯。
3. 選擇性價比目標，並將滑桿調整至所需的設定。
4. 選擇儲存變更。
5. 若要更新 Amazon Redshift Serverless 可分配給工作負載的 RPU 數量上限，請選擇工作群組區段的限制索引標籤。

您可以使用性價比目標滑桿，在成本和效能之間設定所需的平衡。移動滑桿即可選擇下列其中一個選項：

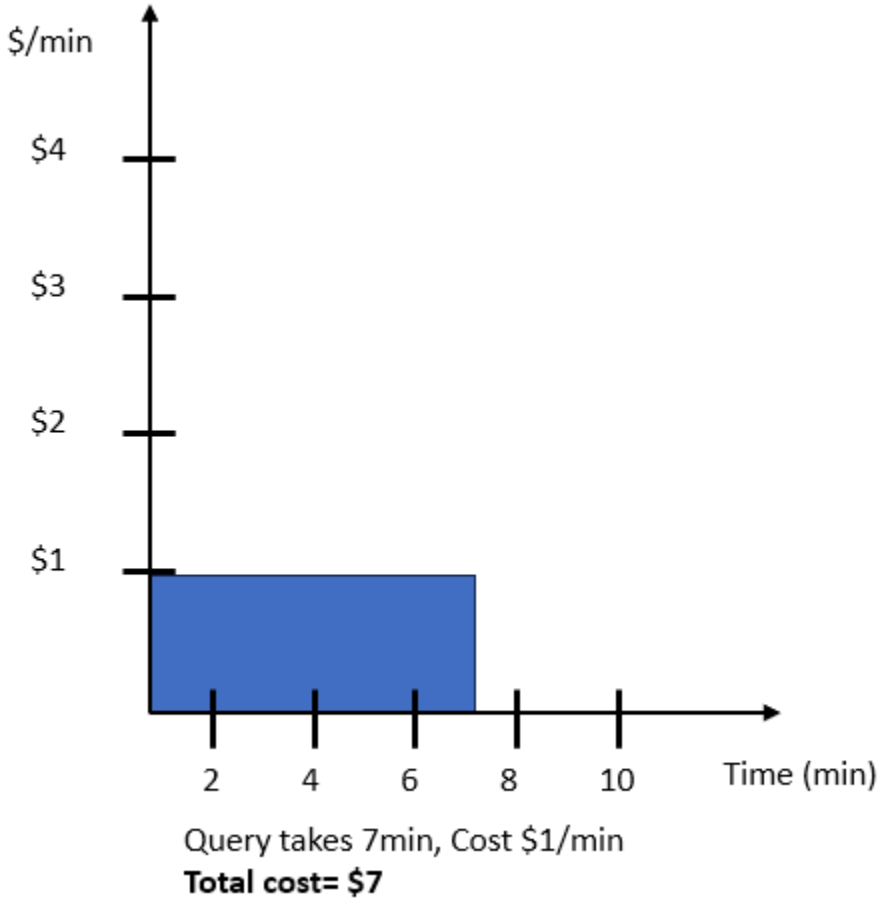
- 針對成本進行最佳化 - 此設定會優先考慮節省成本。Amazon Redshift Serverless 只有在可能降低成本時，才會嘗試自動擴展運算容量。Amazon Redshift Serverless 也會嘗試縮減運算資源以降低成本，不過可能會增加查詢執行時期。
- 平衡 - 此設定會在效能和成本之間取得平衡。Amazon Redshift Serverless 會擴展效能，並可能導致中等的成本增加或減少。這是大多數 Amazon Redshift Serverless 資料倉儲的建議設定。
- 針對效能進行最佳化 - 此設定會優先考慮效能。Amazon Redshift 會積極擴展以實現高效能，因而可能產生較高的成本。
- 中間位置：您也可以將滑桿設定為平衡和針對成本進行最佳化或針對效能進行最佳化之間的兩個中間位置之一。如果完全針對成本或效能最佳化太過極端，請使用這些設定。

選擇性價比目標時的考量事項

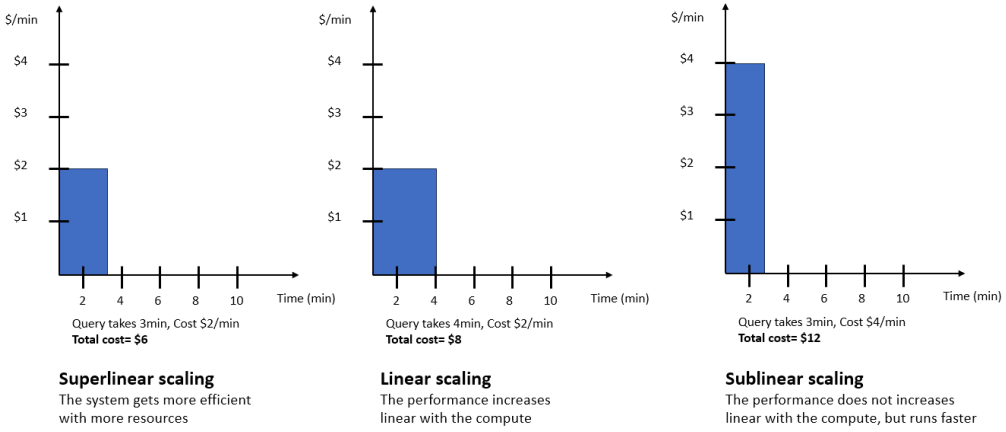
您可以使用性價比滑桿來為工作負載選擇所需的性價比目標。AI 驅動的擴展和最佳化演算法會隨著時間從您的工作負載歷史記錄學習，並改善預測和決策準確性。

範例

此範例假設查詢花 7 分鐘時間，且費用為 7 美元。下圖顯示未擴展的查詢執行時期和成本。



某一特定查詢可能會以幾種不同的方式擴展，如下所示。根據您選擇的價格效能目標，AI 驅動的擴展和最佳化會預測查詢如何權衡效能和成本，並據此進行擴展。選擇不同的滑桿選項會產生下列結果：



- 針對成本進行最佳化 - 使用針對成本進行最佳化選項時，您的資料倉儲會採用有利降低成本的選項進行擴展。在上述範例中，超線性擴展方法示範了此行為。擴展只會在根據擴展模型預測，能夠以符合

成本效益的方式進行時完成。如果擴展模型預測無法針對所指工作負載進行成本最佳化擴展，則資料倉儲將不會擴展。

- 平衡 - 使用平衡選項時，系統會進行擴展，同時平衡成本和效能考量，且可能有限地增加成本。平衡選項會執行超線性、線性及可能亞線性工作負載擴展。
- 針對效能進行最佳化 - 使用針對效能進行最佳化選項，除了先前改善效能的方法之外，系統也會在即使成本較高的情況下擴展，而且可能不會與執行時期改善成正比。透過針對效能進行最佳化，系統會盡可能執行超線性擴展、線性擴展和亞線性擴展。滑桿位置越接近針對效能進行最佳化位置，Amazon Redshift Serverless 越會允許亞線性擴展。

設定性價比滑桿時，請注意下列事項：

- 您隨時可以變更性價比設定，但工作負載擴展不會立即變更。隨著系統了解目前工作負載，擴展也會隨著時間而變更。我們建議監控 Serverless 工作群組 1 到 3 天，以確認新設定的影響情形。
- 性價比滑桿選項最大容量和最大 RPU 小時數可一起運作。最大容量和最大 RPU 小時數是兩個控制項，分別用於限制 Amazon Redshift Serverless 允許資料倉儲擴展的最大 RPU，以及 Amazon Redshift Serverless 允許資料倉儲使用的最大 RPU 小時數。無論性價比目標設定為何，Amazon Redshift Serverless 一律遵守並強制執行這些設定。

監控資源自動擴展

您可以透過下列方式監控 AI 驅動的擴展和最佳化：

- 在 Amazon Redshift 主控台上檢閱使用的 RPU 容量圖。
- 監控 CloudWatch 中 `AWS/Redshift-Serverless` 和 `Workgroup` 下的 `ComputeCapacity` 指標。
- 查詢 SYS_QUERY_HISTORY 檢視。提供特定查詢 ID 或查詢文字以識別時段。使用此時段查詢 SYS_SERVERLESS_USAGE 系統檢視以尋找 `compute_capacity` 值。`compute_capacity` 欄位顯示查詢執行時期中擴展的 RPU。

使用以下範例查詢 SYS_QUERY_HISTORY 檢視。將範例值取代為您的查詢文字。

```
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
```

執行下列查詢，以查看 compute_capacity 在 start_time 到 end_time 這段期間如何擴展。將下列查詢中的 start_time 和 end_time 取代為上述查詢的輸出：

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

如需使用這些功能的逐步指示，請參閱在 Amazon Redshift Serverless 中設定監控、限制和警示，以保持成本的可預測性。

使用 AI 驅動的擴展和最佳化時的考量事項

使用 AI 驅動的擴展和最佳化時，請考慮下列事項：

- 對於需要 8 到 512 Base RPU 的 Amazon Redshift Serverless 上的現有工作負載，請使用 AI 驅動的擴展和最佳化來獲得最佳結果。對於低於或高於此 RPU 範圍的工作負載，我們不建議使用此功能。
- 價格效能目標會自動最佳化工作負載，但結果可能會有所不同。隨著時間使用此功能，以便系統可以透過執行代表性的工作負載來學習您的特定模式。
- AI 驅動的擴展和最佳化會使用最佳時間，根據在 Amazon Redshift Serverless 執行個體上執行的工作負載，將最佳化套用至 Serverless 工作群組。

若要進一步了解 AI 驅動的擴展和最佳化，請觀看下列影片。

Amazon Redshift Serverless 的帳單

運算容量的帳單

您可以透過兩種方式為 Amazon Redshift Serverless 購買容量：

- 您可以購買隨需容量 - 當您選擇隨需運算容量時，只需支付所使用資源的費用。這是剛開始使用 Amazon Redshift Serverless 或還不清楚穩定的用量模式時的首選。隨需提供最大的彈性。如需詳細資訊，請參閱隨需運算容量的帳單。
- 您可以購買保留 - 當您購買特定時間內 (例如一年) 預設數量的運算資源時，保留可提供折扣。這是適合您清楚將要穩定使用某個數量的容量時的選項。由於您可以預測部分容量需求，因此這個選項有助於節省成本。如需詳細資訊，請參閱無伺服器保留的帳單。

您可以同時使用保留和隨需資源。您不必只選用其中之一。

如需詳細的定價資訊，請參閱 Amazon Redshift 定價。

隨需運算容量的帳單

基本容量及其對帳單的影響

執行查詢時，系統會根據指定期間內使用的容量來計費 (以每秒 RPU 小時進行計算)。沒有執行任何查詢時，則不會向您收取運算容量的費用。系統也會根據所儲存的資料量向您收取 Redshift 受管儲存 (RMS) 的費用。

在建立工作群組時，您可以選擇設定運算的基本容量。若要符合工作群組層級的工作負載性價比要求，請將現有工作群組的基本容量調高或調低。從工作群組組態中選取工作群組，然後選擇限制索引標籤，以使用主控台變更基本容量。

隨著查詢數量的增加，Amazon Redshift Serverless 會自動擴展以提供一致的效能。

最大 RPU 時數使用限制

為了讓 Amazon Redshift Serverless 的成本保持可預測性，您可以設定每天、每週或每月能夠使用的 RPU 時數上限。使用主控台或透過 API 即可設定此限制。達到限制時，您可以指定要在系統資料表中寫入日誌項目、接收警示或關閉使用者查詢。設定 RPU 時數上限有助於控制成本。RPU 時數上限的設定會同時在存取資料倉儲資料的查詢和存取外部資料 (例如 Amazon S3 中的外部資料表) 的查詢兩方面適用於工作群組。

以下是範例：

假設您每週設定 100 小時的限制。若要在主控台上執行此操作，請執行下列步驟：

1. 選擇您的工作群組，然後在限制標籤下選擇管理用量限制。
2. 新增用量限制、選擇每週頻率、持續時間為 100 小時，並將動作設定為關閉使用者查詢。

在此範例中，如果您達到一週 100 RPU 小時限制，則關閉查詢。

設定工作群組的 RPU 時數上限並不會限制工作群組的效能或運算資源。您可以隨時調整設定值，而不會中斷查詢作業。設定最大 RPU 時數的目標，是協助您符合價格和效能需求。如需無伺服器帳單的相關資訊，請參閱 Amazon Redshift 定價。

保留 Amazon Redshift Serverless 可預測成本的另一種方法是使用 AWS 成本異常偵測，以減少帳單意外的機會並提供更多控制。

> **Note**
>
> Amazon Redshift 定價計算器有助於估算定價。只要輸入所需的運算資源，其便會提供成本預覽。

設定最大容量以控制運算資源的成本

最大容量設定可作為 Amazon Redshift Serverless 可縱向擴展的 RPU 上限。此有助於控制運算資源的成本。與基本容量設定最小可用運算資源的方式類似，最大容量會設定 RPU 用量上限。這樣就能幫助您的支出符合您的計劃。最大容量特別適用於每個工作群組，並且隨時都會限制運算用量。

最大容量與 RPU 時數用量限制有何不同

最大 RPU 小時限制和最大容量都可協助您控制成本，但它們的運作方式不同。以下幾點說明了不同之處：

- 最大容量 – 此設定會建立 Amazon Redshift Serverless 可用於自動擴展目的的最高 RPUs 計數。
- 最大 RPU 時數用量限制 — 與最大容量不同，此設定不會設定容量上限。不過，此設定會確實執行其他動作來幫助您限制成本。其中包括將項目新增至記錄檔、通知您，或停止執行查詢 (如果您選擇的話)。

您可以只使用最大容量，也可以搭配最大 RPU 小時用量限制中的動作。

最大容量使用案例

每個工作群組都可以有不同的最大容量設定。它可以幫助您執行預算編列要求。下列範例說明其運作方式：

- 您的工作群組的基本容量設定為 256 個 RPU。整個月的大部分時間您的工作負載穩定在 256 RPU 以上。
- 最大容量設定為 512 RPU。

假設您在三天期間內有非預期的大量使用率來產生臨機操作統計報告。在此情況下，您已設定最大容量來避免超過 512 RPU 的運算成本。執行此操作時，您可以確定運算容量不會超過此上限。

最大容量的使須知

這些使用須知可協助您正確設定最大容量：

- 每個 Amazon Redshift Serverless 工作群組都可以有不同的最大容量設定。
- 如果您在一段時間內資源使用率非常高，並且最大容量設定為較低的 RPU 級別，如此可能會延遲工作負載處理，並導致使用者體驗不佳。
- 您可以在 Amazon Redshift Serverless 儀表板上檢視一段時間內使用的容量。如需有關檢視摘要資料的詳細資訊，請參閱使用儀表板檢查 Amazon Redshift Serverless 摘要資料。
- 最大容量變更通常不會中斷執行中的查詢。不過，如果您將最大容量設定為低於目前的基本容量，Amazon Redshift Serverless 會立即減少容量。這可能會影響您執行中的工作負載。
- 最大容量設定為 5632 RPU。

如何設定最大容量

您可以在主控台設定最大容量。對於既有的工作群組，您可以變更工作群組組態下的設定。您也可以使用 CLI 執行下列範例所示的命令來設定：

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

這會為具有指定名稱的工作群組設定最大容量設定。設定完成後，您可以在主控台上檢查該值以進行驗證。您也可以使用 CLI 執行 `get-workgroup` 命令來檢查值。

您可以透過設定為 `-1` 來關閉最大容量設定，如下所示：

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

監控 Amazon Redshift Serverless 用量和成本

有數種方式可以估算 Amazon Redshift Serverless 的用量和帳單。系統檢視會有幫助，因為系統中繼資料 (包括查詢和用量資料) 是及時的，而且您不需要進行任何設定即可查詢。CloudWatch 對於監控 Amazon Redshift Serverless 執行個體的用量也有幫助，並有其他功能可提供洞察和設定動作。

透過查詢系統檢視將用量視覺化

查詢 SYS_SERVERLESS_USAGE 系統資料表即可追蹤用量並取得查詢費用：

```
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
```

此查詢會根據用量，提供 Amazon Redshift Serverless 每天產生的費用。

用於決定用量和成本的用量注意事項

- 您只需就所執行的工作負載支付費用，計費基礎為每秒 RPU 小時，最低費用為 60 秒。
- sys_serverless_usage 系統資料表中的記錄會顯示以 1 分鐘的時間間隔產生的成本。了解下列資料欄很重要：

  charged_seconds 資料欄：
  - 提供在時間間隔內收費的運算單位 (RPU) 秒數。結果裡面包括 Amazon Redshift Serverless 中的任何最低費用。
  - 具有交易完成後運算資源用量的相關資訊。因此，如果交易尚未完成，此資料欄的值可能會是 0。

  compute_seconds 資料欄：
  - 提供即時運算用量資訊。這不包括 Amazon Redshift Serverless 中的任何最低費用。因此，其可能會與間隔期間所計費的收費秒數有某種程度上的不同。
  - 顯示每筆交易期間的用量資訊 (即使交易尚未結束)，因此所提供的資料為即時資料。
- 在某些情況下，compute_seconds 為 0，但 charged_seconds 大於 0，反之亦然。這是由於在系統檢視中記錄資料的方式所產生的正常行為。若要更準確地呈現無伺服器使用情況詳細資訊，建議您彙總 SYS_SERVERLESS_USAGE 中的資料。

如需監控資料表和檢視的相關資訊，請參閱使用 Amazon Redshift Serverless 監控查詢和工作負載。

使用 CloudWatch 將用量視覺化

您可以使用 CloudWatch 中提供的指標來追蹤用量。針對 CloudWatch 所產生的指標為 `ComputeSeconds` (會指出當下這一分鐘使用的總 RPU 秒數) 和 `ComputeCapacity` (會指出該分鐘的總運算容量)。您也可以在 Redshift Serverless 儀表板上的 Redshift 主控台上找到用量指標。如需 CloudWatch 的相關資訊，請參閱什麼是 Amazon CloudWatch？

無伺服器保留的帳單

Amazon Redshift Serverless 可讓您執行和擴展分析，而不需透過依用量計費定價模型佈建和管理叢集。現在透過無伺服器保留，您可以進一步最佳化運算成本，並改善 Redshift Serverless 上現有和新工作負載的成本可預測性。

Amazon Redshift 會在 AWS 付款人帳戶層級管理無伺服器保留，而且可以在多個 AWS 帳戶之間共用保留，讓您可以將 AWS 帳戶中所有 Redshift Serverless 工作負載的運算成本降低高達 24%。Amazon Redshift 的無伺服器保留是按小時計費，並且每秒計算保留，藉此提供全年一致的計費模式，同時維持 Redshift Serverless 提供的彈性。Amazon Redshift 針對超過指定 RPU 層級的任何用量，按標準隨需費率計費。

> **Note**
>
> 如果您想要限制隨需用量，您可以使用最大容量設定來設定工作群組的資源用量限制。如需詳細資訊，請參閱Amazon Redshift Serverless 的帳單。

無伺服器保留的優勢

無伺服器保留是 Amazon Redshift Serverless 的折扣定價選項。無伺服器保留可讓您選擇以隨需 (OD) 費率的折扣價格，承諾指定數量的 Redshift 處理單元 (RPU) 為期一年，且不需預付款項。若預付款項還可獲得更大的折扣。透過無伺服器保留，您可以最佳化運算成本，並改善 Serverless 上現有和新工作負載的成本可預測性。

每個無伺服器保留都是在 AWS 帳戶層級購買，並且可以在相同付款人帳戶中的多個 Amazon Redshift Serverless 工作群組之間共用。如此您就能靈活運用折扣。保留可以在多個具有不同工作負載模式的工作群組之間共用。

無伺服器保留的運作方式

保留 RPU 是一項簡單的程序，只需幾分鐘就能完成。此程序包括指定要保留的 RPU 層級和付款類型。Amazon Redshift Serverless 使用標準 AWS 帳單和成本管理工具，協助您判斷所需的保留層級，並持續監控您的用量。無伺服器保留是在 AWS 付款人帳戶層級管理，並且可以在相同的付款人帳戶下共用，並可讓您在 AWS 帳戶中所有 Redshift Serverless 工作負載上降低高達 24% 的運算成本。無伺服器保留是按小時計費，並且每秒計算，藉此提供全年一致的計費模式，同時維持 Redshift Serverless 提供的彈性。超過指定 RPU 層級的任何用量都會按標準 Redshift Serverless 隨需費率計費。

您可以在同一個 AWS 帳戶中購買多個無伺服器保留。當您購買其他無伺服器保留時，這些保留會彼此層疊。例如，如果您購買兩個保留，並分別為每個保留選擇 100 個 RPU，則會以折扣費率為您提供總計 200 個 RPU。

> **Note**
>
> 如果您想要設定隨需用量的限制，則可以在 Amazon Redshift Serverless 主控台中選擇限制索引標籤，然後選取管理用量限制，以設定工作群組的最大 RPU。

購買無伺服器保留後，該保留會立即生效，並在 Redshift 主控台的無伺服器保留儀表板中顯示。

分析 RPU (Redshift 處理單元) 使用情形來判斷所需的保留層級

Redshift Serverless 保留可讓您鎖定可預測的較低運算成本，方法是承諾特定 Redshift 處理單元 (RPU) 數量為期一年，如此您就能享有隨需定價的折扣。搭配無預付選項可享有高達 20% 的折扣，若全部預付，則最高可享有 24% 的折扣。您在 AWS 付款人帳戶層級購買 Redshift Serverless 預留，而您的節省會自動套用至任何連結帳戶中的任何 AWS Redshift Serverless 工作群組，因此您可以在支援多個團隊的同時集中管理預算。Redshift Serverless 會以每秒精細程度來計算用量，取得每小時平均值，然後按小時計費，確保您只需針對使用的容量付費。Redshift Serverless 保留結合了在帳戶之間彈性應用與期限為基礎的節省方式，為您提供可預測的分析價格，而不會犧牲 Redshift Serverless 的敏捷性。

分析保留的 RPU 使用情形

您可以透過兩種方式之一來判斷 RPU 用量層級：您可以使用 Redshift Serverless 儀表板查看七天的內容，或使用 Cost Explorer 進行長期分析。下列程序示範如何分析 RPU 使用情形：

方法 1：Redshift Serverless 儀表板 (7 天檢視)

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 開啟 Serverless 儀表板。
3. 選擇您的工作群組。
4. 檢視從最後一個小時到一週期間內的 RPU 容量使用情形。

方法 2： AWS Cost Explorer （長期分析）

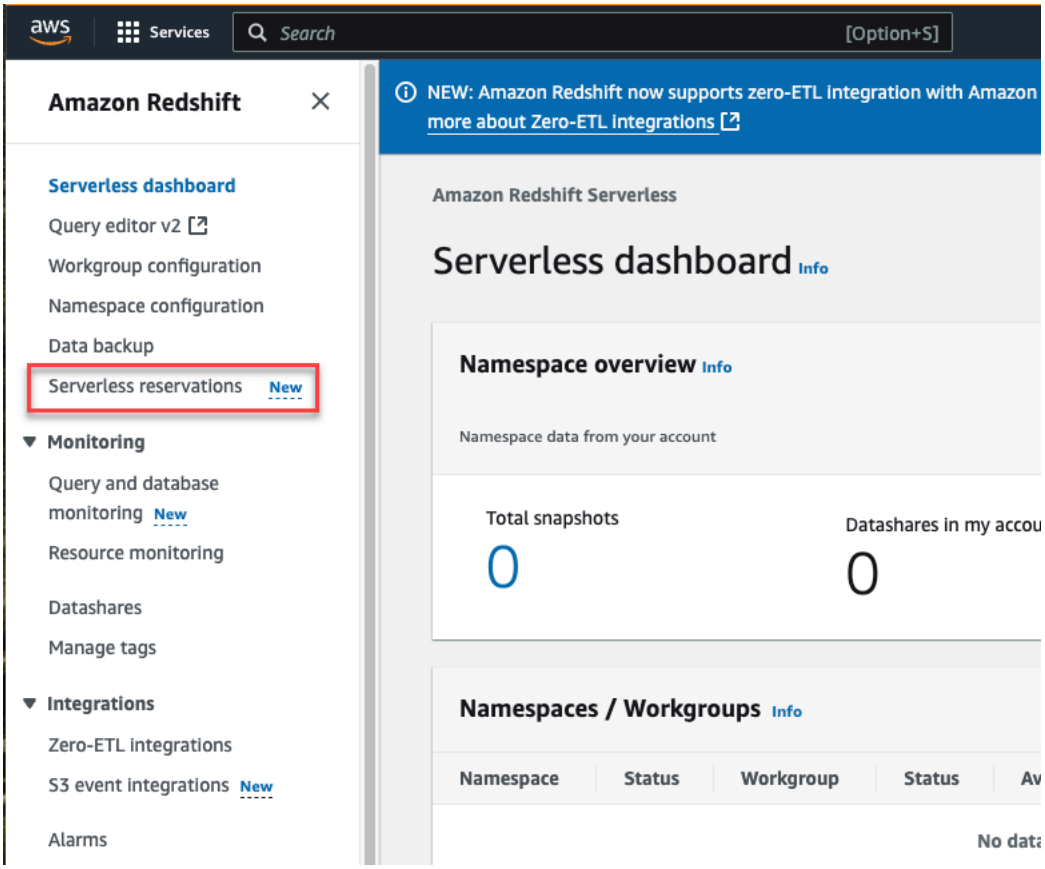
1. 登入 AWS 管理主控台 並開啟位於 https：//https://console.aws.amazon.com/costmanagement/ 的 Cost Explorer 主控台。
2. 將精細程度設定為每小時

3. 依用量類型分組
4. 套用下列篩選條件：
   - 服務：Redshift
   - 區域：您的本機區域
   - 用量類型：篩選條件 Redshift:ServerlessUsage
5. 檢閱成本和用量圖，了解您所選區域中的每小時無伺服器用量

使用主控台購買無伺服器保留

當您購買保留時，您可以選擇將套用折扣的 RPU 層級。在您選取 RPU 層級之前，最好先了解您的基本容量和隨時間使用的隨需容量。本節說明如何判斷您的容量並預留無伺服器保留。

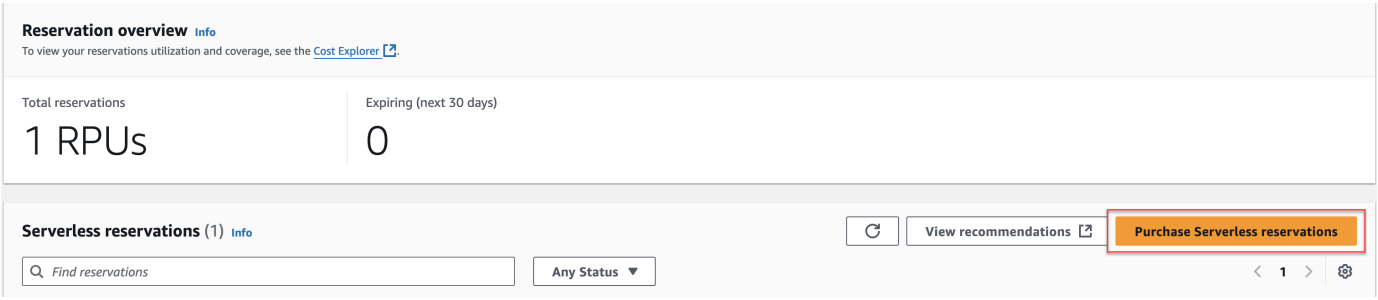
若要開始進行，請在 Redshift 主控台中選擇無伺服器，然後從功能表中選擇無伺服器保留。



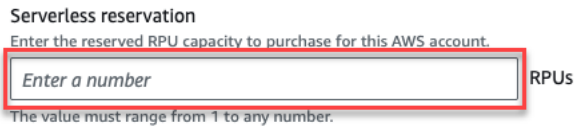
主控台會顯示功能的說明，以及現有保留的清單。您可以在這裡購買保留，也可以使用可用的報告和監控工具來查看目前的用量。這些可協助您判斷 RPU 層級，以及適合保留多少 RPU。

若要購買保留，請完成下列步驟：

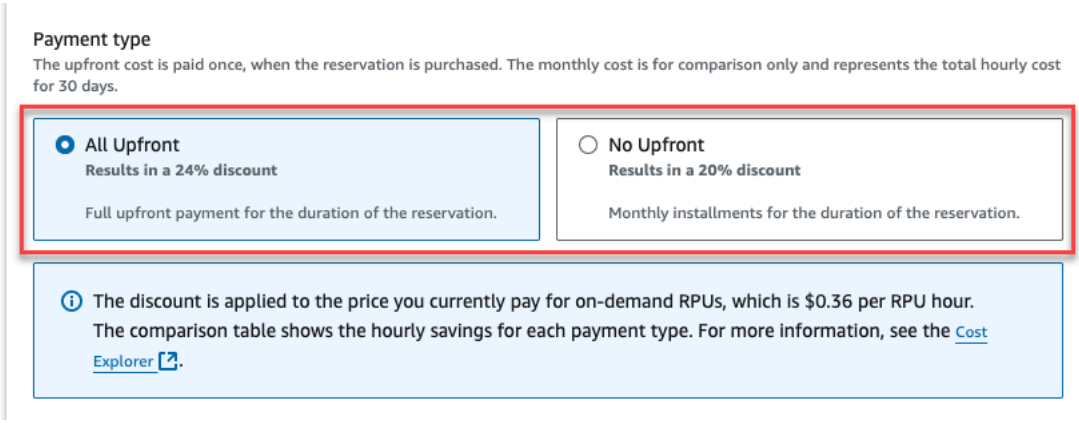
1. 選擇購買無伺服器保留。



2. 此時會出現逐步解說，其中包含一系列選擇。輸入要保留的無伺服器保留 RPU 層級。如果您不確定這應該是何種層級，您可以使用本節後段將說明的工具。

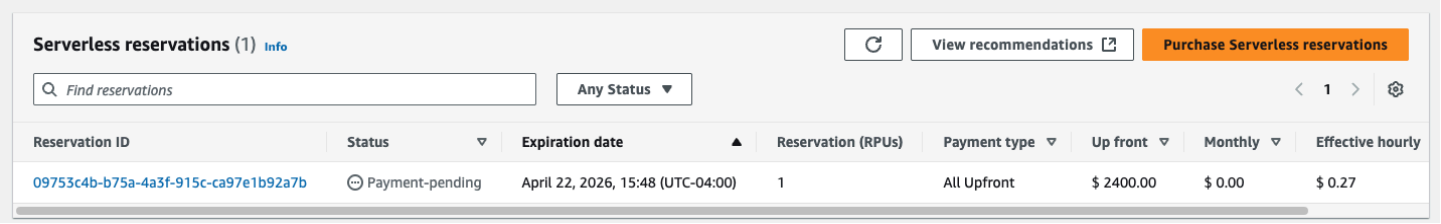


3. 設定付款類型。您可以選擇預付保留的 RPU，也可以每月付款。如果您選擇預付，可以享有更大的折扣。



4. 當您完成選擇時，請選擇購買無伺服器保留，然後選擇確認。

確認保留後，該保留會出現在保留清單中。



使用須知

- 您無法變更或刪除保留。但是您可以建立額外的保留，以擴大涵蓋範圍。
- Redshift Serverless 會在使用隨需 RPU 之前，先對工作負載使用保留的 RPU，以確保節省成本。如果您超過保留的 RPU 數量，這些額外的 RPU 將會按 Redshift Serverless 隨需費率計費。
- Amazon Redshift Serverless 的免費抵用金不適用無伺服器保留，只適用按隨需計費的 RPU。

無伺服器保留範例

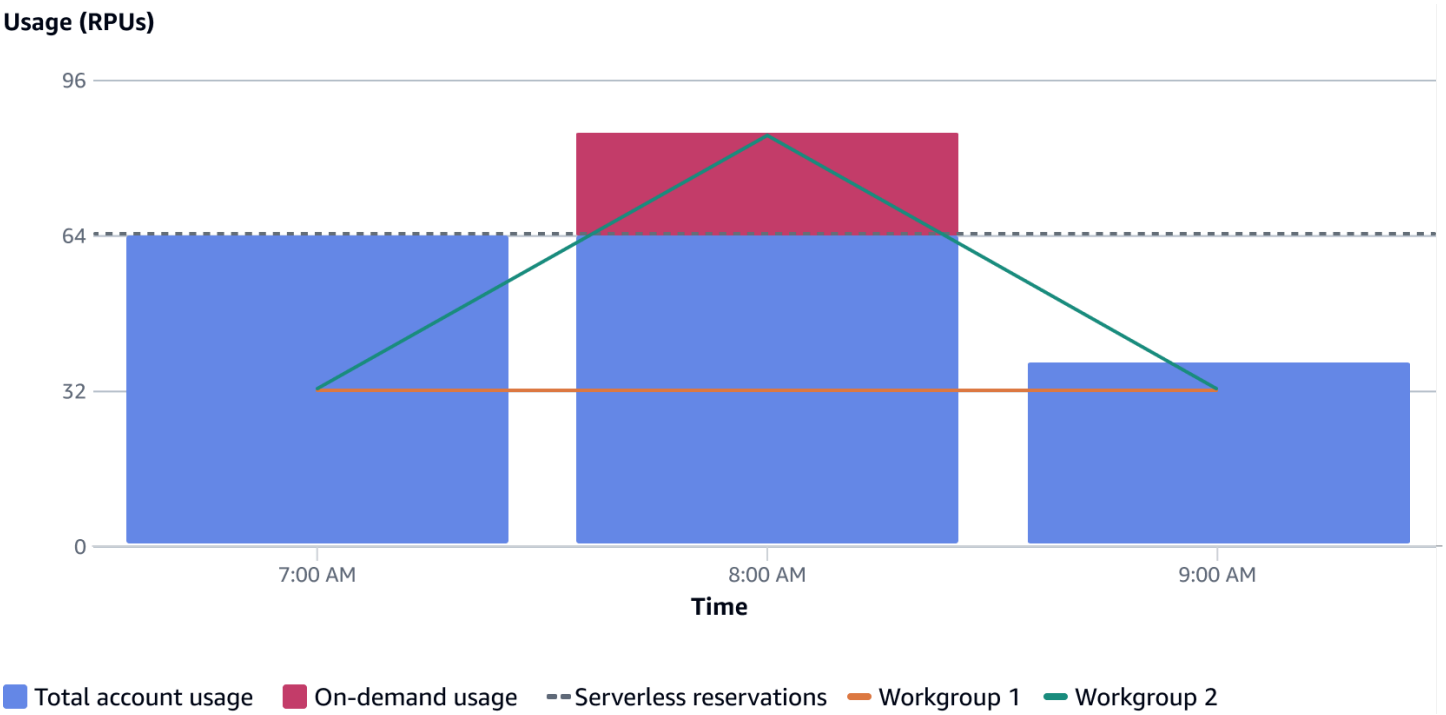
在此案例中，您的 AWS 付款人/連結帳戶有兩個 Amazon Redshift 工作群組：

- 工作群組 1 具有穩定狀態用量，例如商業智慧團隊。
- 工作群組 2 具有無法預測的工作負載，且會發生用量激增的情形，例如用於 ETL 操作。

您想要最佳化這些工作群組的成本，因此購買一年期的無伺服器保留。根據歷史資料，您可以判斷這兩個工作群組都在穩定狀態下使用 64 個 RPU。不過，工作群組 2 偶爾會在短時間內從 32 個 RPU 增加到 48 個 RPU，以及減少至 24 個 RPU。您一開始可以按照歷史趨勢，將保留的 RPU 層級設定為 64 個 RPU。每小時帳單詳細資訊如下：

- 第一個小時類似歷史用量趨勢，兩個工作群組都使用 32 個 RPU，因此帳戶總用量為 64 個 RPU。在這個小時內，所有 RPU 都會以無伺服器保留折扣費率計費。這是因為 64 個 RPU 的使用層級等於 64 個 RPU 無伺服器保留。
- 在第二個小時，工作群組 1 繼續使用 32 個 RPU。然而，工作群組 2 突增到 48 個 RPU，使得帳戶總用量上升到 80 個 RPU。在這個小時內，64 個 RPU 會以無伺服器保留折扣費率計費，而 16 個 RPU 會以 Redshift Serverless 隨需費率計費。
- 在第三個小時，工作群組 1 繼續取用 32 個 RPU，工作群組 2 則減少至 8 個 RPU。在這個小時內，會以 64 個 RPU 無伺服器保留費率對帳戶收費，即使帳戶總計為 40 個 RPU。

請參閱下圖，了解工作群組用量演變，以及隨需和無伺服器保留費率帳單詳細資訊：



使用 CLI AWS 或 Amazon Redshift API 購買無伺服器保留

您可以使用 `create-reservation` 來建立 RPU 保留。以下顯示命令：

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

您將 `capacity` 設定為要保留的 RPU 數量。

儲存的帳單

主要儲存容量會以 Redshift 受管儲存 (RMS) 來計費。儲存會按每月 GB 數計費。儲存帳單不同於運算容量帳單。用於手動快照的儲存體會根據所有作用中手動快照的總唯一資料區塊，以標準備份計費費率計費。保留不到 24 小時的復原點不會收費。

資料傳輸成本和機器學習 (ML) 成本會分開收取，情況與佈建叢集相同。 AWS 區域間的快照複寫和資料共用會按照定價頁面上概述的傳輸速率計費。如需詳細資訊，請參閱 Amazon Redshift 定價。

使用 CloudWatch 將帳單用量視覺化

系統會產生追蹤快照儲存用量的指標 `ServerlessUniqueSnapshotStorage`，並將其傳送至 CloudWatch。如需 CloudWatch 的相關資訊，請參閱什麼是 Amazon CloudWatch？

使用 Amazon Redshift Serverless 免費試用

Amazon Redshift Serverless 會提供免費試用。如果您參與免費試用，便可在 Redshift 主控台中檢視免費試用額度餘額，並在 SYS_SERVERLESS_USAGE 系統檢視中查看免費試用用量。請注意，免費試用用量的帳單詳細資訊不會出現在帳單主控台中。免費試用結束後，您只能在帳單主控台中檢視用量。如需 Amazon Redshift Serverless 免費試用的相關資訊，請參閱 Amazon Redshift Serverless 免費試用。

帳單用量注意事項

- 記錄用量 — 查詢或交易只會在交易完成、回復或停止後才進行計量和記錄。例如，如果交易執行兩天，系統便會在交易完成後記錄 RPU 用量。您可以透過查詢 `sys_serverless_usage` 即時監控持續的使用情況。系統可能會以特定小時和每日使用的 RPU 用量變化和效果成本的形式來反映交易記錄。
- 寫入明確交易 — 請務必將結束交易作為最佳實務。如果您沒有結束或回復開啟的交易，Amazon Redshift Serverless 會繼續使用 RPU。例如，如果你寫入明確的 `BEGIN TRAN`，則務必要有相應的 `COMMIT` 和 `ROLLBACK` 陳述式。
- 已取消的查詢 — 如果您執行查詢並在查詢完成前將其取消，系統仍會就查詢執行過的時間向您收費。
- 擴展 — Amazon Redshift Serverless 執行個體可能會啟動擴展來處理負載較高的時段，以維持一致的效能。您的 Amazon Redshift Serverless 帳單包含相同 RPU 費率的基本運算容量和擴展的容量。
- 縮減規模 — Amazon Redshift Serverless 會從其基本 RPU 容量縱向擴展，以處理負載較高的時段。在某些情況下，RPU 容量可能會在查詢負載下降後於一段時間內保持較高的設定。建議您在主控台中設定 RPU 時數上限，以免產生意外成本。
- 系統資料表 — 當您查詢系統資料表時，系統會就查詢時間計費。
- Redshift Spectrum — 當您有 Amazon Redshift Serverless 並執行查詢時，資料湖查詢不需要另外付費。對儲存在 Amazon S3 中的資料所進行的查詢，按交易時間計算的費用與查詢本機資料時相同。
- 聯合查詢 — 聯合查詢會以特定時間間隔內所使用的 RPU 來收費，方式與資料倉儲或資料湖上的查詢相同。
- 儲存 — 儲存會以每月 GB 數另外計費。
- 最低費用 — 最低費用為 60 秒的資源用量，以每秒為單位來計量。
- 快照帳單 — 快照帳單不會變更。系統會根據儲存來收費，並以每月 GB 費率計費。您可以免費地將資料倉儲還原到過去 24 小時內的特定時間點 (可達 30 分鐘的精細程度)。如需詳細資訊，請參閱 Amazon Redshift 定價。

- 使用額外運算資源執行自動最佳化 - Amazon Redshift Serverless 通常會與使用者查詢一起執行自動最佳化操作。這些操作稱為自主操作，您不需要付費。

  如果您啟用配置額外的運算資源，即使使用者活動量很高，Amazon Redshift 也會在必要時執行自主運算。在這種情況下，您可以支付執行自律運動所花費的時間。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的為自動資料庫最佳化配置額外的運算資源。

保持帳單可預測性的 Amazon Redshift Serverless 最佳實務

以下是有助於保持帳單一致性的最佳實務和內建設定。

- 確實結束每個交易。當您使用 `BEGIN` 開始交易時，請務必也將其 `END`。
- 使用最佳實務錯誤處理來適當地回應錯誤並結束每個交易。盡量減少開啟的交易有助於避免不必要的 RPU 用量。
- 使用 `SESSION TIMEOUT` 來協助結束開啟的交易和閒置的工作階段。其會導致任何閒置或非作用中時間超過 3600 秒 (1 小時) 的工作階段逾時。其會導致任何保持開啟和非作用中狀態超過 21600 秒 (6 小時) 的交易逾時。您可以針對特定使用者明確變更此逾時設定，例如當您想要為長時間執行的查詢保持工作階段開啟狀態時。CREATE USER 主題會顯示如何調整使用者的 `SESSION TIMEOUT`。
  - 在大多數情況下，建議您不要延長 `SESSION TIMEOUT` 值，除非您有特別需要這麼做的使用案例。如果工作階段仍處於閒置狀態，具有開啟中的交易，則可能會導致系統使用 RPU，直到工作階段關閉為止。這會導致不必要的成本。
  - Amazon Redshift Serverless 的執行中查詢時間上限為 86,399 秒 (24 小時)。開啟中交易的非作用中期間上限為六小時，超過之後，Amazon Redshift Serverless 就會結束與交易相關聯的工作階段。如需詳細資訊，請參閱Amazon Redshift Serverless 物件的配額。

具有連線集區的 Amazon Redshift Serverless 帳單

Amazon Redshift Serverless 會將所有傳入查詢視為可計費使用者活動，包括連線集區傳送的輕量型運作狀態檢查查詢。無論查詢是來自應用程式、JDBC/ODBC 驅動器，還是連線集區架構，此行為都適用。每個運作狀態檢查查詢都會觸發運算用量，而且無論查詢用途或出處為何，都會產生費用。因此，即使沒有實際的使用者工作負載在執行，讓連線集區保持開啟仍會產生成本。

連線集區會在應用程式與 Amazon Redshift Serverless 端點之間維護持久的連線集區。為了確保這些連線保持正常運作且可用，集區機制通常會定期傳送輕量型或空的查詢 (例如 `SELECT 1`)。這些自動化查詢會驗證連線狀態。

當您使用連線集區時，請參考以下這些最佳實務，以盡可能減少產生意外的費用：

- 藉由檢閱和最佳化連線集區組態中的運作狀態檢查或保持連線查詢的頻率，以調整運作狀態檢查頻率。
- 藉由設定連線集區來最佳化閒置系統設定，以盡可能減少系統閒置期間不必要的連線流失或背景查詢活動。
- 實作可減少開銷的應用程式層級集區或更有效率的連線生命週期管理。
- 如果您的連線集區組態允許的話，請停用活動訊號或驗證查詢。查看您特定的連線字串參數或組態檔案，以調整這些設定。
- 微調 TCP 保持連線設定：如果您無法停用驅動器的內部活動訊號機制，請在作業系統或應用程式層級調整傳輸控制通訊協定 (TCP) 保持連線設定，以解決連線逾時問題。如需詳細資訊，請參閱您的作業系統、JDBC/ODBC 驅動器或連線集區文件。
- 最佳化資料庫連線集區：設定您的連線集區 (HikariCP、Apache Database Connection Pool)，以管理連線並盡可能降低連線的額外負荷。專注於參數，例如最大連線數、閒置逾時和驗證查詢 (如有需要)。此最佳化有助於使 Amazon Redshift Serverless 運算用量與實際工作負載需求保持一致，進而可能降低成本。

採用零 ETL 的 Amazon Redshift Serverless 成本最佳化

若要在 Amazon Redshift Serverless 上執行零 ETL 整合時最佳化成本，您可以最適化調整環境，並根據工作負載需求調整重新整理設定。請考慮進行下列調整：

- 使用適用於工作負載的 8 個 RPU 較低基本 RPU 容量。
- 設定目標 Redshift 執行個體的 REFRESH_INTERVAL，以平衡最新狀態與成本。較短的間隔可確保近乎即時的更新，但會增加運算成本。較長的間隔 (5 分鐘或更長) 可減少不需即時保持最新狀態的工作負載費用，例如報告或歷史分析。若要編輯 Redshift 目標 REFRESH_INTERVAL，請參閱 ALTER DATABASE 描述中的重新整理間隔子句。
- 藉由並行執行分析工作負載，同時擷取零 ETL 資料的方式，充分利用 Amazon Redshift Serverless 環境。這可確保運算容量主動提供給多種業務用途。

連線至 Amazon Redshift Serverless

在設定 Amazon Redshift Serverless 執行個體後，便可以使用以下概述的各種方法連線到該執行個體。如果您有多個團隊或專案，並且想要分開管理成本，則可以使用不同的 AWS 帳戶。

如需 Amazon Redshift Serverless 可用 AWS 區域 位置的清單，請參閱《》中針對 Redshift Serverless API 列出的端點Amazon Web Services 一般參考。

Amazon Redshift Serverless 會連線到目前 中 AWS 帳戶 的無伺服器環境 AWS 區域。Amazon Redshift Serverless 會在連接埠範圍為 5431 至 5455 和 8191 至 8215 的連接埠範圍內的 VPC 中執行。預設值為 5439。目前，您只能使用 API 操作 UpdateWorkgroup和 AWS CLI 操作 變更連接埠update-workgroup。

連線至 Amazon Redshift Serverless

您可以使用下列語法連線到 Amazon Redshift Serverless 中的資料庫 (名為 dev)。

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

例如，下列連線字串會指定區域 us-east-1。

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

透過 JDBC 驅動程式連線至 Amazon Redshift Serverless

您可以使用下列其中一種方法，透過 Amazon Redshift 所提供的 JDBC 驅動器 2.x 版驅動器，以偏好的 SQL 用戶端連線至 Amazon Redshift Serverless。

若要透過 JDBC 驅動器 2.x 版，使用登入憑證驗證資料庫以進行連線，請使用下列語法。您可以選擇是否使用連接埠號碼；如果未包含，Amazon Redshift Serverless 會預設為連接埠號碼 5439。您可以變更為 5431 至 5455 或 8191 至 8215 之連接埠範圍內的其他連接埠。若要變更無伺服器端點的預設連接埠，請使用 AWS CLI 和 Amazon Redshift API。

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

例如，下列連線字串會指定工作群組預設值、帳戶 ID 123456789012 以及 us-east-2 區域。

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

若要使用 JDBC 驅動器 2.x 版與 IAM 連線，請使用下列語法。您可以選擇是否使用連接埠號碼；如果未包含，Amazon Redshift Serverless 會預設為連接埠號碼 5439。您可以變更為 5431 至 5455 或 8191 至 8215 之連接埠範圍內的其他連接埠。若要變更無伺服器端點的預設連接埠，請使用 AWS CLI 和 Amazon Redshift API。

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

例如，下列連線字串會指定工作群組預設值、帳戶 ID 123456789012 以及 us-east-2 區域。

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

若為 ODBC，請使用下列語法。

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

如果您使用的是 2.1.0.9 之前的 JDBC 驅動程式版本，並使用 IAM 來連線，則需要使用下列語法。

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

例如，下列連線字串會指定工作群組預設值和 AWS 區域 us-east-1。

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

如需驅動程式的相關資訊，請參閱在 Amazon Redshift 中設定連線。

尋找 JDBC 和 ODBC 連線字串

若要使用 SQL 用戶端工具來連線至工作群組，您必須有 JDBC 或 ODBC 連線字串。您可以在 Amazon Redshift Serverless 主控台的工作群組詳細資訊頁面上找到連線字串。

尋找工作群組的連線字串

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單上，選擇 Redshift Serverless。
3. 在導覽選單上，選擇工作群組組態，然後從清單中選擇工作群組名稱以開啟其詳細資訊。
4. General information (一般資訊) 區段中提供了 JDBC URL 和 ODBC URL 連線字串，以及其他詳細資訊。每個字串都以工作群組執行 AWS 的區域為基礎。選擇適當連線字串旁邊的圖示以複製連線字串。

使用資料 API 連線至 Amazon Redshift Serverless

您也可以使用 Amazon Redshift 資料 API 來連線至 Amazon Redshift Serverless。在 AWS CLI 呼叫中使用 `workgroup-name` 參數，而非 `cluster-identifier` 參數。

如需資料 API 的相關資訊，請參閱使用 Amazon Redshift 資料 API。如需會在 Python 中呼叫資料 API 的範例程式碼和其他範例，請參閱開始使用 Redshift 資料 API 並查看 GitHub 中的 `quick-start` 和 `use-cases` 資料夾。

使用 SSL 連線到 Amazon Redshift Serverless

設定與 Amazon Redshift Serverless 的安全連線

為了支援 SSL 連線，Redshift Serverless 會在每一個工作群組上建立並安裝 AWS Certificate Manager (ACM) 發行的 SSL 憑證。ACM 憑證受到大多數作業系統、網頁瀏覽器和用戶端公開信任。如果 SQL 用戶端或應用程式在 `sslmode` 連線選項設為 `require`、`verify-ca` 或 `verify-full` 的情況下，使用 SSL 連線到 Redshift Serverless，您可能需要下載憑證套件。如果用戶端需要憑證，Redshift Serverless 會提供如下的憑證套件：

- 從 https://s3.amazonaws.com/redshift-downloads/amazon-trust-ca-bundle.crt 下載套件。
  - 預期的 MD5 檢查總和碼為 418dea9b6d5d5de7a8f1ac42e164cdcf。
  - sha256 檢查總和碼為 36dba8e4b8041cd14b9d60158893963301bcbb92e1c456847784de2acb5bd550。

  請勿使用位於 `https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt` 的先前憑證套件。
- 在中國 AWS 區域，請從 https：//https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/amazon-trust-ca-bundle.crt 下載套件。
  - 預期的 MD5 檢查總和碼為 418dea9b6d5d5de7a8f1ac42e164cdcf。
  - sha256 檢查總和碼為 36dba8e4b8041cd14b9d60158893963301bcbb92e1c456847784de2acb5bd550。

  請勿使用位於 `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` 的先前憑證套件。

> **Important**
>
> Redshift Serverless 已變更 SSL 憑證的管理方式。您可能需要更新目前的信任根 CA 憑證，才能繼續使用 SSL 連線到工作群組。如需用於 SSL 連線的 ACM 憑證的詳細資訊，請參閱 將 SSL 連線轉換為使用 ACM 憑證。

根據預設，不論連線是否使用 SSL，工作群組資料庫都接受連線。

若要建立只接受 SSL 連線的新工作群組，請使用 `create-workgroup` 命令並將 `require_ssl` 參數設定為 `true`。若要使用下列範例，請將 *yourNamespaceName* 取代為您的命名空間名稱，並將 *yourWorkgroupName* 取代為您的工作群組名稱。

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

若要更新只接受 SSL 連線的現有工作群組，請使用 `update-workgroup` 命令並將 `require_ssl` 參數設定為 `true`。請注意，當您更新 `require_ssl` 參數時，Redshift Serverless 會重新啟動您的工作群組。若要使用下列範例，請將 *yourWorkgroupName* 取代為您的工作群組名稱。

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

Amazon Redshift 支援 Elliptic Curve Diffie—Hellman Ephemeral (ECDHE) 金鑰協議通訊協定。使用 ECDHE 時，用戶端和伺服器各有橢圓曲線公有-私有金鑰對，用於透過不安全通道建立共用秘密。您不需要在 Amazon Redshift 中設定任何項目來啟用 ECDHE。如果您從 SQL 用戶端工具連線，而此工具使用 ECDHE 來加密用戶端和伺服器之間的通訊，Amazon Redshift 會使用提供的密碼清單來建立適當的連線。如需詳細資訊，請參閱 Wikipedia 上的 Elliptic curve diffie—hellman 和 OpenSSL 網站上的 Ciphers。

設定與 FIPS 相容的 Amazon Redshift Serverless SSL 連線

若要建立使用與 FIPS 相容的 SSL 連線的新工作群組，請使用 `create-workgroup` 命令並將 `use_fips_ssl` 和 `require_ssl` 參數設定為 `true`。若要使用下列範例，請將 *yourNamespaceName* 取代為您的命名空間名稱，並將 *yourWorkgroupName* 取代為您的工作群組名稱。

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

若要更新現有工作群組以使用與 FIPS 相容的 SSL 連線，請使用 `update-workgroup` 命令並將 `use_fips_ssl` 和 `require_ssl` 參數設定為 `true`。請注意，當您更新 `use_fips_ssl` 參數時，Redshift Serverless 會重新啟動您的工作群組。若要使用下列範例，請將 *yourWorkgroupName* 取代為您的工作群組名稱。

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

如需設定 Redshift Serverless 以使用與 FIPS 相容的連線的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 use_fips_ssl。

從 Amazon Redshift 受管 VPC 端點連線至 Amazon Redshift Serverless

從其他 VPC 端點連線至 Amazon Redshift Serverless

如需為 Amazon Redshift Serverless 工作群組設定受管 VPC 端點的相關資訊，請參閱使用 Redshift 受管 VPC 端點。

從 VPC 端點介面 (AWS PrivateLink) 連線至 Amazon Redshift Serverless

如需從 VPC 端點介面 (AWS PrivateLink) 連線至 Amazon Redshift Serverless 的相關資訊，請參閱 介面 VPC 端點。

從其他帳戶中的 Redshift VPC 端點連線至 Amazon Redshift Serverless

從跨 VPC 端點連線至 Amazon Redshift Serverless

Amazon Redshift Serverless 會佈建在 VPC 中。您可以授予對其他帳戶中 VPC 的存取權，以存取帳戶中的 Amazon Redshift Serverless。這類似於來自受管理 VPC 端點的連線，但在此情況下，連線起始於另一個帳戶中的資料庫用戶端。您可以執行幾項作業：

- 資料庫擁有者可以將包含 Amazon Redshift Serverless 的 VPC 的存取權授予相同區域中的其他帳戶。
- 資料庫擁有者可以撤銷 Amazon Redshift Serverless 存取權。

跨帳戶存取的主要優點是允許更輕鬆的資料庫協同作業。不必在包含資料庫的帳戶中佈建使用者即可存取資料庫，這樣可減少設定步驟並節省時間。

授予對其他帳戶中 VPC 的存取

若要授予存取權或變更允許的存取權，授予者需要具有下列權限的已指派權限政策：

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

您可能需要 AWS 受管政策 AmazonRedshiftFullAccess 中指定的其他許可。如需詳細資訊，請參閱授予 Amazon Redshift Serverless 的權限。

承授者需要具有下列權限的已指派權限政策：

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

我們建議的最佳實務是，將許可政策附加到 IAM 角色，然後根據需要將其指派給使用者和群組。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。

這是用來設定跨 VPC 存取權的範例資源政策：

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"
                    ]
                }
            }
        }
    ]
}
```

本節接下來的程序假設執行這些權限的使用者具有適當的指派權限，例如，透過具有列出權限的已指派 IAM 角色。這些程序還假設工作群組具有附加適當資源權限的 IAM 角色。

使用主控台授予其他帳戶的 VPC 存取權

此程序顯示當您是資料庫擁有者且想要授予存取權時，用於設定資料庫存取權的步驟。

從擁有者帳戶授予存取權

1. 在 Amazon Redshift Serverless 工作群組的屬性中，在資料存取索引標籤上有一份名為已授予的帳戶清單。它會顯示授予工作群組存取權的帳戶和 VPC。找到清單，然後選擇授予存取權，將帳戶新增至清單。
2. 出現一個視窗，您可以在其中新增承授者資訊。輸入 AWS 帳戶的 ID，其為您要授予存取權的帳戶的 12 位數 ID。
3. 授予承授者的所有 VPC 或特定 VPC 的存取權。如果僅授予特定 VPC 的存取權，則可以透過輸入每個 VPC 並選擇新增 VPC來為這些 VPC 新增 ID。
4. 完成後，選擇儲存變更。

當您儲存變更時，帳戶會出現在已授予的帳戶清單中。此項目會顯示帳戶 ID 和授予存取權的 VPC 清單。

資料庫擁有者也可以撤銷帳戶的存取權。您可以隨時撤消存取權。

撤銷帳戶存取權

1. 您可以從授予的帳戶清單開始。首先，選取一或多個帳戶。
2. 選擇撤銷存取權。

授予存取權之後，承授者的資料庫管理員可以檢查主控台，以判斷他們是否具有存取權。

使用主控台確認已授予存取其他帳戶的存取權

1. 在 Amazon Redshift Serverless 工作群組屬性的資料存取索引標籤上，有一份名為授權的帳戶清單。它顯示可以從此工作組存取的帳戶。承授者無法使用工作群組的端點 URL 直接存取工作群組。若要存取工作組，身為承授者的您移至端點區段，然後選擇建立端點。
2. 然後，身為承授者，您可以提供端點名稱和 VPC 來存取工作群組。
3. 成功建立端點後，它會出現在端點區段中，並且有端點 URL。您可以使用此端點 URL 來存取工作群組。

使用 CLI 命令授予其他帳戶的存取權

授予存取權的帳戶必須先授予其他帳戶的存取權，才能使用連線`put-resource-policy`。資料庫擁有者可以呼叫 `put-resource-policy` 以授權其他帳戶，建立與工作群組的連線。接著，承授者帳戶便可透過其允許的 VPC，使用 `create-endpoint-authorization` 以建立與工作群組的連線。

以下顯示的內容適用於 put-resource-policy，您可以呼叫這些內容以允許存取特定帳戶和 VPC。

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

呼叫命令後，您可以呼叫 get-resource-policy，並指定 resource-arn 以查看允許哪些帳戶和 VPC 存取資源。

承授者可以進行以下呼叫。它顯示有關授予存取權的資訊。具體而言，它會傳回一份包含授予存取權的 VPC 清單。

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

這樣做的目的是讓承授者從授予帳戶取得有關端點授權的資訊。owner-account 是共用帳戶。當您執行此項目時，它會傳回每個工作群組的 CrossAccountVpcs，即為允許的 VPC 清單。以下內容展示了工作群組可用的所有屬性以做為參考：

```
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**
>
> 提醒您，叢集重新定位不是設定其他 Redshift 網路功能的先決條件。您也不需要將其開啟才能啟用下列功能：

- 從跨帳戶或跨區域 VPC 連線至 Redshift – 您可以從一個 AWS 虛擬私有雲端 (VPC) 連線到包含 Redshift 資料庫的另一個，如本節所述。
- 設定自訂網域名稱 — 您可以為 Amazon Redshift 叢集或 Amazon Redshift Serverless 工作群組建立自訂網域名稱 (也稱為自訂 URL)，以使端點名稱更加簡單易記。如需詳細資訊，請參閱使用自訂網域名稱來進行用戶端連線。

其他資源

設定網路流量設定的說明可在具有預設或自訂安全群組組態的公開存取性中取得。其中包括可公開存取叢集的使用案例。

設定網路流量設定的說明可在具有預設或自訂安全群組組態的私有存取性中取得。其中包括叢集無法在網際網路上使用的使用案例。

如需 Amazon Redshift Serverless 安全連線的相關資訊 (包括授予許可、授權其他服務的存取權，以及建立 IAM 角色)，請參閱 Amazon Redshift Serverless 中的身分和存取管理。

在 Amazon Redshift Serverless 中定義要向聯合身分使用者授予的資料庫角色

當您是組織的一分子時，您會有一組相關聯的角色。例如，你有你工作職能的角色，如程式設計人員和管理員。您的角色決定了您可以存取哪些應用程式和資料。大多數組織會使用身分提供者 (例如 Microsoft Active Directory) 將角色指派給使用者和群組。使用角色來控制資源存取的情況變多了，因為組織不必花太多時間管理個別使用者。

Amazon Redshift Serverless 最近引進了角色型存取控制功能。您可以使用資料庫角色來保護對於資料和物件 (例如結構描述或資料表) 的存取。或者，您也可以使用角色來定義一組提升的許可，例如針對系統監控或資料庫管理員來定義。但在向資料庫角色授予資源許可後，還要再進行一個步驟，那就是將使用者的角色從組織連線到資料庫角色。您可以透過執行 SQL 陳述式，在每個使用者初次登入時向其指派資料庫角色，但這需要大量工作。比較簡單的方法是定義要向其授予的資料庫角色，並將其傳遞至 Amazon Redshift Serverless。這麼做的好處是可以簡化初始登入程序。

您可以使用 `GetCredentials` 將角色傳遞至 Amazon Redshift Serverless。當使用者第一次登入 Amazon Redshift Serverless 資料庫時，系統會建立相關聯的資料庫使用者，並將其對應至相符的資料庫角色。本主題會詳細說明將角色傳遞至 Amazon Redshift Serverless 的機制。

傳遞資料庫角色有幾個主要的使用案例：

- 當使用者透過第三方身分提供者 (一般皆已設定聯合功能) 來登入，並透過工作階段標籤傳遞角色時。
- 當使用者透過 IAM 登入憑證來登入，並透過標籤索引鍵和值傳遞其角色時。

如需角色型存取控制的相關資訊，請參閱角色型存取控制 (RBAC)。

定義資料庫角色

您必須先在資料庫中設定資料庫角色，並向其授予適當的資料庫資源許可，然後才能將角色傳遞至 Amazon Redshift Serverless。例如，在簡單的案例中，您可以建立名為 sales 的資料庫角色，並向其授予有銷售資料之資料表的查詢存取權。如需如何建立資料庫角色和授予許可的相關資訊，請參閱 CREATE ROLE 和 GRANT。

定義要向聯合身分使用者授予之資料庫角色的使用案例

以下各節概述了將資料庫角色傳遞至 Amazon Redshift Serverless 可以簡化資料庫資源存取方式的幾個使用案例。

使用身分提供者來登入

第一個使用案例假設您的組織在身分與存取管理服務中具有使用者身分。這項服務可以基於雲端 (例如 JumpCloud 或 Okta)，也可以基於內部部署 (如 Microsoft Active Directory)。目標是在使用者登入用戶端 (例如查詢編輯器 V2) 或使用 JDBC 用戶端時，自動將使用者得自身分提供者的角色對應至資料庫角色。若要進行這方面的設定，您必須先完成幾個組態任務。這些索引標籤包括以下項目：

1. 使用信任關係設定與身分提供者 (IdP) 的聯合整合。這是先決條件。當您完成設定時，身分提供者便會負責透過 SAML 聲明驗證使用者並提供登入憑證。如需詳細資訊，請參閱將第三方 SAML 解決方案供應商與 整合 AWS。您也可以在將 Amazon Redshift 查詢編輯器 V2 的存取與 Active Directory Federation Services (AD FS) 聯合或將 Amazon Redshift 查詢編輯器 v2 的單一登入存取與 Okta 聯合 中找到更多資訊。
2. 使用者必須具有下列政策許可：
   - `GetCredentials` — 提供憑證以供臨時授權登入 Amazon Redshift Serverless。
   - `sts:AssumeRoleWithSAML` – 提供將企業身分存放區或目錄繫結至角色型 AWS 存取的機制。
   - `sts:TagSession` — 身分提供者主體上的標籤工作階段動作許可。

   在這種情況下，`AssumeRoleWithSAML` 會透過 SAML 驗證回應，針對已驗證過的使用者傳回一組安全憑證。此操作提供一種機制，可將身分存放區或目錄繫結至角色型 AWS 存取，而不需要使用

者特定的登入資料。對於具有 AssumeRoleWithSAML 許可的使用者，身分提供者會負責管理用來傳遞角色資訊的 SAML 聲明。

我們建議的最佳實務是，將許可政策連接到 IAM 角色，然後根據需要將其指派給使用者和群組。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。

3. 您可以 role1:role2 格式的冒號分隔角色值來設定 RedshiftDbRoles 標籤。例如 manager:engineer。這些項目可以從身分提供者中所設定的 session-tag 實作來擷取。SAML 驗證請求會以程式設計方式傳遞角色。如需傳遞工作階段標籤的相關資訊，請參閱在 AWS STS中傳遞工作階段標籤。

   如果您傳遞的角色名稱不存在於資料庫中，系統會將其忽略。

在這個使用案例中，當使用者使用聯合身分登入時，系統會透過工作階段標籤的索引鍵和值在授權請求中傳遞其角色。緊接在授權之後，GetCredentials 便會將角色傳遞至資料庫。連線成功後，資料庫角色便會對應，而且使用者可以執行與其角色對應的資料庫任務。操作的基本部分是系統會在初始授權請求中向 RedshiftDbRoles 工作階段標籤指派角色。如需傳遞工作階段標籤的相關資訊，請參閱使用 AssumeRoleWithSAML 傳遞工作階段標籤。

使用 IAM 憑證進行登入

在第二個使用案例中，我們可以向使用者傳遞角色，讓其透過 IAM 憑證存取資料庫的用戶端應用程式。

1. 在此情況下登入的使用者必須獲派下列動作的政策許可：
   - tag:GetResources — 傳回與指定標籤相關聯的已標記資源。
   - tag:GetTagKeys — 傳回目前使用中的標籤索引鍵。

     我們建議的最佳實務是，將許可政策連接到 IAM 角色，然後根據需要將其指派給使用者和群組。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。
2. 我們也需要用來存取資料庫服務 (例如 Amazon Redshift Serverless) 的允許許可。
3. 在這個使用案例中，請在 AWS Identity and Access Management中設定角色的標籤值。您可以選擇編輯標籤，並建立名為 RedshiftDbRoles 的標籤索引鍵，以及包含角色的隨附標籤值字串。例如，manager:engineer。

當使用者登入時，系統會將其角色新增至授權請求並傳遞給資料庫。其會對應至現有的資料庫角色。

其他資源

如使用案例所述，您可以設定 IdP 和 AWS之間的信任關係。如需詳細資訊，請參閱使用依賴方信任設定您的 SAML 2.0 IdP 並新增宣告。

Amazon Redshift Serverless 中的身分和存取管理

存取 Amazon Redshift 需要 AWS 登入資料，可用來驗證您的請求。這些登入資料必須具有存取 AWS 資源的許可，例如 Amazon Redshift Serverless。

下列各節提供如何使用 AWS Identity and Access Management (IAM) 和 Amazon Redshift 的詳細資訊，藉由控制可存取的人員，協助保護您的資源。如需詳細資訊，請參閱Amazon Redshift 中的身分和存取管理。

授予許可給 Amazon Redshift Serverless

若要存取其他 AWS 服務，Amazon Redshift Serverless 需要許可。有些 Amazon Redshift 功能需要 Amazon Redshift 代表您存取其他 AWS 服務。為了讓 Amazon Redshift Serverless 執行個體為您服務，請為其提供安全憑證。提供安全登入資料的偏好方法是指定 AWS Identity and Access Management (IAM) 角色。您也可以透過 Amazon Redshift 主控台建立 IAM 角色，並將其設定為預設值。如需詳細資訊，請參閱建立 IAM 角色做為 Amazon Redshift 的預設角色。

若要存取其他 AWS 服務，請建立具有適當許可的 IAM 角色。您還需要將該角色與 Amazon Redshift Serverless 相關聯。此外，您可以在執行 Amazon Redshift 命令時指定角色的 Amazon Resource Name (ARN)，或指定關鍵字 `default`。

在 https://console.aws.amazon.com/iam/ 中變更 IAM 角色的信任關係時，請務必確定其包含 `redshift-serverless.amazonaws.com` 和 `redshift.amazonaws.com` 作為主體服務名稱。如需有關如何管理 IAM 角色以代表您存取其他 AWS 服務的資訊，請參閱 授權 Amazon Redshift 代表您存取 AWS 服務。

建立 IAM 角色做為 Amazon Redshift 的預設角色

當您透過 Amazon Redshift 主控台建立 IAM 角色時，Amazon Redshift 會以程式設計方式在您的 中建立角色 AWS 帳戶。Amazon Redshift 也會自動將現有的 AWS 受管政策連接到這些政策。此方法意味著您可以保留在 Amazon Redshift 主控台內，而不必切換到 IAM 主控台來建立角色。

您透過主控台為叢集建立的 IAM 角色會自動附加 `AmazonRedshiftAllCommandsFullAccess` 受管政策。此 IAM 角色可讓 Amazon Redshift 複製、卸載、查詢和分析 IAM 帳戶中 AWS 資源的資

料。相關命令包括 COPY、UNLOAD、CREATE EXTERNAL FUNCTION、CREATE EXTERNAL TABLE、CREATE EXTERNAL SCHEMA、CREATE MODEL 和 CREATE LIBRARY。如需如何建立 IAM 角色作為 Amazon Redshift 預設值的相關資訊，請參閱建立 IAM 角色做為 Amazon Redshift 的預設角色。

若要開始建立 IAM 角色做為 Amazon Redshift 的預設角色，請開啟 AWS 管理主控台，選擇 Amazon Redshift 主控台，然後在功能表中選擇 Redshift Serverless。您可以在 Serverless 儀表板中建立新的工作群組。建立步驟會引導您完成選取 IAM 角色或設定新的 IAM 角色的程序。

當您已有 Amazon Redshift Serverless 工作群組且想要為其設定 IAM 角色時，請開啟 AWS 管理主控台。選擇 Amazon Redshift 主控台，然後選擇 Redshift Serverless。在 Amazon Redshift Serverless 主控台上，選擇現有工作群組的命名空間組態。在安全和加密下，您可以編輯許可。

指派 IAM 角色給命名空間

每個 IAM 角色都是具有許可政策的 AWS 身分，可決定每個角色可在 中執行的動作 AWS。角色旨在可由任何需要它的使用者擔任。此外，每個命名空間都是物件的集合，例如資料表和結構描述，以及使用者。當您使用 Amazon Redshift Serverless 時，您可以將多個 IAM 角色與命名空間相關聯。這樣可以讓您更輕鬆地為資料庫物件集合建構適當的許可，以便角色可以對內部和外部資料執行動作。例如，您因此可以在 Amazon Redshift 資料庫中執行 COPY 命令，以從 Amazon S3 擷取資料並填入 Redshift 資料表。

您可以使用主控台將多個角色與命名空間相關聯，如本節先前所述。您也可以使用 API 命令 `CreateNamespace` 或 CLI 命令 `create-namespace`。使用 API 或 CLI 命令時，您可以透過為 `IAMRoles` 填入一或多個角色，將 IAM 角色指派給命名空間。具體而言，您會將特定角色的 ARN 新增至集合。

管理命名空間關聯的 IAM 角色

在 上 AWS 管理主控台 ，您可以管理 中角色的許可政策 AWS Identity and Access Management。您可以使用命名空間組態下的可用設定來管理命名空間的 IAM 角色。如需命名空間及其在 Amazon Redshift Serverless 中的使用方式的相關資訊，請參閱 工作群組和命名空間。

Amazon Redshift IAM 憑證入門

當您第一次登入 Amazon Redshift 主控台並首次嘗試使用 Amazon Redshift Serverless 時，建議您使用已附加 IAM 角色，且該角色具有所需政策的使用者身分來登入。在您開始建立 Amazon Redshift Serverless 執行個體之後，Amazon Redshift 會記錄您用來登入的 IAM 角色名稱。您可以使用相同的憑證登入 Amazon Redshift 主控台和 Amazon Redshift Serverless 主控台。

在建立 Amazon Redshift Serverless 執行個體時，您可以建立資料庫。使用查詢編輯器 v2 透過暫時憑證選項連線至資料庫。

若要新增會針對資料庫保留下來的新管理員使用者名稱和密碼，請選擇自訂管理員使用者憑證，然後輸入新的管理員使用者名稱和管理員使用者密碼。

若要開始使用 Amazon Redshift Serverless，並首次在主控台中建立工作群組和命名空間，請使用已附加許可政策的 IAM 角色。請確定此角色具有管理員許可 `arn:aws:iam::aws:policy/AdministratorAccess` 或已附加至 IAM 政策的完整 Amazon Redshift 許可 `arn:aws:iam::aws:policy/AmazonRedshiftFullAccess`。

下列案例概述您開始使用 Amazon Redshift Serverless 主控台時，Amazon Redshift Serverless 會如何使用您的 IAM 憑證：

- 如果您選擇使用預設設定 — Amazon Redshift Serverless 會將您目前的 IAM 身分轉譯為資料庫超級使用者。您可以將相同的 IAM 身分與 Amazon Redshift Serverless 主控台搭配使用，以在 Amazon Redshift Serverless 的資料庫中執行超級使用者動作。
- 如果您選擇自訂設定而未指定管理員使用者名稱和密碼，則 Amazon Redshift Serverless 會使用您目前的 IAM 憑證作為預設管理員使用者憑證。
- 如果您選擇自訂設定並指定管理員使用者名稱和密碼 Amazon Redshift Serverless — Amazon Redshift Serverless 會將您目前的 IAM 身分轉譯為資料庫超級使用者。Amazon Redshift Serverless 也會建立另一個長期登入使用者名稱和密碼組作為超級使用者。您可以使用目前的 IAM 身分或已建立的使用者名稱和密碼組，以超級使用者身分登入資料庫。

使用資料庫角色許可存取 Amazon Redshift Serverless 資料庫物件

此程序會說明如何透過 Amazon Redshift 資料庫角色授予用來查詢資料表的許可。角色會透過附加至 IAM 中使用者的標籤來指派，並在使用者登入時傳遞至 Amazon Redshift。這是在 Amazon Redshift Serverless 中定義要向聯合身分使用者授予的資料庫角色中的概念範例說明。完成這些步驟的好處是，您可以將使用者與資料庫角色相關聯，避免為每個資料庫物件設定其許可。它可讓您更輕鬆地管理使用者查詢、修改或新增資料到資料表以及執行其他動作的能力。

此程序假設您已經設定 Amazon Redshift Serverless 資料庫，而且您可以在資料庫中授予許可。它還假設您有在 AWS 主控台中建立 IAM 使用者的許可、建立 IAM 角色，以及指派政策許可。

1. 使用 IAM 主控台建立 IAM 使用者 稍後，您會使用此使用者連線到資料庫。
2. 使用查詢編輯器 v2 或其他 SQL 用戶端建立 Redshift 資料庫角色。如需建立資料庫角色的相關資訊，請參閱 CREATE ROLE。

```
CREATE ROLE urban_planning;
```

查詢 SVV_ROLES 系統檢視，以檢查角色是否已建立。它也會傳回系統角色。

```
SELECT * from SVV_ROLES;
```

3. 授予您建立的資料庫角色，以便從資料表中選取。(您建立的 IAM 使用者最終會透過資料庫角色登入並從資料表中選取記錄。) 下列程式碼範例中的角色名稱和資料表名稱為範例。這裡會授予從名為 cities 的資料表中進行選取的許可。

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

4. 使用 AWS Identity and Access Management 主控台建立 IAM 角色。此角色會授予使用查詢編輯器 v2 的許可。建立新的 IAM 角色，然後針對信任的實體類型選擇 AWS 帳戶。然後選擇此帳戶。為該角色提供下列政策許可：

   - AmazonRedshiftReadOnlyAccess
   - tag:GetResources
   - tag:GetTagKeys
   - sqlworkbench 的所有動作，包括 sqlworkbench:ListDatabases 和 sqlworkbench:UpdateConnection。

5. 在 IAM 主控台中，將具有索引鍵 RedshiftDbRoles 的標籤新增至您先前建立的 IAM 使用者。標籤的值應與您在第一個步驟中建立的資料庫角色相符。在範例中，值為 urban_planning。

完成這些步驟後，請指派 IAM 角色給您在 IAM 主控台中建立的使用者。當使用者使用查詢編輯器 v2 登入資料庫時，其在標籤中的資料庫角色名稱會傳遞至 Amazon Redshift 並與其相關聯。因此，其可以透過資料庫角色查詢適當的資料表。為了說明，此範例中的使用者可以透過 urban_planning 資料庫角色查詢 cities 資料表。

將佈建的叢集遷移至 Amazon Redshift Serverless

您可以將現有的佈建叢集移轉至 Amazon Redshift Serverless，以啟用隨需和自動擴展運算資源的功能。將佈建叢集移轉至 Amazon Redshift Serverless 後，您可以僅支付使用的資源，並根據工作負載需求自動擴展容量，以最佳化成本。移轉的常見使用案例包括執行臨機查詢、定期資料處理任務，或在不過度佈建資源的情況下處理無法預測的工作負載。執行下列這組任務，將佈建的 Amazon Redshift 叢集移轉至無伺服器部署選項。

建立已佈建叢集的快照

> **Note**
>
> 當您將已佈建叢集的快照還原到無伺服器命名空間時，Amazon Redshift 會自動將交錯索引鍵轉換為複合索引鍵。

若要將資料從已佈建的叢集傳輸到 Amazon Redshift Serverless，請建立已佈建叢集的快照，然後在 Amazon Redshift Serverless 中還原快照。

> **Note**
>
> 在將資料遷移到無伺服器工作群組之前，請先確保已佈建的叢集需求與您在 Amazon Redshift Serverless 中選擇的 RPU 數量相容。

建立已佈建叢集的快照

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單中，選擇叢集、快照，然後選擇建立快照。
3. 輸入快照定義屬性，然後選擇建立快照。可能要等一會兒才能使用此快照。

若要將已佈建叢集的快照還原至無伺服器命名空間：

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 從 Amazon Redshift 已佈建的叢集主控台開始，然後導覽至叢集、快照頁面。
3. 選擇要使用的快照。
4. 選擇還原快照、還原至無伺服器命名空間。
5. 選擇要作為快照還原目的地的命名空間。
6. 確認您要從快照還原。此動作會使用已佈建叢集中的資料取代無伺服器端點中的所有資料庫。選擇 Restore (還原)。

如需已佈建叢集快照的相關資訊，請參閱 Amazon Redshift 快照。

使用驅動程式連線至 Amazon Redshift Serverless

若要使用偏好的 SQL 用戶端連線至 Amazon Redshift Serverless，您可以使用 Amazon Redshift 所提供的 JDBC 驅動器 2.x 版。我們建議您使用最新版 Amazon Redshift JDBC 驅動器 2.x 版連線至 Amazon Redshift。連接埠號碼是選用的。如果未包含，Amazon Redshift Serverless 會預設為連接埠號碼 5439。您可以變更為 5431 至 5455 或 8191 至 8215 之連接埠範圍內的其他連接埠。若要變更無伺服器端點的預設連接埠，請使用 AWS CLI 和 Amazon Redshift API。

若要尋找用於 JDBC、ODBC 或 Python 驅動程式的確切端點，請參閱 Amazon Redshift Serverless 中的工作群組組態。您也可以使用 Amazon Redshift Serverless API 操作`GetWorkgroup`或 AWS CLI 操作`get-workgroups`來傳回工作群組的相關資訊，然後進行連線。

使用密碼型驗證進行連線

若要使用具有密碼型身分驗證的 Amazon Redshift JDBC 驅動器 2.x 版建立連線，請使用下列語法：

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

若要使用具有密碼型身分驗證的 Amazon Redshift Python 連接器建立連線，請使用下列語法：

```
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
```

若要使用具有密碼型身分驗證的 Amazon Redshift ODBC 驅動器 2.x 版建立連線，請使用下列語法：

```
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 進行連線

如果您偏好使用 IAM 登入，請使用 Amazon Redshift Serverless `GetCredentials` API 操作。

若要使用 IAM 身分驗證，請將 `iam:` 新增至 Amazon Redshift JDBC URL 的 `jdbc:redshift:` 後方，如下列範例所示。

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

此 Amazon Redshift Serverless 端點不支援自訂 dbUser、dbGroup 或自動建立。根據預設，驅動器會在登入時自動建立資料庫使用者。然後，它會根據 IAM 中指定的標籤，或根據身分提供者 (IdP) 中定義的群組，將使用者指派至 Amazon Redshift 資料庫角色。

請確定 AWS 您的身分具有適用於 `redshift-serverless:GetCredentials`動作的正確 IAM 政策。以下是範例 IAM 政策，將正確的許可授予 AWS 身分以連線至 Amazon Redshift Serverless。如需有關 IAM 許可的詳細資訊，請參閱《IAM 使用者指南》中的新增和移除 IAM 身分許可。

JSON

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

若要使用具有 IAM 型身分驗證的 Amazon Redshift Python 連接器建立連線，請在您的程式碼中使用 `iam=true`，如下列語法所示：

```
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
```

對於 IAM credentials，您可以使用任何憑證，包括以下所列：

- AWS 設定檔組態。
- IAM 憑證 (存取金鑰 ID、私密存取金鑰，以及選用的工作階段符記)。
- 身分提供者聯合。

若要使用具有 IAM 型身分驗證的 Amazon Redshift ODBC 驅動器 2.x 版建立連線，請使用下列語法：

```
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;
```

透過 GetClusterCredentials API 使用 IAM 進行連線

> **Note**
>
> 連線至 Amazon Redshift Serverless 時，建議您使用 GetCredentials API。此 API 提供完整的角色型存取控制 (RBAC) 功能，以及 GetClusterCredentials 中還未提供的其他新功能。我們支援 GetClusterCredentials API 來簡化從佈建叢集轉換到無伺服器工作群組的程序，但強烈建議您盡快使用 GetCredentials 進行移轉，以獲得最佳相容性。

您可以使用 GetClusterCredentials API 建立與 Amazon Redshift Serverless 的連線。若要實作此身分驗證方法，請納入下列參數來修改您的用戶端或應用程式：

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

下列範例示範這三種驅動器的 BrowserSAML 外掛程式。這代表數種可用的身分驗證方法之一。這些範例經過修改後，可根據您的特定需求使用替代身分驗證方法或外掛程式。

GetClusterCredentials 的 IAM 政策許可

以下範例 IAM 政策具有使用 GetClusterCredentials 搭配 Amazon Redshift Serverless 時所需的許可：

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}"
            ]
        }
    ]
}
```

若要使用具有 GetClusterCredentials 的 Amazon Redshift JDBC 驅動器 2.x 版建立連線，請使用下列語法：

```
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>"
```

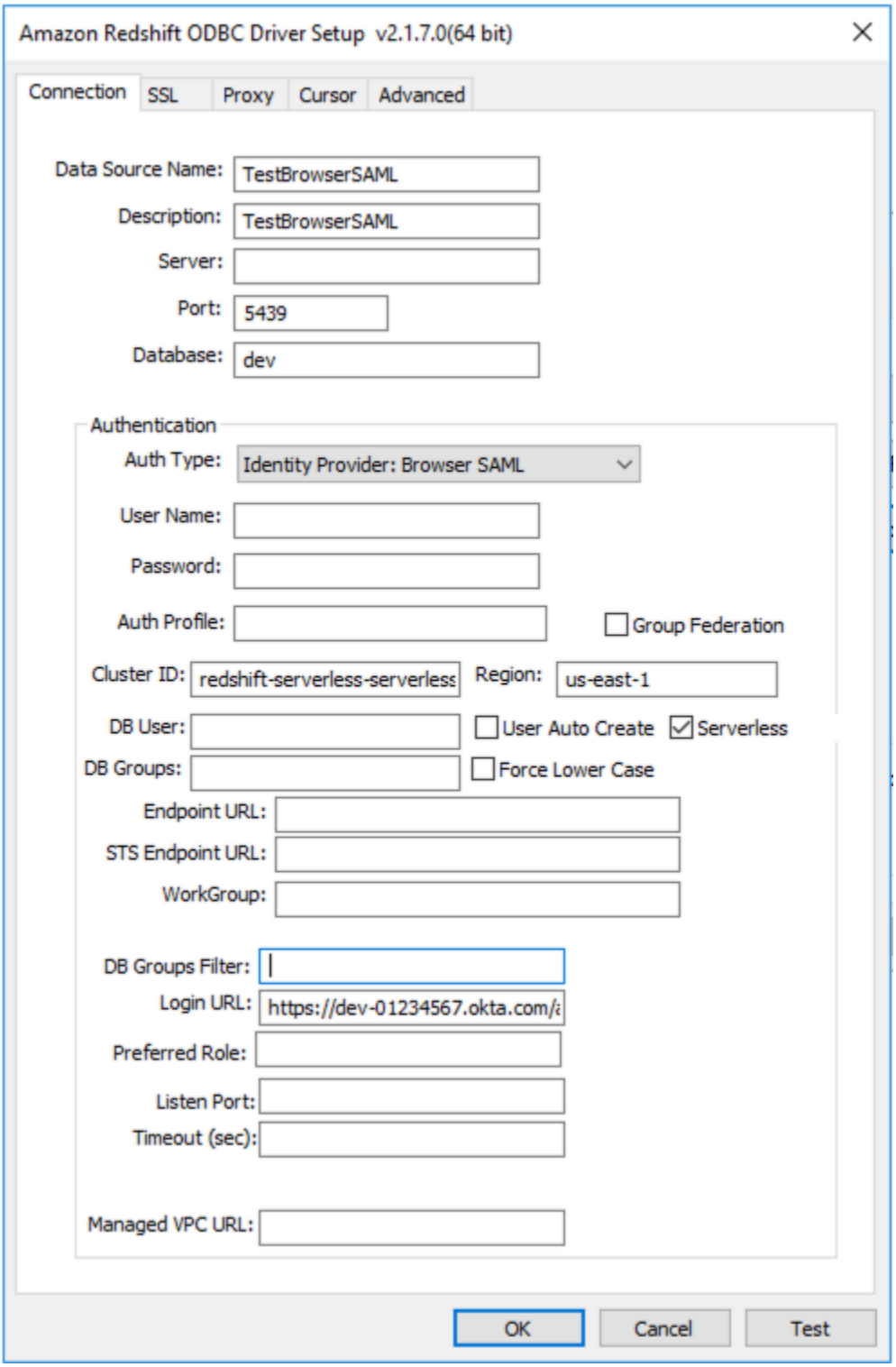
若要使用具有 GetClusterCredentials 的 Amazon Redshift Python 連接器建立連線，請使用下列語法：

```
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
```

若要使用具有 GetClusterCredentials 的 Amazon Redshift ODBC 驅動器 2.x 版建立連線，請使用下列語法：

```
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 ODBC DSN 組態範例：


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 撰寫了任何管理指令碼，則必須使用新的 Amazon Redshift Serverless SDK 來管理 Amazon Redshift Serverless 和相關資源。如需可用 API 操作的相關資訊，請參閱《Amazon Redshift Serverless API 參考指南》。

工作群組和命名空間

若要在 Amazon Redshift Serverless 中隔離工作負載並管理不同的資源，您可以建立命名空間和工作群組，並將儲存和運算資源分開管理。

命名空間是資料庫物件和使用者的集合。儲存體相關的命名空間會組合結構描述、資料表、使用者或 AWS Key Management Service 金鑰來加密資料。儲存屬性包括管理員使用者的資料庫名稱和密碼、許可以及加密與安全。群組到命名空間底下的其他資源包括資料共用、復原點和用量限制。您可以使用 Amazon Redshift Serverless 主控台 AWS Command Line Interface、 或特定資源的 Amazon Redshift Serverless APIs 來設定這些儲存屬性。

工作群組是運算資源的集合。運算相關的工作群組會將運算資源 (例如 RPU、VPC 子網路群組和安全群組) 群組在一起。工作群組的屬性包括網路和安全設定。群組到工作群組底下的其他資源包括存取和用量限制。您可以使用 Amazon Redshift Serverless 主控台 AWS Command Line Interface、 或 Amazon Redshift Serverless APIs 來設定這些運算屬性。

您可以建立一或多個命名空間與工作群組。每個命名空間只能有一個與其相關聯的工作群組。相反地，每個工作群組只能與一個命名空間相關聯。

使用主控台的工作群組和命名空間

設定 Amazon Redshift Serverless 需要逐步完成數個組態步驟。當您遵循步驟來設定 Amazon Redshift Serverless 時，您會建立命名空間和工作群組，並讓其彼此建立關聯。若要使用 Amazon Redshift Serverless 主控台開始設定 Amazon Redshift Serverless 組態，您可以選擇開始使用 Amazon Redshift Serverless 以設定 Amazon Redshift Serverless 並開始與其互動。您可以選擇具有預設設定的環境以加快設定速度，或根據組織的需求明確進行設定。在此過程中，您會指定工作群組和命名空間的設定。

在設定環境後，工作群組屬性和命名空間屬性可協助您熟悉設定。

使用 和 Amazon Redshift Serverless API 的工作群組 AWS Command Line Interface 和命名空間

除了使用 AWS 主控台之外，您也可以使用 AWS CLI 或 Amazon Redshift Serverless API 與工作群組和命名空間互動。下表列出可用來管理快照和復原點的 API 和 CLI 作業。

API 操作	CLI 命令	說明
CreateNamespace	建立命名空間	建立命名空間。根據預設，Amazon Redshift Serverless 會使用預設 AWS Key Management Service 金鑰建立命名空間，但您可以指定另一個金鑰來加密資料。您也可以透過還原快照建立命名空間。如需詳細資訊，請參閱使用快照和復原點。
UpdateNamespace	更新命名空間	更新命名空間。
GetNamespace	獲取命名空間	擷取命名空間的相關資訊。
ListNamepaces	清單命名空間	擷取有關命名空間清單的資訊。
DeleteNamespace	刪除命名空間	刪除命名空間。
CreateWorkgroup	建立工作群組	建立工作群組。建立工作群組時，請確定您具有可與工作群組關聯的現有命名空間。您可以在建立工作群組時指定任何運算資源，例如子網路、安全群組或 RPU。
UpdateWorkgroup	更新工作群組	更新工作群組。
GetWorkgroup	取得工作群組	擷取有關工作群組的資訊。

API 操作	CLI 命令	說明
ListWorkgroups	工作群組清單	擷取有關工作群組清單的資訊。
DeleteWorkgroup	刪除工作群組	刪除工作群組。

工作群組

使用 Amazon Redshift Serverless 即可建立和管理工作群組，以針對不同的工作負載和使用者隔離和控制運算資源。工作群組可讓您設定記憶體和並行擴縮限制等組態選項，並排定工作負載之間執行查詢的優先順序。運算相關的工作群組會將運算資源 (例如 RPU 和 VPC 子網路群組) 群組在一起。

使用命名空間建立工作群組

請完成以下步驟以建立工作群組。如需工作群組組態的詳細資訊，請參閱 工作群組屬性。

1. 選擇 Serverless 儀表板。選擇建立工作群組。
2. 輸入工作群組名稱。
3. 選擇工作群組的 IP 位址類型。選項包括：
   - IPv4 - 使用此選項時，您的 AWS 資源只會透過 IPv4 尋址通訊協定進行通訊。
   - 雙堆疊模式 – 使用此選項，您的 AWS 資源可以透過 IPv4、IPv6 或兩種定址通訊協定進行通訊。此外，您必須將 IPv6 CIDR 區塊與 Amazon VPC 中工作群組所使用的 VPC 和子網路建立關聯。您可以使用 Amazon VPC 主控台來建立 Amazon VPC，或更新現有的 Amazon VPC 以使用 IPv6 尋址。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 VPC 的 IPv6 支援。
4. 選擇 Amazon Redshift Serverless 的虛擬私有雲端 (VPC)。這會將工作群組指派給 AWS 環境中的特定虛擬網路。使用雙堆疊模式時，您選擇的 Amazon VPC 必須支援 IPV6 尋址。如需 VPC 的詳細資訊，請參閱 VPC 和子網路概觀。
5. 選擇一或多個 VPC 安全群組。如需詳細資訊，請參閱使用安全群組控制資源的流量。
6. 選擇是否要為自動最佳化啟用額外的運算資源。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的為自動資料庫最佳化配置額外的運算資源。
7. 在子網路下，指定要與資料庫相關聯的一或多個子網路。這些子網路會包含在您先前選擇的 Amazon VPC 中，且必須位於三個不同的可用區域中。如需詳細資訊，請參閱使用 Amazon Redshift Serverless 時的考量。
8. 選取符合您要求的基本 RPU 容量。

選擇命名空間

1. 選擇建立新的命名空間，然後輸入命名空間名稱，或選擇新增至現有的命名空間，然後從下拉式清單中選取命名空間。
2. 對於資料庫名稱和密碼，指定第一個資料庫的名稱。您也可以透過編輯管理員使用者憑證來指定預設主控台管理員以外的管理員。
3. 對於許可，請選擇關聯 IAM 角色，以將特定 IAM 角色與命名空間和工作群組建立關聯。如需將 IAM 角色與 Amazon Redshift 建立關聯的相關資訊，請參閱 Amazon Redshift 中的身分和存取管理。
4. 您可以透過建立新的金鑰或選擇預設金鑰以外的金鑰來自訂加密設定。對於稽核記錄，選擇要匯出的日誌。每種日誌類型會指定不同的中繼資料 選擇繼續以檢閱您的選擇。

檢閱工作群組選取項目

1. 在檢閱和建立下方檢閱您的設定。其會顯示您在先前步驟中選擇的設定。
2. 選擇儲存。

在建立工作群組後，該工作群組會新增至工作群組清單。

檢視工作群組的屬性

在 Amazon Redshift Serverless 中，工作群組是可用運算資源的集合。當您選擇 Amazon Redshift Serverless 時，您可以在 AWS 主控台中選擇導覽功能表中的工作群組組態，以檢視清單。您可以使用搜尋方塊來尋找符合搜尋條件的工作群組。每個工作群組項目都會顯示一些屬性：

- 工作群組 — 工作群組名稱。您可以選取此項目來檢視和編輯工作群組的屬性。
- 狀態 — 顯示工作群組是否可供使用。
- 命名空間 — 與工作群組相關聯的命名空間。每個工作群組都會與一個命名空間相關聯。
- 建立日期 - 工作群組的建立日期 (UTC)。
- 標籤 - 與工作群組相關聯的標籤。

此外，工作群組組態還有另一個受管工作群組的清單，這些是由 管理的 Amazon Redshift Serverless 工作群組 AWS Glue。如需受管工作群組的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的受管工作群組。

工作群組屬性

您可以透過選擇左側選單中的工作群組組態來列出工作群組。然後，您可以從清單中選擇工作群組。有數個面板會顯示工作群組的屬性。您也可以執行動作。一般資訊會顯示下列項目：

- 工作群組 — 工作群組名稱。
- 命名空間 — 與工作群組相關聯的命名空間。您可以選擇此項目來檢視其屬性。工作群組會與單一命名空間相關聯。
- 建立日期 — 工作群組的建立時間。
- 狀態 — 指出工作群組資源是否可供使用。如果可供使用，您可以透過用戶端連線至 Amazon Redshift Serverless 執行個體，以查詢資料或建立資料庫資源，或者，您也可以使用查詢編輯器 v2 來連線。
- 端點 — URL。
- JDBC URL — 用來建立 JDBC 用戶端連線的 URL。您可以使用此 URL 來與 Amazon Redshift 的 JDBC 驅動程序進行連線。如需詳細資訊，請參閱設定適用於 Amazon Redshift 的 JDBC 驅動器 2.x 版的連線。
- ODBC URL — 用來建立 ODBC 用戶端連線的 URL。此項目包含資料庫和使用者 ID 等屬性以及這些屬性的值。
- 工作群組版本和修補程式版本 - Amazon Redshift Serverless 會定期發行新版本和修補程式。您可以使用工作群組版本和修補程式版本號碼，追蹤 Amazon Redshift Serverless 工作群組的軟體更新。如需特定修補程式中變更和功能的詳細資訊，請參閱 Amazon Redshift 的叢集版本。
- 工作群組 ARN - 工作群組的 Amazon Resource Name。
- 用於自動最佳化的額外運算資源 - Amazon Redshift 是否配置額外的運算資源來執行自動最佳化，即使在大量使用期間也是如此。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的為自動資料庫最佳化配置額外的運算資源。

資料存取索引標籤包含數個面板：

- 網路和安全 - 您可以查看網路屬性，例如虛擬私有雲端 (VPC) 識別碼、VPC 安全群組清單、增強型 VPC 路由、IP 位址類型，以及可公開存取設定。如果您選擇編輯，則可以變更這些設定。此外，您也可以選取開啟增強型 VPC 路由，透過 VPC 來路由傳送無伺服器資料庫和資料儲存庫之間的網路流量，以增強隱私和安全。您也可以選取開啟公開存取，這會讓資料庫可供從 VPC 外部公開存取，讓執行個體和裝置得以連線。

  IP 位址類型可以設定為雙堆疊模式，以同時支援在 IPv4 和 IPv6 上存取工作群組。如需網路層通訊協定網際網路通訊協定 (IP) 的詳細資訊，請參閱維基百科中的網際網路通訊協定。

- Redshift 受管 VPC 端點 — 您可以建立受管 VPC 端點，以便從其他 VPC 存取 Amazon Redshift Serverless。

限制索引標籤具有用於控制 Amazon Redshift Serverless 容量和使用限制的設定。其包含以下面板：

- 以 Redshift 處理單元 (RPU) 為單位的基本容量 — 您可以設定用來處理工作負載的運算資源基本容量。如需詳細資訊，請參閱Amazon Redshift Serverless 的運算容量。
- 用量限制 - 您最多可以為 Amazon Redshift Serverless 執行個體在一段時間內使用的運算資源上限設定四個限制，並選取 Amazon Redshift Serverless 在達到這些限制時要執行的動作。例如，您可以將工作群組設定為兩個限制：500 RPU 小時之一，以及 900 RPU 小時之一。您可以讓 Amazon Redshift Serverless 在達到 500 RPU 小時的第一個限制時傳送警示給您，然後在達到 900 小時的第二個限制時關閉使用者查詢。這些限制有助於控制成本，而更能夠預測成本。
- 查詢限制 — 您可以設定查詢限制，例如逾時設定。這些限制可協助您將成本和效能最佳化。

定位點頁籤具有標籤面板，展示您為工作群組建立的任何標籤。如需標記資源的詳細資訊，請參閱 在 Amazon Redshift Serverless 中標記資源。

受管工作群組屬性

您也可以在受管工作群組清單 AWS Glue Data Catalog 下，選擇由 管理的工作群組。

受管工作群組與一般工作群組擁有不同的屬性。如需受管工作群組的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的受管工作群組。

一般資訊會顯示下列項目：

- 工作群組 - 受管工作群組的名稱。
- 建立日期 - 受管工作群組建立的日期 (UTC)。
- 目錄 ARN - AWS Glue Data Catalog中受管工作群組的 Amazon Resource Name (ARN)。
- 狀態 - 指出受管工作群組的運算資源是否可用。如果資源可用，您可以連線到使用受管工作群組搭配 Apache Iceberg 相容 SQL 用戶端的目錄，以查詢資料或建立資料庫資源。您也可以使用 Amazon Redshift 查詢編輯器 v2 連線到目錄。

查詢和資料庫監控包含受管工作群組效能圖，顯示工作群組中所有查詢隨時間經過的平均經過時間。

查詢歷史記錄索引標籤中列出受管工作群組的所有查詢。其詳細資訊包括執行查詢的使用者、產生查詢的用戶端引擎，以及查詢的 ID 和狀態等資訊。「使用者」索引標籤中列出工作群組的所有使用者。效能指標索引標籤會顯示各種指標，例如平均查詢時間、已完成查詢數量，以及使用的儲存容量百分比。

刪除工作群組

您可以使用主控台刪除工作群組。在執行此操作之前，請確定您已備份資料並備妥快照。在許多情況下，泥無法擷取隨工作群組一起刪除的資源。

請完成下列步驟：

1. 選擇 Amazon Redshift Serverless、選擇工作群組組態，然後選擇刪除 Amazon Redshift Serverless 執行個體。

2. 隨即會開啟對話方塊。當您選擇刪除工作群組時，系統會移除所有用量限制、移除所有 VPC 端點，並移除對 VPC 端點的存取權。

   輸入刪除，然後選取刪除來進行確認。

完成這些步驟後，工作群組的狀態會是正在刪除，橫幅則會指出正在刪除工作群組。刪除程序進行期間，Serverless 儀表板下的某些功能會遭到停用。但是，您可以在佈建叢集儀表板上設定已佈建的叢集。

刪除工作群組後，該工作群組就不會與命名空間一起顯示。您可以選擇建立工作群組按鈕來建立新的工作群組。

您可以刪除現有工作群組，並將具有不同組態的新工作群組與相同的命名空間建立關聯。在建立新的工作群組時，請選擇能與命名空間的相關聯資料大小搭配運作的基本容量。

您可以將工作群組與使用客戶自管金鑰 (CMK) 所建立的命名空間建立關聯。如需 的詳細資訊 AWS KMS，請參閱 AWS KMS 概念。

命名空間

在 Amazon Redshift Serverless 中，命名空間會定義資料庫物件的邏輯容器。其可以包含資料表、工作群組和其他資料庫資源。如果您尚未建立工作群組和命名空間，而且正在尋找如何開始使用 Amazon Redshift Serverless 的指示，請參閱第一次設定 Amazon Redshift Serverless。

命名空間屬性

在 Amazon Redshift Serverless 中，命名空間會定義資料庫物件的容器。您可以從導覽清單中選擇命名空間組態、從清單中選擇命名空間，然後編輯其設定。

命名空間的一般資訊包括以下項目：

- 命名空間 — 名稱。
- 命名空間 ID — 唯一識別碼。
- ARN - 用來指定資源的唯一識別符 AWS。其包含區域和服務等屬性。
- 狀態 — 即狀態，例如可用。
- 建立日期 - 命名空間建立的日期 (UTC)。
- 使用的儲存 — 命名空間及其所有物件所使用的儲存空間。
- 管理員使用者名稱 — 管理員帳戶。這一般是用來建立命名空間的帳戶。
- 資料庫名稱 — 命名空間所含資料庫的名稱。
- 資料表總數 — 所有結構描述中的資料表計數。

命名空間的其他設定和屬性位於數個索引標籤上。這些索引標籤包括以下項目：

- 工作群組 — 顯示與命名空間相關聯的工作群組。
- 資料備份 — 在這個面板上，您可以設定和建立快照，以及設定復原點。
- 安全和加密 — 您可以管理 IAM 角色許可，以及檢視或編輯安全和加密設定。這些項目包括您的加密金鑰狀態，以及用來開啟稽核記錄的設定。如需 Amazon Redshift Serverless 稽核記錄的相關資訊，請參閱 Amazon Redshift Serverless 的稽核記錄。
- 資料共用 — 顯示資料共用。透過資料共用，您不必複製或移動資料即可提供資料存取權。如需資料共用的相關資訊，請參閱 Amazon Redshift Serverless 中的資料共用。

搜尋命名空間

在 Amazon Redshift 選單中，您可以從命名空間清單中進行選擇，以檢視或編輯命名空間的屬性。主控台上的資訊包括命名空間名稱、管理員名稱和其他屬性。

命名空間的設定和屬性位於多個索引標籤上。這些索引標籤包括以下項目：

- 工作群組 — 顯示與命名空間相關聯的工作群組。
- 資料備份 — 您可以設定和建立快照，以及設定復原點。

- 安全和加密 — 您可以管理 IAM 角色許可，以及檢視或編輯安全和加密設定。這些項目包括您的加密金鑰狀態和稽核記錄設定。
- 資料共用 — 顯示資料共用。

編輯安全和加密

Amazon Redshift Serverless 會透過 KMS 加密的方式來獲得保護。您可以透過主控台更新加密設定：

1. 從主控台的主選單選擇命名空間組態、選擇要編輯的命名空間，然後在安全和加密索引標籤上選擇編輯。隨即會出現對話方塊。
2. 您可以選取自訂加密設定，然後選擇 AWS 客戶受管金鑰來變更用於加密資源的金鑰。
3. 對於稽核記錄，選擇要匯出的日誌。每種日誌類型會指定不同的中繼資料
4. 若要完成組態更新，請選擇儲存變更。

變更命名空間的 AWS KMS 金鑰

在 Amazon Redshift 中，加密可保護靜態資料。Amazon Redshift Serverless 會自動使用 AWS KMS 金鑰加密來加密 Amazon Redshift Serverless 資源和快照。作為最佳實務，大多數組織會檢閱其儲存的資料類型，並且會規劃按排程輪換加密金鑰。視您的資料安全政策而定，輪換金鑰的頻率可能會有所不同。Amazon Redshift Serverless 支援變更命名空間的 AWS KMS 金鑰，因此您可以遵循組織的安全政策。

當您變更 AWS KMS 金鑰時，資料保持不變。

使用主控台變更 AWS KMS 金鑰

在 Amazon Redshift 中，加密可保護靜態資料。Amazon Redshift Serverless 會自動使用 AWS KMS 金鑰加密功能來加密 Amazon Redshift Serverless 和快照。作為最佳實務，大多數組織會檢閱其儲存的資料類型，並且會規劃按排程輪換加密金鑰。視您的資料安全政策而定，輪換金鑰的頻率可能會有所不同。Amazon Redshift Serverless 支援變更命名空間的 AWS KMS 金鑰，因此您可以遵循組織的安全政策。

當您變更 AWS KMS 金鑰時，資料保持不變。

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單中，選擇命名空間組態。從清單中選擇您的命名空間。
3. 從安全和加密索引標籤中，選擇編輯。

4. 選擇自訂加密設定，然後選擇命名空間的金鑰。您可以選擇建立新的金鑰。

使用 變更 AWS KMS 加密金鑰 AWS CLI

使用 update-namespace 變更命名空間的 AWS KMS 金鑰。下列內容顯示命令的語法：

```
aws redshift-serverless update-namespace
--namespace-name
[--kms-key-id <id-of-kms-key>]
// other parameters omitted here
```

您必須已建立命名空間，否則 CLI 命令會導致錯誤。

變更金鑰所需的時間取決於 Amazon Redshift Serverless 中的資料量。每 8 TB 的儲存資料一般需要 15 分鐘的時間。

限制

您無法從客戶受管 KMS 金鑰變更為 AWS KMS 金鑰。在此案例中，您必須建立新的命名空間。

金鑰變更期間，您無法執行其他動作。

刪除命名空間

如果要刪除具有關聯工作群組的命名空間，則必須先刪除該工作群組。

在 Amazon Redshift Serverless 主控台上，完成下列步驟：

1. 從左側選單中選擇命名空間組態，然後從清單中選擇要刪除的命名空間。
2. 選擇動作，然後選取刪除命名空間。
3. 隨即會開啟對話方塊。您可以透過在完成刪除操作之前手動建立快照來保留資料。

   輸入刪除，然後選取刪除來進行確認。

使用 Amazon Redshift Serverless 監控查詢和工作負載

您可以使用所提供的系統檢視來監控 Amazon Redshift Serverless 查詢和工作負載。

監控檢視是 Amazon Redshift Serverless 中用來監控查詢和工作負載用量的系統檢視。這些檢視位於 pg_catalog 結構描述中。可用的系統檢視旨在為您提供要監控 Amazon Redshift Serverless 所需的

資訊，這比已佈建叢集所需的資訊簡單得多。SYS 系統檢視旨在與 Amazon Redshift Serverless 搭配運作。若要顯示這些檢視所提供的資訊，請執行 SQL SELECT 陳述式。

定義系統檢視是為了支援下列監控目標。

工作負載監控

您可以監控一段時間內的查詢活動，以便執行下列操作：

- 了解工作負載模式，以知道什麼是正常 (基準) 以及商業服務水準協議 (SLA) 中的內容。
- 快速識別偏離正常的情況，這可能是短暫的問題或需要採取進一步行動的事件。

資料載入與卸載的監控

將資料移入和移出 Amazon Redshift Serverless 是一項關鍵功能。您可以使用 COPY 和 UNLOAD 來載入或卸載資料，而且您必須密切監控每一傳輸資料列的位元組和已完成檔案的進度，以追蹤商業 SLA 的遵循情況。這一般會透過頻繁 (即每分鐘) 執行系統資料表查詢來追蹤進度，並在偵測到重大偏差時提出調查/糾正行動的提醒。

失敗和問題診斷

在某些情況下，您必須針對查詢或執行期的失敗採取行動。開發人員依賴系統資料表來自我診斷問題並確定正確的補救措施。

效能調校

您可能需要針對從一開始就不符合 SLA 要求的查詢，或是隨時間降級的查詢進行微調。若要微調，您必須具有執行期的詳細資訊，包括執行計劃、統計資料、持續時間和資源取用。您需要有違規查詢的基準資料，以判斷偏差的原因，並指導您如何改善效能。

使用者物件事件監控

您需要監控使用者物件上的動作和活動，例如重新整理具體化視觀表、清空和分析。這包括系統管理的事件，如具體化視觀表的自動重新整理 您會想要監控使用者啟動的事件何時結束，或是系統啟動的最後一次成功執行。

帳單的用量追蹤

您可以監控一段時間內的用量趨勢，以便執行下列操作：

- 了解預算規劃和業務擴展估算。
- 找出潛在的成本節約機會，例如移除冷資料。

使用 SYS 系統檢視以監控 Amazon Redshift Serverless。如需 SYS 監控檢視的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SYS 監控檢視。

新增查詢監控政策

超級使用者可以為非超級使用者的使用者提供存取權，讓非超級使用者可以針對所有使用者執行查詢監控。首先，請為使用者或角色新增政策，以提供查詢監控存取權。然後，向使用者或角色授予查詢監控許可。

新增查詢監控政策

1. 選擇 https://console.aws.amazon.com/iam/。
2. 在 Access management (存取管理) 下，請選擇 Policies (政策)。
3. 選擇建立政策。
4. 選擇 JSON，並貼上下列政策定義。

   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. 選擇檢閱政策。
6. 對於 Name (名稱)，輸入政策的名稱，例如 `query-monitoring`。
7. 選擇建立政策。

建立政策後，就可以授予適當許可。

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 中的使用者和群組 AWS IAM Identity Center：

  建立權限合集。請按照《AWS IAM Identity Center 使用者指南》中的建立權限合集說明進行操作。
- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。遵循《IAM 使用者指南》的為第三方身分提供者 (聯合) 建立角色中的指示。
- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照《IAM 使用者指南》的為 IAM 使用者建立角色中的指示。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循《IAM 使用者指南》的新增許可到使用者 (主控台) 中的指示。

對使用者授予查詢監控許可

具有 `sys:monitor` 許可的使用者可以檢視所有查詢。此外，具有 `sys:operator` 許可的使用者可以取消查詢、分析查詢歷史記錄，以及執行清空操作。

向使用者授予查詢監控許可

1. 輸入下列命令來提供系統監控存取權，其中 user-name 是您想要為其提供存取權之使用者的名稱。

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

2. (選擇性) 輸入下列命令來提供系統操作人員存取權，其中 user-name 是您想要為其提供存取權之使用者的名稱。

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

對角色授予查詢監控許可

使用者若有具有 `sys:monitor` 許可的角色，便可以檢視所有查詢。此外，使用者若有具有 `sys:operator` 許可的角色，便可以取消查詢、分析查詢歷史記錄，以及執行清空操作。

向角色授予查詢監控許可

1. 輸入下列命令來提供系統監控存取權，其中 role-name 是您想要為其提供存取權之角色的名稱。

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

2. (選擇性) 輸入下列命令來提供系統操作人員存取權，其中 role-name 是您想要為其提供存取權之角色的名稱。

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

設定用量限制，包括設定 RPU 限制

在工作群組的限制索引標籤下，您可以新增一或多個用量限制，以控制您在給定期間內使用的 RPU 上限，或設定資料共用用量限制。

1. 選擇管理用量限制。限制區段會顯示在按期間計算用量面板的底部。
2. 以小時為單位設定用量限制。
3. 首先，選擇頻率，即每日、每週或每月。這會設定用量限制的期間。在這種情況下，選擇每日可提供更詳細的控制能力。
4. 以小時數為單位設定用量限制。
5. 設定動作。這些項目如下：

   - 記錄至系統資料表 - 將記錄新增至系統檢視 SYS_QUERY_HISTORY。您可以查詢此檢視中的 `usage_limit` 欄，以判斷查詢是否超過限制。
   - 提醒 — 使用 Amazon SNS 設定通知訂閱，並在超出限制時傳送通知。您可以選擇現有的 Amazon SNS 主題，或建立新的主題。
   - 關閉使用者查詢 — 停用查詢以停止 Amazon Redshift Serverless 的使用。其也會傳送通知。

   前兩個動作會提供參考資訊，最後一個動作則會關閉查詢的處理。
6. 您可以選擇性地設定跨區域資料共用用量限制，以限制取用者可查詢多少從生產者區域傳輸至取用者區域的資料。若要這麼做，請選擇新增限制，然後遵循步驟來進行。
7. 在頁面底部選擇儲存變更以儲存您限制。
8. 視需要設定最多 3 個限制。

如需更多有關 RPU 和帳單的概念性資訊，請參閱 Amazon Redshift Serverless 的帳單。

設定查詢限制

在工作群組的限制索引標籤下，您可以新增限制以監控效能和限制。如需查詢監控限制的相關資訊，請參閱 WLM 查詢監控規則。

1. 選擇管理查詢限制。在管理查詢限制對話方塊上選擇新增新的限制。
2. 選擇您要設定的限制類型，然後輸入其對應限制的值。
3. 選擇儲存變更以儲存限制。

當您變更查詢限制和組態參數時，資料庫將會重新啟動。

設定查詢佇列

Amazon Redshift Serverless 支援佇列型查詢資源管理。您可以建立專用的查詢佇列，採用自訂的監控規則來處理不同的工作負載。此功能可讓您精細控制資源用量。

查詢監控規則 (QMR) 僅適用於 Redshift Serverless 工作群組層級，影響此工作群組中執行的所有查詢。佇列式的方法可讓您建立具有不同監控規則的佇列。您可以將這些佇列指派給特定使用者角色和查詢群組。每個佇列都會獨立運作，而規則只會影響該佇列內的查詢。

佇列可讓您設定指標型述詞和自動化回應。例如，您可以設定規則，以自動中止超過時間限制或耗用太多資源的查詢。

考量事項

使用無伺服器佇列時，請考慮下列事項：

- Redshift Serverless 佇列不支援 Amazon Redshift 佈建叢集中使用的下列工作負載管理 (WLM) 組態金
  鑰：`max_execution_time`、`short_query_queue`、`auto_wlm`、`concurrency_scaling`、`priorit`

  此外，Serverless 不支援 change_query_priority 動作。
- Amazon Redshift Serverless 不支援躍點動作（在佇列之間移動查詢）。
- 僅 Amazon Redshift 佈建叢集支援佇列優先順序。
- Amazon Redshift Serverless 會自動管理擴展和資源配置，以獲得最佳效能，因此您不需要手動設定佇列優先順序。

設定查詢佇列

您可以使用 AWS 管理主控台 AWS CLI或 Redshift Serverless API，在無伺服器工作群組的限制索引標籤下建立佇列。

Console

請依照下列步驟，為您的無伺服器工作群組建立佇列。

1. 導覽至 Redshift Serverless 工作群組。
2. 選取限制索引標籤。
3. 在查詢佇列下，選擇啟用佇列。

> **Important**
> 啟用查詢佇列是永久變更。一旦啟用，您就無法還原至無佇列監控。

4. 使用下列參數設定佇列：

   佇列層級參數

   - `name` - 佇列識別符（必要、唯一、非空白）
   - `user_role` - 使用者角色陣列（選用）
   - `query_group` - 查詢群組陣列（選用）
   - `query_group_wild_card` - 0 或 1 以啟用萬用字元比對（選用）
   - `user_group_wild_card` - 0 或 1 以啟用萬用字元比對（選用）
   - `rules` - 監控規則陣列（選用）

   規則層級參數

   - `rule_name` - 唯一識別符，最多 32 個字元（必要）
   - `predicate` - 條件陣列，1-3 個述詞（必要）
   - `action` -「中止」或「日誌」（必要）

   述詞層級參數

   - `metric_name` - 要監控的指標（必要）

- operator - "="、"<" 或 ">"（必要）
- value - 數值閾值（必要）

限制

- 最多 8 個佇列
- 所有佇列最多 25 個規則
- 每個規則最多 3 個述詞
- 規則名稱全域必須是唯一的

範例組態

三個佇列範例：一個用於短逾時的儀表板查詢，一個用於長逾時的 ETL 查詢，另一個用於管理員佇列：

```
[
  {
    "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"]
  }
]
```

在此範例中：

- 如果儀表板查詢執行超過 60 秒，則會中止查詢
- 如果 ETL 查詢執行超過一小時，則會記錄這些查詢
- 管理員佇列沒有任何資源限制

CLI

您可以使用 CreateWorkgroup 或 UpdateWorkgroup APIs搭配 `wlm_json_configuration` 組態參數來管理佇列，以 JSON 格式指定佇列。

```
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\"]}]"}]'
```

最佳實務

當您使用無伺服器佇列時，請記住下列最佳實務。

- 對於具有不同限制需求的工作負載（例如 ETL、報告或臨機操作分析），請使用個別佇列。
- 從簡單的閾值開始，並根據查詢行為和用量模式進行調整。您可以使用系統資料表和檢視中記錄的資料表和檢視來監控查詢用量模式，以取得查詢監控規則。

使用儀表板檢查 Amazon Redshift Serverless 摘要資料

Amazon Redshift Serverless 儀表板包含一組面板，這些面板會顯示一目了然的指標以及有關工作群組和命名空間的資訊。這些面板包括下列項目：

- 資源摘要 — 顯示有關 Amazon Redshift Serverless 的高階資訊，例如已使用的儲存和其他指標。
- 查詢摘要 — 顯示有關查詢的資訊，包括已完成的查詢和執行中的查詢。選擇檢視詳細資訊以移至具有其他篩選條件的畫面。
- 已使用的 RPU 容量 — 顯示給定期間內已使用的整體容量，例如前 10 個小時。
- 資料共用 - 顯示資料共用的計數，用於在 AWS 帳戶之間共用 darta。這個指標會顯示哪些資料共用需要授權以及其他資訊。
- 總運算用量 - 顯示所選時間範圍內所選工作群組的總使用 RPU 時數 (最多過去 7 天)。

您可以從儀表板快速深入了解這些可用指標，以查看有關 Amazon Redshift Serverless 的詳細資訊，或是檢閱查詢或追蹤工作項目。

Amazon Redshift Serverless 的稽核記錄

您可以設定 Amazon Redshift Serverless，以將連線、使用者和使用者活動日誌資料匯出至 Amazon CloudWatch Logs 中的日誌群組。透過 Amazon CloudWatch Logs，您可以執行日誌資料的即時分析，並使用 CloudWatch 來建立警示和檢視指標。您可以使用 CloudWatch Logs 將日誌記錄儲存在耐用的儲存中。

您可以建立 CloudWatch 警示，以使用 Amazon Redshift 主控台追蹤您的指標。如需建立警示的相關資訊，請參閱管理警示。

若要將產生的日誌資料匯出至 Amazon CloudWatch Logs，就必須在主控台上的 Amazon Redshift Serverless 組態設定中選取要匯出的相應日誌。您可以選擇安全和加密底下的命名空間組態設定來執行此操作。

在 CloudWatch 中記錄事件

選取要匯出的 Redshift 日誌後，您就可以在 Amazon CloudWatch Logs 中監控事件。系統會自動為 Amazon Redshift Serverless 建立新的日誌群組，其中 `log_type` 代表日誌類型。

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

當您建立第一個工作群組和命名空間時，預設值是命名空間名稱。日誌群組名稱會根據您命名空間的名稱而有所不同。

例如，如果您匯出連線日誌，則日誌資料會儲存在下列日誌群組中。

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

日誌事件會使用無伺服器日誌串流匯出至日誌群組。該行為取決於下列哪個條件為真：

- 具指定名稱的日誌群組存在。Redshift 會使用現有的日誌群組匯出日誌資料。若要以預先定義的日誌保留期間、指標篩選條件和客戶存取權來建立日誌群組，您可使用自動化組態 (例如 AWS CloudFormation 所提供的組態)。
- 具指定名稱的日誌群組不存在。在執行個體的日誌中偵測到相符的日誌項目時，Amazon Redshift Serverless 會自動在 Amazon CloudWatch Logs 中建立新的日誌群組。日誌群組會使用永不過期的預設日誌保留期間。若要變更日誌保留期間，請使用 Amazon CloudWatch Logs 主控台 AWS CLI、

或 Amazon CloudWatch Logs API。如需在 CloudWatch Logs 中變更日誌保留期間的相關資訊，請參閱使用日誌群組和日誌串流中的變更日誌資料保留。

若要搜尋日誌事件中的資訊，請使用 Amazon CloudWatch Logs 主控台 AWS CLI、 或 Amazon CloudWatch Logs API。如需搜尋和篩選日誌資料的相關資訊，請參閱搜尋和篩選日誌資料。

CloudWatch 指標

Amazon Redshift Serverless 指標分為運算指標以及資料和儲存指標，兩者分別屬於工作群組和命名空間維度集。如需工作群組和命名空間的詳細資訊，請參閱工作群組和命名空間。

CloudWatch 運算指標如下：

指標名稱	單位	說明	維度集
QueriesCompletedPerSecond	查詢數量	每秒完成的查詢次數。	{Database, LatencyRange, Workgroup}, {LatencyRange, Workgroup}
QueryDuration	微秒	完成查詢的平均時間。	{Database, LatencyRange, Workgroup}, {LatencyRange, Workgroup}
QueriesRunning	查詢數量	某個時間點的執行中查詢數目。	{Database , QueryType , Workgroup }, {QueryType, Workgroup}
QueriesQueued	查詢數量	佇列中某個時間點的查詢數目。	{Database , QueryType , Workgroup }, {QueryType, Workgroup}

指標名稱	單位	說明	維度集
DatabaseConnections	連線數量	某個時間點的資料庫連線數目。	{Database, Workgroup}, {Workgroup}
QueryRuntimeBreakdown	毫秒	依查詢階段的查詢所花費總執行時間。	{Database, Stage, Workgroup}, {Stage, Workgroup}
ComputeCapacity	RPU	過去 30 分鐘內配置的運算單元平均數量，四捨五入至最接近的整數。	{Workgroup}
ComputeSeconds	RPU-秒	過去 30 分鐘內使用的累計運算單元秒數。	{Workgroup}
QueriesSucceeded	查詢數量	過去 5 分鐘內成功的查詢數目。	{Database , QueryType , Workgroup }, {QueryType, Workgroup}
QueriesFailed	查詢數量	過去 5 分鐘內失敗的查詢數目。	{Database , QueryType , Workgroup }, {QueryType, Workgroup}

指標名稱	單位	說明	維度集
UsageLimitAvailable	RPU-小時或 TB	根據 UsageType，UsageLimitAvailable 會傳回下列項目： • 如果 UsageType 是 SERVERLESS_COMPUTE，則 UsageLimitAvailable 會傳回工作群組在給定限制內可以查詢的剩餘 RPU-小時數。 • 如果 UsageType 是 CROSS_REGION_DATASHARING，則 UsageLimitAvailable 會傳回客戶可以在給定限制內掃描的剩餘 TB 數。	{UsageLimitId, UsageType, Workgroup}

指標名稱	單位	說明	維度集
UsageLimitConsumed	RPU-小時或 TB	根據 UsageType，UsageLimitConsumed 會傳回下列項目： • 如果 UsageType 是 SERVERLESS_COMPUTE，則 UsageLimitConsumed 會傳回工作群組在給定限制內已查詢的 RPU 小時數。 • 如果 UsageType 是 CROSS_REGION_DATASHARING，則 UsageLimitConsumed 會傳回客戶在給定限制內已用來掃描的 TB 數。	{UsageLimitId, UsageType, Workgroup}

指標名稱	單位	說明	維度集
ExtraComputeForAutomaticOptimizationChargedSeconds	RPU-秒	過去 30 分鐘內針對自動最佳化操作收取的運算單位秒數。	{Workgroup}

CloudWatch 資料和儲存指標如下：

指標名稱	單位	說明	維度集
TotalTableCount	資料表數目	特定時間點的資料表總數。此總計包含永久資料表、暫時資料表、資料共用資料表、外部資料表和具體化檢視。此計數不包含檢視和系統資料表。	{Database, Namespace}
DataStorage	MB	在磁碟或儲存空間中用於 Redshift 資料的 MB 數。	{Namespace}

ServerlessUniqueSnapshotStorage 指標與命名空間和工作群組無關。CloudWatch 的 ServerlessUniqueSnapshotStorage 指標如下：

指標名稱	單位	說明	維度集
ServerlessUniqueSnapshotStorage	MB	在所有作用中手動快照中存放的	{}

指標名稱	單位	說明	維度集
		唯一資料區塊數。	

維度集是套用至指標的分組維度。您可以使用這些維度群組來指定擷取統計資料的方式。

下表詳細說明特定指標的維度和維度值：

維度	描述和值
DatabaseName	資料庫的名稱。自訂值。
Latency	可能的值如下： • 短 — 少於 10 秒 • 中 — 介於 10 秒到 10 分鐘 • 長 — 超過 10 分鐘
QueryType	可能值為 INSERT、DELETE、UPDATE、UNLOAD、LOAD、SELEC 及 OTHER。
stage	查詢的執行階段。可能的值如下： • QueryPlanning：剖析與最佳化 SQL 陳述式花費的時間。 • QueryWaiting：花費在 WLM 佇列中等待的時間。 • QueryExecutingRead：執行讀取查詢所花費的時間。 • QueryExecutingInsert：執行插入查詢所花費的時間。 • QueryExecutingDelete：執行刪除查詢所花費的時間。 • QueryExecutingUpdate：執行更新查詢所花費的時間。

維度	描述和值
	• QueryExecutingCtas：執行建立表格為查詢所花費的時間。 • QueryExecutingUnload：執行卸載查詢所花費的時間。 • QueryExecutingCopy：執行複製查詢所花費的時間。 • QueryCommit：花費在認可的時間。
Namespace	命名空間的名稱。自訂值。
Workgroup	工作群組的名稱。自訂值。
UsageLimitId	用量限制的識別碼。
UsageType	Amazon Redshift Serverless 功能會受限。可能的值如下： • SERVERLESS_COMPUTE • CROSS_REGION_DATASHARING

快照和復原點

Amazon Redshift Serverless 中的備份是命名空間中物件和資料的時間點表示。備份類型有兩種：手動建立的快照和 Amazon Redshift Serverless 自動為您建立的復原點。

Amazon Redshift Serverless 會每 30 分鐘或在每個節點的資料變更每達 5 GB 後自動建立復原點，以先發生者為準。對於較大的資料集 (超過 5 GB × 節點數)，復原點之間的最小間隔為 15 分鐘。所有復原點都會保留 24 小時。

> **Note**
>
> 您無法建立自己的快照排程來控制建立復原點的時間。

Amazon Redshift Serverless 會在 Redshift 受管儲存 (RMS) 中建立快照。如需詳細資訊，請參閱Amazon Redshift Serverless 的運算容量。

> **Note**
>
> RG 或 RA3 佈建叢集和 Amazon Redshift Serverless 工作群組不支援無備份資料表。在 RG 或 RA3 叢集或無伺服器工作群組中標記為無備份的資料表會被視為永久資料表，在拍攝快照時一律會備份，並在從快照還原時一律還原。為了避免無備份資料表的快照成本，請在拍攝快照之前截斷這些資料表。

如果您發現要擷取快照或復原點中的資料，您可以將快照還原至無伺服器命名空間或已佈建的叢集。您可以在三種情況下還原快照：

- 將無伺服器快照還原至無伺服器命名空間。
- 將無伺服器快照還原至已佈建的叢集。
- 將已佈建叢集的快照還原至無伺服器命名空間。

當您將無伺服器快照還原至佈建叢集時，您必須選擇要使用的節點類型，例如 RG 或 RA3，以及節點數量，可讓您在叢集或節點層級控制設定。

若要將已佈建叢集的快照還原至無伺服器命名空間，請從 Redshift 佈建的主控台開始、選擇要還原的快照，然後選擇從快照還原、還原至無伺服器命名空間。當您將已佈建叢集的快照還原到無伺服器命名空間時，Amazon Redshift 會將具有交錯索引鍵的資料表轉換為複合索引鍵。如需排序索引鍵的相關資訊，請參閱使用排序索引鍵。

如果您要新增其他內容，可以使用會提供中繼資料和資訊給快照和復原點的鍵值組來標記快照和復原點。如需標記資源的相關資訊，請參閱標記資源概觀。

最後，您也可以與其他 AWS 帳戶共用快照，讓他們存取快照中的資料並執行查詢。

AWS Backup 整合

您也可以使用全受管服務建立和還原快照 AWS Backup，該服務可協助您集中和自動化跨 AWS 服務、雲端和內部部署的資料保護。如需詳細資訊，請參閱AWS Backup 與 Amazon Redshift 整合。如需詳細資訊 AWS Backup，請參閱什麼是 AWS Backup？ 《AWS Backup 開發人員指南》中的 。

建立快照

> **Note**
>
> RG 或 RA3 佈建叢集和 Amazon Redshift Serverless 工作群組不支援無備份資料表。在 RG 或 RA3 叢集或無伺服器工作群組中標記為無備份的資料表會被視為永久資料表，在拍攝快照時一律會備份，並在從快照還原時一律還原。為了避免無備份資料表的快照成本，請在拍攝快照之前截斷這些資料表。

若要建立快照，請執行下列程序中的步驟。

建立快照

1. 在 Amazon Redshift Serverless 主控台上，選擇資料備份。
2. 選擇建立快照。
3. 選擇要為其建立快照的命名空間。
4. 輸入快照識別碼。
5. (選擇性) 選擇保留期間。如果您選擇自訂值，請選擇天數。您選擇的天數必須在 1 至 3653 天 (含) 之間。預設值為無限期保留。
6. 選擇建立。

從命名空間組態建立快照

1. 在 Amazon Redshift Serverless 主控台上，選擇命名空間組態。
2. 選擇要為其建立快照的命名空間。您只能為與工作群組相關聯且狀態為「可用」的命名空間建立快照。
3. 選擇資料備份索引標籤。
4. 選擇建立快照。
5. 輸入快照識別碼。
6. (選擇性) 選擇保留期間。如果您選擇自訂值，請選擇天數。您選擇的天數必須在 1 至 3653 天 (含) 之間。
7. 選擇建立。

建立最終快照

若要在刪除命名空間之前，先建立命名空間內所有資料的最終快照，請執行下列程序中的步驟。

建立最終快照

1. 在 Amazon Redshift Serverless 主控台上，選擇命名空間組態。
2. 選擇要刪除的命名空間。
3. 選擇 動作、刪除。
4. 選擇建立最終快照。
5. 輸入快照的名稱。
6. 輸入刪除。
7. 選擇 刪除。

共用快照或移除快照許可

若要與其他 AWS 帳戶共用快照，或移除帳戶對快照的存取權，請執行下列程序。

共用或移除快照的存取權

1. 在 Amazon Redshift Serverless 主控台上，選擇資料備份。
2. 選擇要共用的快照。
3. 選擇動作、管理存取權。
4. 若要與其他帳戶共用快照，請輸入 AWS 帳戶 ID。若要移除帳戶的存取權，請選擇移除。
5. 選擇儲存變更。

排程快照

若要精確控制拍攝快照的時間，您可以為特定的命名空間集建立快照排程。建立排程快照時，您可以建立一次性事件或使用 Unix cron 運算式來建立週期性排程。Cron 表達式支援三個欄位，並以空格隔開。

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

欄位	Values (數值)	萬用字元
分鐘	0–59	, - * /
小時	0–23	, - * /
月中的日	1–31	, - * ? / L W
月	1-12 或 JAN-DEC	, - * /
週中的日	1-7 或 SUN-SAT	, - * ? L #
年	1970-2199	, - * /

萬用字元

- , (逗號) 萬用字元包含額外的值。在 `Day-of-week` 欄位，`MON,WED,FRI` 包括週一、週二和週三。每個欄位的總數值限制為 24。
- - (破折號) 萬用字元用於指定範圍。在 `Hour` 欄位中，1–15 會包含指定日的 1 至 15 時。
- * (星號) 包含欄位中所有的值。在 `Hours` 欄位，* 包含每個小時。
- / (斜線) 萬用字元用於指定增量。在 `Hours` 欄位，您可以輸入 **`1/10`** 指定每第 10 小時，從一天的第一小時開始 (例如 01:00、11:00 和 21:00)。
- ? (問號) 萬用字元用於表示不限定任何一個。在 `Day-of-month` 欄位，您可以輸入 7，如果您不在意這個月的 7 號是星期幾，就可以在 Day-of-week (週中的日) 欄位中輸入 ?。
- L 萬用字元在 `Day-of-month` 或 `Day-of-week` 欄位可指定月份或週的最後一天。
- W 萬用字元在 `Day-of-month` 欄位可指定任務日。在 `Day-of-month` 欄位，`3W` 指定的是月份中最接近第三個任務日的日子。
- # 萬用字元在 Day-of-week (週中的日) 欄位可指定某個月中某週特定日子的特定執行個體。例如，3#2 代表則該月的第二個星期二：3 是指星期二，因為它是每週的第三天，2 指的是一個月內該類型的第二天。

> **Note**
>
> 如果您使用 '#' 字元，則只能在星期幾欄位中定義一個表達式。例如："3#1,6#3" 是無效的，因為它被轉譯為兩個表達式。

限制

- 您無法在同一個 cron 表達式中指定 Day-of-month 和 Day-of-week 欄位。如果您在其中一個欄位指定了數值，就必須在另一個欄位中使用 ? (問號)。
- 快照排程不支援下列頻率：
  - 快照排程頻率超過每小時 1 次。
  - 快照排程頻率低於每天 (24 小時) 1 次。

  如果您有重疊的排程，導致 1 個小時內有多個排程快照，將會產生驗證錯誤。

下表包含一些 cron 字串範例。

分鐘	小時	週中的日	意義
0	14-20/1	TUE	週二下午 2 點至 8 點，每小時一次。
0	21	MON-FRI	週一至週五每天晚上 9 點。
30	0/6	SAT-SUN	週六和週日每 6 小時增量，從當日午夜後 30 分鐘開始 (00:30)。結果是在每天 [00:30、06:30、12:30 和 18:30] 進行快照。
30	12/4	*	每天 12:30 開始每 4 小時增量。這解析為 [12:30、16:30、20:30]。

下列範例說明如何建立可從每天 15:15 開始以 2 小時增量執行的排程。

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

您可以使用 Amazon Redshift Serverless 主控台、API 或 AWS CLI 來建立快照排程。

排程快照

1. 在 Amazon Redshift Serverless 主控台上，選擇資料備份。
2. 選擇快照排程。
3. 選擇建立排程。
4. 輸入快照排程的名稱。
5. 選取要為其建立快照的命名空間。
6. 輸入排程的 cron 表達式，或使用排程建置器來建立排程。
7. (選擇性) 選擇保留期間。如果您選擇自訂值，請指定天數。
8. 選擇建立排程。

更新快照保留期間

若要更新快照保留期間，請執行下列程序。

更新快照保留期間

1. 在 Amazon Redshift Serverless 主控台上，選擇資料備份。
2. 選擇要更新的快照。
3. 選擇動作、設定手動快照設定。
4. 選擇保留期間。如果您選擇自訂值，請選擇天數。
5. 選擇儲存變更。

刪除快照

若要刪除快照，請執行下列程序。

刪除快照

> **Note**
>
> 您無法刪除已和其他帳戶共用的快照。您必須先移除該帳戶對快照的存取權，然後才能刪除快照。

1. 在 Amazon Redshift Serverless 主控台上，選擇資料備份。
2. 選擇要刪除的快照。
3. 選擇 動作、刪除。
4. 選擇 刪除。

還原快照

> **Note**
>
> RG 或 RA3 佈建叢集和 Amazon Redshift Serverless 工作群組不支援無備份資料表。在 RG 或 RA3 叢集或無伺服器工作群組中標記為無備份的資料表會被視為永久資料表，在拍攝快照時一律會備份，並在從快照還原時一律還原。為了避免無備份資料表的快照成本，請在拍攝快照之前截斷這些資料表。

將快照還原至無伺服器命名空間後，會將資料庫替換為快照中的資料庫。

將快照還原至無伺服器命名空間會分兩個階段來完成。第一個階段會在幾分鐘內完成、將資料還原至您的命名空間，並使其可用於查詢。還原的第二個階段是微調資料庫，這可能會造成輕微的效能問題。第二個階段可以持續幾小時到幾天，在某些情況下，甚至會持續幾週。時間長短取決於資料大小，但隨著資料庫進行微調，效能會逐步改善。這個階段結束時，您的無伺服器命名空間會徹底完成微調，而且您可以提交查詢而不會出現效能問題。

將快照還原至無伺服器命名空間

1. 在 Amazon Redshift Serverless 主控台上，選擇資料備份。
2. 選擇要還原的快照。您一次只能還原一個快照。
3. 選擇動作、還原至無伺服器命名空間。
4. 選擇要作為還原目的地的可用命名空間。您只能還原至狀態為「可用」的命名空間。
5. 選擇 Restore (還原)。

將快照還原至已佈建的叢集

1. 在 Amazon Redshift Serverless 主控台上，選擇資料備份。
2. 選擇要還原的快照。
3. 選擇動作、還原至佈建叢集。

4. 輸入叢集識別碼。
5. 選擇節點類型。節點數取決於節點類型。
6. 遵循主控台頁面上的頁面說明以輸入叢集組態的屬性。如需詳細資訊，請參閱建立叢集。

如需已佈建叢集快照的詳細資訊，請參閱 Amazon Redshift 快照和備份。

轉換復原點

系統大約每 30 分鐘會在 Amazon Redshift Serverless 中建立一個復原點，並儲存 24 小時。若要將復原點轉換為快照，請執行下列程序中的步驟。

將復原點轉換為快照

1. 在 Amazon Redshift Serverless 主控台上，選擇資料備份。
2. 在復原點底下，選擇您要轉換為快照之復原點的建立時間。
3. 選擇從復原點建立快照。
4. 輸入快照識別碼。
5. 選擇建立。

還原復原點

系統大約每 30 分鐘會在 Amazon Redshift Serverless 中建立一個復原點，並儲存 24 小時。若要將復原點還原至無伺服器命名空間，請執行下列程序中的步驟

將復原點還原至無伺服器命名空間

1. 在 Amazon Redshift Serverless 主控台上，選擇資料備份。
2. 在復原點底下，選擇您要還原之復原點的建立時間。
3. 選擇 Restore (還原)。您只能還原至狀態為「可用」的命名空間。
4. 在文字輸入欄位中輸入還原，然後選擇還原。

將備份複製到另一個備份 AWS 區域

您可以設定 Amazon Redshift Serverless 自動將快照和復原點複製到另一個快照和復原點 AWS 區域。當您在來源 AWS 區域中建立快照時，它會複製到目的地區域。您可以設定命名空間，使其 AWS 區域

一次只能將快照和復原點複製到一個目的地。如需 Amazon Redshift Serverless 可用 AWS 區域 位置的清單，請參閱《》中針對 Redshift Serverless API 列出的端點Amazon Web Services 一般參考。

設定複製備份時，您也可以指定 Amazon Redshift Serverless 應保留複製快照的時間長度的保留期。您無法變更復原點的保留期間 (必須為 1 天)。目標區域中快照的保留期與來源區域中快照的保留期是不同的。依預設，保留期會無限期保留快照。如果您選擇自訂值，請選擇天數。您選擇的天數必須在 1 至 3653 天 (含) 之間。

若要變更要複製快照的目的地區域，請先停用複製備份，然後在重新啟用複製時指定新的目的地區域。

將快照或復原點複製到目的地區域後，您可以使用該快照或復原點將資料還原到該區域。

根據預設，您的資料會使用 為您 AWS 管理的金鑰進行加密。若要使用不同的金鑰，請選擇您要在來源中設定備份複製時使用的金鑰 AWS 區域，Amazon Redshift Serverless 會自動建立授予，以便在目的地啟用快照加密 AWS 區域。

若要將備份複寫到另一個區域，請確定您有下列 IAM 許可：

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

如果您使用自己的 KMS 金鑰加密備份，您還需要下列權限：

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

設定將快照或復原點複製到另一個快照或復原點 AWS 區域

1. 在 Amazon Redshift Serverless 主控台上，選擇要為其設定複製快照或復原點的命名空間。
2. 選擇動作，設定跨區域備份。
3. 選擇要 AWS 區域 複製快照的目的地。
4. (選用) 選擇保留快照的時間長度。如果您選擇自訂值，請選擇天數，您選擇的數量必須在 1-3653 天之間 (包括在內)。預設值為無限期保留。
5. （選用）選擇不同的 AWS KMS 金鑰，用於加密目的地區域中的加密。
6. 選擇 Save configuration (儲存組態)。

還原資料表

您也可以從快照或復原點還原特定資料表。這樣做時，您可以指定來源快照或復原點、資料庫、結構描述、資料表、目標資料庫、結構描述和新資料表名稱。這個新資料表的名稱不能與現有資料表相同。若您想要透過還原資料表來取代現有資料表，則必須先重新命名或捨棄現有資料表，然後再還原資料表。

> **Note**
>
> RG 或 RA3 佈建叢集和 Amazon Redshift Serverless 工作群組不支援無備份資料表。在 RG 或 RA3 叢集或無伺服器工作群組中標記為無備份的資料表會被視為永久資料表，在拍攝快照時一律會備份，並在從快照還原時一律還原。不過，不支援選擇性還原無備份資料表。

已使用資料表的欄位定義、資料表屬性和欄位屬性 (外部索引鍵除外) 建立目標資料表。為避免因相依性而導致的衝突，目標資料表不會從來源資料表繼承外部索引鍵。任何相依性，例如檢視或來源資料表上授予的許可，都不會套用至目標資料表。

若來源資料表存在擁有者，則該使用者是已還原資料表的擁有者，前提是該使用者有足夠的許可以成為指定資料庫和結構描述中關係的擁有者。否則，還原的資料表將由啟動叢集時建立的管理員使用者所擁有。

還原的資料表會恢復到備份時所處的狀態。這包括由 Amazon Redshift 符合可序列化的隔離所定義的交易可見度規則，這表示在備份後開始傳送的交易中立刻可看見該資料。

您可以使用 Amazon Redshift Serverless 主控台從快照還原資料表。

以下為從資料備份還原資料表的限制：

- 您一次只能還原一個資料表。
- 任何相依性，例如檢視或來源資料表上授予的許可，都不會套用至目標資料表。
- 如果為要還原的資料表開啟了資料列層級安全，Amazon Redshift Serverless 會在開啟資料列層級安全的情況下還原資料表。

使用 Amazon Redshift Serverless 主控台還原資料表

1. 在 Amazon Redshift Serverless 主控台上，選擇資料備份。
2. 選擇具有要還原之資料表的快照或復原點。
3. 選擇動作、從快照還原資料表或從復原點還原資料表。

4. 輸入來源快照或復原點和目標資料表的相關資訊，然後選擇還原資料表。

Amazon Redshift Serverless 中的資料共用

透過資料共用，您可以即時存取資料，以在資料有所更新時，讓使用者可以在 Amazon Redshift Serverless 中看到最新且一致的資訊。

您可以在 AWS 帳戶內部或之間的不同 Amazon Redshift Serverless 執行個體之間共用資料以供讀取。

您可以透過使用 SQL 介面或 Amazon Redshift 主控台開始使用資料共用功能。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 Amazon Redshift 中的資料共用。

透過資料共用，Amazon Redshift Serverless 命名空間和佈建叢集可以彼此共用即時資料，無論是 AWS 帳戶 跨 AWS 帳戶或跨 的 AWS 區域。如需詳細資訊，請參閱可使用資料共用的區域。

若要開始在 內共用資料 AWS 帳戶，請開啟 AWS 管理主控台，然後選擇 Amazon Redshift 主控台。選擇命名空間組態，然後選擇資料共用。

若要開始查詢資料共用中的資料，請在具有相關聯工作群組的命名空間中建立資料庫。從指定的資料共用中，選擇具有相關聯工作群組的命名空間，然後建立用來查詢資料的資料庫。

考量事項

在 Amazon Redshift Serverless 中使用資料共用時，請考量以下事項：

- Amazon Redshift 支援在佈建的 RG 和 RA3 叢集以及無伺服器端點工作群組上，以生產者或消費者身分進行資料共用。
- Amazon Redshift Serverless 預設會進行加密。

如需資料共用限制 (包括支援的資料庫物件、加密要求和排序索引鍵要求) 的清單，請參閱《Amazon Redshift 資料庫開發人員指南》中的 Amazon Redshift 中的資料共用考量事項。

授予檢視資料共用的存取權

超級使用者可以為非超級使用者的使用者提供存取權，讓非超級使用者可以檢視所有使用者建立的資料共用。

若要為使用者授予資料共用存取權，請使用以下命令為使用者提供資料共用存取權，其中 datashare_name 是資料共用的名稱，user-name 則是您要為其提供存取權之使用者的名稱。

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

若要為使用者群組授予資料共用存取權，請先建立具有使用者的使用者群組。如需如何建立使用者群組的資訊，請參閱 CREATE GROUP。然後，使用以下命令向使用者授予資料共用存取權，其中 datashare_name 是資料共用的名稱，user-group 則是您要向其授予存取權之使用者群組的名稱。

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

如需如何使用 GRANT 陳述式的資訊，請參閱 GRANT。

將命名空間註冊到 AWS Glue Data Catalog

您可以將整個命名空間註冊到 ， AWS Glue Data Catalog 並建立由 管理的目錄 AWS Glue。您可以使用支援 Apache Iceberg REST API 的任何 SQL 引擎來存取這些目錄。如需從 Amazon Redshift 建立 Apache Iceberg 相容目錄的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 Amazon Redshift 的 Apache Iceberg 相容性。

向 註冊無伺服器命名空間 AWS Glue Data Catalog

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ ：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單上，選擇 Redshift Serverless。「無伺服器」儀表板隨即出現。命名空間/工作群組區段中會列出目前 AWS 區域中您帳戶的命名空間和工作群組。如果您沒有任何命名空間，請選擇建立工作群組來建立工作群組及其對應的命名空間。
3. 選擇您要註冊的命名空間名稱。
4. 在動作中，選擇註冊。 AWS Glue Data Catalog在 AWS Glue Data Catalog註冊快顯方塊隨即出現。
5. 在目的地 AWS 帳戶 ID 下，輸入您要註冊命名空間的帳戶 ID。 這個帳戶 ID 將保存 AWS Glue Data Catalog中的目錄。
6. 在將命名空間註冊為下輸入名稱。這將是命名空間在 Data Catalog 中的名稱。
7. 選擇註冊。您將進入 AWS Lake Formation 主控台。
8. 依照 AWS Lake Formation中的目錄建立程序進行。如需建立目錄的相關資訊，請參閱《 AWS Lake Formation 開發人員指南》中的將 Amazon Redshift 資料帶入 AWS Glue Data Catalog。

在 Amazon Redshift Serverless 中標記資源

在 中 AWS，標籤是使用者定義的標籤，由索引鍵/值對組成。Amazon Redshift Serverless 支援標記功能，以讓您一目了然地查看資源的中繼資料。

資源不需要標籤，但標籤可以協助提供內容。您可能會想要使用具有資源相關資訊的中繼資料來標記資源。例如，假設您想要追蹤哪些資源屬於測試環境和生產環境。您可以建立名為 environment 的索引鍵，並提供值 (test 或 production) 來識別每個環境中使用的資源。如果您在其他 AWS 服務中使用標記，或您的業務有標準類別，我們建議您建立相同的鍵/值對，以保持一致性。

如果您刪除資源，任何關聯的標籤也會遭到刪除。您可以使用 AWS CLI 和 Amazon Redshift Serverless 主控台來標記無伺服器資源。可用的 API 操作有 `TagResource`、`UntagResource` 和 `ListTagsForResource`。

每個資源皆有一個標籤集，此為指派給該資源之一或多個標籤的集合。每個資源每個標籤集最多可擁有 50 個標籤。在您建立資源時與資源建立後，您可以新增標籤。您可以將標籤新增至以下無伺服器資源類型：

- 工作群組
- 命名空間
- 快照
- 復原點

標籤均擁有以下要求：

- 索引鍵字首不能是 `aws:`。
- 索引鍵在標籤集內必須是唯一的。
- 索引鍵必須介於 1 到 128 個允許的字元之間。
- 值必須介於 0 到 256 個允許的字元之間。
- 值在每個標籤集中不需要是唯一的。
- 索引鍵和值的允許字元為 Unicode 字母、數字、空格和下列任何符號：_ . : / = + - @。
- 索引鍵和值會區分大小寫。

管理 Amazon Redshift Serverless 資源的標籤

1. 在 Amazon Redshift Serverless 主控台上，選擇管理標籤。

2. 輸入要搜尋的資源類型，然後選擇搜尋資源。選擇您要管理標籤的資源，然後選擇管理標籤。
3. 指定您要新增至資源的索引鍵和選用值。修改標籤時可以變更標籤的值，但無法變更索引鍵。
4. 在完成標籤的新增、移除或修改後，請選擇儲存變更，然後選擇套用以儲存變更。

Amazon Redshift 佈建叢集

Amazon Redshift 資料倉儲是稱為節點的運算資源的集合，組織成稱為叢集的群組。每個叢集都會執行 Amazon Redshift 引擎，並包含一或多個資料庫。

> **Note**
>
> 目前，可用的是 Amazon Redshift 1.0 版引擎。但是，等引擎更新後，會有多個 Amazon Redshift 引擎版本可供選擇。

Amazon Redshift 中的叢集和節點

Amazon Redshift 叢集由節點組成。每個叢集會擁有一個領導節點和一或多個運算節點。領導節點接收來自用戶端應用程式的查詢，剖析查詢，並制定查詢執行計畫。領導節點會協調在運算節點上平行執行這些計畫，並彙總這些節點傳回的中間結果，最後將結果傳回用戶端應用程式。

運算節點會執行查詢執行計畫，並在它們之間傳輸資料以服務這些查詢。中繼結果在傳回用戶端應用程式之前，會傳給領導節點進行彙總。如需領導節點和運算節點的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的資料倉儲系統架構。

> **Note**
>
> 當您在 Amazon Redshift 主控台 (https://console.aws.amazon.com/redshiftv2/：//) 上建立叢集時，您可以根據資料大小和查詢特性取得叢集組態的建議。若要使用此大小計算器，請在支援 RG 或 RA3 節點類型的 AWS 區域中，尋找 主控台上的協助我選擇。如需詳細資訊，請參閱建立叢集。

啟動叢集時，其中一個要指定的選項為節點類型。節點類型會決定每個節點的 CPU、RAM、儲存容量和儲存磁碟機類型。

Amazon Redshift 提供不同的節點類型來容納您的工作負載，我們建議您根據所需的效能、資料大小和預期的資料成長來選擇 RG 或 RA3。

具有受管儲存的 RG 和 RA3 節點可讓您透過獨立擴展和支付運算和受管儲存來最佳化資料倉儲。使用 RG 或 RA3，您可以根據您的效能需求選擇節點數量，並且只需支付您使用的受管儲存費用。根據您

每天處理的資料量來調整 RG 或 RA3 叢集的大小。您可以在虛擬私有雲端 (VPC) 中啟動使用 RG 或 RA3 節點類型的叢集。如需詳細資訊，請參閱在 VPC 中建立 Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組。

Amazon Redshift 受管儲存在每個 RG 和 RA3 節點中使用大型、高效能 SSDs 來提供快速本機儲存，而 Amazon S3 則可提供長期的耐用儲存。如果節點中的資料成長到超過大型本機 SSD 的大小，Amazon Redshift 受管儲存會自動將資料卸載到 Amazon S3。無論資料是在高效能 SSD 或 Amazon S3 中，您都只需要支付與 Amazon Redshift 受管儲存相同的低費率。針對需要不斷增加儲存空間的工作負載，受管儲存可讓您自動擴展資料倉儲儲存容量，與運算節點分開。

RG 節點透過 RA3 節點類型提供下列功能：

- Graviton 型執行個體類型 – 為分析工作負載提供卓越的價格效能
- 整合式資料湖查詢引擎 – RG 叢集包含整合式資料湖查詢引擎，可直接在叢集自己的運算資源上執行。RA3 叢集使用 Amazon Redshift Spectrum 進行資料湖查詢。

DC2 節點讓您建立包含本機 SSD 儲存的運算密集型資料倉儲。您可以根據資料大小和效能需求來選擇所需的節點數目。DC2 節點將資料儲存在本機以提高效能，並且隨著資料大小的增長，您可以新增更多運算節點來增加叢集的儲存容量。針對 1 TB (壓縮) 以下資料集，我們建議使用 DC2 節點，以便以最低的價格獲得最佳效能。如果您預期資料會增長，我們建議使用 RA3 節點，這樣您就可以獨立調整運算和儲存的大小，以獲得改善的價格效能比。使用 DC2 節點類型的叢集需在虛擬私有雲端 (VPC) 中啟動。如需詳細資訊，請參閱在 VPC 中建立 Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組。

節點類型具備不同的大小。節點大小和節點數量決定了叢集的總儲存量。如需詳細資訊，請參閱節點類型詳細資訊。

部分節點類型允許一個節點 (單節點) 或兩個以上的節點 (多節點)。某些節點類型叢集節點的 數量下限是兩個節點。在單節點叢集上，節點由領導者和運算功能共享。不建議執行生產工作負載時使用單一節點叢集。在多節點叢集上，領導節點與運算節點各自獨立。領導節點與運算節點的節點類型相同。您只需支付運算節點的費用。

Amazon Redshift 會將配額套用至每個 AWS 區域中每個 AWS 帳戶的資源。配額會限制您的帳戶可在 AWS 區域內為指定資源類型建立的資源數量，例如節點或快照。如需套用至 Amazon Redshift 資源之預設配額的詳細資訊，請參閱 Amazon Redshift 中的配額和限制。

叢集的成本取決於 AWS 區域、節點類型、節點數量，以及節點是否事先預留。如需節點成本的相關資訊，請參閱 Amazon Redshift 定價頁面。

節點類型詳細資訊

下表摘要說明各節點類型和大小的節點規格。表格中的標頭具有下列意義：

- vCPU 為各節點的虛擬 CPU 數量。
- RAM 為各節點的記憶體容量，以 GiB 為單位。
- 「每個節點的配量數」是在建立叢集或使用傳統調整大小進行大小調整時，運算節點分割成的配量數。

  如果叢集使用彈性調整來調整大小，則每一節點的配量數目可能會改變。不過，在彈性調整大小之後，叢集中所有運算節點上的切片總數仍然相同。

  當您使用來自快照操作的還原建立叢集時，如果您變更了節點類型，結果叢集的配量數可能會與原始叢集不同。
- 儲存為各節點的儲存容量與類型。
- 節點範圍是 Amazon Redshift 針對節點類型和大小所支援的最小和最大節點數量。

> **Note**
>
> 根據套用至所選 AWS 區域中 AWS 帳戶的配額，您可能會受限於較少的節點。如需套用至 Amazon Redshift 資源之預設配額的詳細資訊，請參閱 Amazon Redshift 中的配額和限制。

- 總容量是叢集的總儲存容量，假如您部署了節點範圍中所指定的節點最大數量。

下表說明 RG 節點的規格。

Node type	vCPU	RAM (GiB)	每個節點的預設配量	每個節點的受管儲存限制 [1]	具有建立叢集的節點範圍	受管儲存總容量 [2]	
rg.large（單一節點）	2	16	2	1 TB	1	1 TB	

Node type	vCPU	RAM (GiB)	每個節點的預設配量	每個節點的受管儲存限制 [1]	具有建立叢集的節點範圍	受管儲存總容量 [2]
rg.large（多節點）	2	16	2	8 TB	2–16	128 TB
rg.xlarge（多節點）	4	32	2	32 TB	2–16[3]	1024 TB[3]
rg.4xlarge	16	128	8	128 TB	2–32[4]	8192 TB[4]
rg.12xlarge	48	384	16	128 TB	2–128	16,384 TB

[1] Amazon Redshift 受管儲存的儲存限制。此為硬性限制。

[2] 受管儲存總限制是節點數目上限乘以每個節點的受管儲存限制。

[3] 您可以使用最多有 16 個節點的 rg.xlarge（多節點）節點類型建立叢集。至於多節點叢集，則可以使用彈性調整大小來將大小調整為最多 32 個節點。

[4] 您可以使用最多 32 個節點的 rg.4xlarge（多節點）節點類型建立叢集。您可以使用彈性調整大小將其大小調整為最多 64 個節點。

下表說明 RA3 節點的規格。

Node type	vCPU	RAM (GiB)	每個節點的預設配量	每個節點的受管儲存限制 [1]	具有建立叢集的節點範圍	受管儲存總容量 [2]
ra3.large (單一節點)	2	16	2	1 TB	1	1 TB[3]
ra3.large (多節點)	2	16	2	8 TB	2-16	128 TB

Node type	vCPU	RAM (GiB)	每個節點的預設配量	每個節點的受管儲存限制 [1]	具有建立叢集的節點範圍	受管儲存總容量 [2]
ra3.xlplus (單節點)	4	32	2	4 TB	1	4 TB[3]
ra3.xlplus (多節點)	4	32	2	32 TB	2–16[4]	1024 TB[4]
ra3.4xlarge	12	96	4	128 TB	2–32[5]	8192 TB[5]
ra3.16xlarge	48	384	16	128 TB	2–128	16,384 TB

[1] Amazon Redshift 受管儲存的儲存限制。此為硬性限制。

[2] 受管儲存總限制是節點數目上限乘以每個節點的受管儲存限制。

[3] 若要將單節點叢集的大小調整為多節點，則只支援傳統調整大小。

[4] 您可以使用具有多達 16 個節點的 ra3.xlplus (多節點) 節點類型來建立叢集。至於多節點叢集，則可以使用彈性調整大小來將大小調整為最多 32 個節點。

[5] 您可以使用具有多達 32 個節點的 ra3.4xlarge 節點類型來建立叢集。您可以使用彈性調整大小將其大小調整為最多 64 個節點。

下表說明密集運算節點的規格。

Node type	vCPU	RAM (GiB)	每個節點的預設配量	每個節點的儲存體	節點範圍	總容量
dc2.large	2	15	2	160 GB NVMe-SSD	1–32	5.12 TB

Node type	vCPU	RAM (GiB)	每個節點的預設配量	每個節點的儲存體	節點範圍	總容量
dc2.8xlarge	32	244	16	2.56 TB NVMe-SSD	2–128	326 TB

> **Note**
>
> 不再提供密集儲存 (DS2) 節點類型。

先前的節點類型名稱

在先前的 Amazon Redshift 版本中，特定節點類型具有不同的名稱。您可以在 Amazon Redshift API 和 中使用先前的名稱 AWS CLI。但我們建議您升級任何參考這些名稱的指令碼，使用最新的名稱。目前名稱和先前名稱如下所示。

目前名稱	先前名稱
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

決定節點數目

Amazon Redshift 會將查詢平行分配到所有叢集的運算節點上執行，因此只要在叢集中新增節點，即可提高查詢效能。執行包含至少兩個運算節點的叢集時，每個節點上的資料會鏡射至另一節點的磁碟上，如此可降低資料遺失的風險。

您可以透過 Amazon Redshift 主控台和 Amazon CloudWatch 指標來監控查詢效能。您也可以視需要新增或移除節點，在價格與效能間為您的叢集取得平衡。要求增加節點時，Amazon Redshift 會處理關於部署、負載平衡和資料維護的所有詳細資訊。如需叢集效能的相關資訊，請參閱 監控 Amazon Redshift 叢集效能。

預留節點適合穩定狀態的生產工作負載，且提供較隨需節點更多的折扣。您可以在執行實驗和概念驗證之後購買預留的節點，以驗證您的生產組態。如需詳細資訊，請參閱預留節點。

暫停叢集時，可讓您在叢集暫停期間暫停隨需帳單。在此暫停期間，您只需支付備份儲存的費用。這樣就不必在需求出現之前預先規劃和購買資料倉儲容量，並讓您能在開發或測試時，以經濟實惠的方式管理環境。

如需隨需節點和預留節點的定價相關資訊，請參閱 Amazon Redshift 定價。

使用 EC2 建立叢集

Amazon Redshift 叢集會在針對您所選 Amazon Redshift 節點類型和大小所設定的 Amazon EC2 執行個體中執行。如需這些聯網平台的詳細資訊，請參閱《Amazon EC2 使用者指南》中的支援的平台。

> **Note**
>
> 為避免 SQL 用戶端工具和 Amazon Redshift 資料庫之間發生連線問題，建議您採取下列兩項動作其中之一。設定傳入規則，允許主機協調封包大小。或者，在 Amazon EC2 執行個體的網路界面 (NIC) 上將最大傳輸單位 (MTU) 設定為 1500，停用 TCP/IP 巨型訊框。如需這些方法的相關資訊，請參閱 查詢似乎沒有回應且有時無法觸達叢集。

Amazon Virtual Private Cloud (Amazon VPC)

使用 Amazon VPC 時，您的叢集會在與 AWS 您的帳戶邏輯隔離的虛擬私有雲端 (VPC) 中執行。如果您透過 Amazon VPC 佈建叢集，則可藉由將一或多個 VPC 安全群組與叢集建立關聯的方式，來控制叢集的存取權。如需詳細資訊，請參閱《Amazon VPC 使用者指南》 中的 VPC 的安全群組。

若要在 VPC 中建立叢集，您必須提供 VPC 的子網路資訊，然後在叢集啟動時提供子網路群組，先建立 Amazon Redshift 叢集子網路群組。如需詳細資訊，請參閱Redshift 資源的子網路。

如需 Amazon Virtual Private Cloud (Amazon VPC) 的相關資訊，請參閱 Amazon VPC 產品詳細資訊頁面。

預設磁碟空間警示

建立 Amazon Redshift 叢集時，您可選擇性設定 Amazon CloudWatch 警示來監控叢集中所有節點所使用的磁碟空間平均百分比。我們將此警示稱為預設磁碟空間警示。

預設磁碟空間警示的作用是幫助您監控叢集的儲存容量。您可依據資料倉儲需求設定此警示。例如，您可用警告做為需要調整叢集大小的指示。您可調整叢集大小為不同的節點類型，或新增節點，或是購買預留節點以供日後擴充。

當磁碟使用量達到或超過指定百分比經過一定的次數和指定持續期間，便會觸發預設磁碟空間警示。根據預設，此警示會在達到指定百分比時觸發，並保持在該百分比之上超過五分鐘。您可在啟動叢集後編輯預設值。

當 CloudWatch 警示觸發時，Amazon Simple Notification Service (Amazon SNS) 會傳送通知給指定的收件人，以警告他們已達到百分比閾值。Amazon SNS 會使用主題來指定收件人和要在通知中傳送的訊息。您可用現有的 Amazon SNS 主題，或在啟動叢集時依指定設定建立主題。您可在啟動叢集後編輯此警示的主題。如需建立 Amazon SNS 主題的相關資訊，請參閱開始使用 Amazon Simple Notification Service。

啟動叢集之後，您可以從叢集在 CloudWatch 警示下的狀態視窗檢視和編輯警示。名稱為 percentage-disk-space-used-default-<*string*>。您可開啟警示，以檢視與其相關聯的 Amazon SNS 主題並編輯警示設定。如果您未選取現有的 Amazon SNS 主題來使用，則系統為您建立的主題名為 <*clustername*>-default-alarms (<*recipient*>)；例如，examplecluster-default-alarms (notify@example.com)。

如需設定及編輯預設磁碟空間警示的相關資訊，請參閱 建立叢集 和 建立磁碟空間警示。

> **Note**
>
> 如果刪除叢集，與叢集相關聯的警示將不會刪除，但也不會觸發。如果不再需要警示，可從 CloudWatch 主控台刪除警示。

叢集狀態

叢集狀態顯示叢集的目前狀態。下表提供各叢集狀態的說明。

狀態	說明
available	叢集正在執行且可使用。
available, prep-for-resize	叢集正在準備進行彈性調整大小。叢集在執行中且可用於讀取和寫入查詢，但無法使用某些叢集操作，例如建立快照。
available, resize-cleanup	彈性調整大小操作正在完成將資料傳輸到新叢集節點。叢集在執行中且可用於讀取和寫入查詢，但無法使用某些叢集操作，例如建立快照。
cancelling-resize	調整大小的作業正在取消。
creating	Amazon Redshift 正在建立叢集。如需詳細資訊，請參閱建立叢集。
deleting	Amazon Redshift 正在刪除叢集。如需詳細資訊，請參閱關閉及刪除叢集。
final-snapshot	Amazon Redshift 正在刪除叢集之前為其建立最後快照。如需詳細資訊，請參閱關閉及刪除叢集。
hardware-failure	叢集發生硬體故障。 如果您有一個單一節點的叢集，該節點將無法替換。若要恢復叢集，請還原快照。如需詳細資訊，請參閱Amazon Redshift 快照和備份。
incompatible-hsm	Amazon Redshift 無法連線到硬體安全模組 (HSM)。請檢查叢集和 HSM 之間的 HSM 組態。如需詳細資訊，請參閱使用硬體安全模組進行加密。
incompatible-network	基本網路組態發生問題。確認您啟動叢集所在的 VPC 存在且其設定正確。如需詳細資訊，請參閱VPC 中的 Redshift 資源。
incompatible-parameters	相關聯參數群組中有一或多個參數值發生問題，且無法套用參數值或數值。請修改參數群組並更新任何無效的值。如需詳細資訊，請參閱Amazon Redshift 參數群組。
incompatible-restore	從快照還原叢集時發生問題。請再次嘗試從其他快照還原叢集。如需詳細資訊，請參閱Amazon Redshift 快照和備份。

狀態	說明
modifying	Amazon Redshift 正在套用變更至叢集。如需詳細資訊，請參閱修改叢集。
paused	叢集已暫停。如需詳細資訊，請參閱暫停和繼續叢集。
rebooting	Amazon Redshift 正在重新啟動叢集。如需詳細資訊，請參閱重新啟動叢集。
renaming	Amazon Redshift 正在套用新名稱至叢集。如需詳細資訊，請參閱重新命名叢集。
resizing	Amazon Redshift 正在調整叢集的大小。如需詳細資訊，請參閱調整叢集大小。
rotating-keys	Amazon Redshift 正在輪換叢集的加密金鑰。如需詳細資訊，請參閱加密金鑰輪換。
storage-full	叢集已達到其儲存容量。請調整叢集的大小，以新增節點或選擇不同的節點大小。如需詳細資訊，請參閱調整叢集大小。
updating-hsm	Amazon Redshift 正在更新 HSM 組態。

使用 Amazon Redshift 佈建叢集的考量事項

建立叢集之後，您可以在本節中找到有關哪些區域中可使用功能、維護任務、節點類型和用量限制等資訊。

區域與可用區域的考量

Amazon Redshift 可在數個 AWS 區域使用。根據預設，Amazon Redshift 會將您的叢集佈建在您選擇的 AWS 區域內隨機選取的可用區域 (AZ) 中。所有叢集節點將佈建在相同的可用區域中。

如果該區域可使用 Amazon Redshift，您可以選擇請求該特定可用區域。例如，若您有一個 Amazon EC2 執行個體在一個可用區域內執行，您可能會想將 Amazon Redshift 叢集建立在相同的區域內，以降低延遲。另一方面，您可能會想選擇其他可用區域以獲得更高的可用性。Amazon Redshift 可能無法在 AWS 區域內的所有可用區域中使用。

如需可佈建 Amazon Redshift 叢集的支援 AWS 區域清單，請參閱 中的 Amazon Redshift 端點Amazon Web Services 一般參考。

叢集維護

Amazon Redshift 會定期執行維護，以套用升級至您的叢集。在這些更新期間，您的 Amazon Redshift 叢集無法執行正常的操作。您有多種方法可以控制我們如何維護您的叢集。例如，您可以控制我們何時將更新部署到您的叢集。您也可以選擇是否讓叢集一律執行最新發行的版本，或在最新發行版本之前發行的版本。最後，您可以選擇將非強制的維護更新延遲為一段期間。

維護時段

Amazon Redshift 會從每個 AWS 區域的 8 小時時段中隨機指派 30 分鐘的維護時段，該時段發生在一週的隨機日期 （週一至週日，包含)。

預設維護時段

下列清單顯示指派預設維護時段的每個 AWS 區域的時段：

- 美國東部 (維吉尼亞北部) 區域：03:00–11:00 UTC
- 美國東部 (俄亥俄) 區域：03:00–11:00 UTC
- 美國西部 (加利佛尼亞北部) 區域：06:00–14:00 UTC
- 美國西部 (奧勒岡) 區域：06:00–14:00 UTC
- 非洲 (開普敦) 區域：20:00–04:00 UTC
- 亞太區域 (香港) 區域：13:00–21:00 UTC
- 亞太區域 (海德拉巴) 區域：16:30–00:30 UTC
- 亞太區域 (雅加達) 區域：15:00–23:00 UTC
- 亞太區域 (馬來西亞) 區域：14:00–22:00 UTC
- 亞太區域 (墨爾本) 區域：12:00–20:00 UTC
- 亞太區域 (孟買) 區域：16:30–00:30 UTC
- 亞太區域 (紐西蘭) 區域：10:00–18:00 UTC
- 亞太區域 (大阪) 區域：13:00–21:00 UTC
- 亞太區域 (首爾) 區域：13:00–21:00 UTC
- 亞太區域 (新加坡) 區域：14:00–22:00 UTC
- 亞太區域 (雪梨) 區域：12:00–20:00 UTC

- 亞太區域 (台北) 區域：14:00–22:00 UTC
- 亞太區域 (泰國) 區域：15:00–23:00 UTC
- 亞太區域 (東京) 區域：13:00–21:00 UTC
- 加拿大 (中部) 區域：03:00–11:00 UTC
- 加拿大西部 (卡加利) 區域：04:00–12:00 UTC
- 中國 (北京) 區域：13:00–21:00 UTC
- 中國 (寧夏) 區域：13:00–21:00 UTC
- 歐洲 (法蘭克福) 區域：06:00–14:00 UTC
- 歐洲 (愛爾蘭) 區域：22:00–06:00 UTC
- 歐洲 (倫敦) 區域：22:00–06:00 UTC
- 歐洲 (米蘭) 區域：21:00–05:00 UTC
- 歐洲 (巴黎) 區域：23:00–07:00 UTC
- 歐洲 (斯德哥爾摩) 區域：23:00–07:00 UTC
- 歐洲 (蘇黎世) 區域：20:00–04:00 UTC
- 以色列 (特拉維夫) 區域：20:00–04:00 UTC
- 墨西哥 (中部) 區域：04:00–12:00 UTC
- 歐洲 (西班牙) 區域：21:00–05:00 UTC
- 中東 (巴林) 區域：13:00–21:00 UTC
- 中東 (阿拉伯聯合大公國) 區域：18:00–02:00 UTC
- 南美洲 (聖保羅) 區域：19:00–03:00 UTC

如果在特定週排程維護事件，事件會在指定的 30 分鐘維護時段內啟動。Amazon Redshift 執行維護時，會終止任何查詢或其他進行中的操作。在 Amazon Redshift 服務水準協議中，計劃維護期間遇到的停機時間不會被視為每月停機時間或無法使用。多數的維護會在 30 分鐘維護時段內完成，但部分的維護作業可能會在時段結束後繼續執行。如果排程的維護時段內沒有要執行的維護作業，叢集將繼續正常運作，直到下次排程的維護時段。

您可以透過程式設計的方式或使用 Amazon Redshift 主控台修改叢集，以變更排程的維護時段。您可以在維護索引標籤下找到維護時段，並為叢集設定進行維護的日期和時間。

叢集有可能在維護時段外重新啟動。有幾個原因會導致這種情況發生。還有一個常見原因是叢集偵測到問題，因此系統執行維護操作以使叢集恢復良好狀態。如需詳細資訊，請參閱為何我的 Amazon Redshift 叢集在維護時段外重新啟動？一文，其中會提供有關為何會發生這種情況的詳細資訊。

延遲維護

若要重新排程叢集的維護時段，您可以將維護延遲最多 60 天。例如，如果叢集的維護時段設為星期三 08:30 – 09:00 UTC，而您需要在這個時間存取叢集，則可以將維護時段延後。

如果您延後維護時段，Amazon Redshift 仍會對叢集套用硬體更新或其他強制性的安全更新。在這些更新期間，您的叢集無法使用。

如果在維護時段即將到來時排定了硬體更新或其他強制性的安全更新，Amazon Redshift 會在待定類別下傳送預先通知給您。若要進一步了解待定事件通知，請參閱 Amazon Redshift 佈建叢集事件通知。

您也可以選擇從 Amazon Simple Notification Service (Amazon SNS) 接收事件通知。如需訂閱 Amazon SNS 事件通知的相關資訊，請參閱Amazon Redshift 叢集事件通知訂閱。

如果您延遲叢集的維護，就無法再將所延遲時段後的下一個維護時段延遲。

> **Note**
>
> 維護開始後就不能延遲。

如需叢集維護的相關資訊，請參閱以下文件：

- 維護時段
- 叢集操作
- 修改叢集

選擇叢集維護軌道

當 Amazon Redshift 發行新的叢集版本時，叢集會在維護時段內更新。您可控制是否要將叢集更新至最新版或前一版。

追蹤可控制在維護時段內套用哪一個叢集版本。當 Amazon Redshift 發行新的叢集版本時，該版本會指派到「目前」軌道，而先前版本會指派到「追蹤」軌道。

如需有關叢集追蹤的資訊，請參閱 Amazon Redshift 佈建叢集和無伺服器工作群組的追蹤。

了解 RG 和 RA3 節點如何分隔運算和儲存

這些區段會詳細說明 RG 和 RA3 節點類型可用的任務，顯示其對一系列使用案例的適用性，並詳細說明其相較於先前可用節點類型的優點。

RG 和 RA3 節點的優點和可用性

RG 和 RA3 節點提供下列優點：

- 其具備彈性，可在不增加儲存成本的情況下增加您的運算容量。且其可在不過度佈建運算容量的情況下擴展您的儲存體。
- 其針對您的熱資料使用高效能 SSD，針對冷資料則使用 Amazon S3。因此，這些節點提供了易於使用且符合成本效益的儲存體及高查詢效能。
- 它們使用建置在 AWS Nitro 系統上的高頻寬聯網，進一步縮短資料卸載至 Amazon S3 並從中擷取所需的時間。

考慮在這些情況下選擇 RG 和 RA3 節點類型：

- 您需要與儲存體分開擴展運算及為其支付費用的彈性時。
- 您只查詢總資料的一小部分。
- 您的資料量正在迅速成長或預期會快速成長。
- 您希望靈活地根據您的效能需求調整叢集大小。

資料湖查詢運算考量 – 在 DC2 和 RA3 佈建叢集上，資料湖查詢會在 Redshift Spectrum 上執行，這會使用獨立於叢集的專用 Amazon Redshift 伺服器。在 RG 佈建叢集上，資料湖查詢會在叢集自己的運算資源上執行，並與其他工作負載共用這些資源。對於具有大量資料湖查詢用量的工作負載，請在調整 RG 叢集大小時考慮這一點。

若要使用 RG 節點類型，您的 AWS 區域必須支援 RG。如需詳細資訊，請參閱AWS 區域中的 RG 節點類型可用性。

若要使用 RA3 節點類型，您的 AWS 區域必須支援 RA3。如需詳細資訊，請參閱AWS 區域中的 RA3 節點類型可用性。

> **Important**
>
> 您只能搭配 1.0.21262 或更新版本的叢集版本使用 ra3.xlplus 節點類型。您可以使用 Amazon Redshift 主控台檢視現有叢集的版本。如需詳細資訊，請參閱判斷工作群組或叢集版本。
> 使用 RG 或 RA3 節點類型時，請務必使用新的 Amazon Redshift 主控台。
> 此外，若要搭配使用軌道的 Amazon Redshift 操作使用 RG 或 RA3 節點類型，維護軌道值必須設定為支援 RG 或 RA3 的叢集版本。如需追蹤的詳細資訊，請參閱 選擇叢集維護軌道。

在使用單節點 RA3 節點類型時，請考量下列事項。

- 支援資料共用生產者和取用者。
- 若要變更節點類型，則只支援傳統調整大小。不支援使用彈性調整大小或快照還原來變更節點類型。支援以下情況：
  - 透過傳統調整大小將單節點 dc2.xlarge 變更為單節點 ra3.xlplus，反之亦然。
  - 透過傳統調整大小將單節點 dc2.xlarge 變更為多節點 ra3.xlplus，反之亦然。
  - 透過傳統調整大小將多節點 dc2.xlarge 變更為單節點 ra3.xlplus，反之亦然。

使用 Amazon Redshift 受管儲存

透過 Amazon Redshift 受管儲存，您可以在 Amazon Redshift 中儲存和處理您所有的資料，同時取得更大的彈性來分別擴展運算和儲存容量。您可以繼續使用 COPY 或 INSERT 命令來擷取資料。為了最佳化效能和管理不同儲存層的自動資料置放，Amazon Redshift 會利用最佳化 (例如資料區塊溫度、資料區塊年齡和工作負載模式)。當需要時，Amazon Redshift 或自動將儲存擴展至 Amazon S3，無須任何手動動作。

如需有關儲存成本的資訊，請參閱 Amazon Redshift 定價。

管理 RG 或 RA3 節點類型

若要利用從儲存體分離運算，您可以使用 RG 或 RA3 節點類型建立或升級叢集。若要使用 RG 或 RA3 節點類型，請在虛擬私有雲端 (EC2-VPC) 中建立叢集。

若要使用 RG 或 RA3 節點類型變更 Amazon Redshift 叢集的節點數量，請執行下列其中一項操作：

- 使用彈性調整大小操作新增或移除節點。在某些情況下，不允許使用彈性調整大小從 RG 或 RA3 叢集移除節點。例如，當 2:1 節點計數升級將每個節點的配量數設為 32 時。如需詳細資訊，請參閱調整叢集大小。如果無法使用彈性調整大小，請使用傳統調整大小。
- 使用傳統調整大小操作新增或移除節點。當您調整大小至無法透過彈性調整大小來使用的組態時，請選擇此選項。彈性調整大小比傳統調整大小更快。如需詳細資訊，請參閱調整叢集大小。

AWS 區域中的 RG 節點類型可用性

RG 節點類型僅適用於下列 AWS 區域：

- 美國東部 (維吉尼亞北部) 區域 (us-east-1)
- 美國東部 (俄亥俄) 區域 (us-east-2)

- 美國西部 (加利佛尼亞北部) 區域 (us-west-1)
- 美國西部 (奧勒岡) 區域 (us-west-2)
- 非洲 (開普敦) 區域 (af-south-1)
- 亞太區域 (香港) 區域 (ap-east-1)
- 亞太區域 (台北) 區域 (ap-east-2)
- 亞太區域 (東京) 區域 (ap-northeast-1)
- 亞太區域 (首爾) 區域 (ap-northeast-2)
- 亞太區域 (大阪) (ap-northeast-3)
- 亞太區域 (孟買) 區域 (ap-south-1)
- 亞太區域 (海德拉巴) 區域 (ap-south-2)
- 亞太區域 (新加坡) 區域 (ap-southeast-1)
- 亞太區域 (雪梨) 區域 (ap-southeast-2)
- 亞太區域 (雅加達) 區域 (ap-southeast-3)
- 亞太區域 (墨爾本) 區域 (ap-southeast-4)
- 亞太區域 (馬來西亞) 區域 (ap-southeast-5)
- 亞太區域 (泰國) (ap-southeast-7)
- 加拿大 (中部) 區域 (ca-central-1)
- 歐洲 (法蘭克福) 區域 (eu-central-1)
- 歐洲 (蘇黎世) 區域 (eu-central-2)
- 歐洲 (斯德哥爾摩) 區域 (eu-north-1)
- 歐洲 （ 米蘭 ） 區域 (eu-south-1) （ 僅限 rg.xlarge 和 rg.4xlarge)
- 歐洲 (西班牙) 區域 (eu-south-2)
- 歐洲 (愛爾蘭) 區域 (eu-west-1)
- 歐洲 (倫敦) 區域 (eu-west-2)
- 歐洲 (巴黎) 區域 (eu-west-3)
- 墨西哥 (中部) (mx-central-1)
- 南美洲 (聖保羅) 區域 (sa-east-1)
- AWS GovCloud （ 美國東部 ） (us-gov-east-1)

- AWS GovCloud （美國西部）(us-gov-west-1)

AWS 區域中的 RA3 節點類型可用性

RA3 節點類型僅適用於下列 AWS 區域：

- 美國東部 (維吉尼亞北部) 區域 (us-east-1)
- 美國東部 (俄亥俄) 區域 (us-east-2)
- 美國西部 (加利佛尼亞北部) 區域 (us-west-1)
- 美國西部 (奧勒岡) 區域 (us-west-2)
- 非洲 (開普敦) 區域 (af-south-1)
- 亞太區域 (香港) 區域 (ap-east-1)
- 亞太區域 (海德拉巴) 區域 (ap-south-2)
- 亞太區域 (雅加達) 區域 (ap-southeast-3)
- 亞太區域 (馬來西亞) 區域 (ap-southeast-5)
- 亞太區域 (墨爾本) 區域 (ap-southeast-4)
- 亞太區域 (孟買) 區域 (ap-south-1)
- 亞太區域 (紐西蘭) 區域 (ap-southeast-6)
- 亞太區域 (大阪) (ap-northeast-3)
- 亞太區域 (首爾) 區域 (ap-northeast-2)
- 亞太區域 (新加坡) 區域 (ap-southeast-1)
- 亞太區域 (雪梨) 區域 (ap-southeast-2)
- 亞太區域 (台北) 區域 (ap-east-2)
- 亞太區域 (泰國) (ap-southeast-7)
- 亞太區域 (東京) 區域 (ap-northeast-1)
- 加拿大 (中部) 區域 (ca-central-1)
- 加拿大西部 (卡加利) 區域 (ca-west-1)
- 中國 (北京) 區域 (cn-north-1)
- 中國 (寧夏) 區域 (cn-northwest-1)
- 歐洲 (法蘭克福) 區域 (eu-central-1)

- 歐洲 (蘇黎世) 區域 (eu-central-2)
- 歐洲 (愛爾蘭) 區域 (eu-west-1)
- 歐洲 (倫敦) 區域 (eu-west-2)
- 歐洲 (米蘭) 區域 (eu-south-1)
- 歐洲 (西班牙) 區域 (eu-south-2)
- 歐洲 (巴黎) 區域 (eu-west-3)
- 歐洲 (斯德哥爾摩) 區域 (eu-north-1)
- 以色列 (特拉維夫) 區域 (il-central-1)
- 墨西哥 (中部) (mx-central-1)
- 中東 (巴林) 區域 (me-south-1)
- 中東 (阿拉伯聯合大公國) 區域 (me-central-1)
- 南美洲 (聖保羅) 區域 (sa-east-1)
- AWS GovCloud （美國東部）(us-gov-east-1)
- AWS GovCloud （美國西部）(us-gov-west-1)

升級到 RG 或 RA3 節點類型

若要將現有節點類型升級至 RG 或 RA3，您有下列選項可變更節點類型：

- 從快照還原 – Amazon Redshift 會使用叢集的最新快照，並將其還原以建立新的 RG 或 RA3 叢集。一旦叢集建立完成（通常在幾分鐘內），RG 或 RA3 節點就可以執行完整的生產工作負載。由於運算與儲存體分開，熱資料可受益於大型的聯網頻寬，快速地傳入本機快取。如果您從最新的 DC2 快照還原，RG 和 RA3 會保留 DC2 工作負載的熱區塊資訊，並將最熱的區塊填入其本機快取。如需詳細資訊，請參閱從快照還原叢集。

  若要為應用程式和使用者保留相同的端點，您可以使用與原始 DC2 叢集相同的名稱重新命名新的 RG 或 RA3 叢集。若要為此叢集重新命名，請在 Amazon Redshift 主控台或 `ModifyCluster` API 操作中修改叢集。如需詳細資訊，請參閱《Amazon Redshift API 參考》中的 重新命名叢集 或 `ModifyCluster` API 操作。
- 彈性調整大小 — 使用彈性調整大小調整叢集的大小。當您使用彈性調整大小來變更節點類型時，Amazon Redshift 會自動建立快照、建立新叢集、刪除舊叢集，並重新命名新的叢集。彈性調整大小操作可隨時執行，或排程在未來特定的時間執行。您可以使用彈性調整大小，將現有的 DC2 節點類型叢集快速升級至 RG 或 RA3。如需詳細資訊，請參閱彈性調整大小。

> **Note**
>
> 若要將現有的 RA3 叢集遷移至 rg.xlarge 或 rg.4xlarge，您的來源叢集必須執行修補程式版本 P201 或更新版本。若要遷移至 rg.large 或 rg.12xlarge，您的來源叢集必須執行 P202 或更新版本。若要建立或遷移至單一節點 rg.large 叢集，您的叢集必須執行 P204 或更新版本。無論修補程式版本為何，DC2 叢集都可以遷移至 RG。

下表顯示升級至 RG 節點類型時的建議。

此表中的建議是啟動叢集節點類型和大小，但取決於工作負載的運算需求。RG 具有不同的效能特性，因此建議的映射是起點。為您的關鍵工作負載建立基準，並根據觀察到的效能調整節點數量。相較於 RA3，每個 vCPU 的 RG 執行個體價格較低，因此新增節點以最佳化效能仍然可以降低整體成本。如需節點類型定價的詳細資訊，請參閱 Amazon Redshift 定價。

為了更準確評估您的需求，請考慮執行概念驗證 (POC)，使用 Test Drive 來執行可能的組態。為您的 POC 資料倉儲佈建叢集，而非 Redshift Serverless。如需執行概念驗證的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的執行 Amazon Redshift 的概念驗證 (POC)。

現有節點類型	現有節點數目	建議的新節點類型	升級動作
ra3.16xlarge	2–128	rg.12xlarge	從每 1 個 ra3.16xlarge1 節點 1 個 rg.12xlarge 節點開始。
ra3.4xlarge	2–64	rg.4xlarge	從每 4 個 ra3.4xlarge[1] 節點的 3 個 rg.4xlarge 節點開始。
ra3.xlplus	1	rg.large	從單一節點 ra3.xlplus 叢集的 rg.large 2 個節點開始[1, 2]。
ra3.xlplus	2–32	rg.xlarge	從每 1 個 ra3.xlplus1 節點的 rg.xlarge 節點開始。

現有節點類型	現有節點數目	建議的新節點類型	升級動作
ra3.large	1–16	rg.large	從每 1 個節點的 ra3.large[1, 2] 的 rg.large 節點開始。
dc2.8xlarge	2–15	rg.4xlarge	從每 2 個 dc2.8xlarge[1] 節點的 3 個 rg.4xlarge 節點開始。
dc2.8xlarge	16–128	rg.12xlarge	從每 2 個 dc2.8xlarge1 節點 1 個 rg.[1]2xlarge 節點開始。
dc2.large	1–3	rg.large	從 dc2.large1[, 2] 每 1 個節點的 rg.large 節點開始。
dc2.large	4	rg.large	從 rg.large[1] 的 3 個節點開始。
dc2.large	5–15	rg.xlarge	從 dc2.large[1] 每 8 個節點的 rg.xlarge 3 個節點開始。
dc2.large	16–32	rg.4xlarge	從每 10 個 dc2.large 1 節點 1 個 rg.4xlarge 節點開始。

[1]視工作負載需求而定，可能需要額外的節點。根據所需查詢效能的運算需求新增或移除節點。相較於 RA3，每個 vCPU 的 RG 執行個體價格較低，因此相較於先前的組態，新增節點仍然可以降低整體成本。如需詳細資訊，請參閱 Amazon Redshift 定價。

[2]如需單一節點叢集，請參閱 下的單一節點叢集升級到 RG 節點時需要考慮的事項。

下表顯示升級至 RA3 節點類型時的建議事項。

此表中的建議是啟動叢集節點類型和大小，但取決於工作負載的運算需求。為了更準確評估您的需求，請考慮執行概念驗證 (POC)，使用 Test Drive 來執行可能的組態。為您的 POC 資料倉儲佈建叢集，而非 Redshift Serverless。如需執行概念驗證的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的執行 Amazon Redshift 的概念驗證 (POC)。

現有節點類型	現有節點數目	建議的新節點類型	升級動作
dc2.8xlarge	2–15	ra3.4xlarge	每 1 個 dc2.8xlarge 節點從 2 個 ra3.4xlarge 節點開始[1]。
dc2.8xlarge	16–128	ra3.16xlarge	每 2 個 dc2.8xlarge 節點從 1 個 ra3.16xlarge 節點開始[1]。
dc2.large	1–4	ra3.large	每 1 個 dc2.large 節點從 1 個 ra3.large 節點開始[1]。 每 2 個 dc2.large 節點從 2 個 ra3.large 節點開始[1]。 每 3 個 dc2.large 節點從 3 個 ra3.large 節點開始[1]。 每 4 個 dc2.large 節點從 3 個 ra3.large 節點開始[1]。
dc2.large	5–15	ra3.xlplus	每 8 個 dc2.large 節點從 3 個 ra3.xlplus 節點開始[1]。
dc2.large	16–32	ra3.4xlarge	每 8 個 dc2.large 節點從 1 個 ra3.4xlarge 節點開始[1,2]。

[1]視工作負載需求而定，可能需要額外的節點。根據所需查詢效能的運算需求新增或移除節點。

[2]具有 dc2.large 節點類型的叢集會限制只能有 32 個節點。

某些 RA3 節點類型的節點數量下限是 2 個節點。在建立 RA3 叢集時，請考慮這一點。

RG 和 RA3 節點支援的聯網功能

RG 和 RA3 節點支援網路功能的集合，不適用於其他節點類型。本節提供每項功能的簡短說明，以及其他文件的連結：

- 佈建叢集 VPC 端點 – 當您建立或還原 RG 或 RA3 叢集時，Amazon Redshift 會使用 5431-5455 或 8191-8215 範圍內的連接埠。當叢集設為這些範圍之一的連接埠時，Amazon Redshift 會自動在 AWS 您的帳戶中為叢集建立 VPC 端點，並將私有 IP 地址連接到該端點。如果您將叢集設定為可公開存取，則 Redshift 會在您的 AWS 帳戶中建立彈性 IP 位址，並將其附加至 VPC 端點。如虛詳細資訊，請參閱設定 Amazon Redshift 叢集或 Amazon Redshift Serverless 工作群組的安全群組通訊設定。
- 單一子網路 RG 或 RA3 叢集 – 您可以使用單一子網路建立 RG 或 RA3 叢集，但無法使用災難復原功能。如果您在子網路沒有多個可用區域 (AZ) 時啟用叢集重新放置，就會發生例外狀況。
- 多子網路 RG 或 RA3 叢集和子網路群組 – 您可以在虛擬私有雲端 (VPC) 中佈建叢集時，建立具有多個子網路的 RG 或 RA3 叢集。叢集子網路群組可讓您在 VPC 中指定一組子網路，且 Amazon Redshift 會在其中一個子網路中建立叢集。在建立子網路群組後，您可以移除先前新增的子網路或新增更多子網路。如需詳細資訊，請參閱 Amazon Redshift 叢集子網路群組。
- 跨帳戶或跨 VPC 端點存取 - 您可以設定 Redshift 受管 VPC 端點來存取佈建叢集或 Amazon Redshift Serverless 工作群組。例如，您可以將其設定為包含叢集或工作群組的 VPC 與執行用戶端工具的 VPC 之間的私有連線。使用這種方法，您不需使用公有 IP 位址或透過網際網路來路由流量，就可以存取資料倉儲。如需詳細資訊，請參閱使用 Redshift 受管 VPC 端點。
- 叢集重新放置 - 當服務中斷時，您可以將叢集移至另一個可用區域 (AZ)，而不會遺失任何資料。您可以在主控台上啟用此功能。如需詳細資訊，請參閱重新定位叢集。
- 自訂網域名稱 — 您可以為 Amazon Redshift 叢集建立自訂網域名稱 (也稱為自訂 URL)。其為易讀的 DNS 記錄，可將 SQL 用戶端連線路由到您的叢集端點。如需詳細資訊，請參閱自訂網域名稱來進行用戶端連線。

升級到 RG 節點時需要考慮的事項

修補程式版本

若要從 RA3 遷移至 RG，您的來源叢集必須符合最低必要修補程式版本：rg.xlarge 為 P201，rg.4xlarge 為 P202，多節點 rg.large 為 P202，單一節點 rg.large 為 rg.12xlarge 和 P204。無論修補程式版本為何，DC2 叢集都可以遷移至 RG。在遷移之前，請先將快照還原至 RG 叢集，再移動生產流量，以測試您的工作負載。如需詳細資訊，請參閱 Amazon Redshift 修補程式 201Amazon Redshift 修補程式 202和 Amazon Redshift 修補程式 204。

單一節點叢集

- 只有 rg.large 支援單一節點 RG 叢集。
- 單一節點 rg.large 叢集最多支援 1 TB 的受管儲存，相較於單一節點 ra3.xlplus 叢集最多支援 4 TB。如果您的資料大於 1 TB 或預期增長超過 1 TB，請使用多節點 rg.large 叢集。
- 單一節點叢集適用於開發和測試工作負載。對於生產工作負載，請使用至少兩個節點。如需詳細資訊，請參閱Amazon Redshift 中的叢集和節點及節點類型詳細資訊。

Spectrum 工作負載

具有大量使用 Spectrum 之工作負載的叢集在遷移至 RG 後，可能會觀察到較慢的查詢效能。與 RA3 和 DC2 不同，RG 會直接在叢集運算節點上執行整合式資料湖查詢引擎，而不是在單獨的 Spectrum 機群上執行。如果您遇到 Spectrum 效能降低，請增加 RG 叢集中的節點數量，或遷移至較大的 RG 節點類型。

RG 叢集也支援查詢位於不同 AWS 區域的 Amazon S3 儲存貯體中的資料，移除適用於使用 Spectrum 的 RA3 和 DC2 叢集的相同區域限制。跨區域查詢會產生額外的資料傳輸費用。如需詳細資訊，請參閱查詢您的資料湖。

游標型工作負載

相較於 RA3 和 DC2，RG 節點具有不同的支援游標大小。如需游標大小限制，請參閱游標限制。如果您的工作負載高度依賴游標，而您遇到游標大小限制，請遷移至較大的 RG 節點類型 — 您可以按比例減少節點數量，以維持類似的總叢集大小和成本。

並行擴展資格

如果您的叢集使用並行擴展且有超過 32 個節點，而且您想要遷移至 rg.12xlarge，請使用彈性調整大小或快照還原來維護並行擴展支援。如果您的叢集有超過 32 個節點，而且您執行傳統大小調整為 rg.12xlarge，則叢集會失去並行擴展支援。超過 32 個節點的新建立 rg.12xlarge 叢集不支援並行擴展。如需詳細資訊，請參閱並行擴展候選項目。

預留節點

如果您的叢集使用預留節點，變更節點類型不會自動變更保留 – 現有的保留會與原始節點類型保持關聯，直到其期限結束為止。遷移使用預留節點的叢集之前，請參閱 在使用預留執行個體的叢集上變更節點類型之前需要考量的事項。

叢集操作

建立叢集之後，您可以執行叢集操作來最佳化效能、控制成本，以及確保高可用性。叢集操作可讓您隨著資料倉儲需求的演進，調整叢集的大小、暫停、繼續，甚至重新建立叢集。

常見的使用案例包括擴展高峰工作負載的運算容量、在非作用中期間暫停叢集以降低成本，以及使用不同的組態或在不同的可用區域中重新建立叢集以進行災難復原。下列各節涵蓋執行各種叢集操作的詳細資訊，協助您有效管理 Amazon Redshift 環境。

建立叢集

您可以使用 Amazon Redshift 建立佈建叢集來啟動新的資料倉儲。佈建叢集是稱為節點的運算資源集合，會組織成單一大量平行處理 (MPP) 系統。

建立叢集之前，請閱讀Amazon Redshift 佈建叢集和 Amazon Redshift 中的叢集和節點。

建立叢集

1. 登入 AWS 管理主控台 ，並在 https：//https://console.aws.amazon.com/redshiftv2/ 開啟 Amazon Redshift 主控台。
2. 在導覽選單上，選擇叢集。系統會 AWS 列出目前區域中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。
3. 選擇 Create cluster (建立叢集) 以建立叢集。
4. 遵循主控台頁面說明以輸入 Cluster configuration (叢集組態) 的屬性。

   下列步驟說明在支援 RG 或 RA3 節點類型的 中執行 AWS 區域 的 Amazon Redshift 主控台。如需 AWS 區域 支援 RG 或 RA3 節點類型的清單，請參閱《Amazon Redshift 管理指南》中的 AWS 區域中的 RG 節點類型可用性和 AWS 區域中的 RA3 節點類型可用性。

   如果您不知道要將叢集的大小調整到多大，請選擇協助我選擇。這樣做會啟動調整大小計算器，詢問您有關計劃儲存在資料倉儲中之資料的大小和查詢特性的問題。如果您知道叢集的所需大小 (也就是節點類型和節點數目)，請選擇我會選擇。然後選擇節點類型和節點數量來調整叢集的大小以進行概念驗證。

> **Note**
>
> 如果您的組織符合資格，而且您的叢集是在未提供 Amazon Redshift Serverless 的 AWS 區域 中建立，您也許可以在 Amazon Redshift 免費試用方案下建立叢集。選擇生產或免費試用來回答問題您打算將此叢集用於什麼目的？ 選擇免費試用時，您可以使用 ra3.large 節點類型建立組態。如需選擇免費試用的詳細資訊，請參閱 Amazon Redshift 免費試用版。如需 Amazon Redshift Serverless 可用 AWS 區域 位置的清單，請參閱 中針對 Redshift Serverless API 列出的端點Amazon Web Services 一般參考。

5. 在資料庫組態區段中，指定管理員使用者名稱的值。針對管理員密碼，您可以從下列選項中選擇：

   - 產生密碼 - 使用 Amazon Redshift 產生的密碼。
   - 手動新增管理員密碼 - 使用您自己的密碼。
   - 在 中管理管理員登入 AWS Secrets Manager資料 – Amazon Redshift 使用 AWS Secrets Manager 來產生和管理管理員密碼。使用 AWS Secrets Manager 產生和管理密碼的秘密會產生費用。如需 AWS Secrets Manager 定價的資訊，請參閱 AWS Secrets Manager 定價。

6. (選用) 遵循主控台頁面說明以輸入 Cluster permissions (叢集許可) 的屬性。如果您的叢集需要為您存取其他 AWS 服務，例如從 Amazon S3 載入資料，請提供叢集許可。

7. 選擇 Create cluster (建立叢集) 以建立叢集。該叢集可能需要幾分鐘方可使用。

其他組態

當您建立叢集時，您可以指定其他屬性進行自訂。您可以在下列清單中找到部分屬性的詳細資訊。

IP 地址類型

選擇叢集的 IP 地址類型。您可以選擇讓資源僅透過 IPv4 定址通訊協定進行通訊，或選擇雙堆疊模式，讓資源同時透過 IPv4 和 IPv6 進行通訊。此功能僅適用於 AWS GovCloud （美國東部） 和 AWS GovCloud （美國西部） 區域。如需 AWS 區域的詳細資訊，請參閱 區域和可用區域。

Virtual Private Cloud (VPC) (虛擬私有雲端 (VPC))

選擇具有叢集子網路群組的 VPC。建立叢集後，就無法變更叢集子網路群組。

參數群組

選擇要與該叢集建立關聯的叢集參數群組。如果您沒有選擇，則該叢集會使用預設參數群組。

加密

選擇您是否要加密叢集及其快照內的所有資料。如果您保留預設設定為 None (無)，加密將不會啟用。如果您想要啟用加密，請選擇要使用 AWS Key Management Service (AWS KMS) 還是硬體安全模組 (HSM)，然後設定相關設定。如需 Amazon Redshift 中加密的相關資訊，請參閱 Amazon Redshift 資料庫加密。

- KMS

  如果您想要啟用加密並使用 AWS KMS 來管理您的加密金鑰，請選擇使用 AWS Key Management Service (AWS KMS)。此外，也請選擇要使用的金鑰。您可以選擇預設金鑰、目前帳戶的金鑰或其他帳戶的金鑰。

  > **Note**
  >
  > 如果您想要使用另一個 AWS 帳戶的金鑰，請輸入要使用的金鑰的 Amazon Resource Name (ARN)。您必須擁有許可才能使用該金鑰。如需有關 中金鑰存取的詳細資訊 AWS KMS，請參閱《 AWS Key Management Service 開發人員指南》中的控制對金鑰的存取。

  如需在 Amazon Redshift 中使用 AWS KMS 加密金鑰的詳細資訊，請參閱 使用 加密 AWS KMS。

- HSM

  如果您想啟用加密並使用硬體安全模組 (HSM) 來管理您的加密金鑰，請選擇 HSM (HSM)。

  若您選擇 HSM，請從 HSM Connection (HSM 連線) 與 HSM Client Certificate (HSM 用戶端憑證) 選擇值。Amazon Redshift 和 HSM 需要這些值才能構建可傳遞叢集金鑰的信任連線。您必須在啟動叢集前先在 Amazon Redshift 中設定 HSM 連線與用戶端憑證。如需設定 HSM 連線與用戶端憑證的相關資訊，請參閱 使用硬體安全模組進行加密。

維護軌道

您可以選擇所使用的叢集版本是否為 Current (目前)、Trailing (追蹤) 或 (有時) Preview (預覽) 軌道。

監控

您可以選擇是否建立 CloudWatch 警示。

設定跨區域快照

您可以選擇是否啟用跨區域快照。

自動快照保留期

您可以在 35 天的範圍內選擇這些快照的保留天數。如果節點類型為 DC2，您可以選擇零 (0) 天，表示不建立自動快照。

手動快照保留期

您可以選擇天數或 `Indefinitely` 來保留這些快照。

用於自動最佳化的額外運算資源

您可以選擇是否要配置額外的運算資源來執行自動最佳化，即使在大量使用期間也是如此。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的為自動資料庫最佳化配置額外的運算資源。

建立磁碟空間警示

您可以監控磁碟空間用量，並設定警示，以便在磁碟空間超過叢集的指定閾值時收到通知。建立磁碟空間用量警示可讓您主動管理儲存容量，並防止磁碟空間不足造成的問題，例如查詢失敗或資料擷取錯誤。下列程序會引導您完成建立磁碟空間用量警示的程序。

為叢集建立磁碟空間使用量警示

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇警示。
3. 針對 Actions (動作)，選擇 Create alarm (建立警示)。Create alarm (建立警示) 頁面隨即出現。
4. 遵循頁面說明以進行操作。
5. 選擇 Create alarm (建立警示)。

檢視叢集

檢視叢集可讓您監控和管理叢集的組態、狀態和效能指標。透過檢視叢集詳細資訊，您可以深入了解資源使用率、查詢執行時間和系統運作狀態。下列程序說明如何取得叢集資訊。

檢視叢集

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單上，選擇叢集。系統會 AWS 列出目前區域中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。如果您沒有任何叢集，請選擇 Create cluster (建立叢集) 以建立一個叢集。
3. 選擇清單中的叢集名稱以檢視該叢集的詳細資訊。

修改叢集

當您修改叢集時，以下選項變更會立即套用：

- VPC security groups (VPC 安全群組)
- 可公開存取
- 管理員使用者密碼
- HSM 連接
- HSM Client Certificate (HSM 用戶端憑證)
- 維護詳細資訊
- Snapshot preferences (快照偏好設定)

只有重新啟動叢集才能使以下選項的變更生效：

- 叢集識別碼

  當您變更叢集識別碼時，Amazon Redshift 會自動重新啟動叢集。
- 增強型 VPC 路由

  當您變更增強型 VPC 路由時，Amazon Redshift 會自動重新啟動叢集。
- 叢集參數群組
- IP 地址類型

  此功能僅適用於 AWS GovCloud （美國東部） 和 AWS GovCloud （美國西部） 區域。如需 AWS 區域的詳細資訊，請參閱區域和可用區域。

若您縮短自動快照保留期間，設定落在新保留期間以外的現有自動快照將被刪除。如需詳細資訊，請參閱Amazon Redshift 快照和備份。

如需叢集屬性的相關資訊，請參閱其他組態。

修改叢集

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集。
3. 選擇要修改的叢集。
4. 選擇編輯。編輯叢集頁面隨即出現。
5. 更新叢集屬性。您可以修改的一些屬性包括：

   - 叢集識別碼
   - 快照保留
   - 叢集重新定位

   為了編輯網路和安全、維護和資料庫組態的設定，主控台會提供適當叢集詳細資訊索引標籤的連結。
6. 選擇儲存變更。

調整叢集大小

當您的資料倉儲容量和效能需要變更時，您可以調整叢集大小，以充分利用 Amazon Redshift 的運算和儲存選項。

當您重新調整叢集規模時，您將指定與目前叢集組態不同的節點數量或節點類型。當叢集正在進行重新規模調整時，您無法對該叢集執行任何寫入或讀寫查詢；您僅能執行讀取查詢。

如需重新調整叢集規模的相關資訊，包括使用不同方法的逐步解說資訊，請參閱 調整叢集大小。

重新調整叢集規模

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集。

3. 選擇要調整大小的叢集。
4. 針對 Actions (動作)，選擇 Resize (調整大小)。Resize cluster (調整叢集大小) 頁面隨即出現。
5. 遵循頁面說明以進行操作。您可以在特定時間立即重新調整叢集的大小一次，或是根據排程增加或減少您叢集的大小。
6. 根據您的選擇，選擇 Resize now (立即調整大小) 或 Schedule resize (排程調整大小)。

如果您有預留節點，您可以升級至 RG 或 RA3 預留節點。您可以在使用主控台從快照還原時或執行彈性調整大小時進行此升級。您可以使用主控台來引導您完成此程序。如需升級至 RG 或 RA3 節點類型的詳細資訊，請參閱升級至 RG 或 RA3 節點類型。

當您執行調整大小操作，以從 DC2.large 節點類型升級至 RA3.large 節點類型時，Amazon Redshift 會自動將交錯排序索引鍵轉換為複合排序索引鍵。此轉換可讓您存取並行擴展功能，此功能不支援對具有交錯排序索引鍵的資料表進行查詢。雖然此自動轉換可確保與 RA3 功能相容，但可能會影響您現有的查詢效能模式。

如果您想要在升級至 RA3 節點後保留交錯排序索引鍵，您可以在調整大小操作完成後，使用所需的排序索引鍵組態重新建立資料表。不過，選擇此選項表示您將無法對這些資料表使用並行擴展。

調整大小操作有兩種類型：

- 彈性調整大小 — 您可以在叢集中新增或移除節點。您也可以變更節點類型，例如從 DC2 節點變更為 RG 或 RA3 節點。彈性調整大小通常會很快完成，平均需要十分鐘。因此，建議您將其作為第一個選項。當您執行彈性調整大小時，它會重新分散資料配量，這些資料配量是在每個節點中配置記憶體和磁碟空間的分割區。當您要進行下列操作時，就適合使用彈性調整大小：
  - 在現有叢集中新增或減少節點，但不變更節點類型 — 這通常稱為「就地」調整大小。當您執行此類型的調整大小時，某些執行中的查詢會成功完成，但其他查詢可能會在操作中遭到捨棄。
  - 變更叢集的節點類型 — 當您變更節點類型時，系統會建立快照，並將資料從來源叢集重新分散至包含新節點類型的叢集。完成後，執行中的查詢會遭到捨棄。和「就地」調整大小一樣，其也會快速完成。
- 傳統調整大小 — 您可以透過類似於彈性調整大小的方式來變更節點類型和 (或) 節點數目。傳統調整大小需要更多時間來完成，但是如果節點計數的變化或要遷移到的節點類型不在彈性調整大小的範圍內，傳統調整大小可能會很有用。例如，當節點計數的變化非常大時，就適用這種方式。

主題

- 彈性調整大小
- Elastic resize (傳統調整大小)

彈性調整大小

當您新增或移除相同類型的節點時，彈性調整大小操作具有下列階段：

1. 彈性調整大小會取得叢集快照。只有 DC2 節點支援無備份資料表。對於 RG 和 RA3 叢集，無備份資料表包含在快照中。如需詳細資訊，請參閱從快照排除資料表。如果您的叢集因為您停用了自動快照而沒有最新的快照，備份操作會需要較長時間。(若要將調整大小操作開始之前的時間最小化，建議您啟用自動快照或在開始調整大小之前建立手動快照。) 當您啟動彈性調整大小且快照操作正在進行時，如果快照操作未在幾分鐘內完成，則調整大小可能會失敗。如需詳細資訊，請參閱Amazon Redshift 快照和備份。
2. 此操作會遷移叢集中繼資料。叢集會有幾分鐘的時間無法使用。大多數查詢會暫時暫停，且會保持連線。但是，某些查詢則可能遭到捨棄。這個階段很快就會結束。
3. 工作階段連線會重新起始，並且查詢會繼續。
4. 彈性調整大小會在背景中將資料重新分散給節點配量。叢集可供讀寫操作，但有些查詢可能需要較長時間來執行。
5. 操作完成後，Amazon Redshift 會傳送事件通知。

當您使用彈性調整大小變更節點類型時，其運作方式與您新增或減少相同類型的節點時相似。首先，系統會建立快照。使用快照中的最新資料佈建新的目標叢集，並在背景中將資料傳輸到新的叢集。在此期間，資料是唯讀狀態。當調整大小操作接近完成時，Amazon Redshift 會更新端點以指向新的叢集，並中斷與來源叢集的所有連線。

彈性調整大小不太可能失敗。不過，若發生失敗，大多數情況下會自動復原，而不需要任何手動介入。

如果您有預留節點 (例如 DC2 預留節點)，則可以在執行調整大小時升級至 RA3 預留節點。您可以在執行彈性調整大小或使用主控台從快照進行還原時進行升級。主控台會引導您完成此程序。如需升級至 RG 或 RA3 節點類型的詳細資訊，請參閱升級至 RG 或 RA3 節點類型。

彈性調整大小不會排序資料表或回收磁碟空間，因此不是清空操作的替代方案。如需詳細資訊，請參閱清空資料表。

彈性調整大小有下列限制：

- 彈性調整大小和資料共用叢集 — 當您在作為資料共用生產者的叢集上新增或減少節點時，您無法在 Amazon Redshift 遷移叢集中繼資料時從取用者連線至該叢集。同樣地，如果您執行彈性調整大小並選擇新的節點類型，則在連線中斷並傳輸到新的目標叢集時，便無法使用資料共用。在這兩種類型的彈性調整大小中，生產者都會有幾分鐘的時間無法使用。

- 從共用快照傳輸資料 — 若要在從共用快照傳輸資料的叢集上執行彈性調整大小，必須至少有一個備份可供叢集使用。您可以在 Amazon Redshift 主控台快照清單、`describe-cluster-snapshots` CLI 命令或 `DescribeClusterSnapshots` API 操作上檢視備份。
- 平台限制 — 彈性調整大小僅供使用 EC2-VPC 平台的叢集使用。如需詳細資訊，請參閱使用 EC2 建立叢集。
- 儲存考量 — 請確定新節點的組態有足夠的儲存可儲存現有資料。您可能必須新增其他節點或變更組態。
- 來源叢集與目標叢集的大小 — 彈性調整大小可以調整大小的節點數目和節點類型，取決於來源叢集中的節點數目和針對已調整大小的叢集所選擇的節點類型。若要判斷可用的組態，您可以使用主控台。或者，您可以使用 `describe-node-configuration-options` AWS CLI 命令搭配 `action-type resize-cluster`選項。如需使用 Amazon Redshift 主控台來調整大小的相關資訊，請參閱調整叢集大小。

  下列範例 CLI 命令會描述可用的組態選項。在此範例中，名為 `mycluster` 的叢集是一個 `rg.4xl` 8 節點的叢集。

  ```
  aws redshift describe-node-configuration-options --cluster-identifier mycluster --
  region eu-west-1 --action-type resize-cluster
  ```

  此命令會傳回一個具有所建議節點類型的選項清單、節點數，以及每個選項的磁碟使用率。傳回的組態會因為特定的輸入叢集而不同。當您指定 `resize-cluster` CLI 命令的選項時，您可以選擇其中一個所傳回的組態。
- 其他節點的上限 — 彈性調整大小會限制您可以新增至叢集的節點數目。例如，RG 或 RA3 叢集支援彈性調整大小，最高可達節點數量的四倍。若要說明，您可以將節點新增至 4 節點 rg.4xl 叢集，使其成為五節點叢集，或新增更多節點，直到達到十六個節點為止。

  > **Note**
  >
  > 增加和減少的限制取決於原始節點類型和原始叢集中的節點數目或其上次的傳統調整大小。如果彈性調整大小會超過增加或減少的限制，請使用傳統調整大小。

  對於某些 rg 和 ra3 節點類型，您可以將節點數量增加到現有計數的四倍。具體來說，假設您的叢集由 ra3.4xlarge 或 ra3.16xlarge 節點組成。然後，您可以使用彈性調整大小，將 8 節點的叢集中的節點數目增加到 32 個。或者，您也可以選擇低於限制的值。(請記住，將叢集增加 4 倍的能力取決於來源叢集的大小。) 如果您的叢集具有 ra3.xlplus 節點，則限制為兩倍。

所有 rg 和 ra3 節點類型都支援將節點數量減少到現有計數的四分之一。例如，您可以將具有 ra3.4xlarge 節點的叢集大小從 12 個節點減少到 3 個，或減少到高於下限的數目。

下表列出每個支援彈性調整大小之節點類型的增加和減少限制。

原始節點類型	增加限制	減少限制
rg.12xlarge	4 倍	減少到四分之一的數目
rg.4xlarge	4 倍	減少到四分之一的數目
rg.xlarge	2 倍	減少到二分之一的數目
rg.large	2 倍	減少到二分之一的數目
ra3.16xlarge	4 倍 (例如，從 4 個節點到 16 個節點)	減少到四分之一的數目 (例如，從 16 個節點到 4 個節點)
ra3.4xlarge	4 倍	減少到四分之一的數目
ra3.xlplus	2 倍 (例如，從 4 個節點到 8 個節點)	減少到四分之一的數目
ra3.large	2 倍	減少到二分之一的數目
dc2.8xlarge	2 倍	減少到二分之一的數目 (例如，從 16 個節點到 8 個節點)
dc2.large	2 倍	減少到二分之一的數目

Note

當您調整 RG 或 RA3 叢集大小時選擇舊版節點類型 – 如果您嘗試將具有 RG 或 RA3 節點的叢集大小調整為其他節點類型，例如 DC2 ，則主控台中會顯示驗證警告訊息，而且調整大小操作不會完成。發生這種情況是因為系統不支援調整大小為舊版節點類型。這是為了防止客戶調整大小為已棄用或即將棄用的節點類型。這同時適用於彈性調整大小和傳統調整大小。

Elastic resize (傳統調整大小)

傳統調整大小會處理彈性調整大小不支援的叢集大小變更或節點類型變更的使用案例。當您執行傳統調整大小時，Amazon Redshift 會建立目標叢集，並將資料和中繼資料從來源叢集遷移到目標叢集。

將傳統大小調整為 RG 或 RA3 可以提供更好的可用性

當目標節點類型為 RG 或 RA3 時，已增強傳統調整大小。其會在來源叢集與目標叢集之間使用備份和還原操作來做到這一點。調整大小開始進行時，來源叢集會重新啟動，並有幾分鐘的時間無法使用。之後，叢集便可用於讀取和寫入操作，同時調整大小會在背景中繼續進行。

檢查叢集

若要確保您在執行傳統調整大小至 RG 或 RA3 叢集時擁有最佳效能和結果，請完成此檢查清單。當您不遵循檢查清單時，可能不會獲得使用 RG 或 RA3 節點進行傳統調整大小的一些優點，例如能夠執行讀取和寫入操作。

1. 資料的大小必須低於 2 PB。(PB 等於 1,000 TB)。若要驗證資料的大小，請建立快照並檢查其大小。您也可以執行以下查詢來檢查大小：

   ```
   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;
   ```

   只有超級使用者才能看到 `svv_table_info` 資料表。
2. 在啟動傳統調整大小之前，請確定您已建立不超過 10 小時的手動快照。如果沒有，請建立快照。
3. 用於執行傳統調整大小的快照不能用於還原資料表或其他目的。
4. 叢集必須在 VPC 中。

從傳統調整大小到 RG 或 RA3 所產生的排序和分佈操作

在傳統調整大小為 RG 或 RA3 期間，做為 EVEN 分佈遷移的 KEY 分佈資料表會轉換回原始分佈樣式。這種情況的持續時間取決於資料大小以及叢集的忙碌程度。查詢工作負載的執行優先順序高過資料遷移。如需詳細資訊，請參閱分散樣式。在此移轉過程中，對資料庫的讀取和寫入都會運作，但查詢可能需要更長的時間才能完成。不過，在這段期間，並行擴展可以透過為查詢工作負載新增資源來提升效能。您可以檢視來自 SYS_RESTORE_STATE 和 SYS_RESTORE_LOG 視觀表的結果，查看資料移轉的進度。後面有關於監控的詳細資訊。

叢集調整大小徹底完成後，會發生下列排序行為：

- 如果調整大小導致叢集有更多的配量，KEY 分佈資料表會變成部分未排序的狀態，但平均分散資料表會保持排序狀態。此外，在調整大小剛完成時，有關已排序資料數量的資訊可能不是最新的。在復原索引鍵後，自動清空功能會隨著時間的推移對資料表進行排序。
- 如果調整大小導致叢集有較少的配量，則 KEY 分佈資料表和平均分散資料表都會變成部分未排序的狀態。自動清空功能會隨著時間的推移對資料表進行排序。

如需自動清空資料的相關資訊，請參閱清空資料表。如需運算節點中配量的相關資訊，請參閱資料倉儲系統架構。

目標叢集為 RG 或 RA3 時的傳統調整大小步驟

當目標叢集類型為 RG 或 RA3，且您符合上一節詳述的先決條件時，傳統調整大小包含下列步驟。

1. 啟動從來源叢集到目標叢集的遷移。佈建新的目標叢集時，Amazon Redshift 會傳送調整大小已開始的事件通知。它會重新啟動您現有的叢集，而這會關閉所有連線。如果您現有的叢集是資料共用生產者叢集，則與取用者叢集的連線也會關閉。重新啟動會進行幾分鐘的時間。
2. 重新啟動之後，資料庫便可進行讀取和寫入。此外，資料共用也會繼續進行，這另外需要幾分鐘的時間來完成。
3. 資料會遷移至目標叢集。當目標節點類型為 RG 或 RA3 時，可在資料遷移期間進行讀取和寫入。
4. 當調整大小程序接近完成時，Amazon Redshift 會更新目標叢集的端點，而且與來源叢集的所有連線都會中斷。目標叢集會成為資料共用的生產者。
5. 調整大小完成。Amazon Redshift 傳送事件通知。

您可以在 Amazon Redshift 主控台上檢視調整大小進度。調整叢集大小所需的時間取決於資料量。

> **Note**
>
> 當您調整 RG 或 RA3 叢集大小時選擇舊版節點類型 – 如果您嘗試將具有 RG 或 RA3 節點的叢集大小調整為其他節點類型，例如 DC2 ，則主控台中會顯示驗證警告訊息，而且調整大小操作不會完成。發生這種情況是因為系統不支援調整大小為舊版節點類型。這是為了防止客戶調整大小為已棄用或即將棄用的節點類型。這同時適用於彈性調整大小和傳統調整大小。

在目標叢集為 RG 或 RA3 時監控傳統調整大小

若要監控正在進行傳統調整大小的佈建叢集 (包括 KEY 分佈)，請使用 SYS_RESTORE_STATE。其會顯示正在轉換的資料表的完成百分比。您必須是超級使用者才能存取該資料。

當您執行傳統調整大小時，請捨棄不需要的資料表。如果您這麼做，就可以更快速地分散現有資料表。

當目標叢集不是 RG 或 RA3 時，傳統調整大小步驟

例如，當目標節點類型不是 RG 或 RA3 時，例如 DC2，傳統調整大小包含下列項目。

1. 啟動從來源叢集到目標叢集的遷移。佈建新的目標叢集時，Amazon Redshift 會傳送調整大小已開始的事件通知。它會重新啟動您現有的叢集，而這會關閉所有連線。如果您現有的叢集是資料共用生產者叢集，則與取用者叢集的連線也會關閉。重新啟動會進行幾分鐘的時間。

   請注意，在傳統調整大小進行期間，系統不會保留使用 BACKUP NO 所建立的任何資料庫關聯，例如資料表或具體化視觀表。如需詳細資訊，請參閱 CREATE MATERIALIZED VIEW。
2. 重新啟動之後，資料庫會是唯讀狀態。資料共用會繼續進行，這另外需要幾分鐘的時間來完成。
3. 資料會遷移至目標叢集。資料庫會保持唯讀狀態。
4. 當調整大小程序接近完成時，Amazon Redshift 會更新目標叢集的端點，而且與來源叢集的所有連線都會中斷。目標叢集會成為資料共用的生產者。
5. 調整大小完成。Amazon Redshift 傳送事件通知。

您可以在 Amazon Redshift 主控台上檢視調整大小進度。調整叢集大小所需的時間取決於資料量。

> **Note**
>
> 當目標叢集不是 RG 或 RA3，或不符合上一節詳述的 RA3 目標叢集先決條件時，可能需要數天或數週的時間來調整具有大量資料的叢集大小。
> 另請注意，叢集所使用的儲存容量在進行傳統調整大小後可能會上升。當叢集因為傳統調整大小產生了額外的資料配量時，這是正常的系統行為。即使叢集中的節點數目保持不變，也可能會像這樣使用額外的容量。

彈性調整大小與傳統調整大小

下表會比較這兩種調整大小類型的行為。

Behavior (行為)	彈性調整大小	Elastic resize (傳統調整大小)	說明			
系統資料保留	彈性調整大小會保留系統日誌資料。	傳統調整大小不會保留系統資料表和資料。	如果您已在來源叢集中啟用稽核記錄，則可以在調整大小後繼續存取 Amazon S3 或 CloudWatch 中的日誌。您可以依照資料政策所指定的，來保留或刪除這			

Behavior (行為)	彈性調整大小	Elastic resize (傳統調整大小)	說明			
			些日誌。			
變更節點類型	當節點類型未變更時的彈性調整大小：就地調整大小，並保留大多數查詢。 選取新節點類型時的彈性調整大小：建立新叢集。當調整大小程序完成時，查詢會遭到捨棄。	傳統調整大小：建立新叢集。調整大小程序進行期間，查詢會遭到捨棄。				

Behavior (行為)	彈性調整大小	Elastic resize (傳統調整大小)	說明			
工作階段和查詢保留	當來源叢集和目標叢集中的節點類型相同時，彈性調整大小會保留工作階段和查詢。如果您選擇新的節點類型，查詢會遭到捨棄。	傳統調整大小不會保留工作階段和查詢。查詢會遭到捨棄。	當查詢遭到捨棄時，效能應該會降低。最好的辦法是在使用量不大時執行調整大小操作。			

Behavior (行為)	彈性調整大小	Elastic resize (傳統調整大小)	說明				
取消調整大小操作	您無法取消彈性調整大小。	您可以在 Amazon Redshift 主控台中，從叢集詳細資訊選擇取消調整大小，以在傳統調整大小操作完成前將其取消。	取消調整大小所需的時間量，取決於取消時調整大小操作的階段。如果您這麼做，則要等到取消操作完成後，叢集才可供使用。如果調整大小操				

Behavior (行為)	彈性調整大小	Elastic resize (傳統調整大小)	說明				
			作處於最終階段，即無法取消。 對於將傳統大小調整為 RG 或 RA3 叢集，您無法取消。				

排定調整大小

您可以為叢集排定調整大小操作，縱向擴展以應付預期的高使用量，或縮減規模以節省成本。排程同時適用於彈性調整大小和傳統調整大小。您可以在 Amazon Redshift 主控台上設定排程。如需詳細資訊，請參閱使用主控台管理叢集下的調整叢集大小。您也可以使用 AWS CLI 或 Amazon Redshift API 操作來排程調整大小。如需詳細資訊，請參閱《AWS CLI 命令參考》中的 create-scheduled-action 或《Amazon Redshift API 參考》中的 CreateScheduledAction。

快照、還原和調整大小

彈性調整大小是調整 Amazon Redshift 叢集大小最快的方法。如果彈性調整大小不是您的其中一個選擇，並且您需要對叢集進行近乎不變的寫入存取，則可以使用下節所述的快照和還原操作搭配傳統調整大小。此方法需要在擷取快照之後寫入至來源叢集的任何資料必須在切換之後手動複製到目標叢集。根據複製所需時間，您可能需要重複此動作數次，直到您在這兩個叢集中具有相同的資料。然後可以切換

至目標叢集。此程序可能對現有查詢具有負面影響，直到可在目標叢集中提供完整資料集。但它的確會將您無法寫入至資料庫的時間量降至最低。

快照、還原和傳統調整大小方法使用下列程序：

1. 擷取現有叢集的快照。現有叢集是來源叢集。
2. 記下快照所花費的時間。這麼做表示稍後您可以識別將需要在哪個時間點重新執行擷取、交易和載入 (ETL) 程序，以將任何後置快照資料載入至目標資料庫。
3. 將快照還原至新叢集。這個新叢集是目標叢集。驗證目標叢集中有範例資料。
4. 調整目標叢集大小。為目標叢集選擇新的節點類型、節點數量和其他設定。
5. 檢閱在您擷取來源叢集快照之後發生之 ETL 程序中的載入。務必依照相同順序將相同資料重新載入至目標叢集。如果您有資料持續載入，請重複此程序數次，直到來源叢集和目標叢集中的資料相同。
6. 停止來源叢集上執行的所有佇列。若要這樣做，您可以重新啟動叢集，或以超級使用者身分登入，並使用 PG_CANCEL_BACKEND 和 PG_TERMINATE_BACKEND 命令。重新啟動叢集是確定叢集無法使用的最簡便方法。
7. 重新命名來源叢集。例如，將它從 `examplecluster` 重新命名為 `examplecluster-source`。
8. 重新命名目標叢集，以使用重新命名之前的來源叢集名稱。例如，將前述的目標叢集重新命名為 `examplecluster`。從此時起，使用包含 `examplecluster` 之端點的任何應用程式都會連接至目標叢集。
9. 在切換至目標叢集之後刪除來源叢集，並驗證所有程序都如預期般運作。

或者，您可以在將資料重新載入目標叢集之前重新命名來源和目標叢集。如果您不需要任何相依系統和報告立即與目標叢集的系統和報告一樣最新，則可採用此方法。在此情況下，步驟 6 將移至上述程序尾端。

只在您想要應用程式繼續使用相同端點連接至叢集時，才需要重新命名程序。如果不需要這樣做，則您可以改為更新任何連接至叢集的應用程式，以使用目標叢集的端點，而無需重新命名叢集。

重複使用叢集名稱有一些優勢。首先，您不需要更新應用程式連線字串，因為端點不會變更，即使基礎叢集變更也一樣。其次，Amazon CloudWatch 警示和 Amazon Simple Notification Service (Amazon SNS) 通知等相關項目會繫結至叢集名稱。這個連結表示您可以繼續使用您為叢集設定的相同警示和通知。這種持續使用主要是生產環境中的疑慮，而您想要在這個環境中能夠靈活地調整群集模範，而無需重新配置相關項目，例如警示和通知。

重新命名叢集

如果想讓叢集使用不同的名稱，您可重新命名叢集。由於叢集的端點包含叢集名稱 (亦稱為叢集識別碼)，因此重新命名後端點將使用新名稱。例如，如果您擁有名為 `examplecluster` 的叢集，並將其重新命名為 `newcluster`，端點會變更為使用 `newcluster` 識別碼。連線至該叢集的任何應用程式都應更新為新端點。

如果您想變更應用程式所連線到的叢集，卻不想變更應用程式中的端點，則可以重新命名叢集。但在此情況下，您必須先重新命名原始叢集，接著變更第二個叢集，以重新使用原始叢集在重新命名之前的名稱。這樣做有其必要性，因為叢集識別碼在帳戶和區域中必須是唯一的，原始叢集和第二個叢集不得使用相同的名稱。假如您要從快照還原叢集，且不想變更任何相依應用程式的連線屬性，或許可這樣做。

> **Note**
>
> 如果要刪除原始叢集，您也必須負責刪除任何不想要的叢集快照。

重新命名叢集時，叢集狀態會變更為 `renaming`，直到流程結束。叢集所用的舊 DNS 名稱會立即刪除，但其會保留在快取內幾分鐘。重新命名的叢集的新 DNS 名稱會在大約 10 分鐘內生效。重新命名的叢集必須等到新名稱生效後才可使用。叢集將重新啟動，且任何與叢集的現有連線將會被捨棄。完成後，端點將變更為使用新名稱。因此，您應在開始重新命名之前停止執行查詢，並在重新命名完成後重新啟動查詢。

叢集快照將保留，且與叢集相關聯的所有快照在重新命名後仍會保持與該叢集的關聯。例如，假設您有一個為生產資料庫服務的叢集，且該叢集有多個快照。如果您命名叢集，接著在生產環境中將叢集換成快照，重新命名後的叢集仍有與其相關聯的現有快照。

Amazon CloudWatch 警示和 Amazon Simple Notification Service (Amazon SNS) 事件通知會與叢集名稱相關聯。如果您重新命名叢集，就必須進行對應的更新。您可以在 CloudWatch 主控台中更新 CloudWatch 警示，也可以在事件窗格上的 Amazon Redshift 主控台中更新 Amazon SNS 事件通知。叢集的負載與查詢資料會繼續顯示重新命名之前和之後的資料。但重新命名流程結束後，效能資料將重設。

如需詳細資訊，請參閱修改叢集。

升級叢集的發行版本

您可以將發行維護版本 Release Status (發行狀態) 值為 New release available (新發行版本已推出) 的叢集升級。升級維護版本時，您可以選擇立即升級或在下一次的維護時段升級。

> **Important**
>
> 如果您立即升級，則在升級完成前，您的叢集均會離線。

將叢集升級至新的發行版本

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ ：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集。
3. 選擇要升級的叢集。
4. 針對 Actions (動作)，選擇 Upgrade cluster version (升級叢集版本)。Upgrade cluster version (升級叢集版本) 頁面隨即出現。
5. 遵循頁面說明以進行操作。
6. 選擇 Upgrade cluster version (升級叢集版本)。

暫停和繼續叢集

如果您的叢集只需要在特定時間可用，則您可以暫停叢集，並於稍後再繼續該叢集。叢集暫停時，隨需計費也會暫停。只有叢集的儲存才會產生費用。如需定價的相關資訊，請參閱 Amazon Redshift 定價頁面。

當您暫停叢集時，Amazon Redshift 會建立快照、開始終止查詢，並將叢集置於暫停狀態。如果您在未請求最終快照的情況下刪除暫停的叢集，則無法還原叢集。啟動暫停或繼續操作後，您無法取消或回復該操作。

您可以使用 或 Amazon Redshift API 操作，在 Amazon Redshift AWS CLI主控台上暫停和繼續叢集。

您可以排定暫停和繼續叢集的動作。當您使用新的 Amazon Redshift 主控台建立經常性排程以暫停和繼續時，系統會針對您選擇的日期範圍建立兩個排程動作。排程動作名稱的字尾為 `-pause` 和 `-resume`。名稱的總長度必須符合排程動作名稱的大小上限。

您無法暫停下列類型的叢集：

- EC2-Classic 叢集。
- 未作用中的叢集，例如目前正在修改的叢集。

- 硬體安全模組 (HSM) 叢集。
- 已關閉自動快照的叢集。

決定暫停叢集時，請考慮下列事項：

- 無法連線或查詢叢集。
- 您無法在 Amazon Redshift 主控台上查看暫停叢集的查詢監控資訊。
- 您無法修改暫停的叢集。叢集上的任何排程動作都不會執行。其中包括建立快照、調整叢集大小，以及叢集維護操作。
- 不會建立硬體指標。如果您已在缺少的指標上設定警示，請更新您的 CloudWatch 警示。
- 您無法將暫停叢集的最新自動快照複製到手動快照。
- 當叢集暫停時，它無法繼續，直到暫停操作完成為止。
- 當您暫停叢集時，計費即會暫停。不過，暫停操作通常會在 15 分鐘內完成，視叢集的大小而定。
- 稽核日誌會進行存檔，而且繼續時不會還原這些稽核日誌。
- 叢集暫停後，就可能沒有追蹤和日誌可供用於針對暫停之前所發生的問題進行故障診斷。
- 如果您使用 管理管理員登入資料 AWS Secrets Manager 並暫停叢集，則不會刪除叢集的秘密，您將繼續支付秘密的費用。如需使用 管理 Redshift 管理員密碼的詳細資訊 AWS Secrets Manager，請參閱 使用 管理 Amazon Redshift 管理員密碼 AWS Secrets Manager。
- 在 RG 和 RA3 執行個體類型的繼續時，會還原叢集上的無備份資料表。這些資料表不會在 DC2 執行個體類型繼續時還原。如需無備份資料表的詳細資訊，請參閱 從快照排除資料表。

當您繼續叢集時，請考慮下列事項：

- 繼續叢集的叢集版本會根據叢集的維護時段更新為維護版本。
- 如果您刪除與暫停叢集相關聯的子網路，則可能會有不相容的網路。在此情況下，請從最新的快照還原叢集。
- 如果您在叢集暫停時刪除彈性 IP 地址，則會請求新的彈性 IP 地址。
- 如果 Amazon Redshift 無法使用先前的彈性網路界面繼續叢集，則 Amazon Redshift 嘗試配置新的叢集。
- 當您繼續叢集時，您的節點 IP 地址可能會變更。您可能需要更新 VPC 設定，以支援這些新的 IP 地址，適用於來自 Secure Shell (SSH) 的 COPY，或來自 Amazon EMR 的 COPY。
- 如果您嘗試繼續未暫停的叢集，則繼續操作會傳回錯誤。如果繼續操作是排程動作的一部分，請修改或刪除排程動作，以防止未來發生錯誤。

- 根據叢集的大小，可能需要幾分鐘的時間來繼續叢集，然後才能處理查詢。此外，在繼續完成之後，叢集重新化合時，查詢效能可能會有一段時間受到影響。

重新啟動叢集

重新啟動叢集是一項叢集操作，會使用重新啟動之前相同的組態重新啟動叢集。您可以重新啟動叢集，以套用待處理的維護更新、重設組態變更、從特定問題復原，或對叢集問題進行故障診斷。重新啟動叢集有助於確保 Amazon Redshift 環境的最佳效能、安全和穩定性。下列程序提供重新啟動 Amazon Redshift 叢集的詳細步驟。

當您重新啟動叢集時，叢集狀態會設為 `rebooting`，當重新啟動完成時也會建立叢集事件。任何等待中的叢集修改都會套用在這次的重新啟動上。

重新啟動叢集

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集。
3. 選擇要重新啟動的叢集。
4. 針對 Actions (動作)，選擇 Reboot cluster (重新啟動叢集)。Reboot cluster (重新啟動叢集) 頁面隨即出現。
5. 選擇 Reboot cluster (重新啟動叢集)。

重新定位叢集

透過在 Amazon Redshift 中使用重新定位功能，您可以讓 Amazon Redshift 將叢集移至另一個可用區域 (AZ)，而不會遺失任何資料或導致應用程式發生變更。透過重新定位，您可以在叢集上發生服務中斷時繼續操作，將所受影響降到最低。

開啟叢集重新定位時，Amazon Redshift 可能會在某些情況下選擇將叢集重新定位。特別是，當目前的可用區域中所發生的問題讓您無法達到最佳化的叢集操作，或是為了改善服務可用性，就會發生這種情況。如果指定可用區域中的資源限制會中斷叢集操作，您也可以叫用重新定位功能。其中一個範例是繼續執行叢集或調整叢集大小的能力。Amazon Redshift 提供重新定位功能，無需額外付費。

將 Amazon Redshift 叢集重新定位到新的可用區域時，新叢集的端點會與原始叢集的端點相同。您的應用程式可以重新連線到該端點並繼續操作，而不會修改或遺失資料。但是，由於指定可用區域中的資源可能有限制，因此不一定可以重新定位。

RG 和 RA3 執行個體類型支援 Amazon Redshift 叢集重新定位。RG 和 RA3 執行個體類型使用 Redshift 受管儲存 (RMS) 作為耐用的儲存層。叢集資料的最新複本一律可在 AWS 區域中的其他可用區域中使用。換句話說，您可以將 Amazon Redshift 叢集重新定位到另一個可用區域，而不會遺失任何資料。

當您開啟叢集的重新定位功能時，Amazon Redshift 會將您的叢集遷移到代理後面。這樣做有助於實作與位置無關的叢集運算資源存取權。遷移會導致叢集重新開機。將叢集重新定位到另一個可用區域時，當新叢集在新的可用區域中重新上線時，會發生中斷情形。不過，您不需要對應用程式進行任何變更，因為即使在叢集重新定位至新的可用區域後，叢集端點仍會維持不變。

在子網路群組包含多個可用區域的新建立或還原的 RG 或 RA3 叢集上，預設會啟用叢集重新定位。在建立佈建叢集時，Amazon Redshift 會指派 5439 作為預設連接埠。您可以變更為 5431 至 5455 或 8191 至 8215 之連接埠範圍內的其他連接埠。(請勿變更為範圍以外的連接埠。這會導致錯誤。) 若要變更佈建叢集的預設連接埠，請使用 Amazon Redshift 主控台 AWS CLI或 Amazon Redshift API。若要變更無伺服器工作群組的預設連接埠，請使用 AWS CLI 或 Amazon Redshift Serverless API。

如果您開啟重新定位，且目前使用領導節點 IP 地址來存取叢集或增強型 VPC 路由，請務必變更該存取。請改為使用與叢集的虛擬私有雲端 (VPC) 端點相關聯的 IP 地址。若要尋找此叢集的 IP 地址，請在叢集詳細資訊頁面的網路和安全區段中尋找並使用 VPC 端點。若要取得 VPC 端點的詳細資訊，請登入 Amazon VPC 主控台。

您也可以使用 AWS Command Line Interface (AWS CLI) 命令`describe-vpc-endpoints`來取得與端點相關聯的彈性網路界面。您可以使用 `describe-network-interfaces` 命令來取得相關聯的 IP 地址。如需 Amazon Redshift AWS CLI 命令的詳細資訊，請參閱《 命令參考》中的可用AWS CLI 命令。

限制

使用 Amazon Redshift 重新定位功能時，請注意下列限制：

- 由於指定可用區域中的資源可能有限制，因此並非所有情況下都能進行叢集重新定位。如果發生這種情況，Amazon Redshift 不會變更原始叢集。
- DC2 執行個體系列產品不支援重新定位。
- 您無法跨 AWS 區域執行重新定位。
- Amazon Redshift 重新定位預設使用連接埠號碼 5439。您也可以變更為 5431-5455 或 8191-8215 之連接埠範圍內的其他連接埠。

使用主控台管理重新定位

您可以使用 Amazon Redshift 主控台來管理叢集重新定位的設定。

建立新叢集時關閉重新定位

使用下列程序在建立新叢集時關閉重新定位功能。

關閉新叢集的重新定位功能

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集。
3. 選擇建立叢集以建立新叢集。如需如何建立叢集的詳細資訊，請參閱《Amazon Redshift 入門指南》中的開始使用 Amazon Redshift 佈建資料倉儲。
4. 在備份下，對於叢集重新定位選擇已停用。重新定位預設為開啟。
5. 選擇 Create Cluster (建立叢集)。

修改現有叢集的重新定位

使用下列程序變更現有叢集的重新定位設定。

修改現有叢集的重新定位設定

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽選單上，選擇叢集。系統會 AWS 列出目前區域中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。
3. 從清單中選擇您要修改之叢集的名稱。隨即會出現叢集詳細資訊頁面。
4. 選擇維護索引標籤，然後在備份詳細資訊區段中選擇編輯。
5. 在備份下，選擇已停用。重新定位預設為開啟。
6. 選擇修改叢集。

重新定位叢集

使用下列程序將叢集手動重新定位到其他可用區域。當您要測試次要可用區域中的網路設定，或當您在目前的可用區域中即將遇到資源限制時，此功能特別有用。

將叢集重新定位到其他可用區域

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽選單上，選擇叢集。系統會 AWS 列出目前區域中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。
3. 從清單中選擇您要移動之叢集的名稱。隨即會出現叢集詳細資訊頁面。
4. 針對動作，選擇重新定位。重新定位叢集頁面隨即出現。
5. (選擇性) 選擇可用區域。如果您未選擇可用區域，Amazon Redshift 會為您選擇。

Amazon Redshift 會啟動重新定位功能，並將叢集顯示為正在重新定位。重新定位完成後，叢集狀態會變更為「可用」。

使用 Amazon Redshift CLI 管理重新定位

您可以使用 AWS 命令列介面 (CLI) 管理叢集重新定位的設定。

使用 AWS CLI，下列範例命令會建立名為 **mycluster**且已開啟重新定位的 Amazon Redshift 叢集。

```
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
```

如果您目前的叢集使用不同的連接埠，您必須先將其修改為使用 5431-5455 或 8191-8215 的連接埠範圍，然後再將其修改為開啟重新定位。預設值為 5439。下列範例命令會修改連接埠，以防您的叢集未使用指定範圍中的連接埠。

```
aws redshift modify-cluster --cluster-identifier mycluster --port 5439
```

下列範例命令包含 Amazon Redshift 叢集上的 availability-zone-relocation 參數。

```
aws redshift modify-cluster --cluster-identifier mycluster --availability-zone-
relocation
```

下列範例命令會關閉 Amazon Redshift 叢集上的 availability-zone-relocation 參數。

```
aws redshift modify-cluster --cluster-identifier mycluster --no-availability-zone-
relocation
```

下列範例命令會叫用 Amazon Redshift 叢集上的重新定位。

```
aws redshift modify-cluster --cluster-identifier mycluster --availability-zone us-
east-1b
```

在叢集上設定用量限制

您可以新增最多四個用量限制，以控制下列各項的用量：

- 並行擴展
- 使用額外運算資源執行自動最佳化
- Redshift Spectrum 用量
- 跨區域資料共用

設定佈建叢集的用量限制

以下是在佈建叢集上設定用量限制的程序：

設定叢集的用量限制

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 導覽至您要為其設定限制的佈建叢集。
3. 從叢集的詳細資訊頁面，從動作下拉式功能表中選取管理用量限制。您也可以選取叢集的維護索引標籤，然後向下捲動並選取建立用量限制。
4. 選取您要設定的用量限制的新增限制。您可以為指定功能新增最多 4 個限制。
5. 設定用量限制的期間，即每日、每週或每月。
6. 設定用量限制。
   - 對於使用額外運算資源限制執行的並行擴展和自動最佳化，用量限制是 Amazon Redshift 在指定期間內使用 功能所花費的時間量。在這種情況下，用量限制是以小時和分鐘為單位設定。
   - 對於 Redshift Spectrum，用量限制是從 Amazon S3 掃描的資料量。在此情況下，用量限制以 TB (TB) 為單位設定。
   - 對於跨區域資料共用，用量限制是從生產者區域傳輸至消費者區域的資料量，供消費者查詢。在此情況下，用量限制以 TB (TB) 為單位設定。

7. 將 Amazon Redshift 的動作設定為在叢集達到限制時採取。這些項目如下：

   - 記錄到系統資料表 - 將記錄新增到系統檢視 SYS_QUERY_HISTORY。您可以在此檢視中查詢 usage_limit 欄，以判斷查詢是否超過限制。
   - 提醒 - 使用 Amazon SNS 設定通知訂閱，並在超過限制時傳送通知。您可以選擇現有的 Amazon SNS 主題、建立新主題，或不使用主題繼續。
   - 停用功能 - 停用功能。您也可以選擇使用 Amazon SNS 傳送通知。使用者可以繼續將叢集用於其他任務。

   前兩個動作是資訊性的，但上次停用此功能。
8. 在頁面底部選擇儲存變更以儲存您限制。如果您一次設定多個限制，儲存變更會一次儲存所有限制。

關閉及刪除叢集

如果想讓叢集停止執行且不再產生費用，可關閉叢集。關閉叢集時，您可選擇性建立最後快照。若您建立最後快照，Amazon Redshift 會在叢集關閉前手動建立叢集的快照。若打算使用您要刪除的相同資料和組態來佈建新的叢集，您需要手動快照。您在稍後可以使用手動快照來還原快照，並繼續使用該叢集。

如果您不再需要叢集和其資料，則不用建立最後快照，然後關閉叢集。在此情況下，叢集和資料將被永久刪除。

無論關閉叢集時是否手動建立最後快照，所有與叢集相關聯的自動化快照都會在叢集關閉後被刪除。但與叢集相關聯的任何手動快照則會保留。如果您在關閉叢集時已無其他執行中的叢集，或如果您超過提供給執行中 Amazon Redshift 叢集的免費可用儲存空間，任何保留的手動快照 (包括選擇性的最後快照) 將依 Amazon Simple Storage Service 儲存費率計費。如需快照儲存費用的相關資訊，請參閱 Amazon Redshift 定價頁面。

刪除叢集也會刪除任何相關聯的 AWS Secrets Manager 秘密。

刪除叢集

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集。
3. 選擇要刪除的叢集。

4. 對於 Actions (動作)，請選擇 Delete (刪除)。Delete cluster (刪除叢集) 頁面隨即出現。
5. 選擇 Delete cluster (刪除叢集)。

> **Note**
>
> 當您刪除叢集並選擇建立最終快照時，如果叢集上正在進行還原操作，則 Amazon Redshift 將停止刪除請求。如果發生這種情況，您可以在沒有最終快照的情況下刪除叢集，也可以在還原完成後，將有最終快照的叢集刪除。

Amazon Redshift 快照和備份

快照是叢集的時間點備份。有兩種類型的快照：自動和手動。Amazon Redshift 會透過使用加密的 Secure Sockets Layer (SSL) 連線，將這些快照儲存在 Amazon S3 內部。

Amazon Redshift 會自動取得增量快照，以追蹤自上一個自動快照以來對叢集的變更。自動快照會保留所有需要的資料以從快照還原叢集。您可以建立快照排程以控制何時進行自動快照，或隨時手動進行快照。

當您從快照進行還原時，Amazon Redshift 會建立一個新的叢集，並在所有資料載入之前讓新叢集可供使用，以便您立即查詢新叢集。叢集會隨需從快照中串流資料以回應活動查詢，然後在背景載入剩餘資料。

當您啟動叢集時，您可以為自動和手動快照設定保留期間。您可以經由修改叢集，以變更自動和手動快照的預設保留期間。您可以在建立快照時，或經由修改快照，以變更手動快照的保留期間。

您可以在 中檢視快照詳細資訊 AWS 管理主控台，或在 CLI 或 DescribeClusterSnapshots API 動作中呼叫 describe-cluster-snapshots，以監控快照的進度。有關正在進行的快照，這些資訊顯示了例如增量快照的大小、傳輸速率、經過時間和預估剩餘時間。

為了確保您的備份永遠可供叢集使用，Amazon Redshift 會將快照儲存在由 Amazon Redshift 管理的內部受管 Amazon S3 儲存貯體中。若要管理儲存費用，請評估您需要保留自動快照的天數，並相應地設定其保留期。刪除您不再需要的任何手動快照。如需備份儲存成本的相關資訊，請參閱 Amazon Redshift 定價頁面。

您也可以使用 建立和還原快照 AWS Backup，這項全受管服務可協助您集中和自動化跨 AWS 服務、雲端和內部部署的資料保護。如需詳細資訊，請參閱AWS Backup 與 Amazon Redshift 整合。如需詳細資訊 AWS Backup，請參閱什麼是 AWS Backup？ 《AWS Backup 開發人員指南》中的 。

在 Amazon Redshift Serverless 中使用快照和備份

Amazon Redshift Serverless 如同佈建叢集，可讓您將備份作為命名空間中物件和資料的時間點表示。Amazon Redshift Serverless 中的備份類型有兩種：手動建立的快照，以及 Amazon Redshift Serverless 自動建立的復原點。您可以在快照和復原點中找到有關使用 Amazon Redshift Serverless 快照的詳細資訊。

您也可以從佈建叢集將快照還原至無伺服器命名空間。如需詳細資訊，請參閱從快照還原無伺服器命名空間。

自動快照

當叢集的自動快照已啟用時，Amazon Redshift 將定期為該叢集建立快照。Amazon Redshift 預設約每 8 小時或每節點資料變更 5 GB 時拍攝一次，以先到者為準。如果您的資料大於 5 GB * 節點數，則兩次自動建立快照之間的最短間隔時間為 15 分鐘。或者，您可以建立快照排程以控制何時進行自動快照。如果您使用自訂排程，則兩次自動快照之間的最短間隔時間為一小時。依據預設，當您建立叢集時，將會啟用自動快照。

> **Note**
>
> Amazon Redshift 會擷取暫停資料共用生產者叢集的自動快照。由於消費者可以寫入暫停的生產者，這些快照會擷取這些資料共用寫入並提供復原點。先前，Amazon Redshift 在暫停時不會拍攝這些叢集的自動快照。這些快照遵循叢集現有的自動化快照排程和保留期，而且不需要採取任何動作。對於非資料共用生產者的暫停叢集，Amazon Redshift 不會在叢集暫停時拍攝自動快照。

在保留期間結束後，自動快照將永久刪除。預設保留期間為一日，但您可以使用 Amazon Redshift 主控台或使用 Amazon Redshift API 或 CLI 以程式設計方式進行修改。

若要停用自動快照，請將保留期間設定為 zero (零)。若您停用了自動快照，Amazon Redshift 將停止拍攝快照並刪除叢集的任何現有自動快照。您無法停用 RG 或 RA3 節點類型的自動快照。您可以將 RG 或 RA3 節點類型自動保留期設定為 1–35 天。

只有 Amazon Redshift 可以刪除自動快照；您無法手動刪除自動快照。在自動快照的保留期結束時、停用叢集的自動快照時，或當您刪除叢集時，Amazon Redshift 會刪除自動快照。Amazon Redshift 會保留最新的自動快照，直到您停用自動快照或刪除叢集為止。

如果想要讓自動快照保留期間更長，請建立該快照複本以做為手動快照。自動快照會保留到保留期間結束為止，但對應的手動快照會保留到您手動刪除它為止，或直到保留期間結束為止。

自動快照排程

若要精確控制何時進行快照，您可以建立快照排程，並將它連接至一或多個叢集。當您修改快照排程時，所有相關聯叢集的排程都會修改。如果叢集沒有連接快照排程，將會使用預設的自動快照排程。

快照排程是一組排程規則。您可以指定時間間隔來定義簡單的排程規則，例如每 8 小時或每 12 小時。您也可以新增規則，在特定的週中的日、指定的時間或特定期間進行快照。您也可以使用類似 Unix 的 cron 運算式來定義規則。

快照排程格式

您可以在 Amazon Redshift 主控台建立快照排程。然後，您可以將排程連接至叢集以觸發系統快照的建立。一個排程可連接至多個叢集，而且您可以在一個排程中建立多個 cron 定義來觸發快照。

您可以使用 cron 語法為您的快照定義排程。這些排程的定義使用經過修改且類似 Unix 的 cron 語法。您以國際標準時間 (UTC) 指定時間。您可以用最大頻率 1 小時以及最小單位 1 分鐘來建立排程。

Amazon Redshift 修改的 cron 運算式有 3 個必要欄位，並且會以空格隔開。

語法

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

欄位	Values (數值)	萬用字元
分鐘	0–59	, - * /
小時	0–23	, - * /
月中的日	1–31	, - * ? / L W
月	1-12 或 JAN-DEC	, - * /
週中的日	1-7 或 SUN-SAT	, - * ? L #
年	1970-2199	, - * /

萬用字元

- , (逗號) 萬用字元包含額外的值。在 `Day-of-week` 欄位，`MON,WED,FRI` 包括週一、週二和週三。每個欄位的總數值限制為 24。

- - (破折號) 萬用字元用於指定範圍。在 Hour 欄位中，1–15 會包含指定日的 1 至 15 時。
- * (星號) 包含欄位中所有的值。在 Hours 欄位，* 包含每個小時。
- / (斜線) 萬用字元用於指定增量。在 Hours 欄位，您可以輸入 **1/10** 指定每第 10 小時，從一天的第一小時開始 (例如 01:00、11:00 和 21:00)。
- ? (問號) 萬用字元用於表示不限定任何一個。在 Day-of-month 欄位，您可以輸入 7，如果您不在意這個月的 7 號是星期幾，就可以在 Day-of-week (週中的日) 欄位中輸入 ?。
- L 萬用字元在 Day-of-month 或 Day-of-week 欄位可指定月份或週的最後一天。
- W 萬用字元在 Day-of-month 欄位可指定任務日。在 Day-of-month 欄位，3W 指定的是月份中最接近第三個任務日的日子。
- # 萬用字元在 Day-of-week (週中的日) 欄位可指定某個月中某週特定日子的特定執行個體。例如，3#2 代表則該月的第二個星期二：3 是指星期二，因為它是每週的第三天，2 指的是一個月內該類型的第二天。

> **Note**
>
> 如果您使用 '#' 字元，則只能在星期幾欄位中定義一個表達式。例如："3#1,6#3" 是無效的，因為它被轉譯為兩個表達式。

限制

- 您無法在同一個 cron 表達式中指定 Day-of-month 和 Day-of-week 欄位。如果您在其中一個欄位指定了數值，就必須在另一個欄位中使用 ? (問號)。
- 快照排程不支援下列頻率：
  - 快照排程頻率超過每小時 1 次。
  - 快照排程頻率低於每天 (24 小時) 1 次。

  如果您有重疊的排程，導致 1 個小時內有多個排程快照，將會產生驗證錯誤。

建立排程時，您可以使用下列 cron 字串範例。

分鐘	小時	週中的日	意義			
0	14-20/1	TUE	週二下午 2 點至 8 點，每小時一次。			

分鐘	小時	週中的日	意義			
0	21	MON-FRI	週一至週五每天晚上 9 點。			
30	0/6	SAT-SUN	週六和週日每 6 小時增量，從當日午夜後 30 分鐘開始 (00:30)。結果是在每天 [00:30、06:30、12:30 和 18:30] 進行快照。			
30	12/4	*	每天 12:30 開始每 4 小時增量。這解析為 [12:30、16:30、20:30]。			

舉例來說，如果要執行排程，在每天的 15:15 開始每 2 小時增量。這解析為 [15:15、17:15、19:15、21:15、23:15]，指定：

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

您可以在排程中建立多個 cron 排程定義。例如，下列 AWS CLI 命令在一個排程中包含兩個 Cron 排程。

```
create-snapshot-schedule --schedule-identifier "my-test" --schedule-definition "cron(0
 17 SAT,SUN)" "cron(0 9,17 MON-FRI)"
```

手動快照

您可以隨時建立手動快照。在預設情況下，即使刪除叢集，手動快照也會無限期地保留。您可以在建立手動快照時指定保留期間，或經由修改快照以變更保留期間。如需變更保留期的相關資訊，請參閱修改手動快照保留期。

若刪除快照，您便無法開始任何參考該快照的新操作。不過，若一個還原操作正在處理中，該還原操作將會繼續執行至完成為止。

Amazon Redshift 具有配額，可限制您可以建立的手動快照總數；此配額是每個 AWS 區域每個 AWS 帳戶。Amazon Redshift 中的配額和限制中會列出預設配額。

快照儲存

由於快照會產生儲存費用，當您不再需要快照時，務必刪除它們。Amazon Redshift 會在各自的快照保留期結束時刪除自動和手動快照。您也可以使用 AWS 管理主控台 或 batch-delete-cluster-snapshots CLI 命令刪除手動快照。

您可以修改手動快照設定，以變更手動快照的保留期間。

您可以使用 Amazon Redshift 主控台或使用 describe-storage CLI 命令，以取得您的快照耗用多少儲存的相關資訊。

從快照排除資料表

依預設，所有使用者定義的永久資料表都包含在快照中。如果不需要備份資料表 (例如臨時資料表)，則可以大幅減少建立快照和從快照還原所需的時間。您也可以使用不備份資料表以在 Amazon S3 上減少儲存空間。若要建立不備份資料表，請在建立資料表時使用 BACKUP NO 參數。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE TABLE 和 CREATE TABLE AS。

> **Note**
>
> RG 或 RA3 佈建叢集和 Amazon Redshift Serverless 工作群組不支援無備份資料表。在 RG 或 RA3 叢集或無伺服器工作群組中標記為無備份的資料表會被視為永久資料表，在拍攝快照時一律會備份，並在從快照還原時一律還原。為了避免無備份資料表的快照成本，請在拍攝快照之前截斷這些資料表。

建立手動快照

您可從快照清單手動建立叢集的快照，方式如下。或者，您也可從叢集組態窗格建立叢集快照。如需詳細資訊，請參閱Amazon Redshift 快照和備份。

> **Note**
>
> RG 或 RA3 佈建叢集和 Amazon Redshift Serverless 工作群組不支援無備份資料表。在 RG 或 RA3 叢集或無伺服器工作群組中標記為無備份的資料表會被視為永久資料表，在拍攝快照時一

律會備份，並在從快照還原時一律還原。為了避免無備份資料表的快照成本，請在拍攝快照之前截斷這些資料表。

建立手動快照

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表中，選擇叢集、快照，然後選擇建立快照。建立手動快照的快照頁面即會出現。
3. 輸入快照定義屬性，然後選擇建立快照。可能要等一會兒才能使用此快照。

建立快照排程

Amazon Redshift 會定期對資料建立自動增量快照，並將快照儲存至 Amazon S3。此外，您隨時都可以手動建立資料快照。

Amazon Redshift 主控台所有的快照作業都從快照清單中啟動。您可使用時間範圍、快照類型，以及與快照相關聯的叢集來篩選快照清單。此外，您可以依日期、大小和快照類型排序清單。快照上可用的選項各不相同，視所選的快照類型而定。

若要精確控制何時進行快照，您可以建立快照排程，並將它連接至一或多個叢集。您可以在建立叢集時附加排程，或修改叢集來附加排程。如需詳細資訊，請參閱自動快照排程。

建立快照排程

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表中，選擇叢集、快照，然後選擇快照排程索引標籤。快照排程即會出現。
3. 選擇 Add schedule (新增排程) 以顯示新增排程的頁面。
4. 輸入排程定義的屬性，然後選擇 Add schedule (新增排程)。
5. 在出現的頁面中，您可將叢集連接至新的快照排程，然後選擇 OK (確定)。

共用快照

您可以透過授權快照的存取，與其他 AWS 客戶帳戶共用現有的手動快照。每個快照最多可授權 20 個，每個 AWS Key Management Service (AWS KMS) 金鑰最多可授權 100 個。也就是說，如果您有

10 個使用單一 KMS 金鑰加密的快照，則可以授權 10 個 AWS 帳戶還原每個快照，或其他組合，這些組合最多可新增 100 個帳戶，且每個快照不超過 20 個帳戶。在其中一個已授權帳戶登入為使用者的人員，接下來可以描述快照或將其還原，以在其帳戶下建立新的 Amazon Redshift 叢集。例如，如果您使用個別 AWS 的客戶帳戶進行生產和測試，使用者可以使用生產帳戶登入，並與測試帳戶中的使用者共用快照。某人若以測試帳戶使用者身分登入，他接下來可以還原快照以建立由測試帳戶所擁有的新叢集，以進行測試或診斷工作。

手動快照由建立快照 AWS 的客戶帳戶永久擁有。僅有在帳戶中擁有快照的使用者，方可授權或撤銷其他帳戶存取該快照。在已授權帳戶中的使用者僅可描述或還原任何與他們分享的快照；他們不得複製或刪除與他們分享的快照。授權將持續生效，除非快照擁有者撤銷之。若授權被撤銷，之前授權的使用者將無法看見該快照，也不能啟動任何參考該快照的新動作。若存取權被撤銷時帳戶正在還原該快照，則該還原會持續進執行至完成。您不得刪除有主動授權的快照；您必須先撤銷所有的授權。

AWS 客戶帳戶一律有權存取帳戶擁有的快照。試圖授權或撤銷擁有者帳戶的存取權將會收到錯誤。您無法還原或描述非作用中 AWS 客戶帳戶擁有的快照。

在您授權存取 AWS 客戶帳戶之後，該帳戶中的任何使用者都無法在快照上執行任何動作，除非他們擔任具有允許他們這樣做之政策的角色。

- 只有當快照擁有者帳戶中的使用者所擔任的角色有 IAM 政策允許他們執行使用包含快照的資源規格動作，他們才可授權或撤銷快照的存取權。例如，下列政策允許 AWS 帳戶中的使用者或角色012345678912授權其他帳戶存取名為 的快照my-snapshot20130829：

  JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
          "redshift:AuthorizeSnapshotAccess",
          "redshift:RevokeSnapshotAccess"
          ],
      "Resource":[
           "arn:aws:redshift:us-east-1:012345678912:snapshot:*/my-
snapshot20130829"
          ]
    }
  ]
}
```

- 已共用快照 AWS 之帳戶中的使用者無法對該快照執行動作，除非他們具有允許這些動作的許可。您可以透過將政策指派給某個角色並擔任該角色來執行此操作。
  - 列出或描述快照；他們必須擁有允許 DescribeClusterSnapshots 動作的 IAM 政策。下列代碼顯示了範例：

    JSON

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

- 若要還原快照，使用者所擔任的角色必須擁有允許 RestoreFromClusterSnapshot 動作的 IAM 政策，並擁有涵蓋嘗試建立的叢集與快照的資源元素。例如，若帳戶 012345678912 中的使用者已經與帳戶 my-snapshot20130829 分享快照 219876543210，以透過還原快照來建立叢集，則帳戶 219876543210 中的使用者所擔任的角色必須擁有的政策如下所示：

  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"
```

```
            ]
        }
    ]
}
```

- 從 AWS 帳戶撤銷快照的存取權後，該帳戶中的任何使用者都無法存取快照。即使這些帳戶有 IAM 政策允許對之前共用的快照資源執行動作也一樣。

使用主控台共用叢集快照

在主控台上，您可授權其他使用者存取您所擁有的手動快照，之後不需要時也可撤銷存取權。

與其他帳戶共用快照

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集、快照，然後選擇要共用的手動快照。
3. 針對 Actions (動作)，選擇 Manual snapshot settings (手動快照設定) 以顯示手動快照的屬性。
4. 在 Manage access (管理存取) 區段輸入要共用的一或多個帳戶，然後選擇 Save (儲存)。

共用加密快照時的安全考量

當您提供加密快照的存取權時，Redshift 會要求將用來建立快照的 AWS KMS 客戶受管金鑰共用給執行還原的一或多個帳戶。如果金鑰未共用，嘗試還原快照會導致存取遭拒錯誤。接收帳戶不需要任何額外權限即可還原共用快照。當您授權快照存取權並共用金鑰時，授權存取權的身分必須具有用來加密快照之金鑰的 `kms:DescribeKey` 許可。AWS KMS 權限中有此許可的更詳細描述。如需詳細資訊，請參閱 Amazon Redshift AI 參考文件中的 DescribeKey。

客戶受管金鑰政策可透過程式設計方式或在 AWS Key Management Service 主控台上更新。

> **Note**
>
> 如果您使用預設 KMS 金鑰，則不需要在 AWS KMS 中執行動作或變更任何內容，即可共用快照。

允許存取加密快照的 AWS KMS 金鑰

若要共用加密快照的 AWS KMS 客戶受管金鑰，請執行下列步驟來更新金鑰政策：

1. 將 KMS 金鑰政策更新為您在 KMS 金鑰政策中，當作 `Principal` 共用之 AWS 帳戶的 Amazon Resource Name (ARN)。
2. 允許 `kms:Decrypt` 動作。

在以下金鑰政策範例中，使用者 111122223333 是 KMS 金鑰的擁有者，使用者 444455556666 是與其共用該金鑰的帳戶。此金鑰政策可讓 AWS 帳戶存取範例 KMS 金鑰，方法是將使用者的根 AWS 帳戶身分 ARN 444455556666做為政策`Principal`的 ，以及允許 `kms:Decrypt`動作。

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": "*"
        }
    ]
}
```

將存取權授予客戶受管 KMS 金鑰後，還原加密快照的帳戶必須建立 AWS Identity and Access Management (IAM) 角色，如果尚未擁有使用者，則必須建立該角色或使用者。此外，該 AWS 帳戶也必須將 IAM 政策連接至該 IAM 角色或使用者，以允許他們使用您的 KMS 金鑰還原加密的資料庫快照。

如需授予 AWS KMS 金鑰存取權的詳細資訊，請參閱《 開發人員指南》中的允許其他帳戶中的使用者使用 KMS 金鑰。 AWS Key Management Service

如需金鑰政策的概觀，請參閱 Amazon Redshift 如何使用 AWS KMS。

複製自動化快照

當快照的保留期間結束，當您停用自動快照，或當您刪除叢集時，系統會自動刪除自動快照。如果想要保留自動快照，可將其複製到手動快照。

複製自動快照

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集、快照，然後選擇要複製的快照。
3. 針對 Actions (動作)，選擇 Copy automated snapshot (複製自動快照) 以複製快照。
4. 更新新快照的屬性，然後選擇 Copy (複製)。

將快照複製到另一個 AWS 區域

您可以設定 Amazon Redshift 自動將叢集的快照 （ 自動或手動 ） 複製到另一個 AWS 區域。在叢集的主要 AWS 區域中建立快照時，快照會複製到次要 AWS 區域。這兩個 AWS 區域分別稱為來源 AWS 區域和目的地 AWS 區域。如果您在另一個 AWS 區域中存放快照的副本，如果有任何影響主要 AWS 區域，您可以從最近的資料還原叢集。您可以設定叢集一次只將快照複製到一個目的地 AWS 區域。如需 Amazon Redshift 區域的清單，請參閱《Amazon Web Services 一般參考》中的區域與端點。

當您啟用 Amazon Redshift 自動將快照複製到另一個 AWS 區域時，您可以指定要複製快照的目的地 AWS 區域。對於自動快照，您也可以指定保留期間，以將其保留在目的地 AWS 區域中。自動快照複製到目的地 AWS 區域並到達該區域的保留期間後，就會從目的地 AWS 區域刪除該快照。執行此動作讓您的快照保持低使用率。若要在目的地 AWS 區域中保留較短或較長的自動快照，請變更此保留期間。

您為複製到目的地 AWS 區域的自動快照設定的保留期與來源區域中自動快照的保留期不同 AWS 。預設快照複本的保留期間為 7 天。該 7 日的保留期間僅適用於自動快照。在來源和目的地 AWS 區域中，手動快照會在快照保留期間結束時或您手動刪除快照時刪除。

您可隨時停用叢集的自動快照複製。當您停用此功能時，快照不會再從來源 AWS 區域複製到目的地 AWS 區域。任何複製到目的地 AWS 區域的自動快照都會在達到保留期間限制時刪除，除非您手動建立快照複本。這些手動快照以及從目的地 AWS 區域複製的任何手動快照都會保留在目的地 AWS 區域中，直到您手動刪除為止。

若要變更您複製快照的目標 AWS 區域，請先停用自動複製功能。然後，重新啟用它，同時指定新的目的地 AWS 區域。

將快照複製到目的地 AWS 區域之後，快照會變成作用中，並可用於還原目的。

若要將 AWS KMS加密叢集的快照複製到另一個 AWS 區域，請建立授予 Amazon Redshift 在目的地區域中使用客戶受管金鑰 AWS 。然後在來源 AWS 區域中啟用快照複製時，選擇該授予。如需設定跨區域快照複本授權的相關資訊，請參閱 將 AWS KMS加密的快照複製到另一個快照 AWS 區域。

從快照還原叢集

快照包含的資料來自正在您叢集上執行的任何資料庫。它也包含叢集的相關資訊，包括節點數、節點類型及管理員使用者名稱。如果您從照還原叢集，則 Amazon Redshift 會使用叢集資訊來建立新的叢集。然後，它會從快照資料還原所有資料庫。

> **Note**
>
> RG 或 RA3 佈建叢集和 Amazon Redshift Serverless 工作群組不支援無備份資料表。在 RG 或 RA3 叢集或無伺服器工作群組中標記為無備份的資料表會被視為永久資料表，在拍攝快照時一律會備份，並在從快照還原時一律還原。

若為從原始快照建立的新叢集，您可以選擇組態，例如節點類型及節點數。除非您在請求中指定另一個可用區域，否則該叢集將在同一個 AWS 區域和系統隨機選擇的可用區域中還原。從快照還原叢集時，您可以選擇性地為新叢集選擇相容的維護軌道。

> **Note**
>
> 搭配不同組態將快照還原至叢集時，必須已在版本為 1.0.10013 或更新版本的叢集上取得快照。

當還原正在進行時，系統通常會以下列順序發出事件：

1. 還原程序開始時，會傳送 RESTORE_STARTED – REDSHIFT-EVENT-2008。
2. 建立了新叢集時，會傳送 RESTORE_SUCCEEDED – REDSHIFT-EVENT-3003。

   叢集可供查詢使用。
3. 當資料傳輸完成時，會傳送 DATA_TRANSFER_COMPLETED – REDSHIFT-EVENT-3537。

> **Note**
>
> RG 和 RA3 叢集只會發出 RESTORE_STARTED 和 RESTORE_SUCCEEDED 事件。RESTORE 成功後就不會進行明確的資料傳輸，因為 RG 和 RA3 節點類型會將資料存放在 Amazon Redshift 受管儲存體中。使用 RG 和 RA3 節點時，資料會在 RG 和 RA3 節點與 Amazon Redshift 受管儲存之間持續傳輸，做為正常查詢處理的一部分。RG 和 RA3 節點會在本機快取熱資料，並在 Amazon Redshift 受管儲存體中自動保留較不頻繁的查詢區塊。

您可以呼叫 DescribeClusters API 操作，或在 AWS 管理主控台檢視叢集詳細資訊，以監控還原進度。有關正在進行的還原，這些資訊顯示了例如快照資料的大小、傳輸速率、經過時間和預估剩餘時間。如需這些指標的說明，請參閱 RestoreStatus。

您不得使用快照將作用中的叢集還原至上一個狀態。

> **Note**
>
> 當您還原快照至新叢集時，除非您指定了不同的值，否則將會使用預設安全群組與參數群組。

基於下列原因，您可能想要搭配不同組態將快照還原至叢集：

- 當叢集是由更小的節點類型組成，而且您想要將其合併成更大的節點類型，而節點更少時。
- 當您已監控工作負載並判定需要移至 CPU 及儲存空間更多的節點類型時。
- 當您想要利用不同節點類型來測量測試工作負載的效能時。

「還原」有下列限制：

- 新節點組態必須具有足夠儲存體可放置現有資料。即使在新增節點時，您的新組態可能因為資料的重新配送方式而沒有足夠儲存體。
- 還原操作會檢查快照是否建立在與新叢集的叢集版本相容的叢集版本上。如果新叢集的版本層級太早，則還原操作會失敗，並且會在錯誤訊息中報告詳細資訊。
- 您可以還原的可能組態 (節點數目和節點類型) 取決於原始叢集中的節點數目，以及新叢集的目標節點類型。若要判斷可用的組態，您可以使用 Amazon Redshift 主控台或 `describe-node-configuration-options` AWS CLI 命令搭配 `action-type restore-cluster`。如需使用 Amazon Redshift 主控台還原的相關資訊，請參閱從快照還原叢集。

下列步驟會使用 AWS CLI取得具有許多節點的叢集，並將其合併為更大的節點類型，而節點數更小。在此範例中，我們從 24 個 節點構成的來源叢集開始。在此情況下，假設我們已建立此叢集的快照，而且想要將其還原至更大的節點類型。

1. 執行下列命令來取得我們 24 節點 叢集的詳細資訊。

```
aws redshift describe-clusters --region eu-west-1 --cluster-identifier
 mycluster-123456789012
```

2. 執行下列命令來取得快照的詳細資訊。

```
aws redshift describe-cluster-snapshots --region eu-west-1 --snapshot-identifier
 mycluster-snapshot
```

3. 執行下列命令來描述可供此快照使用的選項。

```
aws redshift describe-node-configuration-options --snapshot-identifier mycluster-
snapshot --region eu-west-1 --action-type restore-cluster
```

此命令會傳回一個具有所建議節點類型的選項清單、節點數，以及每個選項的磁碟使用率。在此範例中，上述命令列出下列可能的節點組態。我們選擇還原到四節點叢集。

```
{
    "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. 執行下列命令，將快照還原至我們選擇的叢集組態。還原此叢集之後，我們擁有與來源叢集相同的內容，但資料已合併為四個ra3.16xlarge節點。

```
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
```

如果您有預留節點，例如 RA3 預留節點，您可以升級至 RG 預留節點。您可以在從快照還原時或在執行彈性調整大小時進行此升級。您可以使用主控台來引導您完成此程序。如需升級至 RG 節點的詳細資訊，請參閱升級至 RG 或 RA3 節點類型。

在主控台上從快照還原叢集

1. 登入 AWS 管理主控台 ，並在 https：//https://console.aws.amazon.com/redshiftv2/ 開啟 Amazon Redshift 主控台。
2. 在導覽功能表中，選擇叢集、快照，然後選擇要還原的快照。
3. 選擇 Restore from snapshot (從快照還原) 以檢視要使用快照資訊建立之新叢集的 Cluster configuration (叢集組態) 和 Cluster details (叢集詳細資訊) 值。
4. 更新新叢集的屬性，然後選擇 Restore cluster from snapshot (從快照還原叢集)。

還原叢集快照後，還原的資料倉儲會使用拍攝快照時所使用的相同自訂 AWS KMS 金鑰進行加密。如果快照沒有自訂 KMS 金鑰，Amazon Redshift 的備份加密邏輯會取決於下列因素：

- 您要還原快照的目標 Amazon Redshift 資料倉儲類型。
- 拍攝快照時叢集的加密類型。

若要了解從叢集快照還原資料倉儲後，資料倉儲如何進行加密，請參閱下表：

目的地類型	快照加密類型	目的地加密類型
佈建叢集	使用 加密 AWS 受管金鑰	使用 加密 AWS 受管金鑰
佈建叢集	使用 加密 AWS 擁有的金鑰	使用 加密 AWS 擁有的金鑰
無伺服器命名空間	使用 加密 AWS 受管金鑰	使用 加密 AWS 擁有的金鑰
無伺服器命名空間	使用 加密 AWS 擁有的金鑰	使用 加密 AWS 擁有的金鑰

如果在拍攝快照時 AWS Secrets Manager 管理叢集的管理員密碼，您必須繼續使用 AWS Secrets Manager 來管理管理員密碼。您可以在還原叢集之後，透過在叢集詳細資訊頁面中更新叢集的管理員憑證，以選擇不使用機密。

如果您有預留節點，則可以升級至 RG 預留節點。您可以在從快照還原時或在執行彈性調整大小時進行此升級。您可以使用主控台來引導您完成此程序。如需升級至 RG 節點的詳細資訊，請參閱升級至 RG 或 RA3 節點類型。

從快照還原資料表

您可以從快照來還原單一資料表，而無須還原整個叢集。當您自快照還原單一資料表時，您將指定來源快照、資料庫、結構描述與資料表名稱，以及還原的資料表目標資料庫、結構描述和新名稱。

> **Note**
>
> RG 或 RA3 佈建叢集和 Amazon Redshift Serverless 工作群組不支援無備份資料表。在 RG 或 RA3 叢集或無伺服器工作群組中標記為無備份的資料表會被視為永久資料表，在拍攝快照時一律會備份，並在從快照還原時一律還原。不過，不支援選擇性還原無備份資料表。

新資料表名稱不能與現有資料表名稱重複。若您要將從快照還原的資料表取代現有資料表，請在從快照還原之前先重新命名或刪除現有資料表。

已使用資料表的欄位定義、資料表屬性和欄位屬性 (外部索引鍵除外) 建立目標資料表。為避免因相依性而導致的衝突，目標資料表不會從來源資料表繼承外部索引鍵。任何相依性，例如檢視或來源資料表上授予的許可，都不會套用至目標資料表。

若來源資料表存在擁有者，則該資料庫使用者是已還原資料表的擁有者，前提是該使用者有足夠的許可以成為指定資料庫和結構描述中關係的擁有者。否則，還原的資料表將由啟動叢集時建立的管理員使用者所擁有。

還原的資料表會恢復到備份時所處的狀態。這包括由 Amazon Redshift 符合可序列化的隔離所定義的交易可見度規則，這表示在備份後開始傳送的交易中立刻可看見該資料。

以下為從快照還原資料表的限制：

- 您只能將資料表還原至目前、活動執行中的叢集；以及從該叢集拍攝的快照中還原資料表。
- 您一次只能還原一個資料表。
- 您無法從叢集調整大小前拍攝的叢集快照還原資料表。例外情況是，如果節點類型沒有變更，您可以在彈性調整大小之後還原資料表。
- 任何相依性，例如檢視或來源資料表上授予的許可，都不會套用至目標資料表。
- 如果為要還原的資料表開啟了資料列層級安全，Amazon Redshift 會在開啟資料列層級安全的情況下還原資料表。

從快照還原資料表

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後選擇用以還原資料表的叢集。
3. 針對 Actions (動作)，選擇 Restore table (還原資料表) 以顯示 Restore table (還原資料表) 頁面。
4. 輸入要使用的快照、來源資料表和目標資料表資訊，然後選擇 Restore table (還原資料表)。

Example範例：使用 從快照還原資料表 AWS CLI

下列範例使用 `restore-table-from-cluster-snapshot` AWS CLI 命令，從 `sample-database` 中的結構描述還原`my-source-table`資料表`my-snapshot-id`。您可以使用 AWS CLI

命令describe-table-restore-status來檢閱還原操作的狀態。此範例透過名為 mycluster-example 的新資料表還原快照至 my-new-table 叢集。

```
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
```

從快照還原無伺服器命名空間

從快照還原無伺服器命名空間會將所有命名空間的資料庫替換為快照中的資料庫。如需無伺服器快照的詳細資訊，請參閱快照和復原點。當您將佈建叢集快照還原到 Amazon Redshift Serverless 命名空間時，Amazon Redshift 會自動將具有交錯索引鍵的資料表轉換為複合索引鍵。如需排序索引鍵的相關資訊，請參閱使用排序索引鍵。

若要將佈建叢集的快照還原至無伺服器命名空間。

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表中，選擇叢集、快照，然後選擇要使用的快照。
3. 選擇從快照還原、還原至無伺服器命名空間。
4. 選擇想要還原至的命名空間。
5. 確認您要從快照還原。選擇還原。此動作會使用佈建叢集中的資料取代無伺服器命名空間中的所有資料庫。

設定未加密叢集的跨區域快照複本

您可以設定 Amazon Redshift 將叢集的快照複製到另一個 AWS 區域。若要設定跨區域快照複製，您需要為每個叢集啟用此複製功能，並設定複製快照的位置，以及在目的地區域中保留複製自動或手動快照的時間長度 AWS 。當叢集啟用跨區域複製時，所有新的手動和自動快照都會複製到指定的 AWS 區域。複製的快照名稱前面會加上 **copy:**。

設定跨區域快照

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後選擇您要移動快照的叢集。
3. 針對動作，選擇設定跨區域快照。

   隨即會出現 [設定跨區域] 對話方塊。
4. 針對複製快照，選擇是。
5. 在目的地 AWS 區域中，選擇要複製快照 AWS 的區域。
6. 在自動快照保留期間 （天） 中，選擇您希望自動快照在刪除之前保留在目的地 AWS 區域中的天數。
7. 在手動快照保留期間中，選擇代表您希望手動快照在刪除之前保留在目的地 AWS 區域中的天數的值。如果您選擇自訂值，則保留期必須介於 1 到 3653 天之間。
8. 選擇儲存。

為 AWS KMS加密的叢集設定跨區域快照複本

當您啟動 Amazon Redshift 叢集時，您可以在目的地 AWS 區域中為帳戶中的根金鑰設定快照複製授權。如果您未設定授予，則會使用預設 AWS擁有的金鑰加密目的地區域中的快照。如此，即可讓 Amazon Redshift 在目標 AWS 區域中執行加密操作。

下列程序說明為 AWS KMS加密叢集啟用跨區域快照複本的程序。如需 Amazon Redshift 中的加密和快照複製權限授予的相關資訊，請參閱將 AWS KMS加密的快照複製到另一個快照 AWS 區域。

為 AWS KMS加密的叢集設定跨區域快照

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後選擇您要移動快照的叢集。
3. 針對動作，選擇設定跨區域快照。

   隨即會出現 [設定跨區域] 對話方塊。
4. 針對複製快照，選擇是。
5. 在目的地 AWS 區域中，選擇要複製快照 AWS 的區域。

6. 在自動快照保留期間（天）中，選擇您希望自動快照在刪除之前保留在目的地 AWS 區域中的天數。
7. 在手動快照保留期間，選擇代表您希望手動快照在刪除之前保留在目的地 AWS 區域中的天數的值。如果您選擇自訂值，則保留期必須介於 1 到 3653 天之間。
8. 選擇儲存。

修改手動快照保留期

您可變更手動快照的保留期間，只要修改快照設定即可。

變更手動快照保留期間

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表中，選擇叢集、快照，然後選擇要變更的手動快照。
3. 針對 Actions (動作)，選擇 Manual snapshot settings (手動快照設定) 以顯示手動快照的屬性。
4. 輸入快照定義的修訂屬性，然後選擇 Save (儲存)。

修改跨區域快照複本的保留期間

設定跨區域快照複本後，您也許會想變更設定。您可輕易變更保留期間，只要選取新的天數，並儲存變更即可。

> **Warning**
>
> 設定跨 AWS 區域快照複本後，您無法修改目的地區域。
> 如果您想要將快照複製到不同的 AWS 區域，請先停用跨區域快照複本。然後使用新的目的地 AWS 區域和保留期重新啟用它。在您停用跨區域快照複本後，所有複製的自動化快照都會遭到刪除。因此，建議您判斷是否有想要保留的項目，以便先將其複製到手動快照，再停用跨區域快照複本。

修改跨區域快照

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。

2. 在導覽功能表上，選擇叢集，然後選擇您要修改快照的叢集。
3. 對於 Actions (動作)，選擇 Configure cross-region snapshot (設定跨區域快照) 以顯示快照的屬性。
4. 輸入快照定義的修訂屬性，然後選擇 Save (儲存)。

刪除手動快照

您可以透過選取快照清單中的一或多個快照來刪除手動快照。

刪除手動快照

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ ：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表中，選擇叢集、快照，然後選擇要刪除的快照。
3. 針對 Actions (動作)，選擇 Delete snapshot (刪除快照) 以刪除快照。
4. 確認刪除列出的快照，然後選擇 Delete (刪除)。

將叢集註冊到 AWS Glue Data Catalog

您可以將整個叢集註冊到 ， AWS Glue Data Catalog 並建立 管理的目錄 AWS Glue。您可以使用支援 Apache Iceberg REST API 的任何 SQL 引擎來存取這些目錄。如需從 Amazon Redshift 建立 Apache Iceberg 相容目錄的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 Amazon Redshift 的 Apache Iceberg 相容性。

將叢集註冊到 AWS Glue Data Catalog

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ ：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單上，選擇叢集。 AWS 區域 系統會列出目前 中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。如果您沒有任何叢集，請選擇 Create cluster (建立叢集) 以建立一個叢集。
3. 選擇您要註冊的叢集名稱。
4. 在動作中，選擇註冊。 AWS Glue Data Catalog在 AWS Glue Data Catalog註冊快顯方塊隨即出現。
5. 在目的地 AWS 帳戶 ID 下，輸入您要註冊叢集的帳戶 ID。 這個帳戶 ID 將保存 AWS Glue Data Catalog中的目錄。

6. 在將命名空間註冊為下輸入名稱。這將是叢集在 Data Catalog 中的名稱。
7. 選擇註冊。您將進入 AWS Lake Formation 主控台。
8. 依照 AWS Lake Formation中的目錄建立程序進行。如需建立目錄的相關資訊，請參閱《 AWS Lake Formation 開發人員指南》中的將 Amazon Redshift 資料帶入 AWS Glue Data Catalog。

多可用區域部署

Amazon Redshift 支援佈建 RG 或 RA3 叢集的多個可用區域 （ 多可用區域 ） 部署。透過使用多可用區部署，Amazon Redshift 資料倉儲可以在可用區域發生意外事件時，於失敗情況下繼續運作。多可用區部署會在兩個可用區域 (AZ) 中部署運算資源，而這些運算資源可透過單一端點來存取。如果整個可用區域發生故障，便可使用第二個可用區域中的剩餘運算資源繼續處理工作負載。執行多可用區域資料倉儲時，Amazon Redshift 會針對 RG 或 RA3 收取相同的每小時運算費率。儲存成本保持不變，因為會在所有可用區域內部和 AWS 區域共用。

目前，Amazon Redshift 支援零復原點目標 (RPO)，這可讓資料在系統發生故障時保持最新狀態。透過多可用區部署，Amazon Redshift 可進一步增強其現有的復原功能，並減少其復原時間點目標 (RTO)。這是因為多可用區部署可以更快地從故障或災難中復原，進而將 Amazon Redshift 服務水準協議 (SLA) 提升到 99.99%，相較之下，單一可用區資料倉儲只有 99.9%。

設定多可用區部署

若要設定多可用區部署，請選取多可用區選項，並指定要在每個可用區域中佈建的運算節點數目。Amazon Redshift 會自動跨兩個可用區域部署相等的運算資源，而且在正常操作期間，所有運算資源一律可用於讀取和寫入處理，並且可以透過使用次要叢集中的額外運算資源，在機會上協助提高效能。這可讓多可用區部署作為具有單一端點的單一資料倉儲，而不需要在發生災難時變更應用程式。雖然異地同步備份部署只會使用位於一個可用區域中的運算資源來處理個別查詢，但它可以自動將某些同時查詢的處理分配到兩個可用區域，這些可用區域有時但不一定會提高高並行工作負載的整體輸送量。請務必謹記異地同步備份的主要功能，即為叢集提供高可用性解決方案，以確保從意外停機時間或中斷事件中快速復原。

如果您的主要動機是為分析工作負載取得更高的效能，建議您使用異地同步備份以外的其他解決方案，例如：

- 調整規模為較大的節點執行個體類型 （ 調整叢集大小 ）
- 在叢集上啟用並行擴展 (並行擴展)
- 使用資料共用架構來隔離工作負載 (資料共用概觀)

您也可以將現有的單一可用區資料倉儲轉換為多可用區資料倉儲，反之亦然。除了在第二個可用區域中會佈建額外的運算資源外，其他的部署都相同。從現有的單一可用區叢集遷移至多可用區時，您可能需要將所需的叢集節點數目加倍，以便維持單一查詢效能。

如果可用區域發生故障，Amazon Redshift 會自動使用剩餘可用區域中的資源繼續運作。不過，使用者連線可能會中斷，而且必須重新建立。此外，在失敗的可用區域中執行的查詢可能會失敗，必須重試。不過，您可以立即重新連線到叢集並為查詢重新排程，Amazon Redshift 則會處理剩餘可用區域中的查詢。在多可用區資料倉儲復原時，於故障發生當下起所發出的查詢可能會發生執行時間延遲。

> **Note**
>
> 為了獲得更好的效能和更高的可用性，建議您對多可用區叢集使用 SNAPSHOT ISOLATION。如需詳細資訊，請參閱 CREATE DATABASE。

限制

多可用區資料倉儲具有與單一可用區資料倉儲相同的功能，但下列適用於多可用區資料倉儲的限制除外：

- 您無法建立未加密的多可用區資料倉儲。在建立新的多可用區資料倉儲、將單一可用區資料倉儲轉換為多可用區資料倉儲，或將單一可用區資料倉儲轉換為多可用區資料倉儲時，請務必新增加密。
- 您無法為任何 RG 或 RA3 執行個體類型建立單一節點異地同步備份部署。在建立多可用區部署時，請為每個可用區域選擇 2 個以上的節點。
- Amazon Redshift 不支援可支援少於三個可用區域的子網路組態。換句話說，所設定的子網路群組需要三個以上的子網路。
- 多可用區部署無法重新放置到其他可用區域。使用多可用區部署時，Amazon Redshift 會自動判斷和執行重新放置。
- 您無法暫停或繼續多可用區部署。
- 您無法在支援的連接埠範圍 5431 至 5455 和 8191 至 8215 之外執行多可用區部署。
- 您無法將 STL、SVCS、SVL、SVV、STV 檢視與多可用區部署搭配使用，因為它們只支援系統監控檢視 (SYS_* 檢視)。請變更監控查詢以使用系統監控檢視 (SYS_* 檢視)。
- 您無法將彈性 IP 位址附加至已啟用多可用區域的現有叢集。
- 您無法將已附加彈性 IP 位址的叢集從單一可用區轉換為多可用區域。
- Amazon Redshift 異地同步備份部署可在下列位置使用 AWS 區域：

> **Note**
>
> RG 執行個體可在特定區域使用。請參閱 AWS 區域中的 RG 節點類型可用性。

- 美國東部 (俄亥俄) (us-east-2)
- 美國東部 (維吉尼亞北部) (us-east-1)
- 美國西部 (奧勒岡) (us-west-2)
- 非洲 (開普敦) (af-south-1)
- 亞太區域 (香港) (ap-east-1)
- 亞太區域 (台北) (ap-east-2)
- 亞太區域 (海德拉巴) (ap-south-2)
- 亞太區域 (雅加達) (ap-southeast-3)
- 亞太區域 (馬來西亞) (ap-southeast-5)
- 亞太區域 (墨爾本) (ap-southeast-4)
- 亞太地區 (孟買) (ap-south-1)
- 亞太地區 (大阪) (ap-northeast-3)
- 亞太地區 (首爾) (ap-northeast-2)
- 亞太地區 (新加坡) (ap-southeast-1)
- 亞太地區 (雪梨) (ap-southeast-2)
- 亞太區域 (紐西蘭) (ap-southeast-6)
- 亞太區域 (泰國) (ap-southeast-7)
- 亞太地區 (東京) (ap-northeast-1)
- 加拿大 (中部) (ca-central-1)
- 中國 (北京) (cn-north-1)
- 中國 (寧夏) (cn-northwest-1)
- 歐洲 (法蘭克福) (eu-central-1)
- 歐洲 (愛爾蘭) (eu-west-1)
- 歐洲 (倫敦) (eu-west-2)
- 歐洲 (米蘭) (eu-south-1)
- 歐洲 (巴黎) (eu-west-3)

- 歐洲 (西班牙) (eu-south-2)
- 歐洲 (斯德哥爾摩) (eu-north-1)
- 歐洲 (蘇黎世) (eu-central-2)
- 以色列 (特拉維夫) (il-central-1)
- 墨西哥 (中部) (mx-central-1)
- 中東 (巴林) (me-south-1)
- 中東 (阿拉伯聯合大公國) (me-central-1)
- 南美洲 (聖保羅) (sa-east-1)
- AWS GovCloud （美國東部） (us-gov-east-1)
- AWS GovCloud （美國西部） (us-gov-west-1)

- 可公開存取的多可用區域資料倉儲支援的 VPC 安全群組數量，比單一可用區域和可透過私有方式存取的多可用區域倉儲少 1 個。

在建立新叢集時設定多可用區

Amazon Redshift 多可用區一次支援兩個可用區域。Amazon Redshift 會自動根據選取的子網路群組組態來選取可用區域。您可以將現有的單一可用區域資料倉儲轉換為多可用區資料倉儲，或從快照還原以將其設定為多可用區資料倉儲。

您可以使用 Amazon Redshift 主控台輕鬆建立新的多可用區部署。若要使用 Amazon Redshift 主控台建立新的多可用區部署，請在建立資料倉儲時選取多可用區選項。指定單一可用區域中所需的運算節點數量，Amazon Redshift 便會在兩個可用區域中各自部署該數量的節點。在正常操作期間，所有節點都會用於讀取和寫入工作負載的處理。您也可以使用 AWS CLI `create-cluster`命令，使用 `multi-az` 參數建立新的多可用區域資料倉儲。

您可以將現有的單一可用區資料倉儲轉換為多可用區資料倉儲，您可以使用 Amazon Redshift 主控台或使用 `multi-az` 參數的 AWS CLI `modify-cluster`命令。或者，您可以從快照還原，使用 Amazon Redshift 主控台或使用 `multi-az` 參數的 AWS CLI `restore-from-cluster-snapshot`命令，將單一可用區資料倉儲設定為多可用區資料倉儲。

異地同步備份部署僅支援使用 Amazon Redshift 受管儲存 (RMS) 的 RG 或 RA3 節點類型。Amazon Redshift 會將資料存放在 RMS 中，該 RMS 使用 Amazon S3 且可在 中的所有可用區域中存取 AWS 區域，而無需在 Amazon Redshift 層級複寫資料。

您可以使用 Amazon Redshift 主控台或 AWS Command Line Interface在建立新叢集時設定多可用區部署。

使用主控台

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇佈建叢集儀表板，然後選擇叢集。 AWS 區域 系統會列出目前 中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。
3. 選擇建立叢集按鈕以開啟 [建立叢集] 頁面。
4. 輸入叢集的屬性。如需有關建立叢集的一般資訊，請參閱建立叢集。
5. 從節點類型下拉式清單中選擇其中一個 RG 或 RA3 節點類型。只有在您選擇 RG 或 RA3 節點類型時，AZ 組態選項才會變成可用。
6. 在 AZ 組態下，選擇多可用區。
7. 在每個可用區域的節點數量下，為叢集輸入至少兩個節點。
8. 您可以選擇載入範例資料或帶入自己的資料：
   - 在範例資料中，選擇載入範例資料，將範例資料集載入 Amazon Redshift 叢集。Amazon Redshift 會將範例資料集 Tickit 載入預設 dev 資料庫和 public 結構描述。Amazon Redshift 會自動將範例資料集載入您的 Amazon Redshift 叢集。您可以開始使用查詢編輯器 v2 來查詢資料。
   - 若要將自己的資料帶入您的 Amazon Redshift 叢集，請按照將自己的資料帶入 Amazon Redshift 中的步驟進行操作。
9. 向下捲動至其他組態、展開網路和安全，並務必接受預設的叢集子網路群組，或選擇另一個群組。如果您選擇另一個叢集子網路群組，請確定您選取的子網路群組中有 3 個可用區域。
10. 在其他組態下，展開資料庫組態。
11. 若要使用自訂 AWS KMS 金鑰而非預設 AWS擁有的金鑰，請按一下資料庫加密下的自訂加密設定。
12. 在選擇 KMS 金鑰下，您可以選擇 AWS Key Management Service 金鑰或輸入 ARN。或者，您可以按一下 AWS Key Management Service 主控台中的建立 AWS Key Management Service 金鑰。如需建立 KMS 金鑰的相關資訊，請參閱《AWS Key Management Service 開發人員指南》中的建立金鑰。
13. 按一下建立叢集。叢集建立成功時，您可以在叢集詳細資訊頁面中檢視詳細資訊。您可以使用 SQL 用戶端來載入和查詢資料。

使用 AWS Command Line Interface

使用 建立叢集時設定異地同步備份 AWS Command Line Interface

- 從 AWS CLI 使用 `create-cluster`命令和 `multi-az` 參數，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
```

為從快照還原的資料倉儲設定多可用區

若要透過從快照還原的方式建立新的多可用區域叢集，請完成下列程序。

使用主控台

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集、快照，然後選擇要使用的快照。
3. 選擇還原快照、還原至佈建叢集。
4. 輸入叢集的屬性。如需有關建立叢集的一般資訊，請參閱建立叢集。
5. 從節點類型下拉式清單中選擇其中一個 RG 或 RA3 節點類型。只有在您選擇 RG 或 RA3 節點類型時，AZ 組態選項才會變成可用。
6. 在 AZ 組態下，選擇多可用區。
7. 在每個可用區域的節點數量下，為叢集輸入至少兩個節點。
8. 您可以選擇載入範例資料或帶入自己的資料：

- 在範例資料中，選擇載入範例資料，將範例資料集載入 Amazon Redshift 叢集。Amazon Redshift 會將範例資料集 Tickit 載入預設 dev 資料庫和 public 結構描述。Amazon Redshift 會自動將範例資料集載入您的 Amazon Redshift 叢集。您可以開始使用查詢編輯器 v2 來查詢資料。
- 若要將自己的資料帶入您的 Amazon Redshift 叢集，請按照從 Amazon S3 將資料載入 Amazon Redshift 中的步驟進行操作。

9. 向下捲動至其他組態、展開網路和安全，並務必接受預設的叢集子網路群組，或選擇另一個群組。如果您選擇另一個叢集子網路群組，請確定您選取的子網路群組中有 3 個可用區域。
10. 在其他組態下，展開資料庫組態。
11. 在資料庫加密下，若要使用預設擁有金鑰以外的自訂 KMS AWS金鑰，請按一下自訂加密設定。預設會取消選取此選項。
12. 在選擇 KMS 金鑰下，您可以選擇 AWS Key Management Service 金鑰或輸入 ARN。或者，您可以按一下 AWS Key Management Service 主控台中的建立 AWS Key Management Service 金鑰。如需建立 KMS 金鑰的相關資訊，請參閱《AWS Key Management Service 開發人員指南》中的建立金鑰。
13. 按一下從快照還原叢集。叢集還原成功時，您可以在叢集詳細資訊頁面中檢視詳細資訊。

使用 AWS Command Line Interface

- 從 中 AWS CLI，使用 `restore-from-cluster-snapshot`命令，如下所示。

```
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/
```

將單一可用區資料倉儲轉換為多可用區資料倉儲

透過將單一可用區資料倉儲轉換為多可用區資料倉儲，您的資料倉儲將具備 99.99% SLA 保證的高可用性。即使使用多可用區資料倉儲，個別查詢的效能仍會保持不變。對於較高的並行工作負載數量，由於 Amazon Redshift 可以使用兩個可用區域中的運算資源來執行請求，因此整體輸送量會有所提升。

> **Note**
>
> Amazon Redshift 不允許您在從單一可用區轉換為多可用區時分割現有的運算資源，反之亦然。不支援透過此操作來維持一致的個別查詢效能。

使用主控台

使用主控台將單一可用區叢集轉換為多可用區資料倉儲

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ ：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇佈建叢集儀表板，然後選擇叢集。 AWS 區域 系統會列出目前 中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。
3. 選擇您要轉換為多可用區部署的叢集。隨即會出現叢集詳細資訊頁面。
4. 對於動作，選擇啟用多可用區。隨即會出現修改摘要。按一下啟用多可用區。
5. 發生錯誤時，請執行下列其中一項動作，然後按一下啟用多可用區。
   - 叢集加密 — 選擇屬性可在叢集詳細資訊頁面之 [屬性] 索引標籤底下的 [資料庫組態] 區段中編輯加密設定。
   - 子網路群組 — 選擇子網路群組，可透過按一下子網路群組連結來編輯叢集子網路群組設定。如果您選擇另一個叢集子網路群組，請確定您選取的子網路群組中有 3 個可用區域。
   - 連接埠設定 — 選擇屬性可在叢集詳細資訊頁面之 [屬性] 索引標籤底下的 [資料庫組態] 區段中編輯連接埠設定。
6. 您可以使用 SQL 用戶端來載入和查詢資料。

使用 AWS Command Line Interface

- 從 中 AWS CLI，使用 `modify-cluster`命令和 `multi-az` 參數，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
```

您無法將 STL、SVCS、SVL、SVV 或 STV 檢視與多可用區域部署搭配使用，因為它們只支援系統監控檢視 (SYS_* 檢視)。請變更監控查詢以使用系統監控檢視 (SYS_* 檢視)。

將多可用區資料倉儲轉換為單一可用區資料倉儲

透過將多可用區資料倉儲轉換為單一可用區資料倉儲，您的資料倉儲將不會獲得多可用區所提供的 99.99% SLA 保證。個別查詢的效能會保持不變，但是整體輸送量會受到影響，因為無法使用第二個可用區域中的運算資源。您可以選擇啟用並行擴展，自動擴展輸送量以在使用單一可用區時仍可獲得一致的效能。

> **Note**
>
> Amazon Redshift 不允許您在從單一可用區轉換為多可用區時分割現有的運算資源，反之亦然。不支援透過此操作來維持一致的個別查詢效能。

使用主控台

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇佈建叢集儀表板，然後選擇叢集。 AWS 區域 系統會列出目前 中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。
3. 選擇您要轉換為多可用區部署的叢集。隨即會出現叢集詳細資訊頁面。
4. 對於動作，選擇停用多可用區。隨即會出現修改摘要。按一下停用多可用區。

使用 AWS Command Line Interface

- 從 中 AWS CLI，使用 `modify-cluster`命令和 `no-multi-az` 參數，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
```

一旦您的資料倉儲轉換為單一可用區，就會失去 99.99 的 SLA 保證。整體輸送量也會受到影響。儲存變更後，便可以在叢集詳細資訊頁面中檢視詳細資訊。

調整多可用區資料倉儲的大小

您可以調整多可用區資料倉儲的大小，並指定與目前的資料倉儲組態不同的節點數量或節點類型。

使用主控台

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇佈建叢集儀表板，然後選擇叢集。 AWS 區域 系統會列出目前 中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。
3. 選擇您要調整多可用區資料倉儲大小的叢集。隨即會出現叢集詳細資訊頁面。
4. 針對 Actions (動作)，選擇 Resize (調整大小)。Resize cluster (調整叢集大小) 頁面隨即出現。
5. 遵循頁面說明以進行操作。您可以在特定時間立即重新調整叢集的大小一次，或是根據排程增加或減少您叢集的大小。
6. 在新組態下，從節點類型下拉式清單中選擇其中一個 RG 或 RA3 節點類型。
7. 按一下調整叢集大小。

使用 AWS Command Line Interface

使用 調整多可用區域資料倉儲的大小 AWS Command Line Interface

- 從 中 AWS CLI，使用 `resize-cluster`命令來變更單一可用區域的節點數量，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
```

容錯移轉多可用區部署

多可用區資料倉儲是同時部署在兩個可用區域中的集合運算資源。部署在主要可用區域中的運算資源稱為主要運算，部署在次要可用區域中的運算資源則稱為次要運算。在可用區域或基礎設施發生故障等罕見事件時，多可用區資料倉儲不必使用者介入即可自動復原。復原程序包括從主要運算容錯移轉到次要運算，以及將次要運算資源指定為主要運算資源。此外，新的次要運算資源也會佈建到第三個可用區域。自動復原程序會根據 RTO 和 RPO 來進行測量。

- 復原時間點目標 (RTO) — 系統在災難發生後恢復運作狀態所需的時間。換言之，RTO 會測量停機時間。
- 復原點目標 (RPO) — 可能遺失的資料量 (以時間為單位)。對於 Amazon Redshift 多可用區資料倉儲而言，RPO 一般是零，因為所有資料都儲存在 Amazon Redshift 受管儲存 (RMS) 中，並由 Amazon Simple Storage Service 提供支援，其預設具有高耐用性和高可用性。

> **Note**
>
> 發生容錯移轉之後，個別查詢的效能不會發生變化。由於其中一個可用區域中的運算資源無法使用，因此資料倉儲的整體輸送量會短暫降低。不過，Amazon Redshift 會自動取得另一個可用區域中的容量，確保系統會恢復相同的資料倉儲處理容量。

除了自動復原程序外，您也可以使用容錯移轉主要運算選項，針對資料倉儲手動觸發此程序。您可以使用此方法來測試多可用區對於讓應用程式獲得更高的高可用性和更好的連續性有多大幫助。

使用主控台

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 執行以下任意一項：
   - 在導覽功能表上，選擇叢集。在叢集底下，選擇叢集。隨即會出現叢集詳細資訊頁面。
   - 從叢集儀表板中選擇叢集。
3. 從動作中，選擇容錯移轉主要運算。
4. 出現提示時，按一下確認。

使用 AWS Command Line Interface

- 從 中 AWS CLI，使用 `failover-primary-compute`命令，如下所示。

```
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
```

確認上述操作後，Amazon Redshift 便會執行與從可用區域或基礎設施故障自動復原時相同的步驟。此程序會導致主要可用區域中的運算節點無法使用，並將次要可用區域中的運算資源指定為主要運算。成功完成叢集復原後，多可用區部署便會可供使用。多可用區資料倉儲也會在別的第三個可用區域可供使用時自動在其中佈建新的次要運算。

在這個過程中，當叢集自動復原並重新設定回多可用區部署設定時，主控台上的叢集狀態會在整個時間都顯示為正在修改。叢集可以立即接受新的連線。現有連線和執行中的查詢可能會遭到捨棄。您可以立即重試這些項目。

檢視多可用區資料倉儲的查詢和載入

無論叢集的類型、大小和狀態 (暫停或繼續) 為何，您都可以檢視過去 7 天所執行之查詢的相關資訊。

查詢和載入頁面上顯示的資訊會填入來自 Amazon Redshift 系統資料表 (SYS_* 檢視) 的資訊。此資訊可讓您顯示有關查詢的其他資訊，並提供滾動 7 天的保留期。查詢診斷會變得更快，可讓您依資料庫、使用者名稱或 SQL 陳述式類型來篩選資料。若要查看這些額外的篩選條件和所有已執行查詢的資訊，請注意下列先決條件：

- 您必須選擇連線至資料庫來連線至資料庫。
- 資料庫使用者必須具有 sys:operator 或 sys:monitor 角色和許可，才能執行查詢監控。如需有關系統角色的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 Amazon Redshift 系統定義角色。

在連線至資料庫後，便會看到這些額外的篩選條件和查詢資訊。

顯示來自查詢和載入的查詢效能資料

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇查詢和載入以顯示帳戶的查詢清單。
3. 您可能必須連線至資料庫才能查看其他篩選條件。如有必要，請按一下連線至資料庫，然後依照提示進行來連線至資料庫。

   根據預設，此清單會顯示過去 24 小時您所有叢集的查詢。您可在主控台中變更顯示日期的範圍。

顯示來自查詢監控的查詢效能資料

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集。在叢集底下，選取叢集。
3. 選擇查詢監控。
4. 視叢集的組態或版本而定，您可能必須連線至資料庫才能看到其他篩選條件。如有必要，請按一下連線至資料庫，然後依照提示進行來連線至資料庫。

監控多可用區部署中的查詢

多可用區部署會使用同時部署在兩個可用區域的運算資源，並可在指定可用區域中的資源無法使用時繼續運作。系統隨時會使用所有運算資源。這可讓系統以主動-主動的方式跨兩個可用區域進行完整操作，以便執行讀取和寫入操作。

您可以在 pg_catalog 結構描述中查詢 SYS_ 檢視，以監控多可用區部署中的查詢執行時間。SYS_ 檢視會顯示主要和次要叢集的查詢執行時間活動或統計資料。如需監控檢視的清單，請參閱監控檢視。

請依照下列步驟監控多可用區部署內每個可用區域的查詢執行時間：

1. 導覽至 Amazon Redshift 主控台，並連線到多可用區部署中的資料庫，然後透過查詢編輯器執行查詢。
2. 在多可用區 Amazon Redshift 部署上執行任何範例查詢。
3. 對於多可用區部署，您可以透過使用 SYS_QUERY_HISTORY 資料表中的 compute_type 資料欄來識別查詢和用來執行查詢的可用區域。「主要」代表在多可用區部署中的主要叢集上執行的查詢，「次要」代表在多可用區部署中的次要叢集上執行的查詢。

   下列查詢會使用 compute_type 資料欄來監控查詢。

```
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;
```

終止叢集的查詢

此程序同時適用於多可用區叢集和單一可用區叢集。

終止查詢

您也可以使用 Queries (查詢) 頁面終止目前正在進行的查詢。

資料庫使用者必須具有 sys:operator 角色和許可，才能結束執行中的查詢。如需有關系統角色的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 Amazon Redshift 系統定義角色。

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇查詢和載入以顯示帳戶的查詢清單。
3. 選擇清單中要終止的執行中查詢，然後選擇 Terminate query (終止查詢)。

監控 Amazon Redshift 叢集效能

Amazon Redshift 提供您可以追蹤叢集和資料庫之運作狀態和效能的效能指標和資料。在本節中，我們討論您可在 Amazon Redshift (特別是 Amazon Redshift 主控台) 中處理的資料類型。

您可以在 Amazon Redshift 主控台中使用的效能資料之種類有兩種：

- Amazon CloudWatch 指標 — Amazon CloudWatch 指標可協助您監控叢集的實體層面，例如 CPU 利用率、延遲和輸送量。Amazon Redshift 主控台中將直接顯示指標資料。您也可以在 CloudWatch 主控台中檢視此指標資料。或者，您可以用任何其他方式使用指標，例如使用 AWS CLI 或其中一個 AWS SDKs。
- 查詢/載入效能資料 — 效能資料可協助您監控資料庫活動和效能。此資料會在 Amazon Redshift 主控台中彙總，協助您輕鬆地將在 CloudWatch 指標中所見內容與特定資料庫查詢和載入事件關聯在一起。您也可以建立自己的自訂效能查詢，並在資料庫上執行他們。Amazon Redshift 主控台中僅顯示查詢和載入效能資料。不以 CloudWatch 指標形式發佈。

效能資料會整合至 Amazon Redshift 主控台中，以下列方式帶來更豐富的體驗：

- 在您檢視叢集，而您可以需要進行如重新調整大小之與叢集有關的決定時，與叢集關聯的效能資料會隨之顯示。

- 相較於 CloudWatch，某些效能指標會在 Amazon Redshift 主控台中，以更適當的調整後單位顯示。例如，`WriteThroughput` 會以 GB/s 顯示 (在 CloudWatch 中以 bytes/s 顯示)，此單位和一般節點儲存空間的關聯性更高。
- 您可以輕鬆在相同圖表上一起顯示某叢集之節點的效能資料。您可以透過此方式來監控叢集之所有節點的效能。您也可以檢視每個節點的效能資料。

Amazon Redshift 提供效能資料 (CloudWatch 指標以及查詢和載入資料)，不另行收費。效能資料會每秒記錄。您可以在 Amazon Redshift 主控台中存取效能資料的歷史值。如需有關使用 CloudWatch 存取以 CloudWatch 指標形式公開的 Amazon Redshift 效能資料的詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的什麼是 CloudWatch？。

Amazon Redshift 中的效能資料

您可以使用適用於 Amazon Redshift 的 CloudWatch 指標，取得叢集運作狀態和效能，並查看節點層級的資訊。使用這些指標時，請謹記每個指標都有一或多個與其關聯的維度。您可透過這些維度來知道指標適用的項目，也就是指標的範圍。Amazon Redshift 具有以下兩個維度：

- 具有 `NodeID` 維度的指標即為提供叢集的節點之效能資料的指標。這組指標包含領導與運算節點。這些指標的範例包括：`CPUUtilization`、`ReadIOPS`、`WriteIOPS`。
- 僅具有 `ClusterIdentifier` 維度的指標即為提供叢集之效能資料的指標。這些指標的範例包括：`HealthStatus` 和 `MaintenanceMode`。

> **Note**
>
> 在這些指標案例中，叢集特定指標代表的是節點行為的彙總。在這些案例中，請留意指標值的解譯，因為領導節點的行為會透過運算節點來彙總。

如需 CloudWatch 指標和維度的一般資訊，請參閱《Amazon CloudWatch 使用者指南》中的 CloudWatch 概念。

如需 Amazon Redshift 之 CloudWatch 指標的進一步說明，請參閱以下區段。

主題

- Amazon Redshift 指標
- Amazon Redshift 指標的維度
- Amazon Redshift 查詢和載入效能資料

Amazon Redshift 指標

AWS/Redshift 命名空間包含下列指標。除非另有說明，否則會每隔 1 分鐘收集指標。

指標	說明
CommitQueueLength	在指定時間點等待認可的交易數目。 單位：Count 維度：ClusterIdentifier
ConcurrencyScaling ActiveClusters	在任何指定時間主動處理查詢的並行擴展叢集數量。 單位：Count 維度：ClusterIdentifier
ConcurrencyScaling Seconds	有主動查詢處理活動的並行擴展叢集所使用的秒數。 單位：Count 維度：ClusterIdentifier
CPUUtilization	CPU 使用率。以叢集而言，此指標代表所有節點 (領導者與運算) CPU 使用值的彙整。 單位：百分比 維度：ClusterIdentifier 、NodeID 維度：ClusterIdentifier
DatabaseConnections	叢集的資料庫連線數量。 單位：Count 維度：ClusterIdentifier
HealthStatus	表示叢集運作狀態。叢集每分鐘連線至其資料庫並執行建議查詢。如果成功執行此操作，該叢集將被視為運作正常。否則，

指標	說明
	表示該叢集運作不良。不良狀態可能會發生於資料庫處於極高負載，或叢集上的資料庫有組態的問題。 **Note** 在 Amazon CloudWatch 中，此指標會回報為 1 或 0，在 Amazon Redshift 主控台則會與文字 HEALTHY 或 UNHEALTHY 一併顯示以提供便利性。此指標顯示於 Amazon Redshift 主控台時，將忽略取樣平均值，只會顯示 HEALTHY 或 UNHEALTHY 。在 Amazon CloudWatch 中，由於取樣的問題，可能會顯示 1 與 0 以外的值。小於 1 的任何 HealthStatus 值都將回報為 0 (UNHEALTHY)。 單位：計數 (1/0) (在 Amazon Redshift 主控台中為 HEALTHY/UNHEALTHY) 維度：ClusterIdentifier
MaintenanceMode	表示叢集是否處於維護模式。 **Note** 在 Amazon CloudWatch 中，此指標會回報為 1 或 0，在 Amazon Redshift 主控台則會與文字 ON 或 OFF 一併顯示以提供便利性。此指標顯示於 Amazon Redshift 主控台時，將忽略取樣平均值，只會顯示 ON 或 OFF。在 Amazon CloudWatch 中，由於取樣的問題，可能會顯示 1 與 0 以外的值。大於 0 的任何 MaintenanceMode 值都將回報為 1 (ON)。 單位：計數 (1/0) (在 Amazon Redshift 主控台中為 ON/OFF)。 維度：ClusterIdentifier

指標	說明
MaxConfiguredConcurrencyScalingClusters	從參數群組設定檔最大並行擴展叢集數量。如需詳細資訊，請參閱 Amazon Redshift 參數群組。 單位：Count 維度：ClusterIdentifier
NetworkReceiveThroughput	節點或叢集接收資料的速率。 單位：位元組/秒 (在 Amazon Redshift 主控台中為 MB/秒) 維度：ClusterIdentifier 、NodeID 維度：ClusterIdentifier
NetworkTransmitThroughput	節點或叢集寫入資料的速率。 單位：位元組/秒 (在 Amazon Redshift 主控台中為 MB/秒) 維度：ClusterIdentifier 、NodeID 維度：ClusterIdentifier
PercentageDiskSpaceUsed	已使用的磁碟空間的百分比。 單位：百分比 維度：ClusterIdentifier 維度：ClusterIdentifier 、NodeID
QueriesCompletedPerSecond	平均每秒完成的查詢次數。每隔 5 分鐘回報。單一節點叢集不支援此指標。 單位：計數/秒 維度：ClusterIdentifier 、latency 維度：ClusterIdentifier 、wlmid

指標	說明
QueryDuration	完成查詢的平均時間。每隔 5 分鐘回報。單一節點叢集不支援此指標。 單位：微秒 維度：ClusterIdentifier 、NodeID、latency 維度：ClusterIdentifier 、latency 維度：ClusterIdentifier 、NodeID、wlmid
QueryRuntimeBreakdown	查詢階段執行所花費的查詢總時間。每隔 5 分鐘回報。 單位：毫秒 維度：ClusterIdentifier、NodeID、stage 維度：ClusterIdentifier、stage
ReadIOPS	磁碟讀取操作的每秒平均次數。 單位：計數/秒 維度：ClusterIdentifier 、NodeID 維度：ClusterIdentifier
ReadLatency	用於磁碟讀取 I/O 操作的平均時間量。 單位：秒 維度：ClusterIdentifier 、NodeID 維度：ClusterIdentifier

指標	說明
ReadThroughput	平均每秒從磁碟讀取的位元組數目。 單位：位元組 (在 Amazon Redshift 主控台中為 GB/秒) 維度：ClusterIdentifier 、NodeID 維度：ClusterIdentifier
RedshiftManagedStorageTotalCapacity	受管儲存總容量。 單位：MB 維度：ClusterIdentifier
RgUniqueSnapshotStorage	在 RG 執行個體系列的所有作用中手動快照中存放的唯一資料區塊數。 單位：MB
TotalTableCount	特定時間點的資料表總數。此總計包含永久資料表、暫時資料表、資料共用資料表、外部資料表和具體化檢視。此計數不包含檢視和系統資料表。 單位：Count 維度：ClusterIdentifier
WLMQueueLength	等候進入工作負載管理 (WLM) 佇列中的查詢數量。 單位：Count 維度：ClusterIdentifier 、service class 維度：ClusterIdentifier 、QueueName

指標	說明
WLMQueueWaitTime	查詢在工作負載管理 (WLM) 佇列中等待的總時間。每隔 5 分鐘回報。 單位：毫秒。 維度：ClusterIdentifier 、QueryPriority 維度：ClusterIdentifier 、wlmid 維度：ClusterIdentifier 、QueueName
WLMQueriesCompletedPerSecond	工作負載管理 (WLM) 佇列平均每秒完成的查詢次數。每隔 5 分鐘回報。單一節點叢集不支援此指標。 單位：計數/秒 維度：ClusterIdentifier 、wlmid 維度：ClusterIdentifier 、QueueName
WLMQueryDuration	工作負載管理 (WLM) 佇列完成查詢的平均時間長度。每隔 5 分鐘回報。單一節點叢集不支援此指標。 單位：微秒 維度：ClusterIdentifier 、wlmid 維度：ClusterIdentifier 、QueueName
WLMRunningQueries	每個 WLM 佇列從主要叢集和並行擴展叢集執行的查詢數。 單位：Count 維度：ClusterIdentifier 、wlmid 維度：ClusterIdentifier 、QueueName

指標	說明
WriteIOPS	寫入操作的每秒平均次數。 單位：計數/秒 維度：ClusterIdentifier 、NodeID 維度：ClusterIdentifier
WriteLatency	用於磁碟寫入 I/O 操作的平均時間量。 單位：秒 維度：ClusterIdentifier 、NodeID 維度：ClusterIdentifier
WriteThroughput	平均每秒寫入磁碟的位元組數目。 單位：位元組 (在 Amazon Redshift 主控台中為 GB/秒) 維度：ClusterIdentifier 、NodeID 維度：ClusterIdentifier
SchemaQuota	所設定的結構描述配額。 單位：MB 維度：ClusterIdentifier 、Database、Schema 定期/推送：Periodic 頻率：5 minutes 停止條件：已捨棄結構描述或已移除配額

指標	說明
NumExceededSchemaQuotas	超出配額的結構描述數目。 單位：Count 維度：ClusterIdentifier 定期/推送：Periodic 頻率：5 minutes 停止條件：無
StorageUsed	結構描述使用的磁碟或儲存空間。 單位：MB 維度：ClusterIdentifier 、Database、Schema 定期/推送：Periodic 頻率：5 minutes 停止條件：已捨棄結構描述或已移除配額
PercentageQuotaUsed	已使用的磁碟或儲存空間相對於已設定的結構描述配額的百分比。 單位：百分比 維度：ClusterIdentifier 、Database、Schema 定期/推送：Periodic 頻率：5 minutes 停止條件：已捨棄結構描述或已移除配額

指標	說明
UsageLimitAvailable	根據 FeatureType，UsageLimitAvailable 會傳回下列項目： • 如果 FeatureType 為 CONCURRENCY_SCALING ，UsageLimitAvailable 會傳回以 1 分鐘為增量單位的並行擴縮所使用的總時間量。此時間不會計入 的可用時間CONCURRENCY_SCALING_AUTO_TASK ，反之亦然。 • 如果 FeatureType 為 CONCURRENCY_SCALING_AUTO_TASK ，則 UsageLimitAvailable 會傳回並行擴展可用於自動最佳化任務的總時間量。此時間不會計入 的可用時間CONCURRENCY_SCALING ，反之亦然。 • 如果 FeatureType 為 EXTRA_COMPUTE_FOR_AUTOMATIC_OPTIMIZATION ，則 UsageLimitAvailable 會傳回可用於自動最佳化任務的總時間量。 • 如果 FeatureType 為 CROSS_REGION_DATASHARING ，則 UsageLimitAvailable 傳回可以用 1 TB 為增量單位進行掃描的總資料量。 • 如果 FeatureType 為 SPECTRUM，則 UsageLimitAvailable 傳回可以用 1 TB 為增量單位進行掃描的總資料量。 單位：分鐘或 TB 維度：ClusterIdentifier 、FeatureType 、UsageLimitId

指標	說明
UsageLimitConsumed	根據 FeatureType，UsageLimitConsumed 會傳回下列項目： • 如果 FeatureType 為 CONCURRENCY_SCALING ，則 UsageLimitConsumed 會以 1 分鐘的增量傳回並行擴展所使用的總時間量。此時間不會計入用於 的時間CONCURRENCY_SCALING_AUTO_TASK ，反之亦然。 • 如果 FeatureType 為 CONCURRENCY_SCALING_AUTO_TASK ，則 UsageLimitConsumed 會傳回並行擴展用於自動最佳化任務的總時間量。此時間不會計入用於 的時間CONCURRENCY_SCALING ，反之亦然。 • 如果 FeatureType 為 EXTRA_COMPUTE_FOR_AUTOMATIC_OPTIMIZATION ，則 UsageLimitConsumed 會傳回用於自動最佳化任務的總時間量。 • 如果 FeatureType 為 CROSS_REGION_DATASHARING ，則 UsageLimitConsumed 會傳回以 1-TB 增量掃描的資料總量。 • 如果 FeatureType 為 SPECTRUM，則 UsageLimitConsumed 會以 1-TB 的增量傳回掃描的資料總量。 單位：分鐘或 TB 維度：ClusterIdentifier 、FeatureType 、UsageLimitId

Amazon Redshift 指標的維度

Amazon Redshift 資料可依下表中任何維度進行篩選。

維度	說明
latency	可能的值如下： • 短 — 少於 10 秒 • 中 — 介於 10 秒到 10 分鐘

維度	說明
	• 長 — 超過 10 分鐘
NodeID	篩選專屬於某個叢集的節點的資料。NodeID 可以是「Leader」、「Shared」或「Compute-N」，其中的 N 是 0, 1, ...，亦即叢集中的節點數量。「Shared」表示該叢集只有一個節點，例如領導節點與運算節點結合。 領導者節點與運算節點回報的指標只有 CPUUtilization 、NetworkTransmitThroughput 及 ReadIOPS。只有運算節點會回報使用 NodeId 維度的其他指標。
ClusterIdentifier	篩選專屬於某個叢集的資料。專屬於叢集的指標包括 HealthStatus 、MaintenanceMode 及 DatabaseConnections 。通常，此維度中的指標 (例如 ReadIOPS) 也是代表節點指標資料彙總的節點指標。在解譯這些指標時應特別注意，因為它們彙整了領導與運算節點的行為。
service class	WLM 服務類別的識別碼。
stage	查詢的執行階段。可能的值如下： • QueryPlanning：剖析與最佳化 SQL 陳述式花費的時間。 • QueryWaiting：花費在 WLM 佇列中等待的時間。 • QueryExecutingRead：執行讀取查詢所花費的時間。 • QueryExecutingInsert：執行插入查詢所花費的時間。 • QueryExecutingDelete：執行刪除查詢所花費的時間。 • QueryExecutingUpdate：執行更新查詢所花費的時間。 • QueryExecutingCtas：執行建立表格為查詢所花費的時間。 • QueryExecutingUnload：執行卸載查詢所花費的時間。 • QueryExecutingCopy：執行複製查詢所花費的時間。 • QueryCommit：花費在認可的時間。
wlmid	工作負載管理佇列的識別碼。

維度	說明
QueryPriority	查詢的優先順序。可能值為 CRITICAL、HIGHEST、HIGH、NORMAL、LOW 以及 LOWEST。
QueueName	工作負載管理佇列的名稱。
FeatureType	受限於使用限制值的功能。可能值為 CONCURRENCY_SCALING 、CROSS_REGION_DATASHARING 及 SPECTRUM。
UsageLimitId	用於限制用量的識別符。

Amazon Redshift 查詢和載入效能資料

除 CloudWatch 指標外，Amazon Redshift 還提供了查詢和載入效能資料。查詢和載入效能資料可用來協助您了解資料庫效能和叢集指標間的關係。例如，如果您注意到叢集的 CPU 用量高升，您可以在叢集 CPU 圖形上找到峰值並查看在當時執行中的查詢。或者，您可以檢閱特定的查詢，指標資料 (例如 CPU) 會顯示在內容中，讓您可以了解查詢對叢集指標的影響。

查詢和載入效能資料不以 CloudWatch 指標形式發佈，且只能在 Amazon Redshift 主控台中檢視。查詢和載入效能資料會透過查詢資料庫的系統資料表來產生 (如需詳細資訊，請參閱《Amazon Redshift 開發人員指南》中的系統資料表參考)。您也可以產生自己的自定資料庫效能查詢，但我們建議開始從在主控台中呈現的查詢和載入效能資料開始。如需自行測量和監控資料庫效能的相關資訊，請參閱《Amazon Redshift 開發人員指南》中的管理效能。

以下資料表說明您可以在 Amazon Redshift 主控台中存取之查詢和載入資料的不同層面。

查詢/載入資料	說明
查詢摘要	在特定期間的查詢清單。可以根據查詢 ID、查詢執行時間和狀態等值排序清單。請在叢集詳細資訊頁面的 Query monitoring (查詢監控) 標籤中檢視此資料。
查詢詳細資訊	提供包含以下項目之特定查詢的詳細資訊： • 查詢 ID、類型、執行查詢的叢集和執行時間等查詢屬性。 • 查詢狀態和錯誤數之類的詳細資訊。 • 執行的 SQL 陳述式。

查詢/載入資料	說明
	• 解釋計畫 (如果有)。 • 查詢執行期間的叢集效能資料 (如需詳細資訊，請參閱 檢視查詢歷史記錄資料)。
載入摘要	列出在特定期間的所有載入。可以根據查詢 ID、查詢執行時間和狀態等值排序清單。請在叢集詳細資訊頁面的 Query monitoring (查詢監控) 標籤中檢視此資料。
載入詳細資訊	提供包含以下項目之特定載入操作的詳細資訊： • 查詢 ID、類型、執行查詢的叢集和執行時間等載入屬性。 • 載入狀態和錯誤數之類的詳細資訊。 • 執行的 SQL 陳述式。 • 載入檔案清單。 • 載入操作期間的叢集效能資料 (如需詳細資訊，請參閱 檢視查詢歷史記錄資料)。

檢視效能資料

本節說明如何在 Amazon Redshift 主控台中檢視效能資料，包括叢集和查詢效能的資訊。此外，您可以透過 Amazon Redshift 主控台來直接建立對叢集指標的警示。

當您在 Amazon Redshift 主控台中檢視效能資料時，您可以依叢集來進行檢視。叢集的效能資料圖形旨在讓您存取可回答您最常見效能問題的資料。針對某些效能資料 (請參閱 Amazon Redshift 中的效能資料)，您也可以使用 CloudWatch 來進一步自訂您的指標圖表。例如，您可以選擇更長的時間，或是合併叢集的指標。如需使用 CloudWatch 主控台的相關資訊，請參閱CloudWatch 主控台中的效能指標。

觀看以下影片，了解如何使用 Amazon Redshift 主控台上的查詢監控功能來監控、隔離和優化您的查詢：Amazon Redshift 中的查詢監控。

主題

- 檢視叢集效能資料
- 檢視查詢歷史記錄資料
- 檢視資料庫效能資料

- 檢視工作負載並行和並行擴展資料
- 檢視自動最佳化資料
- 檢視查詢和載入
- 檢視和分析查詢詳細資訊
- 在查詢執行時檢視叢集效能
- 檢視載入操作期間的叢集指標
- 檢視叢集工作負載明細圖表

檢視叢集效能資料

使用 Amazon Redshift 中的叢集指標，您可執行以下常見的效能任務：

- 判斷叢集指標在特定時間範圍內是否不正常，以及若是不正常，則辨識影響效能的查詢。
- 確認歷史或目前查詢是否對叢集效能造成影響。如果您發現有問題的查詢，您可以檢視其相關詳細資訊，包含查詢執行期間的叢集效能。您可以使用此資訊來診斷查詢緩慢的原因以及可以執行的事項來改善其效能。

檢視效能資料

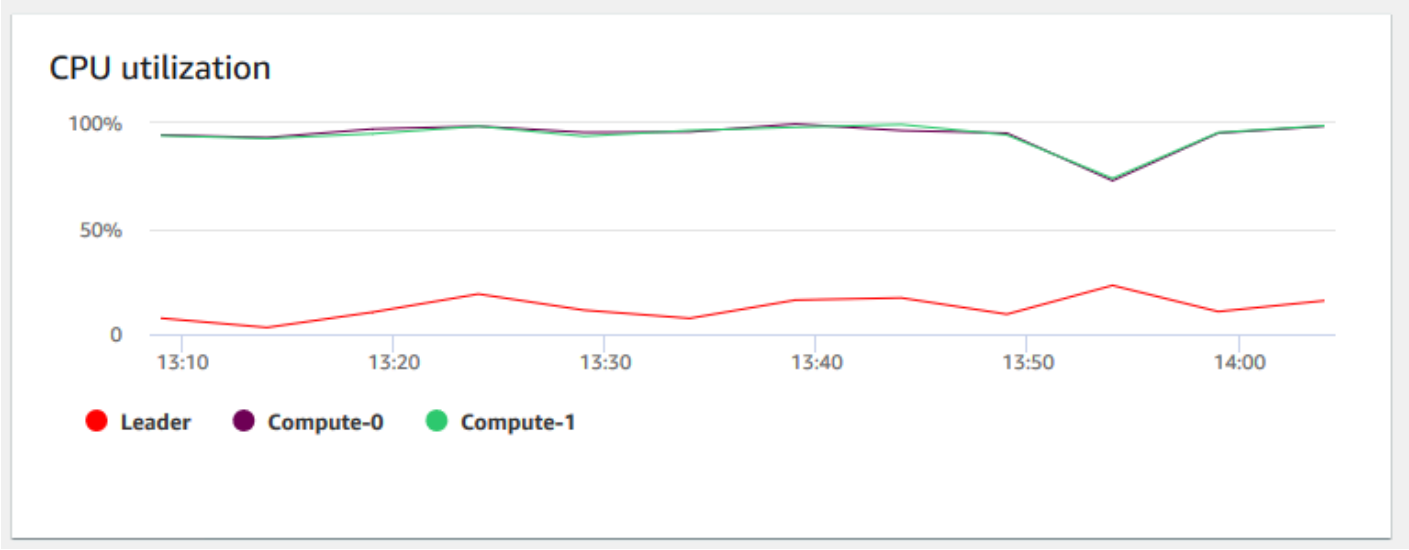
1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後從清單中選擇叢集名稱以開啟其詳細資訊。隨即顯示叢集的詳細資訊，包含叢集效能、查詢監控、資料庫、資料共用、排程、維護和屬性標籤。
3. 選擇 Cluster performance (叢集效能) 標籤，取得包括下列項目的效能資訊：
    - CPU 使用率
    - 使用的磁碟空間百分比
    - 資料庫連線
    - 運作狀態
    - 查詢期間
    - 查詢輸送量
    - Concurrency scaling activity (並行擴展活動)

另有更多指標可用。若要查看可用的指標並選擇要顯示的指標，請選擇 Preferences (偏好設定) 圖示。

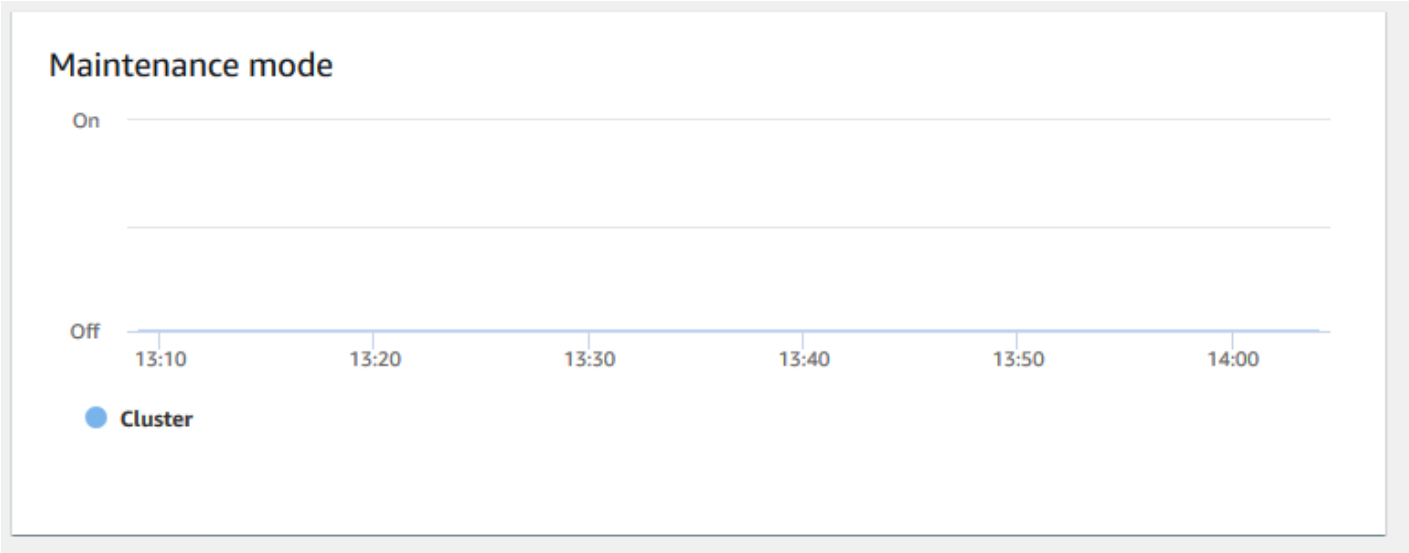
叢集效能圖表

下列範例顯示一些在全新 Amazon Redshift 主控台中顯示的圖表。

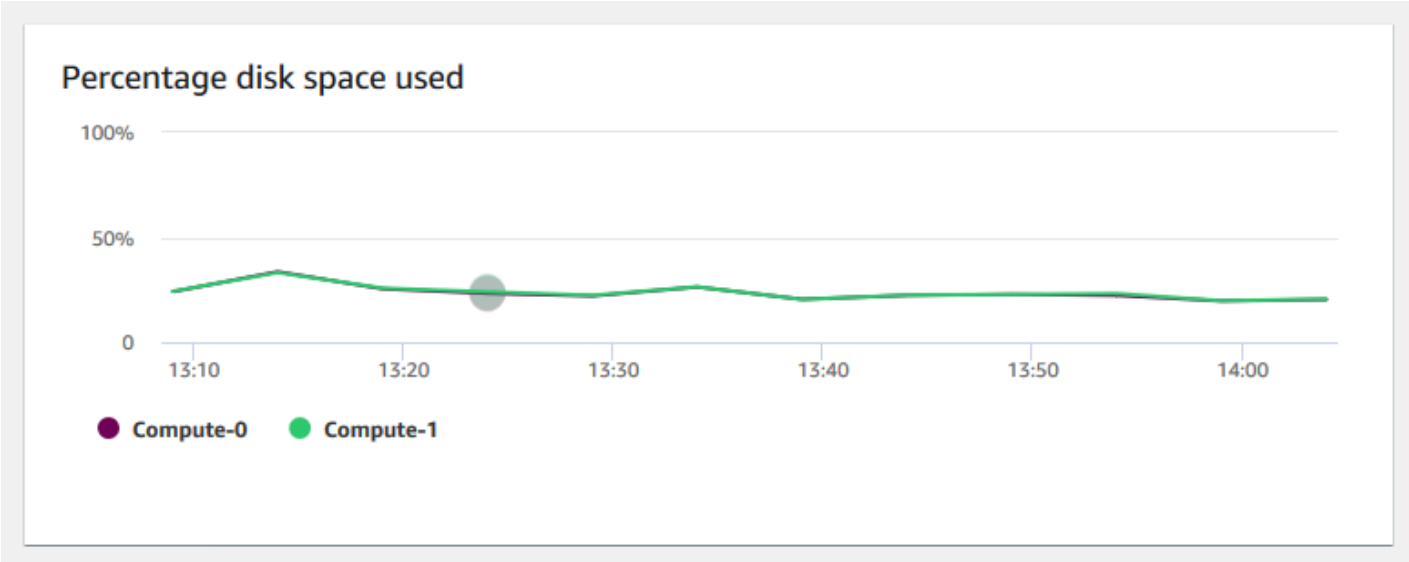
- CPU 使用率 — 顯示所有節點的 CPU 使用率百分比 (領導和運算)。如要在排程叢集遷移或其他資源使用操作前尋找叢集用量最低的時間，請監控此圖表來查看個別或所有節點的 CPU 使用率。



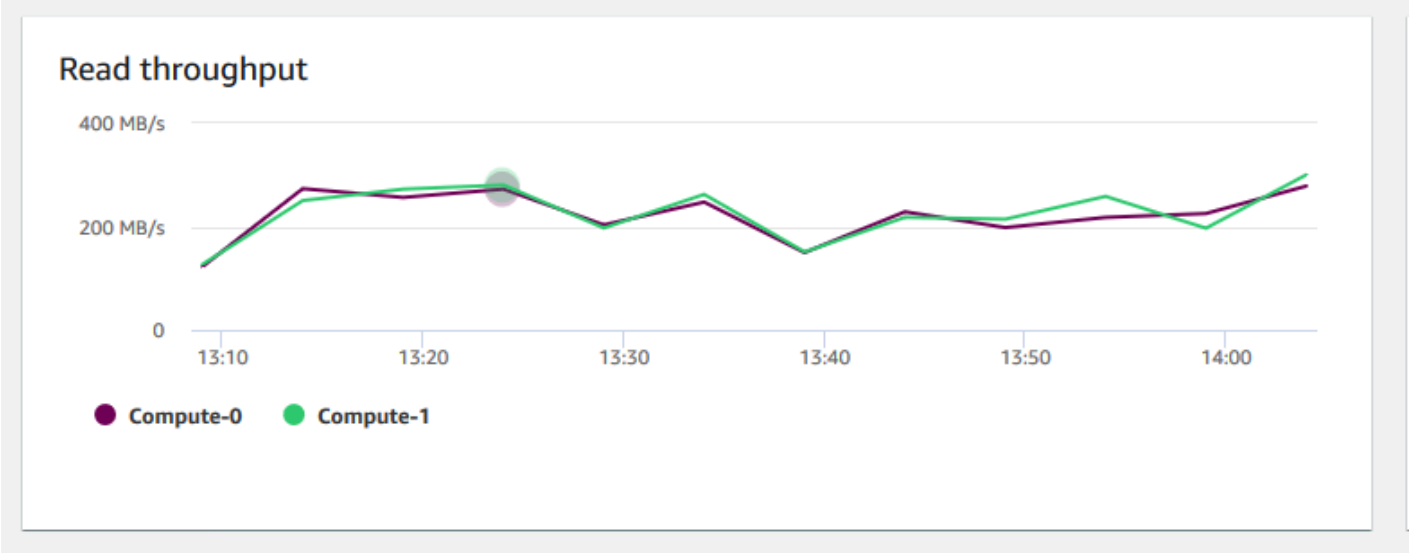
- 維護模式 — 使用 `On` 和 `Off` 指標，顯示叢集在指定時間是否處於維護模式。您可以查看叢集正在進行維護的時間。然後，您可以將此時間與叢集完成的操作產生關聯，以估計其未來在發生經常性事件時的停機時間。



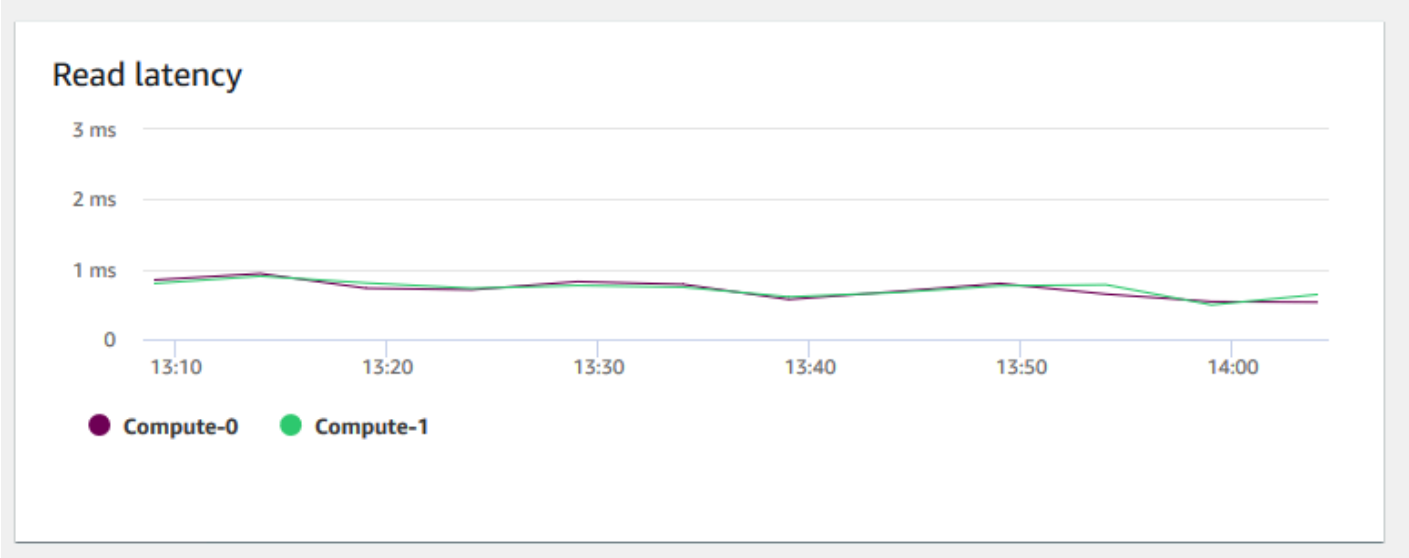
- 使用的磁碟空間百分比 — 顯示每個運算節點磁碟空間的用量百分比，而非叢集整體的用量。您可以探索此圖表來監控磁碟使用率。VACUUM 和 COPY 等維護操作會使用中繼暫時儲存空間來進行排序操作，因此預期會在磁碟用量中出現尖峰。



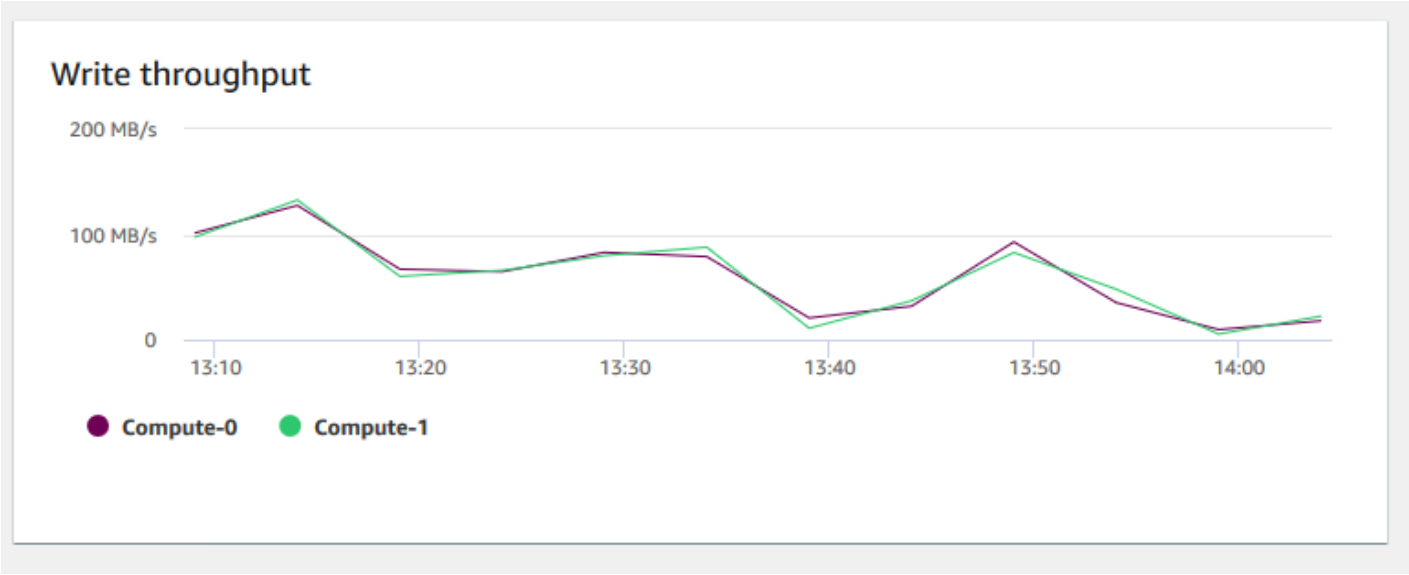
- 讀取輸送量 — 顯示每秒從磁碟讀取的平均 MB 數。您可以評估此圖表，監控叢集的對應實體方面。此輸送量不包含叢集中執行個體及其磁碟區之間的網路流量。



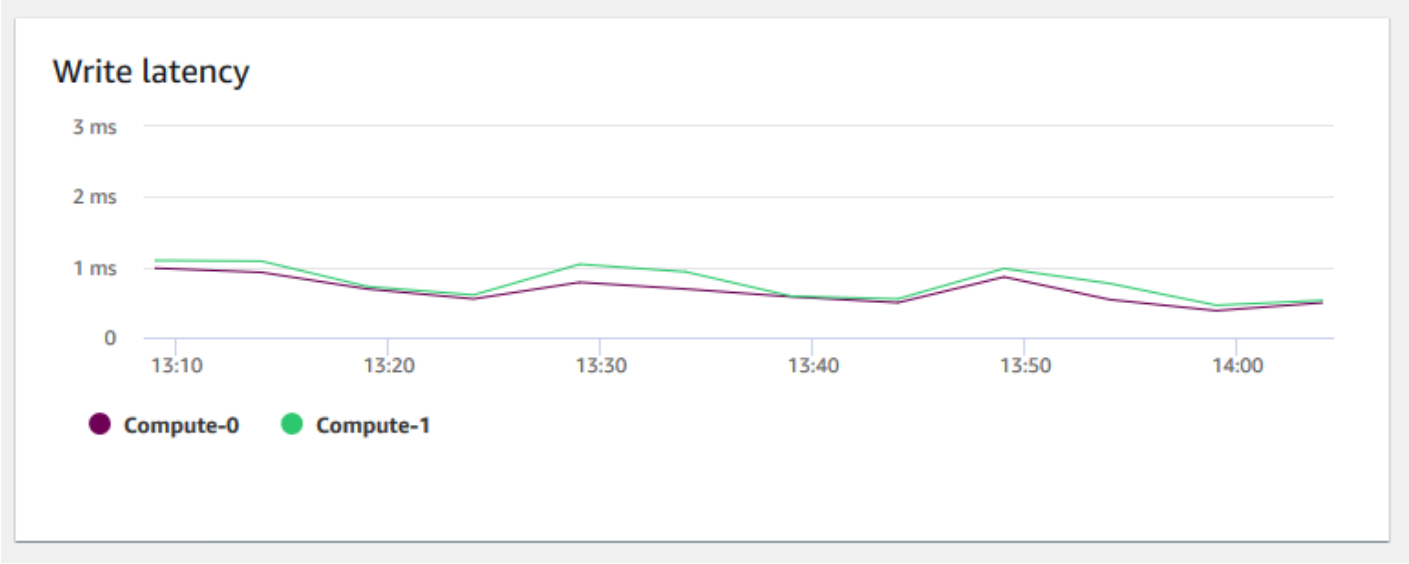
- 讀取延遲 — 顯示每毫秒磁碟讀取 I/O 操作所花費的平均時間長度。您可以檢視要傳回資料的回應時間。當延遲相當高時，這表示寄件者花費較多時間閒置 (而非傳送任何新的封包)，降低輸送量成長的速度。



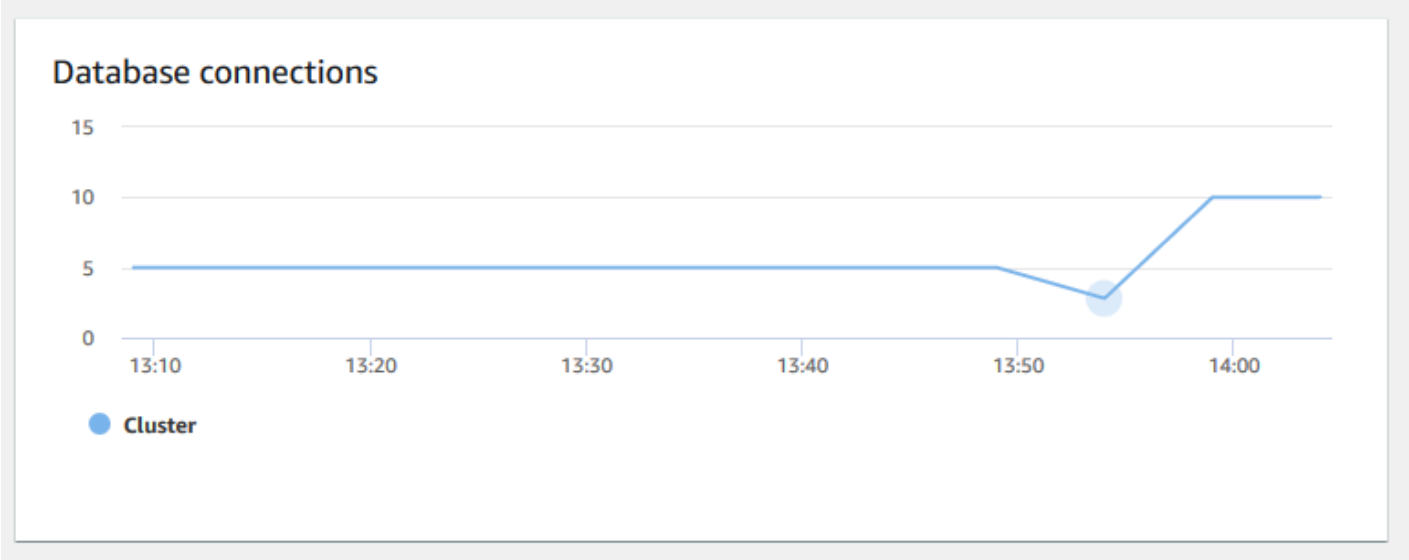
- 寫入輸送量 — 顯示每秒寫入磁碟的平均 MB 數。您可以評估此指標來監控叢集的對應實體方面。此輸送量不包含叢集中執行個體及其磁碟區之間的網路流量。



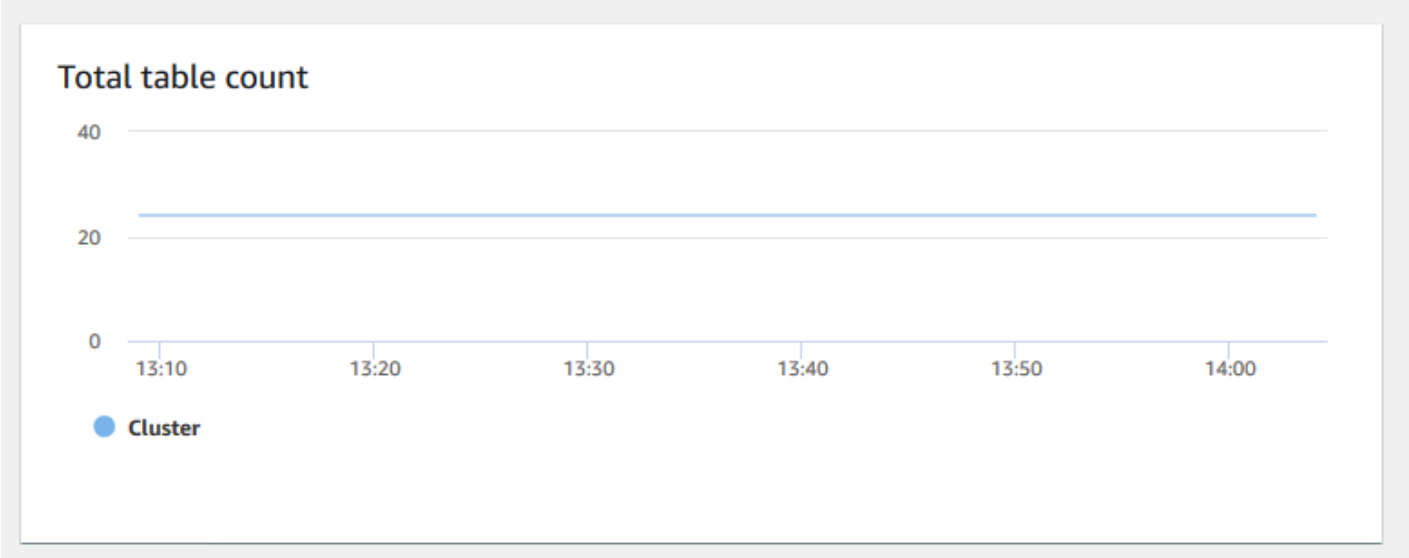
- 寫入延遲 — 顯示磁碟寫入 I/O 操作花費的平均時間長度 (毫秒)。您可以評估傳回寫入確認的時間。當延遲相當高時，這表示寄件者花費較多時間閒置 (而非傳送任何新的封包)，降低輸送量成長的速度。



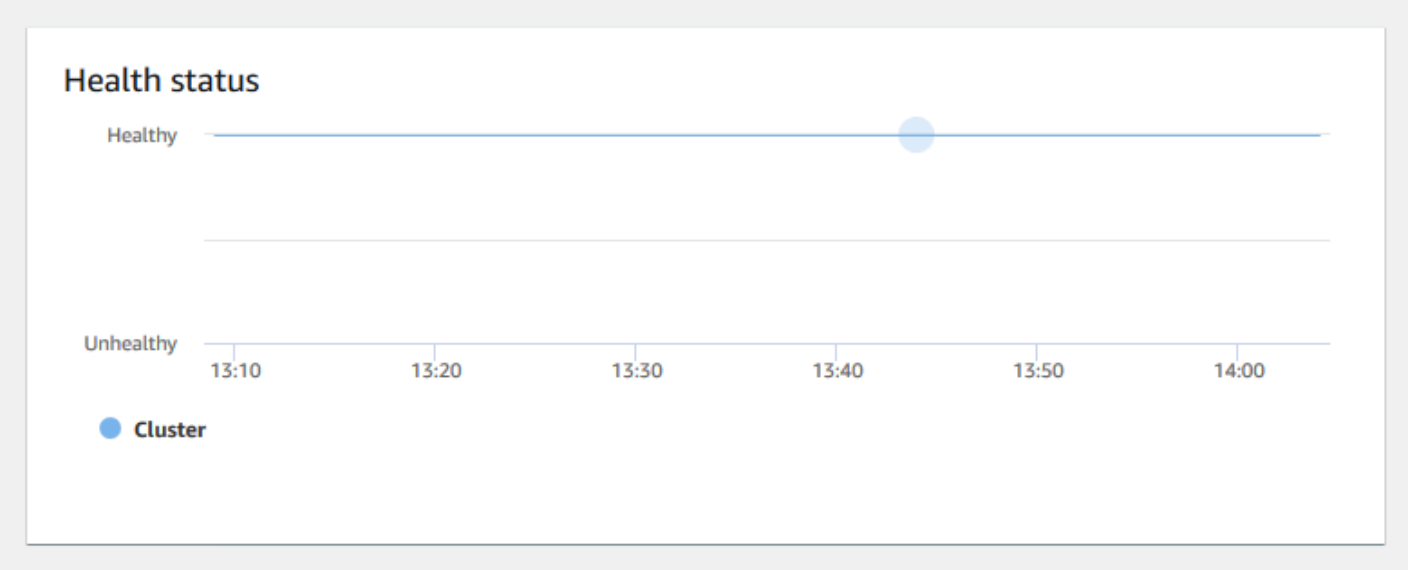
- 資料庫連線 — 顯示叢集的資料庫連線數。您可以使用此圖表來查看與資料庫建立的連線數，並尋找叢集用量最低的時間。



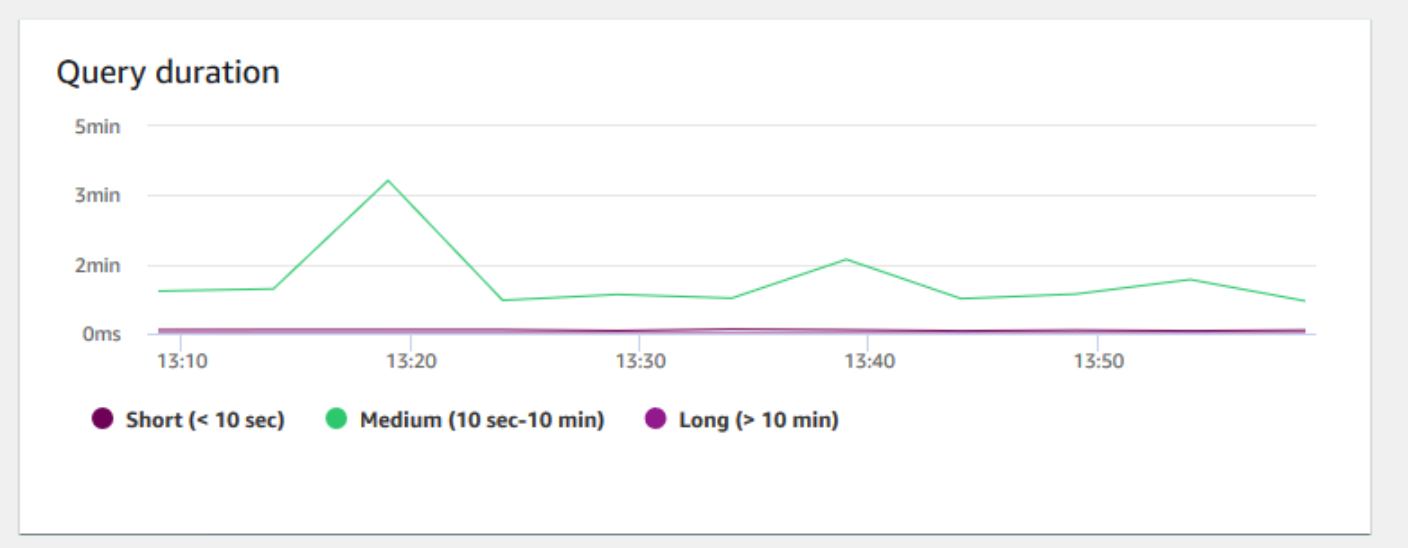
- 總資料表計數 — 顯示叢集內於特定時間點開啟的使用者資料表數。您可以監控開啟資料表計數較高時的叢集效能。



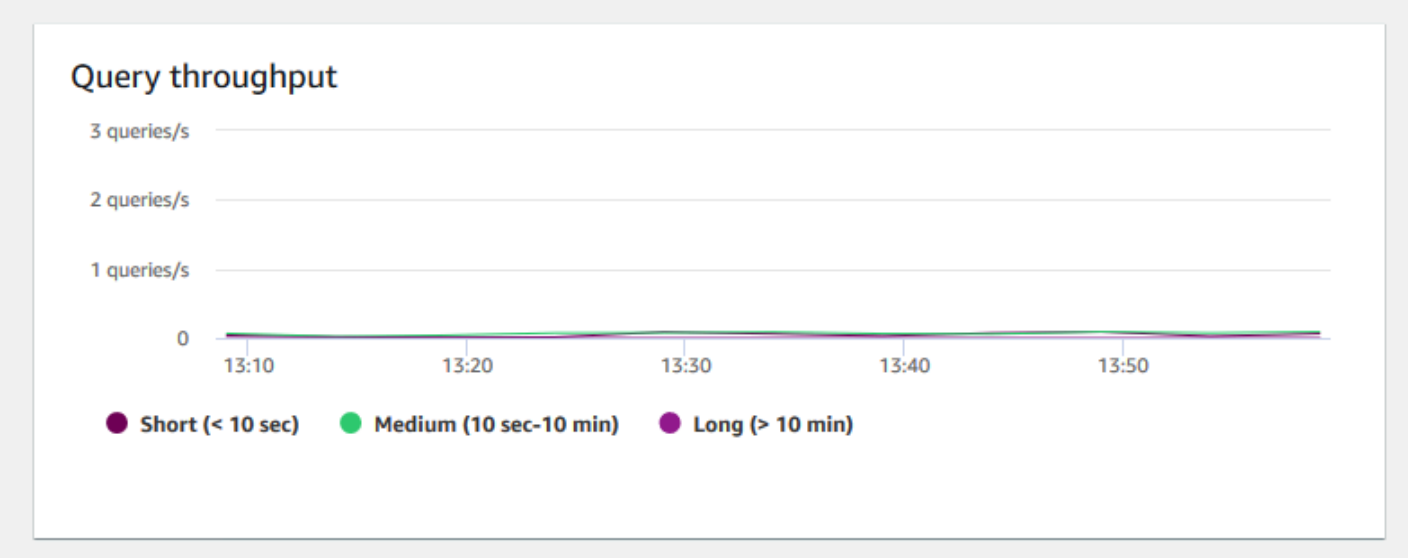
- 運作狀態 — 指出叢集的運作狀態為 Healthy 或 Unhealthy。如果叢集可以連線到其資料庫及成功執行簡易查詢，便會將叢集視為運作狀態良好。否則，表示該叢集運作不良。不良狀態可能會發生於資料庫處於極高負載，或叢集上的資料庫有組態的問題。



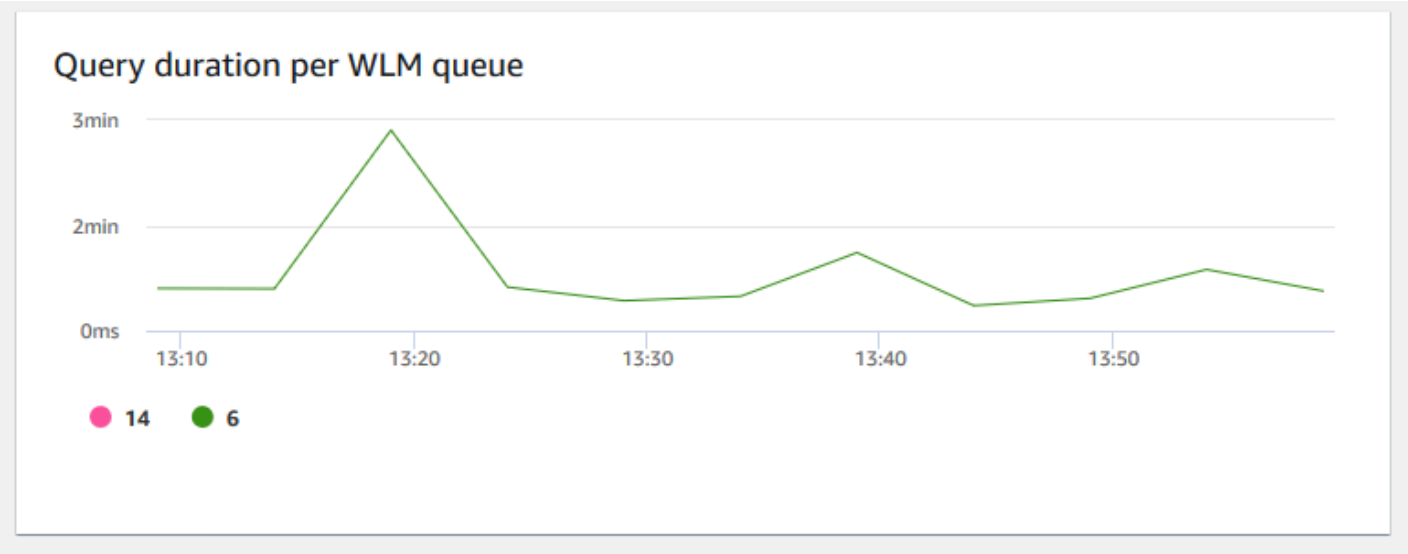
- 查詢持續時間 — 顯示完成查詢的平均時間長度 (微秒)。您可以針對此圖表上的資料進行基準測試，測量叢集內的 I/O 效能，並視需要調校最耗費時間的查詢。



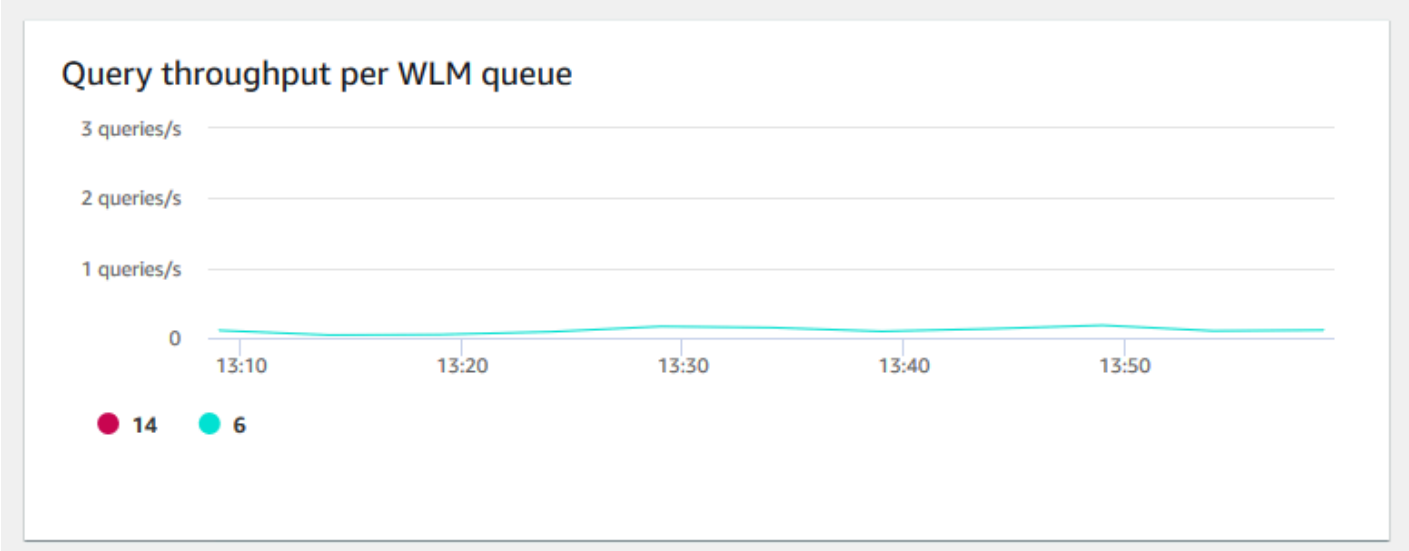
- 查詢輸送量 — 顯示每秒完成的平均查詢數。您可以分析此圖表上的資料，測量資料庫效能，並說明系統以平衡方式支援多使用者的能力。



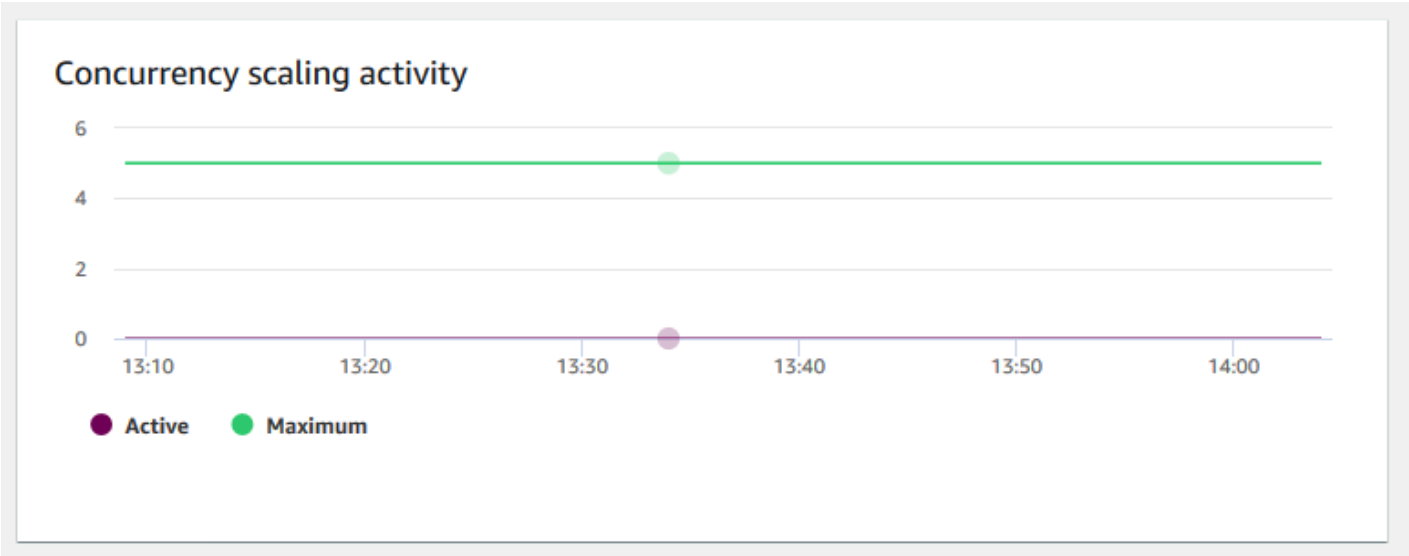
- 每個 WLM 佇列的查詢持續時間 — 顯示完成查詢的平均時間長度 (微秒)。您可以針對此圖表上的資料進行基準測試，測量每個 WLM 佇列的 I/O 效能，並視需要調校最耗費時間的查詢。



- 每個 WLM 佇列的查詢輸送量 — 顯示每秒完成的平均查詢數。您可以分析此圖表上的資料，測量每個 WLM 佇列的資料庫效能。



- 並行擴展活動 — 顯示作用中並行擴展叢集的數量。當並行擴展啟用時，當您需要更多叢集容量以執行增加的並行讀取查詢時，Amazon Redshift 將會自動新增額外的叢集容量。



檢視查詢歷史記錄資料

您可以在 Amazon Redshift 中使用查詢歷史記錄指標來執行以下作業：

- 隔離和診斷查詢效能問題。
- 比較相同時間軸上的查詢執行時間指標和叢集效能指標，查看兩個指標可能的相關情況。您可以透過這麼做來找出執行狀況不佳的查詢、尋找瓶頸查詢並判斷您是否需要重新調整工作負載的叢集大小。

- 在時間軸上選擇，以向下切入特定查詢的詳細資訊。當 Query ID (查詢 ID) 和其他屬性顯示在圖表下方的資料列時，您可以選擇查詢來查看查詢詳細資訊。詳細資訊包括查詢的 SQL 陳述式、執行詳細資訊和查詢計畫。如需詳細資訊，請參閱檢視和分析查詢詳細資訊。
- 判斷您的載入任務是否成功完成，以及滿足您的服務水準協議 (SLA)。

顯示查詢歷史記錄資料

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ ：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後從清單中選擇叢集名稱以開啟其詳細資訊。叢集的詳細資訊隨即顯示，包含叢集效能、查詢監控、資料庫、資料共用、排程、維護和屬性索引標籤。
3. 選擇與查詢有關的指標 Query monitoring (查詢監控) 標籤。
4. 在 Query monitoring (查詢監控) 區段中，選擇 Query history (查詢歷史記錄) 標籤。

   使用視窗上的控制項，您可以在 Query list (查詢清單) 和 Cluster metrics (叢集指標) 之間進行切換。

   當您選擇 Query list (查詢清單) 時，標籤會包含下列圖表：

   - 查詢執行時間 — 時間軸上的查詢活動。使用此圖表以查看在相同時間範圍內執行的查詢。選擇查詢以檢視更多查詢執行詳細資訊。x 軸會顯示選取的期間。您可以透過正在執行、已完成、載入等篩選圖表化查詢。每個長條都代表一個查詢，且長條的長度代表其從長條開始到結束的執行時間。查詢可以包含 SQL 資料處理陳述式 (例如 SELECT、INSERT、DELETE) 和載入 (例如 COPY)。根據預設，會顯示所選取時間期間中前 100 個執行時間最長的查詢。
   - 查詢和載入 — 在該叢集上執行的查詢和載入清單。如果查詢目前正在執行中，視窗便會包含 Terminate query (終止查詢) 選項。

   當您選擇 Cluster metrics (叢集指標) 時，標籤會包含下列圖表：

   - 查詢執行時間 — 時間軸上的查詢活動。使用此圖表以查看在相同時間範圍內執行的查詢。選擇查詢以檢視更多查詢執行詳細資訊。
   - CPU 使用率 — 領導節點和運算節點平均的叢集 CPU 使用率。
   - 已使用的儲存體容量 — 已使用的儲存體容量百分比。
   - 作用中資料庫連線 — 叢集上作用中的資料庫連線數。

請在使用查詢歷史記錄圖表時，請考慮下列事項：

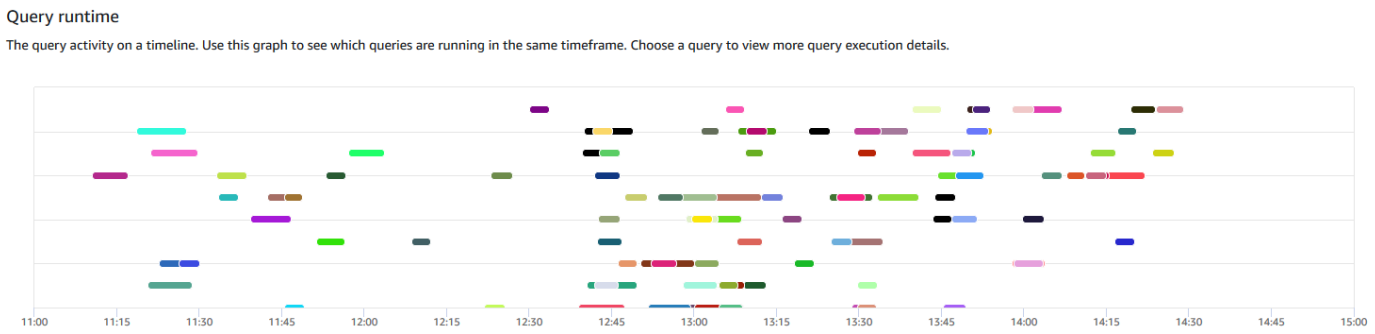
- 在 Query runtime (查詢執行時間) 圖表上選擇代表特定查詢的橫條，以查看該查詢的詳細資訊。您也可以在 Queries and loads (查詢和載入) 清單上選擇查詢 ID 來查看其詳細資訊。
- 您可以滑動以選取 Query runtime (查詢執行時間) 圖表的區段，以放大顯示特定時間期間。
- 在 Query runtime (查詢執行時間) 圖表上，如要查看您所選擇篩選條件考慮的所有資料，請向前翻到所有列在 Queries and loads (查詢和載入) 清單上的頁面。
- 您可以使用顯示在 settings gear icon (設定齒輪圖示) 旁邊的偏好設定視窗，變更在 Queries and loads (查詢和載入) 清單上顯示的資料列數和欄位。
- 您也可以從左側導覽器的查詢圖示、查詢和載入進行瀏覽，以顯示查詢和載入清單。如需詳細資訊，請參閱檢視查詢和載入。

查詢歷史記錄圖表

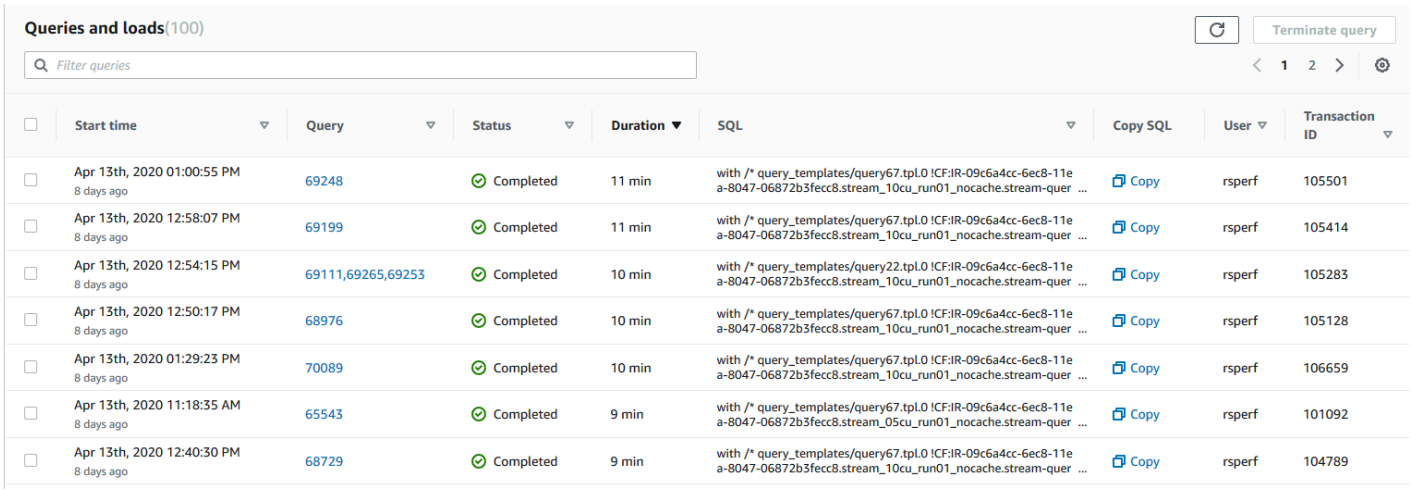
下列範例顯示在全新 Amazon Redshift 主控台中顯示的圖表。

> **Note**
>
> Amazon Redshift 主控台圖表只包含最近 100,000 筆查詢的資料。

- Query runtime (查詢執行時間)



- Queries and loads (查詢和載入)



Queries and loads(100)

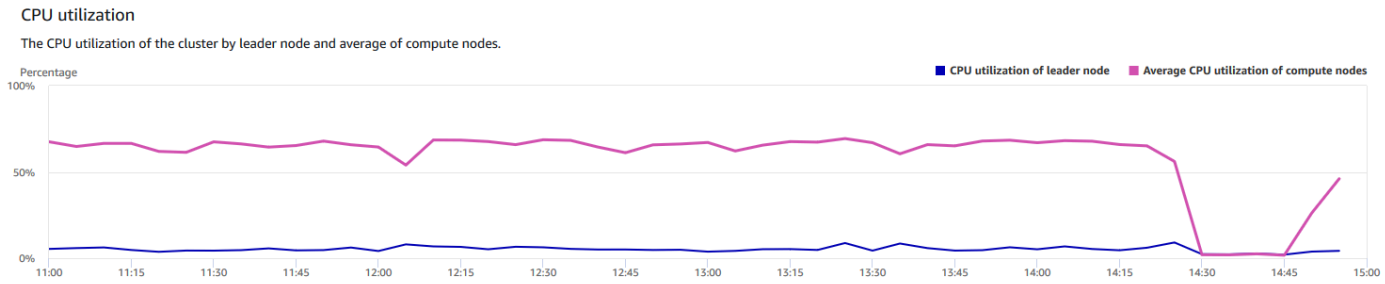
Filter queries

Terminate query

1 2

Start time	Query	Status	Duration	SQL	Copy SQL	User	Transaction ID
Apr 13th, 2020 01:00:55 PM 8 days ago	69248	Completed	11 min	with /* query_templates/query67.tpl.0 !CF:IR-09c6a4cc-6ec8-11ea-8047-06872b3fecc8.stream_10cu_run01_nocache.stream-quer ...	Copy	rsperf	105501
Apr 13th, 2020 12:58:07 PM 8 days ago	69199	Completed	11 min	with /* query_templates/query67.tpl.0 !CF:IR-09c6a4cc-6ec8-11ea-8047-06872b3fecc8.stream_10cu_run01_nocache.stream-quer ...	Copy	rsperf	105414
Apr 13th, 2020 12:54:15 PM 8 days ago	69111,69265,69253	Completed	10 min	with /* query_templates/query22.tpl.0 !CF:IR-09c6a4cc-6ec8-11ea-8047-06872b3fecc8.stream_10cu_run01_nocache.stream-quer ...	Copy	rsperf	105283
Apr 13th, 2020 12:50:17 PM 8 days ago	68976	Completed	10 min	with /* query_templates/query67.tpl.0 !CF:IR-09c6a4cc-6ec8-11ea-8047-06872b3fecc8.stream_10cu_run01_nocache.stream-quer ...	Copy	rsperf	105128
Apr 13th, 2020 01:29:23 PM 8 days ago	70089	Completed	10 min	with /* query_templates/query67.tpl.0 !CF:IR-09c6a4cc-6ec8-11ea-8047-06872b3fecc8.stream_10cu_run01_nocache.stream-quer ...	Copy	rsperf	106659
Apr 13th, 2020 11:18:35 AM 8 days ago	65543	Completed	9 min	with /* query_templates/query67.tpl.0 !CF:IR-09c6a4cc-6ec8-11ea-8047-06872b3fecc8.stream_05cu_run01_nocache.stream-quer ...	Copy	rsperf	101092
Apr 13th, 2020 12:40:30 PM 8 days ago	68729	Completed	9 min	with /* query_templates/query67.tpl.0 !CF:IR-09c6a4cc-6ec8-11ea-8047-06872b3fecc8.stream_10cu_run01_nocache.stream-quer ...	Copy	rsperf	104789

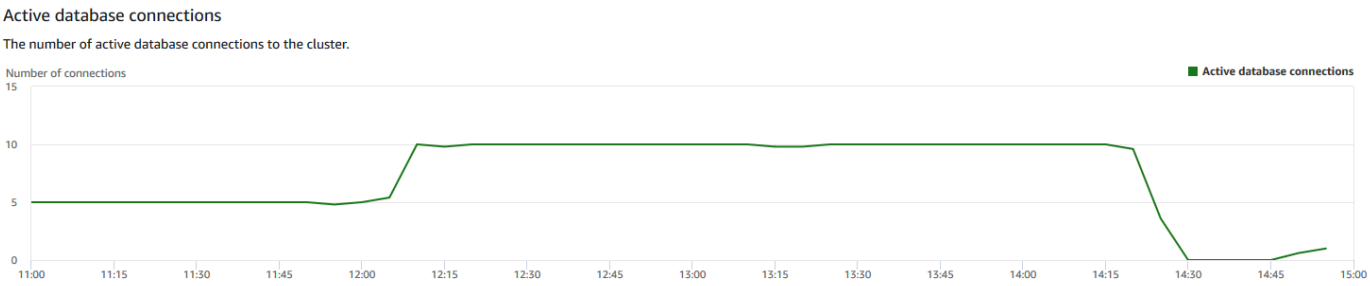
- CPU 使用率



- Storage capacity used (已使用的儲存體容量)



- Active database connections (作用中的資料庫連線)



檢視資料庫效能資料

您可以使用 Amazon Redshift 中的資料庫效能指標來執行以下作業：

- 透過處理階段來分析查詢所花費的時間。您可以尋找在某階段所花費時間中不尋常的趨勢。
- 根據持續時間範圍 (短、中、長) 來分析查詢數、持續時間和查詢的輸送量。
- 根據查詢優先順序 (最低、低、一般、高、最高、關鍵) 尋找查詢等待時間中的趨勢。
- 尋找查詢持續時間、輸送量，或是 WLM 佇列等待時間中的趨勢。

顯示資料庫效能資料

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後從清單中選擇叢集名稱以開啟其詳細資訊。隨即顯示叢集的詳細資訊，包含叢集效能、查詢監控、資料庫、資料共用、排程、維護和屬性標籤。
3. 選擇與查詢有關的指標 Query monitoring (查詢監控) 標籤。
4. 在 Query monitoring (查詢監控) 區段中，選擇 Database performance (資料庫效能) 標籤。

   使用視窗上的控制項，您可以在 Cluster metrics (叢集指標) 和 WLM queue metrics (WLM 佇列指標) 之間進行切換。

   當您選擇 Cluster metrics (叢集指標) 時，標籤會包含下列圖表：

   - 工作負載執行明細項目 — 在查詢處理階段所耗費的時間。
   - 查詢 (依持續時間範圍) — 短、中、長時間的查詢數。
   - 查詢輸送量 — 每秒完成的平均查詢數。
   - 查詢持續時間 — 完成查詢的平均時間長度。
   - 平均佇列等待時間 (依優先順序) — 查詢在 WLM 佇列中所耗費的總時間 (依查詢優先順序)。

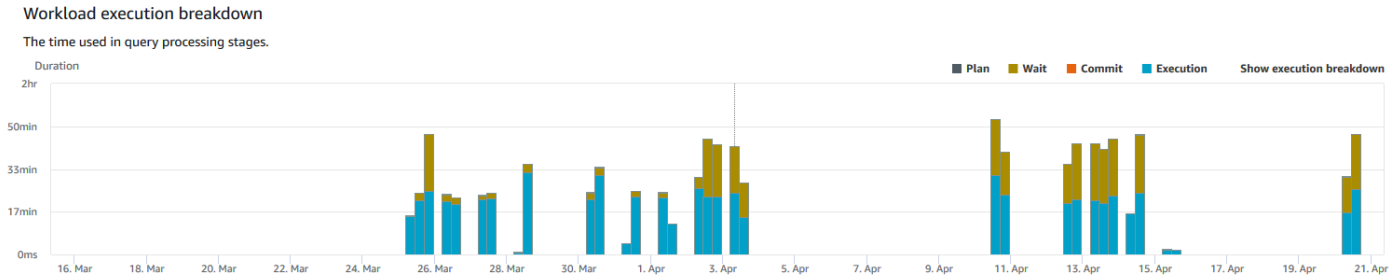
   當您選擇 WLM queue metrics (WLM 佇列指標) 時，標籤會包含下列圖表：

   - 查詢持續時間 (依佇列) — WLM 佇列的平均查詢持續時間。
   - 查詢輸送量 (依佇列) — WLM 佇列每秒完成的平均查詢數。
   - 查詢等待時間 (依佇列) — 查詢在 WLM 佇列耗費的平均持續時間。

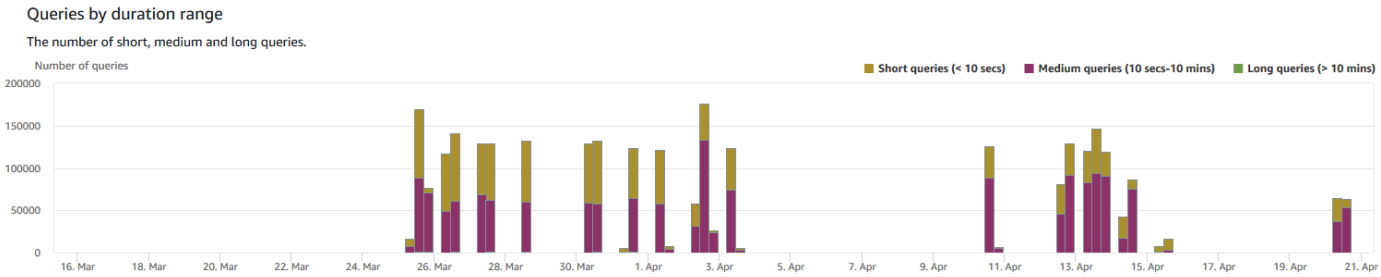
資料庫效能圖表

下列範例顯示在全新 Amazon Redshift 主控台中顯示的圖表。

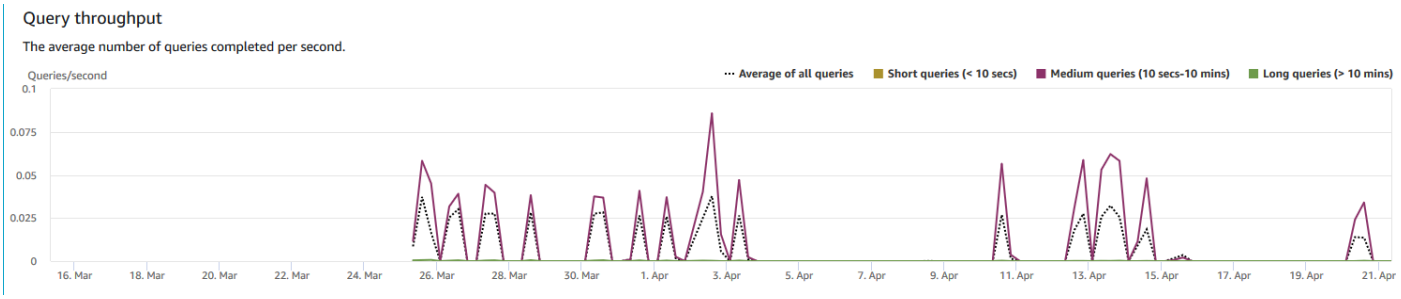
- Workload execution breakdown (工作負載執行明細項目)



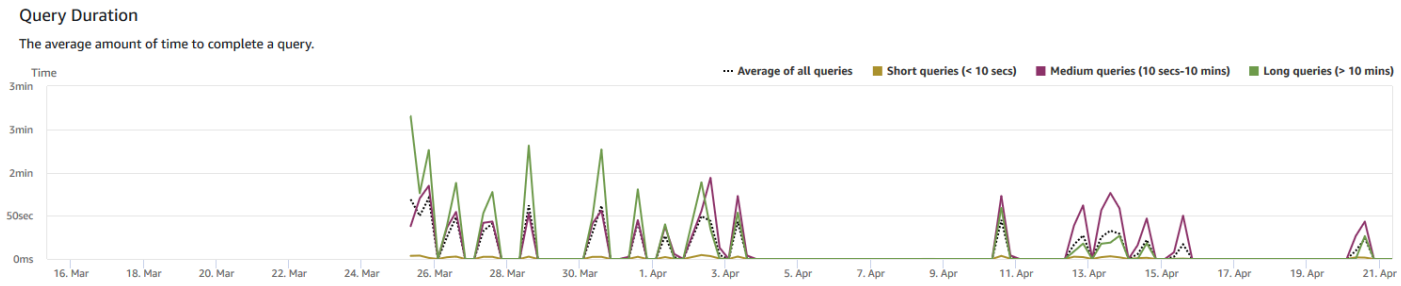
- Queries by duration range (查詢 (依持續時間範圍))



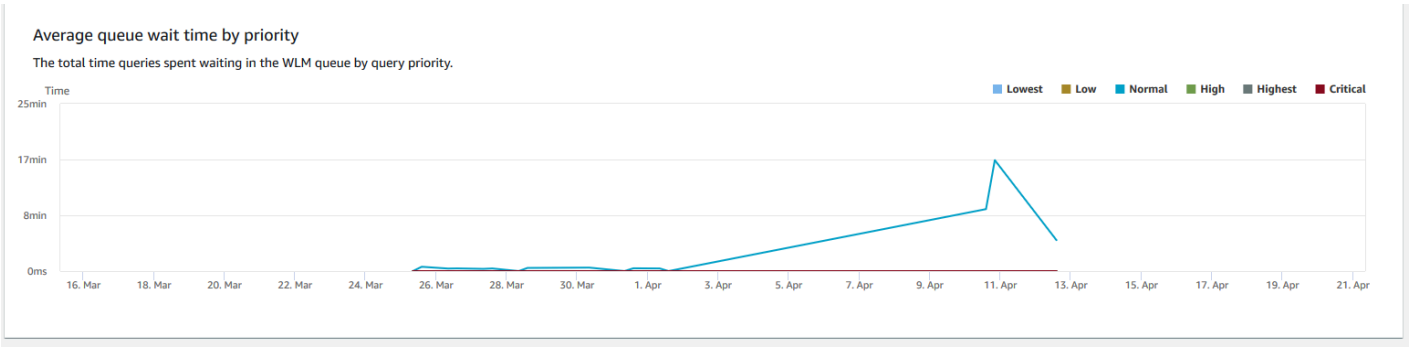
- 查詢輸送量



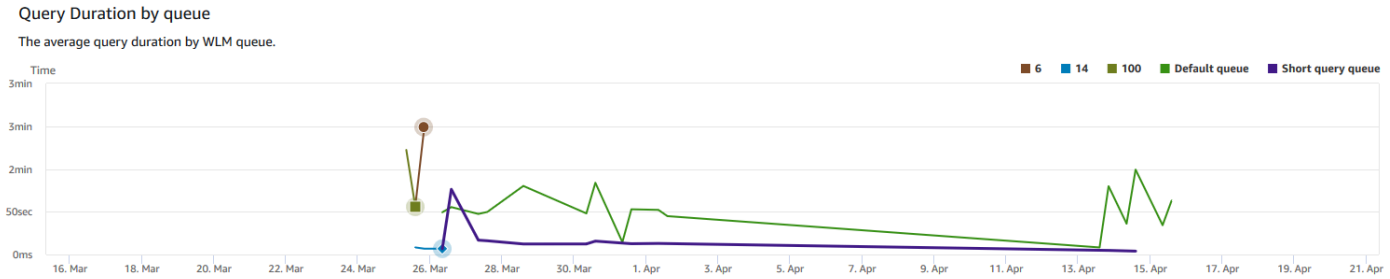
- 查詢期間



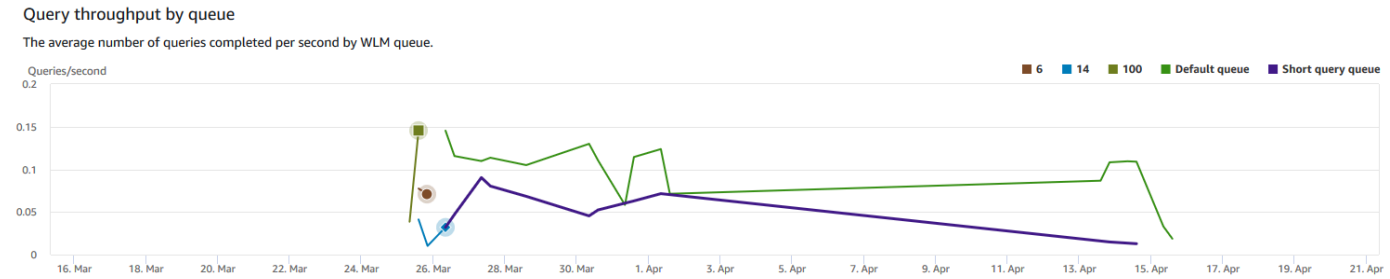
- Average queue wait time by priority (平均佇列等待時間 (依優先順序))



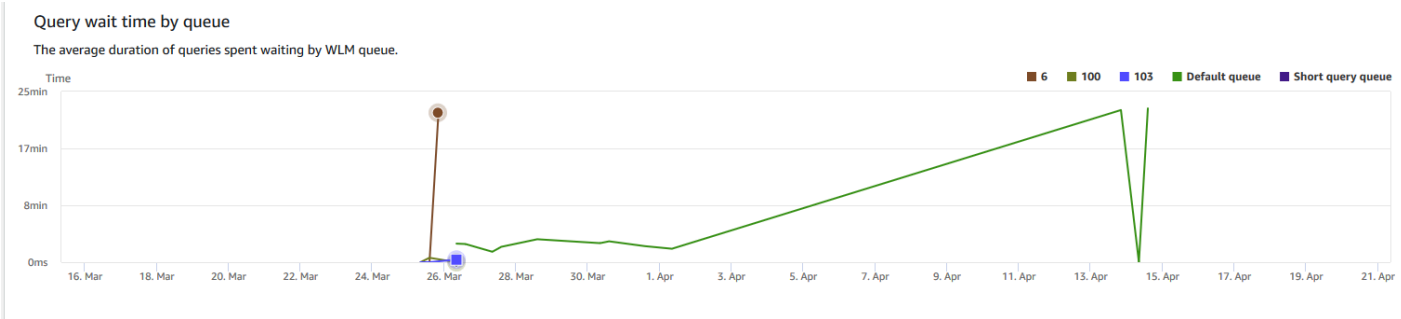
- Query duration by queue (查詢持續時間 (依佇列))



- Query throughput by queue (查詢輸送量 (依佇列))



- Query wait time by queue (查詢等待時間 (依佇列))



檢視工作負載並行和並行擴展資料

透過在 Amazon Redshift 中使用並行擴展指標，您可以執行以下操作：

- 分析您是否可以透過啟用並行擴展來降低排入佇列的查詢數。您可以依 WLM 佇列進行比較，或是針對所有 WLM 佇列進行比較。
- 檢視並行擴展叢集中的並行擴展活動。如此可讓您了解並行擴展是否受限於 `max_concurrency_scaling_clusters`。如果是，您可以選擇提高 DB 參數中的 `max_concurrency_scaling_clusters`。
- 檢視所有並行擴展叢集合計的並行擴展總用量。

顯示並行擴展資料

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ ：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後從清單中選擇叢集名稱以開啟其詳細資訊。叢集的詳細資訊隨即顯示，包含叢集效能、查詢監控、資料庫、資料共用、排程、維護和屬性索引標籤。
3. 選擇與查詢有關的指標 Query monitoring (查詢監控) 標籤。
4. 在 Query monitoring (查詢監控) 區段中，選擇 Workload concurrency (工作負載並行) 標籤。

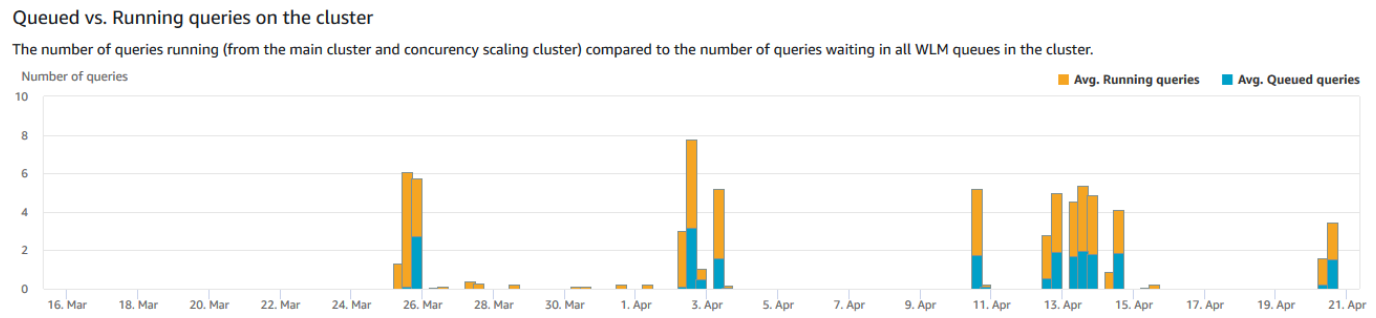
   標籤包括下列圖表：

   - 排入佇列的查詢和正在叢集上執行的查詢 — 正在執行查詢數 (來自主要叢集和並行擴展叢集) 和叢集中所有 WLM 佇列內正在等待查詢數的比較。
   - 每個佇列上排入佇列和正在執行的查詢 — 正在執行查詢數 (來自主要叢集和並行擴展叢集) 和正在每個 WLM 佇列中等待查詢數的比較。
   - 並行擴展活動 — 正在處理查詢的並行擴展叢集數。
   - 並行擴展用量 — 具備作用中查詢處理活動的並行擴展叢集用量。

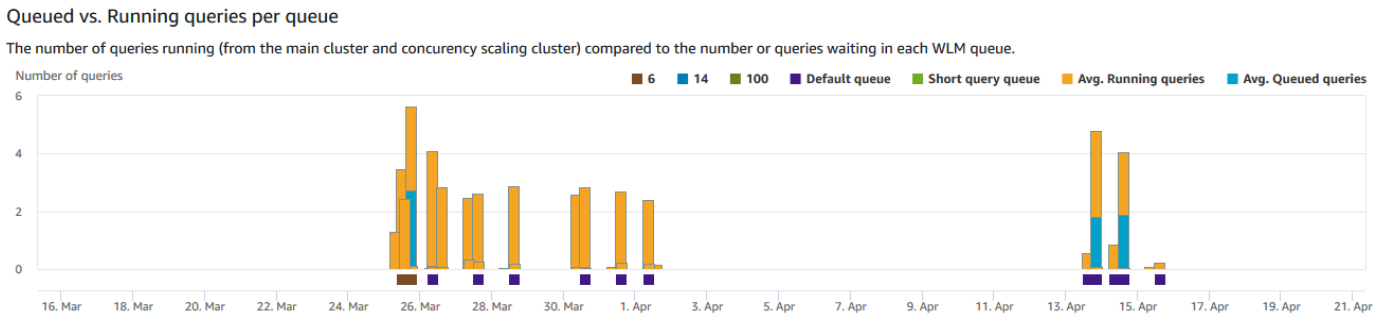
工作負載並行圖表

下列範例顯示在全新 Amazon Redshift 主控台中顯示的圖表。若要在 Amazon CloudWatch 中建立類似圖表，您可以使用並行擴展和 WLM CloudWatch 指標。如需 Amazon Redshift 的 CloudWatch 指標的相關資訊，請參閱「Amazon Redshift 中的效能資料」。

- 排入佇列的查詢和正在叢集上執行的查詢



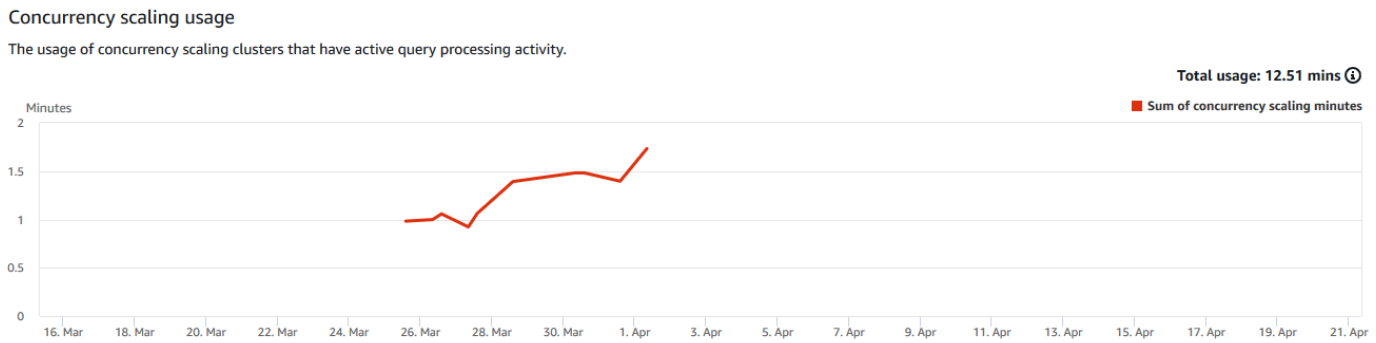
- 每個佇列上排入佇列和正在執行的查詢



- Concurrency scaling activity (並行擴展活動)



- 並行擴展模式



檢視自動最佳化資料

Amazon Redshift 主控台提供使用額外運算資源執行自動最佳化或自主運算的相關資訊。您可以使用此資訊來追蹤用量，並監控是否已達到用量限制。雖然 Amazon Redshift 不會針對佈建叢集本身上執行的自律程式向您收費，但會針對使用額外運算資源執行的自律程式向您收費。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的為自動資料庫最佳化配置額外的運算資源。

檢視額外的運算自律資料

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後從清單中選擇叢集名稱以開啟其詳細資訊。
3. 從叢集的詳細資訊頁面，從動作下拉式功能表中選取管理用量限制。您也可以選取叢集的維護索引標籤，然後向下捲動並選取建立用量限制。
4. 顯示額外運算自律資料的圖表會顯示在標題為額外運算用量限制的區段下方，以進行自動最佳化。圖表顯示 Amazon Redshift 在指定期間內使用額外運算資源執行自主計算的時間量。

檢視查詢和載入

Amazon Redshift 主控台提供在資料庫中執行的查詢和載入資訊。您可以使用此資訊來找出花長時間處理的查詢，以及對其他查詢造成瓶頸而無法有效處理的查詢，並對其進行故障診斷。您可在 Amazon Redshift 主控台中使用查詢資訊監控查詢處理過程。

顯示查詢效能資料

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇查詢和載入以顯示帳戶的查詢清單。

   根據預設，此清單會顯示過去 24 小時您所有叢集的查詢。您可在主控台中變更顯示日期的範圍。

> **Important**
>
> Queries and loads (查詢和負載) 清單顯示系統中執行時間最長的查詢，最多 100 個查詢。

檢視和分析查詢詳細資訊

您可以使用查詢識別碼檢視查詢的詳細資訊。詳細資訊可以包括查詢的完成狀態、持續時間、SQL 陳述式，以及它是否為使用者查詢或由 Amazon Redshift 重寫的查詢。使用者查詢是從 SQL 用戶端提交至 Amazon Redshift 的查詢，或是由商業智慧工具產生的查詢。Amazon Redshift 可能會重寫查詢以進行最佳化，這可能會導致多個重寫的查詢。雖然程序是由 Amazon Redshift 完成的，但您會在查詢詳細資訊頁面上同時看到這類查詢和使用者查詢。

檢視查詢

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇查詢和載入以顯示帳戶的查詢清單。您可能需要變更此頁面的設定，才能找到您的查詢。
3. 在清單中選擇 Query (查詢) 識別碼，以顯示 Query details (查詢詳細資訊)。

   Query details (查詢詳細資訊) 頁面包括附有查詢指標的 Query details (查詢詳細資訊) 和 Query plan (查詢計畫) 標籤。

   指標包括查詢的詳細資訊，例如開始時間、查詢 ID、狀態和持續時間。其他詳細資訊包括查詢是否在主叢集或並行擴展叢集上執行，以及它是否為父系或重新寫入的查詢。

在查詢執行時檢視叢集效能

您可以在查詢執行時監控叢集的效能，以找出潛在的瓶頸並最佳化查詢執行。在查詢執行時檢視叢集效能，可提供系統層級指標的即時檢視，例如 CPU 使用率、磁碟 I/O 和網路流量，以及執行時間、處理的資料量和查詢步驟等查詢層級的詳細資訊。下列程序會引導您存取和解譯效能指標，以有效管理和最佳化佈建叢集。

在查詢執行時顯示叢集效能

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後從清單中選擇叢集名稱以開啟其詳細資訊。叢集的詳細資訊隨即顯示，包含叢集效能、查詢監控、資料庫、資料共用、排程、維護和屬性索引標籤。
3. 選擇 Query monitoring (查詢監控) 標籤以取得詳細資訊。

   如需詳細資訊，請參閱檢視查詢歷史記錄資料。

檢視載入操作期間的叢集指標

檢視載入操作期間叢集效能時，您可以辨識正在耗用資源的查詢並採取動作來降低它們的影響。如果您不希望某個載入的執行完成，可以將其終止。

> **Note**
>
> 在 Amazon Redshift 主控台終止查詢和載入的功能需要特定的許可。如果您希望使用者能夠終止查詢和載入，請務必將 `redshift:CancelQuerySession`動作新增至您的 AWS Identity and Access Management (IAM) 政策。無論您選取 Amazon Redshift 唯讀 AWS受管政策或在 IAM 中建立自訂政策，此要求都適用。具有 Amazon Redshift 完整存取政策之使用者已擁有必要的許可，可終止查詢和載入。如需 Amazon Redshift 之 IAM 政策中動作的相關資訊，請參閱管理 資源的存取。

顯示載入操作期間的叢集效能

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ ：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後從清單中選擇叢集名稱以開啟其詳細資訊。叢集的詳細資訊隨即顯示，包含叢集效能、查詢監控、資料庫、資料共用、排程、維護和屬性索引標籤。
3. 選擇 Query monitoring (查詢監控) 標籤以取得詳細資訊。
4. 在 Queries and loads (查詢和載入) 區段中，選擇 Loads (載入) 檢視叢集的載入操作。如果載入正在執行，您可選擇 Terminate query (終止查詢) 以終止此作業。

檢視叢集工作負載明細圖表

您可以取得工作負載效能的詳細檢視，方法是查看主控台中的工作負載執行明細項目圖表。我們以 QueryRuntimeBreakdown 指標提供的資料來建立圖表。利用此圖表，您可以查看您的查詢花費各種處理階段 (例如等候和規劃) 的時間。

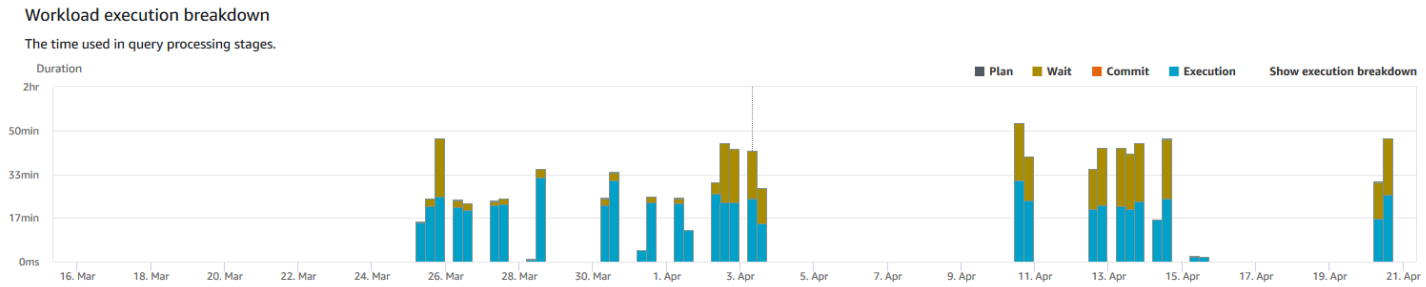
> **Note**
>
> 工作負載執行明細項目圖表不會對單一節點叢集顯示。

以下指標清單說明各種處理階段：

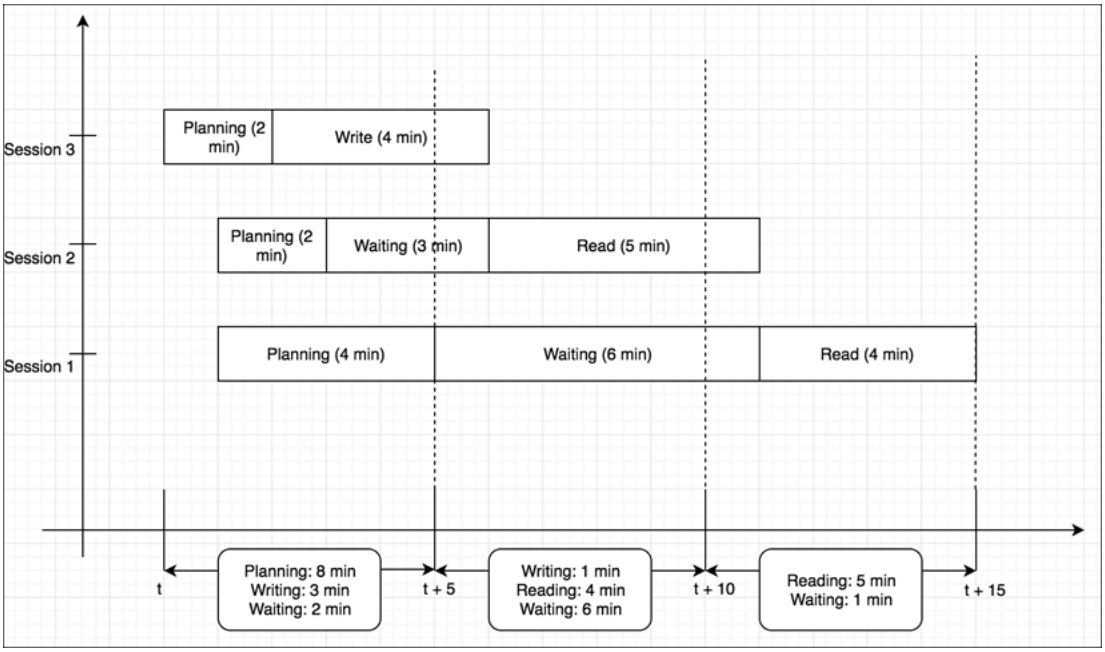
- QueryPlanning：剖析與最佳化 SQL 陳述式花費的時間。
- QueryWaiting：花費在工作負載管理 (WLM) 佇列中等待的時間。
- QueryExecutingRead：執行讀取查詢所花費的時間。
- QueryExecutingInsert：執行插入查詢所花費的時間。
- QueryExecutingDelete：執行刪除查詢所花費的時間。
- QueryExecutingUpdate：執行更新查詢所花費的時間。
- QueryExecutingCtas：執行 CREATE TABLE AS 查詢所花費的時間。
- QueryExecutingUnload：執行卸載查詢所花費的時間。
- QueryExecutingCopy：執行複製查詢所花費的時間。

例如，在 Amazon Redshift 主控台的下列圖示表示查詢在計畫、等待、讀取和寫入階段中所花的時間。您可以將此圖形中的發現與其他指標相結合，以進一步分析。在某些情況下，您的圖表可能顯示較短期間的查詢 (量測方式為 QueryDuration 指標) 在等待階段的所花費時間較長。在這些情況下，您可以為特定佇列增加 WLM 並行率，以增加傳輸量。

以下是工作負載執行細分圖表的範例。在圖表中，y 軸值是指每個階段在指定時間的平均持續時間，以堆疊長條圖顯示。



下圖說明 Amazon Redshift 如何為並行工作階段彙總查詢的處理。



檢視叢集工作負載明細圖表

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後從清單中選擇叢集名稱以開啟其詳細資訊。叢集的詳細資訊隨即顯示，包含叢集效能、查詢監控、資料庫、資料共用、排程、維護和屬性索引標籤。
3. 選擇與查詢有關的指標 Query monitoring (查詢監控) 標籤。
4. 在 Query monitoring (查詢監控) 區段中，選擇 Database performance (資料庫效能) 及選擇 Cluster metrics (叢集指標)。

   下列指標已建立所選時間範圍的堆疊長條圖：

   - Plan (計畫) 階段
   - Wait (等待) 階段
   - Commit (遞交) 時間
   - Execution (執行) 時間

分析查詢執行

您可以分析查詢的執行詳細資訊，藉此了解其執行狀況，並找出可最佳化的層面。分析查詢可獲得查詢計畫的深入資訊，包括涉及的步驟、每個步驟所花的時間，以及處理的資料量。常見的使用案例包括對緩慢執行的查詢進行故障診斷、最佳化資料分佈策略，以及識別查詢重寫或編製索引的機會。

分析查詢

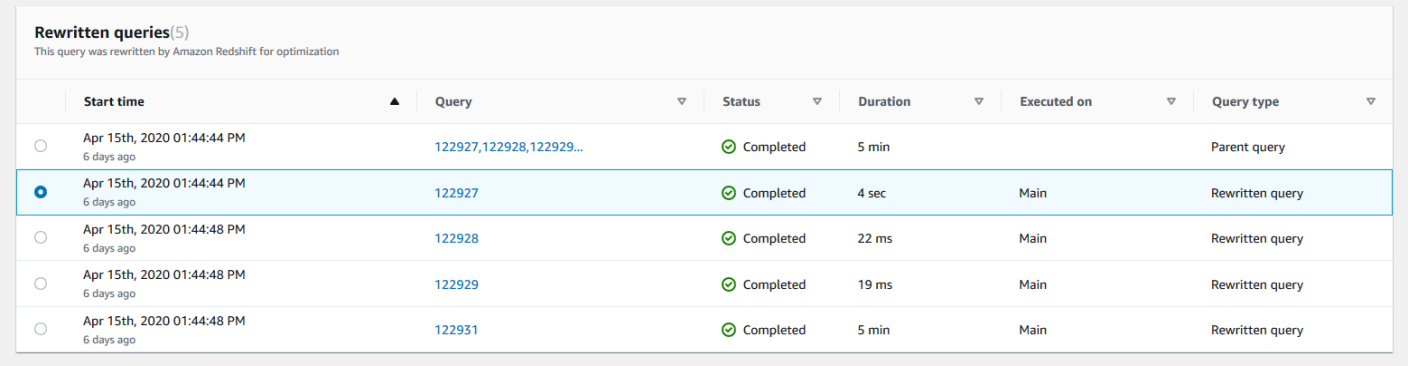
1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇查詢和載入以顯示帳戶的查詢清單。您可能需要變更此頁面的設定，才能找到您的查詢。
3. 在清單中選擇 Query (查詢) 識別碼，以顯示 Query details (查詢詳細資訊)。

   Query details (查詢詳細資訊) 頁面包括附有查詢指標的 Query details (查詢詳細資訊) 和 Query plan (查詢計畫) 標籤。

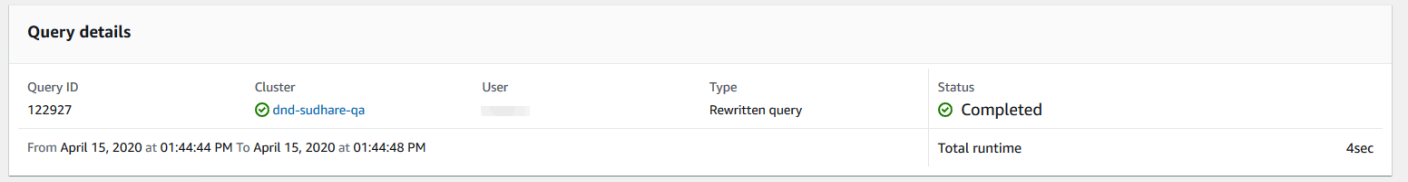
> **Note**
>
> 當您向下切入查詢執行時期圖表中的查詢時，您也可以從叢集詳細資訊頁面的查詢歷史記錄索引標籤瀏覽至查詢詳細資訊頁面。

Query details (查詢詳細資訊) 頁面包含下列各節：

- Rewritten queries (重新撰寫的查詢) 清單，如以下螢幕擷取畫面所示。



- Query details (查詢詳細資訊) 區段，如以下螢幕擷取畫面所示。



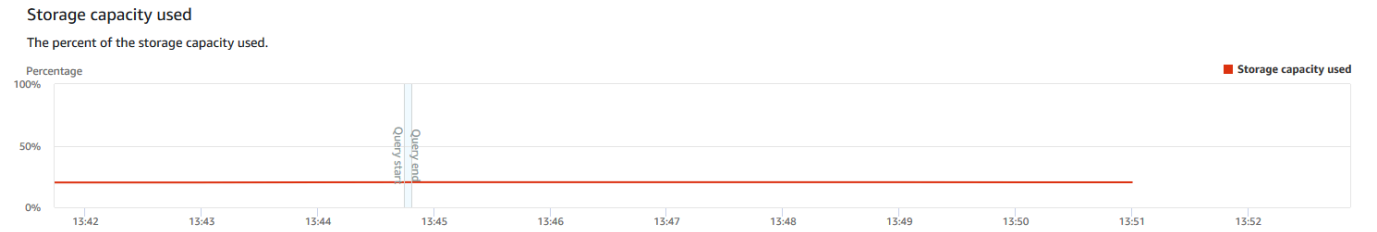
- Query details (查詢詳細資訊) 標籤，包含執行的 SQL 和執行的 Execution details (執行詳細資訊)。
- Query plan (查詢計劃) 標籤，包含 Query plan (查詢計劃) 步驟和其他查詢計劃的相關資訊。此表格也包含執行查詢的叢集相關圖表。
  - Cluster health status (叢集運作狀態)



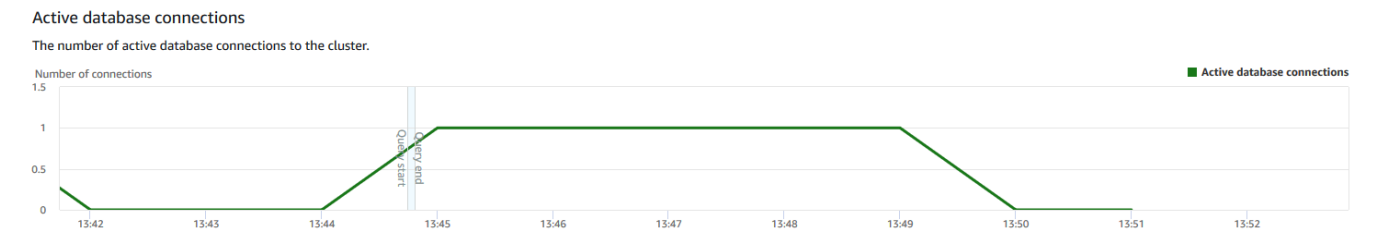
  - CPU 使用率



  - Storage capacity used (已使用的儲存體容量)



  - Active database connections (作用中的資料庫連線)



建立 警示

您在 Amazon Redshift 主控台中建立的警示是 CloudWatch 警示。因為它們可協助您針對叢集和無伺服器執行個體做出主動的決定，因此很有用。您可以對 Amazon Redshift 中的效能資料 中所列的任何指標設定一或多個警示。例如，在叢集節點上設定高 CPUUtilization 的警示，可協助指出節點過度使用的時機。DataStorage 偏高的警示會追蹤無伺服器命名空間用於資料的儲存空間。

您可以從動作中修改或刪除警示。您也可以透過指定 Slack 或 Amazon Chime Webhook URL，建立 Chime 或 Slack 警示，將警示從 CloudWatch 傳送至 Slack 或 Amazon Chime。

您可以在本節中找到如何使用 Amazon Redshift 主控台來建立警示。您可以使用 CloudWatch 主控台或使用指標的任何其他方式建立警示，例如使用 AWS CLI 或 AWS SDK。

使用 Amazon Redshift 主控台建立 CloudWatch 警示

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。

   如果您使用的是 Amazon Redshift Serverless，請選擇儀表板右上角的前往無伺服器。
2. 在導覽功能表上，選擇警示，然後選擇建立警示。
3. 在建立警示頁面中，輸入屬性以建立 CloudWatch 警示。
4. 選擇 Create alarm (建立警示)。

終止執行中的查詢

您也可以使用 Queries (查詢) 頁面終止目前正在進行的查詢。

> **Note**
>
> 在 Amazon Redshift 主控台終止查詢和載入的功能需要特定的許可。如果您希望使用者能夠終止查詢和載入，請務必將 redshift:CancelQuerySession動作新增至您的 AWS Identity and Access Management (IAM) 政策。無論您選取 Amazon Redshift 唯讀受管 AWS 政策或在 IAM 中建立自訂政策，此要求都適用。具有 Amazon Redshift 完整存取政策之使用者已擁有必要的許可，可終止查詢和載入。如需 Amazon Redshift 之 IAM 政策中動作的相關資訊，請參閱管理 資源的存取。

終止執行中的查詢

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇查詢和載入以顯示帳戶的查詢清單。
3. 選擇清單中要終止的執行中查詢，然後選擇 Terminate query (終止查詢)。

CloudWatch 主控台中的效能指標

在 CloudWatch 主控台中使用 Amazon Redshift 指標時，請謹記以下幾點：

- 僅在 Amazon Redshift 主控台中才可使用查詢和載入效能資料。
- CloudWatch 中有些指標所用的單位，和 Amazon Redshift 主控台所用的單位不同。例如，`WriteThroughput` 會以 GB/s 顯示 (在 CloudWatch 中以 Bytes/s 顯示)，此單位和一般節點儲存空間的關聯性更高。

在 CloudWatch 主控台、命令列工具或 Amazon 開發套件中使用 Amazon Redshift 指標時，請謹記以下事項：

1. 首先，指定要處理的指標維度。維度是一組用來單獨辨識指標的名稱值組。Amazon Redshift 的維度是 `ClusterIdentifier` 和 `NodeID`。CloudWatch 主控台會提供 `Redshift Cluster` 和 `Redshift Node` 檢視，以利選取叢集和節點特定維度。如需維度的相關資訊，請參閱《CloudWatch 開發人緣指南》中的維度。
2. 接著，指定指標名稱，例如 `ReadIOPS`。

下表摘要說明可供您使用之 Amazon Redshift 指標維度的類型。視指標而定，每隔 1 分鐘或 5 分鐘可免費取得資料。如需詳細資訊，請參閱Amazon Redshift 指標。

CloudWatch 命名空間	維度	說明
`AWS/Redshift`	`NodeID`	篩選專屬於某個叢集的節點的資料。`NodeID` 可以是「Leader」、「Shared」或「Compute-N」，其中的 N 是 0, 1, ...，亦即叢集中的節點數量。「Shared」表示該叢集只有一個節點，例如領導節點與運算節點結合。

CloudWatch 命名空間	維度	說明
AWS/Redshift	ClusterIdentifier	篩選專屬於某個叢集的資料。專屬於叢集的指標包括 HealthStatus 、MaintenanceMode 及 DatabaseConnections 。通常，此維度中的指標 (例如 ReadIOPS) 也是代表節點指標資料彙總的節點指標。在解譯這些指標時應特別注意，因為它們彙整了領導與運算節點的行為。

閘道和磁碟區指標的使用類似於其他服務指標的使用。CloudWatch 文件羅列許多一般任務，包括：

- 檢視可用的指標
- 取得指標的統計資訊
- 建立 CloudWatch 警示

查詢分析工具

本文件說明查詢分析工具，這是用於分析查詢的元件和效能的圖形化工具。

查詢分析工具是一項查詢監控和故障診斷功能，可透過 Amazon Redshift 主控台檢視。此工具對於分析查詢效能很有幫助。此工具的主要用途在於顯示有關查詢的視覺化圖形執行順序、執行計劃及統計資料，並且讓這些資訊更容易理解和進行故障診斷。查詢分析工具可協助您分析下列類型的查詢元件：

- 子查詢 - 子查詢是查詢工作的一部分。如果比起當做單一大查詢處理會更具效率，Amazon Redshift 就可能會將查詢分為多個子查詢。在分析工具中，您可以查看每個子查詢的屬性。子查詢是由串流和其他子元件所組成。

  查詢分析工具顯示的子查詢類型通常包括以下各項：

  - 暫存資料表查詢：此子查詢的文字開頭為 CREATE TEMP TABLE 命令。此子查詢會為要處理的其他子查詢建立暫存資料表。
  - 統計資料查詢：查詢分析工具會將下列註解新增至此子查詢的開頭，有助於識別該子查詢：

    ```
    -- collect statistics of child query queryID
    ```

    此子查詢會收集 Amazon Redshift 查詢引擎用來最佳化效能的資訊。

> **Note**
>
> 查詢分析工具會顯示使用者提供作為 Amazon Redshift 執行之最終子查詢的查詢。

- 串流 - 串流是分配在可用運算節點切片上的區段集合。每個子查詢都是由一或多個區段組成。在查詢分析工具中，您可以查看每個串流的屬性，例如其執行時間。透過瀏覽串流清單，您可能很快就能找到效能瓶頸。
- 區段 - 區段是單一處理程序可執行的一組數個步驟。區段也是運算節點切片可執行的最小編譯單位。配量是 Amazon Redshift 中平行處理的單位。串流中的區段會平行執行。查詢分析工具不會以圖形方式顯示區段，但您可以在步驟的詳細資訊窗格中存取該步驟的區段資訊。
- 步驟 - 每個區段都是由一系列步驟組成。步驟是查詢工作的一部分。例如，步驟可以包括 hashjoin，或是 scan，這是從資料表讀取記錄的作業。

如需串流、區段和步驟的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的查詢規劃和執行工作流程。

查詢分析工具會顯示 SYS_QUERY_HISTORY、SYS_QUERY_DETAIL、SYS_QUERY_EXPLAIN 和 SYS_CHILD_QUERY_TEXT 檢視傳回的資訊。如需這些檢視的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SYS_QUERY_HISTORY、SYS_QUERY_DETAIL、SYS_QUERY_EXPLAIN 和 SYS_CHILD_QUERY_TEXT。

查詢分析工具只會顯示近期在資料庫上執行之查詢的查詢資訊。若查詢是使用預先填入的快取資料完成，而不是對資料庫執行，則在之前沒有可用資訊的情況下，查詢將不會有查詢設定檔。這是因為 Amazon Redshift 不會為其產生查詢計畫。

使用查詢分析工具的先決條件

SYS 監控檢視的設計目的是易於使用和降低複雜性，提供完整的指標陣列以進行有效的監控和疑難排解。SYS 監控檢視也會保證過去七天的查詢歷史記錄 (無論叢集的大小或活動為何)。使用者只能查看自己執行的查詢，而超級使用者可以查看所有使用者的查詢。

您的 IAM 使用者帳戶或角色需有許可，才能存取主控台的查詢和資料庫監控區段。本節說明如何將許可新增至使用者帳戶或角色。

使用下列政策將最低許可新增至您的 IAM 使用者帳戶或角色：

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:*"
            ]
        }
    ]
}
```

主題

- 對角色授予查詢監控許可
- 對使用者授予查詢監控許可
- 使用 IAM 身分的臨時憑證

對角色授予查詢監控許可

若使用者擁有的角色具有 sys:monitor 許可，則可以檢視所有查詢。若使用者擁有的角色具有 sys:operator 許可，則可以取消查詢、分析查詢歷史記錄，以及執行真空操作。

向角色授予查詢監控許可

1. 使用下列命令來提供系統監控存取權，其中 *role-name* 是您想要為其提供存取權之角色的名稱。

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

2. (選用) 使用下列命令來提供系統操作員存取權，其中 *role-name* 是您想要為其提供存取權之角色的名稱。

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

對使用者授予查詢監控許可

具有 sys:monitor 許可的使用者可以檢視所有查詢。具有 sys:operator 許可的使用者可以取消查詢、分析查詢歷史記錄，以及執行清空操作。

向使用者授予查詢監控許可

1. 使用下列命令來提供系統監控存取權，其中 *user-name* 是您想要為其提供存取權之使用者的名稱。

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

2. (選用) 使用下列命令來提供系統操作員存取權，其中 *-name* 是您想要為其提供存取權之使用者的名稱。

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

使用 IAM 身分的臨時憑證

只有在連線至叢集時才能使用此選項。使用此方法時，查詢分析工具會將使用者名稱對應到您的 IAM 身分，並產生臨時密碼讓您以 IAM 身分連線到資料庫。使用此方法來連線的使用者必須已被允許 redshift:GetClusterCredentialsWithIAM 的 IAM 許可。若要防止使用者使用此方法，請修改其 IAM 使用者或角色以拒絕此許可。

在 Amazon Redshift 主控台中存取查詢分析工具以分析查詢

您可以存取 Amazon Redshift Serverless 或 Amazon Redshift 佈建的查詢分析工具。如需詳細資訊，請參閱下列的章節：

主題

- 在 Amazon Redshift 主控台中存取 Amazon Redshift Serverless 的查詢分析工具

- 在 Amazon Redshift 主控台中存取 Amazon Redshift 佈建的查詢分析工具

在 Amazon Redshift 主控台中存取 Amazon Redshift Serverless 的查詢分析工具

若要存取 Amazon Redshift Serverless 的查詢分析工具，請執行下列操作：

- 開啟 Amazon Redshift Serverless 主控台。
- 在導覽窗格的監控下，選擇查詢和資料庫監控。
- 選擇工作群組。
- 選擇查詢和資料庫監控。
- 選擇查詢。
- 在查詢詳細資訊頁面中選擇查詢計畫索引標籤。

如果有可用的查詢計畫，您會看到子查詢清單。選擇查詢即可在查詢分析工具中檢視該查詢。

在 Amazon Redshift 主控台中存取 Amazon Redshift 佈建的查詢分析工具

若要存取 Amazon Redshift 佈建的查詢分析工具，請執行下列操作：

- 開啟 Amazon Redshift 佈建叢集儀表板。
- 選擇叢集
- 選擇查詢監控。
- 連線至資料庫
- 選擇查詢和資料庫監控。
- 選擇查詢。

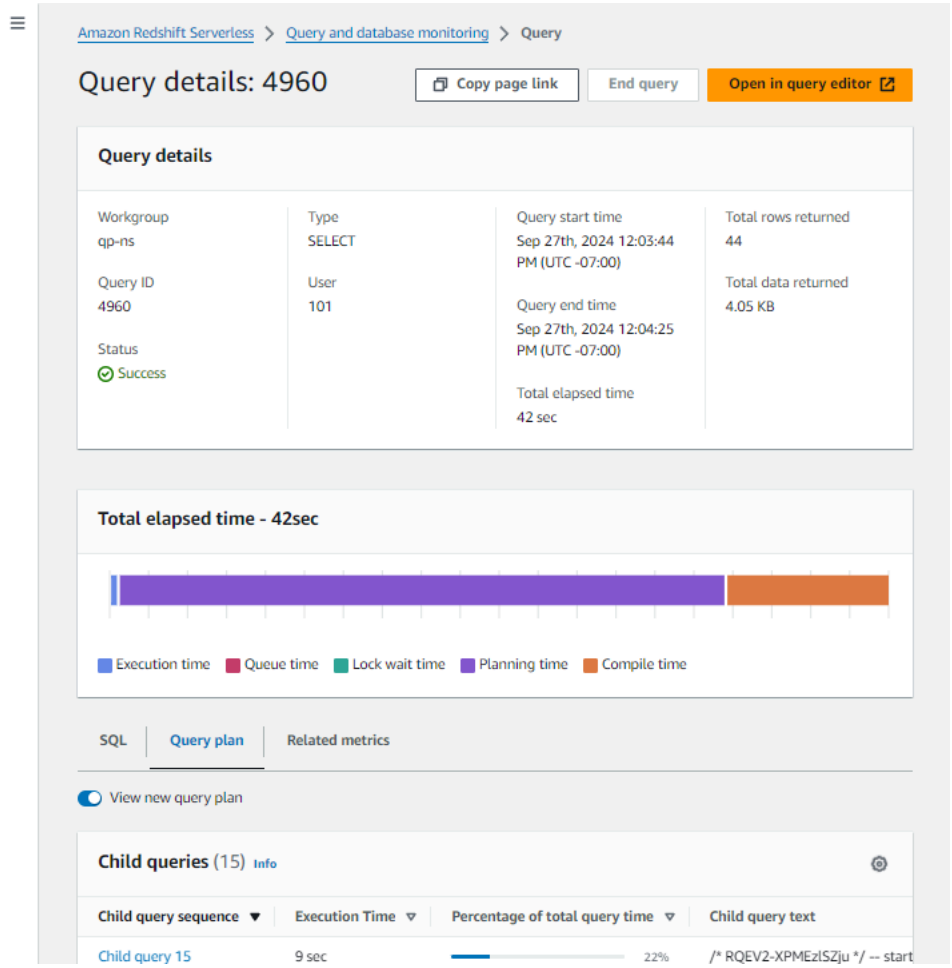
如果有可用的查詢計畫，您會看到子查詢清單。選擇查詢即可在查詢分析工具中檢視該查詢。

查詢分析工具使用者介面

查詢分析工具使用下列頁面來顯示有關查詢的資訊：

- 查詢詳細資訊頁面 - 此頁面顯示查詢的統計資料和子查詢。
- 子查詢頁面 - 此頁面顯示子查詢之執行計畫的統計資料、串流和視覺化呈現。當您從查詢和資料庫監控頁面的子查詢清單中選擇子查詢時，主控台就會顯示此頁面。

查詢詳細資訊頁面



查詢詳細資訊頁面包含下列元件：

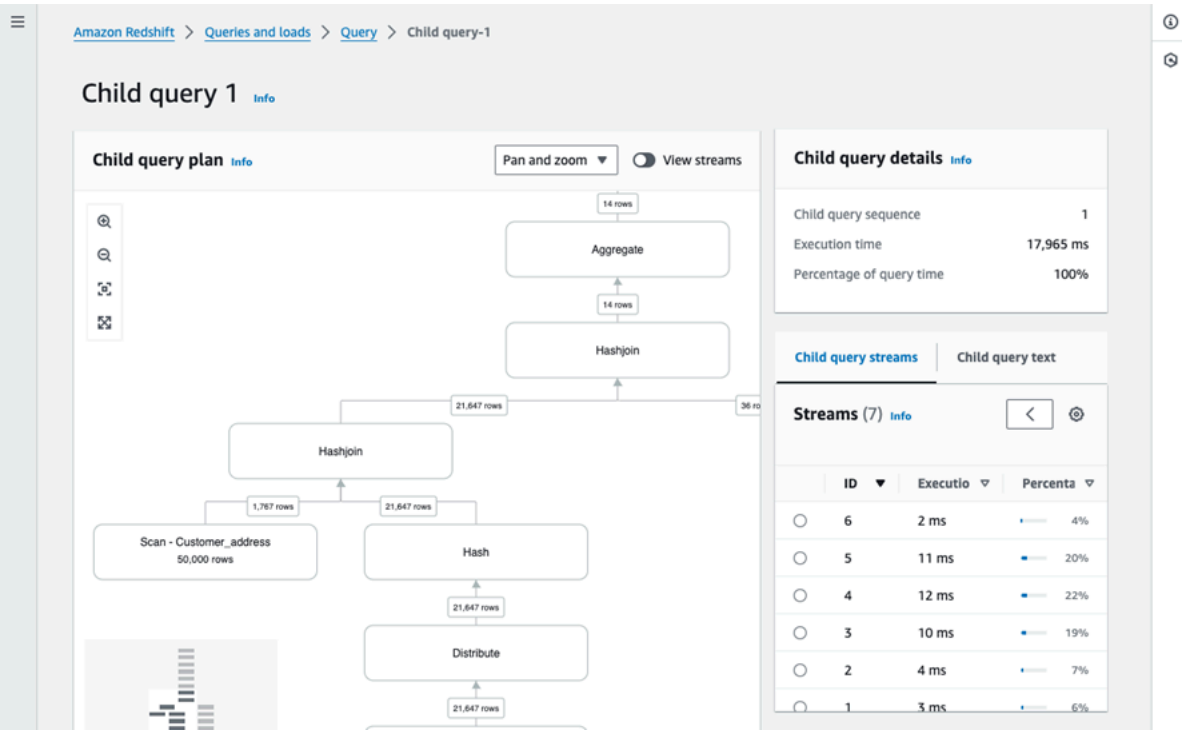
- 頂部窗格 - 頁面頂端的窗格會顯示有關查詢的詳細資訊，例如狀態和類型。如需頂部窗格所顯示資訊的來源相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SYS_QUERY_HISTORY。
- SQL 索引標籤 - 底部面板的此索引標籤會顯示原始使用者查詢的 SQL 文字。
- 查詢計畫索引標籤 - 底部面板的此索引標籤會顯示 Amazon Redshift 用來為使用者查詢準備資料和統計資料的子查詢清單。根據預設，子查詢清單會顯示有關每個子查詢的資訊和彙總統計資料。如需此頁面所顯示資訊的來源相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SYS_QUERY_DETAIL。

  您可以使用「偏好設定」功能表，在子查詢清單中新增或移除欄。
- 相關指標 - 底部面板的此索引標籤會顯示查詢的下列 CloudWatch 指標：

- 使用的 RPU 容量 (適用於無伺服器工作群組)：查詢使用的運算容量，以 Redshift 處理單元 (RPU 為單位計算。如需詳細資訊，請參閱Amazon Redshift Serverless 的運算容量。
- 叢集運作狀態、CPU 使用率、使用的儲存容量 (適用於佈建叢集)：狀態和查詢使用的系統資源。
- 作用中資料庫連線：查詢的 `DatabaseConnections` 指標。

如需 CloudWatch 指標的詳細資訊，請參閱Amazon Redshift 中的效能資料。

子查詢頁面



子查詢詳細資訊頁面包含下列元件：

- 子查詢下拉式清單 - 此控制項會顯示每個子查詢的序列名稱和執行時間。您可以在此控制項中選取其他子查詢，以瀏覽至其他子查詢。
- 側邊面板 - 此面板包含顯示子查詢串流和子查詢文字的索引標籤。
- 子查詢串流索引標籤 - 頂部面板的此索引標籤會顯示以下內容：
  - 串流 - 此窗格顯示子查詢中的串流清單。此窗格顯示 Amazon Redshift 用來最佳化查詢之串流的相關資訊和彙總資料。如需此窗格中詳細資訊的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SYS_QUERY_DETAIL。

    您可以使用「偏好設定」功能表，在串流窗格中新增或移除欄。您可以使用串流窗格中的齒輪圖示來存取「偏好設定」功能表。

  - 子查詢窗格 - 子查詢中步驟的圖形表示。如需有關子查詢窗格的資訊，請參閱下方的 子查詢窗格。
- 子查詢文字索引標籤 - 頂部面板的此索引標籤會顯示子查詢的 SQL 程式碼。
- 子查詢詳細資訊窗格 - 右側面板中的此窗格會顯示子查詢的詳細資訊。如需此窗格中詳細資訊的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SYS_QUERY_DETAIL。
- 串流詳細資訊窗格 - 當您在串流窗格中選擇串流時，串流詳細資訊窗格就會顯示串流的相關資訊。如需此面板中詳細資訊的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SYS_QUERY_DETAIL。
- 步驟詳細資訊窗格 - 當您在串流窗格或子查詢計畫的圖形中選擇步驟時，步驟詳細資訊窗格就會顯示步驟的相關資訊。如需此窗格中詳細資訊的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SYS_QUERY_DETAIL。

子查詢窗格

查詢分析工具會在子查詢窗格中，以所選取子查詢中步驟的圖形表示形式來顯示子查詢。

子查詢窗格會顯示步驟的執行順序和步驟之間的關係。例如，如果某一個步驟聯結來自另外兩個步驟的輸出，則子查詢窗格會將該步驟顯示為樹狀節點，且包含兩個對其饋送的節點：



根據預設，子查詢窗格不會顯示包含步驟的串流。若要顯示 Amazon Redshift 用來邏輯分割子查詢中步驟的串流，請選擇檢視串流。當您選擇檢視串流時，子查詢窗格會顯示查詢的串流所包含的步驟。

子查詢窗格不會顯示區段資訊。若要查看步驟的區段，請選擇步驟。然後步驟詳細資訊窗格就會顯示步驟的區段。

在子查詢窗格中瀏覽

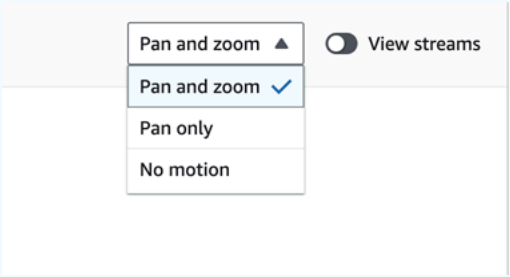
在子查詢窗格中，您可以選擇步驟以查看其詳細資訊。您也可以平移和縮放工作空間，以更清楚地視覺化查詢計畫中的步驟。

您可以使用下列方法選擇節點、平移和放大子查詢計畫：

- 使用滑鼠 - 您可以選擇節點、按一下並拖曳要平移的工作空間，以及按住 **Ctrl** (Windows) 或 **CMD** (Mac) 搭配滑鼠滾輪來縮放。選擇節點會導致工作空間縮放和平移以反白顯示該節點。如果您在工作

空間中選擇串流，該串流會在串流清單中反白顯示。如果您在工作空間中選擇步驟，步驟詳細資訊窗格會顯示該步驟的相關資訊。

- 使用工作空間左上方的縮放和調整控制項 - 這些控制項可讓您放大、縮小、縮放以納入整個工作空間，以及進入全螢幕模式。當您縮放以納入整個查詢計畫時，工作空間會同時將查詢計畫水平和垂直置中。
- 使用工作空間右下方的迷你地圖 - 您可以使用工作空間左下角的迷你地圖控制項平移或縮放工作空間。
- 在串流窗格中選擇串流 - 如果您在串流窗格中選擇某一個串流，工作空間會平移並縮放以顯示選取的串流，以及在串流詳細資訊窗格中顯示該串流的相關資訊。
- 在串流詳細資訊窗格中選擇步驟 - 如果您在串流詳細資訊窗格中選擇某一個步驟，工作空間會平移並縮放以顯示選取的步驟，以及在步驟詳細資訊窗格中顯示該步驟的相關資訊。

> **Note**
>
> 當您在工作空間或不同窗格中選擇步驟時，工作空間會嘗試縮放和平移，以最清楚的方式顯示選取的步驟。
>
> 當您從工作空間或其他窗格選擇串流或步驟時，如果您已在工作空間的右上角控制項中選取平移和縮放，則工作空間只會縮放並平移至該串流或步驟。您可以在下拉式清單中選擇適當的設定，以將此行為限於平移和縮放、僅平移，或不移動。
>
> 
>

使用查詢分析工具對查詢進行故障診斷

如果您要對查詢進行故障診斷，您可以選擇子查詢，以判斷哪個串流使用最高的總查詢時間百分比。這是快速判斷查詢的哪個部分應進一步分析的方法。

在您得知哪個子查詢花最久的時間後，請檢視其步驟，以查看哪些聯結或掃描可能會導致效能變慢。

查詢和資料庫監控

本文件說明查詢和資料庫監控頁面，此功能 AWS 管理主控台 用於分析 Amazon Redshift 佈建叢集或無伺服器工作群組的效能，以及針對這些叢集執行的查詢。

您可以使用查詢和資料庫監控頁面調查下列案例：

- 在指定期間內監控資料倉儲指標
- 查詢如何協助提升資料倉儲的整體效能
- 查看細分成生命週期事件的查詢執行時間，例如鎖定等待時間、編譯時間和執行時間
- 哪些使用者在指定期間內執行資源最密集的查詢
- 監控修補程式事件如何影響查詢效能

主題

- 許可
- 查詢和資料庫監控主控台

許可

AWS 帳戶 您用來存取主控台的 權限會影響查詢和資料庫監控頁面顯示的查詢。根據預設，您只能檢視自己擁有的查詢。若要檢視其他使用者擁有的查詢，請對您的帳戶授予 SYS:MONITOR 角色。若要允許使用者從查詢和資料庫監控頁面結束執行中的查詢，請對使用者授予 SYS:OPERATOR 權限。

若要自動將 sys:monitor 角色指派給 Amazon Redshift Serverless 或佈建的 IAM 使用者或角色，請執行下列命令：

```
create role monitor;
grant role sys:monitor to role monitor;
```

若要更新用於查詢監控的 IAM 角色，請執行下列操作：

1. 選擇 Tags (標籤) 索引標籤。
2. 選擇管理標籤。
3. 新增具有金鑰 **RedshiftDbRoles** 和值 **monitor** 的標籤。
4. 儲存變更

若要將資料庫憑證新增至使用者，請執行下列命令：

```
grant role sys:monitor to <username>
```

如需使用 GRANT 命令的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 GRANT。

若要使用查詢監控，您的 IAM 使用者需具有存取 Amazon Redshift 資料平面的許可。確認您的 IAM 使用者的許可政策中具有下列許可：

```
{
    "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 IAM 身分的臨時憑證

只有在連線至叢集時才能使用此選項。使用此方法時，查詢和資料庫監控會將使用者名稱對應到您的 IAM 身分，並產生能以 IAM 身分連線到資料庫的臨時密碼。使用此方法來連線的使用者必須已被允許 `redshift:GetClusterCredentialsWithIAM` 的 IAM 許可。若要防止使用者使用此方法，請修改其 IAM 使用者或角色以拒絕此許可。

查詢和資料庫監控主控台

本節說明如何使用查詢和資料庫監控主控台頁面。

您可以使用查詢和資料庫監控主控台，快速取得資料倉儲效能的概觀。您可以隨著時間監控資料倉儲的效能，並檢查資料倉儲佈建叢集或個別查詢的效能，以便最快識別瓶頸和其他需要改進的層面。

查詢和資料庫監控頁面包含下列功能：

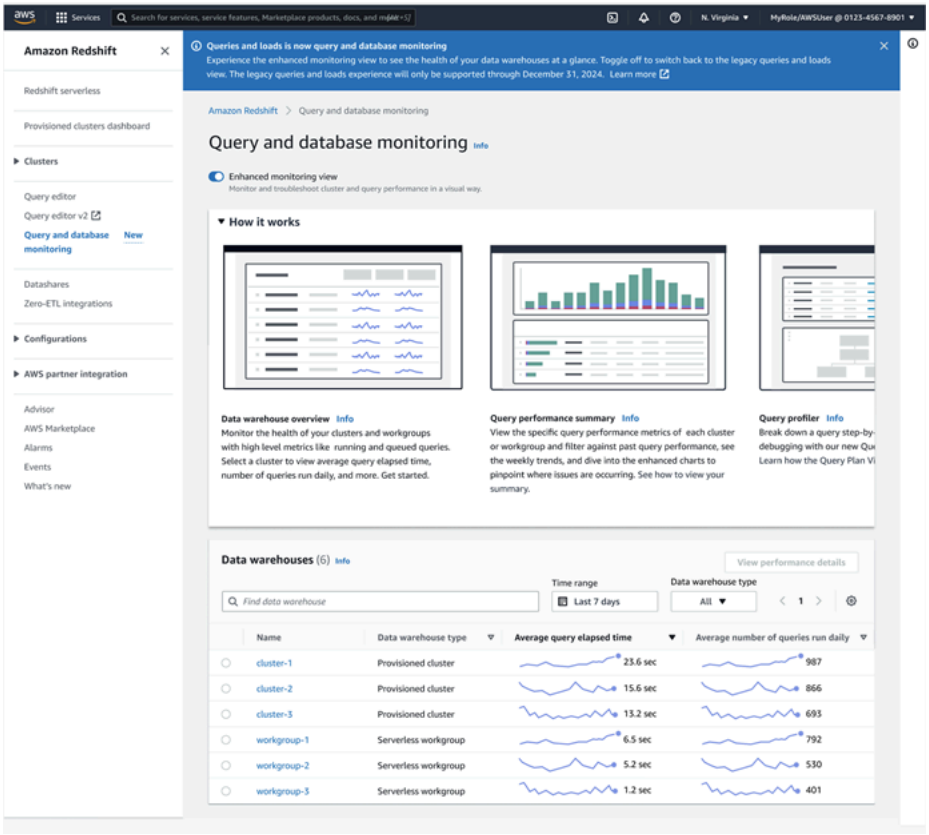
- 提高安全性 - 您需有提升的權限才能監視其他使用者的查詢。如需詳細資訊，請參閱許可。
- 七天查詢歷史記錄 - 保證可存取七天的查詢歷史記錄

- 查詢監控 - 您可以在使用者查詢層級監控佈建叢集和無伺服器工作群組中的查詢。
- 查詢趨勢分析 - 您可以比較符合指定條件之類似查詢的效能。

若要存取查詢和資料庫監控頁面，請執行下列操作：

1. 在 https：//https://console.aws.amazon.com/redshiftv2/ 登入 Amazon Redshift 主控台。
2. 從導覽窗格選擇查詢和資料庫監控。

查詢和資料庫監控主控台頁面顯示如下：



查詢和資料庫監控頁面包含下列元件：

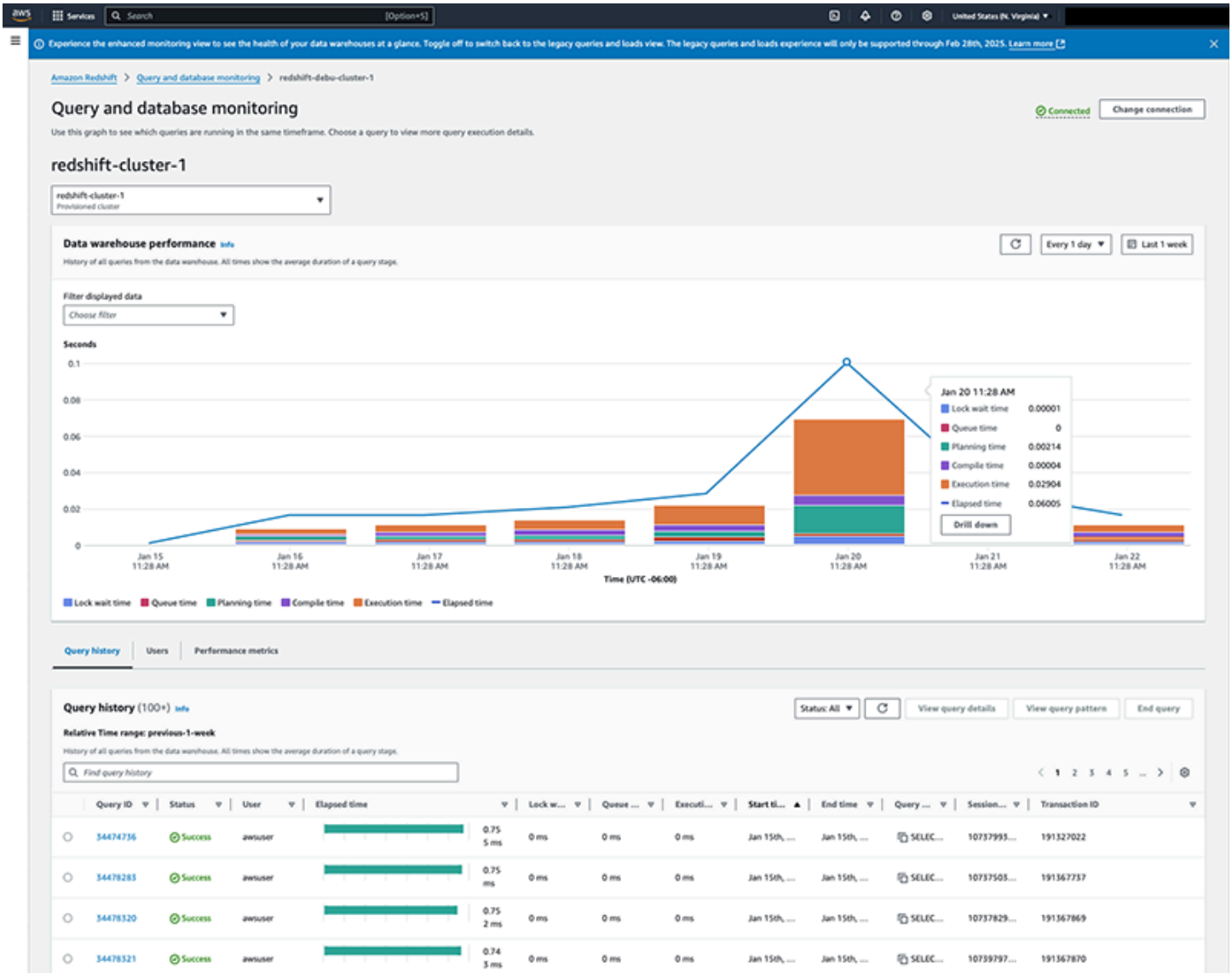
- 資料倉儲概觀 - 監控佈建叢集和無伺服器工作群組的平均查詢效能。您可以查看此頁面上的統計資料是否有高峰或高活動期間，以快速查看特定叢集或工作群組的效能問題。
- 查詢效能摘要 - 監控特定佈建叢集或無伺服器工作群組的平均查詢效能。您也可以按一下資料倉儲概觀清單中的其中一個叢集或工作群組，以存取查詢效能摘要頁面。

主題

- 查詢效能摘要
- 查詢詳細資訊
- 查詢模式

查詢效能摘要

當您從查詢和資料庫監控頁面選擇查詢效能摘要，或從資料倉儲概觀頁面選擇叢集或工作群組時，主控台會顯示個別佈建叢集或無伺服器工作群組的效能摘要。



此頁面包含下列元件：

- 叢集或工作群組下拉式清單 - 選擇您要分析的叢集或工作群組。

- 資料倉儲效能 - 此窗格顯示叢集或工作群組在指定期間內的歷史記錄，其中顯示查詢的每個階段所花費的時間長度。如果您在特定查詢階段看到高峰，例如上圖中 1 月 20 日的規畫和執行時間增加，您可以使用此資訊來找出查詢效能的問題。顯示的預設期間是過去七天，但您可以根據您的分析需求調整期間。
- 查詢歷史記錄 - 此窗格顯示指定篩選條件內執行的每個查詢的效能歷史記錄。您可以使用此資訊來對個別查詢的效能進行故障診斷。若要進一步分析特定查詢，您可以從此圖或資料倉儲效能圖中選擇查詢。

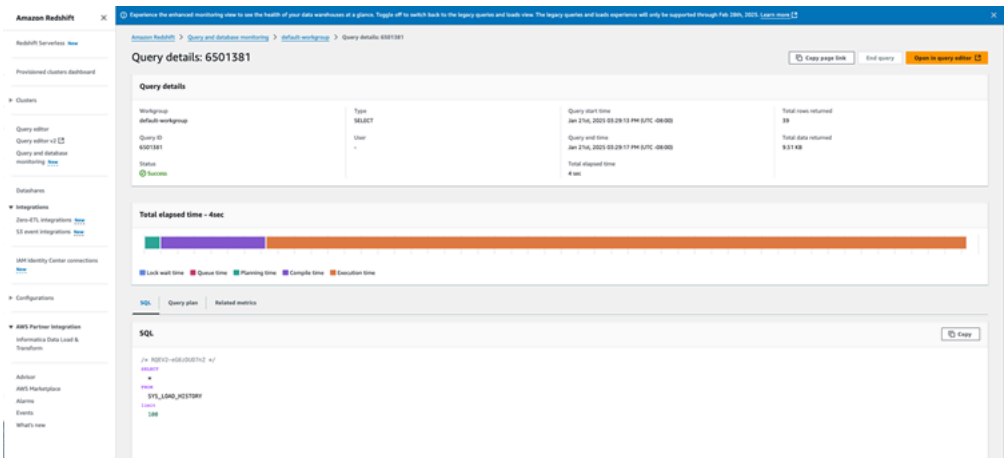
  當您將滑鼠游標暫留在圖表上的資料點上方時，就會出現快顯視窗。此快顯視窗會顯示該資料點的時間指標明細。如果資料點包含查詢資料，您可以選擇深入檢視，將圖表的時間範圍更新為資料點的下一個最小時間單位。這些時間單位如下所示：

  - 天
  - 小時
  - 15 分鐘
  - 5 分鐘
  - 1 分鐘

  例如，如果您選擇深入檢視資料點，則該圖的範圍會變更為一天。若再次選擇深入檢視，則圖的範圍會變更為一小時。
- 查詢分析工具 - 用於監控查詢效能的圖形工具。如需詳細資訊，請參閱查詢分析工具。

查詢詳細資訊

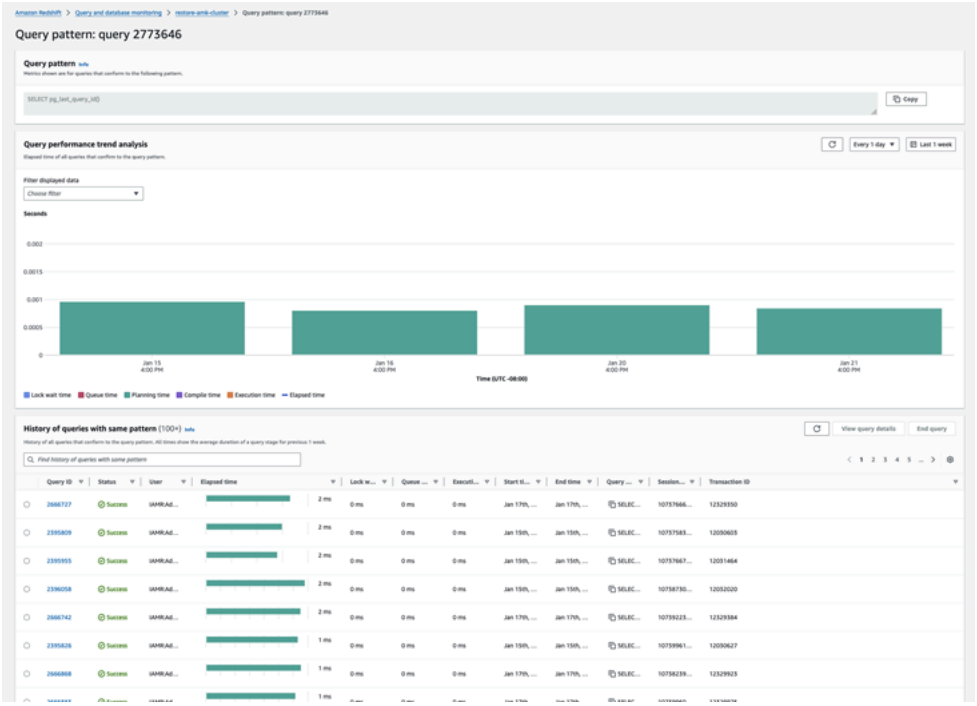
當您從叢集或工作群組詳細資訊頁面的資料倉儲效能或查詢歷史記錄窗格選擇查詢時，查詢詳細資訊頁面隨即開啟。



如需查詢詳細資訊頁面的相關資訊，請參閱 查詢詳細資訊頁面。

查詢模式

您可以在查詢效能摘要頁面的查詢歷史記錄窗格中選擇檢視查詢模式按鈕，以查看相同模式查詢的歷史記錄。查詢模式頁面會顯示過去一星期內，您指定的 SQL 陳述式擷取的所有查詢。



查詢模式頁面包含下列元件：

- 查詢模式 - 擷取要分析之查詢的 SQL 陳述式。
- 查詢效能趨勢分析 - 此長條圖顯示查詢模式選取的所有查詢的經過時間。結果會依天分組。
- 相同模式查詢的歷史記錄 - 查詢模式所選取查詢的每個階段所花的時間。

使用查詢模式頁面可獲得以下深入分析：

- 每天於特定時間執行之查詢的趨勢
- 您對資料倉儲定期執行之查詢的執行時間高峰。

Sys 檢視型查詢和資料庫監控

本文件說明在 Amazon Redshift 主控台中，為查詢和資料庫監控頁面提供資料的 SYS 檢視，這是用於分析查詢元件和效能的工具。如需有關查詢和資料庫監控頁面的資訊，請參閱 查詢和資料庫監控。

查詢和資料庫監控頁面有一項功能，可顯示 SYS 檢視提供的資訊。主控台檢視包含查詢分析工具，該工具會以圖形顯示查詢執行計畫。若要切換至 SYS 型檢視，請依照此處的步驟，對新的查詢和資料庫監控頁面授予適當的存取權和許可。

查詢和資料庫監控頁面的 SYS 型檢視功能具備下列功能：

- 提高安全性 - 您需有提升的權限才能監視其他使用者的查詢
- 七天查詢歷史記錄 - 保證可存取七天的查詢歷史記錄
- 查詢分析工具 - 用於監控查詢效能的圖形工具。如需詳細資訊，請參閱查詢分析工具

根據預設，您只能檢視自己擁有的查詢。若要檢視其他使用者擁有的查詢，請對您的帳戶授予 SYS:MONITOR 角色。若要允許使用者結束執行中的查詢，請對使用者授予 SYS:OPERATOR 權限。

若要對資料庫使用者或角色授予檢視所有使用者擁有之查詢的權限，請執行下列命令：

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

若要自動將 sys:monitor 角色指派給 Amazon Redshift Serverless 或佈建的 IAM 使用者或角色，請執行下列命令：

```
 create role monitor;
grant role sys:monitor to role monitor;
```

若要更新用於查詢監控的 IAM 角色，請執行下列操作：

1. 選擇 Tags (標籤) 索引標籤。
2. 選擇管理標籤。
3. 新增具有金鑰 **RedshiftDbRoles** 和值 **monitor** 的標籤。
4. 儲存變更

若要將資料庫憑證新增至使用者，請執行下列命令：

```
grant role sys:monitor to <username>
```

許可

若要使用查詢監控，您的 IAM 使用者需具有存取 Amazon Redshift 資料平面的許可。確認您的 IAM 使用者的許可政策中具有下列許可：

```
{
    "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"
},
```

連線至資料庫

使用增強型查詢監控功能之前，您必須先連線至資料庫，才能存取 SYS 檢視型資訊。若要連線至資料庫，請使用下列其中一種憑證：

- 使用者名稱和密碼
- 與您的 IAM 角色相關聯的臨時憑證
- 資料庫使用者

請注意下列有關使用增強型查詢監控的事項：

- 對於佈建叢集，您必須連線至資料庫，因為增強型查詢監控會使用 SYS 檢視。這些檢視較為安全，且需要更高的權限才能存取其他使用者所擁有查詢的資料。
- 使用 SYS 檢視型查詢和資料庫監控頁面時，只有在您的使用者帳戶沒有資料庫超級使用者角色時，才會顯示您的 `user_id`。非超級使用者看不見使用者名稱。
- 在 sys 檢視型查詢和資料庫監控頁面體驗中，查詢執行程序 ID (`p_id`) 會出現在欄標題 `session_id` 下。

Amazon Redshift 佈建叢集和無伺服器工作群組的追蹤

當 Amazon Redshift 發行新版本時，會更新 Amazon Redshift 資料倉儲的版本 (無伺服器工作群組或佈建叢集)。您可控制是否要將資料倉儲更新至最新版或前一版。

無伺服器工作群組或佈建叢集的追蹤會決定要版本更新期間套用的發行版本。Amazon Redshift 會在指定的維護時段更新佈建叢集，且通常會在閒置期間更新無伺服器工作群組。如需 Redshift Serverless 何時更新工作群組的詳細資訊，請參閱 更新無伺服器工作群組。

當 Amazon Redshift 發行新版本時，該版本會指派至目前追蹤，而先前版本會指派到後續追蹤。若要設定資料倉儲的追蹤，請指定下列其中一個值：

- 目前 - 透過目前追蹤即可取得包含最新功能、安全更新及效能增強功能的最新認證發行版本。
- 後續 - 透過後續追蹤則會使用先前的認證版本。

例如，假設您的無伺服器工作群組目前執行 1.0.2762 版，且 Amazon Redshift 發行 Redshift Serverless 1.0.3072 版。如果您的追蹤值為目前，您的工作群組就會更新至 1.0.3072 版 (最新發行版本)。如果您將追蹤值設定為後續，則您的工作群組會在下一個後續追蹤版本發行時更新。

若使用後續追蹤功能，您可以選擇執行後續追蹤中的一部分 Amazon Redshift 資料倉儲。如此就能對設定為目前追蹤的資料倉儲進行 1 到 6 週的測試和整合驗證，然後再將發行版本套用至後續追蹤上的資料倉儲。根據預設，Amazon Redshift 會在目前追蹤上建立所有叢集和工作群組，以利用最新的認證發行版本。不過，若在生產環境中使用 Amazon Redshift 後續追蹤，並且在測試和開發環境中使用目前追蹤，您就能進行額外的盡職調查，並有更多時間評估最新發行版本。後續追蹤可確保最大的穩定性，因此非常適合生產環境中的關鍵任務工作負載。

> **Note**
>
> 後續追蹤版本可能在短時間內會與目前追蹤版本相同。目前追蹤尚未升級至下一個版本時，就會發生這種情況。一般而言，目前追蹤版本比後續追蹤版本還新。

在追蹤之間切換

變更 Amazon Redshift 資源的追蹤通常是一次性的決策。變更軌道時請務必謹慎。如需各資料倉儲版本包含哪些功能的相關資訊，請參閱 適用於 Amazon Redshift 的叢集版本。

如果您將追蹤從後續變更為目前，我們會將資料倉儲更新為目前追蹤發行版本。如果您將資料倉儲的追蹤變更為後續，我們會將您的資料倉儲更新如下：

- 對於無伺服器工作群組，我們會在閒置期間更新您的資料倉儲版本。如需 Redshift Serverless 如何更新工作群組版本的詳細資訊，請參閱 更新無伺服器工作群組。
- 對於佈建叢集，在目前追蹤發行版本之後有新版本之前，我們不會更新您的資料倉儲。

追蹤和還原

對於無伺服器工作群組，快照會繼承目標 Amazon Redshift 資料倉儲的追蹤。例如，如果您為設定為後續追蹤的工作群組建立快照，並將該快照套用至設定為目前追蹤的工作群組，則工作群組的追蹤設定將會是目前。

對於佈建叢集，快照會繼承來源 Amazon Redshift 資料倉儲的追蹤。如果您在拍攝快照之後變更來源資料倉儲的追蹤，則快照和來源資料倉儲會在不同的追蹤上。當您從快照還原時，新的資料倉儲會在繼承自快照來源的追蹤上。您可在還原操作完成後變更追蹤。

調整資料倉儲的大小不會影響其追蹤。

更新無伺服器工作群組

當工作群組選擇的追蹤推出新版本時，只要沒有待處理的追蹤更新請求，Amazon Redshift Serverless 通常會在閒置期間套用更新。如果工作群組在 14 天內沒有閒置期間，則 Redshift Serverless 會強制執行版本更新。

Redshift Serverless 只會將您的工作群組更新為下一個較新的版本。即使所選工作群組的追蹤版本低於工作群組的目前版本，Redshift Serverless 也不會略過中繼版本或降級工作群組。除非 `Trailing` 追蹤追上進度，否則您的工作群組將不會進行任何主要版本升級。

例如，假設 `Current` 追蹤為 186 版，而 `Trailing` 追蹤為 185 版。如果您的工作群組的 `Track` 值為 `Current`，其版本為 186，如果您將 `Track` 值變更為 `Trailing`，Redshift Serverless 並不會將工作群組的版本降級為 185。在此案例中，Redshift Serverless 會將工作群組保持在 186 版，直到 `Trailing` 追蹤版本等於或高於 186 版。

如果追蹤變更待定，則 Redshift Serverless 不會將工作群組更新為現有追蹤中的下一個主要版本，直到套用追蹤變更為止。追蹤變更完成後，Redshift Serverless 會評估在新追蹤下將工作群組更新為適當版本的條件。

例如，如果您的工作群組設定為 `Current` 追蹤，而目前追蹤是 186，且您將工作群組變更為 `Trailing` 追蹤，則在套用追蹤變更之後，以及 `Trailing` 版本更新為等於或高於 186 的版本之後，Redshift Serverless 才會更新工作群組。

> **Note**
>
> 工作群組上的任何現有操作 (例如從快照還原、變更 KMS 金鑰或調整大小)，只會發生在現有追蹤上。Redshift Serverless 不會使用待處理追蹤進行無伺服器操作。

如果您有待處理追蹤切換請求，您可以使用 UpdateWorkgroup 將 `track` 參數設回其原始值，以取消請求。

管理版本

追蹤是一系列的版本。您可以決定 Amazon Redshift 資料倉儲是在目前追蹤還是後續追蹤上。如果您將資料倉儲放在目前追蹤上，則其一律會升級到最新的發行版本。如果您將資源放在後續追蹤上，則其一律會執行最新發行版本的前一個發行版本。

對於佈建叢集，Amazon Redshift 主控台的 Amazon Redshift 資料倉儲清單中的發行狀態欄會指出是否有其中一個資源可進行升級。

判斷工作群組或叢集版本

您可以使用 Amazon Redshift 主控台來判斷 Amazon Redshift 無伺服器工作群組版本或佈建叢集版本引擎。

登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。

Serverless workgroups

對於無伺服器工作群組，在導覽功能表上選擇工作群組，然後從清單中選擇工作群組名稱以開啟其詳細資訊。工作群組的詳細資訊隨即顯示。

Provisioned clusters

對於佈建叢集，在導覽功能表上選擇叢集，然後從清單中選擇叢集名稱以開啟其詳細資訊。

叢集的詳細資訊隨即顯示，包含叢集效能、查詢監控、資料庫、資料共用、排程、維護和屬性索引標籤。選擇維護索引標籤以取得詳細資訊。

在 Maintenance (維護) 區段中，尋找 Current cluster version (目前的叢集版本)。

> **Note**
>
> 對於佈建叢集，主控台會以一個欄位顯示版本資訊，但在 Amazon Redshift API 中會是兩個參數。這兩個參數為 `ClusterVersion` 和 `ClusterRevisionNumber`。如需詳細資訊，請參閱《Amazon Redshift API 參考》中的叢集。

零 ETL 整合

零 ETL 整合是一種全受管解決方案，可在 Amazon Redshift 中從多個操作和交易來源提供交易和操作資料。透過這個解決方案，您可以設定從您的來源到 Amazon Redshift 資料倉儲的整合。您不需要維護擷取、轉換和載入 (ETL) 管道。我們會為您處理 ETL，方法是自動建立和管理將資料從資料來源複寫到 Amazon Redshift 叢集或 Redshift Serverless 命名空間的複寫操作。您可以繼續更新和查詢來源資料，同時使用 Amazon Redshift 來處理分析工作負載，例如報告和儀表板。

透過零 ETL 整合，您可以取得更新的分析、AI/ML 和報告資料。可以針對業務儀表板、最佳化遊戲體驗、資料品質監控和客戶行為分析等使用案例，取得更準確且及時的洞見。可以更有信心地進行資料驅動型預測，提高客戶體驗，以及提升整個業務的資料驅動型洞察。

零 ETL 整合目前支援下列來源：

- 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
- 應用程式包括 Salesforce、Salesforce Marketing Cloud Account Engagement、SAP、ServiceNow、Instagram ads、Meta ads 及 Zendesk
- 自我管理的 MySQL、PostgreSQL、SQL Server 和 Oracle

若要建立零 ETL 整合，您必須指定整合來源，並指定 Amazon Redshift 資料倉儲做為目標。初始資料載入後，整合會將來源中的資料複寫到目標資料倉儲。資料便可在 Amazon Redshift 中使用。您可以在建立整合來源、建立零 ETL 整合以及建立 Amazon Redshift 資料倉儲時，控制資料的加密。整合會監控資料管道的運作狀態，並在可能的情況下從問題中復原。您可以建立從相同類型的來源至單一 Amazon Redshift 資料倉儲的整合，以便跨多個應用程式衍生整體洞察。

使用 Amazon Redshift 中的資料，您可以使用 Amazon Redshift 提供的分析。例如，內建機器學習 (ML)、具體化視觀表、資料共用，以及直接存取多個資料存放區和資料湖。對於資料工程師而言，零 ETL 整合可讓您存取可能會因複雜的資料管道中所發生的間歇性錯誤而遭到延遲的時效性資料。您可以在交易資料上執行分析查詢和 ML 模型，以針對時效性事件和商業決策衍生及時的深入分析。

您可以建立 Amazon Redshift 事件通知訂閱，因此，當指定的零 ETL 整合發生事件時，您可以收到通知。。若要檢視與整合相關的事件通知清單，請參閱 使用 Amazon EventBridge 的零 ETL 整合事件通知。建立訂閱最簡單的方式是使用 Amazon SNS 主控台。如需建立和訂閱 Amazon SNS 主題的詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的 Amazon SNS 入門。

當您開始使用零 ETL 整合時，請考慮下列概念：

- 來源資料庫是將其中的資料複寫到 Amazon Redshift 的資料庫。
- 目標資料倉儲是資料複製到的 Amazon Redshift 佈建的叢集或 Redshift Serverless 工作群組。
- 目的地資料庫是您從目標資料倉儲中的零 ETL 整合建立的資料庫。

如需可用來監控零 ETL 整合之系統資料表和檢視的相關資訊，請參閱 監控與 Amazon Redshift 系統檢視的零 ETL 整合。

如需零 ETL 整合 AWS 區域 每個來源支援的清單，請參閱 零 ETL 整合的支援區域。

如需零 ETL 整合的定價資訊，請參閱適當的定價頁面：

- Amazon Redshift 定價
- Amazon Aurora 定價
- Amazon RDS 定價
- Amazon DynamoDB 定價
- AWS Glue 定價

如需有關零 ETL 整合來源的詳細資訊，請參閱下列主題：

- 如需 Aurora 零 ETL 整合的詳細資訊，請參閱 Amazon Aurora 使用者指南中的零 ETL 整合的優點、重要概念、限制、配額和支援的區域。
- 如需 RDS 零 ETL 整合，請參閱 Amazon RDS 使用者指南中的零 ETL 整合的優點、重要概念、限制、配額和支援的區域。
- 對於 DynamoDB 零 ETL 整合，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 與 Amazon Redshift 的零 ETL 整合。
- 如需與應用程式的零 ETL 整合，請參閱《AWS Glue 開發人員指南》中的零 ETL 整合。

主題

- 將零 ETL 整合與 Amazon Redshift 搭配使用的考量
- 開始使用零 ETL 整合
- 在 Amazon Redshift 中建立目的地資料庫
- 在 Amazon Redshift 中查詢複寫的資料
- 檢視零 ETL 整合
- 歷史記錄模式
- 在 Amazon Redshift 中共用資料
- 監控零 ETL 整合
- 零 ETL 整合的指標
- 修改 DynamoDB 的零 ETL 整合
- 刪除 DynamoDB 的零 ETL 整合
- 零 ETL 整合的支援區域
- 對零 ETL 整合進行疑難排解

將零 ETL 整合與 Amazon Redshift 搭配使用的考量

以下考量適用於 Amazon Redshift 的零 ETL 整合。

- 您的目標 Amazon Redshift 資料倉儲必須符合下列先決條件：
  - 執行 Amazon Redshift Serverless 或 RG 或 RA3 節點類型的佈建叢集。
  - 已加密 (如果使用已佈建的叢集)。
  - 已啟用區分大小寫。
- 如果您刪除來源，而該來源是 Amazon Redshift 資料倉儲的授權整合來源，則所有相關聯的整合都會進入 FAILED 狀態。任何先前複寫的資料都會保留在您的 Amazon Redshift 資料庫中，並且可供查詢。
- 目的地資料庫是唯讀的。您無法在目的地資料庫中建立資料表、視觀表或具體化視觀表。不過，您可以在目標資料倉儲中的其他資料表上使用具體化視觀表。
- 具體化視觀表在用於跨資料庫查詢時才會得到支援。如需使用透過零 ETL 整合複寫之資料來建立具體化視觀表的相關資訊，請參閱 透過具體化視觀表查詢複寫的資料。
- 根據預設，您只能查詢目標資料倉儲中處於 Synced 狀態的資料表。若要查詢其他狀態的資料表，請將資料庫參數 QUERY_ALL_STATES 設定為 TRUE。如需有關設定 QUERY_ALL_STATES

的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE DATABASE 和 ALTER DATABASE。如需資料庫狀態的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SVV_INTEGRATION_TABLE_STATE。

- Amazon Redshift 只接受 UTF-8 字元，因此可能不遵守您的來源中定義的定序。排序和比較規則可能會有所不同，這最後可能會變更查詢結果。
- 每個 Amazon Redshift 資料倉儲目標的零 ETL 整合限制為 50 個。
- 整合來源中的資料表必須有主索引鍵。否則，無法將您的資料表複寫至 Amazon Redshift 中的目標資料倉儲。

  如需如何將主索引鍵新增至 Amazon Aurora PostgreSQL 的相關資訊，請參閱 AWS 資料庫部落格中的建立與 Amazon Redshift 的 Amazon Aurora PostgreSQL 零 ETL 整合時，在沒有主索引鍵的情況下處理資料表。如需如何將主索引鍵新增至 Amazon Aurora MySQL 或 RDS for MySQL 的相關資訊，請參閱 AWS 資料庫部落格中的建立與 Amazon Redshift 的 Amazon Aurora MySQL 或 Amazon RDS for MySQL 零 ETL 整合時，在沒有主索引鍵的情況下處理資料表。
- 您可以使用資料篩選來進行 Aurora 零 ETL 整合，以定義從來源 Aurora 資料庫叢集複寫至目標 Amazon Redshift 資料倉儲的範圍。您可以定義一或多項篩選條件，以選擇性地包含或排除複寫某些資料表，而不要將所有資料複寫至目標。如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用資料篩選以便與 Amazon Redshift 進行 Aurora 零 ETL 整合。
- 對於與 Amazon Redshift 的 Aurora PostgreSQL 零 ETL 整合，Amazon Redshift 最多支援來自 Aurora PostgreSQL 的 100 個資料庫。每個資料庫都會單獨從來源複寫至目標。
- 零 ETL 整合不支援將資料從交易資料存放區複寫到 Amazon Redshift 時進行轉換。資料會依原樣從來源資料庫複寫。不過，您可以在 Amazon Redshift 中對複寫的資料套用轉換。
- 零 ETL 整合在 Amazon Redshift 中執行時是使用平行連線。其使用從整合建立資料庫之使用者的憑證執行。
- 您可以針對零 ETL 整合設定 `REFRESH_INTERVAL`，以控制資料複寫至 Amazon Redshift 的頻率。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE DATABASE 和 ALTER DATABASE。
- 您從與 Amazon DynamoDB 的零 ETL 整合建立 Amazon Redshift 資料庫之後，資料庫狀態應該會從建立變更為作用中。此時會在來源 DynamoDB 資料表中開始將資料複寫至目標 Redshift 資料表，這些資料表是在目的地資料庫 (`ddb_rs_customerprofiles_zetl_db`) 的公有結構描述下建立。

在目標上使用歷史記錄模式時的考量事項

以下是在目標資料庫上使用歷史記錄模式時的考量事項。如需詳細資訊，請參閱歷史記錄模式。

- 當您捨棄來源上的某個資料表時，目標上不會捨棄該資料表，但其狀態會變更為 DroppedSource。您可以在 Amazon Redshift 資料庫上捨棄或重新命名資料表。
- 當您截斷來源上的資料表時，目標資料表上就會執行刪除作業。例如，如果來源上的所有記錄都遭到截斷，則目標欄 _record_is_active 上的對應記錄會變更為 false。
- 當您在目標資料表上執行 TRUNCATE 資料表 SQL 時，作用中歷史記錄列會標記為非作用中且包含對應的時間戳記。
- 資料表中的列設定為非作用中時，就可以在短暫 (約 10 分鐘) 延遲後將其刪除。若要刪除非作用中的列，請使用查詢編輯器 v2 或其他 SQL 用戶端連線至零 ETL 資料庫。
- 您只能在歷史記錄模式開啟的狀態下，從資料表中刪除非作用中的列。例如，類似下方所示的 SQL 命令只會刪除非作用中的列。

```
delete from schema.user_table where _record_delete_time <= '2024-09-10 12:34:56'
```

這相當於 SQL 命令，如下所示。

```
delete from schema.user_table where _record_delete_time <= '2024-09-10 12:34:56' and
 _record_is_active = False
```

- 關閉資料表的歷史記錄模式時，所有歷史資料都會儲存到名為 <schema>.<table-name>_historical_<timestamp> 的資料表，同時名為 <schema>.<table-name> 的原始資料表會重新整理。
- 若使用資料表篩選條件從複寫作業中排除歷史記錄模式為開啟狀態的資料表，則所有列都會設定為非作用中並變更為 DroppedSource 狀態。如需資料表篩選條件的詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用資料篩選以便與 Amazon Redshift 進行 Aurora 零 ETL 整合。
- 只有處於 Synced 狀態的資料表，才能將其歷史記錄模式切換為 true 或 false。
- 歷史記錄模式為開啟狀態之資料表的具體化視觀表會以完整重新計算的形式建立。

零 ETL 整合來源為 Aurora 或 Amazon RDS 時的考量事項

以下是與 Amazon Redshift 的 Amazon 和 Amazon RDS 零 ETL 整合的考量事項。

- 您可以使用資料篩選來進行 Aurora 和 RDS for MySQL 零 ETL 整合，以定義從來源資料庫叢集複寫至目標 Amazon Redshift 資料倉儲的範圍。您可以定義一或多項篩選條件，以選擇性地包含或排除複寫某些資料表，而不要將所有資料複寫至目標。如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用資料篩選以便與 Amazon Redshift 進行 Aurora 零 ETL 整合。

- 整合來源中的資料表必須有主索引鍵。否則，無法將您的資料表複寫至 Amazon Redshift 中的目標資料倉儲。

  如需如何將主索引鍵新增至 Amazon Aurora PostgreSQL 的相關資訊，請參閱 AWS 資料庫部落格中的建立與 Amazon Redshift 的 Amazon Aurora PostgreSQL 零 ETL 整合時，在沒有主索引鍵的情況下處理資料表。如需如何將主索引鍵新增至 Amazon Aurora MySQL 或 RDS for MySQL 的相關資訊，請參閱 AWS 資料庫部落格中的建立與 Amazon Redshift 的 Amazon Aurora MySQL 或 Amazon RDS for MySQL 零 ETL 整合時，在沒有主索引鍵的情況下處理資料表。
- Amazon Redshift VARCHAR 資料類型的長度上限為 65,535 個位元組。當來源的內容不符合此限制時，複寫就不會繼續，且資料表會進入失敗狀態。您可以將資料庫參數 TRUNCATECOLUMNS 設定為 TRUE 來截斷內容以符合欄寬。如需有關設定 TRUNCATECOLUMNS 的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE DATABASE 和 ALTER DATABASE。

  如需零 ETL 整合來源和 Amazon Redshift 資料庫之間資料類型差異的詳細資訊，請參閱《Amazon Aurora 使用者指南》中的 Aurora 和 Amazon Redshift 之間的資料類型差異。

對於 Aurora 來源，另請參閱《Amazon Aurora 使用者指南》中的限制。

對於 Amazon RDS 來源，另請參閱《Amazon RDS 使用者指南》中的限制。

零 ETL 整合來源為 DynamoDB 時的考量事項

以下是與 Amazon Redshift 的 DynamoDB 零 ETL 整合的考量事項。

- 不支援 DynamoDB 中長度超過 127 個字元的資料表名稱。
- DynamoDB 零 ETL 整合中的資料會對應至 Amazon Redshift 中的 SUPER 資料類型欄。
- 不支援長度超過 127 個字元的分割區索引鍵或排序索引鍵的欄名稱。
- 來自 DynamoDB 的零 ETL 整合只能對應至一個 Amazon Redshift 資料庫。
- 對於分割區索引鍵和排序索引鍵，精確度和小數位數上限為 (38,18)。DynamoDB 上的數值資料類型支援最高 38 的精確度。Amazon Redshift 也支援最高精確度 38，但 Amazon Redshift 上預設的十進位精確度/小數位數為 (38,10)。這表示小數位數值可以遭到截斷。
- 若要成功進行零 ETL 整合，DynamoDB 項目中的個別屬性 (由 name+value 組成) 不得超過 64 KB。
- 啟用時，零 ETL 整合會匯出完整的 DynamoDB 資料表以填入 Amazon Redshift 資料庫。完成此初始程序所需的時間取決於 DynamoDB 資料表大小。然後，零 ETL 整合會使用 DynamoDB 增量匯出，將更新從 DynamoDB 增量複寫到 Amazon Redshift。這表示 Amazon Redshift 中複寫的 DynamoDB 資料會自動保持為最新。

目前，DynamoDB 零 ETL 整合的最低延遲為 15 分鐘。您可以透過為零 ETL 整合設定非零的 REFRESH_INTERVAL 來進一步增加此值。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE DATABASE 和 ALTER DATABASE。

對於 Amazon DynamoDB 來源，另請參閱《Amazon DynamoDB 開發人員指南》中的先決條件和限制。

零 ETL 整合來源為 Salesforce、SAP、ServiceNow 和 Zendesk 等應用程式時的考量事項

以下是來源為 Salesforce、SAP、ServiceNow 和 Zendesk 等應用程式搭配 Amazon Redshift 時的考量事項。

- 不支援應用程式來源中長度超過 127 個字元的資料表名稱和欄名稱。
- Amazon Redshift VARCHAR 資料類型的長度上限為 65,535 個位元組。當來源的內容不符合此限制時，複寫就不會繼續，且資料表會進入失敗狀態。您可以將資料庫參數 TRUNCATECOLUMNS 設定為 TRUE 來截斷內容以符合欄寬。如需有關設定 TRUNCATECOLUMNS 的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE DATABASE 和 ALTER DATABASE。

  如需零 ETL 整合應用程式來源與 Amazon Redshift 資料庫之間資料類型差異的詳細資訊，請參閱《AWS Glue 開發人員指南》中的零 ETL 整合。
- 與應用程式的零 ETL 整合的最低延遲為 1 小時。您可以透過為零 ETL 整合設定非零的 REFRESH_INTERVAL 來進一步增加此值。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE DATABASE 和 ALTER DATABASE。

對於與應用程式的零 ETL 整合的來源，另請參閱《AWS Glue 開發人員指南》中的零 ETL 整合。

還原零 ETL 整合目標時的考量事項

當您將 Amazon Redshift Serverless 命名空間從快照或復原點還原至相同的無伺服器命名空間時，會自動維護與命名空間相關聯的零 ETL 整合。還原後會啟動完整重新同步，以確保目標 Amazon Redshift 資料庫中的資料與來源一致。適用下列注意事項：

- 將快照還原至不同的命名空間不會維護整合。
- 還原之後，Amazon Redshift 會針對所有維護的零 ETL 整合執行完整重新同步。在重新同步期間，資料擷取效能可能會暫時降低。

- 如果您在目標資料庫上使用歷史記錄模式，請記錄擷取快照到完整重新同步完成為止的版本。重新同步後，歷史記錄模式會恢復正常操作。此行為與任何其他完整重新同步的行為相同。如需詳細資訊，請參閱歷史記錄模式。
- 如果在擷取快照之後建立零 ETL 整合，整合會在還原之後進入 NEEDS_ATTENTION 狀態，因為快照中不存在對應的資料庫。若要解決此問題，您可以從包含整合的最新快照還原，或刪除整合。
- 如果在拍攝快照後刪除零 ETL 整合，則不會還原整合。快照中的資料庫會保留並可刪除。
- 如果在擷取快照之後變更資料表層級篩選條件，則還原的目標會使用整合中的目前篩選條件，而不是快照時的篩選條件。
- 若要在還原期間選擇退出維護整合，請取消勾選 還原頁面上的維護整合方塊 AWS 管理主控台，或者如果您正在使用 AWS CLI，請在呼叫 restore-from-snapshot 或 restore-from-recovery-point API 操作時設定 `--no-maintain-integration` 參數。當您選擇退出時，整合會在還原後進入 FAILED 狀態。您可以刪除整合並重新建立。
- 此功能僅適用於還原至相同無伺服器命名空間的 Amazon Redshift Serverless。佈建叢集上的快照還原不會維持零 ETL 整合。

調整零 ETL 整合目標大小時的考量事項

當您調整做為零 ETL 整合目標的 Amazon Redshift 佈建叢集大小時，調整大小可能會導致整合資料表重新同步。是否發生重新同步取決於調整大小的類型：

- 僅變更節點數量（就地調整大小）的彈性調整大小不會影響零 ETL 整合。資料表會保持同步。
- 變更節點類型的彈性調整大小會導致叢集上零 ETL 整合中的所有資料表重新同步。任何傳統調整大小也會觸發重新同步。這是因為這些調整大小操作會暫時變更資料表的分佈樣式，而服務會將資料重新分佈到新的叢集組態。如需詳細資訊，請參閱調整叢集的大小。

調整大小觸發重新同步時，適用下列考量：

- 整合會保持作用中狀態，並在調整大小完成後自動開始重新同步。您無需執行任何動作。
- 當資料表重新同步時，您無法在 Amazon Redshift 中查詢它。若要在重新同步期間保持可查詢資料表，請先將目的地資料庫TRUE上的 QUERY_ALL_STATES 參數設為 ，再開始調整大小。重新同步期間傳回的資料可能會過時，直到重新同步完成為止。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE DATABASE 和 ALTER DATABASE。
- 重新同步可能需要 20-25 分鐘或更長時間，取決於來源資料庫的大小。
- 您可以使用 SVV_INTEGRATION_TABLE_STATE 系統檢視來監控整合資料表的狀態。資料表會顯示 ResyncRequired或 ResyncInitiated 狀態，直到重新同步完成並返回 Synced。

開始使用零 ETL 整合

這組任務會逐步引導您設定第一個零 ETL 整合。首先，設定您的整合來源，並為其設定必要的參數和許可。然後，您可以從 Amazon Redshift 主控台或 繼續執行其餘的初始設定 AWS CLI。主控台提供為我修正選項，可更正一些組態問題。

主題

- 建立和設定目標 Amazon Redshift 資料倉儲
- 為資料倉儲開啟區分大小寫功能
- 設定 Amazon Redshift 資料倉儲的授權
- 建立零 ETL 整合

建立和設定目標 Amazon Redshift 資料倉儲

在這個步驟中，您會建立和設定目標 Amazon Redshift 資料倉儲，例如 Redshift Serverless 工作群組或佈建叢集。如果您已將 Amazon Redshift 資料倉儲設定為搭配零 ETL 整合使用，則可略過此步驟。

您的目標資料倉儲必須有下列特性：

- 執行 Amazon Redshift Serverless 或 RG 或 RA3 節點類型的佈建叢集。
- 已開啟區分大小寫功能 (`enable_case_sensitive_identifier`)。如需詳細資訊，請參閱為資料倉儲開啟區分大小寫功能。
- 加密 (如果您的目標資料倉儲是 Amazon Redshift 佈建叢集的話)。如需詳細資訊，請參閱Amazon Redshift 資料庫加密。
- 在與整合來源相同的 AWS 區域中建立。

若要為您的零 ETL 整合建立目標資料倉儲，請根據您的部署類型參考下列其中一個主題：

- 若要建立 Amazon Redshift 佈建的叢集，請參閱 建立叢集。
- 若要建立含命名空間的 Amazon Redshift Serverless 工作群組，請參閱 使用命名空間建立工作群組。

當您建立佈建叢集時，Amazon Redshift 也會建立預設參數群組。您無法編輯預設參數群組。但是，您可以在建立新叢集之前先建立自訂參數群組，然後將其與叢集建立關聯。或者，您可以編輯將與建立的

叢集相關聯的參數群組。在建立自訂參數群組時，或在編輯目前的參數群組以使用零 ETL 整合時，也必須為參數群組開啟區分大小寫功能。

若要使用 Amazon Redshift 主控台或 建立自訂參數群組 AWS CLI，請參閱建立參數群組。

為資料倉儲開啟區分大小寫功能

您可以附加參數群組，並在建立期間啟用已佈建叢集的區分大小寫。但是，您只能在建立無伺服器工作群組之後，透過 AWS Command Line Interface (AWS CLI) 更新該工作群組。這是支援區分大小寫的來源資料表和欄的必要項目。`enable_case_sensitive_identifier` 是一個組態值，其決定資料庫、資料表和欄的名稱識別碼是否區分大小寫。必須開啟這個參數才能在資料倉儲中建立零 ETL 整合。如需詳細資訊，請參閱 enable_case_sensitive_identifier。

若為 Amazon Redshift Serverless — 使用 開啟 Amazon Redshift Serverless 的區分大小寫 AWS CLI。請注意，您只能從 AWS CLI為 Amazon Redshift Serverless 開啟區分大小寫功能。

對於 Amazon Redshift 佈建的叢集，請使用下列其中一個主題為目標叢集啟用區分大小寫：

- 使用 Amazon Redshift 主控台為 Amazon Redshift 佈建叢集開啟區分大小寫功能
- 使用 開啟 Amazon Redshift 佈建叢集的區分大小寫 AWS CLI

使用 開啟 Amazon Redshift Serverless 的區分大小寫 AWS CLI

執行下列 AWS CLI 命令，以開啟工作群組的區分大小寫功能。

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

等待工作群組的狀態變為 `Active` 後，再繼續進行下一個步驟。

使用 Amazon Redshift 主控台為 Amazon Redshift 佈建叢集開啟區分大小寫功能

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在左側導覽窗格中，選擇佈建叢集儀表板。
3. 選擇您要將資料複寫到其中的佈建叢集。
4. 在左側導覽窗格中，選擇組態 > 工作負載管理。

5. 在工作負載管理頁面上，選擇參數群組。
6. 選擇參數索引標籤。
7. 選擇編輯參數，然後將 enable_case_sensitive_identifier 變更為 true。
8. 然後選擇 Save (儲存)。

使用 開啟 Amazon Redshift 佈建叢集的區分大小寫 AWS CLI

1. 由於您無法從終端機程式編輯預設參數群組，請執行下列 AWS CLI 命令來建立自訂參數群組。稍後，您會為其與已佈建的叢集建立關聯。

```
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. 執行下列 AWS CLI 命令來開啟參數群組的區分大小寫。

```
aws redshift modify-cluster-parameter-group \
    --parameter-group-name zero-etl-params \
    --parameters ParameterName=enable_case_sensitive_identifier,ParameterValue=true
```

3. 執行以下命令將參數群組與叢集相關聯。

```
aws redshift modify-cluster \
    --cluster-identifier target-cluster \
    --cluster-parameter-group-name zero-etl-params
```

4. 等待佈建叢集變為可用狀態。您可以使用 `describe-cluster` 命令檢查叢集的狀態。然後，執行以下命令重新啟動叢集。

```
aws redshift reboot-cluster \
    --cluster-identifier target-cluster
```

設定 Amazon Redshift 資料倉儲的授權

若要將資料從整合來源複寫到您的 Amazon Redshift 資料倉儲，您最初必須新增下列兩個實體：

- 授權主體 — 識別可在資料倉儲中建立零 ETL 整合的使用者或角色。

- 授權的整合來源 — 識別可更新資料倉儲的來源資料庫。

您可以從 Amazon Redshift 主控台上的資源政策索引標籤或使用 Amazon Redshift `PutResourcePolicy` API 作業，設定授權的整合來源。

新增授權主體

若要在 Redshift Serverless 工作群組或佈建叢集中建立零 ETL 整合，請授權相關聯命名空間或佈建叢集的存取權。

如果下列兩個情況都成立，則可以略過此步驟：

- AWS 帳戶 擁有 Redshift Serverless 工作群組或佈建叢集的 也擁有來源資料庫。
- 該主體與以身分為基礎的 IAM 政策相關聯，且具有可在此 Redshift Serverless 命名空間或佈建叢集中建立零 ETL 整合的許可。

將授權主體新增至 Amazon Redshift Serverless 命名空間

1. 在 Amazon Redshift 主控台中，從左側導覽窗格中選擇 Redshift Serverless。
2. 選擇命名空間組態，然後選擇您的命名空間，然後移至資源政策索引標籤。
3. 選擇新增授權主體。
4. 對於您要新增的每個授權委託人，在命名空間中輸入 AWS 使用者或角色的 ARN，或 AWS 帳戶 您要授予建立零 ETL 整合存取權的 的 ID。帳戶 ID 會儲存為 ARN。
5. 選擇儲存變更。

將授權主體新增至 Amazon Redshift 佈建叢集

1. 在 Amazon Redshift 主控台的左側導覽窗格中，選擇佈建叢集儀表板。
2. 選擇叢集，然後選擇叢集，接著前往資源政策索引標籤。
3. 選擇新增授權主體。
4. 對於您要新增的每個授權委託人，在叢集中輸入 AWS 使用者或角色的 ARN，或 AWS 帳戶 您要授予建立零 ETL 整合之存取權的 ID。帳戶 ID 會儲存為 ARN。
5. 選擇儲存變更。

新增授權的整合來源

若要允許來源更新 Amazon Redshift 資料倉儲，您必須以授權的整合來源的形式將其新增至命名空間。

將授權的整合來源新增至 Amazon Redshift Serverless 命名空間

1. 在 Amazon Redshift 主控台中，前往 無伺服器儀表板。
2. 選擇命名空間的名稱。
3. 前往資源政策索引標籤。
4. 選擇新增授權的整合來源。
5. 針對零 ETL 整合指定來源的 ARN。

> **Note**
>
> 移除授權的整合來源會讓系統不再將資料複寫至命名空間。此動作會停用從該來源到此命名空間的所有零 ETL 整合。

將授權的整合來源新增至 Amazon Redshift 佈建叢集

1. 在 Amazon Redshift 主控台中，前往佈建叢集儀表板。
2. 選擇佈建叢集的名稱。
3. 前往資源政策索引標籤。
4. 選擇新增授權的整合來源。
5. 指定屬於零 ETL 整合資料來源之來源的 ARN。

> **Note**
>
> 移除授權的整合來源會讓系統不再將資料複寫至佈建叢集。此動作會停用從該來源到此 Amazon Redshift 佈建叢集的所有零 ETL 整合。

使用 Amazon Redshift API 設定授權

您可以使用 Amazon Redshift API 操作來設定與零 ETL 整合搭配使用的資源政策。

若要控制可以在命名空間中建立傳入整合的來源，請建立資源政策，並將其附加至命名空間。使用資源政策，您可以指定具有整合存取權的來源。資源政策會附加至目標資料倉儲的命名空間，以允許來源建立傳入整合，將即時資料從來源複寫到 Amazon Redshift。

以下是範例資源政策。

```
{
  "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"
    }
  ]
}
```

以下概述了適用於設定整合資源政策的 Amazon Redshift API 操作：

- 使用 PutResourcePolicy API 作業來保留資源政策。當您提供其他資源政策時，會取代資源上先前的資源政策。使用上一個範本資源政策，該策略會授予下列動作的許可：
  - `CreateInboundIntegration` – 允許來源主體建立輸入整合，以便將資料從來源複寫到目標資料倉儲。

- `AuthorizeInboundIntegration` – 允許 Amazon Redshift 持續驗證目標資料倉儲是否可以接收從來源 ARN 複寫的資料。
- 使用 GetResourcePolicy API 操作是查看現有的資源政策。
- 使用 DeleteResourcePolicy API 作業從資源中移除資源政策。

若要更新資源策略，您也可以使用 put-resource-policy AWS CLI 命令。例如，若要將資源政策放在 DynamoDB 來源的 Amazon Redshift 命名空間 ARN 上，請執行類似以下的 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"
        }
    ]
}
```

建立零 ETL 整合

首先，您會建立零 ETL 整合將來源資料複寫至 Amazon Redshift。

資料的來源決定了要建立的零 ETL 整合類型。

主題

- 建立 Aurora 的零 ETL 整合
- 建立 Amazon RDS 的零 ETL 整合
- 建立 DynamoDB 的零 ETL 整合
- 建立與應用程式的零 ETL 整合

建立 Aurora 的零 ETL 整合

在此步驟中，您會建立與 Amazon Redshift 的 Aurora 零 ETL 整合。

建立與 Amazon Redshift 的 Aurora 零 ETL 整合

1. 如 Amazon Aurora 使用者指南所述，從 Amazon RDS 主控台建立自訂資料庫叢集參數群組。
2. 如 Amazon Aurora 使用者指南所述，從 Amazon RDS 主控台建立來源 Amazon Aurora 資料庫叢集。
3. 從 Amazon Redshift 主控台：建立和設定目標 Amazon Redshift 資料倉儲。
   - 從 AWS CLI 或 Amazon Redshift 主控台：為資料倉儲開啟區分大小寫功能。
   - 從 Amazon Redshift 主控台：設定 Amazon Redshift 資料倉儲的授權。
4. 如 Amazon Aurora 使用者指南中所述，從 Amazon RDS 主控台建立零 ETL 整合。
5. 從 Amazon Redshift 主控台或查詢編輯器 v2，從您的整合建立 Amazon Redshift 資料庫。

   然後，查詢並建立具有複製資料的具體化視觀表。

如需建立 Aurora 零 ETL 整合的詳細資訊，請參閱《Amazon Aurora 使用者指南》中的建立與 Amazon Redshift 的 Amazon Aurora 零 ETL 整合。

建立 Amazon RDS 的零 ETL 整合

在此步驟中，您會建立與 Amazon Redshift 的 Amazon RDS 零 ETL 整合。Redshift 支援與 RDS for MySQL、RDS for PostgreSQL 和 RDS for Oracle 整合。

建立與 Amazon Redshift 的 Amazon RDS 零 ETL 整合

1. 如 Amazon RDS 使用者指南所述，從 Amazon RDS 主控台建立自訂資料庫參數群組。
2. 如 Amazon Aurora 使用者指南所述，從 Amazon RDS 主控台建立來源 Amazon RDS 執行個體。
3. 從 Amazon Redshift 主控台：建立和設定目標 Amazon Redshift 資料倉儲。
   - 從 AWS CLI 或 Amazon Redshift 主控台：為資料倉儲開啟區分大小寫功能。
   - 從 Amazon Redshift 主控台：設定 Amazon Redshift 資料倉儲的授權。
4. 如 Amazon RDS 使用者指南所述，從 Amazon RDS 主控台建立零 ETL 整合。
5. 從 Amazon Redshift 主控台或查詢編輯器 v2，從您的整合建立 Amazon Redshift 資料庫。

   然後，查詢並建立具有複製資料的具體化視觀表。

Amazon RDS 主控台提供逐步的整合建立流程，您可在其中指定來源資料庫和目標 Amazon Redshift 資料倉儲。如果發生問題，您可以選擇讓 Amazon RDS 為您修正問題，而不必在 Amazon RDS 或 Amazon Redshift 主控台上手動修復問題。

如需建立 RDS 零 ETL 整合的詳細指示，請參閱《Amazon RDS 使用者指南》中的建立與 Amazon Redshift 的 Amazon RDS 零 ETL 整合。

如需專門建立 Amazon RDS for Oracle 零 ETL 整合的詳細指示，請參閱《Oracle Database@AWS 使用者指南》中的設定零 ETL 整合。

建立 DynamoDB 的零 ETL 整合

在建立零 ETL 整合之前，請先檢閱 將零 ETL 整合與 Amazon Redshift 搭配使用的考量 中所述的考量事項和需求。依照此一般流程進行，以建立從 DynamoDB 到 Amazon Redshift 的零 ETL 整合

透過零 ETL 整合將 DynamoDB 資料複寫至 Amazon Redshift

1. 確認您的登入憑證允許使用零 ETL 整合搭配 Amazon Redshift 和 DynamoDB 的許可。請參閱 要搭配 DynamoDB 零 ETL 整合使用的 IAM 政策 中的範例 IAM 政策。

2. 從 DynamoDB 主控台，設定您的 DynamoDB 資料表，使其具有時間點復原 (PITR)、資源政策、身分型政策和加密金鑰許可，如《Amazon DynamoDB 開發人員指南》中所述。
3. 從 Amazon Redshift 主控台：建立和設定目標 Amazon Redshift 資料倉儲。
   - 從 AWS CLI 或 Amazon Redshift 主控台：為資料倉儲開啟區分大小寫功能。
   - 從 Amazon Redshift 主控台：設定 Amazon Redshift 資料倉儲的授權。
4. 從 Amazon Redshift 主控台建立零 ETL 整合，如本主題稍後所述。
5. 從 Amazon Redshift 主控台，在您的 Amazon Redshift 資料倉儲中建立目的地資料庫。如需詳細資訊，請參閱在 Amazon Redshift 中建立目的地資料庫。
6. 從 Amazon Redshift 主控台，查詢 Amazon Redshift 資料倉儲中的複寫資料。如需詳細資訊，請參閱在 Amazon Redshift 中查詢複寫的資料。

在此步驟中，您會建立與 Amazon Redshift 的 Amazon DynamoDB 零 ETL 整合。

Amazon Redshift console

使用 Amazon Redshift 主控台建立與 Amazon Redshift 的 Amazon DynamoDB 零 ETL 整合

1. 從 Amazon Redshift 主控台，選擇零 ETL 整合。在列出零 ETL 整合的窗格中，選擇建立零 ETL 整合、建立 DynamoDB 整合。
2. 在建立整合的頁面上，輸入整合的相關資訊，如下所示：
   - 輸入整合名稱 - 這是可用來參考整合的唯一名稱。
   - 輸入說明 - 說明要從來源複寫到目標的資料。
   - 選擇 DynamoDB 來源資料表 - 可以選擇一個 DynamoDB 資料表。必須在資料表上啟用時間點復原 (PITR)。只會顯示資料表大小不超過上限 100 TB (TiB) 的資料表。來源 DynamoDB 資料表必須加密。來源也必須具有包含授權主體和整合來源的資源政策。如果這些政策不正確，您會看到為我修正選項。
   - 選擇目標 Amazon Redshift 資料倉儲 - 資料倉儲可以是 Amazon Redshift 佈建叢集或 Redshift Serverless 工作群組。如果您的目標 Amazon Redshift 位於相同的帳戶中，您就可以選取目標。如果目標位於不同的帳戶中，您可以指定 Redshift 資料倉儲 ARN。目標必須具有包含授權主體和整合來源的資源政策，且 `enable_case_sensitive_identifier` 參數設定為 true。如果您的目標上沒有正確的資源政策，且您的目標位於相同的帳戶中，您可以選取為我修正選項，以便在建立整合程序期間自動套用資源政策。如果您的目標位於不同的 AWS 帳戶中，則須在 Amazon Redshift 倉儲上手動套用資源政策。如果您的目標 Amazon Redshift 資料倉儲未將正確的參數群組選項

enable_case_sensitive_identifier 設定為 true，您可以選取為我修正選項，以在建立整合的過程中自動更新此參數群組並重新啟動倉儲。

- 最多可輸入 50 個標籤索引鍵和選用的值 - 提供有關整合的其他中繼資料。如需詳細資訊，請參閱在 Amazon Redshift 中標記資源。
- 選擇加密選項 - 加密整合。如需詳細資訊，請參閱使用客戶自管金鑰加密 DynamoDB 整合。

當您加密整合時，您也可以新增其他加密內容。如需詳細資訊，請參閱加密內容。

3. 此時會出現檢閱頁面，您可以在此選擇建立 DynamoDB 整合。
4. 此時會出現進度頁面，您可以在此檢視建立零 ETL 整合的各種不同任務的進度。
5. 整合建立並處於作用中狀態後，在整合的詳細資訊頁面上，選擇連線至資料庫。若是第一次建立 Amazon Redshift 資料倉儲，則也會建立資料庫。您需要連線到目標資料倉儲中的任何資料庫，才能為整合建立另一個資料庫。在連線至資料庫頁面中，判斷您是否可以使用最近的連線，並選擇身分驗證方法。根據您的身分驗證方法，輸入資訊以連線至目標中的現有資料庫。此身分驗證資訊可包含現有的資料庫名稱 (通常是 dev)，以及使用 Amazon Redshift 資料倉儲建立資料庫時指定的資料庫使用者。
6. 連線至資料庫之後，選擇從整合建立資料庫，即可建立資料庫，以便從來源接收資料。當您建立資料庫時，您會提供整合 ID、資料倉儲名稱和資料庫名稱。
7. 整合狀態和目的地資料庫為 Active 之後，資料就會開始從 DynamoDB 資料表複寫至目標資料表。隨著您新增資料至來源，資料會自動複寫到目標 Amazon Redshift 資料倉儲。

AWS CLI

若要使用 建立與 Amazon Redshift 的 Amazon DynamoDB 零 ETL 整合 AWS CLI，請使用 create-integration命令搭配下列選項：

- integration-name - 指定整合的名稱。
- source-arn - 指定 DynamoDB 來源的 ARN。
- target-arn - 指定 Amazon Redshift 佈建叢集或 Redshift Serverless 工作群組目標的命名空間 ARN。

下列範例藉由提供整合名稱、來源 ARN 和目標 ARN 來建立整合。整合不會經過加密。

```
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"
}
```

下列範例會使用客戶自管金鑰進行加密來建立整合。建立整合之前：

- 在來源 DynamoDB 資料表的相同帳戶 (範例中稱為 "AccountA") 中建立客戶自管金鑰 (範例中稱為 "CMCMK")。
- 確定用於建立整合的使用者/角色 (範例中稱為 "RoleA") 具有此 KMS 金鑰的 `kms:CreateGrant` 和 `kms:DescribeKey` 許可。
- 將以下各項新增至金鑰政策。

```
{
    "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": []
}
```

要搭配 DynamoDB 零 ETL 整合使用的 IAM 政策

建立零 ETL 整合時，您的登入憑證必須同時具有 DynamoDB 和 Amazon Redshift 動作的許可，以及具有作為整合來源和目標之資源的許可。以下範例示範所需的最低許可。

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>"
        }
    ]
}
```

使用客戶自管金鑰加密 DynamoDB 整合

如果您在建立 DynamoDB 零 ETL 整合 AWS 擁有的金鑰 時指定自訂 KMS 金鑰，而不是 ，則金鑰政策必須提供 Amazon Redshift 服務主體對 CreateGrant動作的存取權。此外，政策必須允許請求者帳戶或角色許可執行 DescribeKey 和 CreateGrant 動作。

下列範例金鑰政策陳述式示範政策中所需的許可。有些範例包含內容索引鍵，可進一步縮減許可範圍。

範例金鑰政策陳述式

下列政策陳述式允許請求者帳戶或角色擷取 KMS 金鑰的相關資訊。

```
{
   "Effect":"Allow",
   "Principal":{
      "AWS":"arn:aws:iam::{account-ID}:role/{role-name}"
   },
   "Action":"kms:DescribeKey",
   "Resource":"*"
}
```

下列政策陳述式允許請求者帳戶或角色新增授權至 KMS 金鑰。kms:ViaService 條件金鑰會將 KMS 金鑰的使用限於來自 Amazon Redshift 的請求。

```
{
   "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"
         ]
      }
   }
}
```

下列政策陳述式允許 Amazon Redshift 服務主體將授權新增至 KMS 金鑰。

```
{
   "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:*"
      }
   }
}
```

如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的建立金鑰政策。

加密內容

當您加密零 ETL 整合時，您可以將金鑰-值對新增為其他加密內容。您可能希望新增這些金鑰-值對，以新增有關複寫資料的其他內容資訊。如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的加密內容。

除了您新增的任何項目之外，Amazon Redshift 還會新增下列加密內容對：

- `aws:redshift:integration:arn` - `IntegrationArn`
- `aws:servicename:id` - `Redshift`

這會將您可以新增的整體配對數量從 8 減少為 6，並對授予限制的整體字元限制造成影響。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的使用授予限制條件。

建立與應用程式的零 ETL 整合

在此步驟中，您會建立與 Amazon Redshift 的應用程式零 ETL 整合。

建立與 Amazon Redshift 的應用程式零 ETL 整合

1. 從 Amazon Redshift 主控台：建立和設定目標 Amazon Redshift 資料倉儲。
   - 從 AWS CLI 或 Amazon Redshift 主控台：為資料倉儲開啟區分大小寫功能。
   - 從 Amazon Redshift 主控台：設定 Amazon Redshift 資料倉儲的授權。
2. 從 AWS Glue 主控台：如 AWS Glue 開發人員指南中所述建立 整合。
3. 完成建立目的地資料庫並開始複寫資料之後，您可以查詢並建立複寫資料的具體化資料。如需詳細資訊，請參閱在 Amazon Redshift 中查詢複寫的資料。

如需建立與應用程式的零 ETL 整合的詳細資訊，請參閱《AWS Glue 開發人員指南》中的零 ETL 整合。

在 Amazon Redshift 中建立目的地資料庫

若要將資料從來源複製到 Amazon Redshift，您必須從 Amazon Redshift 中的整合建立資料庫。

請連線至您的目標 Redshift Serverless 工作群組或佈建叢集，並建立參考整合識別符的資料庫。此識別碼是當您查詢 SVV_INTEGRATION 檢視時針對 `integration_id` 所傳回的值。

> **Important**
>
> 從您的整合建立資料庫之前，您必須先在 Amazon Redshift 主控台上建立零 ETL 整合，且整合必須處於 `Active` 狀態。

在將資料從來源複製到 Amazon Redshift 之前，您必須從 Amazon Redshift 中的整合建立資料庫。您可以使用 Amazon Redshift 主控台或查詢編輯器 v2 來建立資料庫。

Amazon Redshift console

1. 在左側導覽窗格中，選擇零 ETL 整合。
2. 從整合清單中選擇整合。
3. 如果您要使用佈建叢集，則必須先連線至資料庫。選擇 Connect to database (連接至資料庫)。您可以透過使用最近的連線或建立新連線來進行連線。
4. 若要從整合建立資料庫，請選擇從整合建立資料庫。
5. 輸入目的地資料庫名稱。整合 ID 和資料倉儲名稱會預先填入。

   針對 Aurora PostgreSQL、RDS for PostgreSQL、RDS for Oracle 或 Oracle Database@AWS 來源，輸入您在建立零 ETL 整合時指定的來源具名資料庫。在上述情況下，您最多可以將 100 個來源資料庫對應至 Amazon Redshift 資料庫。
6. 選擇建立資料庫。

Amazon Redshift query editor v2

1. 導覽至 Amazon Redshift 主控台，然後選擇查詢編輯器 v2。
2. 在左側面板中選擇 Amazon Redshift Serverless 工作群組或 Amazon Redshift 佈建叢集，然後與其連線。
3. 若要取得整合 ID，請導覽至 Amazon Redshift 主控台上的整合清單。

   或者，執行下列命令來取得 `integration_id` 值：

   ```
   SELECT integration_id FROM SVV_INTEGRATION;
   ```
4. 然後，執行下列命令以建立資料庫。透過指定整合 ID，您可以在資料庫與來源之間建立連線。

   以上一個命令傳回的值替代 `integration_id`。

```
CREATE DATABASE destination_db_name FROM INTEGRATION 'integration_id';
```

對於 Aurora PostgreSQL、RDS for PostgreSQL、RDS for Oracle 或 Oracle Database@AWS 來源，您還必須在建立整合時指定的來源執行個體或叢集中包含具名資料庫的參考。例如：

```
CREATE DATABASE "destination_db_name" FROM INTEGRATION 'integration_id'
 DATABASE "named_db";
```

如需建立資料庫作為零 ETL 整合目標的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE DATABASE。您可以使用 ALTER DATBASE 變更資料庫參數，例如 REFRESH INTERVAL。如需更改零 ETL 整合目標資料庫的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 ALTER DATABASE。

> **Note**
>
> 只有您的整合來源可以更新您從整合建立的資料庫中的資料。若要變更資料表的結構定義，請針對來源中的資料表執行 DDL 或 DML 命令。您可以針對來源中的資料表執行 DDL 和 DML 命令，但您只能在目的地資料庫上執行 DDL 命令和唯讀查詢。

如需檢視目的地資料庫狀態的相關資訊，請參閱 檢視零 ETL 整合。

在建立目的地資料庫後，您可以將資料新增至來源。若要將資料新增至來源，請參閱下列其中一個主題：

- 對於 Aurora 來源，請參閱 Amazon Aurora 使用者指南中的將資料新增至來源資料庫叢集。
- 對於 Amazon RDS 來源，請參閱 Amazon RDS 使用者指南中的將資料新增到來源資料庫執行個體。
- 對於 DynamoDB 來源，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 入門。
- 對於與應用程式的零 ETL 整合來源，請參閱《AWS Glue 開發人員指南》中的零 ETL 整合。

在 Amazon Redshift 中查詢複寫的資料

在將資料新增到來源後，資料會近乎即時地複寫到 Amazon Redshift 資料倉儲，並且可供查詢。如需有關整合指標和表格統計資訊的資訊，請參閱零 ETL 整合的指標。

> **Note**
>
> 由於資料庫與 MySQL 中的結構描述相同，因此 MySQL 資料庫層級會映射至 Amazon Redshift 結構描述層級。當您查詢從 Amazon Aurora MySQL 或 RDS for MySQL 複寫過來的資料時，請注意這個映射上的差異。

查詢複寫的資料

1. 導覽至 Amazon Redshift 主控台，然後選擇查詢編輯器 v2。
2. 連線至您的 Amazon Redshift Serverless 工作群組或 Amazon Redshift 佈建叢集，然後從下拉式清單中選擇您的資料庫。
3. 使用 SELECT 陳述式從您在來源中建立的結構描述和資料表選取所有資料：如需區分大小寫，請針對結構描述、資料表和資料欄名稱使用雙引號 (" ")。例如：

   ```
   SELECT * FROM "schema_name"."table_name";
   ```

   您也可以使用 Amazon Redshift Data API 來查詢資料。

透過具體化視觀表查詢複寫的資料

您可以在本機 Amazon Redshift 資料庫中建立具體化視觀表，以轉換透過零 ETL 整合所複寫的資料。請連線至本機資料庫，並使用跨資料庫查詢來存取目的地資料庫。您可以使用由三個部分組成的標記法 (destination-database-name.schema-name.table-name) 所表示的完整物件名稱，或建立參考目的地資料庫結構描述配對的外部結構描述，並使用由兩個部分組成的標記法 (external-schema-name.table-name)。如需跨資料庫查詢的相關資訊，請參閱跨資料庫查詢資料。

使用下列範例，從來源 *tickit_zetl* 建立範例資料，並將其插入至 *sales_zetl* 和 *event_zetl* 資料表。這兩個資料表會複寫到 Amazon Redshift 資料庫 *zetl_int_db*。

```
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");
```

您可以使用由三個部分組成的標記法建立具體化視觀表，以取得每個活動的總銷售額：

```
--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);
```

您可以使用由兩個部分組成的標記法建立具體化視觀表，以取得每個活動的總銷售額：

```
--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
);
```

若要檢視您建立的具體化視觀表，請使用下列範例。

```
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        |
+-----------+-------------+
```

查詢從 DynamoDB 複寫的資料

當您將資料從 Amazon DynamoDB 複寫至 Amazon Redshift 資料庫時，資料會儲存在 SUPER 資料類型欄的具體化視觀表中。

在此範例中，下列資料會儲存在 DynamoDB 中。

```
{
    "key1": {
        "S": "key_1"
    },
    "key2": {
        "N": 0
    },
    "payload": {
        "L": [
            {
                "S": "sale1"
            },
            {
                "S": "sale2"
            },
        ]
    },
}
```

Amazon Redshift 具體化視觀表的定義如下。

```
CREATE MATERIALIZED VIEW mv_sales
                    BACKUP NO
                    AUTO REFRESH YES
                    AS
                    SELECT "value"."payload"."L"[0]."S"::VARCHAR AS first_payload
                    FROM public.sales;
```

若要檢視具體化視觀表中的資料，請執行 SQL 命令。

```
SELECT first_payload FROM mv_sales;
```

檢視零 ETL 整合

您可以從 Amazon Redshift 主控台檢視零 ETL 整合。您可以在此檢視其組態資訊和目前狀態，並開啟查詢和共用資料的畫面。

Amazon Redshift console

檢視零 ETL 整合的詳細資訊

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 從左側導覽窗格選擇無伺服器或佈建叢集儀表板。然後，選擇零 ETL 整合。
3. 選取您要檢視的零 ETL 整合。針對每個整合，系統會提供以下資訊：
   - 整合 ID 是建立整合時傳回的識別碼。
   - 狀態可以是下列其中之一：
     - `Active` - 零 ETL 整合正在將交易資料傳送至目標 Amazon Redshift 資料倉儲。
     - `Syncing` - 零 ETL 整合遇到可復原的錯誤，且正在重新植入資料。在完成重新同步之前，受影響的資料表無法在 Amazon Redshift 中進行查詢。
     - `Failed` - 零 ETL 整合發生無法復原的事件或無法修正的錯誤。您必須刪除並重新建立零 ETL 整合。
     - `Creating` - 正在建立零 ETL 整合。
     - `Deleting` - 正在刪除零 ETL 整合。

    - `Needs attention` - 零 ETL 整合發生事件或錯誤，需要手動介入才能解決此問題。若要修正問題，請遵循錯誤訊息中的步驟。
- 來源類型是複寫至目標的來源資料類型。類型可以指定其他資料庫管理員，例如 Aurora MySQL-Compatible Edition、Amazon Aurora PostgreSQL、RDS for MySQL，以及來自應用程式 (`GlueSAAS`)。
- 來源 ARN 是來源資料的 ARN。對於大多數來源，這是來源資料庫或資料表的 ARN。對於與應用程式來源的零 ETL 整合，這是 AWS Glue 連線物件的 ARN。
- 目標是接收來源資料的 Amazon Redshift 資料倉儲命名空間。
- 資料庫可以是下列其中之一：
    - `No database` — 沒有用於整合的目的地資料庫。
    - `Creating` — Amazon Redshift 正在建立用於整合的目的地資料庫。
    - `Active` – 正在將資料從整合來源複寫到 Amazon Redshift。
    - `Error` — 整合發生錯誤。
    - `Recovering` — 重新啟動資料倉儲後，整合正在復原。
    - `Resyncing` — Amazon Redshift 正在重新同步整合中的資料表。
- 目標類型是 Amazon Redshift 資料倉儲的類型。
- 建立日期是建立整合時的日期和時間 (UTC)。

> **Note**
>
> 若要檢視資料倉儲的整合詳細資訊，請選擇已佈建叢集或無伺服器命名空間的詳細資料頁面，然後選擇 Zero-ETL 整合索引標籤。

從零 ETL 整合清單中，您可以選擇查詢資料以跳至 Amazon Redshift 查詢編輯器 v2。Amazon Redshift 目標資料庫已啟用 enable_case_sensitive_identifier 參數。當您撰寫 SQL 時，您可能需要使用雙引號括住結構描述、資料表和資料欄的名稱 ("<name>")。如需在 Amazon Redshift 資料倉儲中查詢資料的相關資訊，請參閱使用查詢編輯器 v2 查詢資料庫。

從零 ETL 整合清單中，您可以選擇共用資料以建立資料共用。遵循建立資料共用頁面上的指示，為 Amazon Redshift 資料庫建立資料共用。您必須先建立目的地資料庫，才能在 Amazon Redshift 資料庫中共用資料。如需資料共用的詳細資訊，請參閱 Amazon Redshift 的資料共用概念。

若要重新整理整合，您可以使用 ALTER DATABASE 命令。這樣做會將整合來源中的所有資料複寫到目的地資料庫中。下列範例會重新整理零 ETL 整合中所有已同步處理和失敗的資料表。

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH ALL tables;
```

AWS CLI

若要使用 描述與 Amazon Redshift 的 Amazon DynamoDB 零 ETL 整合 AWS CLI，請使用 `describe-integrations`命令搭配下列選項：

- `integration-arn` - 指定要說明的 DynamoDB 整合的 ARN。
- `integration-name` - 指定選用篩選條件，以指定一或多個要傳回的資源。

以下範例藉由提供整合 ARN 來說明整合。

```
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"
        }
    ]
}
```

您也可以依 integration-arn、source-arn、source-types 或 status 篩選 describe-integrations 的結果。如需詳細資訊，請參閱《Amazon Redshift CLI 指南》中的 describe-integrations。

歷史記錄模式

在歷史記錄模式中，您可以設定零 ETL 整合，直接在 Amazon Redshift 中追蹤來源資料表中記錄的每個版本 (包括更新和刪除)。您可以對所有資料執行進階分析，例如執行歷史分析、建置回顧報告、執行趨勢分析，以及將增量更新傳送至以 Amazon Redshift 為基礎建置的下游應用程式。有多種 Amazon Redshift 零 ETL 整合支援歷史記錄模式，包括 Amazon Aurora MySQL、Amazon Aurora PostgreSQL、Amazon RDS for MySQL 和 Amazon DynamoDB。另外也有數種應用程式支援歷史記錄模式，例如 Salesforce、SAP、ServiceNow 和 Zendesk。

您可以從 Amazon Redshift 主控台 (https://console.aws.amazon.com/redshiftv2/) 開啟和關閉零 ETL 整合的歷史記錄模式。使用歷史記錄模式追蹤整合來源中已刪除或已修改的記錄。追蹤是在目標 Amazon Redshift 資料倉儲中進行。開啟歷史記錄模式不會影響這些資料表上定期分析查詢的效能。

開啟歷史記錄模式後，您在來源內捨棄的資料表並不會在 Amazon Redshift 中捨棄。資料表會改為顯示 DroppedSource 狀態，而您仍然可以查詢這些資料表。您也可以搭配一般 SQL 使用 DROP 和 RENAME 命令。

如果您想要在來源上重複使用相同的資料表名稱，則必須先 DROP 或 RENAME 對應的 DroppedState 資料表，才能將其複寫至 Amazon Redshift。務必先完成上述操作，再於來源上建立資料表。

如需使用歷史記錄模式時應考量事項的相關資訊，請參閱 在目標上使用歷史記錄模式時的考量事項。

管理零 ETL 整合的歷史記錄模式

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 從左側導覽窗格選擇無伺服器或佈建叢集儀表板。然後，選擇零 ETL 整合。
3. 選取您要管理的零 ETL 整合，然後選擇管理歷史記錄模式。管理歷史記錄模式視窗隨即顯示。
4. 若目標資料表是從包含單一來源資料表 (例如 Amazon DynamoDB) 的來源類型複寫而來，則您可以關閉或開啟該資料表的歷史記錄模式。當零 ETL 整合有多個可能的目標資料表時，您可以對所有現有和未來的資料表關閉、對所有現有和未來的資料表開啟，或是管理個別資料表的歷史記錄模式。建立零 ETL 整合時，預設值為歷史記錄模式 off。

歷史記錄模式為 on 時，下列欄會新增至目標資料表，以便追蹤來源中的變更。歷史記錄模式 on 會增加每月用量和成本，因為 Amazon Redshift 不會刪除目標資料表中的任何記錄。任何已刪除或變更的來源記錄都會在目標中建立新記錄，導致目標中擁有多個記錄版本的總列數更多。在來源中刪除或修改記錄時，該記錄不會從目標資料表中刪除。您可以藉由刪除非作用中記錄來管理目標資料表。

欄名稱	資料類型	說明
_record_is_active	Boolean	指出目標中的記錄目前是否在來源中處於作用中狀態。True 表示記錄處於作用中狀態。
_record_create_time	時間戳記	來源記錄處於作用中狀態的開始時間 (UTC)。
_record_delete_time	時間戳記	來源記錄更新或刪除時的結束時間 (UTC)。

您可以藉由篩選欄 `_record_is_active` 為 false 的記錄，從歷史記錄模式資料表中刪除非作用中的記錄。下列 SQL DELETE 命令會從 id 欄小於或等於 100 的資料表中刪除非作用中的記錄。您刪除記錄後，在自動清空刪除執行時，就會回收已刪除記錄的儲存空間。

```
DELETE FROM myschema.mytable where not _record_is_active AND id <= 100;
```

歷史記錄模式為 `off` 時，Amazon Redshift 會複製目標資料庫中包含作用中記錄且沒有新增歷史記錄欄的資料表。Amazon Redshift 會將資料表重新命名為 *table-name*_historical_*timestamp* 以供您使用。如果您不再需要此資料表副本，可以將其捨棄。您可以使用 ALTER TABLE 命令重新命名這些資料表。例如：

```
ALTER TABLE [schema-name.]table-name_historical_timestamp RENAME TO new_table_name;
```

如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 ALTER TABLE。

您也可以使用 SQL 命令 CREATE DATABASE 和 ALTER DATABASE 來管理歷史記錄模式。如需如何設定 HISTORY_MODE 的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE DATABASE 和 ALTER DATABASE。

在 Amazon Redshift 中共用資料

在您將資料新增到來源之後，資料會立即複寫到 Amazon Redshift，並可透過建立資料共用來共用。

您必須先建立目的地資料庫才能共用資料。

使用 Amazon Redshift 主控台在 Amazon Redshift Serverless 中共用資料

1. 在 Amazon Redshift 主控台的左側導覽窗格中，選擇 Amazon Redshift Serverless > 無伺服器儀表板。
2. 從左側導覽窗格中，選擇零 ETL 整合。
3. 選擇 Share data (共用資料)。
4. 在建立資料共用頁面上，依照建立資料共用中的步驟執行。

使用 Amazon Redshift 主控台在 Amazon Redshift 佈建叢集中共用資料

1. 在 Amazon Redshift 主控台的左側導覽窗格中，選擇佈建叢集儀表板。
2. 從左側導覽窗格中，選擇零 ETL 整合。
3. 從整合清單中選擇整合。
4. 在整合詳細資訊頁面上，選擇連線至資料庫。
5. 在連線至資料庫頁面上，您可以建立新連線或使用最近的連線。請確定您已向目的地資料庫建立連線。
6. 如果您建立新的連線，請輸入資料庫的資料庫名稱。然後，按一下連線。
7. 在整合詳細資料頁面上，選擇共用資料。
8. 在建立資料共用頁面上，依照建立資料共用中的步驟執行。

監控零 ETL 整合

您可以藉由查詢系統檢視或透過 Amazon EventBridge 來監控零 ETL 整合。

監控與 Amazon Redshift 系統檢視的零 ETL 整合

您也可以透過在 Amazon Redshift 中查詢下列系統檢視來監控零 ETL 整合。

- SVV_INTEGRATION 會提供零 ETL 整合之組態詳細資訊的相關資訊。
- SYS_INTEGRATION_ACTIVITY 會提供已完成之零 ETL 整合的相關資訊。

- SVV_INTEGRATION_TABLE_MAPPING 提供有關將中繼資料值從來源對應至目標的資訊。
- SVV_INTEGRATION_TABLE_STATE 會提供整合狀態的相關資訊。
- SYS_INTEGRATION_TABLE_ACTIVITY 提供有關插入、刪除和更新整合活動的資訊。
- SYS_INTEGRATION_TABLE_STATE_CHANGE 提供有關整合的資料表狀態變更日誌的資訊。

監控與 Amazon EventBridge 的零 ETL 整合

Amazon Redshift 會將整合相關的事件傳送至 Amazon EventBridge。如需事件及其對應事件 ID 的清單，請參閱 使用 Amazon EventBridge 的零 ETL 整合事件通知。

零 ETL 整合的指標

您可以使用 Amazon Redshift 主控台和 Amazon CloudWatch 中的指標來了解零 ETL 整合的運作狀態和效能。您可以調整指標以顯示較短或較長持續時間的資料，也可以選擇在 CloudWatch 中檢視指標。若要在 Amazon Redshift 主控台上檢視整合的指標，請在左側導覽窗格中選擇零 ETL 整合，然後選擇您的整合 ID。

根據零 ETL 整合的來源資料而定，Amazon Redshift 會在整合詳細資料頁面上提供整合的指標。可能的指標包括下列類型：

- 整合指標索引標籤中提供下列圖形：

指標	Amazon Redshift 主控台中的指標名稱	說明
`IntegrationLag`	`Lag`	從資料遞交到您的來源的時間，到資料可供在 Amazon Redshift 中查詢的時間兩者之間的延遲。 單位：秒 維度：`IntegrationId` 更新頻率：60 秒
`IntegrationNumTablesReplicated`	`Tables replicated`	已從來源資料庫複寫到 Amazon Redshift 的資料表數量。

指標	Amazon Redshift 主控台中的指標名稱	說明
		單位：Count 維度：IntegrationId 更新頻率：60 秒
IntegrationNumTablesFailedReplication	Tables failed	複寫失敗的資料表數量。 單位：Count 維度：IntegrationId 更新頻率：60 秒
IntegrationDataTransferred	Data transferred	資料傳輸量 (以邏輯位元組為單位)。 單位：位元組 維度：IntegrationId 更新頻率：60 秒
IntegrationLatestDetectedChange	Latest detected change	整合上次在複寫佇列中暫存來源變更的時間，以 Unix epoch 秒為單位。 單位：秒 維度：IntegrationId 更新頻率：60 秒
IntegrationLatestAppliedChange	Latest applied change	整合上次在 Amazon Redshift 上完成擷取的時間，以 Unix epoch 秒為單位。 單位：秒 維度：IntegrationId 更新頻率：60 秒

指標	Amazon Redshift 主控台中的指標名稱	說明
IntegrationDuplicateRowsDetected	Duplicate rows detected	在報告間隔期間識別具有重複資料列的資料表數目。 單位：Count 維度：IntegrationId 更新頻率：60 秒
IntegrationAutoRemediationTriggered	Auto-remediation triggered	在報告間隔期間，自動修復移至重新同步的資料表數目。 單位：Count 維度：IntegrationId 更新頻率：60 秒

> **Note**
>
> 對於複寫多個資料庫的整合，IntegrationLatestDetectedChange和IntegrationLatestAppliedChange指標會報告所有資料庫的最小值。這代表最近更新過的資料庫。每個資料庫的值可在 SVV_INTEGRATION 的latest_detected_change_time和 latest_applied_change_time欄中使用。

- 您可以從資料表統計資訊索引標籤檢視目前作用中或發生錯誤的資料表清單。此索引標籤上的統計資料如下所示 (取決於來源類型)：
  - 結構描述名稱 — 資料表所在的結構描述名稱。
  - 資料表名稱 — 來源資料庫中的資料表名稱。
  - 狀態 — 資料表的狀態。可能的值包括 Synced、Failed、Deleted、Resync Required 及 Resync Initiated。
  - 資料庫 — 資料表所在的 Amazon Redshift 資料庫。
  - 上次更新 — 上次對資料表進行更新的日期和時間 (UTC)。
  - 資料表資料列計數 - 資料表中的列數。

- 資料表大小 - 資料表的大小。

您也可以檢視所選取時間範圍內插入、刪除和更新的列數圖。

修改 DynamoDB 的零 ETL 整合

在此步驟中，您會修改與 Amazon Redshift 的 DynamoDB 零 ETL 整合。

Amazon Redshift console

使用 Amazon Redshift 主控台修改與 Amazon Redshift 的 Amazon DynamoDB 零 ETL 整合

1. 從 Amazon Redshift 主控台，選擇零 ETL 整合。在列出零 ETL 整合的窗格中，選擇您要修改的 DynamoDB 整合。
2. 選擇編輯並修改整合名稱或說明。
3. 選擇儲存變更，以儲存您所做的變更。

AWS CLI

若要使用 修改與 Amazon Redshift 的 Amazon DynamoDB 零 ETL 整合 AWS CLI，請使用 `modify-integration`命令搭配下列選項：

- `integration-arn` - 指定要修改的 DynamoDB 整合的 ARN。
- `integration-name` - 指定整合的新名稱。
- `description` - 指定整合的新描述。

以下範例藉由提供整合 ARN、新的說明和新名稱來修改整合。

```
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": []
}
```

刪除 DynamoDB 的零 ETL 整合

當您刪除整合時，目標資料倉儲會保留任何先前複寫的資料。您可以繼續共用和查詢此資料。不過，來源中的新資料不會複寫到目標。

在此步驟中，您會刪除與 Amazon Redshift 的 DynamoDB 零 ETL 整合。

Amazon Redshift console

使用 Amazon Redshift 主控台刪除與 Amazon Redshift 的 Amazon DynamoDB 零 ETL 整合

1. 從 Amazon Redshift 主控台，選擇零 ETL 整合。在列出零 ETL 整合的窗格中，選擇您要刪除的 DynamoDB 整合。
2. 選擇刪除並提供請求的資訊。
3. 選擇刪除以刪除零 ETL 整合。

AWS CLI

若要使用 刪除與 Amazon Redshift 的 Amazon DynamoDB 零 ETL 整合 AWS CLI，請使用 `delete-integration`命令搭配下列選項：

- `integration-arn` - 指定要刪除的 DynamoDB 整合的 ARN。

以下範例藉由提供整合 ARN 來刪除整合。

```
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 整合的支援區域

零 ETL 整合是一種全受管解決方案，可在 Amazon Redshift 中從多個操作和交易來源以及企業應用程式提供交易和操作資料。此頁面列出每一種支援來源的可用區域。

Aurora MySQL

下列區域和引擎版本適用於與 Amazon Redshift 的 Aurora MySQL 零 ETL 整合。

區域	Aurora MySQL
非洲 (開普敦)	Available
亞太地區 (香港)	Available
亞太地區 (東京)	Available

區域	Aurora MySQL
亞太地區 (首爾)	Available
亞太地區 (大阪)	Available
亞太地區 (孟買)	Available
亞太地區 (海德拉巴)	Available
亞太地區 (新加坡)	Available
亞太地區 (悉尼)	Available
亞太地區 (雅加達)	Available
亞太地區 (墨爾本)	Available
亞太地區 (馬來西亞)	不適用
加拿大 (中部)	Available
加拿大西部 (卡加利)	Available
中國 (北京)	Available
中國 (寧夏)	Available
歐洲 (法蘭克福)	Available
歐洲 (蘇黎世)	Available
歐洲 (斯德哥爾摩)	Available
歐洲 (米蘭)	Available
歐洲 (西班牙)	Available
歐洲 (愛爾蘭)	Available
歐洲 (倫敦)	Available

區域	Aurora MySQL
Europe (Paris)	Available
以色列 (特拉維夫)	Available
中東 (阿拉伯聯合大公國)	Available
Middle East (Bahrain)	Available
南美洲 (聖保羅)	Available
美國東部 (維吉尼亞北部)	Available
美國東部 (俄亥俄)	Available
美國西部 (加利佛尼亞北部)	Available
美國西部 (奧勒岡)	Available
AWS GovCloud (US-East)	不適用
AWS GovCloud (US-West)	不適用

Aurora PostgreSQL

下列區域適用於與 Amazon Redshift 的 Aurora PostgreSQL 零 ETL 整合。

區域	Aurora PostgreSQL
非洲 (開普敦)	Available
亞太地區 (香港)	Available
亞太地區 (東京)	Available
亞太地區 (首爾)	Available
亞太地區 (大阪)	Available

區域	Aurora PostgreSQL
亞太地區 (孟買)	Available
亞太地區 (海德拉巴)	Available
亞太地區 (新加坡)	Available
亞太地區 (悉尼)	Available
亞太地區 (雅加達)	Available
亞太地區 (墨爾本)	Available
亞太地區 (馬來西亞)	Available
加拿大 (中部)	Available
加拿大西部 (卡加利)	Available
中國 (北京)	Available
中國 (寧夏)	Available
歐洲 (法蘭克福)	Available
歐洲 (蘇黎世)	Available
歐洲 (斯德哥爾摩)	Available
歐洲 (米蘭)	Available
歐洲 (西班牙)	Available
歐洲 (愛爾蘭)	Available
歐洲 (倫敦)	Available
Europe (Paris)	Available
以色列 (特拉維夫)	Available

區域	Aurora PostgreSQL
中東 (阿拉伯聯合大公國)	Available
Middle East (Bahrain)	Available
南美洲 (聖保羅)	Available
美國東部 (維吉尼亞北部)	Available
美國東部 (俄亥俄)	Available
美國西部 (加利佛尼亞北部)	Available
美國西部 (奧勒岡)	Available
AWS GovCloud (US-East)	不適用
AWS GovCloud (US-West)	不適用

Amazon DynamoDB

下列區域適用於與 Amazon Redshift 的 DynamoDB 零 ETL 整合。

區域	DynamoDB
非洲 (開普敦)	Available
亞太地區 (香港)	Available
亞太區域 (台北)	Available
亞太地區 (東京)	Available
亞太地區 (首爾)	Available
亞太地區 (大阪)	Available
亞太地區 (孟買)	Available

區域	DynamoDB
亞太地區 (海德拉巴)	Available
亞太地區 (新加坡)	Available
亞太地區 (悉尼)	Available
亞太地區 (雅加達)	Available
亞太地區 (墨爾本)	Available
亞太地區 (馬來西亞)	Available
亞太區域 (泰國)	Available
加拿大 (中部)	Available
加拿大西部 (卡加利)	Available
中國 (北京)	Available
中國 (寧夏)	Available
歐洲 (法蘭克福)	Available
歐洲 (蘇黎世)	Available
歐洲 (斯德哥爾摩)	Available
歐洲 (米蘭)	Available
歐洲 (西班牙)	Available
歐洲 (愛爾蘭)	Available
歐洲 (倫敦)	Available
Europe (Paris)	Available
以色列 (特拉維夫)	Available

區域	DynamoDB
中東 (阿拉伯聯合大公國)	Available
Middle East (Bahrain)	Available
墨西哥 (中部)	Available
南美洲 (聖保羅)	Available
美國東部 (維吉尼亞北部)	Available
美國東部 (俄亥俄)	Available
美國西部 (加利佛尼亞北部)	Available
美國西部 (奧勒岡)	Available
AWS GovCloud (US-East)	Available
AWS GovCloud (US-West)	Available

Amazon RDS for MySQL

下列區域可用於與 Amazon Redshift 的 Amazon RDS for MySQL 零 ETL 整合。

區域	RDS for MySQL
非洲 (開普敦)	Available
亞太地區 (香港)	Available
亞太地區 (東京)	Available
亞太地區 (首爾)	Available
亞太地區 (大阪)	Available
亞太地區 (孟買)	Available

區域	RDS for MySQL
亞太地區 (海德拉巴)	不適用
亞太地區 (新加坡)	Available
亞太地區 (悉尼)	Available
亞太地區 (雅加達)	不適用
亞太地區 (墨爾本)	不適用
亞太地區 (馬來西亞)	不適用
加拿大 (中部)	Available
加拿大西部 (卡加利)	不適用
中國 (北京)	不適用
中國 (寧夏)	不適用
歐洲 (法蘭克福)	Available
歐洲 (蘇黎世)	不適用
歐洲 (斯德哥爾摩)	Available
歐洲 (米蘭)	Available
歐洲 (西班牙)	不適用
歐洲 (愛爾蘭)	Available
歐洲 (倫敦)	Available
Europe (Paris)	Available
以色列 (特拉維夫)	不適用
中東 (阿拉伯聯合大公國)	不適用

區域	RDS for MySQL
Middle East (Bahrain)	Available
南美洲 (聖保羅)	Available
美國東部 (維吉尼亞北部)	Available
美國東部 (俄亥俄)	Available
美國西部 (加利佛尼亞北部)	Available
美國西部 (奧勒岡)	Available
AWS GovCloud (US-East)	不適用
AWS GovCloud (US-West)	不適用

企業應用程式

下列區域適用於與 Amazon Redshift 的企業應用程式零 ETL 整合。

區域	企業應用程式
非洲 (開普敦)	不適用
亞太地區 (香港)	Available
亞太地區 (東京)	Available
亞太地區 (首爾)	Available
亞太地區 (大阪)	不適用
亞太地區 (孟買)	不適用
亞太地區 (海德拉巴)	不適用
亞太地區 (新加坡)	Available

區域	企業應用程式
亞太地區 (悉尼)	Available
亞太地區 (雅加達)	不適用
亞太地區 (墨爾本)	不適用
亞太地區 (馬來西亞)	不適用
加拿大 (中部)	Available
加拿大西部 (卡加利)	不適用
中國 (北京)	不適用
中國 (寧夏)	不適用
歐洲 (法蘭克福)	Available
歐洲 (蘇黎世)	不適用
歐洲 (斯德哥爾摩)	Available
歐洲 (米蘭)	不適用
歐洲 (西班牙)	不適用
歐洲 (愛爾蘭)	Available
歐洲 (倫敦)	Available
Europe (Paris)	不適用
以色列 (特拉維夫)	不適用
中東 (阿拉伯聯合大公國)	不適用
Middle East (Bahrain)	不適用
南美洲 (聖保羅)	Available

區域	企業應用程式
美國東部 (維吉尼亞北部)	Available
美國東部 (俄亥俄)	Available
美國西部 (加利佛尼亞北部)	不適用
美國西部 (奧勒岡)	Available
AWS GovCloud (US-East)	不適用
AWS GovCloud (US-West)	不適用

對零 ETL 整合進行疑難排解

下列各節可協助您對使用零 ETL 整合遇到的問題進行故障診斷。

針對 Aurora MySQL 的零 ETL 整合進行故障診斷

使用下列資訊，針對 Aurora MySQL 的零 ETL 整合常見問題進行故障診斷。

主題

- 整合建立失敗
- 資料表沒有主索引鍵
- Aurora MySQL 資料表未複寫至 Amazon Redshift
- 資料表中有不支援的資料類型
- 資料操作語言命令失敗
- 資料來源之間的追蹤變更不相符
- 授權失敗
- 資料表數量超過 10 萬個或結構描述數量超過 4950 個
- Amazon Redshift 無法載入資料
- 工作群組參數的設定不正確
- 未建立資料庫來啟用零 ETL 整合
- 資料表處於需要重新同步處理或重新同步處理已起始狀態
- 整合延遲增長

整合建立失敗

如果零 ETL 整合建立失敗，則整合的狀態為 `Inactive`。請確定您的 Aurora 資料庫叢集符合下列敘述：

- 您已在 Amazon RDS 主控台中建立叢集。
- 您的來源 Aurora 資料庫叢集執行支援的版本。如需支援版本的清單，請參閱支援的區域和 Aurora 資料庫引擎，以與 Amazon Redshift 進行零 ETL 整合。若要驗證此問題，請前往叢集的組態索引標籤，然後檢查引擎版本。
- 您已正確設定叢集的 binlog 參數設定。如果您的 Aurora MySQL binlog 參數未正確設定或未與來源 Aurora 資料庫叢集相關聯，則建立會失敗。請參閱設定資料庫叢集參數。

此外，請確定您的 Amazon Redshift 資料倉儲符合下列敘述：

- 已開啟區分大小寫功能。請參閱 為資料倉儲開啟區分大小寫功能。
- 您已為命名空間新增正確的授權主體和整合來源。請參閱 設定 Amazon Redshift 資料倉儲的授權。

資料表沒有主索引鍵

在目的地資料庫中，一或多個資料表沒有主索引鍵，因此無法同步。

若要解決此問題，請前往整合詳細資訊頁面上的表格統計資訊索引標籤，或使用 SVV_INTEGRATION_TABLE_STATE 來檢視失敗的資料表。您可以在資料表中新增主索引鍵，Amazon Redshift 便會重新同步資料表。或者，雖不建議，但您也可以捨棄 Aurora 上的這些資料表，並建立有主索引鍵的資料表。如需詳細資訊，請參閱 Amazon Redshift 的資料表設計最佳實務。

Aurora MySQL 資料表未複寫至 Amazon Redshift

如果您沒有看到一或多個資料表反映在 Amazon Redshift 中，您可以執行下列命令來將其重新同步。將 *dbname* 取代為您的 Amazon Redshift 資料庫名稱。以及將 *table1* 和 *table2* 取代為要同步的資料表名稱。

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 ALTER DATABASE。

您的資料可能不會複寫，因為一或多個來源資料表沒有主索引鍵。Amazon Redshift 中的監控儀表板會將這些資料表的狀態顯示為 `Failed`，而整體零 ETL 整合的狀態會變更為 `Needs attention`。若

要解決此問題，您可以識別資料表中可成為主索引鍵的現有索引鍵，也可以新增合成主索引鍵。如需詳細解決方案，請參閱AWS 資料庫部落格中的建立與 Amazon Redshift 的 Amazon Aurora MySQL 或 Amazon RDS for MySQL 零 ETL 整合時，在沒有主索引鍵的情況下處理資料表。

此外，確認您的目標是 Amazon Redshift 叢集，如此叢集就不會暫停。

資料表中有不支援的資料類型

在您從 Amazon Redshift 整合建立的資料庫中，以及從 Aurora 資料庫叢集複寫資料的目的地資料庫中，一或多個資料表有不支援的資料類型，因此無法同步。

若要解決此問題，請前往整合詳細資訊頁面上的表格統計資訊索引標籤，或使用 SVV_INTEGRATION_TABLE_STATE 來檢視失敗的資料表。然後，移除這些資料表，並在 Amazon RDS 上重新建立新的資料表。如需有關不受支援資料類型的詳細資訊，請參閱 Amazon Aurora 使用者指南中的 Aurora 和 Amazon Redshift 資料庫之間的資料類型差異。

資料操作語言命令失敗

Amazon Redshift 無法在 Redshift 資料表上執行 DML 命令。若要解決此問題，請使用 SVV_INTEGRATION_TABLE_STATE 檢視失敗的資料表。Amazon Redshift 會自動重新同步資料表以解決此錯誤。

資料來源之間的追蹤變更不相符

當 Amazon Aurora 與 Amazon Redshift 之間的變更不相符，導致整合進入 `Failed` 狀態時，就會發生此錯誤。

若要解決此錯誤，請刪除零 ETL 整合，然後在 Amazon RDS 中重新建立整合。如需詳細資訊，請參閱建立零 ETL 整合和刪除零 ETL 整合。

授權失敗

授權失敗，因為做為 Amazon Redshift 資料倉儲授權整合來源的來源 Aurora 資料庫叢集已遭到移除。

若要解決此錯誤，請刪除零 ETL 整合，然後在 Amazon RDS 上重新建立整合。如需詳細資訊，請參閱建立零 ETL 整合和刪除零 ETL 整合。

資料表數量超過 10 萬個或結構描述數量超過 4950 個

對於目的地資料倉儲，資料表的數量超過 10 萬個或結構描述的數量超過 4950 個。Amazon Aurora 無法將資料傳送到 Amazon Redshift。資料表和結構描述的數量超過所設定的限制。若要解決此問題，請從來源資料庫中移除任何不必要的結構描述或資料表。

Amazon Redshift 無法載入資料

Amazon Redshift 無法將資料載入到零 ETL 整合。

若要解決此問題，請刪除 Amazon RDS 上的零 ETL 整合，然後重新建立整合。如需詳細資訊，請參閱建立零 ETL 整合和刪除零 ETL 整合。

工作群組參數的設定不正確

您的工作群組未開啟區分大小寫功能。

若要解決此問題，請前往整合詳細資訊頁面上的屬性索引標籤、選擇參數群組，然後從屬性索引標籤開啟區分大小寫的識別碼。如果您沒有現有的參數群組，請建立一個已開啟區分大小寫識別碼的參數群組。然後，在 Amazon RDS 上建立新的零 ETL 整合。如需詳細資訊，請參閱建立零 ETL 整合。

未建立資料庫來啟用零 ETL 整合

沒有建立可供零 ETL 整合來啟用的資料庫。

若要解決此問題，請為整合建立資料庫。如需詳細資訊，請參閱在 Amazon Redshift 中建立目的地資料庫。

資料表處於需要重新同步處理或重新同步處理已起始狀態

您的資料表處於需要重新同步處理或重新同步處理已起始狀態。

若要收集有關資料表為何處於該狀態的更詳細錯誤資訊，請使用 SYS_LOAD_ERROR_DETAIL 系統檢視。

整合延遲增長

如果來源資料庫中大量使用 SAVEPOINT，則零 ETL 整合的整合延遲可能會增加。

針對與 Aurora PostgreSQL 的零 ETL 整合進行故障診斷

使用下列資訊，針對 Aurora PostgreSQL 的零 ETL 整合常見問題進行故障診斷。

主題

- 整合建立失敗
- 資料表沒有主索引鍵

- Aurora PostgreSQL 資料表未複寫至 Amazon Redshift
- 資料表中有不支援的資料類型
- 資料操作語言命令失敗
- 資料來源之間的追蹤變更不相符
- 授權失敗
- 資料表數量超過 10 萬個或結構描述數量超過 4950 個
- Amazon Redshift 無法載入資料
- 工作群組參數的設定不正確
- 未建立資料庫來啟用零 ETL 整合
- 資料表處於需要重新同步處理或重新同步處理已起始狀態

整合建立失敗

如果零 ETL 整合建立失敗，則整合的狀態為 `Inactive`。請確定您的 Aurora 資料庫叢集符合下列敘述：

- 您已在 Amazon RDS 主控台中建立叢集。
- 您的來源 Aurora 資料庫叢集執行支援的版本。如需支援版本的清單，請參閱支援的區域和 Aurora 資料庫引擎，以與 Amazon Redshift 進行零 ETL 整合。若要驗證此問題，請前往叢集的組態索引標籤，然後檢查引擎版本。
- 您已正確設定叢集的 binlog 參數設定。如果您的Aurora PostgreSQL binlog 參數未正確設定或未與來源 Aurora 資料庫叢集相關聯，則建立會失敗。請參閱設定資料庫叢集參數。

此外，請確定您的 Amazon Redshift 資料倉儲符合下列敘述：

- 已開啟區分大小寫功能。請參閱 為資料倉儲開啟區分大小寫功能。
- 您為 your endterm="zero-etl-using.redshift-iam.title"/> 新增了正確的授權主體和整合來源。

資料表沒有主索引鍵

在目的地資料庫中，一或多個資料表沒有主索引鍵，因此無法同步。

若要解決此問題，請前往整合詳細資訊頁面上的表格統計資訊索引標籤，或使用 SVV_INTEGRATION_TABLE_STATE 來檢視失敗的資料表。您可以在資料表中新增主索引

鍵，Amazon Redshift 便會重新同步資料表。或者，雖不建議，但您也可以捨棄 Aurora 上的這些資料表，並建立有主索引鍵的資料表。如需詳細資訊，請參閱 Amazon Redshift 的資料表設計最佳實務。

Aurora PostgreSQL 資料表未複寫至 Amazon Redshift

如果您沒有看到一或多個資料表反映在 Amazon Redshift 中，您可以執行下列命令來將其重新同步。將 *dbname* 取代為您的 Amazon Redshift 資料庫名稱。以及將 *table1* 和 *table2* 取代為要同步的資料表名稱。

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 ALTER DATABASE。

您的資料可能不會複寫，因為一或多個來源資料表沒有主索引鍵。Amazon Redshift 中的監控儀表板會將這些資料表的狀態顯示為 `Failed`，而整體零 ETL 整合的狀態會變更為 `Needs attention`。若要解決此問題，您可以識別資料表中可成為主索引鍵的現有索引鍵，也可以新增合成主索引鍵。如需詳細解決方案，請參閱AWS 資料庫部落格中的建立與 Amazon Redshift 的 Amazon Aurora PostgreSQL 零 ETL 整合時，在沒有主索引鍵的情況下處理資料表。

此外，確認您的目標是 Amazon Redshift 叢集，如此叢集就不會暫停。

資料表中有不支援的資料類型

在您從 Amazon Redshift 整合建立的資料庫中，以及從 Aurora 資料庫叢集複寫資料的目的地資料庫中，一或多個資料表有不支援的資料類型，因此無法同步。

若要解決此問題，請前往整合詳細資訊頁面上的表格統計資訊索引標籤，或使用 SVV_INTEGRATION_TABLE_STATE 來檢視失敗的資料表。然後，移除這些資料表，並在 Amazon RDS 上重新建立新的資料表。如需有關不受支援資料類型的詳細資訊，請參閱 Amazon Aurora 使用者指南中的 Aurora 和 Amazon Redshift 資料庫之間的資料類型差異。

資料操作語言命令失敗

Amazon Redshift 無法在 Redshift 資料表上執行 DML 命令。若要解決此問題，請使用 SVV_INTEGRATION_TABLE_STATE 檢視失敗的資料表。Amazon Redshift 會自動重新同步資料表以解決此錯誤。

資料來源之間的追蹤變更不相符

當 Amazon Aurora 與 Amazon Redshift 之間的變更不相符，導致整合進入 `Failed` 狀態時，就會發生此錯誤。

若要解決此錯誤，請刪除零 ETL 整合，然後在 Amazon RDS 中重新建立整合。如需詳細資訊，請參閱建立零 ETL 整合和刪除零 ETL 整合。

授權失敗

授權失敗，因為做為 Amazon Redshift 資料倉儲授權整合來源的來源 Aurora 資料庫叢集已遭到移除。

若要解決此錯誤，請刪除零 ETL 整合，然後在 Amazon RDS 上重新建立整合。如需詳細資訊，請參閱建立零 ETL 整合和刪除零 ETL 整合。

資料表數量超過 10 萬個或結構描述數量超過 4950 個

對於目的地資料倉儲，資料表的數量超過 10 萬個或結構描述的數量超過 4950 個。Amazon Aurora 無法將資料傳送到 Amazon Redshift。資料表和結構描述的數量超過所設定的限制。若要解決此問題，請從來源資料庫中移除任何不必要的結構描述或資料表。

Amazon Redshift 無法載入資料

Amazon Redshift 無法將資料載入到零 ETL 整合。

若要解決此問題，請刪除 Amazon RDS 上的零 ETL 整合，然後重新建立整合。如需詳細資訊，請參閱建立零 ETL 整合和刪除零 ETL 整合。

工作群組參數的設定不正確

您的工作群組未開啟區分大小寫功能。

若要解決此問題，請前往整合詳細資訊頁面上的屬性索引標籤、選擇參數群組，然後從屬性索引標籤開啟區分大小寫的識別碼。如果您沒有現有的參數群組，請建立一個已開啟區分大小寫識別碼的參數群組。然後，在 Amazon RDS 上建立新的零 ETL 整合。如需詳細資訊，請參閱建立零 ETL 整合。

未建立資料庫來啟用零 ETL 整合

沒有建立可供零 ETL 整合來啟用的資料庫。

若要解決此問題，請為整合建立資料庫。如需詳細資訊，請參閱在 Amazon Redshift 中建立目的地資料庫。

資料表處於需要重新同步處理或重新同步處理已起始狀態

您的資料表處於需要重新同步處理或重新同步處理已起始狀態。

若要收集有關資料表為何處於該狀態的更詳細錯誤資訊，請使用 SYS_LOAD_ERROR_DETAIL 系統檢視。

針對 RDS for MySQL 的零 ETL 整合進行故障診斷

使用下列資訊，針對與 RDS for MySQL 的零 ETL 整合常見問題進行故障診斷。

主題

- 整合建立失敗
- 資料表沒有主索引鍵
- RDS for MySQL 資料表未複寫至 Amazon Redshift
- 資料表中有不支援的資料類型
- 資料操作語言命令失敗
- 資料來源之間的追蹤變更不相符
- 授權失敗
- 資料表數量超過 10 萬個或結構描述數量超過 4950 個
- Amazon Redshift 無法載入資料
- 工作群組參數的設定不正確
- 未建立資料庫來啟用零 ETL 整合
- 資料表處於需要重新同步處理或重新同步處理已起始狀態

整合建立失敗

如果零 ETL 整合建立失敗，則整合的狀態為 `Inactive`。請確定您的來源 RDS DB 執行個體符合下列敘述：

- 您已在 Amazon RDS 主控台中建立執行個體。
- 您的來源 RDS 資料庫執行個體執行支援的 RDS for MySQL 版本。如需支援版本的清單，請參閱支援的區域和資料庫引擎，以與 Amazon Redshift 進行 Amazon RDS 零 ETL 整合。若要驗證此問題，請前往執行個體的組態索引標籤，然後檢查引擎版本。
- 您已正確設定執行個體的 binlog 參數設定。如果您的 RDS for MySQL binlog 參數未正確設定或未與來源 RDS 資料庫執行個體相關聯，則建立會失敗。請參閱設定資料庫執行個體參數。

此外，請確定您的 Amazon Redshift 資料倉儲符合下列敘述：

- 已開啟區分大小寫功能。請參閱 為資料倉儲開啟區分大小寫功能。
- 您已為命名空間新增正確的授權主體和整合來源。請參閱 設定 Amazon Redshift 資料倉儲的授權。

資料表沒有主索引鍵

在目的地資料庫中，一或多個資料表沒有主索引鍵，因此無法同步。

若要解決此問題，請前往整合詳細資訊頁面上的表格統計資訊索引標籤，或使用 SVV_INTEGRATION_TABLE_STATE 來檢視失敗的資料表。您可以在資料表中新增主索引鍵，Amazon Redshift 便會重新同步資料表。或者，雖不建議，但您也可以捨棄 RDS 上的這些資料表，並建立有主索引鍵的資料表。如需詳細資訊，請參閱 Amazon Redshift 的資料表設計最佳實務。

RDS for MySQL 資料表未複寫至 Amazon Redshift

如果您沒有看到一或多個資料表反映在 Amazon Redshift 中，您可以執行下列命令來將其重新同步。將 *dbname* 取代為您的 Amazon Redshift 資料庫名稱。以及將 *table1* 和 *table2* 取代為要同步的資料表名稱。

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 ALTER DATABASE。

您的資料可能不會複寫，因為一或多個來源資料表沒有主索引鍵。Amazon Redshift 中的監控儀表板會將這些資料表的狀態顯示為 `Failed`，而整體零 ETL 整合的狀態會變更為 `Needs attention`。若要解決此問題，您可以識別資料表中可成為主索引鍵的現有索引鍵，也可以新增合成主索引鍵。如需詳細解決方案，請參閱AWS 資料庫部落格中的建立與 Amazon Redshift 的 Aurora MySQL-Compatible Edition 或 RDS for MySQL 零 ETL 整合時，在沒有主索引鍵的情況下處理資料表。

此外，確認您的目標是 Amazon Redshift 叢集，如此叢集就不會暫停。

資料表中有不支援的資料類型

在您於 Amazon Redshift 建立的目的地資料庫中，以及從 RDS 資料庫執行個體複寫資料的目的地資料庫中，一或多個資料表有不支援的資料類型，因此無法同步。

若要解決此問題，請前往整合詳細資訊頁面上的表格統計資訊索引標籤，或使用 SVV_INTEGRATION_TABLE_STATE 來檢視失敗的資料表。然後，移除這些資料表，並在 Amazon RDS 上重新建立新的資料表。如需有關不受支援資料類型的詳細資訊，請參閱 Amazon RDS 使用者指南中的 RDS 和 Amazon Redshift 資料庫之間的資料類型差異。

資料操作語言命令失敗

Amazon Redshift 無法在 Redshift 資料表上執行 DML 命令。若要解決此問題，請使用 SVV_INTEGRATION_TABLE_STATE 檢視失敗的資料表。Amazon Redshift 會自動重新同步資料表以解決此錯誤。

資料來源之間的追蹤變更不相符

當 Amazon Aurora 與 Amazon Redshift 之間的變更不相符，導致整合進入 `Failed` 狀態時，就會發生此錯誤。

若要解決此錯誤，請刪除零 ETL 整合，然後在 Amazon RDS 中重新建立整合。如需詳細資訊，請參閱建立零 ETL 整合和刪除零 ETL 整合。

授權失敗

授權失敗，因為做為 Amazon Redshift 資料倉儲授權整合來源的來源 RDS 資料庫執行個體已遭到移除。

若要解決此錯誤，請刪除零 ETL 整合，然後在 Amazon RDS 上重新建立整合。如需詳細資訊，請參閱建立零 ETL 整合和刪除零 ETL 整合。

資料表數量超過 10 萬個或結構描述數量超過 4950 個

對於目的地資料倉儲，資料表的數量超過 10 萬個或結構描述的數量超過 4950 個。Amazon Aurora 無法將資料傳送到 Amazon Redshift。資料表和結構描述的數量超過所設定的限制。若要解決此問題，請從來源資料庫中移除任何不必要的結構描述或資料表。

Amazon Redshift 無法載入資料

Amazon Redshift 無法將資料載入到零 ETL 整合。

若要解決此問題，請刪除 Amazon RDS 上的零 ETL 整合，然後重新建立整合。如需詳細資訊，請參閱建立零 ETL 整合和刪除零 ETL 整合。

工作群組參數的設定不正確

您的工作群組未開啟區分大小寫功能。

若要解決此問題，請前往整合詳細資訊頁面上的屬性索引標籤、選擇參數群組，然後從屬性索引標籤開啟區分大小寫的識別碼。如果您沒有現有的參數群組，請建立一個已開啟區分大小寫識別碼的參數群組。然後，在 Amazon RDS 上建立新的零 ETL 整合。如需詳細資訊，請參閱建立零 ETL 整合。

未建立資料庫來啟用零 ETL 整合

沒有建立可供零 ETL 整合來啟用的資料庫。

若要解決此問題，請為整合建立資料庫。如需詳細資訊，請參閱在 Amazon Redshift 中建立目的地資料庫。

資料表處於需要重新同步處理或重新同步處理已起始狀態

您的資料表處於需要重新同步處理或重新同步處理已起始狀態。

若要收集有關資料表為何處於該狀態的更詳細錯誤資訊，請使用 SYS_LOAD_ERROR_DETAIL 系統檢視。

針對 DynamoDB 的零 ETL 整合進行故障診斷

使用下列資訊，針對 Amazon DynamoDB 零 ETL 整合的常見問題進行故障診斷。

主題

- 整合建立失敗
- 資料表中有不支援的資料類型
- 不支援的資料表和屬性名稱
- 授權失敗
- Amazon Redshift 無法載入資料
- 工作群組或叢集參數設定不正確
- 未建立資料庫來啟用零 ETL 整合
- 來源 DynamoDB 資料表上未啟用時間點復原 (PITR)
- KMS 金鑰存取遭拒
- Amazon Redshift 無法存取 DynamoDB 資料表金鑰

整合建立失敗

如果零 ETL 整合建立失敗，則整合的狀態為 `Inactive`。請確定您的 Amazon Redshift 資料倉儲和來源 DynamoDB 資料表符合下列敘述：

- 已開啟資料倉儲的區分大小寫。請參閱《Amazon Redshift 管理指南》中的開啟區分大小寫。
- 您已為 Amazon Redshift 中的命名空間新增正確的授權主體和整合來源。請參閱《Amazon Redshift 管理指南》中的設定 Amazon Redshift 資料倉儲的授權。

- 您已將正確的資源型政策新增至來源 DynamoDB 資料表。請參閱《IAM 使用者指南》中的 IAM 中的政策和許可。

資料表中有不支援的資料類型

DynamoDB 號碼在 Amazon Redshift 中轉換為 DECIMAL(38,10。超過此精確度範圍的數字會自動轉換為 (38,10)。刪除整合並統一數字精確度，然後重新建立整合。

不支援的資料表和屬性名稱

Amazon Redshift 支援的資料表和屬性名稱上限為 127 個字元。如果 DynamoDB 資料表名稱或分割區索引鍵或排序索引鍵欄名稱等長名稱造成整合失敗，請使用較短的名稱修正，然後重新建立整合。

授權失敗

若作為 Amazon Redshift 資料倉儲授權整合來源的來源 DynamoDB 資料表遭到移除，則授權可能會失敗。

若要解決此問題，請刪除零 ETL 整合，然後使用 Amazon DynamoDB 重新建立整合。

Amazon Redshift 無法載入資料

Amazon Redshift 無法從到零 ETL 整合資料載入。

若要解決此問題，請使用 ALTER DATABASE 重新整理整合。

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH ALL TABLES
```

工作群組或叢集參數設定不正確

您的工作群組或叢集未開啟區分大小寫功能。

若要解決此問題，請前往整合詳細資訊頁面上的屬性索引標籤、選擇參數群組，然後從屬性索引標籤開啟區分大小寫的識別碼。如果您沒有現有的參數群組，請建立一個已開啟區分大小寫識別碼的參數群組。然後，在 DynamoDB 上建立新的零 ETL 整合。請參閱《Amazon Redshift 管理指南》中的開啟區分大小寫。

未建立資料庫來啟用零 ETL 整合

沒有建立可供零 ETL 整合來啟用的資料庫。

若要解決此問題，請為整合建立資料庫。請參閱《Amazon Redshift 管理指南》中的在 Amazon Redshift 中建立目的地資料庫。

來源 DynamoDB 資料表上未啟用時間點復原 (PITR)

必須啟用 PITR，DynamoDB 才能匯出資料。確認一律啟用 PITR。如果您在整合處於作用中狀態時關閉 PITR，您將需要依照錯誤訊息中的指示執行，並使用 ALTER DATABASE 重新整理整合。

```
ALTER DATABASE sample_integration_db INTEGRATION REFRESH ALL TABLES
```

KMS 金鑰存取遭拒

用於來源資料表或整合的 KMS 金鑰必須設有足夠的許可。如需資料表加密和解密的相關資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 DynamoDB 靜態加密。

Amazon Redshift 無法存取 DynamoDB 資料表金鑰

如果來源資料表加密是 AWS 受管金鑰，則切換到 AWS 擁有的金鑰 或客戶受管金鑰。如果已使用客戶自管金鑰加密資料表，請確認政策沒有任何條件索引鍵。

針對應用程式的零 ETL 整合進行故障診斷

使用下列資訊，針對 Salesforce、SAP、ServiceNow 和 Zendesk 等應用程式的零 ETL 整合常見問題進行故障診斷。

主題

- 整合建立失敗
- 資料表未複寫至 Amazon Redshift
- 資料表中有不支援的資料類型
- 工作群組參數的設定不正確
- 未建立資料庫來啟用零 ETL 整合
- 資料表處於需要重新同步處理或重新同步處理已起始狀態

整合建立失敗

如果零 ETL 整合建立失敗，則整合的狀態為 `Inactive`。請確定您的 Amazon Redshift 資料倉儲符合下列敘述：

- 已開啟區分大小寫功能。請參閱 為資料倉儲開啟區分大小寫功能。
- 您已為命名空間新增正確的授權主體和整合來源。請參閱 設定 Amazon Redshift 資料倉儲的授權。

資料表未複寫至 Amazon Redshift

在目的地資料庫中，一或多個資料表沒有主索引鍵，因此無法同步。

若要解決此問題，請前往整合詳細資訊頁面上的表格統計資訊索引標籤，或使用 SVV_INTEGRATION_TABLE_STATE 來檢視失敗的資料表。您可以在資料表中新增主索引鍵，Amazon Redshift 便會重新同步資料表。您可以執行下列命令來進行重新同步。將 *dbname* 取代為您的 Amazon Redshift 資料庫名稱。以及將 *table1* 和 *table2* 取代為要同步的資料表名稱。

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 ALTER DATABASE。

資料表中有不支援的資料類型

在您於 Amazon Redshift 中從整合建立，且其中資料是從應用程式的零 ETL 整合複寫而來的資料庫中，有一或多個資料表包含不支援的資料類型且無法同步。

若要解決此問題，請前往整合詳細資訊頁面上的表格統計資訊索引標籤，或使用 SVV_INTEGRATION_TABLE_STATE 來檢視失敗的資料表。然後，移除這些資料表，並在來源重新建立新的資料表。如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的零 ETL 整合。

工作群組參數的設定不正確

您的工作群組未開啟區分大小寫功能。

若要解決此問題，請前往整合詳細資訊頁面上的屬性索引標籤、選擇參數群組，然後從屬性索引標籤開啟區分大小寫的識別碼。如果您沒有現有的參數群組，請建立一個已開啟區分大小寫識別碼的參數群組。然後建立新的零 ETL 整合。如需詳細資訊，請參閱《AWS Glue 開發人員指南》中的零 ETL 整合。

未建立資料庫來啟用零 ETL 整合

沒有建立可供零 ETL 整合來啟用的資料庫。

若要解決此問題，請為整合建立資料庫。如需詳細資訊，請參閱在 Amazon Redshift 中建立目的地資料庫。

資料表處於需要重新同步處理或重新同步處理已起始狀態

您的資料表處於需要重新同步處理或重新同步處理已起始狀態。

若要收集有關資料表為何處於該狀態的更詳細錯誤資訊，請使用 SYS_LOAD_ERROR_DETAIL 系統檢視。

查詢資料庫

若要查詢由 Amazon Redshift 叢集所託管的資料庫，您有兩個選項：

- 連接至您的叢集，並使用查詢編輯器在 上執行 AWS 管理主控台 查詢。

  如果在 Amazon Redshift 主控台中使用查詢編輯器，您即不必下載和設定 SQL 用戶端應用程式。
- 透過 SQL 用戶端工具連接到您的叢集，例如 SQL Workbench/J。

  Amazon Redshift 可支援 SQL 用戶端工具透過 Java 資料庫連線 (JDBC) 和開放式資料庫連線 (ODBC) 進行連線。Amazon Redshift 不會提供或安裝任何 SQL 用戶端工具或程式庫，因此您必須在用戶端電腦或 Amazon EC2 執行個體上加以安裝才能使用。您可以使用支援 JDBC 或 ODBC 驅動程式的大多數 SQL 用戶端工具。

> **Note**
>
> 當您撰寫預存程序時，我們建議您採用保護敏感值的最佳作法：
> 不要在預存程序邏輯中對任何敏感資訊進行硬式編碼。例如，請勿在預存程序主體的 CREATE USER 陳述式中指派使用者密碼。這會造成安全性風險，因為硬式編碼值可以記錄為目錄資料表中的結構描述資料。應改為透過參數將敏感值 (例如密碼) 當做引數傳遞給預存程序。
> 如需預存程序的相關資訊，請參閱 CREATE PROCEDURE 和在 Amazon Redshift 中建立預存程序。如需目錄資料表的相關資訊，請參閱系統目錄資料表。

連線至 Amazon Redshift

您可以使用下列語法連線至資料庫。

```
cluster-name.account-number.aws-region.redshift.amazonaws.com/database-name
```

語法元素定義如下。

- `cluster-name`

  您的叢集名稱。
- `account-number`

與指定 中的 AWS 帳戶號碼相關聯的唯一識別符 AWS 區域。由特定 AWS 區域 中特定帳戶建立的所有叢集會擁有相同的 account-number。

- aws-region

  AWS 區域 叢集所在的 程式碼。

- database-name

  您的資料庫名稱。

例如，下列連線字串會在 us-east-1 中指定my-cluster叢集中的my-db資料庫 AWS 區域。

```
my-cluster.123456789012.us-east-1.redshift.amazonaws.com/my-db
```

使用查詢編輯器 v2 查詢資料庫

查詢編輯器 v2 是另外一種網頁式 SQL 用戶端應用程式，可用來在 Amazon Redshift 資料倉儲上編寫和執行查詢。查詢編輯器 v2 主要用來編輯和執行查詢、將查詢結果視覺化，以及與團隊共用您的工作。透過查詢編輯器 v2，您可以建立資料庫、結構描述、資料表和使用者定義的函數 (UDF)。在樹狀檢視面板中，您可以針對每個資料庫檢視其結構描述。對於每個結構描述，您可以檢視其資料表、檢視、UDF 和預存程序。查詢編輯器 v2 會取代先前的查詢編輯器。

> **Note**
>
> 查詢編輯器 v2 提供商業版本 AWS 區域。如需查詢編輯器 v2 可用 AWS 區域 位置的清單，請參閱《》中針對 Redshift 查詢編輯器 v2 列出的端點Amazon Web Services 一般參考。

如需查詢編輯器 v2 的示範，請觀看下列影片。Amazon Redshift 查詢編輯器 v2。

如需資料分析的示範，請觀看下列影片。使用 Amazon Redshift 查詢編輯器 v2 進行資料分析。

如需使用查詢編輯器 v2 透過隔離或共用連線來執行多個查詢的示範，請觀看下列影片。使用查詢編輯器 v2 執行並行查詢。

查詢編輯器 v2 有一組豐富的功能可供您管理和執行 SQL 陳述式。以下各節中的主題可協助您開始使用其中的許多功能。請自行探索查詢編輯器 v2 以熟悉其功能。

設定您的 AWS 帳戶

您可以執行這組任務來設定查詢編輯器 v2 來查詢 Amazon Redshift 資料庫。透過適當的許可，您可以存取 AWS 帳戶 目前 所擁有的 Amazon Redshift 叢集或工作群組中的資料 AWS 區域。

管理員第一次為您的 設定查詢編輯器 v2 時 AWS 帳戶，會選擇用來加密查詢編輯器 v2 資源 AWS KMS key 的 。根據預設， AWS 擁有的金鑰會用來加密資源。另外，管理員也可以在組態頁面中選擇金鑰的 Amazon Resource Name (ARN)，以使用客戶自管金鑰。

設定帳戶後，無法變更 AWS KMS 加密設定。如需以查詢編輯器 v2 建立和使用客戶受管金鑰的相關資訊，請參閱建立 AWS KMS 客戶受管金鑰以搭配查詢編輯器 v2 使用。管理員也可以選擇性地選擇用於某些功能 (例如，從檔案載入資料) 的 S3 儲存貯體和路徑。如需詳細資訊，請參閱從本機檔案設定和工作流程載入資料。

Amazon Redshift 查詢編輯器 v2 支援身分驗證、加密、隔離和合規功能，以確保靜態資料和傳輸中的資料安全無虞。如需資料安全和查詢編輯器 v2 的相關資訊，請參閱下列內容：

- 靜態加密
- 傳輸中加密
- Amazon Redshift 中的組態與漏洞分析

AWS CloudTrail 會擷取由 發出或代表發出的 API 呼叫和相關事件， AWS 帳戶 並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。您可以識別呼叫的使用者和帳戶 AWS、進行呼叫的來源 IP 地址，以及呼叫的時間。若要進一步了解查詢編輯器 v2 在 AWS CloudTrail上的執行方式，請參閱使用 CloudTrail 進行記錄。如需 CloudTrail 的相關資訊，請參閱《AWS CloudTrail 使用者指南》。

查詢編輯器 v2 對其某些資源具有可調整的配額。如需詳細資訊，請參閱Amazon Redshift 物件的配額。

使用查詢編輯器 v2 所建立的資源

在查詢編輯器 v2 內，您可以建立儲存的查詢和圖表等資源。查詢編輯器 v2 中的所有資源都會與 IAM 角色或使用者相關聯。建議您將政策連接至 IAM 角色，並將該角色指派給使用者。

在查詢編輯器 v2 中，您可以為儲存的查詢和圖表新增及移除標籤。您可以在設定自訂 IAM 政策或搜尋資源時使用這些標籤。您也可以使用標籤編輯器來管理 AWS Resource Groups 標籤。

您可以使用 IAM 政策設定 IAM 角色，在 中與相同 AWS 帳戶 中的其他人共用查詢 AWS 區域。

建立 AWS KMS 客戶受管金鑰以搭配查詢編輯器 v2 使用

若要建立對稱加密的客戶受管金鑰：

您可以使用 AWS KMS 主控台或 AWS KMS API 操作，建立對稱加密客戶受管金鑰來加密查詢編輯器 v2 資源。如需建立金鑰的指示，請參閱《 AWS Key Management Service 開發人員指南》中的建立對稱加密 AWS KMS 金鑰。

金鑰政策

金鑰政策會控制客戶受管金鑰的存取權限。每個客戶受管金鑰都必須只有一個金鑰政策，其中包含決定誰可以使用金鑰及其使用方式的陳述式。在建立客戶受管金鑰時，可以指定金鑰政策。如需詳細資訊，請參閱《 AWS Key Management Service 開發人員指南》中的管理對 AWS KMS 金鑰的存取。

若要將客戶受管金鑰與 Amazon Redshift 查詢編輯器 v2 搭配使用，就必須在金鑰政策中允許下列 API 操作：

- `kms:GenerateDataKey` — 產生唯一一個用來加密資料的對稱資料金鑰。
- `kms:Decrypt` — 解密使用客戶受管金鑰所加密的資料。
- `kms:DescribeKey` — 提供客戶受管金鑰的詳細資訊，以便服務可以驗證金鑰。

以下是 AWS 帳戶 的範例 AWS KMS 政策111122223333。在第一部分中，`kms:ViaService` 會限制金鑰只能用於查詢編輯器 v2 服務 (在政策中名為 `sqlworkbench.`*`region`*`.amazonaws.com`)。AWS 帳戶 使用 金鑰的 必須是 111122223333。在第二個區段中， 的 AWS 帳戶 根使用者和金鑰管理員111122223333可以存取金鑰。

當您建立 時 AWS 帳戶，您會從一個名為 AWS 帳戶 theroot 使用者的登入身分開始，該身分具有對所有 AWS 服務 和 資源的完整存取權。強烈建議不要使用根使用者來執行日常任務。有關需要根使用者憑證的任務，請參閱《IAM 使用者指南》中的需要根使用者憑證的任務。

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"
        }
    ]
}
```

下列資源提供有關 AWS KMS 金鑰的詳細資訊：

- 如需 AWS KMS 政策的詳細資訊，請參閱《 AWS Key Management Service 開發人員指南》中的在政策中指定許可。
- 如需有關故障診斷 AWS KMS 政策的資訊，請參閱《 AWS Key Management Service 開發人員指南》中的故障診斷金鑰存取。
- 如需金鑰的相關資訊，請參閱《AWS Key Management Service 開發人員指南》中的 AWS KMS 金鑰。

存取查詢編輯器 v2

若要存取查詢編輯器 v2，您需要獲得許可。管理員可以將下列其中一個 AWS 受管政策連接至角色，以授予許可。(建議您將政策連接至 IAM 角色，並將該角色指派給使用者。) 這些 AWS 受管政策是以不同的選項撰寫，可控制標記資源如何允許共用查詢。您可以使用 IAM 主控台 (https://console.aws.amazon.com/iam/) 來連接 IAM 政策。

- AmazonRedshiftQueryEditorV2FullAccess – 授予 Amazon Redshift 查詢編輯器 v2 操作和資源的完整存取權。此政策也會授予其他必要服務的存取權。
- AmazonRedshiftQueryEditorV2NoSharing – 授予使用 Amazon Redshift 查詢編輯器 v2 的能力，而不共用資源。此政策也會授予其他必要服務的存取權。
- AmazonRedshiftQueryEditorV2ReadSharing – 授予使用 Amazon Redshift 查詢編輯器 v2 的能力，並可有限度地共用資源。獲得授予的主體可以讀取與其團隊共用的資源，但無法更新這些資源。此政策也會授予其他必要服務的存取權。
- AmazonRedshiftQueryEditorV2ReadWriteSharing – 授予使用 Amazon Redshift 查詢編輯器 v2 的能力，並可共用資源。獲得授予的主體可以讀取和更新與其團隊共用的資源。此政策也會授予其他必要服務的存取權。

您也可以根據所提供受控政策中允許和拒絕的許可來建立自己的政策。如果您使用 IAM 主控台政策編輯器建立自己的政策，請選擇 SQL Workbench 作為您在視覺化編輯器中為其建立政策的服務。查詢編輯器 v2 會在視覺化編輯器和 IAM 政策模擬器中使用服務名稱 AWS SQL Workbench。

若要讓主體 (已獲派 IAM 角色的使用者) 能夠連線到 Amazon Redshift 叢集，就需要其中一個查詢編輯器 v2 受管政策中的許可。他們也需要叢集的其中一個 `redshift:GetClusterCredentialsWithIAM`或 `redshift:GetClusterCredentials`許可。若要獲得此許可，具有系統管理許可的人員可以使用臨時憑證，將政策連接至用來連線到叢集的 IAM 角色。您可以將政策的範圍限定在特定叢集或更為普遍的範圍。如需使用臨時登入資料之許可的詳細資訊，請參閱建立具有呼叫 GetClusterCredentialsWithIAM 或 GetClusterCredentials 許可的 IAM 角色或使用者 GetClusterCredentials。

若要讓主體 (一般是已獲派 IAM 角色的使用者) 能夠在帳戶設定頁面中為帳戶中的其他使用者開啟匯出結果集的功能，他們需要該角色連接的 `sqlworkbench:UpdateAccountExportSettings` 許可。此許可包含在 `AmazonRedshiftQueryEditorV2FullAccess` AWS 受管政策中。

隨著新功能新增至查詢編輯器 v2， AWS 受管政策會視需要更新。如果您根據所提供受管策略中允許和拒絕的許可建立自己的政策，請編輯您的政策，讓政策隨著受管政策的變更進行更新以保持在最新狀態。如需 Amazon Redshift 中受管政策的相關資訊，請參閱 AWS Amazon Redshift 的 受管政策。

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 中的使用者和群組 AWS IAM Identity Center：

  建立權限合集。請按照《AWS IAM Identity Center 使用者指南》中的建立權限合集說明進行操作。
- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。遵循《IAM 使用者指南》的為第三方身分提供者 (聯合) 建立角色中的指示。
- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照《IAM 使用者指南》的為 IAM 使用者建立角色中的指示。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循《IAM 使用者指南》的新增許可到使用者 (主控台) 中的指示。

> **Note**
>
> 如果 AWS IAM Identity Center 管理員移除整個帳戶中特定許可集的所有許可集關聯，則無法再存取原本與已移除許可集關聯的任何查詢編輯器資源。如果之後重新建立了相同的許可，系統會建立新的內部識別碼。由於內部識別碼已變更，因此使用者無法再存取其先前擁有之查詢編輯器資源的存取權。我們的建議是，在管理員刪除許可集之前，該許可集的使用者先將查詢編輯器資源 (例如筆記本和查詢) 匯出為備份。

設定主體標籤以從查詢編輯器 v2 連接叢集或工作群組

若要使用聯合身分使用者選項連線到叢集或工作群組，請使用主體標籤設定 IAM 角色或使用者。或者，也可以設定身分提供者 (IdP) 以傳入 `RedshiftDbUser` 和 (選擇性) `RedshiftDbGroups`。如需使用 IAM 來管理標籤的相關資訊，請參閱《IAM 使用者指南》中的在 AWS Security Token Service中傳遞工作階段標籤。若要使用 設定存取權 AWS Identity and Access Management，管理員可以使用 IAM 主控台 (https://console.aws.amazon.com/iam/：//) 新增標籤。

將主體標籤新增至 IAM 角色

1. 登入 AWS 管理主控台 並開啟位於 https：//https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 在導覽窗格中，選擇 Roles (角色)。
3. 選擇需要使用聯合身分使用者存取查詢編輯器 v2 的角色。
4. 選擇 Tags (標籤) 索引標籤。

5. 選擇管理標籤。
6. 選擇新增標籤，並以 RedshiftDbUser 形式輸入索引鍵，然後輸入聯合身分使用者名稱的值。
7. 選擇性地選擇新增標籤，並以 RedshiftDbGroups 形式輸入索引鍵，然後輸入要與使用者建立關聯之群組名稱的值。
8. 選擇儲存變更以檢視與所選擇的 IAM 角色相關聯的標籤清單。傳播變更的過程可能需要幾秒鐘的時間。
9. 若要使用聯合身分使用者，請在變更傳播完成後重新整理查詢編輯器 v2 的頁面。

設定您的身分提供者 (IdP) 以傳遞主體標籤

使用身分提供者 (IdP) 設定標籤的程序隨 IdP 而異。如需如何將使用者和群組資訊傳遞至 SAML 屬性的指示，請參閱您的 IdP 文件。正確設定時，下列屬性會出現在您的 SAML 回應中，供 AWS Security Token Service 用來填入 RedshiftDbUser和 的委託人標籤RedshiftDbGroups。

```
<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>
```

選擇性的 *db_group* 必須是以冒號分隔的清單，例如 group1:group2:group3。

此外，您也可以設定 TransitiveTagKeys 屬性以在鏈結角色期間保留標籤。

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys">
  <AttributeValue>RedshiftDbUser</AttributeValue>
  <AttributeValue>RedshiftDbGroups</AttributeValue>
</Attribute>
```

如需設定查詢編輯器 v2 的相關資訊，請參閱要使用查詢編輯器 v2 所需的許可。

> **Note**
>
> 當您使用查詢編輯器 v2 的聯合身分使用者連線選項連線到叢集或工作群組時，身分提供者 (IdP) 可以為 RedshiftDbUser 和 RedshiftDbGroups 提供自訂主體標籤。目前， AWS IAM Identity Center Dimers 不支援將自訂主體標籤直接傳遞至查詢編輯器 v2。

開啟查詢編輯器 v2

您可以透過 Amazon Redshift，於 Amazon Redshift 主控台中使用查詢編輯器 v2 對資料倉儲叢集執行 SQL 查詢。查詢編輯器 v2 是一種 Web 型工具，提供易用的介面來執行臨機查詢、探索資料，以及執行資料分析任務。下列各節將引導您完成在主控台中開啟查詢編輯器 v2 並有效利用其功能的程序。

開啟查詢編輯器 v2

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ ：// 開啟 Amazon Redshift 主控台。
2. 從導覽器選單中選擇編輯器，然後選擇查詢編輯器 V2。查詢編輯器 v2 便會在新的瀏覽器標籤中開啟。

查詢編輯器頁面會有一個導覽器選單可供您選擇檢視，如下所示：

編輯器



您可以管理和查詢組織為資料表並包含在資料庫中的資料。資料庫可以包含儲存的資料，也可以包含儲存在其他地方 (例如 Amazon S3) 之資料的參考。您可以連線到包含在叢集或無伺服器工作群組內的資料庫。

在編輯器檢視中工作時，您可以使用下列控制項：

- 叢集或工作群組欄位會顯示您目前所連線項目的名稱。資料庫欄位會顯示叢集或工作群組內的資料庫。您在資料庫檢視中執行的動作預設會對您選取的資料庫採取行動。
- 叢集或工作群組、資料庫和結構描述的樹狀檢視階層式檢視。在結構描述下，您可以處理您的資料表、檢視、函數和預存程序。樹狀檢視中的每個物件都支援可針對物件執行相關動作 (例如重新整理或捨棄) 的內容選單。
- 

  建立動作可建立資料庫、結構描述、資料表和函數。
- 

  載入資料動作可將資料從 Amazon S3 或從本機檔案載入到資料庫。
- 

  儲存圖示可儲存查詢。

- ⌘

  捷徑圖示可顯示編輯器的鍵盤快速鍵。

- •••

  更多圖示可在編輯器中顯示更多動作。例如：

  - 與我的團隊共用可與團隊共用查詢或筆記本。如需詳細資訊，請參閱以團隊的形式協作和共用。
  - 捷徑可顯示編輯器的鍵盤快速鍵。
  - 標籤歷史記錄可在編輯器中顯示標籤的歷史記錄。
  - 重新整理自動完成可在編寫 SQL 時重新整理所顯示的建議。

- ＋

  編輯器區域，可供您輸入和執行查詢。

  執行查詢後會出現結果標籤，內有查詢結果。你可以在此開啟圖表以將結果視覺化。您也可以匯出結果。

- ＋

  筆記本區域，可供您新增區段以便輸入和執行 SQL 或新增 Markdown。

  執行查詢後會出現結果標籤，內有查詢結果。您可以在此匯出結果。

查詢



查詢包含用來管理和查詢資料庫中所含資料的 SQL 命令。當您使用查詢編輯器 v2 載入範例資料時，其也會為您建立並儲存範例查詢。

當您選擇已儲存的查詢時，您可以使用內容 (右鍵) 選單來將其開啟、重新命名和刪除。您可以選擇查詢詳細資訊來檢視已儲存查詢的屬性，例如查詢 ARN。您還可以檢視其版本歷史記錄、編輯連接至查詢的標籤，以及將其與您的團隊共用。

筆記本



SQL 筆記本包含 SQL 和 Markdown 儲存格。使用筆記本可在單一文件中組織、註釋和共用多個 SQL 命令。

當您選擇已儲存的筆記本時，您可以使用內容 (右鍵) 選單來將其開啟、重新命名、複製和刪除。您可以選擇筆記本詳細資訊來檢視已儲存筆記本的屬性，例如筆記本 ARN。您還可以檢視其版本歷史記錄、編輯連接至筆記本的標籤、匯出筆記本，以及將其與您的團隊共用。如需詳細資訊，請參閱Amazon Redshift 中的筆記本。

圖表



圖表會以視覺化方式呈現資料。查詢編輯器 v2 提供了工具供您建立多種類型的圖表並將其儲存起來。

當您選擇已儲存的圖表時，您可以使用內容 (右鍵) 選單來將其開啟、重新命名和刪除。您可以選擇圖表詳細資訊來檢視已儲存圖表的屬性，例如圖表 ARN。您也可以編輯連接至圖表的標籤，並將其匯出。如需詳細資訊，請參閱將查詢結果視覺化。

歷史記錄



查詢歷史記錄會列出您使用 Amazon Redshift 查詢編輯器 v2 執行的查詢。這些查詢會以個別查詢的形式執行，或是作為 SQL 筆記本的一部分來執行。如需詳細資訊，請參閱檢視查詢和標籤歷史記錄。

已排程的查詢



已排程的查詢是設定為在特定時間啟動的查詢。

所有查詢編輯器 v2 檢視都有下列圖示：

- 

  視覺化模式圖示可在淺色模式和深色模式之間進行切換。

- 

  設定圖示可顯示不同設定畫面的選單。

  - 

    編輯器偏好設定圖示可讓您在使用查詢編輯器 v2 時編輯您的偏好設定。您可以在此編輯工作區設

定，以變更字型大小、標籤大小和其他顯示設定。您也可以開啟 (或關閉) 自動完成，以在您輸入 SQL 時顯示建議。

- 

  連線圖示可檢視編輯器標籤所使用的連線。

  連線可用來擷取資料庫中的資料。連線會針對特定資料庫來建立。使用隔離連線時，在某個編輯器標籤中會變更資料庫的 SQL 命令 (例如建立暫存資料表) 結果，並無法在另一個編輯器標籤中看到。當您在查詢編輯器 v2 中開啟編輯器標籤時，預設會使用隔離連線。當您建立共用連線時 (也就是如果您關閉隔離工作階段切換開關的話)，相同資料庫的共用連線彼此能看到其他共用連線的結果。但是，使用資料庫共用連線的編輯器標籤並不會平行執行。使用相同連線的查詢必須等到連線為可用狀態時才能執行。某個資料庫的連線不能與另一個資料庫共用，因此無法在不同的資料庫連線中看見 SQL 結果。

  帳戶中的任何使用者可以擁有的作用中連線數目會由查詢編輯器 v2 管理員控制。

- 

  帳戶設定圖示可讓管理員變更帳戶中所有使用者的某些設定。如需詳細資訊，請參閱帳戶設定。

使用查詢編輯器 v2 時的考量

在使用查詢編輯器 v2 時，請考量下列事項：

- 查詢的持續時間上限為 24 小時。
- 查詢結果大小上限為 100 MB。如果呼叫傳回超過 100 MB 的回應資料，則會傳回前 100 MB 並顯示警告。
- 您可以執行長度最多 300,000 個字元的查詢。
- 您可以儲存長度最多 30,000 個字元的查詢。
- 根據預設，查詢編輯器 v2 會自動遞交每個執行的個別 SQL 命令。在提供了 BEGIN 陳述式時，BEGIN-COMMIT 或 BEGIN-ROLLBACK 區塊內的陳述式會以單一交易的形式來執行。如需交易的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 BEGIN。
- 在執行 SQL 陳述式時，查詢編輯器 v2 可顯示的警告數量上限為 10 個。例如，在執行預存程序時，不會顯示超過 10 個 RAISE 陳述式。
- 查詢編輯器 v2 不支援包含逗號 (,) 的 IAM `RoleSessionName`。您可能會看到類似以下的錯誤：Error Message : "'AROA123456789EXAMPLE:mytext,yourtext' is not a valid value for TagValue - it contains illegal characters"。當您定義包含逗號的 IAM `RoleSessionName`，然後使用查詢編輯器 v2 搭配該 IAM 角色時，就會發生此問題。

如需 IAM RoleSessionName 的詳細資訊，請參閱《IAM 使用者指南》中的 RoleSessionName SAML 屬性。

帳戶設定

具有正確 IAM 許可的使用者可以檢視和變更相同 AWS 帳戶中其他使用者的帳戶設定。此管理員可以檢視或設定下列項目：

- 帳戶中每一使用者的最大並行資料庫連線數。這包括隔離工作階段的連線。當您變更此值時，其可能需要 10 分鐘才會生效。
- 允許帳戶中的使用者將 SQL 命令的整個結果集匯出至檔案。
- 載入並顯示包含一些相關聯之已儲存查詢的範例資料庫。
- 指定帳戶使用者用來從本機檔案載入資料的 Amazon S3 路徑。
- 檢視用來加密查詢編輯器 v2 資源的 KMS 金鑰 ARN。

連線至 Amazon Redshift 資料庫

若要連線至資料庫，請在樹狀檢視面板中選擇叢集或工作群組的名稱。若出現提示，請輸入連線參數。

在連線至叢集或工作群組及其資料庫時，通常會提供資料庫名稱。您也可以提供下列其中一種身分驗證方法所需的參數：

IAM Identity Center

若使用此方法，您可以使用身分提供者 (IdP) 提供的單一登入憑證連線到 Amazon Redshift 資料倉儲。您的叢集或工作群組必須針對 Amazon Redshift 主控台中的 IAM Identity Center 來啟用。如需協助您設定 IAM Identity Center 連線，請參閱 將 Redshift 與 AWS IAM Identity Center 連線，以獲得單一登入體驗。

聯合身分使用者

使用此方法時，IAM 角色或使用者的主體標籤必須提供連線詳細資訊。您可以在 AWS Identity and Access Management 或身分提供者 (IdP) 中設定這些標籤。查詢編輯器 v2 依賴以下標籤。

- `RedshiftDbUser` — 此標籤會定義查詢編輯器 v2 所使用的資料庫使用者。這是必要標籤。
- `RedshiftDbGroups` — 此標籤會定義在連線至查詢編輯器 v2 時所加入的資料庫群組。這是選擇性標籤，其值必須是以冒號分隔的清單，例如 `group1:group2:group3`。系統會忽略空白值，也就是說，會將 `group1::::group2` 解譯為 `group1:group2`。

這些標籤會轉送至 `redshift:GetClusterCredentials` API，以取得叢集的憑證。如需詳細資訊，請參閱[設定主體標籤以從查詢編輯器 v2 連接叢集或工作群組](#)。

使用資料庫使用者名稱的臨時憑證

只有在連線至叢集時才能使用此選項。使用此方法時，查詢編輯器 v2 會為資料庫提供使用者名稱。查詢編輯器 v2 會產生能以資料庫使用者名稱的身分連線到資料庫的臨時密碼。使用此方法來連線的使用者必須已被允許 `redshift:GetClusterCredentials` 的 IAM 許可。若要防止使用者使用此方法，請修改其 IAM 使用者或角色以拒絕此許可。

使用 IAM 身分的臨時憑證

只有在連線至叢集時才能使用此選項。使用此方法時，查詢編輯器 v2 會將使用者名稱對應到您的 IAM 身分，並產生能以 IAM 身分連線到資料庫的臨時密碼。使用此方法來連線的使用者必須已被允許 `redshift:GetClusterCredentialsWithIAM` 的 IAM 許可。若要防止使用者使用此方法，請修改其 IAM 使用者或角色以拒絕此許可。

資料庫使用者名稱和密碼

使用此方法時，也會為您要連線的資料庫提供使用者名稱和密碼。查詢編輯器 v2 會在 AWS Secrets Manager中代表您建立秘密。此秘密包含用來連線至資料庫的憑證。

AWS Secrets Manager

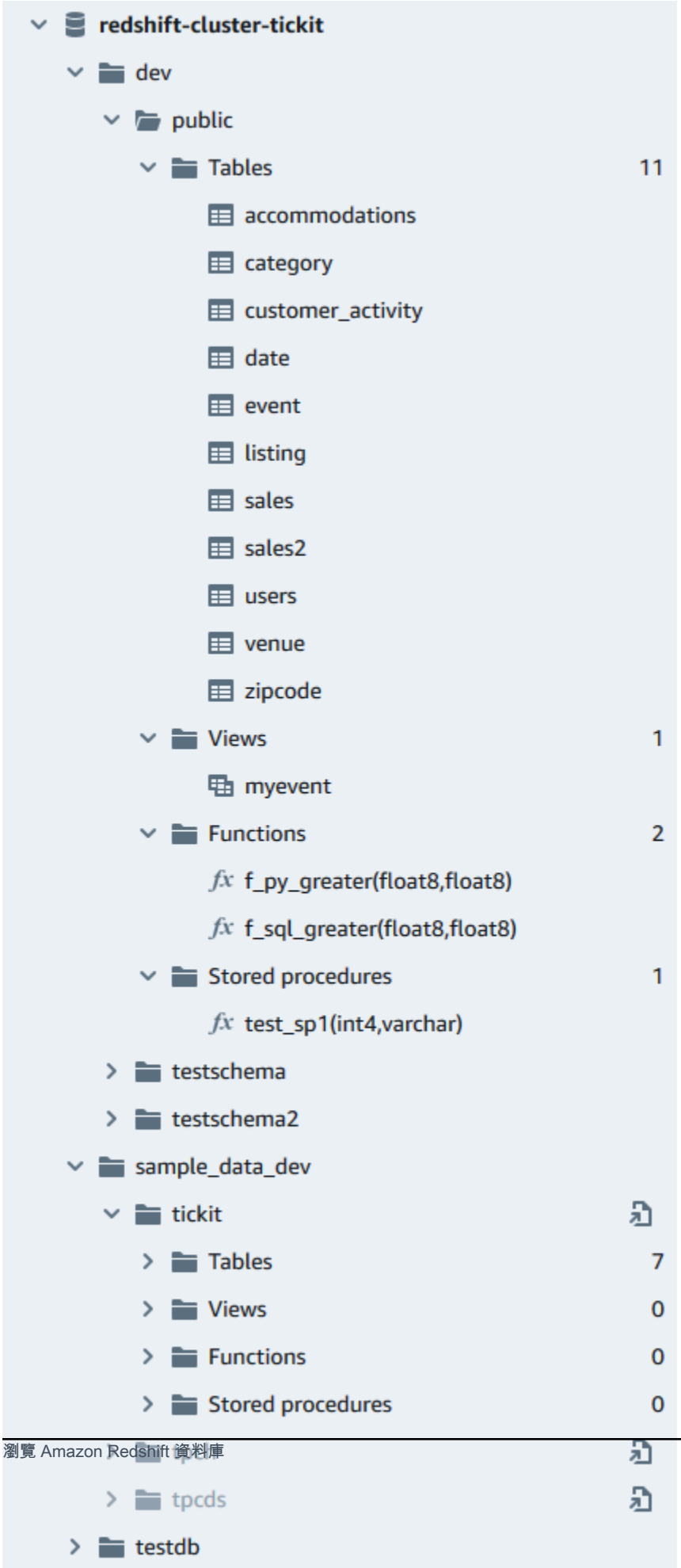
使用此方法而不是資料庫名稱時，請提供儲存在 Secrets Manager、包含資料庫和登入憑證的祕密。如需有關建立機密的資訊，請參閱 [建立資料庫連線憑證的機密](#)。

當您使用查詢編輯器 v2 選取叢集或工作群組時，視內容而定，您可以使用內容 (右鍵) 選單來建立、編輯和刪除連線。您可以透過選擇連線詳細資訊，檢視連線的連線 ARN 等屬性。您也可以編輯連接至連線的標籤。

瀏覽 Amazon Redshift 資料庫

在資料庫內，您可於樹狀檢視面板中管理結構描述、資料表、檢視、函數和預存程序。檢視中的每個物件在內容 (右鍵) 選單中都有與其相關聯的動作。

階層式樹狀檢視面板會顯示資料庫物件。若要重新整理樹狀檢視面板，以顯示上次顯示樹狀檢視之後可能已建立的資料庫物件，請選擇  圖示。開啟物件的內容 (右鍵) 選單即可查看您能夠執行的動作。


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

在選擇資料表後，您可以執行下列操作：

- 若要使用會查詢資料表中所有資料欄的 SELECT 陳述式在編輯器中開始查詢，請使用選取資料表。
- 若要查看屬性或資料表，請使用顯示資料表定義。使用此選項可查看資料欄名稱、資料欄類型、編碼、分佈索引鍵、排序索引鍵，以及資料欄是否可以包含 Null 值。如需資料表屬性的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE TABLE。
- 若要刪除資料表，請使用刪除。您可以使用截斷資料表刪除資料表中的所有資料列，或使用捨棄資料表從資料庫中移除資料表。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 TRUNCATE 和 DROP TABLE。

選擇要重新整理的結構描述或捨棄結構描述。

選擇要顯示檢視定義的檢視或捨棄檢視。

選擇要顯示函數定義的函數或捨棄函數。

選擇要顯示程序定義的預存程序或捨棄程序。

建立資料庫物件

您可以建立資料庫物件，包括資料庫、結構描述、資料表和使用者定義函數 (UDF)。您必須連線至叢集或工作群組和資料庫，才能建立資料庫物件。

建立資料庫

您可以使用查詢編輯器 v2 在叢集或工作群組中建立資料庫。

若要建立資料庫

如需有關資料庫的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE DATABASE。

1. 選擇 建立，然後選擇資料庫。
2. 輸入資料庫名稱。
3. (選擇性) 選取使用者和群組，然後選擇資料庫使用者。

4. (選擇性) 您可以從資料共用或 AWS Glue Data Catalog建立資料庫。如需 的詳細資訊 AWS Glue，請參閱什麼是 AWS Glue？ 《 AWS Glue 開發人員指南》中的 。
   - (選擇性) 選取使用資料共用建立，然後選擇選取資料共用。此清單包含可用來在目前的叢集或工作群組中建立取用者資料共用的生產者資料共用。
   - （選用） 選取使用 建立 AWS Glue Data Catalog，然後選擇選擇 AWS Glue 資料庫。在資料目錄結構描述中，輸入在參考由三個部分組成的名稱 (database.schema.table) 中的資料時，會用於結構描述的名稱。
5. 選擇建立資料庫。

   樹狀檢視面板中隨即會顯示新的資料庫。

   當您選擇選擇性步驟來查詢從資料共用建立的資料庫時，請連線到叢集或工作群組中的 Amazon Redshift 資料庫 (例如，預設資料庫 dev)，並使用由三個部分組成的標記法 (datase.schema.table) (其會參考您選取使用資料共用建立時所建立的資料庫名稱)。資料共用資料庫會列在查詢編輯器 v2 的編輯器標籤中，但未啟用直接連線功能。

   當您選擇選用步驟來查詢從 建立的資料庫 AWS Glue Data Catalog時，請連線至叢集或工作群組中的 Amazon Redshift 資料庫 （ 例如，預設資料庫 dev)，並使用三部分表示法 (database.schema.table)，以參考您選取使用 建立 AWS Glue Data Catalog時建立的資料庫名稱、您在資料目錄結構描述中命名的結構描述，以及 中的資料表 AWS Glue Data Catalog。類似於：

   ```
   SELECT * FROM glue-database.glue-schema.glue-table
   ```

   > **Note**
   >
   > 確認您已使用 連線方法連線至預設資料庫 使用 IAM 身分的暫時登入資料，且您的 IAM 登入資料已獲得 AWS Glue 資料庫的使用權限。
   >
   > ```
   > GRANT USAGE ON DATABASE glue-database to "IAM:MyIAMUser"
   > ```

   AWS Glue 資料庫會列在查詢編輯器 v2 編輯器索引標籤中，但不會啟用直接連線。

   如需查詢 的詳細資訊 AWS Glue Data Catalog，請參閱《Amazon Redshift 資料庫開發人員指南》中的以消費者身分使用 Lake Formation 受管資料共用和以生產者身分使用 Lake Formation 受管資料共用。

以資料共用取用者身分建立資料庫的範例

下列範例描述的是所使用的一個特定案例，目的是使用查詢編輯器 v2 從資料共用建立資料庫。請檢閱此案例，以了解如何在環境中從資料共用建立資料庫。此案例使用兩個叢集，分別是 cluster-base (生產者叢集) 和 cluster-view (取用者叢集)。

1. 使用 Amazon Redshift 主控台在叢集 cluster-base 中為資料表 category2 建立資料共用。生產者資料共用會命名為 datashare_base。

   如需有關建立資料共用的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的在 Amazon Redshift 中跨叢集共用資料。

2. 使用 Amazon Redshift 主控台接受資料共用 datashare_base 作為叢集 cluster-view 中資料表 category2 的取用者。
3. 檢視查詢編輯器 v2 中的樹狀檢視面板，該面板會將 cluster-base 的階層顯示為：

   - 叢集：cluster-base
     - 資料庫：dev
       - 結構描述：public
         - 資料表：category2
4. 選擇  建立，然後選擇資料庫。
5. 在資料庫名稱中輸入 see_datashare_base。
6. 選取使用資料共用建立，然後選擇選取資料共用。選擇 datashare_base 作為您要建立之資料庫的來源。

   查詢編輯器 v2 中的樹狀檢視面板會將 cluster-view 的階層顯示為：

   - 叢集：cluster-view
     - 資料庫：see_datashare_base
       - 結構描述：public
         - 資料表：category2
7. 當您查詢資料時，請連線到叢集 cluster-view 的預設資料庫 (名稱一般是 dev)，但在 SQL 中請參考資料共用資料庫 see_datashare_base。

> **Note**
>
> 在查詢編輯器 v2 的編輯器檢視中，所選取的叢集為 cluster-view。所選取的資料庫為 dev。資料庫 see_datashare_base 會列出，但未啟用直接連線功能。您選擇 dev 資料庫，並在執行的 SQL 中參考 see_datashare_base。

```
SELECT * FROM "see_datashare_base"."public"."category2";
```

查詢會從叢集 cluster_base 中的資料共用 datashare_base 擷取資料。

從 建立資料庫的範例 AWS Glue Data Catalog

下列範例說明使用 AWS Glue Data Catalog 查詢編輯器 v2 從 建立資料庫的特定案例。檢閱此案例，了解如何從 AWS Glue Data Catalog 環境中的 建立資料庫。這個案例會使用 cluster-view 這個叢集來包含您建立的資料庫。

1. 選擇 建立，然後選擇資料庫。
2. 在資料庫名稱中輸入 data_catalog_database。
3. 選取使用 建立 AWS Glue Data Catalog，然後選擇選擇 AWS Glue 資料庫。選擇 glue_db 作為您要建立之資料庫的來源。

   選擇資料目錄結構描述，並輸入 myschema 作為要在由三個部分組成的標記法中使用的結構描述名稱。

   查詢編輯器 v2 中的樹狀檢視面板會將 cluster-view 的階層顯示為：

   - 叢集：cluster-view
     - 資料庫：data_catalog_database
       - 結構描述：myschema
         - 資料表：category3
4. 當您查詢資料時，請連線到叢集 cluster-view 的預設資料庫 (名稱一般是 dev)，但在 SQL 中請參考資料庫 data_catalog_database。

> **Note**
>
> 在查詢編輯器 v2 的編輯器檢視中，所選取的叢集為 `cluster-view`。所選取的資料庫為 `dev`。資料庫 `data_catalog_database` 會列出，但未啟用直接連線功能。您選擇 `dev` 資料庫，並在執行的 SQL 中參考 `data_catalog_database`。

```
SELECT * FROM "data_catalog_database"."myschema"."category3";
```

查詢會擷取由 AWS Glue Data Catalog分類的資料。

建立結構描述

您可以使用查詢編輯器 v2 在叢集或工作群組中建立結構描述。

建立結構描述

如需有關結構描述的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的結構描述。

1. 選擇 建立，然後選擇結構描述。
2. 輸入結構描述名稱。
3. 選擇本機或外部作為結構描述類型。

   如需本機結構描述的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE SCHEMA。如需外部結構描述的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 CREATE EXTERNAL SCHEMA。
4. 如果您選擇外部，則會有下列外部結構描述選項。

   - Glue Data Catalog — 可在 Amazon Redshift 中建立會參考 AWS Glue中資料表的外部結構描述。除了選擇 AWS Glue 資料庫之外，也請選擇與叢集相關聯的 IAM 角色，以及與 Data Catalog 相關聯的 IAM 角色。
   - PostgreSQL — 可在 Amazon Redshift 中建立會參考 Amazon RDS for PostgreSQL 或 Amazon Aurora PostgreSQL 相容版本資料庫的外部結構描述。也請提供連線資訊給資料庫。如需聯合

查詢的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的使用聯合查詢來查詢資料。

- MySQL — 可在 Amazon Redshift 中建立會參考 Amazon RDS for MySQL 或 Amazon Aurora MySQL 相容版本資料庫的外部結構描述。也請提供連線資訊給資料庫。如需聯合查詢的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的使用聯合查詢來查詢資料。

5. 選擇建立結構描述。

   樹狀檢視面板中便會出現新的結構描述。

建立資料表

您可以使用查詢編輯器 v2 在叢集或工作群組中建立資料表。

若要建立資料表

您可以根據指定的逗號分隔值 (CSV) 檔案來建立資料表，也可以定義資料表的每個資料欄。如需有關資料表的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的設計資料表和 CREATE TABLE。

請選擇在編輯器中開啟查詢，先檢視及編輯 CREATE TABLE 陳述式，再執行查詢以建立資料表。

1. 選

   
   擇 建立，然後選擇資料表。
2. 選擇結構描述。
3. 輸入資料表名稱。
4. 選擇

   
   新增欄位以新增資料欄。
5. 使用 CSV 檔案作為資料表定義的範本：

   a. 選擇從 CSV 載入。

   b. 瀏覽至檔案位置。

      如果您使用 CSV 檔案，請確定檔案的第一個資料列包含資料欄標題。

   c. 選擇檔案，然後選擇開啟。確認資料欄名稱和資料類型符合所需。

6. 針對每一個資料欄，請選擇該資料欄，然後選擇您想要的選項：
   - 為編碼選擇一個值。
   - 選擇預設值。
   - 如果您希望資料欄的值遞增，請開啟自動遞增。然後指定自動遞增種子和自動遞增步長的值。
   - 如果資料欄一律應該包含值，請開啟非 NULL。
   - 輸入資料欄的大小值。
   - 如果您想讓資料欄成為主索引鍵，請開啟主索引鍵。
   - 如果您想讓資料欄成為唯一索引鍵，請開啟唯一索引鍵。
7. (選擇性) 選擇資料表詳細資訊，然後選擇下列任何選項：
   - 分佈索引鍵資料欄和樣式。
   - 排序索引鍵資料欄和排序類型。
   - 開啟備份，以將資料表包含在快照中。
   - 開啟暫存資料表以將資料表建立為暫存資料表。
8. 選擇在編輯器中開啟查詢以繼續指定用來定義資料表的選項，或選擇建立資料表以建立資料表。

建立函數

您可以使用查詢編輯器 v2 在叢集或工作群組中建立函數。

建立 函數

1. 選擇 建立，然後選擇函數。
2. 在類型中，選擇 SQL 或 Python。
3. 選擇結構描述的值。
4. 輸入函數名稱的值。
5. 輸入函數波動的值。
6. 以輸入參數的順序，依資料類型選擇參數。
7. 在傳回中，選擇資料類型。
8. 輸入函數的 SQL 程式或 Python 程式的程式碼。

9. 選擇建立。

如需使用者定義函數 (UDF) 的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的建立使用者定義函數。

檢視查詢和標籤歷史記錄

您可以使用查詢編輯器 v2 檢視查詢歷史記錄。只有使用查詢編輯器 v2 執行的查詢才會出現在查詢歷史記錄中。使用編輯器標籤或筆記本標籤所執行的查詢都會顯示出來。您可以依期間 (例如 `This week`) 篩選所顯示的清單，其中一週會定義為星期一至星期日。查詢清單一次會擷取 25 列符合篩選條件的查詢。選擇載入更多便可查看下一組。選擇查詢，然後從動作選單中選擇。可用的動作取決於所選擇的查詢是否已儲存。您可以選擇下列操作：

- 檢視查詢詳細資訊 — 顯示查詢詳細資訊頁面，其中包含已執行之查詢的詳細資訊。
- 在新標籤中開啟查詢 — 開啟新的編輯器標籤，並使用所選擇的查詢讓其做好準備。如果仍然連線，則系統會自動選取叢集或工作群組和資料庫。若要執行查詢，請先確認是否已選擇正確的叢集或工作群組和資料庫。
- 開啟來源標籤 — 如果仍開啟，請導覽至查詢在執行時所屬的編輯器或筆記本標籤。查詢執行後，編輯器或筆記本的內容可能已有所變更。
- 開啟儲存的查詢 — 導覽至編輯器或筆記本標籤，並開啟查詢。

您也可以檢視在編輯器標籤中執行之查詢的歷史記錄，或檢視在筆記本標籤中執行之查詢的歷史記錄。若要查看標籤中的查詢歷史記錄，請選擇標籤歷史記錄。在標籤歷史記錄內，您可以執行下列操作：

- 複製查詢 — 將查詢版本的 SQL 內容複製到剪貼簿。
- 在新標籤中開啟查詢 — 開啟新的編輯器標籤，並使用所選擇的查詢讓其做好準備。若要執行查詢，您必須選擇叢集或工作群組和資料庫。
- 檢視查詢詳細資訊 — 顯示查詢詳細資訊頁面，其中包含已執行之查詢的詳細資訊。

與 Amazon Q 生成式 SQL 互動

> **Note**
>
> Amazon Q 生成式 SQL 支援僅適用於下列項目 AWS 區域：
>
> - 美國東部 (維吉尼亞北部) 區域 (us-east-1)

- 美國東部 (俄亥俄) 區域 (us-east-2)
- 美國西部 (奧勒岡) 區域 (us-west-2)
- 亞太區域 (孟買) 區域 (ap-south-1)
- 亞太區域 (首爾) 區域 (ap-northeast-2)
- 亞太區域 (新加坡) 區域 (ap-southeast-1)
- 亞太區域 (雪梨) 區域 (ap-southeast-2)
- 亞太區域 (東京) 區域 (ap-northeast-1)
- 加拿大 (中部) 區域 (ca-central-1)
- 歐洲 (法蘭克福) 區域 (eu-central-1)
- 歐洲 (愛爾蘭) 區域 (eu-west-1)
- 歐洲 (倫敦) 區域 (eu-west-2)
- 歐洲 (巴黎) 區域 (eu-west-3)
- 南美洲 (聖保羅) 區域 (sa-east-1)

如需有關資料在哪裡處理的資訊，請參閱《Amazon Q Developer 使用者指南》中的 Amazon Q Developer 中的跨區域推論。

您可以在 Amazon Redshift 查詢編輯器 v2 中與 Amazon Q 生成式 SQL 功能互動。這是一個編碼助理，可根據您的提示和資料庫模式產生 SQL 陳述式。當您在查詢編輯器 v2 中編寫筆記本時，可以使用此編碼助理。產生的 SQL 適用於筆記本所連線的資料庫。

與 Amazon Q 生成式 SQL 互動時，請提出特定問題，在有複雜的請求時進行迭代，並驗證答案的準確性。

以自然語言提供分析請求時，請盡可能具體地協助編碼助理了解您需要的內容。而不是詢問「尋找售出最多門票的熱門會場」，而是提供更多詳細資訊，例如「尋找 2008 年售出最多門票的前三個會場的名稱/ID」。在資料庫中使用一致且特定的物件名稱 (如果您知道的話)。例如資料庫中定義的結構描述、資料表和欄名稱，而不要以不同的方式參考相同的物件，這樣可能造成助理混淆。

將複雜的請求分解為多個簡單的陳述式，這些陳述式更容易讓助理解譯。反覆提出後續問題，以從助理那裡獲得更詳細的分析。例如，首先要問「哪個州的場地最多？」 然後根據回應，詢問「哪個是這個州最受歡迎的會場？」。

執行產生的 SQL 之前，請先檢閱產生的 SQL 以確保準確性 如果產生的 SQL 查詢有錯誤或不符合您的意圖，請提供指示給助理，說明如何更正，而不是改寫整個請求。例如，如果查詢缺少年份的述詞子句，請詢問「從 2008 年提供會場」。

將您執行生成式 SQL 所收到的錯誤文字作為提示提交至 Amazon Q 生成式 SQL。它會從這些錯誤中學習，以產生更佳的 SQL。

將您的結構描述新增至 SQL 搜尋路徑，以表示應使用結構描述。例如，當資料位於 tickit 結構描述而非公有結構描述中時，新增 tickit 結構描述。

```
set search_path to '$user', tickit;
```

與 Amazon Q 生成式 SQL 互動時的考量事項

在聊天面板中工作時，請考慮以下幾點。

- 您帳戶的查詢編輯器 v2 管理員必須已在生成式 SQL 設定頁面中開啟聊天功能。
- 若要使用 Amazon Q 生成式 SQL，除了查詢編輯器 v2 的 AWS 受管政策中指定的其他許可之外，還需要 IAM 政策`sqlworkbench:GetQSqlRecommendations`中的許可。如需 AWS 受管政策的詳細資訊，請參閱 存取查詢編輯器 v2。
- 您的問題必須以英文撰寫。
- 您的問題必須參考叢集或工作群組中已連接的資料庫。為了避免空白狀態錯誤，應該有至少一個資料表和資料庫中的一些資料。
- 您的問題必須參照儲存在連線資料庫中的資料。但無法參考外部結構描述。如需受支援之結構描述的詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的建立結構描述。
- 任何導致 SQL 變更連線資料庫的問題,都可能會產生警告。
- 生成式 AI 技術是一種新穎的技術，在反應中可能會出現錯誤，有時也稱為幻覺。在您的環境或工作負載中使用之前，請先測試並檢閱所有程式碼是否有錯誤和漏洞。
- 您可以透過帳戶中其他使用者執行來分享 SQL 查詢，以改善建議。您的帳戶管理員可以執行下列 SQL 命令，以允許存取帳戶的查詢歷程記錄。

  ```
  GRANT ROLE SYS:MONITOR to "IAMR:role-name";
  GRANT ROLE SYS:MONITOR to "IAM:user-name";
  GRANT ROLE SYS:MONITOR to "database-username";
  ```

  如需有關 SYS:MONITOR 的詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的 Amazon Redshift 系統定義角色。

- 您的資料是安全和私有的。您的資料不會跨帳戶共用。您的查詢、資料和資料庫結構描述不會用於訓練生成式 AI 基礎模型 (FM)。您的輸入會用作 FM 的上下文提示詞，僅做為對您的查詢的回答。

使用生成式 SQL

設定正確的許可權後，在查詢編輯器 v2 中使用筆記本時，您可以選擇圖示來開始對話。

與 Amazon Q 生成式 SQL 聊天功能進行互動以產生 SQL

1. 在查詢編輯器 v2 的編輯器索引標籤中，開啟筆記本。
2. 選擇

   生成式 SQL 圖示，然後按照指令在聊天面板中詢問您對 Amazon Redshift 查詢編輯器 v2 生成式 SQL 的問題。

   您在提示欄位中提供問題，Amazon Q 生成式 SQL 就會使用建議的 SQL 來回應。遇到的任何錯誤，都會在聊天面板中傳回給您。
3. 選擇新增至筆記本，新增 Markdown 儲存格與您的提示，以及含有建議 SQL 的 SQL 儲存格以加入至您的筆記本。
4. (選用) 藉由選擇

   實用意見回饋圖示或

   不實用意見回饋圖示，提供有關所產生 SQL 的意見回饋。您可以將不實用的意見回饋分類為 `Incorrect tables/columns`、`Incorrect predicates/literals/group bys`、`Incorrect SQL structure` 或 `Other`。此外，您可以隨意見回饋提供一些自由格式的文字來敘述 SQL 的準確性。
5. (選用) 選擇重新產生 SQL，為相同的提示產生另一個回應。您可以選擇為目前的提示重新產生 SQL 一次。
6. (選用) 在生成式 SQL 聊天面板中，選擇

   更多圖示，然後選擇重新整理資料庫以重新整理描述已連線之資料庫的中繼資料。此中繼資料包括資料庫中結構描述定義、資料表和欄的定義。

以管理員身分更新生成式 SQL 設定

具有正確 IAM 許可的使用者可以檢視和變更相同 AWS 帳戶中其他使用者的生成式 SQL 設定。除了查詢編輯器 v2 的 AWS 受管政策中指定的其他許可之外，此管理員還必須在其 IAM 政策`sqlworkbench:UpdateAccountQSqlSettings`中擁有許可。如需受管政策的詳細資訊，請參閱要使用查詢編輯器 v2 所需的許可。

讓系統管理員為帳戶中的所有使用者開啟生成式 SQL 聊天

1. 選擇

   設定圖示以顯示不同設定畫面的選單。
2. 然後選擇

   生成式 SQL 設定圖示，以顯示 Q 生成式 SQL 設定頁面。
3. 選取 Q 生成式 SQL，為帳戶中的使用者開啟生成式 SQL 功能。

   開啟 Amazon Q 生成式 SQL 之後，您可以檢視配置中剩餘的提示數目。查詢編輯器 v2 管理員可以讓帳戶中的使用者使用 Amazon Q Developer 專業方案。若要使用專業方案，請使用 IAM Identity Center 設定您的使用者，並為每個使用者訂閱 Amazon Q Developer 專業方案。如需設定 IAM Identity Center 搭配 Amazon Redshift 的相關資訊，請參閱 將 Redshift 與 AWS IAM Identity Center 連線，以獲得單一登入體驗。如需有關 Amazon Q Developer 定價的資訊，請參閱 Amazon Q Developer 定價。

   使用 Amazon Q Developer Free 方案時， 所有使用者的提示總數每月限制 AWS 帳戶 為 1，000 個。使用 Amazon Q Developer 專業方案時，任何個別使用者每月可提交的提示總數限制為 1,000 個。您可以在設定頁面上檢視可用的提示數目。如需有關 Amazon Q Developer 定價的資訊，請參閱 Amazon Q Developer 定價。

自訂內容

查詢編輯器 v2 管理員可以指定自訂內容，以根據您的環境量身打造生成式 SQL。自訂內容提供領域知識和偏好設定，可精細控制 SQL 產生。自訂內容是在 JSON 檔案中定義，查詢編輯器 v2 管理員可將其上傳到 Amazon Q 生成式 SQL。

以下是用來將資料倉儲的生成式 SQL 個人化的 JSON 金鑰。

所有資料表參考都需要遵循三部分標記法 `database.schema.table`。

Resources

資源會指定套用自訂內容的資料資產範圍或部分。

ResourceId

指定資源的唯一識別碼。對於 Amazon Redshift 叢集，請指定 `cluster id`。對於 Redshift Serverless 工作群組，請指定 `workgroup name`。

ResourceType

有效值：`REDSHIFT_WAREHOUSE`。

TablesToInclude

指定 SQL 生成時參考的一組資料表。當您想要將 SQL 查詢的範圍限於一組定義的可用資料表子集時，此欄位會非常重要。此欄位可透過減少不必要的資料表參考，協助最佳化生成過程。您可以將此欄位與 `TablesToExclude` 配對，以更精確地控制查詢生成。

TablesToExclude

指定要從 SQL 生成中排除的一組資料表。當某些資料表不相關或不應在查詢生成過程中考慮時，可使用此項。

TableAnnotations

提供有關使用中資料表的中繼資料或補充資訊。這些註釋可以包括資料表說明、使用須知或任何其他屬性，以協助 Amazon Q 生成式 SQL 進一步了解資料表的內容或結構。這可讓資料表定義更清楚明確，對於提高 SQL 生成的準確性來說非常寶貴。

ColumnsToInclude

定義生成 SQL 查詢時要包含指定資料表中的哪些欄。此欄位有助於 Amazon Q 生成式 SQL 聚焦相關欄，並透過縮小資料擷取範圍來改善效能。此欄位可確保 Amazon Q 生成式 SQL 僅提取指定查詢內容所需的資料。

ColumnsToExclude

指定 SQL 生成時要省略不予考慮的欄。此項可在某些欄包含不相關或 Amazon Q 生成式 SQL 不應考慮的多餘資料時使用。透過管理包含和排除欄，您就可以精簡結果，並維持對所擷取資料的控制。

ColumnAnnotations

與 TableAnnotations 類似，此欄位提供個別欄特定的中繼資料或註釋。這些註釋可提供深入了解欄定義或特殊處理指示的資訊。此資訊有助於引導 SQL 生成過程，並確保在查詢中適當地使用欄。

CuratedQueries

一組預先定義的問題和答案範例，其中問題是以自然語言 (NLQ) 撰寫，而答案是對應的 SQL 查詢。這些範例可協助 Amazon Q 生成式 SQL 了解預期生成的查詢類型。這些範例可作為參考點來提高 Amazon Q 生成式 SQL 輸出的準確性和關聯性。

CustomDocuments

提供給 Amazon Q 生成式 SQL 的額外資訊或提示，例如定義、領域特定知識或解釋。例如，如果您的業務單位使用獨特的方法來計算值，例如「製造部門的總銷售額是價格 * 收入」，則可記載於此處。這些文件透過提供額外的內容，增強了 Amazon Q 生成式 SQL 解譯自然語言輸入的能力。

AdditionalTables

指定 SQL 生成時應考慮的任何其他資料表，但這些資料表不屬於儲存在資料倉儲中的資料。這可讓 Amazon Q 生成式 SQL 將外部資料來源整合到其 SQL 生成邏輯中，以擴大其處理複雜資料環境的容量。

AppendToPrompt

提供給 Amazon Q 生成式 SQL 的額外指示或指南，以引導 SQL 生成程序。這可包括說明如何建構查詢的特定指令、特定 SQL 建構模組的偏好設定，或任何其他提高 Amazon Q 生成式 SQL 輸出品質的高階指示。

下列範例自訂內容會顯示 JSON 檔案的格式，並定義下列項目：

- 定義叢集 mycluster 的 Amazon Redshift 資料倉儲自訂內容。
- 定義要包含和排除的特定資料表和欄，以協助最佳化 SQL 生成程序。
- 定義要包含之資料表和欄的註釋。
- 定義 Amazon Q 生成式 SQL 要使用的範例策劃查詢。
- 定義生成 SQL 時要使用的自訂文件和防護機制。
- 定義生成 SQL 時要使用的額外資料表的 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."
        }
    ]
}
```

教學課程：搭配 TICKIT 資料使用 Amazon Q 生成式 SQL 功能

若要撰寫生成式 SQL 的有效提示詞，您必須了解您的資料庫結構描述和資料。TICKIT 資料包含七個資料表：兩個事實資料表和五個維度資料表。範例資料包含 2008 年發生之娛樂活動服務員銷售的相關記錄。如需 TICKIT 資料結構描述的詳細資訊，請參閱 Amazon Redshift 資料庫開發人員指南中的範例資料庫。您可以透過 Amazon Redshift 主控台和查詢編輯器 v2 中的各種方法，將 TICKIT 資料載入資料庫。查詢編輯器 v2 提供一種將 TICKIT 資料載入資料庫 `sample_data_dev` 的方法。如需相關資訊，請參閱將資料載入到資料庫。查詢編輯器 v2 也提供 TICKIT 資料的範例提示。下列案例說明與生成式 SQL 的對話，以產生有關 TICKIT 範例資料的 SQL。在這個案例中，TICKIT 範例資料已經建立於 Amazon Redshift 叢集的 `dev` 資料庫中。

> **Note**
>
> 這個範例是有關對話的圖解。生成式 SQL 的回應可能不符合使用相同提示詞的結果。

與 Amazon Q 生成式 SQL 的對話範例

1. 在編輯器中，連接至包含 TICKIT 範例資料的叢集或工作群組。
2. 建立空白筆記本，然後選擇

   生成式 SQL 圖示以開啟聊天面板。
3. 輸入以下提示以產生 SQL，以驗證 VENUE 資料表中的記錄數：

   ```
   How many venues are there?
   ```

   ```
   SELECT
     COUNT(*) AS num_venues
   FROM
     tickit.venue
   ```

   選擇新增至筆記本，新增兩個儲存格至開啟的筆記本。一個 Markdown 儲存格「有多少個會場？」 和一個包含生成式 SQL。

   在 SQL 儲存格中，選擇執行以接收結果：

   ```
   count
   ------
   202
   ```
4. 若要請求其他版本的 SQL，請選擇重新產生 SQL 並接收下列答案：

   ```
   SELECT
     venuestate,
     COUNT(*) AS num_venues
   FROM
     tickit.venue
   GROUP BY
     venuestate
   ```

   選擇新增至筆記本，多新增兩個儲存格至開啟的筆記本。一個 Markdown 儲存格「有多少個會場？」 和一個包含生成式 SQL。

   在 SQL 儲存格中，選擇執行以按編號接收列出會場狀態的結果：

```
venuestate num_venues
----------------------
MA         4
OH         8
MI         5
...
```

5. 生成式 SQL 助理可以假設您請求的內容。儘管資料庫中沒有小工具，但您仍然可以詢問。在這種情況下，它會提供你一條訊息，表示無法產生 SQL。

```
How many widgets are there?
```

```
I was not able to generate any SQL
```

6. 當您了解更多有關資料庫的資訊時，可以提出更具體的問題來產生 SQL。例如，您知道資料庫包含 2008 年中幾個月份的銷售資料。輸入下列提示詞以產生 SQL，尋找 2008 年 11 月的銷售數量：

```
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
```

選擇新增至筆記本，新增兩個儲存格至開啟的筆記本。一個 Markdown 儲存格「2008 年 11 月有多少銷量量？」 和一個包含生成式 SQL。

在 SQL 儲存格中，選擇執行以接收結果：

```
num_sales
-----
```

```
0
```

您了解到這不是你期望的結果。

7. 您注意到述詞 date.month='November' 預期 DATE 資料表中的月份會以月份名稱縮寫來表示。您可以將述詞變更為 d.month='NOV' 並重新執行 SQL。

```
SELECT
  COUNT(*)
FROM
  sales
  JOIN date ON sales.dateid = date.dateid
WHERE
  date.month = 'NOV'
  AND date.year = 2008
```

在 SQL 儲存格中，選擇執行以取得新結果。

```
count
-----
14261
```

8. 如果您問的問題會嘗試變更連線的資料庫，則會傳回警告訊息以及任何建議的 SQL。輸入下列提示詞以產生 SQL，將資料插入資料表：

```
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.
```

如果您選擇新增至筆記本，將兩個儲存格新增至開啟的筆記本，並執行 SQL，然後 SQL 會失敗。

```
ERROR: syntax error at or near "," Position: 132 [ErrorId:
 1-6546764a-011df2691778846219ce6ec2]
```

這個案例只說明了與 Amazon Q 生成式 SQL 互動的一些基本方法。您可以使用這種生成式 AI 技術進行更多實驗，幫助您開始編寫 SQL 來查詢資料庫。

將資料載入到資料庫

您可以使用查詢編輯器 v2 將資料載入到 Amazon Redshift 叢集或工作群組中的資料庫。本節說明如何載入範例資料、從 S3 載入資料，以及從本機檔案設定和工作流程載入資料。

範例資料

查詢編輯器 v2 隨附範例資料和筆記本，可供您載入到範例資料庫和對應的結構描述。

若要載入範例資料，請選擇與您要載入的範例資料相關聯的



圖示。然後，查詢編輯器 v2 便會將資料載入資料庫 `sample_data_dev` 的結構描述中，並建立已儲存筆記本的資料夾。

以下是可用的範例資料集。

tickit

Amazon Redshift 文件中的多數範例都使用名為 `tickit` 的範本資料。這份資料包含七個資料表：兩個事實資料表和五個維度資料表。在載入這份資料時，系統便會以範例資料更新結構描述 `tickit`。如需 `tickit` 資料的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的範例資料庫。

tpch

這份資料會用於決策支援基準。在載入這份資料時，系統便會以範例資料更新結構描述 `tpch`。如需 `tpch` 資料的相關資訊，請參閱 TPC-H。

tpcds

這份資料會用於決策支援基準。在載入這份資料時，系統便會以範例資料更新結構描述 `tpcds`。如需 `tpcds` 資料的相關資訊，請參閱 TPC-DS。

從 Amazon S3 載入資料

您可以將 Amazon S3 資料載入到現有資料表或新的資料表。

將資料載入到現有資料表

查詢編輯器 v2 會使用 COPY 命令從 Amazon S3 載入資料。在查詢編輯器 v2 載入資料精靈中產生和使用的 COPY 命令，支援許多可供 COPY 命令語法從 Amazon S3 進行複製的參數。如需 COPY 命令及其用於從 Amazon S3 載入資料之選項的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的從 Amazon Simple Storage Service 進行 COPY。

1. 確認您要將資料載入到其中的資料庫內已建立資料表。
2. 確認您已連線至查詢編輯器 v2 的樹狀檢視面板中的目標資料庫後再繼續。您可以使用內容功能表 (按一下滑鼠右鍵) 對要在其中載入資料的叢集或工作群組建立連線。

   選擇

   載入資料。
3. 對於資料來源，選擇從 S3 儲存貯體載入。
4. 在 S3 URI 中選擇瀏覽 S3，以尋找要載入的資料所在的 Amazon S3 儲存貯體。
5. 如果指定的 Amazon S3 儲存貯體與目標資料表不 AWS 區域 相同，請為 AWS 區域 資料所在的選擇 S3 檔案位置。
6. 如果 Amazon S3 檔案實際上是包含多個 Amazon S3 儲存貯體 URI 的資訊清單，請選擇此檔案是資訊清單檔案。
7. 為要上傳的檔案選擇檔案格式。支援的資料格式有 CSV、JSON、DELIMITER、FIXEDWIDTH、SHAPEFILE、AVRO、PARQUET 和 ORC。根據指定的檔案格式，您可以選擇相應的檔案選項。如果資料有加密，您也可以選取資料已加密，然後輸入用來加密資料之 KMS 金鑰的 Amazon Resource Name (ARN)。

   如果您選擇 CSV 或 DELIMITER」，則也可以選擇分隔符號字元，以及是否要忽略標頭資料列 (如果指定的資料列數目實際上是資料欄名稱而非要載入的資料)。
8. 選擇壓縮方法來壓縮檔案。預設為沒有壓縮。
9. (選擇性) 進階設定支援各種資料轉換參數和載入操作。視需要為您的檔案輸入此資訊。

   如需資料轉換和資料載入參數的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的資料轉換參數和資料載入操作。
10. 選擇下一步。

11. 選擇載入現有資料表。
12. 確認或選擇在其中載入資料的目標資料表的位置，包括叢集或工作群組、資料庫、結構描述以及資料表名稱。
13. 選擇具有從 Amazon S3 載入資料所需許可的 IAM 角色。
14. (選擇性) 選擇資料欄名稱，在資料欄映射中輸入以依照輸入資料檔案的順序映射資料欄。
15. 選擇載入資料以開始載入資料。

    當載入完成時，查詢編輯器會顯示所產生來載入資料的 COPY 命令。系統會顯示 COPY 的結果。如果成功，您便可立即使用 SQL 從所載入的資料表中選取資料。如果發生錯誤，請查詢系統檢視 STL_LOAD_ERRORS 以取得更多詳細資訊。如需有關 COPY 命令錯誤的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 STL_LOAD_ERRORS。

當您將資料載入到新的資料表時，查詢編輯器 v2 會先在資料庫中建立資料表，再於相同工作流程中以個別動作的形式載入資料。

將資料載入到新的資料表

查詢編輯器 v2 會使用 COPY 命令從 Amazon S3 載入資料。在查詢編輯器 v2 載入資料精靈中產生和使用的 COPY 命令，支援許多可供 COPY 命令語法從 Amazon S3 進行複製的參數。如需 COPY 命令及其用於從 Amazon S3 載入資料之選項的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的從 Amazon Simple Storage Service 進行 COPY。

1. 確認您已連線至查詢編輯器 v2 的樹狀檢視面板中的目標資料庫後再繼續。您可以使用內容功能表 (按一下滑鼠右鍵) 對要在其中載入資料的叢集或工作群組建立連線。

   選擇

   載入資料。
2. 對於資料來源，選擇從 S3 儲存貯體載入。
3. 在 S3 URI 中選擇瀏覽 S3，以尋找要載入的資料所在的 Amazon S3 儲存貯體。
4. 如果指定的 Amazon S3 儲存貯體與目標資料表不 AWS 區域 相同，請為 AWS 區域 資料所在的選擇 S3 檔案位置。
5. 如果 Amazon S3 檔案實際上是包含多個 Amazon S3 儲存貯體 URI 的資訊清單，請選擇此檔案是資訊清單檔案。
6. 為要上傳的檔案選擇檔案格式。支援的資料格式有 CSV、JSON、DELIMITER、FIXEDWIDTH、SHAPEFILE、AVRO、PARQUET 和 ORC。根據

指定的檔案格式，您可以選擇相應的檔案選項。如果資料有加密，您也可以選取資料已加密，然後輸入用來加密資料之 KMS 金鑰的 Amazon Resource Name (ARN)。

如果您選擇 CSV 或 DELIMITER」，則也可以選擇分隔符號字元，以及是否要忽略標頭資料列 (如果指定的資料列數目實際上是資料欄名稱而非要載入的資料)。

7. 選擇壓縮方法來壓縮檔案。預設為沒有壓縮。
8. (選擇性) 進階設定支援各種資料轉換參數和載入操作。視需要為您的檔案輸入此資訊。

   如需資料轉換和資料載入參數的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的資料轉換參數和資料載入操作。
9. 選擇下一步。
10. 選擇載入新資料表。

    資料表資料欄會從輸入資料推斷而來。您可以透過新增資料欄和資料表詳細資訊，修改資料表結構描述的定義。若要還原至查詢編輯器 v2 推斷的資料表結構描述，請選擇還原為預設值。
11. 確認或選擇在其中載入資料的目標資料表的位置，包括叢集或工作群組、資料庫和結構描述。輸入要建立的資料表名稱。
12. 選擇具有從 Amazon S3 載入資料所需許可的 IAM 角色。
13. 選擇建立資料表，使用所顯示的定義建立資料表。

    此時會顯示資料表定義的檢閱摘要。資料表會建立到資料庫中。稍後若要刪除資料表，請執行 DROP TABLE SQL 命令。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 DROP TABLE。
14. 選擇載入資料以開始載入資料。

    當載入完成時，查詢編輯器會顯示所產生來載入資料的 COPY 命令。系統會顯示 COPY 的結果。如果成功，您便可立即使用 SQL 從所載入的資料表中選取資料。如果發生錯誤，請查詢系統檢視 STL_LOAD_ERRORS 以取得更多詳細資訊。如需有關 COPY 命令錯誤的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 STL_LOAD_ERRORS。

從本機檔案設定和工作流程載入資料

您可以將資料從本機檔案載入到現有資料表或新的資料表。

從本機檔案載入資料的管理員設定

您的查詢編輯器 v2 管理員必須在帳戶設定視窗中指定常用的 Amazon S3 儲存貯體。您必須已為帳戶使用者設定適當的許可。

- 必要的 IAM 許可 — 從本機檔案載入資料的使用者必須具有 `s3:ListBucket`、`s3:GetBucketLocation`、`s3:putObject`、`s3:getObject` 和 `s3:deleteObject` 許可。您可以指定 *`optional-prefix`*，以限制只有具有此字首的物件能透過查詢編輯器 v2 使用此儲存貯體。如果您透過查詢編輯器 v2 以外的工具使用這個相同的 Amazon S3 儲存貯體，便可以使用此選項。如需儲存貯體和字首的相關資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的管理使用者對特定資料夾的存取。為了確保系統不會允許跨使用者的資料存取，建議查詢編輯器 v2 管理員使用 Amazon S3 儲存貯體政策來根據 `aws:userid` 限制物件的存取。下列範例會向字首為 `aws:userid` 的 Amazon S3 物件允許具有唯讀/唯寫存取權限的 *`<staging-bucket-name>`* Amazon S3 許可。

  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}/*"
            ]
        }
```

```
    ]
}
```

- 資料分隔 — 建議不要讓使用者擁有存取彼此資料的權限 (即使很短暫也不行)。從本機檔案載入的方式會使用查詢編輯器 v2 管理員所設定的暫存 Amazon S3 儲存貯體。請設定暫存儲存貯體的儲存貯體政策，以在使用者之間提供資料分隔機制。下列範例顯示的儲存貯體政策會在 *<staging-bucket-name>* 的使用者之間分隔資料。

  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}/*"
            ]
         }
    ]
}
```

從本機檔案載入資料

將本機檔案的資料載入到現有資料表

查詢編輯器 v2 管理員必須在帳戶設定視窗中指定通用的 Amazon S3 儲存貯體。查詢編輯器 v2 會自動將本機檔案上傳到您帳戶使用的通用 Amazon S3 儲存貯體，然後使用 COPY 命令載入資料。查詢編輯器 v2 的載入本機檔案視窗所產生和執行的 COPY 命令，支援許多可供 COPY 命令語法從 Amazon S3 進行複製的參數。如需 COPY 命令及其用於從 Amazon S3 載入資料之選項的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的從 Amazon S3 進行 COPY。

1. 確認您要將資料載入到其中的資料庫內已建立資料表。
2. 確認您已連線至查詢編輯器 v2 的樹狀檢視面板中的目標資料庫。您可以使用內容功能表 (按一下滑鼠右鍵) 對要在其中載入資料的叢集或工作群組建立連線。

3. 選擇

   載入資料。

4. 對於資料來源，選擇從本機檔案載入。

5. 選擇瀏覽以尋找包含要載入檔案之資料的檔案。依預設，系統會顯示副檔名為 `.csv`、`.avro`、`.parquet` 和 `.orc` 的檔案，但您也可以選擇其他檔案類型。 檔案大小上限為 100 MB。

6. 為要上傳的檔案選擇檔案格式。支援的資料格式有 CSV、JSON、DELIMITER、FIXEDWIDTH、SHAPEFILE、AVRO、PARQUET 和 ORC。根據指定的檔案格式，您可以選擇相應的檔案選項。如果資料有加密，您也可以選取資料已加密，然後輸入用來加密資料之 KMS 金鑰的 Amazon Resource Name (ARN)。

   如果您選擇 CSV 或 DELIMITER」，則也可以選擇分隔符號字元，以及是否要忽略標頭資料列 (如果指定的資料列數目實際上是資料欄名稱而非要載入的資料)。

7. (選擇性) 進階設定支援各種資料轉換參數和載入操作。視需要為您的檔案輸入此資訊。

   如需資料轉換和資料載入參數的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的資料轉換參數和資料載入操作。

8. 選擇下一步。

9. 選擇載入現有資料表。

10. 確認或選擇在其中載入資料的目標資料表的位置，包括叢集或工作群組、資料庫、結構描述以及資料表名稱。

11. (選擇性) 您可以選擇資料欄名稱，在資料欄映射中輸入以依照輸入資料檔案的順序映射資料欄。

12. 選擇載入資料以開始載入資料。

    載入完成時，無論載入是否成功，系統都會顯示訊息。如果成功，您便可立即使用 SQL 從所載入的資料表中選取資料。如果發生錯誤，請查詢系統檢視 STL_LOAD_ERRORS 以取得更多詳細資訊。如需有關 COPY 命令錯誤的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 STL_LOAD_ERRORS。

    用來載入資料的 COPY 命令範本會出現在您的查詢歷史記錄中。此 COPY 命令範本會顯示一些已使用的參數，但其無法直接在編輯器索引標籤中執行。如需查詢歷史記錄的相關資訊，請參閱檢視查詢和標籤歷史記錄。

當您將資料載入到新的資料表時，查詢編輯器 v2 會先在資料庫中建立資料表，再於相同工作流程中以個別動作的形式載入資料。

將本機檔案資料載入到新的資料表

您的查詢編輯器 v2 管理員必須在帳戶設定視窗中指定常用的 Amazon S3 儲存貯體。本機檔案會自動上傳到您帳戶使用的通用 Amazon S3 儲存貯體，然後查詢編輯器 v2 會使用 COPY 命令載入資料。查詢編輯器 v2 的載入本機檔案視窗所產生和執行的 COPY 命令，支援許多可供 COPY 命令語法從 Amazon S3 進行複製的參數。如需 COPY 命令及其用於從 Amazon S3 載入資料之選項的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的從 Amazon S3 進行 COPY。

1. 確認您已連線至查詢編輯器 v2 的樹狀檢視面板中的目標資料庫。您可以使用內容功能表 (按一下滑鼠右鍵) 對要在其中載入資料的叢集或工作群組建立連線。
2. 選擇

   載入資料。
3. 對於資料來源，選擇從本機檔案載入。
4. 選擇瀏覽以尋找包含要載入檔案之資料的檔案。依預設，系統會顯示副檔名為 `.csv`、`.avro`、`.parquet` 和 `.orc` 的檔案，但您也可以選擇其他檔案類型。 檔案大小上限為 100 MB。
5. 為要上傳的檔案選擇檔案格式。支援的資料格式有 CSV、JSON、DELIMITER、FIXEDWIDTH、SHAPEFILE、AVRO、PARQUET 和 ORC。根據指定的檔案格式，您可以選擇相應的檔案選項。如果資料有加密，您也可以選取資料已加密，然後輸入用來加密資料之 KMS 金鑰的 Amazon Resource Name (ARN)。

   如果您選擇 CSV 或 DELIMITER」，則也可以選擇分隔符號字元，以及是否要忽略標頭資料列 (如果指定的資料列數目實際上是資料欄名稱而非要載入的資料)。
6. (選擇性) 進階設定支援各種資料轉換參數和載入操作。視需要為您的檔案輸入此資訊。

   如需資料轉換和資料載入參數的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的資料轉換參數和資料載入操作。
7. 選擇下一步。
8. 選擇載入新資料表。
9. 確認或選擇在其中載入資料的目標資料表的位置，包括叢集或工作群組、資料庫和結構描述。輸入要建立的資料表名稱。
10. 選擇建立資料表，使用所顯示的定義建立資料表。

此時會顯示資料表定義的檢閱摘要。資料表會建立到資料庫中。稍後若要刪除資料表，請執行 DROP TABLE SQL 命令。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 DROP TABLE。

11. 選擇載入資料以開始載入資料。

    載入完成時，會顯示訊息以指出載入是否成功。如果成功，您便可立即使用 SQL 從所載入的資料表中選取資料。如果發生錯誤，請查詢系統檢視 STL_LOAD_ERRORS 以取得更多詳細資訊。如需有關 COPY 命令錯誤的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 STL_LOAD_ERRORS。

    用來載入資料的 COPY 命令範本會出現在您的查詢歷史記錄中。此 COPY 命令範本會顯示一些已使用的參數，但其無法直接在編輯器索引標籤中執行。如需查詢歷史記錄的相關資訊，請參閱檢視查詢和標籤歷史記錄。

使用 Amazon Redshift 編寫查詢

您可以在編輯器中輸入查詢，或從查詢清單中選取已儲存的查詢，然後選擇執行。

系統預設會設定限制 100 個以將結果限制為 100 個資料列。您可以關閉此選項，以傳回較大的結果集。如果您關閉這個選項，又想要避免結果集太大，則可以在 SQL 陳述式中包含 LIMIT 選項。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 ORDER BY 子句。

若要在結果區域中顯示查詢計劃，請開啟解釋。開啟解釋圖形，讓結果也顯示以圖形表示的解釋計劃。

若要將查詢儲存至查詢資料夾，請選擇儲存。

如果查詢成功，則會出現成功訊息。如果查詢傳回資訊，結果會顯示在結果區段中。如果結果數目超過顯示區域，結果區域的頂端會出現數字。您可以選擇數字以顯示連續的結果頁面。

您可以為每個資料欄篩選和排序結果。若要在結果資料欄標頭中輸入篩選條件，請將游標暫留在該資料欄上，以查看可供您輸入資料欄篩選條件的選單 (☰)。

如果查詢中包含錯誤，查詢編輯器 v2 會在結果區域中顯示錯誤訊息。該訊息會提供有關如何更正查詢的資訊。

您可以在結果區域中使用內容 (右鍵) 選單來匯出或複製查詢結果，如下所示：

- 選擇複製到和 JSON 或 CSV，將選取的資料列下載至檔案。

- 選擇複製資料列，以將選取的資料列複製到剪貼簿。
- 選擇複製具有標頭的資料列，以將具有欄標頭的所選資料列複製到剪貼簿。

您也可以在結果區域中選擇匯出，然後選擇 JSON 或 CSV，以將整組資料列結果下載至檔案。結果集內的資料列數目可能會受限於查詢中的限制選項或 SQL `limit` 子句。所下載結果集的大小上限為 5 MB。

您也可以使用 Windows 上的快速鍵 Ctrl+C 或 macOS 上的 Cmd+C 將資料從目前的結果頁面複製到剪貼板。如果未選取任何資料列，則會將焦點所在的儲存格複製到剪貼簿。如果已選取資料列，則會將選取的資料列複製到剪貼簿。

若要新增查詢標籤，請選擇



圖示，然後選擇編輯器，其會出現在具有查詢標籤的資料列中。查詢標籤不一定會使用 `Isolated session`。使用隔離工作階段時，SQL 命令的結果 (例如，在某個編輯器標籤中建立暫存資料表) 無法在另一個編輯器標籤中看到。當您在查詢編輯器 v2 中開啟編輯器標籤時，預設會使用隔離工作階段。

若要執行查詢

1. 在查詢區域中，執行以下其中一項操作：
   - 輸入查詢。
   - 貼上您複製的查詢。
   - 選擇查詢資料夾、開啟內容選單 (右鍵) 儲存的查詢，然後選擇開啟查詢。
2. 確認您已為打算執行的 SQL 選擇了正確的叢集或工作群組和資料庫值。

   最初，您可以在樹狀檢視中選擇叢集或工作群組。也請在樹狀檢視中選擇資料庫。

   您可以使用每個編輯器索引標籤的隔離工作階段標頭附近的下拉式控制項，在每個編輯器索引標籤內變更叢集或工作群組和資料庫。

   對於每個編輯器標籤，您要選擇是否要在隔離工作階段中執行 SQL。隔離工作階段有自己的資料庫連線。請使用該連線來執行與其他查詢編輯器工作階段隔離的 SQL。如需連線的相關資訊，請參閱開啟查詢編輯器 v2。
3. 選擇執行。

   隨即會開啟結果區域，並顯示查詢結果。

顯示查詢的解釋計劃

1. 選取查詢。
2. 開啟解釋。

   預設也會開啟解釋圖形。
3. 選擇執行。

   查詢便會開始執行，解釋計劃則會顯示在查詢的結果區域中。

查詢編輯器 v2 支援以下功能：

- 您可以在一個查詢標籤中使用多個 SQL 陳述式編寫查詢。系統會依序執行查詢，並針對每個查詢開啟多個結果標籤。
- 您可以使用工作階段變數和暫存資料表編寫查詢。
- 您可以使用以 ${*parameter*} 指定的可取代參數來編寫查詢。您可以使用多個可取代參數來編寫 SQL 查詢，並在 SQL 陳述式中的多個位置使用相同的參數。

  在執行查詢時，系統會顯示一個視窗供您輸入參數值。每次執行查詢時，系統都會顯示視窗供您輸入參數值。

  如需範例，請參閱 範例：銷售額大於特定參數。
- 系統會自動為查詢建立版本。您可以選擇較早的查詢版本來執行。
- 不需要等待查詢完成，就能繼續您的工作流程。即使您關閉查詢編輯器，查詢仍會繼續執行。
- 在編寫查詢時，系統支援自動完成結構描述、資料表和資料欄名稱。

SQL 編輯器支援以下功能：

- SQL 中使用的開始和結束括號具有相符的顏色。編輯器中會顯示垂直線以幫助您讓括號相符。
- 您可以收合和展開 SQL 的各個部分。
- 您可以搜尋和取代 SQL 中的文字。
- 您可以使用快速鍵來執行數個常見的編輯任務。
- 編輯器中會反白顯示 SQL 錯誤，以方便您找到問題所在區域。

如需編輯功能的示範，請觀看下列影片：Amazon Redshift 查詢編輯器 v2 中的全新和增強編輯體驗。

查詢範例

您可以在下面找到可以執行的各種查詢類型的描述。

其中許多查詢所使用的資料來自 tickit 範例結構描述。如需載入範例 tickit 資料的相關資訊，請參閱將資料載入到資料庫。如需 tickit 範例資料的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的範例資料庫。

當您執行這些範例查詢時，請確認您在編輯器中選擇了正確的資料庫，例如 sample_data_dev。

主題

- 範例：設定工作階段變數
- 範例：按總銷售額排序的最佳活動
- 範例：銷售額大於特定參數
- 範例：建立暫存資料表
- 範例：從暫存資料表中選取

範例：設定工作階段變數

下列命令會針對工作階段將 search_path 伺服器組態參數設定為「public]。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SET 和 search_path。

```
set search_path to public;
```

範例：按總銷售額排序的最佳活動

下列查詢會尋找銷售額最多的活動。

```
select eventname, count(salesid) totalorders, sum(pricepaid) totalsales
from sales, event
where sales.eventid=event.eventid
group by eventname
order by 3;
```

以下是結果的部分清單。

```
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
...
```

範例：銷售額大於特定參數

下列查詢會尋找銷量大於 ${numberoforders} 所指定參數的銷售額。當參數值為 7 時，結果有 60 列。當您執行查詢時，查詢編輯器 v2 會顯示執行查詢表單視窗，以收集 SQL 陳述式中的參數值。

```
select salesid, qtysold
from sales
where qtysold > ${numberoforders}
order by 2;
```

以下是結果的部分清單。

```
salesid qtysold
20005 8
21279 8
130232 8
42737 8
74681 8
67103 8
105533 8
91620 8
121552 8
...
```

範例：建立暫存資料表

下列陳述式會透過選取「sales」和「event」資料表中的資訊來建立暫存資料表「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;
```

範例：從暫存資料表中選取

下列陳述式會從按訂單總量排序的暫存資料表「eventsalestemp」中選取活動、訂單總量和銷售總額。

```
select eventname,  totalorders,  totalsales
from eventsalestemp
order by 2;
```

以下是結果的部分清單。

```
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 中的筆記本

您可以使用筆記本在單一文件中組織、註釋和共用多個 SQL 查詢。您可以將多個 SQL 查詢和 Markdown 儲存格新增至筆記本。筆記本可讓您透過使用多個查詢和 Markdown 儲存格，將與資料分析相關聯的查詢和說明分組到單一文件中。您可以使用 Markdown 語法新增文字並設定外觀格式，以提供資料分析任務的內容和其他資訊。您可以與團隊成員共用筆記本。

若要使用筆記本，您必須將筆記本的許可新增至 IAM 主體 (IAM 使用者或 IAM 角色)。我們建議的最佳實務是，將許可政策附加到 IAM 角色，然後根據需要將其指派給使用者和群組。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。您可以將許可新增至其中一個查詢編輯器 v2 受管政策。如需詳細資訊，請參閱存取查詢編輯器 v2。

您可以循序執行筆記本的所有儲存格。筆記本的 SQL 查詢儲存格與查詢編輯器索引標籤的功能大多相同。如需詳細資訊，請參閱使用 Amazon Redshift 編寫查詢。以下是查詢編輯器索引標籤與筆記本中的 SQL 儲存格之間的差異。

- 沒有控制項可以在筆記本中的 SQL 陳述式上執行 `Explain`。
- 在筆記本中，每個 SQL 儲存格只能建立一個圖表。

您可以將筆記本匯出和匯入至使用查詢編輯器 v2 建立的檔案。檔案的副檔名為 .ipynb，檔案大小最多可以為 5 MB。SQL 和 Markdown 儲存格會儲存在檔案中。叢集或工作群組與資料庫不會儲存在匯出的筆記本中。當您開啟匯入的筆記本時，您要選擇叢集或工作群組，以及要在其中執行叢集或工作群組的資料庫。執行 SQL 儲存格之後，您便可以在結果索引標籤中選擇是否要將目前的結果頁面顯示為圖表。查詢的結果集不會儲存在筆記本中。

建立筆記本

您可以建立筆記本，以便在單一文件中組織、註釋和共用多個 SQL 查詢。

若要建立筆記本

1. 從導覽器功能表中選擇編輯器圖示
   ()。
2. 選擇加號圖示
   ()，
   然後選擇筆記本。

   根據預設，SQL 查詢儲存格會出現在筆記本中。
3. 在 SQL 查詢儲存格中，執行以下其中一項操作：
   - 輸入查詢。
   - 貼上您複製的查詢。
4. (選用) 選擇加號圖示
   ()，
   然後選擇Markdown 以新增 Markdown 儲存格，您可以在其中使用標準 Markdown 語法提供描述性或說明性的文字。
5. (選用) 選擇加號圖示
   ()，
   然後選擇 SQL 以插入 SQL 儲存格。

您可以使用鉛筆圖示
()
重新命名筆記本。

在功能表圖示

(•••)

中，您還可以在筆記本上執行下列操作：

- 與我的團隊共用 — 依標籤所定義與您的團隊共用筆記本。若要與團隊共用筆記本，請確定您已將主體標籤 sqlworkbench-team 設定為與帳戶中其他團隊成員相同的值。例如，管理員可以針對會計部門中的每個人，將值設定為 accounting-team。如需範例，請參閱 要使用查詢編輯器 v2 所需的許可。
- 匯出 — 將筆記本匯出至副檔名為 .ipynb 的本機檔案。
- 匯入查詢 - 將查詢從本機檔案匯入筆記本中的儲存格。您可以匯入具有 .sql 和 .txt 副檔名的檔案。
- 儲存版本 — 建立筆記本的版本。若要查看筆記本的版本，請導覽至已儲存的筆記本，然後開啟版本歷史記錄。
- 複製 — 建立筆記本的副本，並在新的筆記本索引標籤中開啟它。
- 捷徑 — 在編寫筆記本時顯示可用的捷徑。

匯入至筆記本

您可以將整個筆記本或個別 SQL 儲存格匯入查詢編輯器 v2 筆記本中。

若要將整個筆記本從本機檔案匯入至我的筆記本，請選擇

匯入，然後選擇匯入筆記本。瀏覽至包含筆記本的 .ipynb 檔案。筆記本會匯入到目前開啟的筆記本資料夾。您接著可以在筆記本編輯器中開啟筆記本。

若要將查詢從本機檔案匯入至筆記本中的 SQL 儲存格，請選擇

匯入，然後選擇匯入查詢。在匯入查詢視窗中，依照畫面上的指示，選擇可匯入至新筆記本或現有筆記本作為查詢的檔案和資料夾。檔案的副檔名必須為 `.sql` 或 `.txt`。每個查詢最多可包含 10,000 個字元。當您新增至現有筆記本時，您會從已儲存筆記本清單的所有筆記本中選擇筆記本。匯入的查詢會在筆記本結尾新增為 SQL 儲存格。當您選擇新的筆記本時，您會選擇筆記本的名稱，而筆記本會在目前開啟的已儲存筆記本資料夾中建立。

> **Note**
>
> 使用 TextEdit 應用程式在 macOS 上建立 `.sql` 檔案時，您可能會遇到檔案增加了額外的隱藏副檔名的問題。例如，在 TextEdit 中建立名為 `Test.sql` 的檔案，最終可能會儲存為 `Test.sql.rtf`。查詢編輯器 v2 不支援副檔名為 `.rtf` 的檔案。不過，如果您使用 TextEdit 建立 `.sql` 檔案，並將其儲存為純文字檔案，則該檔案會有額外的隱藏副檔名 `.txt`。例如，名為 `Text.sql` 的檔案可能會儲存為 `Text.sql.txt`。與 `.rtf` 副檔名不同的是，查詢編輯器 v2 支援副檔名為 `.txt` 的檔案，因此將查詢匯入至筆記本時支援 `Text.sql.txt`。

如需筆記本的示範，請觀看下列影片：查詢編輯器 v2 中的 Amazon Redshift SQL 筆記本。

查詢 AWS Glue Data Catalog

您可以使用查詢編輯器 v2 來查詢 中目錄化的資料 AWS Glue Data Catalog ，方法是使用特定 SQL 命令並授予本節中概述的許可。依預設， AWS Glue Data Catalog 會列為名為 `awsdatacatalog` 的查詢編輯器 v2 資料庫。並非所有 Amazon Redshift AWS 區域都可查詢 AWS Glue Data Catalog 。請使用 SHOW 命令來判斷此功能是否可用。如需詳細資訊 AWS Glue，請參閱什麼是 AWS Glue？ 《AWS Glue 開發人員指南》中的 。

> **Note**
>
> 只有 Amazon Redshift RG 或 RA3 節點類型叢集和 Amazon Redshift Serverless AWS Glue Data Catalog 才支援查詢 。

您可以設定資料倉儲，並使用下列 SQL 命令檢視目錄化的 AWS Glue 資料庫物件：

- SHOW — 顯示目前連線的資料倉儲是否已掛載 `awsdatacatalog`。例如，若要顯示 `data_catalog_auto_mount` 參數值，請執行：

```
SHOW data_catalog_auto_mount;
```

如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SHOW。

- ALTER SYSTEM — 變更 data_catalog_auto_mount 的系統層級組態。例如，若要將 data_catalog_auto_mount 參數值變更為 on，請執行：

```
ALTER SYSTEM SET data_catalog_auto_mount = on;
```

當佈建叢集重新啟動或無伺服器工作群組自動暫停並繼續時，變更便會生效。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 ALTER SYSTEM。

- SHOW SCHEMAS — 顯示結構描述的清單。名為 的資料庫中的結構描述awsdatacatalog代表 中編製目錄的 AWS Glue 資料庫 AWS Glue Data Catalog。例如，若要顯示這些結構描述，請執行：

```
SHOW SCHEMAS FROM DATABASE awsdatacatalog;
```

如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SHOW SCHEMAS。

- SHOW TABLES — 顯示結構描述中的資料表清單。例如，若要顯示在結構描述myglue執行中名為 awsdatacatalog的 AWS Glue Data Catalog 資料庫中的資料表：

```
SHOW TABLES FROM SCHEMA awsdatacatalog.myschema;
```

如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SHOW TABLES。

- SHOW COLUMNS — 顯示資料表中的資料欄清單。例如，若要顯示在結構描述myglue和資料表mytable執行中名為 awsdatacatalog的 AWS Glue Data Catalog 資料庫中的資料欄：

```
SHOW COLUMNS FROM TABLE awsdatacatalog.myglue.mytable;
```

如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SHOW COLUMNS。

若要授予 IAM 使用者或角色查詢 AWS Glue Data Catalog的許可，

1. 在樹狀檢視窗格中，使用資料庫使用者名稱和密碼身分驗證方法，連線至佈建叢集或無伺服器工作群組中的初始資料庫。例如，使用您在建立叢集或工作群組時使用的管理員使用者和密碼連線至 dev 資料庫。
2. 在編輯器索引標籤中，執行下列 SQL 陳述式以向 IAM 使用者授予 AWS Glue Data Catalog的存取權。

```
GRANT USAGE ON DATABASE awsdatacatalog to "IAM:myIAMUser"
```

其中 *IAM:myIAMUser* 是您想要向其授予 AWS Glue Data Catalog使用權限的 IAM 使用者。或者，您也可以將 *IAMR:myIAMRole* 的使用權限授予 IAM 角色。

3. 在樹狀檢視窗格中，編輯或刪除先前建立的叢集或工作群組連線。使用下列其中一種方式連線至叢集或工作群組：
   - 若要從叢集存取 awsdatacatalog 資料庫，您必須使用身分驗證方法使用您的 IAM 身分的暫時憑證。如需此身分驗證方法的相關資訊，請參閱連線至 Amazon Redshift 資料庫。您的查詢編輯器 v2 管理員可能需要設定帳戶的帳戶設定，才能在連線視窗上顯示此身分驗證方法。
   - 若要從工作群組存取 awsdatacatalog 資料庫，您必須使用身分驗證方法聯合身分使用者。如需此身分驗證方法的相關資訊，請參閱連線至 Amazon Redshift 資料庫。
4. 透過所授予的權限，您可以使用 IAM 身分來針對您的 AWS Glue Data Catalog執行 SQL。

連線後，您可以使用查詢編輯器 v2 來查詢 AWS Glue Data Catalog中所編目的資料。在查詢編輯器 v2 的樹狀檢視窗格上，選擇叢集或工作群組和 awsdatacatalog 資料庫。在編輯器或筆記本窗格中，確認您已選擇正確的叢集或工作群組。所選擇的資料庫應該是初始的 Amazon Redshift 資料庫，例如 dev。如需有關編寫查詢的資訊，請參閱使用 Amazon Redshift 編寫查詢和Amazon Redshift 中的筆記本。系統會保留名為 awsdatacatalog 的資料庫，以參考帳戶中的外部 Data Catalog 資料庫。對 awsdatacatalog 資料庫所進行的查詢只能是唯讀的。請使用由三個部分組成的標記法來參考 SELECT 陳述式中的資料表。其中第一部分是資料庫名稱，第二部分是 AWS Glue 資料庫名稱，第三部分是 AWS Glue 資料表名稱。

```
SELECT * FROM awsdatacatalog.<aws-glue-db-name>.<aws-glue-table-name>;
```

您可以執行各種案例來讀取 AWS Glue Data Catalog 資料並填入 Amazon Redshift 資料表。

下列範例 SQL 會聯結其中定義的兩個資料表 AWS Glue。

```
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;
```

下列範例 SQL 會建立 Amazon Redshift 資料表，並填入來自兩個 AWS Glue 資料表聯結的資料。

```
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;
```

查詢 Amazon S3 資料表 (預覽版)

您可以使用查詢編輯器 v2 來查詢掛載至 AWS Glue Data Catalog的 Amazon S3table 目錄中保存的資料。Amazon S3 資料表目錄會在建立 AWS Glue Data Catalog 時掛載到 ，並自動顯示為相同帳戶 AWS 區域 下相同 中所有佈建叢集和無伺服器工作群組上的外部資料庫。如需使用 Amazon Redshift 存取 Amazon S3 資料表的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的使用 Amazon Redshift 存取 Amazon S3 資料表。

查詢資料湖

您可以依照本教學課程中的任務集，查詢 Amazon S3 資料湖中的資料。首先，建立外部結構描述以參考 AWS Glue Data Catalog 中的外部資料庫。然後，您就可以查詢 Amazon S3 資料湖中的資料。

示範：查詢資料湖

如需如何查詢資料湖的示範，請觀看以下影片。從 Amazon Redshift 查詢編輯器 v2 查詢資料湖。

先決條件

在查詢編輯器 v2 中使用資料湖之前，請先確認您已在 Amazon Redshift 環境中設定以下項目：

- 使用 編目 Amazon S3 資料， AWS Glue 並啟用 Data Catalog AWS Lake Formation。
- 使用 AWS Glue 已啟用的 Data Catalog 為 Amazon Redshift 建立 IAM 角色 AWS Lake Formation。如需此程序的詳細資訊，請參閱使用 AWS Glue Data Catalog 啟用的 為 Amazon Redshift 建立 IAM 角色 AWS Lake Formation。如需使用 Redshift Spectrum 和 Lake Formation 的詳細資訊，請參閱搭配使用 Redshift Spectrum。 AWS Lake Formation
- 授予資料表的 SELECT 許可，以在 Lake Formation 資料庫中進行查詢。如需有關此程序的詳細資訊，請參閱授予資料表的 SELECT 許可，以在 Lake Formation 資料庫中進行查詢。

  您可以在 Lake Formation 主控台 (https://console.aws.amazon.com/lakeformation/)、許可區段、資料湖許可頁面中驗證 IAM 角色、 AWS Glue 資料庫和資料表是否具有適當的許可。
- 確認已連線的使用者擁有在 Amazon Redshift 資料庫中建立結構描述和存取資料湖中資料的許可。當您在查詢編輯器 v2 中連線到資料庫時，您要選擇包含憑證 (可以是資料庫使用者或 IAM 使用者)

的驗證方法。已連線的使用者必須具備適當的許可和資料庫權限，例如 `superuser`。建立叢集或工作群組的 Amazon Redshift `admin` 使用者擁有 `superuser` 權限，而且可以建立結構描述和管理 Redshift 資料庫。如需使用查詢編輯器 v2 來連線至資料庫的相關資訊，請參閱連線至 Amazon Redshift 資料庫。

建立外部結構描述

若要查詢 Amazon S3 資料湖中的資料，請先建立外部結構描述。外部結構描述會參考 AWS Glue Data Catalog 中的外部資料庫。

1. 在查詢編輯器 v2 的編輯器檢視中，選擇

   建立，然後選擇結構描述。
2. 輸入結構描述名稱。
3. 對於結構描述類型，選擇外部。
4. 在資料目錄詳細資訊中，區域預設為 AWS 區域 Redshift 資料庫所在的 。
5. 選擇外部結構描述將對應的AWS Glue 資料庫，以及包含 AWS Glue 資料表參考的資料庫。
6. 選擇具有在 Amazon S3 上查詢資料所需許可的 Amazon Redshift IAM 角色。
7. 選擇性地選擇具有 Data Catalog 許可的 IAM 角色。
8. 選擇建立結構描述。

   結構描述會出現在樹狀檢視面板中的資料庫下方。

在建立結構描述時，如果您收到資料庫的許可遭拒錯誤，請檢查已連線的使用者是否具有建立結構描述的資料庫權限。

在 Amazon S3 資料湖中查詢資料

請使用您在上一個程序中建立的結構描述。

1. 在樹狀檢視面板中，選擇結構描述。
2. 若要檢視資料表定義，請選擇資料表。隨即會顯示資料表資料欄和資料類型。
3. 若要查詢資料表，請選擇該資料表，然後在內容功能表 (按一下滑鼠右鍵) 中，選擇選取資料表以產生查詢。
4. 在編輯器中執行查詢。

下列範例 SQL 是由查詢編輯器 v2 產生，用於查詢名為 AWS Glue 資料表中的所有資料列flightscsv。為了簡單起見，輸出中顯示的資料欄和資料列已遭到截斷。

```
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
...
```

資料共用

您可以建立資料共用讓其他叢集上的使用者可以查詢資料。包含您要共用之資料的叢集稱為生產者叢集。您必須在生產者叢集上針對要共用的資料庫物件建立資料共用。您可以共用結構描述、資料表、檢視表和 SQL 使用者定義函數 (UDF)。要作為資料共用對象的叢集稱為取用者叢集。在取用者叢集上，您必須從資料共用建立資料庫。然後，取用者叢集上的使用者就可以查詢資料。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的開始使用資料共用。

建立資料共用

您必須在要作為生產者叢集的叢集上建立資料共用。若要進一步了解資料共用的考量，請參閱《Amazon Redshift 資料庫開發人員指南》中的 Amazon Redshift 中的資料共用考量。

1. 選擇您要使用的生產者叢集上的資料庫。
2. 建立資料共用。例如：

```
create datashare mysource;
```

3. 設定資料共用的許可。例如：

```
grant alter, share on datashare mysource to admin;
```

4. 設定您要共用之資料庫物件的許可。例如：

```
alter datashare mysource add schema public;
```

```
alter datashare mysource add table public.event;
```

5. 在取用者叢集命名空間上設定用來存取資料共用的許可。例如：

```
grant usage on datashare mysource to namespace '2b12345-1234-5678-9012-
bb1234567890';
```

顯示資料共用

您可以顯示已在生產者叢集上建立的資料共用。

1. 選擇生產者叢集。
2. 顯示資料共用。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
```

建立取用者資料庫

在取用者叢集上，您必須從資料共用建立資料庫。這些步驟會描述如何在同一帳戶的兩個叢集之間共用資料。如需跨 AWS 帳戶共用資料的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的跨 AWS 帳戶共用資料。

您可以使用 SQL 命令或查詢編輯器 v2 樹狀檢視面板來建立資料庫。

使用 SQL

1. 從您帳戶的資料共用和生產者叢集的命名空間建立資料庫。例如：

```
create database share_db from datashare mysource of account '123456789012'
 namespace 'p1234567-8765-4321-p10987654321';
```

2. 設定許可，以便使用者可以存取資料庫和結構描述。例如：

```
grant usage on database share_db to usernames;
```

```
grant usage on schema public to usernames;
```

使用查詢編輯器 v2 樹狀檢視面板

1. 選擇 建立，然後選擇資料庫。
2. 輸入資料庫名稱。
3. (選擇性) 選取使用者和群組，然後選擇資料庫使用者。
4. 選擇使用資料共用建立。
5. 選擇資料共用。
6. 選擇建立資料庫。

   新的資料共用資料庫便會顯示在查詢編輯器 v2 的樹狀檢視面板中。
7. 設定許可，以便使用者可以存取資料庫和結構描述。例如：

```
grant usage on database share_db to usernames;
```

```
grant usage on schema public to usernames;
```

查詢資料共用物件

在取用者叢集上，您可以使用由三個部分組成的標記法表示的完全合格物件名稱來查詢資料共用物件：資料庫、結構描述和物件名稱。

1. 在查詢編輯器 v2 的樹狀檢視面板中，選擇結構描述。
2. 若要檢視資料表定義，請選擇資料表。

   隨即會顯示資料表資料欄和資料類型。
3. 若要查詢資料表，請選擇該資料表，然後使用內容選單 (右鍵) 選擇選取資料表。
4. 使用 SELECT 命令查詢資料表。例如：

```
select top 10 * from test_db.public.event;
```

使用查詢編輯器 v2 的排程查詢

使用 Amazon Redshift 查詢編輯器 v2 可讓 SQL 查詢按照排程自動執行。排程查詢是依指定時間或間隔自動執行的 SQL 陳述式，可讓您有效率地管理重複的資料操作和分析任務。如果您希望在 Amazon Redshift 環境中簡化批次處理、產生定期報告或維護資料管道，建議您排程查詢。

排程查詢有助於自動化擷取、轉換和載入 (ETL) 工作流程、重新整理儀表板以顯示最新洞察，並且操作各種資料管理常式。以下頁面詳細說明建立、設定和管理排程查詢的程序，以最佳化 Amazon Redshift 工作負載。

使用查詢編輯器 v2 建立查詢排程

您可以使用 Amazon Redshift 查詢編輯器 v2 建立排程來執行 SQL 陳述式。您可以建立排程，以符合您業務需求的時間間隔執行 SQL 陳述式。當已排程查詢的執行時間來臨，查詢便會由 Amazon EventBridge 啟動，並使用 Amazon Redshift 資料 API。

建立排程以執行 SQL 陳述式

1. 在編輯器

   檢視中，選擇

   排程以建立排程來執行 SQL 陳述式。
2. 在定義排程時，您會提供下列資訊。

   - 擔任執行查詢所需許可的 IAM 角色。此 IAM 角色也會附加至您的叢集或工作群組。

- AWS Secrets Manager 或臨時登入資料的身分驗證值，以授權存取您的叢集或工作群組。資料 API 支援這些身分驗證方法。如需詳細資訊，請參閱驗證已排程的查詢。
- 資料庫所在的叢集或工作群組。
- 所要查詢的資料所在資料庫的名稱。
- 已排程查詢的名稱及其描述。查詢編輯器 v2 會在您提供的已排程查詢名稱前面加上「QS2-」。查詢編輯器 v1 會在其已排程查詢的名稱前面加上「QS-」。
- 要依排程執行的 SQL 陳述式。
- 排程頻率和重複選項，或定義排程的 cron 格式值。如需詳細資訊，請參閱《Amazon CloudWatch Events 使用者指南》中的 Cron 運算式。
- 您也可以選擇啟用標準 Amazon SNS 通知來監控所排程的查詢。您可能需要確認提供給 Amazon SNS 通知的電子郵件地址。檢查您的電子郵件是否有用來確認 Amazon SNS 通知電子郵件地址的連結。如需詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的電子郵件通知。如果您的查詢正在執行中，但在 SNS 主題中看不到發佈的訊息，請參閱《Amazon EventBridge 使用者指南》中的我的規則已執行，但我沒看到任何訊息發佈到我的 Amazon SNS 主題。

3. 選擇排程查詢以儲存並啟用排程，並將排程新增至已排程的查詢檢視中的查詢清單。

已排程的查詢



檢視會列出叢集和工作群組的已排程查詢。使用此檢視時，您可以顯示排程查詢詳細資訊、啟用或停用排程、編輯排程，以及刪除已排程的查詢。當您檢視查詢詳細資訊時，您也可以檢視依排程執行查詢的歷史記錄。

> **Note**
>
> 排程查詢執行只會在排程歷史記錄清單中存在 24 小時。依排程執行的查詢未出現在查詢編輯器 v2 的查詢歷史記錄檢視中。

排程查詢的示範

如需排程查詢的示範，請觀看下列影片。排程查詢的影片示範。

設定用來排程查詢的許可

若要排程查詢，定義排程的 AWS Identity and Access Management (IAM) 使用者以及與排程相關聯的 IAM 角色必須設定 IAM 許可，才能使用 Amazon EventBridge 和 Amazon Redshift Data API。若要接收來自已排程查詢的電子郵件，則還必須設定您選擇性指定的 Amazon SNS 通知。

以下說明使用 AWS 受管政策來提供許可的任務，但根據您的環境，您可能想要縮小允許許可的範圍。

若為已登入查詢編輯器 v2 的 IAM 使用者，請使用 IAM 主控台 (https://console.aws.amazon.com/iam/) 編輯 IAM 使用者。

- 除了執行 Amazon Redshift 和查詢編輯器 v2 操作的許可外，請將 `AmazonEventBridgeFullAccess`和 `AmazonRedshiftDataFullAccess` AWS 受管政策連接到 IAM 使用者。
- 或者，請將許可指派給角色，然後將角色指派給使用者。

  將允許 `sts:AssumeRole` 許可的政策附加到您在定義已排程查詢時所指定 IAM 角色的資源 ARN。如需有關擔任角色的相關資訊，請參閱《IAM 使用者指南》中的向使用者授予切換角色的許可。

  下列範例顯示在帳戶 `123456789012` 中擔任 IAM 角色 `myRedshiftRole` 的許可政策。IAM 角色 `myRedshiftRole` 也是附加至已排程查詢執行所在叢集或工作群組的 IAM 角色。

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AssumeIAMRole",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": [
                "arn:aws:iam::123456789012:role/myRedshiftRole"
            ]
        }
    ]
}
```

  更新用於排程查詢的 IAM 角色信任政策，以允許 IAM 使用者擔任該角色。

```
{
```

```
            "Sid": "AssumeRole",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/myIAMusername"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

針對您指定要允許已排程的查詢執行的 IAM 角色，請使用 IAM 主控台 (https://console.aws.amazon.com/iam/) 編輯 IAM 角色。

- 將 AmazonRedshiftDataFullAccess和 AmazonEventBridgeFullAccess AWS 受管政策連接至 IAM 角色。AmazonRedshiftDataFullAccess 受管政策只會針對標記了索引鍵 RedshiftDataFullAccess 的 Redshift Serverless 工作群組允許 redshift-serverless:GetCredentials 許可。

驗證已排程的查詢

在排程查詢時，您會在 SQL 執行時使用下列其中一種身分驗證方法。每種方法都需要在查詢編輯器 v2 上使用不同的輸入組合。用於執行 SQL 陳述式的資料 API 支援這些身分驗證方法。

用於執行查詢的資料庫使用者或角色必須具備必要的資料庫權限。例如，若要授予資料表 mytable 的 IAMR:MyRedshiftQEv2Scheduler 權限，請執行下列 SQL 命令。

```
GRANT all ON TABLE mytable TO "IAMR:MyRedshiftQEv2Scheduler";
```

若要檢視叢集或工作群組中的資料庫使用者清單，請查詢系統檢視 PG_USER_INFO。

> **Note**
>
> 您為其排程查詢的任何 Redshift Serverless 工作群組也必須標記索引鍵 RedshiftDataFullAccess。如需詳細資訊，請參閱授權 Amazon Redshift 資料 API 的存取。
>
> 若要替代標記工作群組的做法，您也可以將內嵌政策新增至允許 redshift-serverless:GetCredentials 的 IAM 角色 (隨排程指定)。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

使用此方法時，請為儲存在 AWS Secrets Manager中的 secret-arn 提供機密值。此秘密包含用來連線至資料庫的憑證。您可能已在建立叢集或工作群組時，建立了具有適當憑證的機密。此機密必須標記了索引鍵 `RedshiftDataFullAccess`。如果標籤索引鍵尚未存在，請使用 AWS Secrets Manager 主控台來新增它。如需有關建立機密的資訊，請參閱 建立資料庫連線憑證的機密。

如需最基本許可的相關資訊，請參閱《AWS Secrets Manager 使用者指南》中的使用 AWS Secrets Manager建立和管理機密。

臨時憑證

使用此方法時，請在連線到叢集中的資料庫時提供您的資料庫名稱和資料庫使用者值。您只需要在連線至工作群組中的資料庫時提供資料庫名稱。

在連線至叢集時，`AmazonRedshiftDataFullAccess` 政策會允許名為 `redshift_data_api_user` 的資料庫使用者獲得 `redshift:GetClusterCredentials` 的許可。如果您想要使用不同的資料庫使用者來執行 SQL 陳述式，請在附加至叢集的 IAM 角色中新增政策以允許 `redshift:GetClusterCredentials`。下列範例政策會允許資料庫使用者 `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"
            ]
        }
    ]
}
```

設定用來檢視排程查詢歷史記錄的許可

若要允許使用者檢視排程查詢歷史記錄，請編輯 IAM 角色 (隨排程指定) 信任關係以新增許的。

以下是 IAM 角色中的信任政策範例，該政策允許 IAM 使用者 *myIAMusername* 檢視排程查詢歷史記錄。您不必允許 IAM 使用者獲得 `sts:AssumeRole` 許可，而是可以選擇允許 IAM 角色獲得此許可。

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"
        }
    ]
}
```

監控已排程的查詢

對於您指定要傳送電子郵件通知的 Amazon SNS 主題，請透過導覽至 SNS 通知區段、開啟監控，然後使用建立 SNS 主題建立主題，以使用查詢編輯器 v2 建立 Amazon SNS 主題。查詢編輯器 v2 會建立 Amazon SNS 主題，並將服務主體新增至 Amazon EventBridge 的存取政策。以下是在 Amazon SNS 主題中建立的存取政策範例。在此範例中，使用 the AWS 區域 *us-west-2*、 AWS 帳戶 *123456789012* 和 Amazon SNS 主題 *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"
    }
```

```
  ]
}
```

排程查詢執行時，Amazon SNS 會傳送 AWS 通知電子郵件。下列範例顯示針對在 AWS 帳戶 *123456789012* 中的 AWS 區域 *eu-north-1* 上執行的已排程查詢 *QS2-may25a*，使用 Amazon SNS 通知主題 *may25a-SNS* 傳送至 *myemail@example.com* 的電子郵件。

```
{"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
```

針對排程查詢的設定進行故障診斷

如果您在排程查詢時遇到問題，請考慮下列事項。

查詢未執行

檢查排程中使用的 IAM 角色是否有可取得暫時叢集憑證的許可。已佈建叢集的許可為 `redshift:GetClusterCredentialsWithIAM`。Redshift Serverless 工作群組的許可為 `redshift-serverless:GetCredentials`。

已排程歷史記錄未顯示

用於登入 AWS 主控台的 IAM 使用者或 IAM 角色並未新增至用於排程查詢之 IAM 角色的信任政策中。

使用 AWS Secrets Manager 進行排程查詢連線時，請確認秘密已加上金鑰 的標籤`RedshiftDataFullAccess`。

如果排程的查詢使用 AWS Secrets Manager 連線，則用於排程查詢的 IAM 角色必須`SecretsManagerReadWrite`具有連接到角色的同等受管政策。

查詢歷史記錄狀態為 **Failed**

如需查詢失敗原因的詳細資訊，請檢視 SYS_QUERY_HISTORY 系統檢視。常見問題是用來執行查詢的資料庫使用者或角色可能沒有要執行 SQL 所需的權限。如需詳細資訊，請參閱驗證已排程的查詢。

下列 SQL 會查詢 SYS_QUERY_HISTORY 檢視以傳回失敗的查詢。

```
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';
```

若要了解特定的失敗已排程查詢的詳細資訊，請參閱使用 檢視排程查詢的結果 AWS CloudShell。

使用 檢視排程查詢的結果 AWS CloudShell

您可以使用 AWS CloudShell 來了解排程查詢的詳細資訊。您必須擁有適當的許可，才能執行下列程序中顯示的 AWS CLI 命令。

檢視已排程查詢的結果

1. 在 AWS 主控台上，開啟 AWS CloudShell 命令提示字元。如需詳細資訊 AWS CloudShell，請參閱AWS CloudShell 《 使用者指南》中的什麼是 AWS CloudShell 。
2. 擔任已排程查詢的 IAM 角色。若要擔任該角色，請在查詢編輯器 v2 中尋找與已排程查詢相關聯的 IAM 角色，並在 AWS CloudShell中的 AWS CLI 命令內使用該角色。例如，針對角色 `scheduler`，請輸入 AWS STS 命令來擔任已排程的查詢所使用的角色。

```
aws sts assume-role --role-arn "arn:aws:iam::123456789012:role/scheduler" --role-
session-name "scheduler-test"
```

傳回的憑證類似以下內容。

```
"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. AWS CLI 使用擔任 IAM 角色中顯示的登入資料，在 中建立環境變數。您必須使用這些字符，直到其到期。例如，您可以在 中輸入下列項目 AWS CloudShell。

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
export AWS_SESSION_TOKEN=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY...
```

4. 若要檢視失敗查詢的錯誤，請執行 AWS CLI 命令來描述陳述式。SQL 陳述式的 ID 來自查詢編輯器 v2 中已排程查詢之排程歷史記錄區段中顯示的 ID。

```
aws redshift-data describe-statement --id 130d2620-05d2-439c-b7cf-815d9767f513
```

在此範例中，已排程的 SQL `select * from users limit 100` 會導致 `users` 資料表不存在的 SQL 錯誤。

```
{
"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"
}
```

將查詢結果視覺化

在執行查詢並顯示結果後，您可以開啟圖表來顯示透過圖形加以視覺化的目前結果頁面。您可以使用下列控制項來定義圖表的內容、結構和外觀：



追蹤

代表圖表中的一組相關圖形標記。您可以在圖表中定義多個追蹤。

Type

您可以定義追蹤類型以將資料表示為下列其中一項：

- 散佈圖或氣泡圖。
- 長條圖，以垂直或水平長條表示資料類別。
- 區域圖，以定義填充區域。
- 直方圖，以使用長條表示頻率的分佈。
- 圓餅圖，以圓形表示資料，其中的每個切片各表示在整體中所佔的百分比。
- 漏斗圖或漏斗區域圖，以透過各個程序階段表示資料。
- OHLC (開盤-最高-最低-收盤) 圖，通常用於財務資料，以沿著 x 軸 (通常代表時間間隔) 表示開盤價、最高價、最低價和收盤價。
- 蠟燭圖，在時間線上表示某個類別一系列的值。
- 瀑布圖，透過一系列中間值來表示初始值的增減變化。這些值可以表示時間間隔或類別。
- 折線圖，表示值隨著時間推移的變化情形。

X 軸

您必須指定要沿著 X 軸繪製、含有值的資料表資料欄。包含描述性值的資料欄通常代表維度資料。包含量化值的資料欄通常代表事實資料。

Y 軸

您必須指定要沿著 Y 軸繪製、含有值的資料表資料欄。包含描述性值的資料欄通常代表維度資料。包含量化值的資料欄通常代表事實資料。

子圖

您可以定義圖表資料的其他呈現方式。

轉換

您可以定義轉換以篩選追蹤資料。您必須使用分割轉換以顯示來自單一來源追蹤的多個追蹤。您必須使用彙總轉換以將追蹤顯示為平均值或最小值。您必須使用排序轉換來排序追蹤。

一般外觀

您可以針對下列項目設定預設值：背景顏色、邊界顏色、用來設計色盤的色階、文字樣式和大小、標題樣式和大小，以及模式列。您可以定義拖曳、點按和暫留的互動效果。您可以定義中繼文字。您可以定義追蹤、軸、圖例和註釋的預設外觀。

建立圖表

1. 執行查詢並取得結果。
2. 開啟圖表。
3. 選擇追蹤並開始將資料視覺化。
4. 從下列其中一個選項中選擇圖表樣式：
   - 散佈圖
   - 長條圖
   - 區域圖
   - 直方圖
   - 圓餅圖
   - 漏斗圖
   - 漏斗區域圖
   - OHLC (開盤-最高-最低-收盤) 圖
   - 蠟燭圖
   - 瀑布圖
   - 折線圖
5. 選擇樣式以自訂外觀，包括顏色、軸、圖例和註釋。您可以新增文字、形狀和圖像。
6. 選擇註釋以新增文字、形狀和圖像。
7. 選擇重新整理以更新圖表顯示。選擇全螢幕以展開圖表顯示。

範例：建立圓餅圖以將查詢結果視覺化

下列範例會使用範例資料庫的「Sales」資料表。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的範例資料庫。

以下是您為了提供圓餅圖資料所執行的查詢。

```
select top 5 eventname, count(salesid) totalorders, sum(pricepaid) totalsales
from sales, event
where sales.eventid=event.eventid group by eventname
order by 3;
```

為按總銷售額排序的最佳活動建立圓餅圖

1. 執行查詢。
2. 在查詢結果區域中，開啟圖表。
3. 選擇追蹤。
4. 對於類型，選擇圓餅圖。
5. 對於值，選擇 totalsales。
6. 對於標籤，選擇 eventname。
7. 選擇樣式，然後選擇一般。
8. 在色階底下，選擇分類，然後選擇 Pastel2。



範例：建立用於比較營收與銷售額的組合圖表

執行此範例中的步驟，以建立由營收資料的長條圖與銷售額資料的折線圖組合而成的圖表。下列範例會使用 tickit 範例資料庫的「Sales」資料表。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的範例資料庫。

以下是您為了提供圖表資料所執行的查詢。

```
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;
```

建立用於比較營收與銷售額的組合圖表

1. 執行查詢。
2. 在查詢結果區域中，開啟圖表。
3. 在追蹤 o 底下，對於類型，選擇長條圖。
4. 對於 X，選擇 eventname。
5. 對於 Y，請選擇 total_price。

   長條圖會沿著 X 軸顯示活動名稱。
6. 在樣式底下，選擇追蹤。
7. 對於名稱，輸入營收。
8. 在樣式底下，選擇軸。
9. 對於標題，選擇 Y 並輸入營收。

   營收標籤隨即會顯示在左側 Y 軸上。
10. 在結構底下，選擇追蹤。
11. 選擇

    ＋

    追蹤。

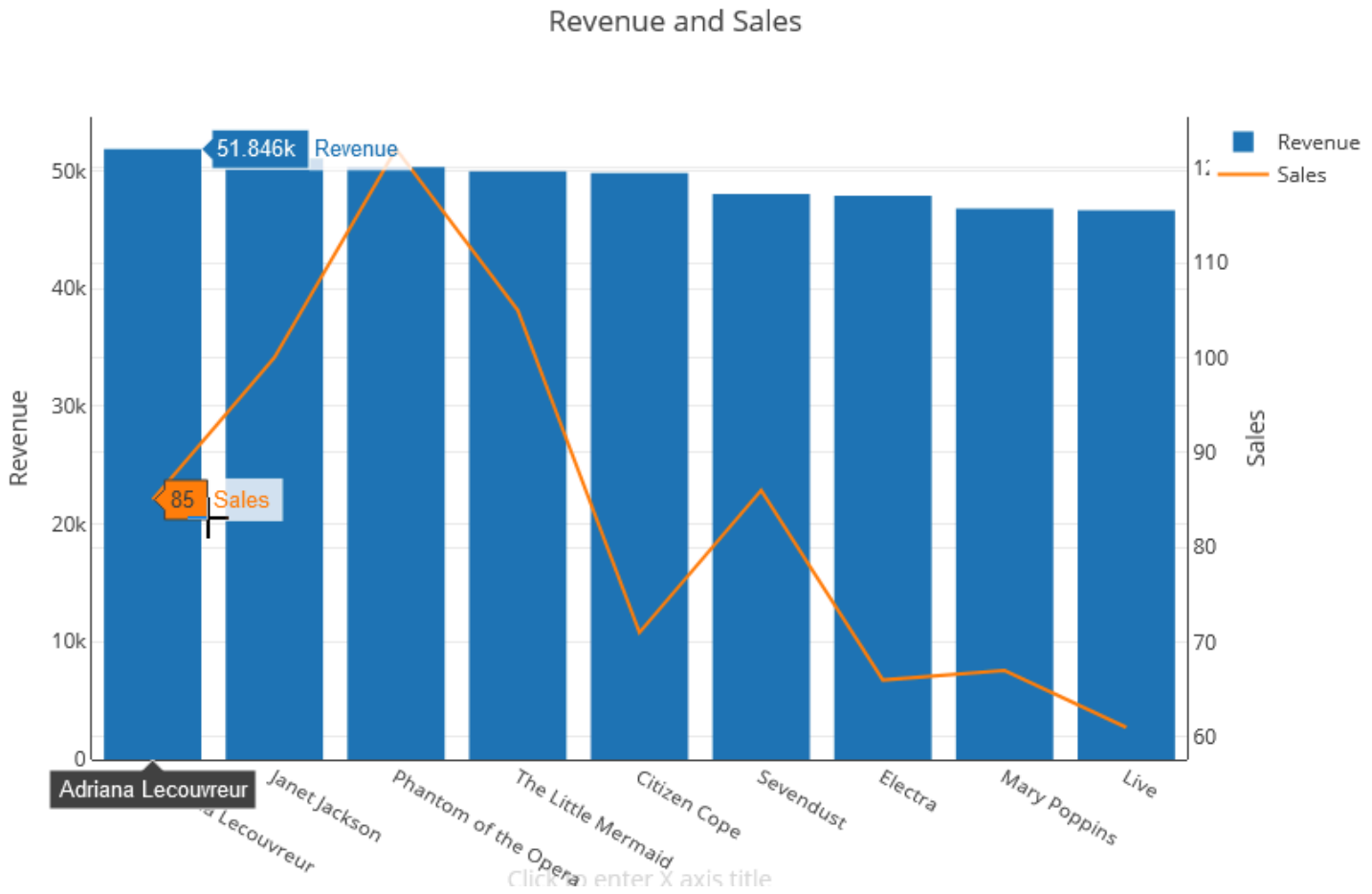
追蹤 1 的選項隨即會顯示出來。

12. 對於類型，選擇折線圖。
13. 對於 X，選擇 eventname。
14. 對於 Y，選擇 total_qty_sold。
15. 在要使用的軸底下，對於 Y 軸選擇

    +

    。

    Y 軸便會顯示 Y2。
16. 在樣式底下，選擇軸。
17. 在標題底下，選擇 Y2。
18. 對於名稱，輸入 Sales。
19. 在折線圖底下，選擇 Y:Sales。
20. 在軸線底下，選擇顯示，然後對於位置，選擇右。



示範：使用 Amazon Redshift 查詢編輯器 v2 建置視覺效果

如需如何建置視覺效果的示範，請觀看下列影片。使用 Amazon Redshift 查詢編輯器 v2 建置視覺效果。

以團隊的形式協作和共用

您可以與團隊共用查詢。

您可以為一組會協作和共用查詢編輯器 v2 資源的使用者定義團隊。管理員可以透過將標籤新增至 IAM 角色來建立團隊。如需詳細資訊，請參閱要使用查詢編輯器 v2 所需的許可。

儲存和瀏覽查詢

在與團隊共用查詢前，請先儲存查詢。您可以檢視和刪除已儲存的查詢。

儲存查詢

1. 準備您的查詢並選擇儲存。
2. 為查詢輸入標題。
3. 選擇儲存。

瀏覽已儲存的查詢

1. 從導覽窗格選擇查詢。
2. 您可以檢視屬於我的查詢、由我共用或共用給我的團隊的查詢。這些查詢能以個別查詢的形式出現，也能出現在您建立的資料夾內。

共用查詢

您可以與團隊共用查詢。您也可以檢視已儲存查詢的歷史記錄並管理查詢版本。

若要與團隊共用查詢，請確定您已將主體標籤 `sqlworkbench-team` 設定為與帳戶中其他團隊成員相同的值。例如，管理員可以針對會計部門中的每個人，將值設定為 `accounting-team`。如需範例，請參閱 要使用查詢編輯器 v2 所需的許可。

與團隊共用查詢

1. 從導覽窗格選擇查詢。

2. 開啟您要共用之查詢的內容 (右鍵) 選單，然後選擇與我的團隊共用。
3. 選擇您要與其共用查詢的團隊，然後選擇儲存共用選項。

管理查詢版本

每次儲存 SQL 查詢時，查詢編輯器 v2 都會將其儲存為新的版本。您可以瀏覽較舊的查詢版本、儲存查詢的複本或還原查詢。

管理查詢版本

1. 從導覽窗格選擇查詢。
2. 開啟您要使用之查詢的內容 (右鍵) 選單。
3. 選擇版本歷史記錄以開啟查詢的版本清單。
4. 在版本歷史記錄頁面上，您可以執行下列操作：
   - 回復為選取項 — 回復為選取的版本，並繼續使用此版本。
   - 另存選取項 — 在編輯器中建立新查詢。

使用 Amazon Redshift 查詢編輯器 v1 查詢資料庫

若要在 Amazon Redshift 叢集託管的資料庫上執行查詢，簡單的辦法是使用查詢編輯器。建立叢集後，您可以在 Amazon Redshift 主控台中使用查詢編輯器，立即執行查詢。

> **Note**
>
> 您無法使用這個原始查詢編輯器在 Amazon Redshift Serverless 中查詢資料。請改用 Amazon Redshift 查詢編輯器 v2。

2021 年 2 月，我們部署了更新的查詢編輯器，並變更了可使用查詢編輯器的授權許可。新的查詢編輯器會使用 Amazon Redshift 資料 API 來執行查詢。`AmazonRedshiftQueryEditor` 政策是 AWS 受管 AWS Identity and Access Management (IAM) 政策，已更新為包含必要的許可。如果您有自訂 IAM 政策，請務必加以更新。請使用 `AmazonRedshiftQueryEditor` 作為指南。`AmazonRedshiftQueryEditor` 的變更包括下列內容：

- 可管理查詢編輯器陳述式結果的許可需要陳述式擁有者使用者。

- 已新增使用 Secrets Manager 連線至資料庫的許可。

如需詳細資訊，請參閱要使用 Amazon Redshift 主控台查詢編輯器所需的許可。

當您從新的查詢編輯器連線到叢集時，您可以使用下列兩種身分驗證方法之一。

您可以使用查詢編輯器執行下列作業：

- 執行單一 SQL 陳述式查詢。
- 將最大 100 MB 的結果集下載到逗號分隔值 (CSV) 檔案。
- 儲存查詢供重複使用。您無法在歐洲 (巴黎) 區域、亞太區域 (大阪) 區域、亞太區域 (香港) 區域或中東 (巴林) 區域儲存查詢。
- 檢視使用者定義資料表的查詢執行時間詳細資訊。
- 將查詢排程在未來的時間執行。
- 檢視您在查詢編輯器中所建立查詢的歷史記錄。
- 使用增強的 VPC 路由對叢集執行查詢。

查詢編輯器考量

在使用查詢編輯器時，請考慮下列有關使用查詢的事項：

- 查詢的持續時間上限為 24 小時。
- 查詢結果大小上限為 100 MB。如果呼叫傳回的回應資料超過 100 MB，系統會終止呼叫。
- 查詢結果的保留時間上限為 24 小時。
- 查詢陳述式的大小上限為 100 KB。
- 叢集必須位於以 Amazon VPC 服務為基礎的虛擬私有雲端 (VPC) 中。
- 您不能在查詢編輯器中使用交易。如需交易的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 BEGIN。
- 您可以儲存長度最多 3,000 個字元的查詢。

使用 SQL 用戶端工具連線至 Amazon Redshift 資料倉儲

您可以從 SQL 用戶端工具透過 Java 資料庫連線 (JDBC)、Python 和開放式資料庫連線 (ODBC) 的連線來連線至 Amazon Redshift 資料倉儲。Amazon Redshift 不會提供或安裝任何 SQL 用戶端工具或

程式庫。若要使用這些工具或程式庫來處理資料倉儲中的資料，請將其安裝在用戶端電腦或 Amazon EC2 執行個體上。您可以使用支援 JDBC、Python 或 ODBC 驅動程式的大多數 SQL 用戶端工具。

使用本主題結尾的章節清單，可協助您逐步完成將用戶端電腦或 Amazon EC2 執行個體設定為使用 JDBC、Python 或 ODBC 連線的程序。這些主題也會討論用戶端與伺服器連線的相關安全選項。此外，請尋找有關從 SQL 用戶端工具 (例如 Amazon Redshift RSQL) 進行設定和連線的資訊。如果您還沒有商業智慧工具可使用，則可以嘗試這些工具。您也可以利用本節來了解如何連線至叢集。最後，如果您在嘗試連線至資料倉儲時遇到問題，您可以檢閱故障診斷資訊來找出解決方案。

透過用戶端工具連線的建議

如果您使用 IP 位址連線至 Redshift 叢集，當發生中斷或連線中斷，且叢集在新的可用區域 (AZ) 中上線時，可能會導致更長的停機時間。不過，如果您仍希望應用程式使用 IP 位址連線到 Redshift，請使用附加到叢集的虛擬私有雲端 (VPC) 端點的私有 IP 位址。您可以在屬性索引標籤下，於網路和安全的叢集詳細資訊中找到此資訊。

> **Note**
>
> 如果您的應用程式使用領導節點的 IP 位址來存取 Redshift 叢集，則建議的最佳實務是改為使用叢集端點 URL。如需詳細資訊，請參閱在 Amazon Redshift 中設定連線。

主題

- 在 Amazon Redshift 中設定連線
- 探索中繼資料的最佳實務
- 設定連線的安全選項
- 從用戶端工具和程式碼連線
- 使用身分驗證設定檔連線至 Amazon Redshift
- 針對 Amazon Redshift 中的連線問題進行疑難排解

在 Amazon Redshift 中設定連線

在下一節中，您會了解如何設定 JDBC、Python 和 ODBC 連線，以從 SQL 用戶端工具連線至叢集。本節描述如何設定 JDBC、Python 和 ODBC 連線。也描述如何使用 Secure Sockets Layer (SSL) 和伺服器憑證，以加密用戶端與伺服器之間的通訊。

適用於 Amazon Redshift 的 JDBC、Python 和 ODBC 驅動程式

若要處理叢集中的資料，您必須具有 JDBC、Python 或 ODBC 驅動程式，以便從用戶端電腦或執行個體來連線。將應用程式編寫為使用 JDBC、Python 或 ODBC 資料存取 API 操作，並使用支援 JDBC、Python 或 ODBC 的 SQL 用戶端工具。

Amazon Redshift 提供了 JDBC、Python 和 ODBC 驅動程式供您下載。支援這些驅動程式 支援。PostgreSQL 驅動程式未經過測試，因此未獲得 Amazon Redshift 團隊的支援。在連線至 Amazon Redshift 叢集時，請使用 Amazon Redshift 特定的驅動程式。Amazon Redshift 驅動程式有下列優點：

- 支援 IAM、SSO 和聯合身分驗證。
- 支援新的 Amazon Redshift 資料類型。
- 支援身分驗證設定檔。
- 與 Amazon Redshift 增強功能搭配使用而提升了效能。

如需如何下載 JDBC 和 ODBC 驅動程式及設定叢集連線的相關資訊，請參閱 設定適用於 Amazon Redshift 的 JDBC 驅動器 2.x 版的連線、Amazon Redshift Python 連接器 和 設定 Amazon Redshift ODBC 驅動程式 2.x 版連線。

如需如何管理 IAM 身分的相關資訊 (包括 IAM 角色的最佳實務)，請參閱 Amazon Redshift 中的身分和存取管理。

尋找叢集連線字串

若要使用 SQL 用戶端工具來連線至叢集，您必須有叢集連線字串。您可以在 Amazon Redshift 主控台的叢集詳細資訊頁面上找到叢集連線字串。

尋找叢集的連線字串

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後從清單中選擇叢集名稱以開啟其詳細資訊。
3. General information (一般資訊) 區段中提供了 JDBC URL 和 ODBC URL 連線字串，以及其他詳細資訊。每個字串都以叢集執行 AWS 的區域為基礎。按一下適當連線字串旁邊的圖示以複製連線字串。

若要連線至叢集端點，您可以從 DescribeClusters API 請求使用叢集端點 URL。以下是叢集端點 URL 的範例。

```
mycluster.cmeaswqeuae.us-east-2.redshift.amazonaws.com
```

如果您已經為叢集設定了自訂網域名稱，您也可以使用該名稱來連線至叢集。如需建立自訂網域名稱的相關資訊，請參閱設定自訂網域名稱。

> **Note**
>
> 連線時，請勿使用叢集節點的 IP 地址或 VPC 端點的 IP 地址。請一律使用 Redshift 端點，以避免發生不必要的中斷。唯一會使用端點 URL 的例外狀況是當您使用自訂網域名稱時。如需詳細資訊，請參閱使用自訂網域名稱來進行用戶端連線。

設定適用於 Amazon Redshift 的 JDBC 驅動器 2.x 版的連線

您可以從許多第三方 SQL 用戶端工具使用 JDBC 驅動器 2.x 版連線來連線至 Amazon Redshift 叢集。Amazon Redshift JDBC 連接器會提供開放原始碼的解決方案。您可以瀏覽原始碼、請求增強功能、回報問題並提供貢獻。

如需 JDBC 驅動器變更的最新資訊，請參閱變更日誌。

依預設，Amazon Redshift JDBC 驅動程式會設定為使用 TCP 保持連線，以防止連線逾時。您可以指定驅動程式何時開始傳送保持連線封包，或在連線 URL 中設定相關屬性來關閉此功能。如需連線 URL 語法的相關資訊，請參閱建置連線 URL。

屬性	說明
`TCPKeepAlive`	若要關閉 TCP 保持連線，請將此屬性設定為 `FALSE`。

主題

- 下載 Amazon Redshift JDBC 驅動程式 2.x 版
- 安裝 Amazon Redshift JDBC 驅動程式 2.x 版
- 取得 JDBC URL
- 建置連線 URL
- 使用 Apache Maven 設定 JDBC 連線

- 設定身分驗證和 SSL
- 設定 記錄
- 資料類型轉換
- 使用預備陳述式支援
- JDBC 驅動程式的 2.x 和 1.x 版本之間的差異
- 建立 JDBC 驅動器 2.x 版的初始化 (.ini) 檔案
- JDBC 驅動器 2.x 版組態的選項
- JDBC 驅動器 2.x 版之前的版本

下載 Amazon Redshift JDBC 驅動程式 2.x 版

> **Note**
>
> Amazon Redshift JDBC 2.x 驅動器並非專為執行緒安全所設計。若有兩個或多個執行緒嘗試以並行方式使用相同連線，可能會導致死結、錯誤、不正確的結果或其他非預期的行為。如果您擁有多執行緒應用程式，建議您同步驅動器的存取，以避免並行存取。

Amazon Redshift 為與 JDBC 4.2 API 相容的工具提供了驅動程式。此驅動程式的類別名稱為 `com.amazon.redshift.Driver`。

如需如何安裝 JDBC 驅動程式、參考 JDBC 驅動程式庫，以及註冊驅動程式類別的詳細資訊，請參閱下列主題。

對於您使用 Amazon Redshift JDBC 驅動器 2.x 版的每部電腦，請確定其已安裝 Java 執行階段環境 (JRE) 8.0。

如果您使用 Amazon Redshift JDBC 驅動程式進行資料庫身分驗證，請確定 Java 類別路徑中有 適用於 Java 的 AWS SDK 2.40.5 或更新版本。如果您尚未 適用於 Java 的 AWS SDK 安裝 ，請使用 AWS 開發套件的 JDBC 4.2 相容驅動程式和驅動程式相依程式庫下載 ZIP 檔案：

- JDBC 4.2 相容驅動程式 2.x 版和 AWS SDK 驅動程式相依程式庫

  此 ZIP 檔案包含 JDBC 4.2 相容的驅動程式 2.x 版和適用於 Java 的 AWS SDK 2.x 驅動程式相依程式庫檔案。將相依 jar 檔案解壓縮到與 JDBC 驅動程式相同的位置。只有 JDBC 驅動程式需要在 CLASSPATH 中。

此 ZIP 檔案不包含適用於 Java 2.x 的完整 AWS SDK。不過，它包含 AWS Identity and Access Management (IAM) 資料庫身分驗證所需的適用於 Java 的 AWS SDK 2.x 驅動程式相依程式庫。

將此 Amazon Redshift JDBC 驅動程式與 IAM 資料庫身分驗證所需的 AWS SDK 搭配使用。

若要安裝完整的適用於 Java 的 AWS SDK 2.x，請參閱《適用於 Java 的 AWS SDK 開發人員指南》中的AWS 適用於 Java 的 SDK 2.x。

- 與 JDBC 4.2 相容的驅動程式 2.x 版（不含 AWS SDK)

檢閱 JDBC 驅動器 2.x 版軟體授權和變更日誌檔案：

- JDBC 驅動器 2.x 版授權
- JDBC 驅動器 2.x 版變更日誌

JDBC 驅動程式 1.2.27.1051 版和更新版本支援 Amazon Redshift 預存程序。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的在 Amazon Redshift 中建立預存程序。

安裝 Amazon Redshift JDBC 驅動程式 2.x 版

若要安裝適用於 AWS SDK 的 Amazon Redshift JDBC 4.2 相容驅動程式 2.x 版和驅動程式相依程式庫，請將 ZIP 封存檔中的檔案擷取至您選擇的目錄。

若要安裝 Amazon Redshift JDBC 4.2 相容驅動器 2.x 版 (不含 AWS SDK)，請將 JAR 檔案複製到您選擇的目錄。

若要使用 Amazon Redshift JDBC 驅動程式存取 Amazon Redshift 資料存放區，您需要如下所述地執行組態。

主題

- 參考 JDBC 驅動程式庫
- 註冊驅動程式類別

參考 JDBC 驅動程式庫

您用來連線至資料的 JDBC 應用程式或 Java 程式碼必須存取驅動程式 JAR 檔案。請在應用程式或程式碼中，指定您從 ZIP 封存檔中解壓縮的所有 JAR 檔案。

在 JDBC 應用程式中使用驅動程式

JDBC 應用程式通常會提供一組用於新增驅動程式庫檔案清單的組態選項。請使用所提供的選項，將 ZIP 封存檔中的所有 JAR 檔案納入為應用程式中驅動程式組態的一部分。如需詳細資訊，請參閱 DNS 應用程式的文件。

在 Java 程式碼中使用驅動程式

您必須在類別路徑中包含所有驅動程式庫檔案。Java 執行階段環境會在此路徑中搜尋類別和其他資源檔案。如需詳細資訊，請參閱適當的 Java SE 文件，以便為您的作業系統設定類別路徑。

- 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：預設 macOS 類別路徑是安裝 JDBC 驅動程式的目錄。

註冊驅動程式類別

請務必為您的應用程式註冊適當的類別。您可以使用下列類別將 Amazon Redshift JDBC 驅動程式連線到 Amazon Redshift 資料存放區：

- `Driver` 類別會擴展 `java.sql.Driver`。
- `DataSource` 類別會擴展 `javax.sql.DataSource` 和 `javax.sql.ConnectionPoolDataSource`。

該驅動程式支援下列完整類別名稱 (與 JDBC 版本無關)：

- `com.amazon.redshift.jdbc.Driver`
- `com.amazon.redshift.jdbc.DataSource`

下列範例示範如何使用 DriverManager 類別建立 JDBC 4.2 連線。

```
            private static Connection connectViaDM() throws Exception
{
Connection connection = null;
connection = DriverManager.getConnection(CONNECTION_URL);
return connection;
}
```

下列範例示範如何使用 DataSource 類別建立連線。

```
 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;
}
```

取得 JDBC URL

您需要先知道叢集的 JDBC URL，才能從 SQL 用戶端工具連線至 Amazon Redshift 叢集。JDBC URL 採用下列格式：jdbc:redshift://*endpoint*:*port*/*database*

前面格式的欄位具有下列值。

欄位	Value
jdbc	連線的通訊協定。
redshift	子通訊協定，指定使用 Amazon Redshift 驅動程式來連線至資料庫。
endpoint	Amazon Redshift 叢集的端點。
port	您啟動叢集時指定的連接埠號碼。如果有防火牆，請確定此連接埠已開放給您使用。
database	您為叢集建立的資料庫。

以下是 JDBC URL 範例：jdbc:redshift://examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com:5439/dev

如果您的 URL 值包含下列任何 URI 保留字元，則值必須經過 URL 編碼：

- ;
- +
- {
- }
- [
-]
- &
- =
- ?
- 空格

例如，如果您的 PWD 值是 password:password，則使用該值的連線 URL 看起來如下：

```
jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/
dev;UID=amazon;PWD=password%3Apassword
```

如需如何取得 JDBC 連線的資訊，請參閱尋找叢集連線字串。

如果用戶端電腦無法連線至資料庫，您可以對可能的問題進行疑難排解。如需詳細資訊，請參閱針對 Amazon Redshift 中的連線問題進行疑難排解。

建置連線 URL

使用連線 URL 為您要存取的資料存放區提供連線資訊。以下是 Amazon Redshift JDBC 驅動器 2.x 版的連線 URL 格式。在這裡，[Host] 是 Amazon Redshift 伺服器的端點，[Port] 是伺服器用來接聽用戶端請求的傳輸控制通訊協定 (TCP) 連接埠號碼。

```
jdbc:redshift://[Host]:[Port]
```

以下是會指定某些選擇性設定的連線 URL 格式。

```
jdbc:redshift://[Host]:[Port]/[database];[Property1]=[Value];
[Property2]=[Value];
```

如果您的 URL 值包含下列任何 URI 保留字元，則值必須經過 URL 編碼：

- ;
- +
- {
- }
- [
-]
- &
- =
- ?
- 空格

例如，如果您的 PWD 值是 password:password，則使用該值的連線 URL 看起來如下：

```
jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/
dev;UID=amazon;PWD=password%3Apassword
```

例如，假設您想要連線至 AWS上美國西部 (加利佛尼亞北部) 區域中 Amazon Redshift 叢集上的連接埠 9000。您也想要存取名為 dev 的資料庫，並使用資料庫使用者名稱和密碼來驗證連線。在此情況下，您會使用下列連線 URL。

```
jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/
dev;UID=amazon;PWD=amazon
```

您可以使用下列字元將組態選項與 URL 字串的其餘部分隔開：

- ;
- ?

例如，下面兩個 URL 字串是相等的：

```
jdbc:redshift://my_host:5439/dev;ssl=true;defaultRowFetchSize=100
```

```
jdbc:redshift://my_host:5439/dev?ssl=true;defaultRowFetchSize=100
```

您可以使用下列字元將 URL 字串中的組態選項彼此隔開：

- ;
- &

例如，下面兩個 URL 字串是相等的：

```
jdbc:redshift://my_host:5439/dev;ssl=true;defaultRowFetchSize=100
```

```
jdbc:redshift://my_host:5439/dev;ssl=true&defaultRowFetchSize=100
```

下列 URL 範例會指定日誌層級 6 以及日誌的路徑。

```
jdbc:redshift://redshift.amazonaws.com:5439/dev;DSILogLevel=6;LogPath=/home/user/logs;
```

請勿複製連線 URL 中的屬性。

如需可指定之組態選項的完整清單，請參閱 JDBC 驅動器 2.x 版組態的選項。

> **Note**
>
> 連線時，請勿使用叢集節點的 IP 地址或 VPC 端點的 IP 地址。請一律使用 Redshift 端點，以避免發生不必要的中斷。唯一會使用端點 URL 的例外狀況是當您使用自訂網域名稱時。如需詳細資訊，請參閱使用自訂網域名稱來進行用戶端連線。

使用 Apache Maven 設定 JDBC 連線

Apache Maven 是軟體專案管理和理解工具。 適用於 Java 的 AWS SDK 支援 Apache Maven 專案。如需詳細資訊，請參閱《適用於 Java 的 AWS SDK 開發人員指南》中的搭配使用 SDK 與 Apache Maven。

如果您使用 Apache Maven，您可以將專案設定和建置為使用 Amazon Redshift JDBC 驅動程式來連線至 Amazon Redshift 叢集。若要執行此作業，請在專案的 pom.xml 檔案中，將 JDBC 驅動程式新增為相依性。如果您使用 Maven 來建置專案，且想要使用 JDBC 連線，請採取下節中的步驟。

將 JDBC 驅動程式設定為 Maven 相依性

1. 將 Amazon 儲存庫或 Maven Central 儲存庫新增到 pom.xml 檔案的儲存庫區段。

> **Note**
>
> 下列程式碼範例中的 URL 在瀏覽器中使用時會傳回錯誤。僅在 Maven 項目的內容中使用此 URL。

若要使用 Secure Sockets Layer (SSL) 來連線，請將下列儲存庫新增至 pom.xml 檔案。

```
<repositories>
    <repository>
      <id>redshift</id>
      <url>https://s3.amazonaws.com/redshift-maven-repository/release</url>
    </repository>
</repositories>
```

若為 Maven Central 儲存庫，請將以下內容新增到 pom.xml 檔案。

```
<repositories>
    <repository>
      <id>redshift</id>
      <url>https://repo1.maven.org/maven2</url>
    </repository>
</repositories>
```

2. 在 pom.xml 檔案的相依性區段中，宣告您要使用的驅動程式版本。

   Amazon Redshift 為與 JDBC 4.2 API 相容的工具提供了驅動程式。如需這些驅動程式所支援功能的相關資訊，請參閱下載 Amazon Redshift JDBC 驅動程式 2.x 版。

   以您的驅動程式版本取代下列範例中的 *driver-version*，例如 2.1.0.1。若為 JDBC 4.2 相容驅動程式，請使用以下內容。

   ```
   <dependency>
      <groupId>com.amazon.redshift</groupId>
      <artifactId>redshift-jdbc42</artifactId>
      <version>driver-version</version>
   </dependency>
   ```

   此驅動程式的類別名稱為 com.amazon.redshift.Driver。

當您使用 IAM 資料庫身分驗證時，Amazon Redshift Maven 驅動程式需要下列選擇性相依性。

```
<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>
```

若要將 Amazon Redshift JDBC 驅動程式升級或變更為最新版本，請先將相依性的版本區段修改為驅動程式的最新版本。然後以 Maven Clean Plugin 來清理專案，如下所示。

```
mvn clean
```

設定身分驗證和 SSL

為了防止有人未經授權就存取資料，Amazon Redshift 資料存放區會要求所有連線使用使用者憑證進行身分驗證。某些資料存放區還會要求必須透過 Secure Sockets Layer (SSL) 通訊協定來進行連線 (無論是否使用單向身分驗證)。

Amazon Redshift JDBC 驅動器 2.x 版會針對這些身分驗證通訊協定提供完整支援。

驅動程式支援的 SSL 版本取決於您使用的 JVM 版本。如需每個 Java 版本所支援之 SSL 版本的相關資訊，請參閱 Java Platform Group Product Management 部落格上的診斷 TLS、SSL 和 HTTPS。

用於連線的 SSL 版本是驅動程式和伺服器支援的最高版本，而這會在連線時確定。

將 Amazon Redshift JDBC 驅動器 2.x 版設定為根據您要連線之 Redshift 伺服器的安全需求驗證您的連線。

您一律必須提供 Redshift 使用者名稱和密碼來驗證連線。根據伺服器上是否啟用且需要 SSL，您可能還需要將驅動程式設定為透過 SSL 連接。或者，您也可以使用單向 SSL 身分驗證，以便用戶端 (驅動程式本身) 驗證伺服器的身分。

您要在連線 URL 中提供組態資訊給驅動程式。如需連線 URL 語法的相關資訊，請參閱建置連線 URL。

SSL 表示 TLS/SSL，兩者分別是 Transport Layer Security 和 Secure Sockets Layer。此驅動程式支援業界標準的 TLS/SSL 版本。

設定 IAM 身分驗證

如果您要使用 IAM 身分驗證連線至 Amazon Redshift 伺服器，請將以下屬性設定為資料來源連線字串的一部分。

如需 IAM 身分驗證的相關資訊，請參閱 Amazon Redshift 中的身分和存取管理。

若要使用 IAM 身分驗證，請使用下列其中一種連線字串格式：

連接字串	說明
`jdbc:redshift:iam:// [host]:[port]/[db]`	一般連線字串。驅動程式會從主機推斷 ClusterID 和 Region。
`jdbc:redshift:iam:// [cluster-id]: [region]/[db]`	驅動程式會擷取主機資訊 (在給定 ClusterID D 和 Region 的情況下)。
`jdbc:redshift:iam:// [host]/[db]`	驅動程式會預設為連接埠 5439，並從主機推斷 ClusterID 和 Region。根據您在建立、修改或遷移叢集時所選取的連接埠而定，允許存取選取的連接埠。

指定設定檔

如果您要使用 IAM 身分驗證，則可以在設定檔名稱下指定任何其他必要或選擇性的連線屬性。這樣做可讓您避免將某些資訊直接放在連線字串中。您可以使用 Profile 屬性在連線字串中指定設定檔名稱。

設定檔可以新增至 AWS 登入資料檔案。此檔案的預設位置是：`~/.aws/credentials`。

您可以在下列環境變數中設定路徑來變更預設值：`AWS_CREDENTIAL_PROFILES_FILE`

如需設定檔的相關資訊，請參閱《適用於 Java 的 AWS SDK》中的使用 AWS 憑證。

使用執行個體設定檔憑證

如果您要在與 IAM 角色關聯的 Amazon EC2 執行個體上執行應用程式，您可以使用執行個體設定檔憑證進行連線。

若要這麼做，請使用前述資料表中的其中一種 IAM 連線字串格式，並將 dbuser 連線屬性設定為您要用來連線的 Amazon Redshift 使用者名稱。

如需執行個體設定檔的相關資訊，請參閱《IAM 使用者指南》中的存取管理。

使用憑證提供者

此驅動程式也支援下列服務的憑證提供者外掛程式：

- AWS IAM Identity Center
- Active Directory Federation Service (ADFS)
- JSON Web 權杖 (JWT) 服務
- Microsoft Azure Active Directory (AD) 服務和瀏覽器 Microsoft Azure Active Directory (AD) 服務
- Okta 服務
- PingFederate 服務
- 適用於 SAML 服務的瀏覽器 SAML，例如 Okta、Ping 或 ADFS

如果您使用這些服務的其中一個，連線 URL 必須指定下列屬性：

- Plugin_Name — 憑證提供者外掛程式類別的完整類別路徑。
- IDP_Host — 您用來對 Amazon Redshift 進行身分驗證的服務主機。
- IdP_Port — 用於身分驗證服務的主機監聽的連接埠。Okta 不需要此屬性。
- User - idp_host 伺服器的使用者名稱。
- Password - 與 idp_host 使用者名稱相關聯的密碼。
- DbUser - 您用來連線的 Amazon Redshift 使用者名稱。
- SSL_Insecure — 指出是否應驗證 IDP 伺服器憑證。
- Client_ID - 與 Azure AD 入口網站中的使用者名稱相關聯的用戶端 ID。僅用於 Azure AD。

- Client_Secret — 與 Azure AD 入口網站中的用戶端 ID 相關聯的用戶端祕密。僅用於 Azure AD。
- IdP_Tenant — Amazon Redshift 應用程式的 Azure AD 租用戶 ID。僅用於 Azure AD。
- App_ID — Amazon Redshift 應用程式的 Okta 應用程式 ID。僅用於 Okta。
- App_Name — Amazon Redshift 應用程式的 Okta 應用程式名稱 (選擇性)。僅用於 Okta。
- Partner_SPID — 選擇性的合作夥伴 SPID (服務提供者 ID) 值。僅用於 PingFederate。
- Idc_Region – IAM Identity Center AWS 區域 AWS 執行個體所在的 。僅用於 AWS IAM Identity Center。
- Issuer_Url – AWS IAM Identity Center 伺服器的執行個體端點。僅用於 AWS IAM Identity Center。

如果您使用這些服務其中之一的瀏覽器外掛程式，連線 URL 也可以包括：

- Login_URL — 透過瀏覽器外掛程式使用安全性聲明標記語言 (SAML) 或 Azure AD 服務時，身分提供者網站上資源的 URL。如果您使用瀏覽器外掛程式，則必須要有此參數。

- Listen_Port – 透過瀏覽器外掛程式使用 SAML、Azure AD 或 IAM Identity Center AWS 服務時，驅動程式用來從身分提供者取得 SAML 回應的連接埠。

- IdP_Response_Timeout – 透過瀏覽器外掛程式使用 SAML、Azure AD 或 IAM Identity Center AWS 服務時，驅動程式等待身分提供者 SAML 回應的時間，以秒為單位。

如需其他連線字串屬性的資訊，請參閱 JDBC 驅動器 2.x 版組態的選項。

僅使用使用者名稱和密碼

如果您要連線的伺服器未使用 SSL，您只需要提供 Redshift 使用者名稱和密碼即可驗證連線。

僅使用 Redshift 使用者名稱和密碼來設定身分驗證

1. 將 UID 屬性設定為用於存取 Amazon Redshift 伺服器的 Redshift 使用者名稱。
2. 將 PWD 屬性設定為與 Redshift 使用者名稱對應的密碼。

使用 SSL 但不進行身分驗證

如果您要連線的伺服器使用 SSL，但不需要進行身分驗證，則您可以將驅動程式設定為使用非驗證 SSL Factory。

設定沒有身分驗證的 SSL 連線

1. 將 UID 屬性設定為用於存取 Amazon Redshift 伺服器的 Redshift 使用者名稱。
2. 將 PWD 屬性設定為與 Redshift 使用者名稱對應的密碼。
3. 將 SSLFactory 屬性設定為 com.amazon.redshift.ssl.NonValidatingFactory。

使用單向 SSL 身分驗證

如果您要連線的伺服器使用 SSL 且具有憑證，則您可以將驅動程式設定為使用單向身分驗證來驗證伺服器的身分。

單向身分驗證需要已簽署的受信任 SSL 憑證來驗證伺服器的身分。您可以將驅動程式設定為使用特定憑證，或存取包含適當憑證的 TrustStore。如果您未指定憑證或 TrustStore，則驅動程式會使用預設的 Java TrustStore (一般是 jssecacerts 或 cacerts)。

設定單向 SSL 身分驗證

1. 將 UID 屬性設定為用於存取 Amazon Redshift 伺服器的 Redshift 使用者名稱。
2. 將 PWD 屬性設定為與 Redshift 使用者名稱對應的密碼。
3. 將 SSL 屬性設為 true。
4. 將 SSLRootCert 屬性設定為根 CA 憑證的位置。
5. 如果您未使用其中一個預設的 Java TrustStore，請執行以下其中一項操作：
   - 若要指定伺服器憑證，請將 SSLRootCert 屬性設定為憑證的完整路徑。
   - 若要指定 TrustStore，請執行以下操作：
     a. 使用 keytool 程式，將伺服器憑證新增至您要使用的 TrustStore。
     b. 指定要在使用驅動程式啟動 Java 應用程式時使用的 TrustStore 和密碼。例如：

```
-Djavax.net.ssl.trustStore=[TrustStoreName]
-Djavax.net.ssl.trustStorePassword=[TrustStorePassword]
-Djavax.net.ssl.trustStoreType=[TrustStoreType]
```

6. 選擇一項：
   - 若要驗證憑證，請將 SSLMode 屬性設定為 verify-ca。
   - 若要驗證憑證並驗證憑證中的主機名稱，請將 SSLMode 屬性設定為 verify-full。

設定 記錄

您可以在驅動程式中開啟日誌記錄功能，以協助診斷問題。

您可以使用下列方法記錄驅動程式資訊：

- 若要將記錄的資訊儲存在 .log 檔案中，請參閱使用日誌檔案。
- 若要將記錄的資訊傳送到 DriverManager 中指定的 LogStream 或 LogWriter，請參閱使用 LogStream 或 LogWriter。

您要在連線 URL 中提供組態資訊給驅動程式。如需連線 URL 語法的相關資訊，請參閱建置連線 URL。

使用日誌檔案

日誌記錄功能應該要開啟到足以找到問題的時間長度就好。日誌記錄功能會降低效能，而且還會消耗大量磁碟空間。

在連線 URL 中設定 LogLevel 機碼以開啟日誌記錄功能，並指定日誌檔案中包含的詳細資訊數量。下表列出 Amazon Redshift JDBC 驅動器 2.x 版所提供的日誌記錄層級 (排序方式為最不詳細到最詳細)。

LogLevel 值	說明
1	記錄會導致驅動程式中止的嚴重錯誤事件。
2	記錄可能允許驅動程式繼續執行的錯誤事件。
3	未採取動作時可能會導致錯誤的日誌事件。這個層級的日誌記錄和這個層級以上的日誌記錄層級也會記錄使用者的查詢。
4	記錄用於描述驅動程式進展的一般資訊。
5	記錄有助於偵錯驅動程式的詳細資訊。
6	記錄所有驅動程式活動。

設定會使用日誌檔案的日誌記錄功能

1. 將 LogLevel 屬性設定為要包含在日誌檔案中的所需資訊層級。

2. 將 LogPath 屬性設定為用來儲存日誌檔案之資料夾的完整路徑。

   例如，下列連線 URL 會啟用日誌記錄層級 3，並將日誌檔案儲存在 C:\temp 資料夾：`jdbc:redshift://redshift.company.us-west-1.redshift.amazonaws.com:9000/Default;DSILogLevel=3;LogPath=C:\temp`

3. 若要確保新設定生效，請重新啟動 JDBC 應用程式並重新連線至伺服器。

   Amazon Redshift JDBC 驅動程式會在 LogPath 屬性中指定的位置產生下列日誌檔案：

   - redshift_jdbc.log 檔案，會記錄非連線特定的驅動程式活動。
   - redshift_jdbc_connection_[Number].log 檔案，會針對向資料庫建立的每個連線產生此檔案，其中 [Number] 是用來識別每個日誌檔案的編號。此檔案會記錄連線特定的驅動程式活動。

如果 LogPath 值無效，則驅動程式會將日誌記錄資訊傳送至標準輸出串流 (`System.out`)

使用 LogStream 或 LogWriter

日誌記錄功能應該要開啟到足以找到問題的時間長度就好。日誌記錄功能會降低效能，而且還會消耗大量磁碟空間。

在連線 URL 中設定 LogLevel 機碼以開啟日誌記錄功能，並指定傳送至 DriverManager 中所指定 LogStream 或 LogWriter 的詳細資訊數量。

若要開啟使用 LogStream 或 LogWriter 的日誌記錄功能：

1. 若要設定驅動程式以記錄描述驅動程式進度的一般資訊，請將 LogLevel 屬性設定為 1 或 INFO。
2. 若要確保新設定生效，請重新啟動 JDBC 應用程式並重新連線至伺服器。

資料類型轉換

Amazon Redshift JDBC 驅動器 2.x 版支援許多常見的資料格式，可在 Amazon Redshift、SQL 和 Java 資料類型之間進行轉換。

下表列出支援的資料類型映射。

Amazon Redshift 類型	SQL 類型	Java 類型
BIGINT	SQL_BIGINT	Long

Amazon Redshift 類型	SQL 類型	Java 類型
BOOLEAN	SQL_BIT	Boolean
CHAR	SQL_CHAR	String
DATE	SQL_TYPE_DATE	java.sql.Date
DECIMAL	SQL_NUMERIC	BigDecimal
DOUBLE PRECISION	SQL_DOUBLE	Double
GEOMETRY	SQL_ LONGVARBINARY	byte[]
INTEGER	SQL_INTEGER	Integer
OID	SQL_BIGINT	Long
SUPER	SQL_LONGVARCHAR	String
REAL	SQL_REAL	Float
SMALLINT	SQL_SMALLINT	Short
TEXT	SQL_VARCHAR	String
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	String

使用預備陳述式支援

Amazon Redshift JDBC 驅動程式支援預備陳述式。您可以使用預備陳述式來改善需要在相同連線期間執行多次的參數化查詢效能。

預備陳述式是在伺服器端編譯但不會立即執行的 SQL 陳述式。所編譯的陳述式會以 PreverredStatement 物件的形式儲存在伺服器上，直到您關閉物件或連線為止。當該物件存在時，您可以視需要使用不同的參數值執行預備陳述式多次，而不必重新編譯陳述式。由於額外負荷降低，因此該組查詢的執行速度會變快。

如需預備陳述式的相關資訊，請參閱《Oracle 提供的 JDBC 基本概念教學課程》https://docs.oracle.com/javase/tutorial/jdbc/basics/prepared.html中的 < 使用預備陳述式 > 。

您可以準備包含多個查詢的陳述式。例如，下列預備陳述式包含兩個 INSERT 查詢：

```
PreparedStatement pstmt = conn.prepareStatement("INSERT INTO
MyTable VALUES (1, 'abc'); INSERT INTO CompanyTable VALUES
(1, 'abc');");
```

請注意，這些查詢不會依賴相同預備陳述式中所指定其他查詢的結果。由於準備步驟期間不會執行查詢，因此系統尚未傳回結果，因此沒有結果可供相同預備陳述式中的其他查詢使用。

例如，系統不會允許下列會先建立資料表，然後將值插入該新建資料表的預備陳述式：

```
PreparedStatement pstmt = conn.prepareStatement("CREATE
TABLE MyTable(col1 int, col2 varchar); INSERT INTO myTable
VALUES (1, 'abc');");
```

如果您嘗試準備這個陳述式，伺服器會傳回錯誤，指出目的地資料表 (myTable) 尚不存在。必須先執行 CREATE 查詢，才可以準備 INSERT 查詢。

JDBC 驅動程式的 2.x 和 1.x 版本之間的差異

本節說明 2.x 和 1.x 版 JDBC 驅動程式傳回的資訊差異。JDBC 驅動程式 1.x 版已停止使用。

下表列出每個 JDBC 驅動程式版本的 getDatabaseProductName() 和 getDatabaseProductVersion() 函數所傳回的 DatabaseMetadata 資訊。JDBC 驅動程式 2.x 版會在建立連線時取得值。JDBC 驅動程式 1.x 版會從查詢結果取得值。

JDBC 驅動程式版本	getDatabaseProductName() 結果	getDatabaseProductVersion() 結果
2.x	Redshift	8.0.2

JDBC 驅動程式版本	getDatabaseProductName() 結果	getDatabaseProductVersion() 結果
1.x	PostgreSQL	08.00.0002

下表列出每個 JDBC 驅動程式版本的 getTypeInfo 函數所傳回的 DatabaseMetadata 資訊。

JDBC 驅動程式版本	getTypeInfo 結果
2.x	與 Redshift 資料類型一致
1.x	與 PostgreSQL 資料類型一致

建立 JDBC 驅動器 2.x 版的初始化 (.ini) 檔案

透過使用 Amazon Redshift JDBC 驅動器 2.x 版的初始化 (.ini) 檔案，您可以指定系統層級的組態參數。例如，每個應用程式的聯合 IdP 身分驗證參數可能會有所不同。.ini 檔案會提供通用位置讓 SQL 用戶端取得所需的組態參數。

您可以建立包含 SQL 用戶端組態選項的 JDBC 驅動器 2.x 版初始化 (.ini) 檔案。檔案的預設名稱為 `rsjdbc.ini`。JDBC 驅動器 2.x 版會在下列位置 (依優先順序列出) 檢查 .ini 檔案：

- 連線 URL 或 SQL 用戶端的連線屬性對話方塊中的 `IniFile` 參數。請確定 `IniFile` 參數包含 .ini 檔案的完整路徑 (包括檔案名稱)。如需 `IniFile` 參數的詳細資訊，請參閱 IniFile。如果 `IniFile` 參數未正確指定 .ini 檔案的位置，則會顯示錯誤。
- 具有完整路徑 (包括檔案名稱在內) 的環境變數，例如 AMAZON_REDSHIFT_JDBC_INI_FILE。您可以使用 `rsjdbc.ini` 或指定檔案名稱。如果 AMAZON_REDSHIFT_JDBC_INI_FILE 環境變數未正確指定 .ini 檔案的位置，則會顯示錯誤。
- 驅動程式 JAR 檔案所在的目錄。
- 使用者主目錄。
- 系統的臨時目錄。

您可以將 .ini 檔案組織為多個驅動，例如 [DRIVER]。每個區段包含指定各種連線參數的鍵值對。您可以使用 `IniSection` 參數來指定 .ini 檔案中的區段。如需 `IniSection` 參數的詳細資訊，請參閱 IniSection。

以下是 .ini 檔案格式的範例，其中包含 [DRIVER]、[DEV]、[QA] 和 [PROD] 區段。[DRIVER] 區段可套用至任何連線。

```
[DRIVER]
key1=val1
key2=val2

[DEV]
key1=val1
key2=val2

[QA]
key1=val1
key2=val2

[PROD]
key1=val1
key2=val2
```

JDBC 驅動器 2.x 版會從下列位置 (依優先順序列出) 載入組態參數：

- 應用程式程式碼中的預設組態參數。
- .ini 檔案中的 [DRIVER] 區段屬性 (如果包含的話)。
- 自訂區段組態參數 (如果有在連線 URL 或 SQL 用戶端的連線屬性對話方塊中提供 `IniSection` 選項的話)。
- 來自 `getConnection` 呼叫中所指定連線屬性物件的屬性。
- 連線 URL 中所指定的組態參數。

JDBC 驅動器 2.x 版組態的選項

您可以在下面找到您可以為 Amazon Redshift JDBC 驅動程式 2.x 版指定的選項說明。組態選項不區分大小寫。

您可以使用連線 URL 設定組態屬性。如需詳細資訊，請參閱建置連線 URL。

主題

- AccessKeyID
- AllowDBUserOverride

- App_ID
- App_Name
- ApplicationName
- AuthProfile
- AutoCreate
- Client_ID
- Client_Secret
- ClusterID (ClusterID)
- 壓縮
- connectTimeout
- connectionTimezone
- databaseMetadataCurrentDbOnly
- DbUser
- DbGroups
- DBNAME
- defaultRowFetchSize
- DisableIsValidQuery
- enableFetchRingBuffer
- enableMultiSqlSupport
- EnableTableTypes
- fetchRingBufferSize
- ForceLowercase
- 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
- 密碼
- Plugin_Name
- PORT
- preferPQ
- Preferred_Role
- 設定檔
- PWD
- queryGroup
- readOnly
- 區域
- reWriteBatchedInserts

- reWriteBatchedInsertsSize
- roleArn
- roleSessionName
- scope
- SecretAccessKey
- SessionToken
- serverlessAcctId
- serverlessWorkGroup
- socketFactory
- socketTimeout
- SSL
- SSL_Insecure
- SSLCert
- SSLFactory
- SSLKey
- SSLMode
- SSLPassword
- SSLRootCert
- StsEndpointUrl
- tcpKeepAlive
- token
- token_type
- UID
- 使用者
- webIdentityToken

AccessKeyID

- **預設值 — 無**

- 資料類型 — 字串

您可以指定此參數以輸入使用者或角色的 IAM 存取金鑰。您通常可以透過查看和現有字串或使用者設定檔來找到金鑰。如果您指定此參數，則還必須指定 SecretAccessKey 參數。如果在 JDBC URL 中傳遞，則 AccessKeyID 必須經過 URL 編碼。

此為選用參數。

AllowDBUserOverride

- 預設值 — 0
- 資料類型 — 字串

此選項會指定驅動程式是使用 SAML 聲明中的 DbUser 值，還是使用連線 URL 中 DbUser 連線屬性所指定的值。

此為選用參數。

1

驅動程式會使用 SAML 聲明中的 DbUser 值。

如果 SAML 聲明未針對 DBUser 指定值，則驅動程式會使用 DBUser 連線屬性中指定的值。如果連線屬性也未指定值，則驅動程式會使用連線設定檔中指定的值。

0

驅動程式會使用 DBUser 連線屬性中指定的 DBUser 值。

如果 DBUser 連線屬性未指定值，則驅動程式會使用連線設定檔中指定的值。如果連線設定檔也未指定值，則驅動程式會使用 SAML 聲明中的值。

App_ID

- 預設值 — 無
- 資料類型 — 字串

與您的 Amazon Redshift 應用程式相關聯、Okta 所提供的唯一 ID。

如果透過 Okta 服務進行驗證，則需要此參數。

App_Name

- 預設值 — 無
- 資料類型 — 字串

您用來驗證與 Amazon Redshift 之連線的 Okta 應用程式名稱。

此為選用參數。

ApplicationName

- 預設值 — Null
- 資料類型 — 字串

要傳遞給 Amazon Redshift 以進行稽核的應用程式名稱。

此為選用參數。

AuthProfile

- 預設值 — 無
- 資料類型 — 字串

用於連線至 Amazon Redshift 的身分驗證設定檔名稱。

此為選用參數。

AutoCreate

- 預設值 — false
- 資料類型 — 布林值

此選項會指定當指定的使用者不存在時，驅動程式是否要導致系統建立新的使用者。

此為選用參數。

true

如果 DBUser 或唯一 ID (UID) 指定的使用者不存在，則系統會建立具有該名稱的新使用者。

false

驅動程式不會導致系統建立新的使用者。如果指定的使用者不存在，則身分驗證會失敗。

Client_ID

- 預設值 — 無
- 資料類型 — 字串

使用 Azure AD 服務來驗證連線時要使用的用戶端 ID。

如果透過 Azure AD 服務進行驗證，則需要此參數。

Client_Secret

- 預設值 — 無
- 資料類型 — 字串

使用 Azure AD 服務來驗證連線時要使用的用戶端秘密。

如果透過 Azure AD 服務進行驗證，則需要此參數。

ClusterID (ClusterID)

- 預設值 — 無
- 資料類型 — 字串

您要連線至的 Amazon Redshift 叢集的名稱。驅動程式會嘗試從給定的主機偵測此參數。如果您使用 Network Load Balancer (NLB) 並透過 IAM 進行連線，則驅動程式無法偵測此參數，因此您可以使用此連線選項設定此參數。

此為選用參數。

壓縮

- 預設值 - off
- 資料類型 — 字串

用於 Amazon Redshift 伺服器與用戶端或驅動程式之間的有線通訊協定通訊的壓縮方法。

此為選用參數。

您可以指定下列值：

- lz4

  將用於與 Amazon Redshift 進行有線通訊協定通訊的壓縮方法設定為 lz4。
- off

  不要對與 Amazon Redshift 進行的有線通訊協定通訊使用壓縮。

connectTimeout

- 預設值 — 10
- 資料類型 — 整數

用於通訊端連線操作的逾時值。如果建立 Amazon Redshift 連線所需的時間超過此值，系統就會將此連線視為無法使用。逾時會以秒為單位來指定。值為 0 表示未指定逾時。

此為選用參數。

connectionTimezone

- 預設值 — LOCAL
- 資料類型 — 字串

工作階段層級時區。

此為選用參數。

您可以指定下列值：

LOCAL

將工作階段層級時區設定為 LOCAL JVM 時區。

SERVER

將工作階段層級時區設定為在 Amazon Redshift 伺服器上為使用者設定的時區。您可以使用以下命令為使用者設定工作階段層級時區：

```
ALTER USER
[...]
SET TIMEZONE TO [...];
```

databaseMetadataCurrentDbOnly

- 預設值 — true
- 資料類型 — 布林值

此選項指定中繼資料 API 會從所有可存取的資料庫擷取資料，還是僅從已連線的資料庫擷取資料。

此為選用參數。

您可以指定下列值：

true

應用程式會從單一資料庫擷取中繼資料。

false

應用程式會從所有可存取的資料庫擷取中繼資料。

DbUser

- 預設值 — 無
- 資料類型 — 字串

要與您的 Amazon Redshift 帳戶搭配使用的使用者 ID。如果您已啟用 AutoCreate 屬性，則可以使用目前不存在的 ID。

此為選用參數。

DbGroups

- 預設值 — PUBLIC
- 資料類型 — 字串

DBUser 針對目前的工作階段所加入之現有資料庫群組名稱的逗號分隔清單。

此為選用參數。

DBNAME

- 預設值 — Null
- 資料類型 — 字串

要連線到之資料庫的名稱。您可以使用此選項在 JDBC 連線 URL 中指定資料庫名稱。

此為必要參數。您必須在連線 URL 或用戶端應用程式的連線屬性中指定資料庫名稱。

defaultRowFetchSize

- 預設值 — 0
- 資料類型 — 整數

此選項會指定 getFetchSize 的預設值。

此為選用參數。

您可以指定下列值：

0

在單一操作中擷取所有資料列。

正整數

針對 ResultSet 的每次擷取迭代，要從資料庫擷取的資料列數目。

DisableIsValidQuery

- 預設值 — False
- 資料類型 — 布林值

此選項會指定驅動程式在使用 Connection.isValid() 方法判斷資料庫連線是否處於作用中狀態時，是否要提交新的資料庫查詢。

此為選用參數。

true

驅動程式在使用 Connection.isValid() 判斷資料庫連線是否處於作用中狀態時，不會提交查詢。如果資料庫伺服器意外關閉，這可能會造成驅動程式錯誤地將資料庫連線識別為作用中狀態。

false

驅動程式在使用 Connection.isValid() 判斷資料庫連線是否處於作用中狀態時，會提交查詢。

enableFetchRingBuffer

- 預設值 — true
- 資料類型 — 布林值

此選項指定驅動程式會在個別執行緒上使用環形緩衝器來擷取資料列。fetchRingBufferSize 參數會指定環形緩衝區的大小。

環形緩衝區會在 JDBC 中實作自動記憶體管理，以防止在資料擷取操作期間out-of-memory(OOM) 錯誤。環形緩衝區會即時監控緩衝資料的實際大小，確保驅動程式的總記憶體用量保持在定義的限制內。達到緩衝容量時，驅動程式會暫停資料擷取操作，防止記憶體溢出，而不需要手動介入。此內建防護功能可自動消除 OOM 錯誤，無需使用者設定。

如果交易偵測到包含多個以分號分隔的 SQL 命令的陳述式，該交易的擷取環形緩衝區會設為 false。enableFetchRingBuffer 的值不會變更。

此為選用參數。

> **Note**
>
> 當環緩衝區停用且擷取大小未正確設定時，可能會發生out-of-memory(OOM) 問題。如需設定擷取大小的詳細資訊，請參閱此處。

enableMultiSqlSupport

- 預設值 — true
- 資料類型 — 布林值

此選項會指定是否要在陳述式中處理以分號分隔的多個 SQL 命令。

此為選用參數。

您可以指定下列值：

true

驅動程式會在陳述式物件中處理以分號分隔的多個 SQL 命令。

false

驅動程式會在單一陳述式中傳回多個 SQL 命令的錯誤。

EnableTableTypes

- 預設值 — true
- 資料類型 — 布林值

此選項指定驅動程式是否在 getTables和 getTableTypesDatabaseMetaData方法的結果中辨識來自資料來源的詳細資料表類型資訊。根據預設，驅動程式會辨識詳細的資料表類型。

此為選用參數。它適用於驅動程式 2.2.8 版和更新版本。

true

驅動程式會辨識下列資料表類型：TABLE、VIEW、 SYSTEM TABLE、 SYSTEM VIEW、EXTERNAL TABLE 和 LOCAL TEMPORARY。

false

驅動程式會將詳細資料表類型資訊標準化為一般 TABLE 和 VIEW 資料表類型。

fetchRingBufferSize

- 預設值 — 1G
- 資料類型 — 字串

此選項會指定在擷取結果集時使用的環形緩衝區大小。您可以指定以位元組為單位的大小，例如 1K 表示 1 KB、5000 表示 5,000 個位元組、1M 表示 1 MB、1G 表示 1 GB，依此類推。您也可以指定堆積記憶體的百分比。驅動程式會在達到限制時停止擷取資料列。當應用程式讀取資料列並釋放環形緩衝區中的空間時，擷取操作便會繼續。

此為選用參數。

ForceLowercase

- 預設值 — false
- 資料類型 — 布林值

此選項會指定在使用單一登入身分驗證時，驅動程式是否會將從身分提供者傳送至 Amazon Redshift 的所有資料庫群組 (DbGroups) 轉換為小寫。

此為選用參數。

true

驅動程式會將從身分提供者傳送的所有資料庫群組轉換為小寫。

false

驅動程式不會變更資料庫群組。

groupFederation

- 預設值 — false
- 資料類型 — 布林值

此選項會指定是否使用 Amazon Redshift IDP 群組。這會由 GetClusterCredentialsV2 API 支援。

此為選用參數。

true

使用 Amazon Redshift 身分提供者 (IdP) 群組。

false

針對使用者聯合使用 STS API 和 GetClusterCredentials，並明確指定連線的 DbGroups。

HOST

- 預設值 — Null

- 資料類型 — 字串

要連線到之 Amazon Redshift 伺服器的主機名稱。您可以使用此選項在 JDBC 連線 URL 中指定主機名稱。

此為必要參數。您必須在連線 URL 或用戶端應用程式的連線屬性中指定主機名稱。

IAMDisableCache

- 預設值 — false
- 資料類型 — 布林值

此選項會指定是否快取 IAM 憑證。

此為選用參數。

true

不快取 IAM 憑證。

false

快取 IAM 憑證。例如，當對於 API 閘道的請求遭到限流時，這可以提高效能。

IAMDuration

- 預設值 — 900
- 資料類型 — 整數

臨時 IAM 憑證到期之前的時間長度 (以秒為單位)。

- 最小值 — 900
- 最大值 — 3,600

此為選用參數。

Idc_Client_Display_Name

- 預設值 - Amazon Redshift JDBC 驅動器

- 資料類型 — 字串

使用 BrowserIdcAuthPlugin 的用戶端要使用的顯示名稱。

此為選用參數。

Idc_Region

- 預設值 — 無
- 資料類型 — 字串

IAM Identity Center 執行個體所在的 AWS 區域。

只有在 plugin_name 組態選項中使用 `BrowserIdcAuthPlugin` 進行身分驗證時，才需要此參數。

IdP_Host

- 預設值 — 無
- 資料類型 — 字串

您用來對 Amazon Redshift 進行身分驗證的 IdP (身分提供者) 主機。這可以在連線字串或設定檔中指定。

此為選用參數。

IdP_Partition

- 預設值 — 無
- 資料類型 — 字串

指定設定身分提供者 (IdP) 的雲端分割區。這會決定驅動程式連線到哪個 IdP 身分驗證端點。

如果此參數保留空白，驅動程式會預設為商業分割區。可能值為：

- `us-gov`：如果您的 IdP 是在 Azure Government 中設定，請使用此值。例如，Azure AD Government 會使用端點 `login.microsoftonline.us`。
- `cn`：如果您的 IdP 是在中國雲端分割區中設定，請使用此值。例如，Azure AD 中國使用端點 `login.chinacloudapi.cn`。

此為選用參數。

IdP_Port

- 預設值 — 無
- 資料類型 — 字串

IdP (身分提供者) 使用的連接埠。您可以在連線字串或設定檔中指定連接埠。預設連接埠為 5439。根據您在建立、修改或遷移叢集時所選取的連接埠而定，允許存取選取的連接埠。

此為選用參數。

IdP_Tenant

- 預設值 — 無
- 資料類型 — 字串

Amazon Redshift 應用程式的 Azure AD 租用戶 ID。

如果透過 Azure AD 服務進行驗證，則需要此參數。

IdP_Response_Timeout

- 預設值 — 120
- 資料類型 — 整數

透過瀏覽器外掛程式使用 SAML 或 Azure AD 服務時，驅動程式等待身分提供者傳回 SAML 回應的時間長度 (以秒為單位)。

此為選用參數。

IniFile

- 預設值 — 無
- 資料類型 — 字串

.ini 檔案的完整路徑 (包括檔案名稱)。例如：

```
IniFile="C:\tools\rsjdbc.ini"
```

如需 .ini 檔案的相關資訊，請參閱建立 JDBC 驅動器 2.x 版的初始化 (.ini) 檔案。

此為選用參數。

IniSection

- 預設值 — 無
- 資料類型 — 字串

.ini 檔案中包含組態選項之區段的名稱。如需 .ini 檔案的相關資訊，請參閱建立 JDBC 驅動器 2.x 版的初始化 (.ini) 檔案。

下列範例會指定 .ini 檔案的 [Prod] 區段：

```
IniSection="Prod"
```

此為選用參數。

isServerless

- 預設值 — false
- 資料類型 — 布林值

此選項會指定 Amazon Redshift 端點主機是否為無伺服器執行個體。驅動程式會嘗試從給定的主機偵測此參數。如果您使用 Network Load Balancer (NLB)，則驅動程式無法偵測此參數，因此您可以在此設定此參數。

此為選用參數。

true

Amazon Redshift 端點主機是無伺服器執行個體。

false

Amazon Redshift 端點主機是已佈建的叢集。

Issuer_Url

- 預設值 — 無

- 資料類型 — 字串

指向 AWS IAM Identity Center 伺服器的執行個體端點。

只有在 plugin_name 組態選項中使用 `BrowserIdcAuthPlugin` 進行身分驗證時，才需要此參數。

Listen_Port

- 預設值 — 7890
- 資料類型 — 整數

透過瀏覽器外掛程式使用 SAML、Azure AD 或 AWS Identity Center 服務時，驅動程式用來從身分提供者或授權碼接收 SAML 回應的連接埠。

此為選用參數。

Login_URL

- 預設值 — 無
- 資料類型 — 字串

透過瀏覽器外掛程式使用 SAML 或 Azure AD 服務時，身分提供者網站上資源的 URL。

如果透過瀏覽器外掛程式向 SAML 或 Azure AD 服務進行驗證，則需要此參數。

loginTimeout

- 預設值 — 0
- 資料類型 — 整數

連線至伺服器並進行驗證時，逾時之前等待的秒數。如果建立連線所花的時間超過此臨界值，連線會中止。

當此屬性設為 0 時，連線不會逾時。

此為選用參數。

loginToRp

- 預設值 — `urn:amazon:webservices`

- 資料類型 — 字串

您要用於 AD FS 身分驗證類型的依賴方信任。

此為選用參數。

LogLevel

- 預設值 — 0
- 資料類型 — 整數

使用此屬性可在驅動程式中開啟或關閉日誌記錄，以及指定日誌檔案中包含的詳細資訊數量。

日誌記錄功能應該要啟用到足以找到問題的時間長度就好。日誌記錄功能會降低效能，而且還會消耗大量磁碟空間。

此為選用參數。

將此參數設為以下其中一個值：

0

停用所有日誌記錄功能。

1

在 FATAL 層級上啟用日誌記錄功能，這會記錄會導致驅動程式中止的極嚴重錯誤事件。

2

在 ERROR 層級上啟用日誌記錄功能，這會記錄可能仍允許驅動程式繼續執行的錯誤事件。

3

在 WARNING 層級上啟用日誌記錄功能，這會記錄如果未採取動作可能會導致錯誤的事件。

4

在 INFO 層級上啟用日誌記錄功能，這會記錄描述驅動程式進度的一般資訊。

5

在 DEBUG 層級上啟用日誌記錄功能，這會記錄有助於偵錯驅動程式的詳細資訊。

6

在 TRACE 層級上啟用日誌記錄功能，這會記錄所有驅動程式活動。

在啟用日誌記錄功能時，驅動程式會在 LogPath 屬性所指定的位置產生下列日誌檔案：

- **redshift_jdbc.log** - 記錄非連線特定驅動器活動的檔案。
- **redshift_jdbc_connection_[Number].log** — 向資料庫建立的每個連線的檔案，其中 [Number] 是用來區別每個日誌檔案的編號。此檔案會記錄連線特定的驅動程式活動。

如果 LogPath 值無效，則驅動程式會將日誌記錄資訊傳送至標準輸出串流 System.out。

LogPath

- 預設值 — 目前的工作目錄。
- 資料類型 — 字串

啟用 DSILogLevel 屬性時，驅動程式用來儲存日誌檔案之資料夾的完整路徑。

為了確定連線 URL 會與所有 JDBC 應用程式相容，建議您輸入另一個反斜線來逸出檔案路徑中的反斜線 (\)。

此為選用參數。

OverrideSchemaPatternType

- 預設值 — Null
- 資料類型 — 整數

此選項會指定是否要覆寫 getTables 呼叫中使用的查詢類型。

0

無結構描述通用查詢

1

本機結構描述查詢

2

外部結構描述查詢

此為選用參數。

Partner_SPID

- 預設值 — 無
- 資料類型 — 字串

使用 PingFederate 服務驗證連線時，要使用的合作夥伴 SPID (服務提供者 ID) 值。

此為選用參數。

密碼

- 預設值 — 無
- 資料類型 — 字串

透過 IDP 使用 IAM 身分驗證進行連線時，這是 IDP_Host 伺服器的密碼。使用標準身分驗證時，這可以用於 Amazon Redshift 資料庫密碼，而不是 PWD。

此為選用參數。

Plugin_Name

- 預設值 — 無
- 資料類型 — 字串

用來實作特定憑證提供者外掛程式的完整類別名稱。

此為選用參數。

支援的提供者選項如下：

- **`AdfsCredentialsProvider`** - Active Directory Federation Service。
- **`AzureCredentialsProvider`** - Microsoft Azure Active Directory (AD) 服務。

- **BasicJwtCredentialsProvider** - JSON Web Tokens (JWT) 服務。
- **BasicSamlCredentialsProvider** — 可與許多 SAML 服務提供者搭配使用的安全性聲明標記語言 (SAML) 憑證。
- **BrowserAzureCredentialsProvider** - 瀏覽器 Microsoft Azure Active Directory (AD) 服務。
- **BrowserAzureOAuth2CredentialsProvider** - 適用於原生身分驗證的瀏覽器 Microsoft Azure Active Directory (AD) 服務。
- **BrowserIdcAuthPlugin** – 使用 IAM Identity Center AWS 的授權外掛程式。
- **BrowserSamlCredentialsProvider** - 適用於 SAML 服務的瀏覽器 SAML，例如 Okta、Ping 或 ADFS。
- **IdpTokenAuthPlugin** – 從連結至 IAM Identity Center AWS 的任何 Web 身分提供者接受 IAM Identity Center 字符或 OpenID Connect (OIDC) JSON 型身分字符 (JWT) AWS 的授權外掛程式。
- **OktaCredentialsProvider** - Okta 服務。
- **PingCredentialsProvider** - PingFederate 服務。

PORT

- 預設值 — Null
- 資料類型 — 整數

要連線到之 Amazon Redshift 伺服器的連接埠。您可以使用此選項在 JDBC 連線 URL 中指定連接埠。

此為選用參數。

preferPQ

- 預設值 — true
- 資料類型 — 布林值

此選項控制驅動程式是否在 TLS 1.3 ClientHello 中公告量子後混合式金鑰交換群組。提供後量子群組且伺服器也支援後量子群組時，伺服器會選取一個。否則，交握會透明地回復到傳統金鑰交換。

此為選用參數。它適用於驅動程式 2.2.8 版和更新版本。

true

當執行時間支援時，驅動程式會公告後量子混合 TLS 金鑰交換群組。

false

驅動程式不會公告後量子群組。如果網路中間設備因為無法辨識驅動程式提供的後量子金鑰交換群組，而封鎖 TLS 交握，請使用此設定。

Preferred_Role

- 預設值 — 無
- 資料類型 — 字串

您要在連線到 Amazon Redshift 期間擔任的 IAM 角色。

此為選用參數。

設定檔

- 預設值 — 無
- 資料類型 — 字串

要用於 IAM 身分驗證之設定檔的名稱。此設定檔包含未在連線字串中指定的任何其他連線屬性。

此為選用參數。

PWD

- 預設值 — 無
- 資料類型 — 字串

與您使用屬性 UID 提供的 Amazon Redshift 使用者名稱對應的密碼。

此為選用參數。

queryGroup

- 預設值 — Null
- 資料類型 — 字串

此選項會在執行期將查詢指派給適當的查詢群組，以便將查詢指派給佇列。系統會針對工作階段設定查詢群組。在連線上執行的所有查詢都屬於此查詢群組。

此為選用參數。

readOnly

- 預設值 — false
- 資料類型 — 布林值

此屬性會指定驅動程式是否處於唯讀模式。

此為選用參數。

true

連線處於唯讀模式，且無法寫入到資料存放區。

false

連線不處於唯讀模式，可以寫入到資料存放區。

區域

- 預設值 — Null
- 資料類型 — 字串

此選項指定叢集所在的 AWS 區域。如果您指定 StsEndPoint 選項，系統會忽略區域選項。Redshift `GetClusterCredentials` API 操作也會使用區域選項。

此為選用參數。

reWriteBatchedInserts

- 預設值 — false
- 資料類型 — 布林值

此選項可實現最佳化，以重寫相容的 INSERT 陳述式並將這些陳述式合併為批次。

此為選用參數。

reWriteBatchedInsertsSize

- 預設值 — 128

- 資料類型 — 整數

此選項可實現最佳化，以重寫相容的 INSERT 陳述式並將這些陳述式合併為批次。此值必須以 2 的冪呈指數增加。

此為選用參數。

roleArn

- 預設值 — 無
- 資料類型 — 字串

角色的 Amazon Resource Name (ARN)。當您指定 Plugin_Name 選項的 BasicJwtCredentialsProvider 時，請務必指定此參數。請以下列格式指定 ARN：

`arn:`*`partition`*`:`*`service`*`:`*`region`*`:`*`account-id`*`:`*`resource-id`*

如果您指定 Plugin_Name 選項的 BasicJwtCredentialsProvider，則必須有此參數。

roleSessionName

- 預設值 — jwt_redshift_session
- 資料類型 — 字串

擔任角色工作階段的識別碼。一般來說，您會傳遞與應用程式使用者相關聯的名稱或識別碼。應用程式使用的臨時安全憑證會與該使用者相關聯。當您指定 Plugin_Name 選項的 BasicJwtCredentialsProvider 時，您可以指定此參數。

此為選用參數。

scope

- 預設值 — 無
- 資料類型 — 字串

使用者可以同意的範圍清單 (以空格分隔)。指定這個參數可讓您的 Microsoft Azure 應用程式針對您想要呼叫的 API 取得同意。當您為 Plugin_Name 選項指定 BrowserAzureOAuth2CredentialsProvider 時，便可以指定此參數。

BrowserAzureOAuth2CredentialsProvider 外掛程式需要此參數。

SecretAccessKey

- 預設值 — 無
- 資料類型 — 字串

使用者或角色的 IAM 存取金鑰。如果指定了此參數，則也必須指定 AccessKeyID。如果在 JDBC URL 中傳遞，則 SecretAccessKey 必須經過 URL 編碼。

此為選用參數。

SessionToken

- 預設值 — 無
- 資料類型 — 字串

與您用來驗證的 IAM 角色相關聯的臨時 IAM 工作階段權杖。如果在 JDBC URL 中傳遞，則臨時 IAM 工作階段權杖必須經過 URL 編碼。

此為選用參數。

serverlessAcctId

- 預設值 — Null
- 資料類型 — 字串

Amazon Redshift Serverless 帳戶 ID。驅動程式會嘗試從給定的主機偵測此參數。如果您使用 Network Load Balancer (NLB)，則驅動程式無法偵測此參數，因此您可以在此設定此參數。

此為選用參數。

serverlessWorkGroup

- 預設值 — Null
- 資料類型 — 字串

Amazon Redshift Serverless 工作群組名稱。驅動程式會嘗試從給定的主機偵測此參數。如果您使用 Network Load Balancer (NLB)，則驅動程式無法偵測此參數，因此您可以在此設定此參數。

此為選用參數。

socketFactory

- 預設值 — Null
- 資料類型 — 字串

此選項會指定用於建立通訊端的通訊端 Factory。

此為選用參數。

socketTimeout

- 預設值 — 0
- 資料類型 — 整數

在通訊端讀取操作期間，逾時之前等待的秒數。如果操作所花的時間超過此臨界值，連線會關閉。當此屬性設為 0 時，連線不會逾時。

此為選用參數。

SSL

- 預設值 — TRUE
- 資料類型 — 字串

使用此屬性可開啟或關閉連線的 SSL。

此為選用參數。

您可以指定下列值：

TRUE

驅動程式會透過 SSL 連線到伺服器。

FALSE

驅動程式會以不使用 SSL 的方式連線到伺服器。IAM 身分驗證不支援此選項。

或者，您也可以設定 AuthMech 屬性。

SSL_Insecure

- 預設值 — true
- 資料類型 — 字串

此屬性會指出是否應驗證 IDP 主機伺服器憑證。

此為選用參數。

您可以指定下列值：

true

驅動程式不會檢查 IDP 伺服器憑證的真實性。

false

驅動程式會檢查 IDP 伺服器憑證的真實性。

SSLCert

- 預設值 — 無
- 資料類型 — 字串

包含可在使用 SSL 時驗證 Amazon Redshift 伺服器執行個體之其他受信任 CA 憑證的 .pem 或 .crt 檔案的完整路徑。

如果指定 SSLKey，則需要此參數。

SSLFactory

- 預設值 — 無
- 資料類型 — 字串

在不使用伺服器憑證的情況下，透過 TLS/SSL 連線至伺服器時要使用的 SSL Factory。

SSLKey

- 預設值 — 無

- 資料類型 — 字串

包含用於驗證 SSLCert 中所指定憑證的 PKCS8 金鑰檔案的 .der 檔案完整路徑。

如果指定 SSLCert，則需要此參數。

SSLMode

- 預設值 — verify-ca
- 資料類型 — 字串

使用此屬性可指定在已啟用 TLS/SSL 時，驅動程式會如何驗證憑證。

此為選用參數。

您可以指定下列值：

verify-ca

驅動程式會驗證憑證是否來自受信任的憑證授權機構 (CA)。

verify-full

驅動程式會驗證憑證是否來自受信任的 CA，以及憑證中的主機名稱是否符合連線 URL 中指定的主機名稱。

> **Note**
>
> 我們建議您盡可能使用 `verify-full`進行 SSL 連線。

SSLPassword

- 預設值 — 0
- 資料類型 — 字串

SSLKey 中所指定已加密金鑰檔案的密碼。

如果指定了 SSLKey 且金鑰檔案已加密，則需要此參數。

SSLRootCert

- 預設值 — 無
- 資料類型 — 字串

包含可在使用 SSL 時驗證 Amazon Redshift 伺服器執行個體之根 CA 憑證的 .pem 或 .crt 檔案的完整路徑。

StsEndpointUrl

- 預設值 — Null
- 資料類型 — 字串

您可以指定 AWS Security Token Service (AWS STS) 端點。如果您指定此選項，系統會忽略區域選項。您只能為此端點指定安全的通訊協定 (HTTPS)。

tcpKeepAlive

- 預設值 — TRUE
- 資料類型 — 字串

使用此屬性可開啟或關閉 TCP 保持連線。

此為選用參數。

您可以指定下列值：

TRUE

驅動程式會使用 TCP 保持連線來防止連線逾時。

FALSE

驅動程式不會使用 TCP 保持連線。

token

- 預設值 — 無
- 資料類型 — 字串

IAM Identity Center AWS 提供存取字符或與 IAM Identity Center 連結的 Web 身分提供者提供的 OpenID Connect (OIDC) JSON Web Token AWS (JWT)。您的應用程式必須透過使用 IAM Identity Center AWS 或與 IAM Identity Center 連結的身分提供者來驗證應用程式的使用者來產生此字符 AWS 。

此參數適用於 `IdpTokenAuthPlugin`。

token_type

- 預設值 — 無
- 資料類型 — 字串

`IdpTokenAuthPlugin` 中使用的權仗類型。

您可以指定下列值：

ACCESS_TOKEN

如果您使用 AWS IAM Identity Center 提供的存取權杖，請輸入此選項。

EXT_JWT

如果您使用與 AWS IAM Identity Center 整合的 Web 型身分提供者所提供的 OpenID Connect (OIDC) JSON Web Token (JWT)，請輸入此項。

此參數適用於 `IdpTokenAuthPlugin`。

UID

- 預設值 — 無
- 資料類型 — 字串

您用來存取資料庫的資料庫使用者名稱。

此為必要參數。

使用者

- 預設值 — 無
- 資料類型 — 字串

透過 IDP 使用 IAM 身分驗證進行連線時，這是 idp_host 伺服器的使用者名稱。使用標準身分驗證時，這可以用於 Amazon Redshift 資料庫使用者名稱。

此為選用參數。

webIdentityToken

- 預設值 — 無
- 資料類型 — 字串

由身分提供者提供的 OAuth 2.1 存取權杖或 OpenID Connect ID 權杖。您的應用程式必須透過使用 Web 身分提供者對應用程式的使用者進行身分驗證來獲取此權杖。當您指定 Plugin_Name 選項的 BasicJwtCredentialsProvider 時，請務必指定此參數。

如果您指定 Plugin_Name 選項的 BasicJwtCredentialsProvider，則必須有此參數。

JDBC 驅動器 2.x 版之前的版本

請在您的工具需要特定版本的驅動器時，才下載 Amazon Redshift JDBC 驅動器 2.x 版之前的版本。

下列是之前的 JDBC 4.2 相容 JDBC 驅動器 2.x 版驅動器：

- 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 連接器

透過使用適用於 Python 的 Amazon Redshift 連接器，您可以將工作與適用於 Python (Boto3) 的 AWS SDK，以及 pandas 和數值 Python (NumPy) 整合。如需 pandas 的相關資訊，請參閱 pandas GitHub 儲存庫。如需 NumPy 的相關資訊，請參閱 NumPy GitHub 儲存庫。

Amazon Redshift Python 連接器提供了開放原始碼的解決方案。您可以瀏覽原始碼、請求增強功能、回報問題並提供貢獻。

若要使用 Amazon Redshift Python 連線器，請確定您的 Python 是 3.6 版或更新版本。如需詳細資訊，請參閱 Amazon Redshift Python 驅動程式授權合約。

Amazon Redshift Python 連接器可提供以下功能：

- AWS Identity and Access Management (IAM) 身分驗證。如需詳細資訊，請參閱Amazon Redshift 中的身分和存取管理。
- 使用聯合 API 存取的身分提供者身分驗證。以下企業身分提供者可支援聯合 API 存取：
  - Azure AD。如需詳細資訊，請參閱 AWS 大數據部落格文章使用 Microsoft Azure AD 單一登入聯合 Amazon Redshift 存取。
  - Active Directory Federation Services。如需詳細資訊，請參閱 AWS 大數據部落格文章將 Amazon Redshift 叢集的存取與 Active Directory Federation Services (AD FS) 聯合：第 1 部分。
  - Okta。如需詳細資訊，請參閱 AWS 大數據部落格文章以身分提供者 Okta 聯合 Amazon Redshift 存取。
  - PingFederate。如需詳細資訊，請參閱 PingFederate 網站。
  - JumpCloud。如需詳細資訊，請參閱 JumpCloud 網站。
- Amazon Redshift 資料類型。

Amazon Redshift Python 連接器會實作 Python Database API Specification 2.0。如需詳細資訊，請參閱 Python 網站上的 PEP 249—Python Database API Specification v2.0。

主題

- 安裝 Amazon Redshift Python 連接器
- Amazon Redshift Python 連接器的組態選項
- 匯入 Python 連接器
- 將 Python 連接器與 NumPy 進行整合
- 將 Python 連接器與 pandas 進行整合
- 使用身分提供者外掛程式
- 使用 Amazon Redshift Python 連接器的範例
- Amazon Redshift Python 連接器的 API 參考

安裝 Amazon Redshift Python 連接器

您可以使用下列任一方法來安裝 Amazon Redshift Python 連線器：

- Python Package Index (PyPI)
- Conda
- 複製 GitHub 儲存庫

從 PyPI 安裝 Python 連接器

若要從 Python Package Index (PyPI) 安裝 Python 連線器，您可以使用 pip。若要進行這項動作，請執行以下命令。

```
>>> pip install redshift_connector
```

您可以在虛擬環境內安裝連接器。若要進行這項動作，請執行以下命令。

```
>>> pip install redshift_connector
```

或者，您也可以選擇安裝 pandas 和 NumPy 與連接器。

```
>>> pip install 'redshift_connector[full]'
```

如需 pip 的相關資訊，請參閱 pip 網站。

從 Conda 安裝 Python 連接器

您可以從 Anaconda.org 安裝 Python 連接器。

```
>>>conda install -c conda-forge redshift_connector
```

從 複製 GitHub 儲存庫來安裝 Python 連接器 AWS

若要從來源安裝 Python 連接器，請從 AWS複製 GitHub 儲存庫。在安裝 Python 和 virtualenv 後，請透過執行以下命令來設定環境並安裝所需的相依性。

```
$ 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 連接器的組態選項

接下來，您可以找到可為 Amazon Redshift Python 連接器指定之選項的描述。除非另有指明，否則以下選項適用於最新推出的連接器版本。

access_key_id

- 預設值 — 無
- 資料類型 — 字串

針對 IAM 資料庫身分驗證而設定之 IAM 角色或使用者的存取金鑰。

此為選用參數。

allow_db_user_override

- 預設值 — False
- 資料類型 — 布林值

True

指定連接器會使用安全性聲明標記語言 (SAML) 聲明中的 `DbUser` 值。

False

指定使用 `DbUser` 連線參數中的值。

此為選用參數。

app_name

- 預設值 — 無
- 資料類型 — 字串

用於身分驗證之身分提供者 (IdP) 應用程式的名稱。

此為選用參數。

application_name

- 預設值 — 無
- 資料類型 — 字串

要傳遞至 Amazon Redshift 以進行稽核的用戶端應用程式名稱。您提供的應用程式名稱會出現在 SYS_CONNECTION_LOG 資料表的 'application_name' 欄中。此名稱有助於在偵錯問題時，追蹤連線來源並進行故障診斷。

此為選用參數。

auth_profile

- 預設值 — 無
- 資料類型 — 字串

具有 JSON 形式連線屬性之 Amazon Redshift 身分驗證設定檔的名稱。如需有關命名連線參數的詳細資訊，請參閱 `RedshiftProperty` 類別。`RedshiftProperty` 類別會儲存最終使用者提供的連線參數，以及 (適用的話) 在 IAM 身分驗證程序 (例如臨時 IAM 憑證) 進行期間產生的連線參數。如需詳細資訊，請參閱 RedshiftProperty 類別。

此為選用參數。

auto_create

- 預設值 — False
- 資料類型 — 布林值

此值會指出是否要在使用者不存在時建立使用者。

此為選用參數。

client_id

- 預設值 — 無
- 資料類型 — 字串

來自 Azure IdP 的用戶端 ID。

此為選用參數。

client_secret

- 預設值 — 無
- 資料類型 — 字串

來自 Azure IdP 的用戶端祕密。

此為選用參數。

cluster_identifier

- 預設值 — 無
- 資料類型 — 字串

Amazon Redshift 叢集的叢集識別碼。

此為選用參數。

credentials_provider

- 預設值 — 無
- 資料類型 — 字串

用於以 Amazon Redshift 進行驗證的 IdP。有效值如下：

- `AdfsCredentialsProvider`
- `AzureCredentialsProvider`
- `BrowserAzureCredentialsProvider`
- `BrowserAzureOAuth2CredentialsProvider`
- `BrowserIdcAuthPlugin` - 使用 AWS IAM Identity Center 的授權外掛程式。
- `BrowserSamlCredentialsProvider`
- `IdpTokenAuthPlugin` – 從連結至 AWS IAM Identity Center AWS 的任何 Web 身分提供者接受 IAM Identity Center 字符或 OpenID Connect (OIDC) JSON 型身分字符 (JWT) 的授權外掛程式。
- `PingCredentialsProvider`
- `OktaCredentialsProvider`

此為選用參數。

資料庫

- 預設值 — 無
- 資料類型 — 字串

要連線到之資料庫的名稱。

此為必要參數。

database_metadata_current_db_only

- 預設值 — True
- 資料類型 — 布林值

此值會指出應用程式是否支援多資料庫資料共用型錄。預設值 True 會指出應用程式不會為了回溯相容性而支援多資料庫資料共用型錄。

此為選用參數。

db_groups

- 預設值 — 無
- 資料類型 — 字串

DbUser 所指出的使用者針對目前的工作階段所加入之現有資料庫群組名稱的逗號分隔清單。

此為選用參數。

db_user

- 預設值 — 無
- 資料類型 — 字串

要與 Amazon Redshift 搭配使用的使用者 ID。

此為選用參數。

enable_table_types

- 預設值 — True
- 資料類型 — 布林值

指定連接器是否在 get_tables和get_table_types中繼資料方法的結果中辨識來自資料來源的詳細資料表類型資訊。根據預設，連接器會辨識詳細的資料表類型。

- 真：連接器會辨識下列資料表類型：TABLE、VIEW、 SYSTEM TABLE、 SYSTEM VIEW、EXTERNAL TABLE 和 LOCAL TEMPORARY。
- 錯誤：連接器會將詳細資料表類型資訊標準化為一般 TABLE 和 VIEW 資料表類型。

此為選用參數。它適用於連接器 2.1.16 版和更新版本。

endpoint_url

- 預設值 — 無
- 資料類型 — 字串

Amazon Redshift 端點的 URL。此選項僅供 AWS 內部使用。

此為選用參數。

group_federation

- 預設值 — False
- 資料類型 — 布林值

此選項會指定是否使用 Amazon Redshift IDP 群組。

此為選用參數。

true

使用 Amazon Redshift 身分提供者 (IdP) 群組。

false

針對使用者聯合使用 STS API 和 GetClusterCredentials，並指定連線的 db_groups。

託管

- 預設值 — 無
- 資料類型 — 字串

Amazon Redshift 叢集的主機名稱。

此為選用參數。

iam

- 預設值 — False
- 資料類型 — 布林值

會啟用 IAM 身分驗證。

此為必要參數。

iam_disable_cache

- 預設值 — False
- 資料類型 — 布林值

此選項會指定是否快取 IAM 憑證。預設會快取 IAM 憑證。當對於 API 閘道的請求遭到限流時，這可以提高效能。

此為選用參數。

idc_client_display_name

- 預設值 - Amazon Redshift Python 連接器
- 資料類型 — 字串

使用 BrowserIdcAuthPlugin 的用戶端要使用的顯示名稱。

此為選用參數。

idc_region

- 預設值 — 無

- 資料類型 — 字串

IAM Identity Center AWS 執行個體所在的 AWS 區域。

只有在 credentials_provider 組態選項中使用 `BrowserIdcAuthPlugin` 進行身分驗證時，才需要此參數。

idp_partition

- 預設值 — 無
- 資料類型 — 字串

指定設定身分提供者 (IdP) 的雲端分割區。這會決定驅動程式連線到哪個 IdP 身分驗證端點。

如果此參數保留空白，驅動程式會預設為商業分割區。可能值為：

- `us-gov`：如果您的 IdP 是在 Azure Government 中設定，請使用此值。例如，Azure AD Government 會使用端點 `login.microsoftonline.us`。
- `cn`：如果您的 IdP 是在中國雲端分割區中設定，請使用此值。例如，Azure AD 中國使用端點 `login.chinacloudapi.cn`。

此為選用參數。

idpPort

- 預設值 — 7890
- 資料類型 — 整數

IdP 會將 SAML 聲明傳送到的接聽連接埠。

此為必要參數。

idp_response_timeout

- 預設值 — 120
- 資料類型 — 整數

從 IdP 擷取 SAML 聲明的逾時。

此為必要參數。

idp_tenant

- 預設值 — 無
- 資料類型 — 字串

IdP 租用戶。

此為選用參數。

issuer_url

- 預設值 — 無
- 資料類型 — 字串

指向 AWS IAM Identity Center 伺服器的執行個體端點。

只有在 credentials_provider 組態選項中使用 `BrowserIdcAuthPlugin` 進行身分驗證時，才需要此參數。

listen_port

- 預設值 — 7890
- 資料類型 — 整數

透過瀏覽器外掛程式使用 SAML、Azure AD 或 IAM Identity Center AWS 服務時，驅動程式用來從身分提供者或授權碼接收 SAML 回應的連接埠。

此為選用參數。

login_url

- 預設值 — 無
- 資料類型 — 字串

IdP 的單一登入 Url。

此為選用參數。

max_prepared_statements

- 預設值 — 1000
- 資料類型 — 整數

每個連線將快取的預備陳述式數量上限。將此參數設為 0 會停用快取機制。針對此參數輸入負數會將其設定為預設值。

此為選用參數。

numeric_to_float

- 預設值 — False
- 資料類型 — 布林值

此選項指定連接器是否會將數值資料類型值從 decimal.Decimal 轉換為浮點數。依預設，連接器會接收 decimal.Decimal 形式的數值資料類型值，且不會對值進行轉換。

對於需要精確度的使用案例，不建議您啟用 numeric_to_float，因為得到的結果可能會經過四捨五入。

如需 decimal.Decimal 以及其與浮點數之間權衡的相關資訊，請參閱 Python 網站上的小數 — 固定小數點和浮點數。

此為選用參數。

partner_sp_id

- 預設值 — 無
- 資料類型 — 字串

針對以 Ping 進行的身分驗證所使用的合作夥伴 SP ID。

此為選用參數。

password

- 預設值 — 無
- 資料類型 — 字串

用於身分驗證的密碼。

此為選用參數。

port

- 預設值 — 5439
- 資料類型 — 整數

Amazon Redshift 叢集的連接埠號碼。

此為必要參數。

preferred_role

- 預設值 — 無
- 資料類型 — 字串

目前連線偏好的 IAM 角色。

此為選用參數。

principal_arn

- 預設值 — 無
- 資料類型 — 字串

要為其產生政策之使用者或 IAM 角色的 Amazon Resource Name (ARN)。建議您將政策連接至角色，然後將角色指派給您的使用者以提供存取權。

此為選用參數。

profile

- 預設值 — 無
- 資料類型 — 字串

AWS 登入資料檔案中包含 AWS 登入資料的設定檔名稱。

此為選用參數。

provider_name

- 預設值 — 無
- 資料類型 — 字串

Redshift 原生身分驗證提供者的名稱。

此為選用參數。

region

- 預設值 — 無
- 資料類型 — 字串

AWS 區域 叢集所在的 。

此為選用參數。

role_arn

- 預設值 — 無
- 資料類型 — 字串

呼叫者所擔任角色的 Amazon Resource Name (ARN)。`JwtCredentialsProvider` 所指出的提供者會使用此參數。

若為 `JwtCredentialsProvider` 提供者，此為必要參數。否則，此為選擇性參數。

role_session_name

- 預設值 — jwt_redshift_session
- 資料類型 — 字串

擔任角色工作階段的識別碼。一般來說，您會傳遞與應用程式使用者相關聯的名稱或識別碼。應用程式使用的臨時安全憑證會與該使用者相關聯。`JwtCredentialsProvider` 所指出的提供者會使用此參數。

此為選用參數。

scope

- 預設值 — 無
- 資料類型 — 字串

使用者可以同意的範圍清單 (以空格分隔)。指定這個參數可讓您的應用程式針對您想要呼叫的 API 取得同意。當您為 credentials_provider 選項指定 BrowserAzureOAuth2CredentialsProvider 時，便可以指定此參數。

BrowserAzureOAuth2CredentialsProvider 外掛程式需要此參數。

secret_access_key_id

- 預設值 — 無
- 資料類型 — 字串

針對 IAM 資料庫身分驗證而設定之 IAM 角色或使用者的私密存取金鑰。

此為選用參數。

session_token

- 預設值 — 無
- 資料類型 — 字串

針對 IAM 資料庫身分驗證而設定之 IAM 角色或使用者的存取金鑰。如果使用臨時 AWS 登入資料，則需要此參數。

此為選用參數。

serverless_acct_id

- 預設值 — 無
- 資料類型 — 字串

Amazon Redshift Serverless 帳戶 ID。

此為選用參數。

serverless_work_group

- 預設值 — 無
- 資料類型 — 字串

Amazon Redshift Serverless 工作群組名稱。

此為選用參數。

ssl

- 預設值 — True
- 資料類型 — 布林值

會啟用 Secure Sockets Layer (SSL)。

此為必要參數。

ssl_insecure

- 預設值 — False
- 資料類型 — 布林值

此值會指定是否要停用驗證 IdP 主機的伺服器 SSL 憑證。將此參數設定為 True 會停用驗證 IdP 主機的伺服器 SSL 憑證。建議您在生產環境中保留預設值 False。

此為選用參數。

sslmode

- 預設值 — verify-ca
- 資料類型 — 字串

連線至 Amazon Redshift 的安全。您可以指定下列任一項：

- verify-ca
- verify-full

> **Note**
>
> 我們建議您盡可能使用 `verify-full`進行 SSL 連線。

此為必要參數。

tcp_keepalive

- 預設值 — True
- 資料類型 — 布林值

是否使用 TCP 保持連線來防止連線逾時。您可以指定下列值：

- True：驅動器將使用 TCP 保持連線來防止連線逾時。
- False：驅動器不會使用 TCP 保持連線。

此為選用參數。

tcp_keepalive_count

- 預設值 — 無
- 資料類型 — 整數

在判斷連線為非作用中之前，要傳送的未確認探查數目。例如，將值設定為 3，表示驅動器會先傳送 3 個未回應的保持連線封包，才確定連線非作用中。

如果未指定此參數，Amazon Redshift 會使用系統的預設值。

此為選用參數。

tcp_keepalive_interval

- 預設值 — 無
- 資料類型 — 整數

後續保持連線探查之間的間隔 (以秒為單位)，如果驅動器未在此間隔之前收到探查的確認。如果您指定此參數，其必須是正整數。

如果未指定此參數，Amazon Redshift 會使用系統的預設值。

此為選用參數。

tcp_keepalive_idle

- 預設值 — 無
- 資料類型 — 整數

無活動的持續時間 (以秒為單位)，經過這段時間後，驅動器會傳送第一個保持連線探查。例如，將值設定為 120，表示驅動器會等待 2 分鐘無活動的時間，才傳送第一個保持連線封包。如果您指定此參數，其必須是正整數。

如果未指定此參數，Amazon Redshift 會使用系統的預設值。

此為選用參數。

timeout

- 預設值 — 無
- 資料類型 — 整數

對伺服器的連線在逾時前要經過的秒數。

此為選用參數。

token

- 預設值 — 無
- 資料類型 — 字串

IAM Identity Center AWS 提供存取字符或由與 IAM Identity Center 連結的 Web 身分提供者提供的 OpenID Connect (OIDC) JSON Web Token AWS (JWT)。您的應用程式必須透過使用 IAM Identity Center AWS 或與 IAM Identity Center 連結的身分提供者驗證應用程式的使用者來產生此字符 AWS 。

此參數適用於 `IdpTokenAuthPlugin`。

token_type

- 預設值 — 無

- 資料類型 — 字串

IdpTokenAuthPlugin 中使用的權仗類型。

您可以指定下列值：

ACCESS_TOKEN

如果您使用 AWS IAM Identity Center 提供的存取權杖，請輸入此選項。

EXT_JWT

如果您使用與 AWS IAM Identity Center 整合的 Web 型身分提供者所提供的 OpenID Connect (OIDC) JSON Web Token (JWT)，請輸入此項。

此參數適用於 IdpTokenAuthPlugin。

user

- 預設值 — 無
- 資料類型 — 字串

用於身分驗證的使用者名稱。

此為選用參數。

web_identity_token

- 預設值 — 無
- 資料類型 — 字串

由身分提供者提供的 OAuth 2.0 存取權杖或 OpenID Connect ID 權杖。請確保您的應用程式會透過以 Web 身分提供者對應用程式的使用者進行驗證來取得此權杖。JwtCredentialsProvider 所指出的提供者會使用此參數。

若為 JwtCredentialsProvider 提供者，此為必要參數。否則，此為選擇性參數。

匯入 Python 連接器

若要匯入 Python 連接器，請執行以下命令。

```
>>> import redshift_connector
```

若要使用 AWS 登入資料連線至 Amazon Redshift 叢集，請執行下列命令。

```
conn = redshift_connector.connect(
    host='examplecluster.abc123xyz789.us-west-1.redshift.amazonaws.com',
    port=5439,
    database='dev',
    user='awsuser',
    password='my_password'
 )
```

將 Python 連接器與 NumPy 進行整合

以下是將 Python 連接器與 NumPy 進行整合的範例。

```
>>>  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)
```

結果如下。

```
[['One Hundred Years of Solitude' 'Gabriel García Márquez']
['A Brief History of Time' 'Stephen Hawking']]
```

將 Python 連接器與 pandas 進行整合

以下是將 Python 連接器與 pandas 進行整合的範例。

```
>>> 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)
```

使用身分提供者外掛程式

如需如何使用身分提供者外掛程式的一般資訊，請參閱提供 IAM 登入資料的選項。如需如何管理 IAM 身分的相關資訊 (包括 IAM 角色的最佳實務)，請參閱 Amazon Redshift 中的身分和存取管理。

使用 ADFS 身分提供者外掛程式進行身分驗證

以下是使用 Active Directory Federation Service (ADFS) 身分提供者外掛程式來針對連線至 Amazon Redshift 資料庫的使用者進行驗證的範例。

```
>>> 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 身分提供者外掛程式進行身分驗證

以下是使用 Azure 身分提供者外掛程式進行身分驗證的範例。您可以為 Azure 企業應用程式建立 `client_id` 和 `client_secret` 的值，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 身分提供者外掛程式進行身分驗證

以下是使用 IAM Identity Center AWS 身分提供者外掛程式進行身分驗證的範例。

```
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 Browser 身分提供者外掛程式進行身分驗證

以下是使用 Azure Browser 身分提供者外掛程式來針對連線至 Amazon Redshift 資料庫的使用者進行驗證的範例。

由使用者提供登入憑據的瀏覽器會進行多重要素驗證。

```
>>>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 身分提供者外掛程式進行身分驗證

以下是使用 Okta 身分提供者外掛程式進行身分驗證的範例。您可以透過 Okta 應用程式取得 `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'
)
```

搭配使用 JumpCloud 與一般 SAML 瀏覽器身分提供者外掛程式來進行身分驗證

以下是搭配使用 JumpCloud 與一般 SAML 瀏覽器身分提供者外掛程式來進行身分驗證的範例。

密碼是必要參數。但是，您不必輸入此參數，因為瀏覽器會進行多重要素驗證。

```
>>> 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 連接器的範例

以下是如何使用 Amazon Redshift Python 連接器的範例。若要執行這些範例，您必須先安裝 Python 連接器。如需安裝 Amazon Redshift Python 連接器的相關資訊，請參閱安裝 Amazon Redshift Python 連接器。如需可與 Python 連接器搭配使用之組態選項的相關資訊，請參閱 Amazon Redshift Python 連接器的組態選項。

主題

- 使用 AWS 登入資料連線至和查詢 Amazon Redshift 叢集
- 啟用自動遞交
- 設定游標 paramstyle
- 使用 COPY 從 Amazon S3 儲存貯體複寫資料，然後使用 UNLOAD 將資料寫入至 Amazon S3 儲存貯體

使用 AWS 登入資料連線至和查詢 Amazon Redshift 叢集

下列範例會引導您使用 AWS 登入資料連線至 Amazon Redshift 叢集，然後查詢資料表並擷取查詢結果。

```
#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'])
```

啟用自動遞交

自動遞交屬性會預設為關閉，以遵循 Python Database API Specification。執行復原命令後，您可以使用以下命令開啟連線的 autocommit 屬性，以確保沒有進行中的交易。

```
#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
```

設定游標 paramstyle

游標的 paramstyle 可透過 cursor.paramstyle 進行修改。所使用的預設 paramstyle 是 `format`。此參數的有效值為 `qmark`、`numeric`、`named`、`format` 和 `pyformat`。

以下是使用各種 paramstyle 將參數傳遞至範例 SQL 陳述式的範例。

```
# 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 從 Amazon S3 儲存貯體複寫資料，然後使用 UNLOAD 將資料寫入至 Amazon S3 儲存貯體

下列範例顯示如何將資料從 Amazon S3 儲存貯體複製到資料表，然後從該資料表卸載回儲存貯體。

名為 `category_csv.txt` 且包含下列資料的文字檔案會上傳至 Amazon S3 儲存貯體：。

```
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 程式碼範例，其會先連線至 Amazon Redshift 資料庫。然後，會建立名為 `category` 的資料表，並將 S3 儲存貯體中的 CSV 資料複製到資料表中。

```
#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'])
```

如果您未將 autocommit 設定為 true，請在執行 execute() 陳述式後使用 conn.commit() 遞交。

資料會卸載到 S3 儲存貯體中的 unloaded_category_csv.text0000_part00 檔案中，其內容如下：

```
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 連接器的 API 參考

接下來，您可以找到 Amazon Redshift Python 連線器 API 操作的描述。

redshift_connector

接下來，您可以找到 redshift_connector API 操作的描述。

connect(user, database, password[, port, …])

建立對 Amazon Redshift 叢集的連線。此函數會驗證使用者輸入、選擇性地使用身分提供者外掛程式進行驗證，然後建構連線物件。

apilevel

支援 DBAPI 層級，目前為「2.0」。

paramstyle, str(object='') -> str str(bytes_or_buffer[, encoding[, errors]]) -> str

要全域使用的資料庫 API 參數樣式。

連線

接下來，您可以找到 Amazon Redshift Python 連接器的連線 API 操作描述。

__init__(user, password, database[, host, …])

將原始連線物件初始化。

cursor

建立繫結至此連線的游標物件。

commit

遞交目前的資料庫交易。

rollback

復原目前的資料庫交易。

close

關閉資料庫連線。

execute(cursor, operation, vals)

執行指定的 SQL 命令。根據 redshift_connector.paramstyle 的值，您可以提供序列或映射形式的參數。

run(sql[, stream])

執行指定的 SQL 命令。您可以選擇性提供與 COPY 命令搭配使用的串流。

xid(format_id, global_transaction_id, …)

建立交易 ID。只有 global_transaction_id 參數會用於 postgres。format_id 和 branch_qualifier 不會用於 postgres。global_transaction_id 可以是由傳回元組 (format_id, global_transaction_id, branch_qualifier) 的 postgres 支援的任何字串識別碼。

tpc_begin(xid)

使用由格式 ID、全域交易 ID 和分支限定詞組成的交易 ID xid 來開始 TPC 交易。

`tpc_prepare`

執行以 .tpc_begin 開始之交易的第一個階段。

`tpc_commit([xid])`

以無引數方式呼叫時，.tpc_commit 會遞交先前使用 .tpc_prepare() 所準備的 TPC 交易。

`tpc_rollback([xid])`

以無引數方式呼叫時，.tpc_rollback 會復原 TPC 交易。

`tpc_recover`

傳回適合與 .tpc_commit(xid) 或 .tpc_rollback(xid) 搭配使用的待定交易 ID 清單。

游標

接下來，您可以找到游標 API 操作的描述。

`__init__(connection[, paramstyle])`

將原始游標物件初始化。

`insert_data_bulk(filename, table_name, parameter_indices, column_names, delimiter, batch_size)`

執行大量 INSERT 陳述式。

`execute(operation[, args, stream, …])`

執行資料庫操作。

`executemany(operation, param_sets)`

準備資料庫操作，然後針對所提供的所有參數序列或映射執行此操作。

`fetchone`

擷取查詢結果集的下一列。

`fetchmany([num])`

擷取下一組查詢結果資料列。

`fetchall`

擷取查詢結果的所有剩餘資料列。

`close`

立即關閉游標。

`__iter__`

可對游標物件進行迭代以從查詢中擷取資料列。

`fetch_dataframe([num])`

傳回上次查詢結果的資料框。

`write_dataframe(df, table)`

將相同結構資料框寫入到 Amazon Redshift 資料庫。

`fetch_numpy_array([num])`

傳回上次查詢結果的 NumPy 陣列。

`get_catalogs`

Amazon Redshift 不支援來自單一連線的多個型錄。Amazon Redshift 只會傳回目前的型錄。

`get_tables([catalog, schema_pattern, …])`

傳回系統內使用者定義的唯一公有資料表。

`get_columns([catalog, schema_pattern, …])`

傳回 Amazon Redshift 資料庫中特定資料表內所有資料欄的清單。

AdfsCredentialsProvider 外掛程式

以下是 Amazon Redshift Python 連接器的 AdfsCredentialsProvider 外掛程式 API 操作語法。

```
redshift_connector.plugin.AdfsCredentialsProvider()
```

AzureCredentialsProvider 外掛程式

以下是 Amazon Redshift Python 連接器的 AzureCredentialsProvider 外掛程式 API 操作語法。

```
redshift_connector.plugin.AzureCredentialsProvider()
```

BrowserAzureCredentialsProvider 外掛程式

以下是 Amazon Redshift Python 連接器的 BrowserAzureCredentialsProvider 外掛程式 API 操作語法。

```
redshift_connector.plugin.BrowserAzureCredentialsProvider()
```

BrowserSamlCredentialsProvider 外掛程式

以下是 Amazon Redshift Python 連接器的 BrowserSamlCredentialsProvider 外掛程式 API 操作語法。

```
redshift_connector.plugin.BrowserSamlCredentialsProvider()
```

OktaCredentialsProvider 外掛程式

以下是 Amazon Redshift Python 連接器的 OktaCredentialsProvider 外掛程式 API 操作語法。

```
redshift_connector.plugin.OktaCredentialsProvider()
```

PingCredentialsProvider 外掛程式

以下是 Amazon Redshift Python 連接器的 PingCredentialsProvider 外掛程式 API 操作語法。

```
redshift_connector.plugin.PingCredentialsProvider()
```

SamlCredentialsProvider 外掛程式

以下是 Amazon Redshift Python 連接器的 SamlCredentialsProvider 外掛程式 API 操作語法。

```
redshift_connector.plugin.SamlCredentialsProvider()
```

Apache Spark 的 Amazon Redshift 整合

Apache Spark 是一種分散式處理架構和程式設計模型，可協助您執行機器學習、串流處理或圖形分析。與 Apache Hadoop 類似，Spark 是一種開放原始碼、分散式處理系統，通常用於大數據的工作負

載。Spark 具有經過最佳化的有向無環圖 (DAG) 執行引擎，並且會主動快取記憶體中的資料。這可以提高效能，對於某些算法和互動式查詢來說更是如此。

這項整合可為您提供 Spark 連接器，讓您用於建置在 Amazon Redshift 和 Amazon Redshift Serverless 中讀取和寫入資料的 Apache Spark 應用程式。這些應用程式不會影響應用程式效能或資料的交易一致性。這項整合會自動納入到 Amazon EMR 和 AWS Glue，因此您可以立即執行會在資料擷取和轉換管道中存取資料並將資料載入到 Amazon Redshift 的 Apache Spark 任務。

目前，您可以將 3.3.x、3.4.x、3.5.x、4.0.x 和 4.1.x 版的 Spark 與此整合搭配使用。

這項整合會提供下列功能：

- AWS Identity and Access Management (IAM) 身分驗證。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。
- 述詞和查詢下推，以提升效能。
- Amazon Redshift 資料類型。
- 對 Amazon Redshift 和 Amazon Redshift Serverless 的連線。

使用 Spark 連接器時的考量和限制

- tempdir URI 會指向 Amazon S3 位置。此臨時目錄不會自動清理，因此可能會增加額外的費用。建議您使用《Amazon Simple Storage Service 使用者指南》中的 Amazon S3 生命週期政策來定義 Amazon S3 儲存貯體的保留規則。
- 根據預設，如果 Amazon S3和 Redshift 叢集位於不同的 AWS 區域，Amazon S3 和 Redshift 之間的複本將無法運作。若要使用不同的 AWS 區域，請將 `tempdir_region` 參數設定為用於 的 S3 儲存貯體區域`tempdir`。
- 如果使用 `tempformat` 參數寫入 Parquet 資料，則會在 S3 和 Redshift 之間進行跨區域寫入。
- 建議您使用 Amazon S3 伺服器端加密來加密所用的 Amazon S3 儲存貯體。
- 建議您封鎖對 Amazon S3 儲存貯體的公開存取。
- 建議您不要將 Amazon Redshift 叢集設為可公開存取。
- 建議您開啟 Amazon Redshift 稽核記錄。
- 建議您開啟 Amazon Redshift 靜態加密。
- 建議您針對從 Amazon EMR 上的 Spark 到 Amazon Redshift 的 JDBC 連線開啟 SSL。
- 建議您使用 Amazon Redshift 身分驗證參數的 `aws_iam_role` 參數傳遞 IAM 角色。

使用 Spark 連接器進行身分驗證

下圖描述了 Amazon S3、Amazon Redshift、Spark 驅動程式和 Spark 執行程式之間的身分驗證。

Redshift 和 Spark 之間的身分驗證

您可以使用 Amazon Redshift 提供的 JDBC 驅動器 2.x 版，透過指定登入憑證使用 Spark 連接器連線到 Amazon Redshift。若要使用 IAM，請設定 JDBC url 以使用 IAM 身分驗證。若要從 Amazon EMR 或 連線至 Redshift 叢集 AWS Glue，請確定您的 IAM 角色具有擷取臨時 IAM 登入資料的必要許可。下列清單描述 IAM 角色要擷取憑證和執行 Amazon S3 操作所需的所有許可。

- Redshift:GetClusterCredentials (適用於佈建的 Redshift 叢集)
- Redshift:DescribeClusters (適用於佈建的 Redshift 叢集)
- Redshift:GetWorkgroup (適用於 Amazon Redshift Serverless 工作群組)
- Redshift:GetCredentials (適用於 Amazon Redshift Serverless 工作群組)
- s3:ListBucket
- s3:GetBucket
- s3:GetObject
- s3:PutObject
- s3:GetBucketLifecycleConfiguration

如需 GetClusterCredentials 的詳細資訊，請參閱 GetClusterCredentials 的 IAM 政策。

您還必須確保 Amazon Redshift 可以在 COPY 和 UNLOAD 操作期間擔任 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 驅動程式，該驅動程式會自動管理從 Amazon Redshift 自我簽署憑證到 ACM 憑證的轉換。但是，您必須指定 JDBC url 的 SSL 選項。

以下是如何指定 JDBC 驅動程式 URL 和 aws_iam_role 以連線到 Amazon Redshift 的範例。

```
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 之間的身分驗證

如果您使用 IAM 角色在 Spark 和 Amazon S3 之間進行驗證，請使用下列其中一種方法：

- 適用於 Java 的 AWS SDK 會使用 DefaultAWSCredentialsProviderChain 類別實作的預設登入資料提供者鏈結，自動嘗試尋找 AWS 登入資料。如需詳細資訊，請參閱使用預設憑證提供者鏈結。
- 您可以透過 Hadoop 組態屬性指定 AWS 金鑰。例如，如果您的 tempdir 組態指向 s3n:// 檔案系統，請在 Hadoop XML 組態檔案中設定 fs.s3n.awsAccessKeyId 和 fs.s3n.awsSecretAccessKey 屬性，或呼叫 sc.hadoopConfiguration.set() 以變更 Spark 的全域 Hadoop 組態。

例如，假設您使用 s3n 檔案系統，請新增：

```
sc.hadoopConfiguration.set("fs.s3n.awsAccessKeyId", "YOUR_KEY_ID")
sc.hadoopConfiguration.set("fs.s3n.awsSecretAccessKey", "YOUR_SECRET_ACCESS_KEY")
```

若為 s3a 檔案系統，請新增：

```
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 Python，請使用下列操作：

```
sc._jsc.hadoopConfiguration().set("fs.s3n.awsAccessKeyId", "YOUR_KEY_ID")
sc._jsc.hadoopConfiguration().set("fs.s3n.awsSecretAccessKey",
 "YOUR_SECRET_ACCESS_KEY")
```

- 在 `tempdir` URL 中編碼身分驗證金鑰。例如，URI `s3n://ACCESSKEY:SECRETKEY@bucket/path/to/temp/dir` 會編碼金鑰對 (ACCESSKEY,SECRETKEY)。

Redshift 和 Amazon S3 之間的身分驗證

如果您在查詢中使用 COPY 和 UNLOAD 命令，則也必須向 Amazon S3 授予 Amazon Redshift 的存取權，以代表您執行查詢。若要這麼做，請先授權 Amazon Redshift 存取其他 AWS 服務，然後使用 IAM 角色授權 COPY 和 UNLOAD 操作。

我們建議的最佳實務是，將許可政策附加到 IAM 角色，然後根據需要將其指派給使用者和群組。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。

與 整合 AWS Secrets Manager

您可以從 AWS Secrets Manager中儲存的機密擷取 Redshift 的使用者名稱和密碼憑證。若要自動提供 Redshift 憑證，請使用 `secret.id` 參數。如需如何建立 Redshift 憑證機密的相關資訊，請參閱建立 AWS Secrets Manager 資料庫機密。

GroupID	ArtifactID	支援的修訂	說明
com.amazonaws.secretsmanager	aws-secretsmanager-jdbc	1.0.12	適用於 Java 的 AWS Secrets Manager SQL Connection Library 可讓 Java 開發人員使用 中存放的秘密輕鬆連線至 SQL 資料庫 AWS Secrets Manager。

> **Note**
>
> 確認：本文件包含由 Apache Software Foundation 根據 Apache 2.0 授權所開發的範例程式碼和語言。

透過下推功能改善效能

Spark 連接器會自動套用述詞和查詢下推功能，以針對效能進行最佳化。這項支援表示，如果您在查詢中使用受支援的函數，Spark 連接器便會將該函數轉換為 SQL 查詢，並在 Amazon Redshift 中執行該查詢。此最佳化可減少所擷取的資料，因此 Apache Spark 可以處理較少的資料，獲得較好的效能。依預設，系統會自動啟動下推功能。若要加以停用，請將 `autopushdown` 設定為 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()
```

下推功能支援下列函數。如果您使用的函數不在此清單中，Spark 連接器會在 Spark 中執行此函數，而不是在 Amazon Redshift 中執行，因此無法獲得最佳效能。如需 Spark 中函數的完整清單，請參閱內建函數。

- 彙總函數
  - avg
  - count
  - max
  - min
  - sum
  - stddev_samp
  - stddev_pop
  - var_samp
  - var_pop
- 布林值運算子
  - in
  - isnull
  - isnotnull
  - contains
  - endswith

  - startswith
- **邏輯運算子**
  - 及
  - 或
  - not (or !)
- **數學函式**
  - +
  - -
  - *
  - /
  - - (unary)
  - abs
  - acos
  - asin
  - atan
  - ceil
  - cos
  - exp
  - floor
  - greatest
  - least
  - log10
  - pi
  - pow
  - round
  - sin
  - sqrt
  - tan
- **其他函數**
  - cast

- coalesce
- decimal
- 如
- in

- 關係運算子
  - !=
  - =
  - >
  - >=
  - <
  - <=
- 字串函數
  - ascii
  - lpad
  - rpad
  - translate
  - upper
  - lower
  - length
  - trim
  - ltrim
  - rtrim
  - like
  - substring
  - concat
- 日期和時間函數
  - add_months
  - date
  - date_add
  - date_sub

  - date_trunc
  - timestamp
  - trunc
- 數學運算
  - CheckOverflow
  - PromotePrecision
- 關係運算
  - Aliases (例如 AS)
  - CaseWhen
  - Distinct
  - InSet
  - Joins 和 cross joins
  - 限制
  - Unions, union all
  - ScalarSubquery
  - Sorts (遞增和遞減)
  - UnscaledValue

其他組態選項

在此頁面上，您可以找到可為 Amazon Redshift Spark 連接器指定之選項的說明。

字串欄的大小上限

在建立儲存為 VARCHAR(256) 的資料表時，Redshift 會將字串資料欄建立為文字資料欄。如果你想要支援更大尺寸的資料欄，則可以使用 maxlength 來指定字串資料欄的最大長度。以下範例說明如何指定 `maxlength`。

```
columnLengthMap.foreach { case (colName, length) =>
  val metadata = new MetadataBuilder().putLong("maxlength", length).build()
  df = df.withColumn(colName, df(colName).as(colName, metadata))
}
```

資料欄類型

若要設定資料欄類型，請使用 redshift_type 欄位。

```
columnTypeMap.foreach { case (colName, colType) =>
  val metadata = new MetadataBuilder().putString("redshift_type", colType).build()
  df = df.withColumn(colName, df(colName).as(colName, metadata))
}
```

欄的壓縮編碼

若要在資料欄上使用特定的壓縮編碼，請使用編碼欄位。如需支援壓縮編碼的完整清單，請參閱壓縮編碼。

欄的說明

若要設定描述，請使用 description 欄位。

Redshift 和 Amazon S3 之間的身分驗證

根據預設，結果會以 parquet 格式卸載到 Amazon S3。若要將結果卸載為以豎線分隔的文字檔案，請指定下列選項。

```
.option("unload_s3_format", "TEXT")
```

Pushdown 陳述式

參數	必要	預設	說明
spark.datasource.redshift.community.autopushdown.lazyMode	否	True	指定連接器是否應延遲執行下推陳述式 Redshift。 如果為 true，則 Spark 連接器會在執行查詢之前先擷取所有相關模型和資訊，這通常會產生較佳的效能。

參數	必要	預設	說明
			如果為 false，則 Spark 連接器會立即在主 Spark 驅動程式執行緒中執行下推陳述式，並跨運算式序列化。

連接器參數

Spark SQL 中的參數映射或 OPTIONS 支援以下設定。

參數	必要	預設	說明
dbtable	是，除非指定查詢	N/A	要在 Redshift 中建立或讀取的資料表。將資料儲存回 Redshift 時，需要此參數。
query	是，除非指定 dbtable	N/A	要在 Redshift 中讀取的查詢。
user	否	N/A	Redshift 的使用者名稱。必須與密碼參數搭配使用。只有在使用者和密碼不是 URL 中的參數時才有效。同時使用兩者會導致錯誤。
password	否	N/A	Redshift 的密碼。必須與使用者參數搭配使用。只有在使用者和密碼不是 URL 中的參數時才有效。同

參數	必要	預設	說明
			時使用兩者會導致錯誤。
url	否	N/A	JDBC URL。格式為 jdbc:subprotocol://host:port/database?user=username&password=password。 視您已載入的 JDBC 驅動程序而定，子通訊協定可以是 postgresql 或 Redshift。請注意，一個與 Redshift 相容的驅動程式必須位於類別路徑中，並符合此 URL。 主機和連接埠應指向 Redshift 主節點，因此您必須將安全群組和/或 VPC 設定為允許從驅動程式應用程式存取。 資料庫是 Redshift 的資料庫名稱。 使用者和密碼是用來存取資料庫的憑證，必須內嵌在 JDBC 的這個 URL 中，而且資料庫使用者必須具備用來存取資料表的必要許可。

參數	必要	預設	說明
aws_iam_role	只有在使用 IAM 角色來授權 Redshift COPY/UNLOAD 操作時	N/A	附加至 Redshift 叢集之 IAM 角色的完整指定 ARN。
forward_spark_s3_credentials	否	False	指出此程式庫是否應自動探索 Spark 用來連線到 Amazon S3 的憑證，以及是否透過 JDBC 驅動程式將這些憑證轉送至 Redshift。這些憑證會在 JDBC 查詢中傳送。因此，建議您在使用此選項時，為 JDBC 連線啟用 SSL 加密。
temporary_aws_access_key_id	否	N/A	AWS 存取金鑰。必須具有 S3 儲存貯體的寫入許可。
temporary_aws_secret_access_key	否	N/A	AWS 對應至存取金鑰的私密存取金鑰。
temporary_aws_session_token	否	N/A	AWS 對應於所提供存取金鑰的工作階段字符。

參數	必要	預設	說明
tempdir	否	N/A	Amazon S3 中的可寫入位置。用於在讀取時卸載資料，並在寫入時將 Avro 資料載入到 Redshift。如果您在一般 ETL 管道中使用適用於 Spark 的 Redshift 資料來源，則在儲存貯體上設定生命週期政策並將其作為此資料的暫存位置會很有用。
jdbcdriver	否	由 JDBC URL 的子通訊協定決定	要使用之 JDBC 驅動程式的類別名稱。這個類別必須在類別路徑上。在大多數情況下，應該不需要指定此選項，因為 JDBC URL 的子通訊協定應該會自動決定適當的驅動程式類別名稱。
diststyle	否	Even	在建立資料表時要使用的 Redshift 分散樣式。有效選項為 EVEN、KEY 或 ALL。使用 KEY 時，也必須使用 distkey 選項設定分散索引鍵。
distkey	否，除非使用 DISTSTYLE_KEY	N/A	要在建立資料表時作為分散索引鍵的資料表資料欄名稱。

參數	必要	預設	說明
sortkeyspec	否	N/A	完整的 Redshift 排序索引鍵定義。
include_column_list	否	False	指出此程式庫是否應根據資料欄映射選項，自動從結構描述擷取資料欄，並將其新增至 COPY 命令。
description	否	N/A	資料表的描述。描述可使用 SQL COMMENT 命令加以設定，並且會出現在大多數查詢工具中。請參閱 `description` 中繼資料以設定個別資料欄的描述。

參數	必要	預設	說明
preactions	否	N/A	要在載入 COPY 命令前執行的分號分隔 SQL 命令清單。在載入新資料前先執行 DELETE 命令或類似命令可能會很有用。如果命令中包含 %s，則系統會在執行期之前先將資料表名稱格式化 (如果您使用的是暫存資料表)。如果此命令失敗，則系統會將其視為例外狀況。如果您使用的是暫存資料表，且預先處理失敗，則系統會回復變更並還原備份資料表。
extracopyoptions	否	N/A	載入資料時要附加至 Redshift COPY 命令的額外選項清單 (例如 TRUNCATECOLUMNS 或 MAXERROR n)。如需可用參數的完整清單，請參閱選用參數。 請注意，由於這些選項會附加到 COPY 命令的結尾，因此您只能使用在命令結尾上具有意義的選項。這應該涵蓋大多數可能的使用案例。

參數	必要	預設	說明
sse_kms_key	否	N/A	在 Redshift UNLOAD 操作期間用於 S3 伺服器端加密的 AWS KMS AWS 金鑰 ID，而非預設加密。Redshift IAM 角色必須具有 KMS 金鑰的存取權才能使用該金鑰來進行寫入，而 Spark IAM 角色必須具有該金鑰的存取權才能進行讀取操作。只要 Spark 的 IAM 角色具有適當的存取權，讀取加密的資料就不需要變更 (AWS 處理此問題）。
tempformat	否	AVRO	在寫入到 Redshift 時，用來在 Amazon S3 中儲存暫存檔案的格式。有效值為 AVRO、CSV、CSV GZIP (壓縮 CSV) 和 PARQUET。
csvnullstring (實驗性)	否	Null	使用 CSV tempformat 時為 Null 值寫入的字串值。這應該是不會出現在實際資料中的值。

參數	必要	預設	說明
autopushdown	否	True	指出是否要透過擷取和分析 SQL 操作的 Spark 邏輯計畫，以套用述詞和查詢下推。這些操作會轉換成 SQL 查詢，然後在 Redshift 中執行以提高效能。
autopushdown.s3_result_cache	否	False	快取查詢 SQL 以在記憶體中卸載 Amazon S3 路徑映射資料，因此不需要在相同的 Spark 工作階段中再次執行相同的查詢。只有在開啟 autopushdown 時才支援。在混合讀取和寫入操作時，不建議使用這個參數，因為快取的結果可能包含過時的資訊。
unload_s3_format	否	Parquet	用來卸載查詢結果的格式。有效選項為 Parquet 和文字，其會指定以豎線分隔的文字格式卸載查詢結果。

參數	必要	預設	說明
extraunloadoptions	否	N/A	要附加至 Redshift UNLOAD 命令的額外選項。並非所有選項都能保證運作，因為某些選項可能會與連接器內設定的其他選項衝突。
copydelay	否	30000	Redshift COPY 操作重試之間的延遲時間 (以毫秒為單位)。
copyretrycount	否	2	要重試 Redshift COPY 操作的次數。

參數	必要	預設	說明
tempdir_region	否	N/A	tempdir 所在的 AWS 區域。設定此選項可改善連接器與 tempdir 互動時的效能，並且會在連接器的讀取和寫入操作期間，自動於 COPY 和 UNLOAD 操作中提供此值。 建議在以下情況使用此設定： 1) 當連接器在外部執行時 AWS，因為自動區域探索將會失敗，並對連接器效能造成負面影響。 2) 當 tempdir 位於與 Redshift 叢集不同的區域時，因為使用此設定就比較不需要使用 extracopyoptions 和 extraunloadoptions 參數手動提供區域。使用 PARQUET 做為 tempformat 時，即使使用此參數，tempdir 也不能位於與 Redshift 叢集不同的區域。

參數	必要	預設	說明
			3) 當連接器在與 `tempdir` 不同的區域中執行時，因為其會改善連接器對於 `tempdir` 的存取效能。
secret.id	否	N/A	儲存在 AWS Secrets Manager之機密的名稱或 ARN。您可以使用此參數自動提供 Redshift 憑證，但前提是使用者、密碼和 `DbUser` 憑證未傳遞到 JDBC URL 或做為其他選項來傳遞。

參數	必要	預設	說明
secret.region	否	N/A	搜尋`secret.id` 值的主要 AWS 區域，例如美國東部（維吉尼亞北部）。 如果未指定此區域，連接器會嘗試使用預設憑證提供者鏈結來解析 `secret.id` 的區域。在某些情況下，例如您使用外部的連接器的話，連接器便找不到該區域。建議在以下情況使用此設定： 1) 連接器在外部執行時 AWS，因為自動區域探索會失敗，並防止使用 Redshift 進行身分驗證 當連接器在與 `secret.id` 不同的區域中執行時，因為其會改善連接器對於機密的存取效能。
secret.vpcEndpointUrl	否	N/A	在覆寫預設憑證提供者鏈結時， AWS Secrets Manager 的 PrivateLink DNS 端點 URL。

參數	必要	預設	說明
secret.vpcEndpoint Region	否	N/A	在覆寫預設憑證提供者鏈結時， AWS Secrets Manager 的 PrivateLink DNS 端點區域。
jdbc.*	否	N/A	要傳遞給基礎 JDBC 驅動程式的其他參數，其中萬用字元是 JDBC 參數的名稱，例如 jdbc.ssl。請注意，在傳遞給 JDBC 驅動程序之前，會先刪除 `jdbc` 字首。若要查看 Redshift JDBC 驅動器的所有可能選項，請參閱 JDBC 驅動器 2.x 版組態的選項。

參數	必要	預設	說明
label	否	" "	使用連接器執行查詢時，要包含在查詢群組集內的識別碼。必須在 100 個字元以下，且所有字元都必須是有效的 unicodeIdentifierParts 。如果您的識別碼有超過 100 個字元，系統會移除超過的部分。使用連接器執行查詢時，查詢群組會設定為 JSON 格式字串，例如 `{"spark-redshift-connector":{"svc":"","ver":"5.1.0-amzn-1-spark_3.3","op":"Read","lbl":""}}`) 。此選項會取代 lbl 索引鍵的值。

> **Note**
>
> 確認：本文件包含由 Apache Software Foundation 根據 Apache 2.0 授權所開發的範例程式碼和語言。

支援的資料類型

Spark 連接器支援 Amazon Redshift 中的下列資料類型。如需 Amazon Redshift 中支援的資料類型完整清單，請參閱資料類型。如果資料類型不在下表中，則表示 Spark 連接器不支援該資料類型。

資料類型	別名
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	不含時區的時間戳記
TIMESTAMPTZ	含時區的時間戳記
SUPER	
TIME	不含時區的時間
TIMETZ	含時區的時間
VARBYTE	VARBINARY、BINARY VARYING

複雜資料類型

您可以使用 Spark 連接器在 Redshift SUPER 資料類型資料欄中讀取和寫入 Spark 複雜資料類型 (例如 ArrayType、MapType 和 StructType)。如果您在讀取操作期間提供結構描述，資料欄中的資料會轉換為其在 Spark 中對應的複雜類型，包括任何巢狀類型。此外，如果已啟用 autopushdown，則巢狀屬性、映射值和陣列索引的投影會向下推送至 Redshift，在只存取一部分資料時，就不再需要卸載整個巢狀資料結構。

當您從連接器寫入 DataFrames 時，任何類型為 MapType (使用StringType)、StructType、或 ArrayType 的資料欄都會寫入到 Redshift SUPER 資料類型資料欄。在撰寫這些巢狀資料結構時，tempformat 參數的類型必須是 CSV、CSV GZIP 或 PARQUET。使用 AVRO 會導致例外狀況。撰寫具有 StringType 以外索引鍵類型的 MapType 資料結構也會導致例外狀況。

StructType

下列範例會示範如何建立具有 SUPER 資料類型且包含以下結構的資料表

```
create table contains_super (a super);
```

然後，您可以使用連接器，在使用如下列範例中的結構描述的資料表中，從 SUPER 資料欄 a 查詢 StringType 欄位 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")
```

下列範例會示範如何將結構寫入到資料欄 a。

```
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

如果您偏好使用 MapType 來表示資料，則可以在結構描述中使用 MapType 資料結構，並擷取映射中與索引鍵對應的值。請注意，MapType 資料結構中的所有索引鍵都必須是字串類型，並且所有值都必須是相同類型，例如 int。

下列範例會示範如何取得資料欄 a 中索引鍵 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

如果資料欄包含陣列而不是結構，您可以使用連接器來查詢陣列中的第一個元素。

```
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

搭配 Spark 連接器使用複雜資料類型時有下列限制：

- 所有巢狀的結構欄位名稱和映射索引鍵都必須小寫。如果要查詢包含大寫字母的複雜欄位名稱，因應措施是您可以嘗試省略結構描述，並使用 `from_json` Spark 函數在本機轉換傳回的字串。
- 讀取或寫入操作中使用的任何映射欄位都必須只有 `StringType` 索引鍵。
- 只有 `CSV`、`CSV GZIP` 和 `PARQUET` 才是支援將複雜類型寫入到 Redshift 的 tempformat 值。嘗試使用 `AVRO` 會擲回例外狀況。

設定 Amazon Redshift ODBC 驅動程式 2.x 版連線

您可以從許多第三方 SQL 用戶端工具和應用程式使用 ODBC 連線來連線至 Amazon Redshift 叢集。如果用戶端工具支援 JDBC，您可以選擇使用這種連線，而非 ODBC，因為 JDBC 的設定簡單。不過，如果您的用戶端工具不支援 JDBC，您可以按照本節中的步驟，在用戶端電腦或 Amazon EC2 執行個體上設定 ODBC 連線。

Amazon Redshift 為 Linux、Windows 和 Mac 作業系統提供 64 位元 ODBC 驅動程式；32 位元 ODBC 驅動程式已停用。除了緊急的安全修補程式外，不會再針對 32 位元 ODBC 驅動程式發佈其他更新。

如需 ODBC 驅動程式變更的最新資訊，請參閱變更日誌。

主題

- 取得 ODBC URL

- 在 Microsoft Windows 上使用 Amazon Redshift ODBC 驅動器
- 在 Linux 上使用 Amazon Redshift ODBC 驅動器
- 在 Apple macOS 上使用 Amazon Redshift ODBC 驅動程式
- 身分驗證方法
- 資料類型轉換
- ODBC 驅動器選項
- 先前的 ODBC 驅動程式版本
- ODBC 驅動程式的 2.x 和 1.x 版本之間的差異

取得 ODBC URL

Amazon Redshift 會在 Amazon Redshift 主控台中顯示叢集的 ODBC URL。此 URL 包含要在用戶端電腦和資料庫之間設定連線所需的資訊。

ODBC URL 採用下列格式：

```
Driver={driver}; Server=endpoint_host; Database=database_name; UID=user_name;
 PWD=password; Port=port_number
```

前面格式的欄位具有下列值：

欄位	Value
Driver	要使用的 64 位元 ODBC 驅動程式的名稱：Amazon Redshift ODBC 驅動程式 (x64)。
Server	Amazon Redshift 叢集的端點主機。
Database	您為叢集建立的資料庫。
UID	有許可而可連線至資料庫之資料庫使用者帳戶的使用者名稱。雖然這個值是資料庫層級的許可而非叢集層級的許可，但您可以使用您在啟動叢集時所設定的 Redshift 管理員使用者帳戶。
PWD	用來連線至資料庫之資料庫使用者帳戶的密碼。

欄位	Value
Port	您啟動叢集時指定的連接埠號碼。如果有防火牆，請確定此連接埠已開放給您使用。

以下是 ODBC URL 範例：

```
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
```

如需有關可在何處找到 ODBC URL 的資訊，請參閱尋找叢集連線字串。

在 Microsoft Windows 上使用 Amazon Redshift ODBC 驅動器

您必須在存取 Amazon Redshift 資料倉儲的用戶端電腦上安裝 Amazon Redshift ODBC 驅動程式。針對安裝驅動程式的每一台電腦，有下列最低要求：

- 機器上的管理員權限。
- 機器符合下列系統要求：
  - 下列其中一個操作系統：
    - Windows 10 或 8.1。
    - Windows Server 2025、2022、2019、2016 或 2012。
  - 100 MB 的可用磁碟空間。
  - 已安裝 Visual C++ Redistributable for Visual Studio 2015，適用於 64 位元。您可以在 Microsoft 網站上的下載適用於 Visual Studio 2022 的 Visual C++ 可轉散發套件下載安裝套件。

下載並安裝 Amazon Redshift ODBC 驅動器

使用下列程序來下載並安裝適用於 Windows 作業系統的 Amazon Redshift ODBC 驅動程式。僅在您執行的第三方應用程式經認證可與 Amazon Redshift 搭配使用，且該應用程式需要特定驅動程式時，才使用不同的驅動程式。

若要下載並安裝 ODBC 驅動程式：

1. 下載下列驅動程式：64 位元 ODBC 驅動程式 2.2.4.0 版

此驅動程式的名稱是 Amazon Redshift ODBC 驅動程式 (x64)。

2. 檢閱 Amazon Redshift ODBC 驅動程式 2.x 版授權。
3. 按兩下 .msi 檔案，然後遵循精靈中的步驟來安裝驅動程式。

建立 ODBC 連線的系統 DSN 項目

在下載並安裝 ODBC 驅動程式之後，請將資料來源名稱 (DSN) 項目新增至用戶端電腦或 Amazon EC2 執行個體。SQL 用戶端工具可以使用此資料來源以連線至 Amazon Redshift 資料庫。

建議您建立系統 DSN，而不是使用者 DSN。部分應用程式會使用不同資料庫使用者帳戶載入資料，而且可能無法偵測到在其他資料庫使用者帳戶下建立的使用者 DSN。

> **Note**
>
> 對於使用 AWS Identity and Access Management (IAM) 登入資料或身分提供者 (IdP) 登入資料的身分驗證，需要額外的步驟。如需詳細資訊，請參閱設定 JDBC 或 ODBC 連線以使用 IAM 憑證。

若要建立 ODBC 連線的系統 DSN 項目：

1. 在開始功能表中，輸入「ODBC 資料來源」。選擇 ODBC 資料來源。

   確定您選擇的 ODBC 資料來源管理員具有與您要用來連線到 Amazon Redshift 的用戶端應用程式相同的位元。
2. 在 ODBC 資料來源管理員中，選擇驅動程式索引標籤並尋找下列驅動程式資料夾：Amazon Redshift ODBC 驅動程式 (x64)。
3. 選擇系統 DSN索引標籤，為電腦上的所有使用者設定驅動程式，或選擇使用者 DSN索引標籤，只為您的資料庫使用者帳戶設定驅動程式。
4. 選擇新增。Create New Data Source (建立新的資料來源) 視窗會開啟。
5. 選擇 Amazon Redshift ODBC 驅動程式 (x64)，然後選擇完成。Amazon Redshift ODBC 驅動程式 DSN 設定視窗會開啟。
6. 在連線設定區段下，輸入下列資訊：
   - 資料來源名稱

輸入資料來源的名稱。例如，如果您遵循了《Amazon Redshift 入門指南》，則可能會輸入 `exampleclusterdsn`，以便輕鬆記住與此 DSN 建立關聯的叢集。

- Server

  指定 Amazon Redshift 叢集的端點主機。您可以在 Amazon Redshift 主控台的叢集詳細資訊頁面上找到此資訊。如需詳細資訊，請參閱在 Amazon Redshift 中設定連線。

- 站點

  輸入資料庫使用的連接埠號碼。根據您在建立、修改或遷移叢集時所選取的連接埠而定，允許存取選取的連接埠。

- 資料庫

  輸入 Amazon Redshift 資料庫的名稱。如果您在未指定資料庫名稱的情況下啟動了叢集，請輸入 `dev`。否則，請使用您在啟動過程選擇的名稱。如果您遵循了《Amazon Redshift 入門指南》，請輸入 `dev`。

7. 在身分驗證區段下，指定要設定標準或 IAM 身分驗證的組態選項。
8. 選擇 SSL 選項，並指定下列項目的值：

   - 身分驗證方式

     選擇用於處理 Secure Sockets Layer (SSL) 的模式。在測試環境中，您可以使用 `prefer`。不過，若為生產環境且需要交換安全資料時，請使用 `verify-ca` 或 `verify-full`。

   - 最低 TLS

     或者，選擇驅動器允許資料存放區用於加密連線的最低 TLS/SSL 版本。例如，如果您指定 TLS 1.2，則無法使用 TLS 1.1 來加密連線。預設版本是 TLS 1.2。

9. 在代理索引標籤中，指定任何代理連線設定。
10. 在游標索引標籤中，指定有關如何將查詢結果傳回給 SQL 用戶端工具或應用程式的選項。
11. 在進階選項中，指定 `logLevel`、`logPath`、`compression` 和其他選項的值。
12. 選擇測試。如果用戶端電腦可以連線至 Amazon Redshift 資料庫，則會出現下列訊息：連線成功。如果用戶端電腦無法連線至資料庫，您可以透過產生日誌檔案並聯絡 AWS 支援來疑難排解可能的問題。如需有關產生日誌的資訊，請參閱 (LINK)。
13. 選擇確定。

在 Linux 上使用 Amazon Redshift ODBC 驅動器

您必須在存取 Amazon Redshift 資料倉儲的用戶端電腦上安裝 Amazon Redshift ODBC 驅動程式。針對安裝驅動程式的每一台電腦，有下列最低要求：

- 機器上的根存取。
- 下列其中一個發行版本：
  - Amazon Linux 2
  - Amazon Linux 2023
  - Red Hat® Enterprise Linux® (RHEL) 8 或更新版本。
  - CentOS 8 或更新版本
  - Debian 11 或更新版本
- 150 MB 的可用磁碟空間
- unixODBC 2.2.14 或更新版本。
- glibc 2.26 或更新版本。

下載並安裝 Amazon Redshift ODBC 驅動器

若要下載並安裝適用於 Linux 的 Amazon Redshift ODBC 驅動程式 2.x 版：

1. 下載下列驅動程式：
   - x86 64 位元 RPM 驅動程式版本 2.2.4.0
     - Linux 成品簽章金鑰：金鑰
     - Linux 成品簽署雜湊：雜湊
   - ARM 64 位元 RPM 驅動程式 2.2.4.0 版
     - Linux 成品簽章金鑰：金鑰
     - Linux 成品簽署雜湊：雜湊
   - x86 64 位元 DEB 驅動程式 2.2.4.0 版
     - Linux 成品簽章金鑰：金鑰
     - Linux 成品簽署雜湊：雜湊
   - ARM 64 位元 DEB 驅動程式 2.2.4.0 版
     - Linux 成品簽章金鑰：金鑰
     - Linux 成品簽署雜湊：雜湊

> **Note**
>
> 32 位元的 ODBC 驅動程式已停用。除了緊急的安全修補程式外，不會再發佈其他更新。

2. 安裝之前，請確認 已 AWS 簽署驅動程式套件。下載套件、其簽章金鑰 （憑證） 及其已簽章的雜湊 （已移除的簽章） 之後，請執行下列命令：

```
openssl x509 -in SignatureKeyFile -pubkey -noout > redshift-odbc-pubkey.pem
openssl dgst -sha256 -verify redshift-odbc-pubkey.pem \
    -signature SignedHashFile \
    PackageFile
```

如果套件是真實的， 會openssl列印 `Verified OK`。如果輸出是其他任何項目，請勿安裝套件。重新下載檔案或聯絡 AWS Support。

例如，若要驗證 x86 64 位元 RPM：

```
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. 移至您下載套件的位置，然後執行下列其中一個命令。使用與您的 Linux 發行版本相應的指令。

在 RHEL 和 CentOS 作業系統 (RPM) 上執行下列命令：

```
sudo yum --nogpgcheck localinstall RPMFileName
```

將 `RPMFileName` 取代為 RPM 套件檔案名稱。例如，下列命令示範安裝 64 位元驅動程式：

```
sudo yum --nogpgcheck localinstall AmazonRedshiftODBC-64-bit-2.x.xx.xxxx.x86_64.rpm
```

在 Debian 和 Ubuntu 作業系統 (DEB) 上，導覽至包含下載的 .deb 檔案的目錄，然後執行下列命令：

```
sudo apt install ./DEBFileName
```

將 取代DEBFileName為 DEB 套件檔案名稱。例如：

```
sudo apt install ./AmazonRedshiftODBC-64-bit-2.x.xx.xxxx.amd64.deb
```

使用 ODBC 驅動器管理員設定 ODBC 驅動器

在 Linux 上，您使用的是 ODBC 驅動程式管理員來設定 ODBC 連線設定。ODBC 驅動程式管理員使用組態檔案來定義及設定 ODBC 資料來源和驅動程式。您使用的 ODBC 驅動程式管理員視您使用的作業系統而定。

使用 unixODBC 驅動程式管理員來設定 ODBC 驅動程式

需要下列檔案才能設定 Amazon Redshift ODBC 驅動程式：

- amazon.redshiftodbc.ini
- odbc.ini
- odbcinst.ini

如果是安裝到預設位置，則 amazon.redshiftodbc.ini 組態檔案位於 /opt/amazon/redshiftodbcx64。

此外，在 /opt/amazon/redshiftodbcx64 下，您可以找到 odbc.ini 和 odbcinst.ini 範例檔案。您可以使用這些檔案做為設定 Amazon Redshift ODBC 驅動程式和資料來源名稱 (DSN) 的範例。

不建議使用 Amazon Redshift ODBC 驅動程式安裝目錄來存放組態檔案。所安裝目錄中的範例檔案僅供當作範例。如果您後來重新安裝 Amazon Redshift ODBC 驅動程式，或升級為更新版本，安裝目錄會遭到覆寫。您會遺失對安裝目錄中的檔案所做的任何變更。

若要避免這種情況，請將 amazon.redshiftodbc.ini 檔案複製到安裝目錄以外的目錄。如果您將此檔案複製到使用者的主目錄，請在檔案名稱開頭加上一點 (.)，將它變成隱藏檔。

對於 odbc.ini 和 odbcinst.ini 檔案，請使用位於使用者主目錄中的組態檔案，或在另一個目錄中建立新版本。根據預設，您的 Linux 作業系統應該在使用者的主目錄 (/home/$USER 或 ~/.) 中具

有 odbc.ini 檔案和 odbcinst.ini 檔案。這些預設檔案是隱藏檔，每個檔案名稱前面以點表示。僅當使用 -a 旗標來列出目錄內容時，才會顯示這些檔案。

無論對 odbc.ini 和 odbcinst.ini 檔案選擇何種選項，都會修改這些檔案來新增驅動程式和 DSN 組態資訊。如果建立新檔案，則還需要設定環境變數來指定這些組態檔案所在的位置。

根據預設，ODBC 驅動程式管理員會設定為使用隱藏版本的 odbc.ini 和 odbcinst.ini 組態檔 (命名為 .odbc.ini 和 .odbcinst.ini)，這些組態檔位於主目錄中。它們也會設定為使用驅動程式安裝目錄中的 amazon.redshiftodbc.ini 檔案。如果您將這些組態檔儲存在其他位置，請設定如下所述的環境變數，以便驅動程式管理員可以找到這些檔案。

如果您使用 unixODBC，請執行以下操作：

- 將 ODBCINI 設定為 odbc.ini 檔案的完整路徑和檔案名稱。
- 將 ODBCSYSINI 設定為包含 odbcinst.ini 檔案之目錄的完整路徑。
- 將 AMAZONREDSHIFTODBCINI 設定為 amazon.redshiftodbc.ini 檔案的完整路徑和檔案名稱。

以下是設定上述值的範例：

```
export ODBCINI=/usr/local/odbc/odbc.ini
export ODBCSYSINI=/usr/local/odbc
export AMAZONREDSHIFTODBCINI=/etc/amazon.redshiftodbc.ini
```

在 Linux 上使用資料來源名稱 (DSN) 來設定連線

在使用資料來源名稱 (DSN) 連線至資料存放區時，請設定 odbc.ini 檔案來定義資料來源名稱 (DSN)。在 odbc.ini 檔案中設定屬性，以建立 DSN 來指定資料倉庫的連線資訊。

在 Linux 作業系統上使用下列格式：

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file
Host=cluster_endpoint
Port=port_number
```

```
Database=database_name
locale=locale
```

以下範例示範 Linux 作業系統上使用 64 位元 ODBC 驅動程式的 odbc.ini 組態。

```
[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 上設定不使用 DSN 的連線

若要透過沒有 DSN 的連線連接至您的資料存放區，請在 odbcinst.ini 檔案中定義驅動程式。然後，在應用程式中提供無 DSN 的連線字串。

在 Linux 作業系統上使用下列格式：

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/driver_file

...
```

以下範例示範 Linux 作業系統上使用 64 位元 ODBC 驅動程式的 odbcinst.ini 組態。

```
[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 驅動程式

您必須在存取 Amazon Redshift 資料倉儲的用戶端電腦上安裝 Amazon Redshift ODBC 驅動程式。對於您安裝驅動程式的每個電腦，下列最低需求如下：

- 機器上的根存取。
- Apple macOS 系統需求：
  - 需要 64 位元版本的 Apple macOS 11.7 版或更新版本（例如 Apple macOS Big Sur、Monterey、Ventura 或更新版本）。Redshift ODBC 驅動程式僅支援 64 位元用戶端應用程式。
  - 150 MB 的可用磁碟空間
  - 驅動程式支援使用 iODBC 3.52.9+ 或 unixODBC 2.3.7+ 建置的應用程式。

下載並安裝 Amazon Redshift ODBC 驅動器

使用下列程序在 Apple macOS 下載並安裝 Amazon Redshift ODBC 驅動程式。僅在您執行的第三方應用程式經認證可與 Amazon Redshift 搭配使用，且該應用程式需要特定驅動程式時，才使用不同的驅動程式。

若要下載並安裝 ODBC 驅動程式：

1. 下載下列驅動程式：64 位元 ODBC 驅動程式 2.2.4.0 版

   x86_64 和 arm64 架構都支援此驅動程式。此驅動程式的名稱是 Amazon Redshift ODBC 驅動程式 (x64)。
2. 檢閱 Amazon Redshift ODBC 驅動程式 2.x 版授權。
3. 按兩下 .pkg 檔案，然後依照精靈中的步驟安裝驅動程式。或者，執行下列命令：

   ```
   sudo installer -pkg PKGFileName -target /
   ```

   將 取代PKGFileName為 pkg 套件檔案名稱。例如，下列命令示範安裝 64 位元驅動程式：

   ```
   sudo installer -pkg ./AmazonRedshiftODBC-64-bit.X.X.XX.X.universal.pkg -target /
   ```

使用 ODBC 驅動器管理員設定 ODBC 驅動器

在 Mac 上，您可以使用 ODBC 驅動程式管理員來設定 ODBC 連線設定。ODBC 驅動程式管理員使用組態檔案來定義及設定 ODBC 資料來源和驅動程式。您使用的 ODBC 驅動程式管理員視您使用的作業系統而定。

使用 iODBC 或 unixODBC 驅動程式管理員設定 ODBC 驅動程式

需要下列檔案才能設定 Amazon Redshift ODBC 驅動程式：

- amazon.redshiftodbc.ini
- odbc.ini
- odbcinst.ini

如果是安裝到預設位置，則 amazon.redshiftodbc.ini 組態檔案位於 /opt/amazon/redshiftodbcx64。

此外，在 /opt/amazon/redshiftodbcx64 下，您可以找到 odbc.ini 和 odbcinst.ini 範例檔案。您可以使用這些檔案做為設定 Amazon Redshift ODBC 驅動程式和資料來源名稱 (DSN) 的範例。所安裝目錄中的範例檔案僅供當作範例。

不建議使用 Amazon Redshift ODBC 驅動程式安裝目錄來存放組態檔案。如果您後來重新安裝 Amazon Redshift ODBC 驅動程式，或升級為更新版本，安裝目錄會遭到覆寫。您會遺失對安裝目錄中的檔案所做的任何變更。

若要避免這種情況，請將 odbc.iniodbcinst.ini和 amazon.redshiftodbc.ini 檔案複製到安裝目錄以外的目錄。如果您將這些檔案複製到使用者的主目錄，請在這些檔案名稱的開頭新增句點 (.)，使其成為隱藏檔案。

修改檔案以新增 DSN 組態資訊。建立新檔案時，您也需要設定環境變數，以指定這些組態檔案的位置。

以下是設定環境變數的範例：

```
export ODBCINI=/Library/ODBC/odbc.ini
export ODBCSYSINI=/Library/ODBC
export ODBCINSTINI=${ODBCSYSINI}/odbcinst.ini
```

對於命令列應用程式：將匯出命令新增至 shell 啟動檔案 （ 例如 ~/.bash_profile或 ~/.zshrc)。

如需驅動程式管理員的支援版本，請參閱此處

在 Apple macOS 上使用資料來源名稱 (DSN) 設定連線

在使用資料來源名稱 (DSN) 連線至資料存放區時，請設定 odbc.ini 檔案來定義資料來源名稱 (DSN)。將 odbc.ini 檔案中的屬性設定為建立 DSN，以指定 Redshift 資料倉儲的連線資訊。

在 Apple macOS 上，使用下列格式：

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file
Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

下列範例顯示 Apple macOS 上odbc.ini具有 64 位元 ODBC 驅動程式的 組態。

```
[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 上設定沒有 DSN 的連線

若要透過沒有 DSN 的連線連線至 Redshift 資料倉儲，請在 odbcinst.ini 檔案中定義驅動程式。然後，在應用程式中提供無 DSN 的連線字串。

在 Apple macOS 上，使用下列格式：

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
```

```
Description=driver_description
Driver=path/driver_file

...
```

下列範例顯示 Apple macOS 上odbcinst.ini具有 64 位元 ODBC 驅動程式的 組態。

```
[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
```

身分驗證方法

為了防止有人未經授權就存取資料，Amazon Redshift 資料存放區會要求所有連線使用使用者憑證進行身分驗證。

下表說明可用來連線至 Amazon Redshift ODBC 驅動程式 2.x 版之每種身分驗證方法的必要和選用連線選項：

身分驗證方法	必要	選用
標準	• 主機 • 站點 • 資料庫 • UID • 密碼	
IAM 設定檔	• 主機 • 站點 • 資料庫 • IAM • 設定檔	• ClusterID (ClusterID) • 區域 • AutoCreate • EndpointURL • StsEndpointURL • InstanceProfile

身分驗證方法	必要	選用
		Note 如果 ClusterID 和 Region 未單獨設定，則必須在 Host 中加以設定。
IAM 登入資料	• 主機 • 站點 • 資料庫 • IAM • AccessKeyID • SecretAccessKey	• ClusterID (ClusterID) • 區域 • AutoCreate • EndpointURL • StsEndpointURL • SessionToken • UID **Note** 如果 ClusterID 和 Region 未單獨設定，則必須在 Host 中加以設定。

身分驗證方法	必要	選用
AD FS	• 主機 • 站點 • 資料庫 • IAM • plugin_name • UID • 密碼 • IdP_Host • IdP_Port	• ClusterID (ClusterID) • 區域 • AutoCreate • EndpointUrl • StsEndpointUrl • Preferred_Role • loginToRp • SSL_Insecure **Note** 如果 ClusterID 和 Region 未單獨設定，則必須在 Host 中加以設定。
Azure AD	• 主機 • 站點 • 資料庫 • IAM • plugin_name • UID • 密碼 • IdP_Tenant • Client_ID • Client_Secret	• ClusterID (ClusterID) • 區域 • AutoCreate • EndpointUrl • StsEndpointUrl • Preferred_Role • dbgroups_filter **Note** 如果 ClusterID 和 Region 未單獨設定，則必須在 Host 中加以設定。

身分驗證方法	必要	選用
JWT	• 主機 • 站點 • 資料庫 • IAM • plugin_name • web_identity_token	• provider_name
Okta	• 主機 • 站點 • 資料庫 • IAM • plugin_name • UID • 密碼 • IdP_Host • App_Name • App_ID	• ClusterID (ClusterID) • 區域 • AutoCreate • EndpointUrl • StsEndpointUrl • Preferred_Role **Note** 如果 ClusterID 和 Region 未單獨設定，則必須在 Host 中加以設定。

身分驗證方法	必要	選用
Ping Federate	• 主機 • 站點 • 資料庫 • IAM • plugin_name • UID • 密碼 • IdP_Host • IdP_Port	• ClusterID (ClusterID) • 區域 • AutoCreate • EndpointUrl • StsEndpointUrl • Preferred_Role • SSL_Insecure • partner_spid **Note** 如果 ClusterID 和 Region 未單獨設定，則必須在 Host 中加以設定。

身分驗證方法	必要	選用
瀏覽器 Azure AD	• 主機 • 站點 • 資料庫 • IAM • plugin_name • IdP_Tenant • Client_ID • UID	• ClusterID (ClusterID) • 區域 • AutoCreate • EndpointUrl • StsEndpointUrl • Preferred_Role • dbgroups_filter • IdP_Response_Timeout • listen_port **Note** 如果 ClusterID 和 Region 未單獨設定，則必須在 Host 中加以設定。

身分驗證方法	必要	選用
瀏覽器 SAML	• 主機 • 站點 • 資料庫 • IAM • plugin_name • login_url • UID	• ClusterID (ClusterID) • 區域 • AutoCreate • EndpointUrl • StsEndpointUrl • Preferred_Role • dbgroups_filter • IdP_Response_Timeout • listen_port **Note** 如果 ClusterID 和 Region 未單獨設定，則必須在 Host 中加以設定。
Auth Profile	• 主機 • 站點 • 資料庫 • AccessKeyID • SecretAccessKey	

身分驗證方法	必要	選用
瀏覽器 Azure AD OAUTH2	• 主機 • 站點 • 資料庫 • IAM • plugin_name • IdP_Tenant • Client_ID • UID	• ClusterID (ClusterID) • 區域 • EndpointUrl • IdP_Response_Timeout • listen_port • scope • provider_name **Note** 如果 ClusterID 和 Region 未單獨設定，則必須在 Host 中加以設定。
AWS IAM Identity Center	• 主機 • 資料庫 • plugin_name • idc_region • issuer_url	• idc_client_display_name • idp_response_timeout • listen_port

使用外部憑證服務

除了內建支援 AD FS、Azure AD 和 Okta 外，Windows 版本的 Amazon Redshift ODBC 驅動程式也支援其他憑證服務。驅動程式可以使用您選擇的任何 SAML 型憑證提供者外掛程式來驗證連線。

若要在 Windows 上設定外部憑證服務：

1. 建立會指定憑證提供者外掛程式和視需要指定其他身分驗證參數的 IAM 設定檔。設定檔必須採用 ASCII 編碼，且必須包含下列鍵值組，其中 `PluginPath` 是外掛程式應用程式的完整路徑：

```
plugin_name = PluginPath
```

例如：

```
plugin_name = C:\Users\kjson\myapp\CredServiceApp.exe
```

如需有關如何建立設定檔的資訊，請參閱《Amazon Redshift 叢集管理指南》中的使用組態設定檔。

2. 將驅動程式設定為使用此設定檔。驅動程式會偵測並使用設定檔中指定的驗證設定。

資料類型轉換

Amazon Redshift ODBC 驅動程式 2.x 版支援許多常見的資料格式，可在 Amazon Redshift 和 SQL 資料類型之間進行轉換。

下表列出支援的資料類型映射。

Amazon Redshift 類型	SQL 類型
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 類型	SQL 類型
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 驅動器選項

您可以使用驅動程式組態選項來控制 Amazon Redshift ODBC 驅動程式的行為。驅動程式選項不區分大小寫。

在 Microsoft Windows 中，您通常於設定資料來源名稱 (DSN) 時設定驅動程式選項。您也可以在連接時以程式設計方式在連線字串中設定驅動程式選項，或在 `HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\`*`your_DSN`* 中新增或變更登錄機碼。

在 Linux 中，您可以在 `odbc.ini` 和 `amazon.redshiftodbc.ini` 檔案中設定驅動器組態選項。在 `amazon.redshiftodbc.ini` 檔案中設定的組態選項會套用至所有連線。相反地，`odbc.ini` 檔案中設定的組態選項是針對某個連線。`odbc.ini` 中設定的組態選項優先於 `amazon.redshiftodbc.ini` 中設定的組態選項。

以下是您可以為 Amazon Redshift ODBC 2.x 版驅動程式指定之選項的描述。

主題

- AccessKeyID
- app_id
- ApplicationName
- app_name

- AuthProfile
- AuthType
- AutoCreate
- BoolsAsChar
- CaFile
- client_id
- client_ secret
- ClusterId
- 壓縮
- 資料庫
- DatabaseMetadataCurrentDbOnly
- dbgroups_filter
- 驅動程式
- DSN
- EnableTableTypes
- EndpointUrl
- 擷取
- 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
- 密碼 | PWD
- plugin_name
- Port | PortNumber
- Prefer_PQ
- preferred_role
- 設定檔
- provider_name
- ProxyHost

- ProxyPort
- ProxyPwd
- ProxyUid
- ReadOnly
- region
- SecretAccessKey
- SessionToken
- Server | HostName | Host
- ssl_insecure
- SSLMode
- StreamingCursorRows
- StsConnectionTimeout
- StsEndpointUrl
- token
- token_type
- UID | User | LogonID
- UseDeclareFetch
- UseUnicode
- web_identity_token

AccessKeyID

- 預設值 — 無
- 資料類型 — 字串

使用者或角色的 IAM 存取金鑰。如果設定此參數，則也必須指定 SecretAccessKey。

此為選用參數。

app_id

- 預設值 — 無

- 資料類型 — 字串

與您的 Amazon Redshift 應用程式相關聯、Okta 所提供的唯一 ID。

此為選用參數。

ApplicationName

- 預設值 — 無
- 資料類型 — 字串

要傳遞至 Amazon Redshift 以進行稽核的用戶端應用程式名稱。您提供的應用程式名稱會出現在 SYS_CONNECTION_LOG 資料表的 'application_name' 欄中。此名稱有助於在偵錯問題時，追蹤連線來源並進行故障診斷。

此為選用參數。

app_name

- 預設值 — 無
- 資料類型 — 字串

您用來驗證與 Amazon Redshift 之連線的 Okta 應用程式名稱。

此為選用參數。

AuthProfile

- 預設值 — 無
- 資料類型 — 字串

用來管理連線設定的身分驗證設定檔。如果設定此參數，則也必須指定 AccessKeyID 和 SecretAccessKey。

此為選用參數。

AuthType

- 預設值 — 標準

- 資料類型 — 字串

此選項指定當您使用 [Amazon Redshift ODBC 驅動程式 DSN 設定] 對話方塊設定 DSN 時，驅動程式會使用的身分驗證模式：

- 標準：使用 Amazon Redshift 使用者名稱和密碼的標準身分驗證。
- AWS 設定檔：使用設定檔進行 IAM 身分驗證。
- AWS IAM 登入資料：使用 IAM 登入資料的 IAM 身分驗證。
- 身分提供者：AD FS：使用 Active Directory Federation Services (AD FS) 進行 IAM 身分驗證。
- 身分提供者：Auth 外掛程式：一種授權外掛程式，可接受來自任何連結至 AWS IAM Identity Center 的 Web 身分提供者的 IAM Identity Center 字符或 OpenID Connect (OIDC) JSON 型身分字符 AWS (JWT)。
- 身分提供者：Azure AD：使用 Azure AD 入口網站進行 IAM 身分驗證。
- 身分提供者：JWT：使用 JSON Web 權杖 (JWT) 進行 IAM 身分驗證。
- 身分提供者：Okta：使用 Okta 進行 IAM 身分驗證。
- 身分提供者：PingFederate：使用 PingFederate 進行 IAM 身分驗證。

只有當您使用 Windows 驅動程式中的 [Amazon Redshift ODBC 驅動程式 DSN 設定] 對話方塊設定 DSN 時，才能使用此選項。當您使用連線字串或非 Windows 機器設定連線時，驅動程式會根據您指定的登入資料，自動決定是否要使用標準、 AWS 設定檔或 AWS IAM 登入資料驗證。若要使用身分提供者，您必須設定 plugin_name 屬性。

此為必要參數。

AutoCreate

- 預設值 — 0
- 資料類型 — 布林值

指定當所指定的使用者不存在時，驅動程式是否會建立新使用者的布林值。

- 1 | TRUE：如果透過 UID 指定的使用者不存在，驅動程式會建立新使用者。
- 0 | FALSE：驅動程式不會建立新使用者。如果指定的使用者不存在，則身分驗證會失敗。

此為選用參數。

BoolsAsChar

- 預設值 — 0
- 資料類型 — 布林值

指定驅動程式是否將 BOOLEAN 資料欄報告為字元類型，而非位元類型。

- 1 | TRUE：驅動程式會將 BOOLEAN 資料欄報告為 SQL_VARCHAR，資料欄大小為 1。驅動程式會以字元字串 "0" 和 "1" 傳回布林值。
- 0 | FALSE：驅動程式會將 BOOLEAN 資料欄報告為 SQL_BIT。這是預設行為。

如果您的應用程式依賴以字元資料傳回的 BOOLEAN 資料欄，請使用此選項。

此為選用參數。此選項適用於驅動程式 2.2.1 版和更新版本。

CaFile

- 預設值 — 無
- 資料類型 — 字串

用於某些 IAM 身分驗證形式之 CA 憑證檔案的檔案路徑。

這個參數只能在 Linux 上使用。

此為選用參數。

client_id

- 預設值 — 無
- 資料類型 — 字串

與 Azure AD 中 Amazon Redshift 應用程式相關聯的用戶端 ID。

如果透過 Azure AD 服務進行驗證，則需要此參數。

client_ secret

- 預設值 — 無

- 資料類型 — 字串

與 Azure AD 中 Amazon Redshift 應用程式相關聯的私密金鑰。

如果透過 Azure AD 服務進行驗證，則需要此參數。

ClusterId

- 預設值 — 無
- 資料類型 — 字串

您想要連線到之 Amazon Redshift 叢集的名稱。其會用於 IAM 身分驗證。伺服器參數中不會指定叢集 ID。

此為選用參數。

壓縮

- 預設值 - off
- 資料類型 — 字串

用於 Amazon Redshift 伺服器與用戶端或驅動程式之間的有線通訊協定通訊的壓縮方法。

您可以指定下列值：

- lz4：將用於與 Amazon Redshift 進行有線通訊協定通訊的壓縮方法設定為 `lz4`。
- zstd：將用於與 Amazon Redshift 進行有線通訊協定通訊的壓縮方法設定為 `zstd`。
- off：不要對與 Amazon Redshift 進行的有線通訊協定通訊使用壓縮。

此為選用參數。

資料庫

- 預設值 — 無
- 資料類型 — 字串

您想要存取之 Amazon Redshift 資料庫的名稱。

此為必要參數。

DatabaseMetadataCurrentDbOnly

- 預設值 — 1
- 資料類型 — 布林值

指定驅動程式是否會從多個資料庫和叢集傳回中繼資料的布林值。

- 1 | TRUE：驅動程式只會從目前的資料庫傳回中繼資料。
- 0 | FALSE。驅動程式會傳回多個 Amazon Redshift 資料庫和叢集的中繼資料。

此為選用參數。

dbgroups_filter

- 預設值 — 無
- 資料類型 — 字串

當您使用 Azure、瀏覽器 Azure 和瀏覽器 SAML 身分驗證類型時，可指定來篩選出從 Amazon Redshift 的 SAML 回應收到之 DbGroups 的規則運算式。

此為選用參數。

驅動程式

- 預設值 — Amazon Redshift ODBC 驅動程式 (x64)
- 資料類型 — 字串

驅動程式的名稱。唯一支援的值是 Amazon Redshift ODBC 驅動程式 (x64)。

如果您未設定 DSN，則需要此參數。

DSN

- 預設值 — 無
- 資料類型 — 字串

驅動程式資料來源名稱的名稱。應用程式會在 SQLDriverConnect API 中指定 DSN。

如果您未設定驅動程式，則需要此參數。

EnableTableTypes

- 預設值 — 1
- 資料類型 — 布林值

指定當應用程式呼叫 SQLTables 目錄函數時，驅動程式是否從資料來源辨識詳細的資料表類型資訊。根據預設，驅動程式會辨識詳細的資料表類型。

- 1 | TRUE：驅動程式會辨識下列資料表類型：TABLE、VIEW、 SYSTEM TABLE、 SYSTEM VIEW、EXTERNAL TABLE 和 LOCAL TEMPORARY。
- 0 | FALSE：驅動程式會將詳細資料表類型資訊標準化為一般 TABLE 和 VIEW 資料表類型。

此為選用參數。它適用於驅動程式 2.2.1 版和更新版本。

EndpointUrl

- 預設值 — 無
- 資料類型 — 字串

用來與 Amazon Redshift Coral Service 通訊以進行 IAM 身分驗證的覆寫端點。

此為選用參數。

擷取

- 預設值 – 100
- 資料類型 — 整數

啟用 Declare/Fetch 模式時，此選項會控制驅動程式每次往返擷取的資料列數，讓您調整工作負載的記憶體使用量和網路額外負荷之間的平衡。較大的值會一次擷取更多資料列，以每次擷取更高的記憶體成本減少往返伺服器的次數。較小的值會降低記憶體使用量，但會增加驅動程式返回伺服器的頻率。當 `UseDeclareFetch`啟用時，它會設定驅動程式一次傳回的資料列數。如需詳細資訊，請參閱UseDeclareFetch。

此為選用參數。

ForceLowercase

- 預設值 — 0
- 資料類型 — 布林值

指定在使用單一登入身分驗證時，驅動程式是否會將從身分提供者傳送至 Amazon Redshift 的所有 DbGroups 轉換為小寫的布林值。

- 1 | TRUE：驅動程式會將從身分提供者傳送的所有 DbGroups 轉換為小寫。
- 0 | FALSE：驅動程式不會改變 DbGroups。

此為選用參數。

group_federation

- 預設值 — 0
- 資料類型 — 布林值

此布林值指定是否在佈建叢集中使用 `getClusterCredentialsWithIAM` API 取得臨時叢集憑證。此選項可讓 IAM 使用者與佈建叢集中的 Redshift 資料庫角色整合。請注意，此選項不適用於 Redshift Serverless 命名空間。

- 1 | TRUE：驅動器使用 `getClusterCredentialsWithIAM` API 在佈建叢集中取得臨時叢集憑證。
- 0 | FALSE：驅動器使用預設 `getClusterCredentials` API 在佈建叢集中取得臨時叢集憑證。

此為選用參數。

https_proxy_host

- 預設值 — 無
- 資料類型 — 字串

您想要透過其通過 IAM 身分驗證程序之代理伺服器的主機名稱或 IP 地址。

此為選用參數。

https_proxy_password

- 預設值 — 無
- 資料類型 — 字串

您用來存取代理伺服器的密碼。其會用於 IAM 身分驗證。

此為選用參數。

https_proxy_port

- 預設值 — 無
- 資料類型 — 整數

代理伺服器用來接聽用戶端連線的連接埠號碼。其會用於 IAM 身分驗證。

此為選用參數。

https_proxy_username

- 預設值 — 無
- 資料類型 — 字串

您用來存取代理伺服器的使用者名稱。其會用於 IAM 身分驗證。

此為選用參數。

IAM

- 預設值 — 0
- 資料類型 — 布林值

指定驅動程式是否會使用 IAM 身分驗證方法來驗證連線的布林值。

- 1 | TRUE：驅動程式會使用其中一種 IAM 身分驗證方法 (使用存取金鑰和私密金鑰組、或設定檔或憑證服務)。
- 0 | FALSE。驅動程式會使用標準身分驗證 (使用資料庫使用者名稱和密碼)。

此為選用參數。

idc_client_display_name

- 預設值 - Amazon Redshift ODBC 驅動器
- 資料類型 — 字串

使用 BrowserIdcAuthPlugin 的用戶端要使用的顯示名稱。

此為選用參數。

idc_region

- 預設值 — 無
- 資料類型 — 字串

IAM Identity Center AWS 執行個體所在的 AWS 區域。

只有在 plugin_name 組態選項中使用 `BrowserIdcAuthPlugin` 進行身分驗證時，才需要此參數。

idp_host

- 預設值 — 無
- 資料類型 — 字串

您用來對 Amazon Redshift 進行身分驗證的 IdP (身分提供者) 主機。

此為選用參數。

idp_port

- 預設值 — 無
- 資料類型 — 整數

您用來對 Amazon Redshift 進行身分驗證的 IdP (身分提供者) 連接埠。根據您在建立、修改或遷移叢集時所選取的連接埠而定，允許存取選取的連接埠。

此為選用參數。

idp_response_timeout

- 預設值 — 120
- 資料類型 — 整數

透過瀏覽器外掛程式使用 SAML 或 Azure AD 服務時，驅動程式等待身分提供者所傳 SAML 回應的秒數。

此為選用參數。

idp_tenant

- 預設值 — 無
- 資料類型 — 字串

與 Amazon Redshift 應用程式相關聯的 Azure AD 租用戶 ID。

如果透過 Azure AD 服務進行驗證，則需要此參數。

idp_partition

- 預設值 — 無
- 資料類型 — 字串

指定設定身分提供者 (IdP) 的雲端分割區。這會決定驅動程式連線到哪個 IdP 身分驗證端點。

如果此參數保留空白，驅動程式會預設為商業分割區。可能值為：

- us-gov：如果您的 IdP 是在 Azure Government 中設定，請使用此值。例如，Azure AD Government 會使用端點 `login.microsoftonline.us`。
- cn：如果您的 IdP 是在中國雲端分割區中設定，請使用此值。例如，Azure AD 中國使用端點 `login.chinacloudapi.cn`。

此為選用參數。

idp_use_https_proxy

- 預設值 — 0

- 資料類型 — 布林值

指定驅動程式是否會透過代理伺服器來通過身分提供者 (IdP) 身分驗證程序的布林值。

- 1 | TRUE：驅動程式會透過代理伺服器來通過 IdP 身分驗證程序。
- 0 | FALSE。驅動程式不會透過代理伺服器來通過 IdP 身分驗證程序。

此為選用參數。

InstanceProfile

- 預設值 — 0
- 資料類型 — 布林值

指定驅動程式在設定為使用設定檔進行身分驗證時，是否會使用 Amazon EC2 執行個體設定檔的布林值。

- 1 | TRUE：驅動程式會使用 Amazon EC2 執行個體設定檔。
- 0 | FALSE。驅動程式會改用 [設定檔名稱] 選項 (設定檔) 所指定的鏈結角色設定檔。

此為選用參數。

issuer_url

- 預設值 — 無
- 資料類型 — 字串

指向 AWS IAM Identity Center 伺服器的執行個體端點。

只有在 plugin_name 組態選項中使用 `BrowserIdcAuthPlugin` 進行身分驗證時，才需要此參數。

KeepAlive

- 預設值 — 1
- 資料類型 — 布林值

指定驅動程式是否會使用 TCP 保持連線來防止連線逾時的布林值。

- 1 | TRUE：驅動程式會使用 TCP 保持連線來防止連線逾時。
- 0 | FALSE。驅動程式不會使用 TCP 保持連線。

此為選用參數。

KeepAliveCount

- 預設值 — 0
- 資料類型 — 整數

在認為連線已中斷之前可遺失的 TCP 保持連線封包數。此參數設為 0 時，驅動程式會使用此設定的系統預設值。

此為選用參數。

KeepAliveInterval

- 預設值 — 0
- 資料類型 — 整數

每一次 TCP 保持連線重新傳輸之間的秒數。此參數設為 0 時，驅動程式會使用此設定的系統預設值。

此為選用參數。

KeepAliveIdle

- 預設值 — 0
- 資料類型 — 整數

驅動程式傳送 TCP 保持連線封包之前的閒置秒數。此參數設為 0 時，驅動程式會使用此設定的系統預設值。

此為選用參數。

listen_port

- 預設值 — 7890
- 資料類型 — 整數

透過瀏覽器外掛程式使用 SAML、Azure AD 或 IAM Identity Center AWS 服務時，驅動程式用來從身分提供者或授權碼接收 SAML 回應的連接埠。

此為選用參數。

login_url

- 預設值 — 無
- 資料類型 — 字串

使用一般瀏覽器 SAML 外掛程式時，身分提供者網站上資源的 URL。

如果透過瀏覽器外掛程式向 SAML 或 Azure AD 服務進行驗證，則需要此參數。

loginToRp

- 預設值 — urn:amazon:webservices
- 資料類型 — 字串

您要用於 AD FS 身分驗證類型的依賴方信任。

此為選用字串。

LogLevel

- 預設值 — 0
- 資料類型 — 整數

> **Note**
>
> `LogLevel` 和 `LogPath` 是全域設定。驅動程式會依下列優先順序讀取：
>
> 1. 連線字串 (`LogLevel=6;LogPath=/path;`) — 最高優先順序，在連線時間覆寫所有其他設定。
> 2. 在 Windows 上：DSN 設定對話方塊 （進階索引標籤）。
>
>   在 Linux 和 macOS 上： `odbc.ini` 檔案中的 `[ODBC]`區段。
> 3. 檔案中的 **`[DRIVER]`**區段 `amazon.redshiftodbc.ini` — 如果未從上述來源啟用記錄，則用作備用。

在個別 DSN 區段 （例如 [MyDSN]) LogPath中設定 LogLevel或 沒有效果。這些選項不會從每個 DSN 區段讀取。

使用此屬性可在驅動程式中啟用或停用日誌記錄，以及指定日誌檔案中包含的詳細資訊數量。日誌記錄建議只啟用到足以擷取到問題就好，因為日誌記錄會降低效能，而且可能會耗用大量磁碟空間。

將此屬性設為以下其中一個值：

- 0：OFF。停用所有日誌記錄功能。
- 1：嚴重。保留為嚴重錯誤。
- 2：錯誤。記錄錯誤事件，例如連線失敗和身分驗證錯誤。
- 3：WARN。除了錯誤之外，還會記錄警告和非預期條件。
- 4：INFO。記錄一般進度資訊，例如連線狀態和伺服器參數。
- 5：DEBUG。記錄所有驅動程式活動，包括具有引數和傳回值的 ODBC API 函數呼叫。
- 6：TRACE。記錄所有驅動程式活動，包括其他低階通訊協定和內部狀態詳細資訊。

Important

日誌層級 5 (DEBUG) 或更高版本會擷取所有驅動程式活動，其中可能包含敏感資訊，例如連線憑證和查詢資料。僅暫時啟用這些日誌層級以擷取問題，並確保在與支援或任何第三方共用日誌之前，所有敏感資訊都已修訂。

在啟用日誌記錄功能時，驅動程式會在您於 LogPath 屬性中指定的位置產生下列日誌檔案：

- 記錄在連線交握期間發生之驅動程式活動的 `redshift_odbc.log.1` 檔案。
- 在與資料庫建立連線後所有驅動程式活動的 `redshift_odbc.log` 檔案。

此為選用參數。

LogPath

- 預設值 — 作業系統特定的 TEMP 目錄

- 資料類型 — 字串

> **Note**
>
> LogLevel 和 LogPath 是全域設定。驅動程式會依下列優先順序讀取：
>
> 1. 連線字串 (LogLevel=6;LogPath=/path;) — 最高優先順序，在連線時間覆寫所有其他設定。
> 2. 在 Windows 上：DSN 設定對話方塊 （進階索引標籤）。
>
>   在 Linux 和 macOS 上： odbc.ini 檔案中的 [ODBC]區段。
> 3. amazon.redshiftodbc.ini 檔案中的 **[DRIVER]**區段 — 如果未從上述來源啟用記錄，則用作備用。
>
> 在個別 DSN 區段 （例如 [MyDSN]) LogPath中設定 LogLevel或 沒有效果。這些選項不會從每個 DSN 區段讀取。

LogLevel 大於 0 時，驅動程式用來儲存日誌檔案之資料夾的完整路徑。

此為選用參數。

LoginTimeout

- 預設值 — 0
- 資料類型 — 整數

在逾時之前，驅動程式等待與伺服器的連線建立的秒數。值 0 表示沒有逾時 （驅動程式無限期等待）。

您可以在連線字串 （使用 LoginTimeout或短別名 LT)、DSN 組態中設定此選項，或以程式設計方式使用 SQLSetConnectAttr搭配 SQL_ATTR_LOGIN_TIMEOUT。在 Windows 上，此選項也可用於 DSN 設定對話方塊 （進階索引標籤）。

此為選用參數。

MaxLongVarcharSize

- 預設值 – 65535

- 資料類型 — 整數

驅動程式針對資料欄回報的資料欄大小，由 MaxVarcharSize 重新分類為大型文字 (SQL_LONGVARCHAR)。此值會取代中繼資料回應中資料欄的實際宣告大小。

建議使用預設值 65，535，因為某些中介軟體層，例如 MSDASQL，拒絕報告大小介於 8，001 和 32，768 個位元組之間的大型文字資料欄。透過報告 65，535，驅動程式可確保接受重新分類的資料欄，而不會發生錯誤。大多數使用者不需要變更此值。

當 MaxVarcharSize 設定為 0（停用重新分類）時，此參數沒有作用。

此為選用參數。它適用於驅動程式 2.1.16 版和更新版本。

MaxVarcharSize

- 預設值 – 255
- 資料類型 — 整數

驅動程式報告為一般文字資料欄 () 的資料欄大小上限SQL_VARCHAR。驅動程式會將大於此值的任何 CHAR和 VARCHAR欄重新分類為大型文字 (SQL_LONGVARCHAR 或 UseUnicode 啟用SQL_WLONGVARCHAR時）。

此重新分類可防止應用程式或中介軟體層中無法處理超過特定大小之一般文字資料欄的錯誤。例如，透過 MSDASQL (OLE 資料庫） 存取 Amazon Redshift 的 SQL Server 連結伺服器無法處理大於大約 8，000 個位元組SQL_VARCHAR的資料欄，因此將過大的資料欄重新分類為大型文字可避免轉換失敗。

- 255 （預設）：大於 255 個字元的資料欄報告為 SQL_LONGVARCHAR。
- 0：停用重新分類。無論大小SQL_VARCHAR為何，所有資料欄都會報告為 。
- 自訂值：任何正整數。大於指定值的資料欄會報告為 SQL_LONGVARCHAR。

重新分類資料欄時，驅動程式會使用 MaxLongVarcharSize 中指定的值來報告其大小。

此為選用參數。它適用於驅動程式 2.1.16 版和更新版本。

Min_TLS

- 預設值 – 1.2
- 資料類型 — 字串

驅動程式允許資料存放區用於加密連線的 TLS/SSL 最低版本。例如，如果指定 TLS 1.2，則無法使用 TLS 1.1 來加密連線。

Min_TLS 接受以下值：

- 1.0：連線必須至少使用 TLS 1.0。
- 1.1：連線必須至少使用 TLS 1.1。
- 1.2：連線至少必須使用 TLS 1.2。

此為選用參數。

partner_spid

- 預設值 — 無
- 資料類型 — 字串

使用 PingFederate 服務驗證連線時，要使用的合作夥伴 SPID (服務提供者 ID) 值。

此為選用參數。

密碼 | PWD

- 預設值 — 無
- 資料類型 — 字串

與您在 [使用者] 欄位 (UID | User | LogonID) 中所提供資料庫使用者名稱對應的密碼。

此為選用參數。

plugin_name

- 預設值 — 無
- 資料類型 — 字串

您想要用於身分驗證的憑證提供者外掛程式名稱。

支援下列值：

- ADFS：使用 Active Directory Federation Services 進行身分驗證。

- AzureAD：使用 Microsoft Azure Active Directory (AD) Service 進行身分驗證。
- BrowserAzureAD：使用 Microsoft Azure Active Directory (AD) Service 的瀏覽器外掛程式進行身分驗證。
- BrowserIdcAuthPlugin ：使用 AWS IAM Identity Center 的授權外掛程式。
- BrowserSAML：使用 SAML 服務 (例如 Okta 或 Ping) 的瀏覽器外掛程式進行身分驗證。
- IdpTokenAuthPlugin：從連結至 AWS IAM Identity Center 的任何 Web 身分提供者接受 IAM Identity Center 字符或 OpenID Connect (OIDC) JSON 型身分字符 (JWT) AWS 的授權外掛程式。
- JWT：使用 JSON Web 權杖 (JWT) 進行身分驗證。
- Ping：使用 PingFederate 服務進行身分驗證。
- Okta：使用 Okta 服務進行身分驗證。

此為選用參數。

Port | PortNumber

- 預設值 — 5439
- 資料類型 — 整數

Amazon Redshift 伺服器用來接聽用戶端連線的 TCP 連接埠號碼。

此為選用參數。

Prefer_PQ

- 預設值 — 1
- 資料類型 — 布林值

指定驅動程式是否在 TLS 1.3 ClientHello 中公告量子後混合式金鑰交換群組。提供後量子群組且伺服器也支援後量子群組時，伺服器會選取一個。否則，交握會透明地回復到傳統金鑰交換。

- 1 | TRUE：驅動程式公告量子後混合 TLS 金鑰交換群組。
- 0 | FALSE：驅動程式不會公告後量子群組。如果網路中間設備因為無法辨識驅動程式提供的後量子金鑰交換群組，而封鎖 TLS 交握，請使用此設定。

此為選用參數。它適用於驅動程式 2.2.1 版和更新版本。

preferred_role

- 預設值 — 無
- 資料類型 — 字串

您要在連線到 Amazon Redshift 期間擔任的角色。其會用於 IAM 身分驗證。

此為選用參數。

設定檔

- 預設值 — 無
- 資料類型 — 字串

用於向 Amazon Redshift 驗證的使用者 AWS 設定檔名稱。

- 如果 Use Instance Profile 參數 (InstanceProfile 屬性) 設為 1 | TRUE，則該設定會優先，而且驅動程式會改用 Amazon EC2 執行個體設定檔。
- 包含設定檔之憑證檔案的預設位置為 `~/.aws/Credentials`。可使用 `AWS_SHARED_CREDENTIALS_FILE` 環境變數以指向不同的憑證檔案。

此為選用參數。

provider_name

- 預設值 — 無
- 資料類型 — 字串

使用者使用 CREATE IDENTITY PROVIDER 查詢所建立的身分驗證提供者。其會用於原生 Amazon Redshift 身分驗證。

此為選用參數。

ProxyHost

- 預設值 — 無
- 資料類型 — 字串

您想要透過其進行連線之代理伺服器的主機名稱或 IP 地址。

此為選用參數。

ProxyPort

- 預設值 — 無
- 資料類型 — 整數

代理伺服器用來接聽用戶端連線的連接埠號碼。

此為選用參數。

ProxyPwd

- 預設值 — 無
- 資料類型 — 字串

您用來存取代理伺服器的密碼。

此為選用參數。

ProxyUid

- 預設值 — 無
- 資料類型 — 字串

您用來存取代理伺服器的使用者名稱。

此為選用參數。

ReadOnly

- 預設值 — 0
- 資料類型 — 布林值

指定驅動程式是否處於唯讀模式的布林值。

- 1 | TRUE：連線處於唯讀模式，無法寫入到資料存放區。

- 0 | FALSE：連接不處於唯讀模式，可以寫入到資料存放區。

此為選用參數。

region

- 預設值 — 無
- 資料類型 — 字串

叢集所在的 AWS 區域。

此為選用參數。

SecretAccessKey

- 預設值 — 無
- 資料類型 — 字串

使用者或角色的 IAM 私密金鑰。如果設定此參數，則也必須指定 AccessKeyID。

此為選用參數。

SessionToken

- 預設值 — 無
- 資料類型 — 字串

與您用來驗證的 IAM 角色相關聯的臨時 IAM 工作階段權杖。

此為選用參數。

Server | HostName | Host

- 預設值 — 無
- 資料類型 — 字串

要連線到的端點伺服器。

此為必要參數。

ssl_insecure

- 預設值 — 0
- 資料類型 — 布林值

指定驅動程式是否會檢查 IdP 伺服器憑證真確性的布林值。

- 1 | TRUE：驅動程式不會檢查 IdP 伺服器憑證的真確性。
- 0 | FALSE：驅動程式會檢查 IdP 伺服器憑證的真確性。

此為選用參數。

SSLMode

- 預設值 — `verify-ca`
- 資料類型 — 字串

連線到 Amazon Redshift 時要使用的 SSL 憑證驗證模式。可能的值如下：

- `verify-full`：僅使用 SSL、受信任的憑證授權機構和符合憑證的伺服器名稱進行連線。
- `verify-ca`：僅使用 SSL 和受信任的憑證授權機構進行連線。
- `require`：僅使用 SSL 進行連線。
- `prefer`：使用 SSL (如果可用) 進行連線。否則，以不使用 SSL 的方式進行連線。
- `allow`：預設會以不使用 SSL 的方式進行連線。如果伺服器需要 SSL 連線，則使用 SSL。
- `disable`：以不使用 SSL 的方式進行連線。

> **Note**
>
> 我們建議您盡可能使用 `verify-full`進行 SSL 連線。

此為選用參數。

StreamingCursorRows

- 預設值 — 0

- 資料類型 — 整數

限制驅動程式針對僅限前向游標一次在記憶體中保留的資料列數，因此大型結果集不必一次全部載入記憶體。

將此屬性設為以下其中一個值：

- 0：驅動程式會將整個結果集載入記憶體，然後再將資料列傳回應用程式（預設）。非常大的結果集可能會耗用大量記憶體。
- 正整數：對於僅向前游標，驅動程式一次只會在記憶體中保留這麼多資料列。它會將這些資料列傳回給應用程式、釋出它們，然後在應用程式使用目前批次後擷取下一個批次。

此選項不會影響可捲動游標，其一律使用記憶體內行為。請注意，在 Windows 上，Amazon Redshift ODBC 驅動程式 DSN 設定對話方塊會以 100 的值預先填入此欄位。

串流游標模式需要僅向前游標。如果啟用用戶端游標 (CSC) 選項，則會優先考慮並停用串流游標模式。

> **Note**
>
> StreamingCursorRows 和 UseDeclareFetch 兩個批次結果，而不是將整個結果集載入記憶體。 UseDeclareFetch使用伺服器端游標，通常會提供更高的輸送量，因此當擷取效能時， 會比較喜歡它。 StreamingCursorRows 不需要交易包裝或伺服器端游標；當這些屬性比峰值速度重要時，請選擇它：具有限制記憶體佔用量的直接串流。如果啟用 UseDeclareFetch ，則會優先並強制 StreamingCursorRows 為 0。如需詳細資訊，請參閱UseDeclareFetch。

此為選用參數。

StsConnectionTimeout

- 預設值 — 0
- 資料類型 — 整數

IAM 連線的等待時間上限 (以秒為單位)。如果設定為 0 或未指定，驅動程式會等待每次 AWS STS 呼叫 60 秒。

此為選用參數。

StsEndpointUrl

- 預設值 — 無
- 資料類型 — 字串

此選項會指定用來與 AWS Security Token Service (AWS STS) 進行通訊的覆寫端點。

此為選用參數。

token

- 預設值 — 無
- 資料類型 — 字串

IAM Identity Center AWS 提供存取字符或與 IAM Identity Center 連結的 Web 身分提供者提供的 OpenID Connect (OIDC) JSON Web Token AWS (JWT)。您的應用程式必須透過使用 IAM Identity Center AWS 或與 IAM Identity Center 連結的身分提供者來驗證應用程式的使用者來產生此字符 AWS 。

此參數適用於 `IdpTokenAuthPlugin`。

token_type

- 預設值 — 無
- 資料類型 — 字串

`IdpTokenAuthPlugin` 中使用的權仗類型。

您可以指定下列值：

ACCESS_TOKEN

如果您使用 AWS IAM Identity Center 提供的存取權杖，請輸入此選項。

EXT_JWT

如果您使用與 AWS IAM Identity Center 整合的 Web 型身分提供者所提供的 OpenID Connect (OIDC) JSON Web Token (JWT)，請輸入此項。

此參數適用於 IdpTokenAuthPlugin。

UID | User | LogonID

- 預設值 — 無
- 資料類型 — 字串

您用來存取 Amazon Redshift 伺服器的使用者名稱。

若您使用資料庫身分驗證，則此為必要參數。

UseDeclareFetch

- 預設值 — 0
- 資料類型 — 布林值

當查詢傳回的結果集太大而無法保留在記憶體中時，啟用此選項可讓驅動程式以固定大小批次處理它，而不是一次緩衝整個結果，無論查詢傳回多少資料列，都保持用戶端的記憶體使用量邊界且可預測。這是一個布林值，指定驅動程式是否使用宣告/擷取模式，一次傳回一組資料列數。若要設定每個批次傳回的資料列數，請使用 Fetch選項。如需詳細資訊，請參閱擷取。

- 1 | TRUE：驅動程式使用 Declare/Fetch 模式，並一次傳回 Fetch選項指定的資料列數。
- 0 | FALSE：驅動程式一次傳回整個查詢結果。

此為選用參數。

> **Note**
>
> 如果UseDeclareFetch啟用 ，無論 DSN 或連線字串中設定的值為何，驅動程式都會停用串流游標模式並覆寫StreamingCursorRows為 0。如需詳細資訊，請參閱StreamingCursorRows。

UseUnicode

- 預設值 — 0
- 資料類型 — 布林值

布林值，指定驅動程式將 Redshift 資料傳回為 Unicode 或一般 SQL 類型。

- 1 | TRUE：驅動程式會傳回字元資料類型的廣泛 SQL 類型。
  - SQL_WCHAR 會傳回，而不是 SQL_CHAR。
  - SQL_WVARCHAR 會傳回，而不是 SQL_VARCHAR。
  - SQL_WLONGVARCHAR 會傳回，而不是 SQL_LONGVARCHAR。
- 0 | FALSE：驅動程式會傳回字元資料類型的一般 SQL 類型。
  - SQL_CHAR 會傳回，而不是 SQL_WCHAR。
  - SQL_VARCHAR 會傳回，而不是 SQL_WVARCHAR。
  - SQL_LONGVARCHAR 會傳回，而不是 SQL_WLONGVARCHAR。

此為選用參數。它適用於驅動程式 2.1.15 版和更新版本。

web_identity_token

- 預設值 — 無
- 資料類型 — 字串

身分提供者所提供的 OAUTH 權杖。其會用於 JWT 外掛程式。

如果您將 plugin_name 參數設定為 BasicJwtCredentialsProvider，則此為必要參數。

先前的 ODBC 驅動程式版本

請在您的工具需要特定版本的驅動程式時，才下載 Amazon Redshift ODBC 驅動程式 2.x 版之前的版本。

使用先前的 ODBC 驅動程式版本 (適用於 Microsoft Windows)

以下是適用於 Microsoft Windows 的舊版 Amazon Redshift ODBC 驅動程式 2.x 版：

- 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

使用先前的 ODBC 驅動程式版本 (適用於 Linux)

以下是適用於 Linux 的舊版 Amazon Redshift ODBC 驅動程式 2.x 版：

- 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 Apple macOS 的舊版 ODBC 驅動程式

以下是 Amazon Redshift ODBC 驅動程式 2.x for Apple macOS 的先前版本：

- 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 驅動程式的 2.x 和 1.x 版本之間的差異

本節說明 ODBC 1.x 驅動程式與 ODBC 2.x 驅動程式之間的差異，並提供如何遷移至 2.x 驅動程式的指引。它說明可能會影響您應用程式的變更，以及如何解決這些變更。

要解決的主要差異

下列動作可解決最常見的遷移問題。大多數應用程式只需要其中一部分。

DSN 和連線字串

- 檢閱您的 DSN 是否有可能已移除或重新命名的驅動程式選項。ODBC 2.x 驅動程式不支援某些 ODBC 1.x 驅動程式選項。其他 已重新命名。如需詳細資訊，請參閱 連線字串和 DSN 選項。
- **`UseUnicode=true`** 如果您的應用程式取決於寬字元類型代碼 (**`SQL_WVARCHAR`**、)，請設定 **`SQL_WCHAR`**。在 ODBC 2.x 驅動程式中， `UseUnicode` 預設為 `false`，報告窄字元類型。

- 將記錄設定移至 **[ODBC]**區段。在 Linux 和 macOS 上， LogLevel和 LogPath 必須在 的全域[ODBC]區段中設定odbc.ini，而不是在個別 DSN 區段中設定。

查詢和結構描述

- 包含在**SQLTables**類型篩選條件**EXTERNAL TABLE**中。在 ODBC 2.x 中，Amazon Redshift Spectrum 資料表和資料共用資料表都會報告為 EXTERNAL TABLE。如果您SQLTables依類型篩選，請新增 EXTERNAL TABLE 以繼續查看這些物件。或者，將 EnableTableTypes選項設定為 0，將詳細資料表類型資訊標準化為一般 TABLE 和 VIEW 資料表類型，如以下項目所述。
- 詳細資料表類型預設為啟用。ODBC 2.x 預設會啟用 EnableTableTypes選項。1.x 驅動程式預設會停用此選項。使用 2.x 預設時， 會SQLTables報告詳細類型，例如 SYSTEM TABLE、SYSTEM VIEW、EXTERNAL TABLE 和 LOCAL TEMPORARY。若要將每個資料表報告為一般類型 TABLE，並將每個檢視報告為 VIEW，請將 EnableTableTypes設定為 0。
- **INTERVALVARCHAR**為不支援間隔資料類型的應用程式轉換為 。有些用戶端不支援 ODBC 間隔類型。對於這些用戶端，請在查詢中轉換 欄： SELECT col::VARCHAR FROM ...

應用程式碼

- 驗證查詢逾時設定。ODBC 2.x SQL_ATTR_QUERY_TIMEOUT會根據 ODBC 規格正確強制執行。ODBC 1.x 無提示地忽略此設定。先前成功的長時間執行查詢現在可能會失敗，並出現逾時錯誤。視需要檢閱和調整您的逾時值。
- 提供data-at-execution參數的資料長度。 2.x 報告SQL_NEED_LONG_DATA_LEN為 Y(1.x 報告 N)。繫結SQL_DATA_AT_EXEC參數的應用程式現在必須在 中預先提供總資料長度StrLen_or_IndPtr。

連線字串和 DSN 選項

下表顯示已在 ODBC 2.x 驅動程式中重新命名或具有直接對等項目的 ODBC 1.x 驅動程式選項。

1.x 選項	2.x 同等	備註
MaxLongVarChar	MaxLongVarcharSize	預設從 8190 變更為 65535。
ConnectionTimeout	LoginTimeout	相同的連線逾時，已重新命名。預設為 0（無逾時）。

1.x 選項	2.x 同等	備註
VpcEndpointUrl	vpc_endpoint_url	
SSLCertPath	TrustStore 或 CaFile	用於驗證伺服器的 CA 憑證路徑。在 Windows 上，在 DSN 設定對話方塊的信任存放區欄位中設定此項目；對話方塊沒有CaFile欄位。如果兩者都已設定， TrustStore 會優先處理。

目前的 2.x 驅動程式不支援下列 1.x 選項。2.x 驅動程式會忽略它們，因此它們不會影響您的連線。從 DSN 中移除它們是選用的，但建議避免混淆。

- SingleRowMode – 若要限制用戶端記憶體，請UseDeclareFetch改用 StreamingCursorRows或 。
- UseSystemTrustStore – 不支援。在 Windows 上，1.x 驅動程式可以針對 Windows 系統憑證存放區驗證伺服器憑證。2.x 驅動程式會根據 CA 憑證檔案進行驗證：預設會使用綁定的 Amazon Redshift 根憑證，或是您在 TrustStore或 中指定的檔案CaFile。
- TextAsLongVarchar、CheckCertRevocation、EnableAwsSdkLogs、UseLogPrefix、UseMulti EnforceSingleStatement – 目前版本中沒有同等項目。Amazon Redshift 團隊正在評估未來版本中這些選項的同等項目或替代方案。

如需支援 2.x 選項的完整清單，請參閱 ODBC 驅動器選項。

資料類型變更

下表顯示在 ODBC 2.x 中變更以符合 ODBC 規格的 Amazon Redshift 資料類型映射。大多數應用程式不受影響，因為它們會依索引或名稱繫結資料欄，而不是依類型代碼繫結資料欄。

Amazon Redshift 資料類型	ODBC 1.x	ODBC 2.x
DOUBLE PRECISION	SQL_FLOAT (6)	SQL_DOUBLE (8)
INTERVAL YEAR TO MONTH	SQL_VARCHAR (12)、文字	SQL_INTERVAL_YEAR_TO_MONTH (107)，間隔結構

Amazon Redshift 資料類型	ODBC 1.x	ODBC 2.x
INTERVAL DAY TO SECOND	SQL_VARCHAR (12)、文字	SQL_INTERVAL_DAY_TO_SECOND (110)、間隔結構
VARCHAR / CHAR / TEXT	廣泛類型 (SQL_WVARCHAR 、SQL_WCHAR 、SQL_WLONGVARCHAR)；預設 UseUnicode=true	縮小類型 (SQL_VARCHAR 、SQL_CHAR、SQL_LONGVARCHAR)。UseUnicode=true 設定為還原各種類型。
GEOMETRY / GEOGRAPHY （做為 SQL_C_BINARY ）	十六進位編碼 ASCII 字串	原始二進位位元組

對於 GEOMETRY、 和 SUPER資料類型GEOGRAPHY， 傳回COLUMN_SIZE的值SQLColumns也會變更，現在會傳回 NULL，以根據 ODBC 規格指示未調整大小的資料欄。根據 配置緩衝的應用程式COLUMN_SIZE必須使用預設緩衝區大小來處理 NULL。

遷移後疑難排解

下表說明遷移後可能遇到的常見問題，以及如何解決這些問題。

徵狀	原因	處理方式
結構描述瀏覽器中看不到外部資料表	Spectrum 和資料共用資料表報告為 EXTERNAL TABLE	包含在您的SQLTables 類型篩選條件EXTERNAL TABLE中，或將 EnableTableTypes 設定為 0，將詳細資料表類型標準化為一般 TABLE 和 VIEW 類型。
間隔資料欄上的 pyodbc 錯誤	pyodbc 不支援 ODBC 間隔類型	在查詢VARCHAR中將間隔轉換為 。
字元資料會顯示為非預期的字元 (mojibake)	UseUnicode 預設變更為 false。預期使用廣字元 (UTF-16) 資料的應用程式可能會錯誤地將窄字元位元組解譯	在 DSN UseUnicode=true 中設定 ，或更新您的應用程式以繫結資料

徵狀	原因	處理方式
	為寬對，產生亂碼輸出。資料本身保持不變。	欄，SQL_C_CHAR 而非 SQL_C_WCHAR 。
長時間執行的查詢會傳回逾時錯誤	SQL_ATTR_QUERY_TIMEOUT 現已強制執行。ODBC 1.x 無提示地忽略此設定。	QueryTimeout 從 DSN 中增加或移除，或在應用程式中將 SQL_ATTR_QUERY_TIMEOUT 設定為 0。
設定SQL_ATTR_CURRENT_CATALOG 傳回 HY011	屬性無法在開啟的連線上設定。不支援在開放連線上切換資料庫。	在連線之前設定 屬性，或關閉並重新開啟目標資料庫上的連線。
連線在雙堆疊環境中懸置	<= 2.1.16 版的限制	升級至 2.1.17 或更新版本。
錯誤訊息剖析會傳回非預期的結果	訊息格式已變更；SQLSTATE 代碼不變	剖析 SQLSTATE 代碼，而非訊息文字。
Data-at-execution參數繫結的行為不同	SQL_NEED_LONG_DATA_LEN 從 FALSE 變更為 TRUE；使用 的應用程式現在SQL_DATA_AT_EXEC 必須預先提供總資料長度	在繫結SQL_DATA_AT_EXEC 參數StrLen_or_IndPtr 時，在 中設定長度值。
不會產生驅動程式日誌	LogLevel 和 LogPath 必須位於 的[ODBC]全域區段odbc.ini，而非個別 DSN 區段	將記錄設定從 DSN 移至 [ODBC]區段。

從舊版 ODBC 2.x 升級

如果您已執行較舊的 ODBC 2.x 版本，請升級至最新版本。如需所有版本的完整增強功能和錯誤修正清單，請參閱 GitHub 上的 Amazon Redshift ODBC 驅動程式變更日誌。

其他資訊

- 設定 Amazon Redshift ODBC 驅動程式 2.x 版連線
- ODBC 驅動器選項

- Amazon Redshift ODBC 驅動程式變更日誌

設定 ODBC 驅動程式 1.x 版連線

您可以從許多第三方 SQL 用戶端工具和應用程式使用 ODBC 連線來連線至 Amazon Redshift 叢集。若要這麼做，請在用戶端電腦或 Amazon EC2 執行個體上設定連線。如果用戶端工具支援 JDBC，您可能選擇使用這種連線，而非 ODBC，因為 JDBC 的設定簡單。但是，如果用戶端工具不支援 JDBC，請遵循本節的步驟來設定 ODBC 連線。

Amazon Redshift 為 Linux、Windows 和 macOS 作業系統提供 64 位元 ODBC 驅動程式。32 位元的 ODBC 驅動程式已停止使用。除了緊急的安全修補程式外，不會再發佈其他更新。

如需 ODBC 驅動程式功能和先決條件的最新資訊，請參閱 Amazon Redshift ODBC 驅動程式版本備註。

如需 Amazon Redshift ODBC 驅動程式的安裝與組態資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南。

主題

- 取得 ODBC URL
- 在 Microsoft Windows 上使用 Amazon Redshift ODBC 驅動器
- 在 Linux 上使用 Amazon Redshift ODBC 驅動器
- 在 macOS 上使用 Amazon Redshift ODBC 驅動程式
- ODBC 驅動器選項
- 先前的 ODBC 驅動程式版本

取得 ODBC URL

Amazon Redshift 會在 Amazon Redshift 主控台中顯示叢集的 ODBC URL。此 URL 包含要在用戶端電腦和資料庫之間設定連線的資訊。

ODBC URL 採用下列格式：`Driver={driver};Server=endpoint;Database=database_name;UID=user_name;PWD=pass`

前面顯示的格式欄位具有下列值。

欄位	Value
Driver	要使用的 64 位元 ODBC 驅動程式的名稱：Amazon Redshift (x64)。 32 位元 ODBC 驅動程式的名稱：Amazon Redshift (x86)。
Server	Amazon Redshift 叢集的端點。
Database	您為叢集建立的資料庫。
UID	有許可連接至資料庫之使用者帳戶的使用者名稱。雖然可使用啟動叢集時所設定的管理員使用者帳戶，但此值是資料庫許可，不是 Amazon Redshift 許可。
PWD	用於連接至資料庫之使用者帳戶的密碼。
Port	您啟動叢集時指定的連接埠號碼。如果有防火牆，請確定此連接埠已開放給您使用。

上述資料表中的欄位可包含下列特殊字元：

```
[]{}(),;?*=!@
```

如果您使用這些特殊字元，則必須用大括號括住值。例如，連線字串中的密碼值 Your;password123 會表示為 PWD={Your;password123};。

由於 Field=value 配對會以分號分隔，因此 } 與 ; 的組合中間若有任意數量的空格，將會被視為是 Field={value}; 配對的結尾。建議您避免在欄位值中使用連續的 };。例如，如果您將密碼值設定為 PWD={This is a passwor} ;d};，則密碼會是 This is a passwor} ;，因此 URL 會發生錯誤。

以下是 ODBC URL 範例。

```
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
```

如需如何取得 ODBC 連線的資訊，請參閱尋找叢集連線字串。

在 Microsoft Windows 上使用 Amazon Redshift ODBC 驅動器

請在存取 Amazon Redshift 資料倉儲的用戶端電腦上安裝 Amazon Redshift ODBC 驅動程式。您安裝驅動程式的每一台電腦必須符合最低系統需求清單：如需最低系統要求的相關資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南。

主題

- 下載並安裝 Amazon Redshift ODBC 驅動器
- 建立 ODBC 連線的系統 DSN 項目

下載並安裝 Amazon Redshift ODBC 驅動器

使用下列程序來下載適用於 Windows 作業系統的 Amazon Redshift ODBC 驅動程式。僅在您執行的第三方應用程式經認證可與 Amazon Redshift 搭配使用，且需要有特定驅動程式時，才使用不同於以下的驅動程式。

安裝 ODBC 驅動程式

1. 根據 SQL 用戶端工具或應用程式的系統架構，下載下列其中一個項目：

   - 64 位元 ODBC 驅動程式 1.6.3 版

     這個驅動程式的名稱是 Amazon Redshift (x64)。

   - 32 位元 ODBC 驅動程式版本 1.4.52

     這個驅動程式的名稱是 Amazon Redshift (x86)。32 位元的 ODBC 驅動程式已停止使用。除了緊急的安全修補程式外，不會再發佈其他更新。

   > **Note**
   >
   > 下載對應於 SQL 用戶端工具或應用程式之系統架構的 MSI 套件。例如，假設 SQL 用戶端工具是 64 位元，請安裝 64 位元驅動程式。

   然後下載並檢閱 Amazon Redshift ODBC 和 JDBC 驅動程式授權合約。

2. 按兩下 .msi 檔案，然後遵循精靈中的步驟來安裝驅動程式。

建立 ODBC 連線的系統 DSN 項目

在下載並安裝 ODBC 驅動程式之後，請將資料來源名稱 (DSN) 項目新增至用戶端電腦或 Amazon EC2 執行個體。SQL 用戶端工具會使用此資料來源以連線至 Amazon Redshift 資料庫。

建議您建立系統 DSN，而不是使用者 DSN。某些應用程式會使用不同的使用者帳戶載入資料。這些應用程式可能無法偵測在另一個使用者帳戶下建立的使用者 DSN。

> **Note**
>
> 對於使用 AWS Identity and Access Management (IAM) 登入資料或身分提供者 (IdP) 登入資料的身分驗證，需要額外的步驟。如需詳細資訊，請參閱步驟 5：設定 JDBC 或 ODBC 連線使用 IAM 登入資料。

如需如何建立系統 DSN 項目的相關資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南。

在 Windows 上建立 ODBC 連線的系統 DSN 項目

1. 在 Start (開啟) 功能表中，開啟 ODBC Data Sources (ODBC 資料來源)。

   確定您選擇的 ODBC 資料來源管理員具有與您要用來連線到 Amazon Redshift 的用戶端應用程式相同的位元。
2. 在 ODBC 資料來源管理員中，選擇驅動程式索引標籤並尋找驅動程式資料夾：

   - Amazon Redshift ODBC 驅動程式 (64 位元)
   - Amazon Redshift ODBC 驅動程式 (32 位元)
3. 選擇 System DSN (系統 DSN) 標籤，為電腦上的所有使用者設定驅動程式，或選擇 User DSN (使用者 DSN) 標籤，只為您的使用者帳戶設定驅動程式。
4. 選擇新增。Create New Data Source (建立新的資料來源) 視窗會開啟。
5. 選擇 Amazon Redshift ODBC 驅動程式，然後選擇完成。Amazon Redshift ODBC 驅動程式 DSN 設定視窗會開啟。
6. 在 Connection Settings (連線設定) 下，輸入下列資訊：

   資料來源名稱

   輸入資料來源的名稱。您可以使用稍後建立叢集連線時要用來識別資料來源的任何名稱。例如，如果您遵循了《Amazon Redshift 入門指南》，則可能會輸入 `exampleclusterdsn`，以便輕鬆記住與此 DSN 建立關聯的叢集。

Server

指定 Amazon Redshift 叢集的端點。您可以在 Amazon Redshift 主控台的叢集詳細資訊頁面上找到此資訊。如需詳細資訊，請參閱在 Amazon Redshift 中設定連線。

站點

輸入資料庫使用的連接埠號碼。請使用在叢集啟動或修改時設定給叢集使用的連接埠。

資料庫

輸入 Amazon Redshift 資料庫的名稱。如果您在未指定資料庫名稱的情況下啟動了叢集，請輸入 *dev*。否則，請使用您在啟動過程選擇的名稱。如果您遵循了《Amazon Redshift 入門指南》，請輸入 *dev*。

7. 在 Authentication (身分驗證) 下，指定要設定標準或 IAM 身分驗證的組態選項。如需不同身分驗證選項的相關資訊，請參閱《Amazon Redshift ODBC 連接器安裝和組態指南》中的「在 Windows 上設定身分驗證」。
8. 在 SSL Settings (SSL 設定) 下，指定下列這一項的值：

   SSL authentication (SSL 身分驗證)

   選擇用於處理 Secure Sockets Layer (SSL) 的模式。在測試環境中，您可以使用 `prefer`。不過，若為生產環境且需要交換安全資料時，請使用 `verify-ca` 或 `verify-full`。如需在 Windows 上使用 SSL 的相關資訊，請參閱《Amazon Redshift ODBC 連接器安裝和組態指南》中的「在 Windows 上設定 SSL 驗證」。
9. 在 Additional Options (其他選項) 下，指定如何將查詢結果傳回給 SQL 用戶端工具或應用程式的選項。如需詳細資訊，請參閱《Amazon Redshift ODBC 連接器安裝和組態指南》中的「在 Windows 上設定其他選項」。
10. 在 Logging Options (記錄選項) 中，指定記錄選項的值：如需詳細資訊，請參閱《Amazon Redshift ODBC 連接器安裝和組態指南》中的「在 Windows 上設定記錄選項」。

    然後選擇 OK (確定)。
11. 在 Data Type Options (資料類型選項) 下，指定資料類型的值。如需詳細資訊，請參閱《Amazon Redshift ODBC 連接器安裝和組態指南》中的「在 Windows 上設定資料類型選項」。

    然後選擇 OK (確定)。
12. 選擇測試。如果用戶端電腦可以連線至 Amazon Redshift 資料庫，您會看到下列訊息：連線成功。

如果用戶端電腦無法連線至資料庫，您可以對可能的問題進行疑難排解。如需詳細資訊，請參閱針對 Amazon Redshift 中的連線問題進行疑難排解。

13. 在 Windows 上設定 TCP 保持連線，以防止連線逾時。如需如何在 Windows 上設定 TCP 保持連線的相關資訊，請參閱《Amazon Redshift ODBC 連接器安裝和組態指南》。
14. 若要協助疑難排解，請設定記錄。如需如何在 Windows 上設定記錄的相關資訊，請參閱《Amazon Redshift ODBC 連接器安裝和組態指南》。

在 Linux 上使用 Amazon Redshift ODBC 驅動器

請在存取 Amazon Redshift 資料倉儲的用戶端電腦上安裝 Amazon Redshift ODBC 驅動程式。您安裝驅動程式的每一台電腦必須符合最低系統需求清單：如需最低系統要求的相關資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南。

主題

- 下載並安裝 Amazon Redshift ODBC 驅動器
- 使用 ODBC 驅動器管理員設定驅動器

下載並安裝 Amazon Redshift ODBC 驅動器

使用本節的步驟來下載 Amazon Redshift ODBC 驅動程式並安裝在 Linux 發行版本上。安裝程序會將驅動程式檔案安裝在下列目錄中：

- `/opt/amazon/redshiftodbc/lib/64` (適用於 64 位元驅動程式)
- `/opt/amazon/redshiftodbc/ErrorMessages`
- `/opt/amazon/redshiftodbc/Setup`
- `/opt/amazon/redshiftodbc/lib/32` (適用於 32 位元驅動程式)

安裝 Amazon Redshift ODBC 驅動程式

1. 根據 SQL 用戶端工具或應用程式的系統架構，下載下列其中一個項目：
   - 64 位元 RPM 驅動程式 1.6.3 版
   - 64 位元 Debian 驅動程式 1.6.3 版
   - 32 位元驅動程式版本 1.4.52

其中每一個驅動程式的名稱都是 Amazon Redshift ODBC 驅動程式。32 位元的 ODBC 驅動程式已停止使用。除了緊急的安全修補程式外，不會再發佈其他更新。

> **Note**
>
> 下載對應於 SQL 用戶端工具或應用程式之系統架構的套件。例如，假設用戶端工具是 64 位元，請安裝 64 位元驅動程式。

然後下載並檢閱 Amazon Redshift ODBC 和 JDBC 驅動程式授權合約。

2. 移至您下載套件的位置，然後執行下列其中一個命令。使用與您的 Linux 發行版本相應的指令。

   - 在 RHEL 和 CentOS 作業系統上，執行下列命令。

     ```
     yum -nogpgcheck localinstall RPMFileName
     ```

     將 *RPMFileName* 取代為 RPM 套件檔案名稱。例如，下列命令示範安裝 64 位元驅動程式。

     ```
     yum -nogpgcheck localinstall AmazonRedshiftODBC-64-bit-1.x.xx.xxxx-x.x86_64.rpm
     ```

   - 在 SLES 上，執行下列命令。

     ```
     zypper install RPMFileName
     ```

     將 *RPMFileName* 取代為 RPM 套件檔案名稱。例如，下列命令示範安裝 64 位元驅動程式。

     ```
     zypper install AmazonRedshiftODBC-1.x.x.xxxx-x.x86_64.rpm
     ```

   - 在 Debian 上，執行下列命令。

     ```
     sudo apt install ./DEBFileName.deb
     ```

     將 *DEBFileName.deb* 取代為 Debian 套件檔案名稱。例如，下列命令示範安裝 64 位元驅動程式。

     ```
     sudo apt install ./AmazonRedshiftODBC-1.x.x.xxxx-x.x86_64.deb
     ```

> **Important**
>
> 驅動程式安裝完成時，請設定來用於您的系統。如需驅動程式組態的相關資訊，請參閱使用 ODBC 驅動器管理員設定驅動器。

使用 ODBC 驅動器管理員設定驅動器

在 Linux 作業系統上，您會使用 ODBC 驅動器管理員來設定 ODBC 連線設定。ODBC 驅動程式管理員使用組態檔案來定義及設定 ODBC 資料來源和驅動程式。您使用的 ODBC 驅動程式管理員視您使用的作業系統而定。對於 Linux，這是 unixODBC 驅動器管理員。

如需支援用來設定 Amazon Redshift ODBC 驅動器的 ODBC 驅動器管理員的詳細資訊，請參閱 Linux 作業系統的 在 Linux 上使用 Amazon Redshift ODBC 驅動器。另請參閱 Amazon Redshift ODBC 連接器安裝和組態指南中的「在非 Windows 機器上指定 ODBC 驅動程式管理員」。

設定 Amazon Redshift ODBC 驅動程式需要三個檔案：`amazon.redshiftodbc.ini`、`odbc.ini` 和 `odbcinst.ini`。

如果是安裝到預設位置，則 `amazon.redshiftodbc.ini` 組態檔案位於下列其中一個目錄：

- `/opt/amazon/redshiftodbc/lib/64` (適用於 Linux 作業系統上的 64 位元驅動程式)
- `/opt/amazon/redshiftodbc/lib/32` (適用於 Linux 作業系統上的 32 位元驅動程式)

此外，在 Linux 的 `/opt/amazon/redshiftodbc/Setup` 下可找到範例 `odbc.ini` 和 `odbcinst.ini` 檔案。您可以使用這些檔案做為設定 Amazon Redshift ODBC 驅動程式和資料來源名稱 (DSN) 的範例。

不建議使用 Amazon Redshift ODBC 驅動程式安裝目錄來存放組態檔案。`Setup` 目錄中的範例檔案僅供當作範例。如果您後來重新安裝 Amazon Redshift ODBC 驅動程式，或升級為更新版本，安裝目錄會遭到覆寫。然後，您對這些檔案可能做的任何變更都會遺失。

若要避免這種情況，請將 `amazon.redshiftodbc.ini` 檔案複製到安裝目錄以外的目錄。如果您將此檔案複製到使用者的主目錄，請在檔案名稱開頭加上一點 (.)，將它變成隱藏檔。

對於 `odbc.ini` 和 `odbcinst.ini` 檔案，請使用位於使用者主目錄中的組態檔案，或在另一個目錄中建立新版本。根據預設，Linux 作業系統的使用者主目錄 (/home/$USER 或 ~/) 中，應該有 `odbc.ini` 檔案和 `odbcinst.ini` 檔案。這些預設檔案是隱藏檔，每個檔案名稱前面以點表示。僅當使用 `-a` 旗標來列出目錄內容時，才會顯示這些檔案。

無論對 odbc.ini 和 odbcinst.ini 檔案選擇何種選項，都會修改這些檔案來新增驅動程式和 DSN 組態資訊。如果建立新檔案，則還需要設定環境變數來指定這些組態檔案所在的位置。

根據預設，ODBC 驅動程式管理員會設定為使用隱藏版本的 odbc.ini 和 odbcinst.ini 組態檔 (命名為 .odbc.ini 和 .odbcinst.ini)，這些組態檔位於主目錄中。它們也會設定為使用驅動程式安裝目錄的 /lib 子資料夾中的 amazon.redshiftodbc.ini 檔案。如果您將這些組態檔儲存在其他位置，請設定如下所述的環境變數，以便驅動程式管理員可以找到這些檔案。如需詳細資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南中的「指定驅動程式組態檔案的位置」。

在 Linux 作業系統上建立資料來源名稱

使用資料來源名稱 (DSN) 連接至資料倉庫時，請設定 odbc.ini 檔案來定義 DSN。在 odbc.ini 檔案中設定屬性，以建立 DSN 來指定資料倉庫的連線資訊。

如需有關如何設定odbc.ini檔案的資訊，請參閱《Amazon Redshift ODBC 連接器安裝和組態指南

在 Linux 作業系統上使用下列格式。

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file

Host=cluster_endpoint
Port=port_number
Database=database_name
locale=locale
```

以下範例示範 Linux 作業系統上使用 64 位元 ODBC 驅動程式的 odbc.ini 組態。

```
[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
```

以下範例示範 Linux 作業系統上使用 32 位元 ODBC 驅動程式的 odbc.ini 組態。

```
[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 作業系統上設定沒有 DSN 的連線

若要透過沒有 DSN 的連線連接至您的資料存放區，請在 odbcinst.ini 檔案中定義驅動程式。然後，在應用程式中提供無 DSN 的連線字串。

如需如何在此情況下設定 odbcinst.ini 檔案的相關資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南中的「在非 Windows 機器上設定無 DSN 的連線」。

在 Linux 作業系統上使用下列格式。

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/driver_file

...
```

以下範例針對 Linux 作業系統上安裝於預設目錄中的 64 位元驅動程式示範 odbcinst.ini 組態。

```
[ODBC Drivers]
Amazon Redshift (x64)=Installed

[Amazon Redshift (x64)]
Description=Amazon Redshift ODBC Driver (64-bit)
Driver=/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so
```

以下範例針對 Linux 作業系統上安裝於預設目錄中的 32 位元驅動程式示範 odbcinst.ini 組態。

```
[ODBC Drivers]
```

```
Amazon Redshift (x86)=Installed

[Amazon Redshift (x86)]
Description=Amazon Redshift ODBC Driver (32-bit)
Driver=/opt/amazon/redshiftodbc/lib/32/libamazonredshiftodbc32.so
```

設定環境變數

使用正確的 ODBC 驅動程式管理員來載入正確的驅動程式。若要這樣做，請設定程式庫路徑環境變數。如需詳細資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南中的「在非 Windows 機器上指定 ODBC 驅動程式管理員」。

根據預設，ODBC 驅動程式管理員會設定為使用隱藏版本的 `odbc.ini` 和 `odbcinst.ini` 組態檔 (命名為 `.odbc.ini` 和 `.odbcinst.ini`)，這些組態檔位於主目錄中。它們也會設定為使用驅動程式安裝目錄的 `/lib` 子資料夾中的 `amazon.redshiftodbc.ini` 檔案。如果您將這些組態檔儲存在其他位置，請設定環境變數，以便驅動程式管理員可以找到這些檔案。如需詳細資訊，請參閱《Amazon Redshift ODBC 連接器安裝和組態指南》中的「指定驅動程式組態檔案的位置」。

設定連線功能

您可以將下列連線功能設定給 ODBC 設定：

- 設定 ODBC 驅動程式以提供憑證，並驗證 Amazon Redshift 資料庫的連線。
- 設定 ODBC 驅動程式，以連線到啟用 Secure Sockets Layer (SSL) 的通訊端 (如果您連線到已啟用 SSL 的 Amazon Redshift 伺服器)。
- 設定 ODBC 驅動程式，以透過 Proxy 伺服器連線到 Amazon Redshift。
- 將 ODBC 驅動程式設定為使用查詢處理模式，以防止查詢耗用太多記憶體。
- 設定 ODBC 驅動程式，以透過代理伺服器傳遞 IAM 身分驗證程序。
- 設定 ODBC 驅動程式，以使用 TCP 保持連線來防止連線逾時。

如需這些連線功能的相關資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南。

在 macOS 上使用 Amazon Redshift ODBC 驅動程式

請在存取 Amazon Redshift 資料倉儲的用戶端電腦上安裝驅動程式。您安裝驅動程式的每一台電腦必須符合最低系統需求清單：如需最低系統要求的相關資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南。

主題

- 下載並安裝 Amazon Redshift ODBC 驅動器
- 使用 ODBC 驅動器管理員設定驅動器

下載並安裝 Amazon Redshift ODBC 驅動器

使用本節中的步驟，在支援的 macOS 版本上下載並安裝 Amazon Redshift ODBC 驅動程式。安裝程序會將驅動程式檔案安裝在下列目錄中：

- `/opt/amazon/redshift/lib/universal`
- `/opt/amazon/redshift/ErrorMessages`
- `/opt/amazon/redshift/Setup`

在 macOS 上安裝 Amazon Redshift ODBC 驅動程式

1. 若要在 macOS 上安裝 Amazon Redshift ODBC 驅動程式，請下載 macOS 驅動程式 1.6.3 版。

   然後下載並檢閱 Amazon Redshift ODBC 和 JDBC 驅動程式授權合約。
2. 按兩下 AmazonRedshiftODBC.pkg 來執行安裝程式。
3. 依照安裝程式中的步驟完成驅動程式安裝程序。若要執行安裝，請同意授權合約的條款。

> **Important**
>
> 驅動程式安裝完成時，請設定來用於您的系統。如需驅動程式組態的相關資訊，請參閱使用 ODBC 驅動器管理員設定驅動器。

使用 ODBC 驅動器管理員設定驅動器

在 macOS 作業系統上，您可以使用 ODBC 驅動程式管理員來設定 ODBC 連線設定。ODBC 驅動程式管理員使用組態檔案來定義及設定 ODBC 資料來源和驅動程式。您使用的 ODBC 驅動程式管理員視您使用的作業系統而定。對於 macOS 作業系統，它是 iODBC 驅動程式管理員。

如需有關支援 ODBC 驅動程式管理員以設定 Amazon Redshift ODBC 驅動程式的詳細資訊，請參閱 macOS 作業系統在 macOS 上使用 Amazon Redshift ODBC 驅動程式的 。另請參閱 Amazon Redshift ODBC 連接器安裝和組態指南中的「在非 Windows 機器上指定 ODBC 驅動程式管理員」。

設定 Amazon Redshift ODBC 驅動程式需要三個檔案：amazon.redshiftodbc.ini、odbc.ini 和 odbcinst.ini。

如果是安裝到預設位置，則 amazon.redshiftodbc.ini 組態檔案位於 /opt/amazon/redshift/lib。

此外，在 macOS /opt/amazon/redshift/Setup的 下，有範例odbc.ini和odbcinst.ini檔案。您可以使用這些檔案做為設定 Amazon Redshift ODBC 驅動程式和資料來源名稱 (DSN) 的範例。

不建議使用 Amazon Redshift ODBC 驅動程式安裝目錄來存放組態檔案。Setup 目錄中的範例檔案僅供當作範例。如果您後來重新安裝 Amazon Redshift ODBC 驅動程式，或升級為更新版本，安裝目錄會遭到覆寫。然後，您對這些檔案可能做的任何變更都會遺失。

若要避免這種情況，請將 amazon.redshiftodbc.ini 檔案複製到安裝目錄以外的目錄。如果您將此檔案複製到使用者的主目錄，請在檔案名稱開頭加上一點 (.)，將它變成隱藏檔。

對於 odbc.ini 和 odbcinst.ini 檔案，請使用位於使用者主目錄中的組態檔案，或在另一個目錄中建立新版本。根據預設，您的 macOS 作業系統應該在使用者的主目錄中具有 檔案odbc.ini和 odbcinst.ini 檔案 (/home/$USER 或 .)~/。這些預設檔案是隱藏檔，每個檔案名稱前面以點表示。僅當使用 -a 旗標來列出目錄內容時，才會顯示這些檔案。

無論對 odbc.ini 和 odbcinst.ini 檔案選擇何種選項，都會修改這些檔案來新增驅動程式和 DSN 組態資訊。如果建立新檔案，則還需要設定環境變數來指定這些組態檔案所在的位置。

根據預設，ODBC 驅動程式管理員會設定為使用隱藏版本的 odbc.ini 和 odbcinst.ini 組態檔 (命名為 .odbc.ini 和 .odbcinst.ini)，這些組態檔位於主目錄中。它們也會設定為使用驅動程式安裝目錄的 /lib 子資料夾中的 amazon.redshiftodbc.ini 檔案。如果您將這些組態檔儲存在其他位置，請設定如下所述的環境變數，以便驅動程式管理員可以找到這些檔案。如需詳細資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南中的「指定驅動程式組態檔案的位置」。

在 macOS X 作業系統上建立資料來源名稱

使用資料來源名稱 (DSN) 連接至資料倉庫時，請設定 odbc.ini 檔案來定義 DSN。在 odbc.ini 檔案中設定屬性，以建立 DSN 來指定資料倉庫的連線資訊。

如需有關如何設定odbc.ini檔案的資訊，請參閱《Amazon Redshift ODBC 連接器安裝和組態指南》中的「在非 Windows 機器上建立資料來源名稱」。

在 macOS 作業系統上使用下列格式。

```
[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
```

下列範例顯示 macOS 作業系統odbc.ini上的 組態。

```
[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 作業系統上設定沒有 DSN 的連線

若要透過沒有 DSN 的連線連接至您的資料存放區，請在 odbcinst.ini 檔案中定義驅動程式。然後，在應用程式中提供無 DSN 的連線字串。

如需如何在此情況下設定 odbcinst.ini 檔案的相關資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南中的「在非 Windows 機器上設定無 DSN 的連線」。

在 macOS 作業系統上使用下列格式。

```
[ODBC Drivers]
driver_name=Installed
...

[driver_name]
Description=driver_description
Driver=path/lib/amazonredshiftodbc.dylib

...
```

下列範例顯示安裝在 macOS 作業系統預設目錄中的驅動程式odbcinst.ini組態。

```
[ODBC Drivers]
Amazon RedshiftODBC DSN=Installed

[Amazon RedshiftODBC DSN]
Description=Amazon Redshift ODBC Driver for macOS
Driver=/opt/amazon/redshift/lib/amazonredshiftodbc.dylib
```

設定環境變數

使用正確的 ODBC 驅動程式管理員來載入正確的驅動程式。若要這樣做，請設定程式庫路徑環境變數。如需詳細資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南中的「在非 Windows 機器上指定 ODBC 驅動程式管理員」。

根據預設，ODBC 驅動程式管理員會設定為使用隱藏版本的 `odbc.ini` 和 `odbcinst.ini` 組態檔 (命名為 `.odbc.ini` 和 `.odbcinst.ini`)，這些組態檔位於主目錄中。它們也會設定為使用驅動程式安裝目錄的 `/lib` 子資料夾中的 `amazon.redshiftodbc.ini` 檔案。如果您將這些組態檔儲存在其他位置，請設定環境變數，以便驅動程式管理員可以找到這些檔案。如需詳細資訊，請參閱《Amazon Redshift ODBC 連接器安裝和組態指南》中的「指定驅動程式組態檔案的位置」。

設定連線功能

您可以將下列連線功能設定給 ODBC 設定：

- 設定 ODBC 驅動程式以提供憑證，並驗證 Amazon Redshift 資料庫的連線。
- 設定 ODBC 驅動程式，以連線到啟用 Secure Sockets Layer (SSL) 的通訊端 (如果您連線到已啟用 SSL 的 Amazon Redshift 伺服器)。
- 設定 ODBC 驅動程式，以透過 Proxy 伺服器連線到 Amazon Redshift。
- 將 ODBC 驅動程式設定為使用查詢處理模式，以防止查詢耗用太多記憶體。
- 設定 ODBC 驅動程式，以透過代理伺服器傳遞 IAM 身分驗證程序。
- 設定 ODBC 驅動程式，以使用 TCP 保持連線來防止連線逾時。

如需這些連線功能的相關資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南。

ODBC 驅動器選項

您可以使用組態選項來控制 Amazon Redshift ODBC 驅動程式的行為。

在 Microsoft Windows 中，您通常於設定資料來源名稱 (DSN) 時設定驅動程式選項。您也可以在連接時以程式設計方式在連線字串中設定驅動程式選項，或在 HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI*your_DSN* 中新增或變更登錄機碼。如需設定 DSN 的相關資訊，請參閱在 Microsoft Windows 上使用 Amazon Redshift ODBC 驅動器。

在 macOS 中，您可以在 odbc.ini和 amazon.redshiftodbc.ini 檔案中設定驅動程式組態選項，如中所述使用 ODBC 驅動器管理員設定驅動器。在 amazon.redshiftodbc.ini 檔案中設定的組態選項會套用至所有連線。相反地，odbc.ini 檔案中設定的組態選項是針對某個連線。odbc.ini 中設定的組態選項優先於 amazon.redshiftodbc.ini 中設定的組態選項。

如需如何設定 ODBC 驅動程式組態選項的相關資訊，請參閱 Amazon Redshift ODBC 連接器安裝和組態指南。

先前的 ODBC 驅動程式版本

只有在您的工具需要特定版本的驅動程式時，才需要下載舊版的 Amazon Redshift ODBC 驅動程式。

先前的 ODBC 驅動器版本 (適用於 Windows)

以下是 64 位元驅動程式：

- 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 位元的驅動程式已停止使用，先前的版本已不受支援。

先前的 ODBC 驅動器版本 (適用於 Linux)

以下是 64 位元驅動程式版本：

- 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 位元的驅動程式已停止使用，先前的版本已不受支援。

適用於 macOS 的舊版 ODBC 驅動程式

以下是適用於 macOS 的 Amazon Redshift ODBC 驅動程式版本：

- 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

探索中繼資料的最佳實務

Amazon Redshift 提供多種方法來探索中繼資料。對於應用程式和工具，包括使用 AI 代理器建置的應用程式和工具，我們建議您使用驅動程式中繼資料 API 或SHOW命令來實現一致、可靠和高效能的中繼資料探索。

主題

- 將 Amazon Redshift 驅動程式中繼資料 API 用於應用程式和工具
- 使用 SHOW 命令
- 查詢系統資料表
- 設定應用程式名稱連線屬性

將 Amazon Redshift 驅動程式中繼資料 API 用於應用程式和工具

對於連接到 Amazon Redshift 的應用程式和工具，例如商業智慧工具或查詢編輯器，我們建議您使用 Amazon Redshift JDBC 2.x、ODBC 2.x 或 Python 驅動程式提供的驅動程式中繼資料 API，來探索有關資料倉儲物件的中繼資料，包括資料庫、結構描述、資料表、資料欄和資料類型。或者，您可以使用 Amazon Redshift SHOW命令。

使用驅動程式中繼資料 API 享有下列優點：

- 規格合規。JDBC 和 ODBC 驅動程式實作標準中繼資料界面 (DatabaseMetaData在 JDBC 中，SQLTables在 ODBC SQLColumns中）。由於 Python 的資料庫 API (PEP 249) 未定義中繼資料 API 規格，因此 Amazon Redshift Python 驅動程式會遵循 JDBC DatabaseMetaData 規格，提供同等的方法，例如 get_tables()、 get_columns()和 get_schemas()。這些 APIs遵循明確定義的規格，因此您的整合程式碼是可攜式的。隨著 Amazon Redshift 發展其內部系統資料表，您的應用程式不需要變更。
- 效能最佳化。驅動程式中繼資料 API 經過最佳化，可有效率地傳回中繼資料。 AWS 會繼續投資驅動程式中繼資料 API 效能。
- 向前相容。Amazon Redshift 遵循 JDBC、ODBC 和 Python 連接器規格。當您針對這些標準 APIs編寫程式碼時，您的應用程式會受到保護，免於基礎系統目錄結構的變更。

範例：使用 JDBC DatabaseMetaData.getTables() 擷取資料表中繼資料

```
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"});
```

範例：使用 Python cursor.get_columns() 擷取資料欄中繼資料

```
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', '%')
```

範例：使用 ODBC SQLPrimaryKeys() 擷取主要金鑰中繼資料

```
// 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);
    }
}
```

範例：使用 ODBC SQLTables() 列出資料庫和結構描述

ODBC API 不提供個別的函數來列出目錄或結構描述。反之，您可以使用 的特殊呼叫慣例SQLTables()來擷取此資訊。

列出所有資料庫 （ 目錄 ）

呼叫 SQLTables()，並將 CatalogName設定為 SQL_ALL_CATALOGS。將 SchemaName和 TableName 設定為空白字串。結果集只會傳回資料TABLE_CAT欄中的有效值。所有其他資料欄都包含 NULLs。

```
// 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)
```

列出所有結構描述

呼叫 SQLTables()，並將 SchemaName設定為 SQL_ALL_SCHEMAS。將 CatalogName和 TableName 設定為空白字串。

```
// 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 規格僅定義為結構描述列舉TABLE_SCHEM的有效值。Amazon Redshift 也會填入 ，TABLE_CAT因為它支援跨資料庫中繼資料探索，而且每個結構描述的範圍都限定在特定資料庫。

使用 SHOW 命令

如果驅動程式中繼資料 API 未涵蓋您的使用案例，您可以使用 Amazon Redshift SHOW命令來擷取中繼資料。 SHOW命令已針對快速中繼資料擷取進行最佳化。當您在互動式 SQL 工作階段中需要中繼資料，或您的應用程式透過不會公開驅動程式中繼資料 API 的用戶端連線時，它們很有用。

常見的探索操作支援下列SHOW命令：

- 顯示資料庫
- 顯示結構描述
- 顯示資料表
- 顯示資料欄
- 顯示限制條件
- 顯示授予
- 顯示 函數
- 顯示程序
- 顯示參數

範例

```
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     |
```

如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 SHOW。

查詢系統資料表

如果驅動程式中繼資料 API 和SHOW命令都不符合您的需求，您可以查詢 PostgreSQL 目錄資料表 (pg_*)、資訊結構描述 (information_schema) 或 Amazon Redshift 系統檢視，例如SVV_TABLE_INFO直接。

此方法適用於您從互動式 SQL 查詢執行的臨機操作探索、偵錯或管理任務，例如在 Amazon Redshift 查詢編輯器 v2 或 中psql。不過，系統目錄結構可能會在 Amazon Redshift 版本之間變更。我們不建議您依賴直接系統資料表查詢做為生產應用程式整合的中繼資料界面。

當您查詢系統資料表時，請遵循下列實務：

- 指定資料庫、結構描述和資料表。為了獲得最佳效能，請在查詢篩選條件中包含資料庫名稱、結構描述名稱和資料表名稱。縮小範圍可減少 Amazon Redshift 掃描和傳回所需的中繼資料數量。
- 使用等式或 LIKE 篩選條件。在子WHERE句中使用等式 (=) 或LIKE述詞來縮小結果範圍。避免對大型目錄檢視進行未篩選的查詢。在具有許多物件的叢集上，它們可能會變慢。
- 避免篩選條件述詞中的函數。當您篩選系統資料表查詢時，請勿在子WHERE句中使用 LOWER()或 字串串連等函數。這些可防止有效率的中繼資料查詢，並可能大幅降低查詢效能。

反模式與建議查詢的範例

```
-- 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';
```

設定應用程式名稱連線屬性

當您連線到 Amazon Redshift 時，請設定選用的應用程式名稱驅動程式屬性。此屬性會識別建立每個連線的應用程式或工具，讓您更輕鬆地監控、稽核和疑難排解 Amazon Redshift 環境。

您提供的應用程式名稱會顯示在 的 application_name 欄中SYS_CONNECTION_LOG。您可以使用它來執行下列動作：

- 追蹤連線來源。識別哪些應用程式、服務或工具連接到您的叢集。
- 對效能進行故障診斷。將查詢與原始應用程式建立關聯，以隔離效能問題。
- 稽核存取權。了解整個資料倉儲的應用程式層級用量模式。

下表顯示如何設定每個 Amazon Redshift 驅動程式的應用程式名稱。

驅動程式	屬性	範例
JDBC	ApplicationName	jdbc:redshift:// *endpoint*:*port*/*databas* ApplicationName=my-etl-pipeline
ODBC	ApplicationName	在 DSN 或連線字串ApplicationName=my-bi-tool中設定 。
Python	application_name	redshift_connector.connect(..., application_name='my-data-catalog')

如需每個驅動程式應用程式名稱屬性的詳細資訊，請參閱下列內容：

- JDBC：ApplicationName
- ODBC：ApplicationName
- Python：application_name

> **Note**
>
> 針對應用程式名稱使用描述性、一致的命名慣例，例如 *team*-*service*-*environment*（例如 quicksight-analytics-dashboard-prod)。這種方法可讓您更輕鬆地篩選和分析SYS_CONNECTION_LOG資料。

設定連線的安全選項

Amazon Redshift 支援 Secure Sockets Layer (SSL) 連線，可將資料和伺服器憑證加密，以驗證用戶端所連線的伺服器憑證。

SSL

為了支援 SSL 連線，Amazon Redshift 會在每一個叢集上建立並安裝 AWS Certificate Manager (ACM) 發行的 SSL 憑證。ACM 憑證受到大多數作業系統、網頁瀏覽器和用戶端公開信任。如果 SQL 用戶端或應用程式在 sslmode 連線選項設為 require、verify-ca 或 verify-full 的情況下，使用 SSL 連線到 Amazon Redshift，您可能需要下載憑證套件。如果用戶端需要憑證，Amazon Redshift 會提供如下的憑證套件：

> **Note**
>
> 我們建議您盡可能使用 verify-full進行 SSL 連線。

- 從 https://s3.amazonaws.com/redshift-downloads/amazon-trust-ca-bundle.crt 下載套件。
  - 預期的 MD5 檢查總和碼為 418dea9b6d5d5de7a8f1ac42e164cdcf。
  - sha256 檢查總和碼為 36dba8e4b8041cd14b9d60158893963301bcbb92e1c456847784de2acb5bd550。

  請勿使用位於 https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt 的先前憑證套件。

- 在中國 AWS 區域，請從 https：//https://s3.cn-north-1.amazonaws.com.cn/redshift-downloads-cn/amazon-trust-ca-bundle.crt 下載套件。
  - 預期的 MD5 檢查總和碼為 418dea9b6d5d5de7a8f1ac42e164cdcf。
  - sha256 檢查總和碼為 36dba8e4b8041cd14b9d60158893963301bcbb92e1c456847784de2acb5bd550。

  請勿使用位於 `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` 的先前憑證套件。

> **Important**
>
> Amazon Redshift 已變更 SSL 憑證的管理方式。您可能需要更新目前的信任根 CA 憑證，才能繼續使用 SSL 連線到叢集。如需詳細資訊，請參閱將 SSL 連線轉換為使用 ACM 憑證。

根據預設，不論連線是否使用 SSL，叢集資料庫都接受連線。若要將叢集設為需要 SSL 連線，請在與叢集相關聯的參數群組中，將 `require_SSL` 參數設為 `true`。

Amazon Redshift 支援符合美國聯邦資訊處理標準 (FIPS) 140-2 的 SSL 模式。符合 FIPS 的 SSL 模式預設為停用。

> **Important**
>
> 只在系統需要符合 FIPS 時啟用與 FIPS 相容的 SSL 模式。

若要啟用與 FIPS 相容的 SSL 模式，請在與 Amazon Redshift 叢集或 Redshift Serverless 工作群組相關聯的參數群組中，將 `use_fips_ssl` 參數和 `require_SSL` 參數都設為 `true`。如需修改叢集上參數群組的相關資訊，請參閱 Amazon Redshift 參數群組。如需修改工作群組上參數群組的相關資訊，請參閱 設定與 FIPS 相容的 Amazon Redshift Serverless SSL 連線。

Amazon Redshift 支援 Elliptic Curve Diffie—Hellman Ephemeral (ECDHE) 金鑰協議通訊協定。使用 ECDHE 時，用戶端和伺服器各有橢圓曲線公有-私有金鑰對，用於透過不安全通道建立共用秘密。您不需要在 Amazon Redshift 中設定任何項目來啟用 ECDHE。如果您從 SQL 用戶端工具連線，而此工具使用 ECDHE 來加密用戶端和伺服器之間的通訊，Amazon Redshift 會使用提供的密碼清單來建立適當的連線。如需詳細資訊，請參閱 Wikipedia 上的 Elliptic curve diffie—hellman 和 OpenSSL 網站上的 Ciphers。

ODBC 中的 SSL 和信任 CA 憑證

如果您使用最新的 Amazon Redshift ODBC 驅動程式 (1.3.7.1000 版或更新版本) 來連接，則可略過本節。若要下載最新的驅動程式，請參閱設定 Amazon Redshift ODBC 驅動程式 2.x 版連線。

您可能需要更新目前的信任根 CA 憑證，才能繼續使用 SSL 連線到叢集。如需詳細資訊，請參閱SSL。

您可以驗證您下載的憑證是否符合預期的 MD5 檢查總和碼。若要這樣做，您可以在 Linux 作業系統上使用 Md5sum 程式，或在 Windows 和 macOS 作業系統上使用其他工具。

ODBC DSN 包含 `sslmode` 設定，可決定如何處理用戶端連線和伺服器憑證驗證的加密。Amazon Redshift 支援來自用戶端連線的下列 `sslmode` 值：

- `disable`

  停用 SSL，不加密連線。
- `allow`

  如果伺服器需要，就使用 SSL。
- `prefer`

  如果伺服器支援，就使用 SSL。Amazon Redshift 支援 SSL，因此當您將 `sslmode` 設定為 `prefer` 時，系統會使用 SSL。
- `require`

  需要 SSL。
- `verify-ca`

  必須使用 SSL 且必須驗證伺服器憑證。
- `verify-full`

  必須使用 SSL。必須驗證伺服器憑證，且伺服器主機名稱必須符合憑證上的主機名稱屬性。

您可以判斷用戶端和伺服器之間的連線是否使用 SSL 及是否驗證伺服器憑證。若要執行此作業，您需要在用戶端檢閱 ODBC DSN 的 `sslmode` 設定，並在伺服器上檢閱 Amazon Redshift 叢集的 `require_SSL` 設定。下表描述各種用戶端和伺服器設定組合的加密結果：

sslmode (用戶端)	require_SSL (伺服器)	結果
disable	false	不加密連線。
disable	true	無法建立連線，因為伺服器需要 SSL，但用戶端對連線停用 SSL。
allow	true	加密連線。
allow	false	不加密連線。
prefer 或 require	true	加密連線。
prefer 或 require	false	加密連線。
verify-ca	true	加密連線且驗證伺服器憑證。
verify-ca	false	加密連線且驗證伺服器憑證。
verify-full	true	加密連線且驗證伺服器憑證，並驗證主機名稱。
verify-full	false	加密連線且驗證伺服器憑證，並驗證主機名稱。

在 Microsoft Windows 上搭配 ODBC 使用伺服器憑證進行連線

如果您想要使用 SSL 和伺服器憑證連線到叢集，請先將憑證下載到用戶端電腦或 Amazon EC2 執行個體。然後，設定 ODBC DSN。

1. 將 Amazon Redshift 憑證授權機構套件下載到用戶端電腦驅動程式安裝目錄中的 lib 資料夾，並將檔案儲存為 root.crt。如需下載資訊，請參閱 SSL。
2. 開啟 ODBC 資料來源管理員，然後新增或編輯 ODBC 連線的系統 DSN 項目。在 SSL 模式中，選取 verify-full (除非您使用 DNS 別名)。如果您使用 DNS 別名，請選取 verify-ca。然後選擇 Save (儲存)。

如需設定 ODBC DSN 的相關資訊，請參閱設定 Amazon Redshift ODBC 驅動程式 2.x 版連線。

Java 中的 SSL 和伺服器憑證

SSL 會將用戶端和伺服器之間移動的資料加密，多一道安全性。使用伺服器憑證可透過驗證叢集是否為 Amazon Redshift 叢集來提供額外一層的安全。在作法上是檢查您佈建的所有叢集上自動安裝的伺服器憑證。如需搭配 JDBC 使用伺服器憑證的相關資訊，請前往 PostgreSQL 文件中的 Configuring the client。

在 Java 中使用信任 CA 憑證進行連線

> **Important**
>
> Amazon Redshift 已變更 SSL 憑證的管理方式。您可能需要更新目前的信任根 CA 憑證，才能繼續使用 SSL 連線到叢集。如需詳細資訊，請參閱SSL。

使用信任 CA 憑證來連接

您可以使用 `redshift-keytool.jar` 檔案，將 Amazon Redshift 憑證授權機構套件中的 CA 憑證匯入至 Java TrustStore 或您的私有信任憑證庫。

1. 如果您使用 Java 命令列 `-Djavax.net.ssl.trustStore` 選項，可能的話，請從命令列中移除。
2. 下載 redshift-keytool.jar。
3. 執行以下任意一項：

   - 若要將 Amazon Redshift 憑證授權機構套件匯入 Java TrustStore，請執行下列命令。

     ```
     java -jar redshift-keytool.jar -s
     ```

   - 若要將 Amazon Redshift 憑證授權機構套件匯入您的私有 TrustStore，請執行下列命令：

     ```
     java -jar redshift-keytool.jar -k <your_private_trust_store> -
     p <keystore_password>
     ```

將 SSL 連線轉換為使用 ACM 憑證

Amazon Redshift 將會以 AWS Certificate Manager (ACM) 發行的憑證取代您叢集上的 SSL 憑證。ACM 是大多數現行系統信任的公有憑證授權機構 (CA)。您可能需要更新目前的信任根 CA 憑證，才能繼續使用 SSL 連線到叢集。

只有在下列所有情況都存在時，此變更才會影響您：

- SQL 用戶端或應用程式使用 SSL 來連接至 Amazon Redshift 叢集，且 `sslMode` 連線選項設為 `require`、`verify-ca` 或 `verify-full` 組態選項。
- 您不使用 Amazon Redshift ODBC 或 JDBC 驅動程式，或使用的 Amazon Redshift 驅動程式是 ODBC 1.3.7.1000 版或 JDBC 1.2.8.1005 版以前。

如果此變更會在 Amazon Redshift 商業區域影響到您，則您必須在 2017 年 10 月 23 日之前更新目前的信任根 CA 憑證。Amazon Redshift 會從現在開始到 2017 年 10 月 23 日之前，將您的叢集轉換為使用 ACM 憑證。此變更對叢集效能或可用性的影響應該非常小，或完全不影響。

如果此變更影響 AWS GovCloud (US)（美國）區域，則您必須在 2020 年 4 月 1 日之前更新目前的信任根 CA 憑證，以避免服務中斷。從這個日期起，使用 SSL 加密連線連線至 Amazon Redshift 叢集的用戶端需要額外的受信任憑證授權單位 (CA)。用戶端在連線到 Amazon Redshift 叢集時，會使用受信任的憑證授權單位來確認該叢集的身分。您必須採取動作，才能更新 SQL 用戶端和應用程式，以使用包含新信任 CA 的更新憑證套件。

> **Important**
>
> 在 2021 年 1 月 5 日的中國區域中，Amazon Redshift 會將叢集上的 SSL 憑證取代為 AWS Certificate Manager (ACM) 發行的憑證。如果此變更會在中國 (北京) 區域或中國 (寧夏) 區域影響到您，則您必須在 2021 年 1 月 5 日之前更新目前的信任根 CA 憑證，以避免服務中斷。從這個日期起，使用 SSL 加密連線連線至 Amazon Redshift 叢集的用戶端需要額外的受信任憑證授權單位 (CA)。用戶端在連線到 Amazon Redshift 叢集時，會使用受信任的憑證授權單位來確認該叢集的身分。您必須採取動作，才能更新 SQL 用戶端和應用程式，以使用包含新信任 CA 的更新憑證套件。

- 使用最新的 Amazon Redshift ODBC 或 JDBC 驅動程式
- 使用較舊的 Amazon Redshift ODBC 或 JDBC 驅動程式
- 使用其他 SSL 連線類型

使用最新的 Amazon Redshift ODBC 或 JDBC 驅動程式

建議使用最新的 Amazon Redshift ODBC 或 JDBC 驅動程式。從 ODBC 1.3.7.1000 版和 JDBC 1.2.8.1005 版開始的 Amazon Redshift 驅動程式，將會自動處理從 Amazon Redshift 自簽憑證轉換到 ACM 憑證。若要下載最新的驅動程式，請參閱設定適用於 Amazon Redshift 的 JDBC 驅動器 2.x 版的連線。

如果您使用最新的 Amazon Redshift JDBC 驅動程式，最好不要在 JVM 選項中使用 `-Djavax.net.ssl.trustStore`。如果您必須使用 `-Djavax.net.ssl.trustStore`，請將 Redshift 憑證授權機構套件匯入其指向的信任憑證庫。如需下載資訊，請參閱 SSL。如需詳細資訊，請參閱將 Amazon Redshift 憑證授權機構套件匯入 TrustStore。

使用較舊的 Amazon Redshift ODBC 或 JDBC 驅動程式

- 如果 ODBC DSN 設定為 `SSLCertPath`，請覆寫指定路徑中的憑證檔案。
- 如果未設定 `SSLCertPath`，請覆寫驅動程式 DLL 位置中名為 `root.crt` 的憑證檔案。

如果您必須使用的 1.2.8.1005 版以前的 Amazon Redshift JDBC 驅動程式，請執行下列其中一個動作：

- 如果 JDBC 連線字串使用 `sslCert` 選項，請移除 `sslCert` 選項。然後將 Redshift 憑證授權機構套件匯入 Java TrustStore。如需下載資訊，請參閱 SSL。如需詳細資訊，請參閱將 Amazon Redshift 憑證授權機構套件匯入 TrustStore。
- 如果您使用 Java 命令列 `-Djavax.net.ssl.trustStore` 選項，可能的話，請從命令列中移除。然後將 Redshift 憑證授權機構套件匯入 Java TrustStore。如需下載資訊，請參閱 SSL。如需詳細資訊，請參閱將 Amazon Redshift 憑證授權機構套件匯入 TrustStore。

將 Amazon Redshift 憑證授權機構套件匯入 TrustStore

您可以使用 `redshift-keytool.jar` 將 Amazon Redshift 憑證授權機構套件中的 CA 憑證匯入至 Java TrustStore 或您的私有信任憑證庫。

將 Amazon Redshift 憑證授權機構套件匯入 TrustStore

1. 下載 redshift-keytool.jar。
2. 執行以下任意一項：
    - 若要將 Amazon Redshift 憑證授權機構套件匯入 Java TrustStore，請執行下列命令。

```
java -jar redshift-keytool.jar -s
```

- 若要將 Amazon Redshift 憑證授權機構套件匯入您的私有 TrustStore，請執行下列命令：

```
java -jar redshift-keytool.jar -k <your_private_trust_store> -
p <keystore_password>
```

使用其他 SSL 連線類型

如果您使用下列任何一項來連接，請遵循本節的步驟：

- 開放原始碼 ODBC 驅動程式
- 開放原始碼 JDBC 驅動程式
- Amazon Redshift RSQL 命令列界面
- 任何以 libpq 為基礎的語言繫結，例如 psycopg2 (Python) 和 ruby-pg (Ruby)

若要對其他 SSL 連線類型使用 ACM 憑證，請執行下列動作：

1. 下載 Amazon Redshift 憑證授權機構套件。如需下載資訊，請參閱 SSL。
2. 將套件中的憑證放入 `root.crt` 檔案中。
   - 在 Linux 和 macOS 作業系統上，檔案為 `~/.postgresql/root.crt`。
   - 在 Microsoft Windows 上，此檔案是 `%APPDATA%\postgresql\root.crt`。

從用戶端工具和程式碼連線

Amazon Redshift 提供了 Amazon Redshift 查詢編輯器 v2 供您連線到叢集和工作群組。如需詳細資訊，請參閱使用查詢編輯器 v2 查詢資料庫。

本節提供一些可供第三方工具進行連線的選項。此外，還描述如何以程式設計方式連接至叢集。

主題

- 使用 Amazon Redshift RSQL 進行連線
- 使用 Amazon Redshift RSQL 連線至叢集
- Amazon Redshift RSQL 中繼命令

- Amazon Redshift RSQL 變數
- Amazon Redshift RSQL 錯誤代碼
- Amazon Redshift RSQL 環境變數

使用 Amazon Redshift RSQL 進行連線

Amazon Redshift RSQL 是用於與 Amazon Redshift 叢集和資料庫互動的命令列用戶端。您可以連線到 Amazon Redshift 叢集、描述資料庫物件、查詢資料，以及檢視各種輸出格式的查詢結果。

Amazon Redshift RSQL 支援 PostgreSQL psql 命令列工具的功能，以及一組專屬於 Amazon Redshift 的額外功能。這些索引標籤包括以下項目：

- 您可以使用 AD FS、PingIdentity、Okta、Azure ADm 或其他 SAML/JWT 型身分提供者，來使用單一登入身分驗證。您也可以使用以瀏覽器為基礎的 SAML 身分供應商來進行多重要素驗證 (MFA)。
- 您可以描述 Amazon Redshift 物件的屬性，例如資料表分佈索引鍵、資料表排序索引鍵、晚期繫結視觀表 (LBV) 和具體化視觀表。您也可以描述 AWS Glue 目錄或 Apache Hive 中繼存放區中外部資料表的屬性或屬性、Amazon RDS for PostgreSQL、Amazon Aurora PostgreSQL 相容版本、RDS for MySQL （預覽） 和 Amazon Aurora MySQL 相容版本 （預覽） 中的外部資料庫，以及使用 Amazon Redshift 資料共用的資料表。
- 您也可以使用增強的控制流程命令，例如 `IF` (`\ELSEIF`、`\ELSE`,、`\ENDIF`)、`\GOTO` 和 `\LABEL`。

透過 Amazon Redshift RSQL 批次模式 (該模式會執行以輸入參數形式傳遞的指令碼)，您可以執行同時包含 SQL 和複雜商業邏輯的指令碼。如果您有現有的自我管理內部部署資料倉儲，則可以使用 Amazon Redshift RSQL 取代現有的擷取、轉換、載入 (ETL) 和自動化指令碼，例如 Teradata BTEQ 指令碼。使用 RSQL 可以幫助您避免在程序語言中手動重新實作指令碼。

Amazon Redshift RSQL 適用於 Linux、Windows 和 macOS 作業系統。

若要報告 Amazon Redshift RSQL 的問題，請寫信至 `<redshift-rsql-support@amazon.com>`。

主題

- 開始使用 Amazon Redshift RSQL
- Amazon Redshift RSQL 變更日誌

開始使用 Amazon Redshift RSQL

在搭載 Linux、macOS 或 Microsoft Windows 作業系統的電腦上安裝 Amazon Redshift RSQL。

下載 RSQL

- Linux 64 位元 RPM：RSQL 1.1.2 版
  - Linux 成品簽章金鑰：金鑰
  - Linux 成品簽署雜湊：雜湊
- Mac OS 64 位元 PKG：RSQL 1.1.2 版
- Windows 64 位元 MSI：RSQL 1.1.2 版

若要查看舊版的變更日誌和下載，請瀏覽 Amazon Redshift RSQL 變更日誌。

安裝適用於 Linux 的 RSQL

請遵循下列步驟以安裝適用於 Linux 的 RSQL。

1. 使用以下命令安裝驅動程式管理員：

```
sudo yum install unixODBC
```

2. 安裝 ODBC 驅動程式：下載並安裝 Amazon Redshift ODBC 驅動器。
3. 將 ini 檔案複製到主目錄：

```
cp /opt/amazon/redshiftodbcx64/odbc.ini ~/.odbc.ini
```

4. 將環境變數設定為指向檔案的位置：

```
export ODBCINI=~/.odbc.ini
export ODBCSYSINI=/opt/amazon/redshiftodbcx64/
export AMAZONREDSHIFTODBCINI=/opt/amazon/redshiftodbcx64/amazon.redshiftodbc.ini
```

5. 您現在可以執行下列命令來安裝 RSQL。

```
sudo rpm -i AmazonRedshiftRsql-<version>.rhel.x86_64.rpm
```

安裝適用於 Mac 的 RSQL

請依照下列步驟安裝 RSQL for Mac OSX。

1. 使用以下命令安裝驅動程式管理員：

```
brew install unixodbc --build-from-source
```

2. 安裝 ODBC 驅動程式：下載並安裝 Amazon Redshift ODBC 驅動器。
3. 將 ini 檔案複製到主目錄：

```
cp /opt/amazon/redshift/Setup/odbc.ini ~/.odbc.ini
```

4. 將環境變數設定為指向檔案的位置：

```
export ODBCINI=~/.odbc.ini
export ODBCSYSINI=/opt/amazon/redshift/Setup
export AMAZONREDSHIFTODBCINI=/opt/amazon/redshift/lib/amazon.redshiftodbc.ini
```

5. 將 DYLD_LIBRARY_PATH 設定為 libodbc.dylib 的位置 (如果它不在 /usr/local/lib 的話)。

```
export DYLD_LIBRARY_PATH=$DYLD_LIBRARY_PATH:/usr/local/lib
```

6. 按兩下 pkg 檔案來執行安裝程式。
7. 依照安裝程式中的步驟完成安裝。同意授權合約中的條款。

安裝適用於 Windows 的 RSQL

請依照下列步驟安裝 RSQL for Windows。

1. 安裝 ODBC 驅動程式：下載並安裝 Amazon Redshift ODBC 驅動器。
2. 按兩下 RSQL 下載檔案以執行安裝程式，然後遵循提示以完成安裝。

Amazon Redshift RSQL 變更日誌

1.1.2 (2025-12-11)

錯誤修正

- 修正使用 \goto 和 \label 命令失敗的錯誤。
- 修正當變數以引號括住時，導致 RSQL 無法列印變數值的錯誤。
- 修正在啟用 UseDeclareFetch 時，查詢結果超過 ODBC DSN Fetch 參數大小時，導致 RSQL 損毀的錯誤。
- 修正即使開啟分頁器，仍會一次傳回多頁結果的問題。

- 修正在交易區塊內查詢失敗時觸發 RSQL 損毀的錯誤。

1.1.1 (2025-11-20)

錯誤修正

- 修正 RSQL 在使用 -c 旗標時錯誤剖析查詢的問題。此修正適用於所有平台。
- 修正 Mac 上導致使用者無法在 RSQL 中使用 \s 命令的錯誤。

1.1.0 (2025-11-11)

錯誤修正

- 已解決導致 rSQL 中意外當機的記憶體流失問題。
- 從 RSQL 移除 OpenSSL 相依性。
- 修正連結與相同環境中 libpq/psql 安裝的衝突。
- 已改善 Amazon Linux 2023、Windows 和 macOS 的平台相容性。
- 已修正輸出在超過目前顯示大小時會遭到截斷的問題。

1.0.8 (2023-06-19)

錯誤修正

- 修正了使用 SHOW 命令會截斷輸出的問題。
- 已新增對用於描述外部 Kinesis 串流和 Kafka 主題之 \de 的支援。

1.0.7 (2023-03-22)

錯誤修正

- 修正了 RSQL 無法描述具體化視觀表的問題。
- 修正了使用 Amazon Redshift Serverless 時，stl_connection_log 上的許可遭拒錯誤。
- 修正了 RSQL 可能無法正確處理 \GOTO 標籤的問題。
- 修正了 SSL 訊息會以靜音模式列印的問題。
- 修正了描述預存程序時會顯示隨機字元的問題。

- 修正了會列印重複 ERROR/INFO 訊息的問題。

新增

- RSQL 現在會直接從 ODBC 驅動程式取得 SSL 資訊。

1.0.6 (2023-02-21)

錯誤修正

- 在 Redshift 修補程式 1.0.46086 (P173) 上修正了 \d 會擲出錯誤 (整數的輸入語法無效: "xid") 的問題。

新增

- 已重新命名安裝檔案以反映支援的架構。

1.0.5 (2022-06-27)

錯誤修正

- 將 SQL 錯誤訊息傳送到標準錯誤 (stderr)。
- 修正了使用 ON_ERROR_STOP 時結束代碼所發生的問題。指令碼現在會在遇到錯誤後結束，並返回正確的結束代碼。
- Maxerror 現在不區分大小寫。

新增

- 已新增對 ODBC 2.x 驅動程式的支援。

1.0.4 (2022-03-19)

- 新增對 RSPASSWORD 環境變數的支援。設定密碼以連線到 Amazon Redshift。例如 `export RSPASSW0RD=TestPassw0rd`。

1.0.3 (2021-12-08)

錯誤修正

- 修正了在 Windows 作業系統中使用 \c 或 \logon 在資料庫之間進行切換時的快顯對話方塊。
- 修正了檢查 ssl 資訊時的損毀問題。

Amazon Redshift RSQL 的先前版本

請根據您的作業系統選擇其中一個連結，以下載您需要的 Amazon Redshift RSQL 版本。

Linux 64 位元 RPM

- RSQL 1.1.1 版
- RSQL 1.1.0 版
- RSQL 1.0.8 版
- RSQL 1.0.7 版
- RSQL 1.0.6 版
- RSQL 1.0.5 版
- RSQL 1.0.4 版
- RSQL 1.0.3 版
- RSQL 1.0.1 版

Mac OS 64 位元 DMG/PKG

- RSQL 1.1.1 版
- RSQL 1.1.0 版
- RSQL 1.0.8 版
- RSQL 1.0.7 版
- RSQL 1.0.6 版
- RSQL 1.0.5 版
- RSQL 1.0.4 版
- RSQL 1.0.3 版
- RSQL 1.0.1 版

Windows 64 位元 MSI

- RSQL 1.1.1 版
- RSQL 1.1.0 版
- RSQL 1.0.8 版
- RSQL 1.0.7 版
- RSQL 1.0.6 版
- RSQL 1.0.5 版
- RSQL 1.0.4 版
- RSQL 1.0.3 版
- RSQL 1.0.1 版

使用 Amazon Redshift RSQL 連線至叢集

透過 Amazon Redshift，您可以使用 RSQL 連線到叢集並與其互動。這是一種命令列工具，提供了安全的方法來查詢資料、建立資料庫物件及管理 Amazon Redshift 叢集。下列各節將引導您完成在使用和不使用資料來源名稱 (DSN) 搭配 RSQL 的情況下，建立與叢集的連線的步驟。

在不使用 DSN 的情況下進行連線

1. 在 Amazon Redshift 主控台上，選擇要連線到的叢集，並記下端點、資料庫和連接埠。
2. 在命令提示字元中，使用命令列參數指定連線資訊。

```
rsql -h <endpoint> -U <username> -d <databasename> -p <port>
```

這裡會套用下列項目：

- *<endpoint>* 是您在前一步驟中記錄的 Endpoint (端點)。
- *<username>* 是有許可而可以連線至叢集的使用者名稱。
- *<databasename>* 是您在前一步驟中記錄的 Database Name (資料庫名稱)。
- *<port>* 是您在上一個步驟中記錄的連接埠。*<port>* 是選用參數。

範例如下。

```
rsql -h testcluster.example.amazonaws.com -U user1 -d dev -p 5439
```

3. 在密碼提示中，輸入 *<username>* 使用者的密碼。

   成功連線的回應看起來如下。

```
% 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=#
```

用來連線的命令在 Linux、Mac OS 和 Windows 上具有相同的參數。

在使用 DSN 的情況下進行連線

您可以使用 DSN 將 RSQL 連線至 Amazon Redshift，以簡化連線屬性的組織。本主題包含 ODBC 驅動器安裝的指示，以及 DSN 屬性的說明。

使用具有密碼的 DSN 連線

以下示範使用密碼的 DSN 連線組態範例。Mac OSX 的預設 <path to driver> 是 /opt/amazon/redshift/lib/libamazonredshiftodbc.dylib，Linux 的則是 /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
```

以下輸出來自成功的連線。

```
% 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=#
```

使用單一登入 DSN

您可以設定 DSN 以進行單一登入身分驗證。以下示範使用 Okta 單一登入的 DSN 連線組態範例。

```
[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>
```

成功連線的範例輸出。

```
% 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=#
```

以下範例示範使用 Azure 單一登入的 DSN 連線組態範例。

```
[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>
```

搭配 IAM 設定檔使用 DSN 連線

您可以使用所設定的 IAM 設定檔連線到 Amazon Redshift。IAM 設定檔必須具有呼叫 `GetClusterCredentials` 的權限。以下範例示範要使用的 DSN 屬性。只有在 `Host` 不是 Amazon 提供的端點 (例如 `examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com`) 時，才需要 `ClusterID` 和 `Region` 參數。

```
[testiam]
Driver=Default
Host=testcluster.example.com
Database=dev
DbUser=testuser
ClusterID=rsqltestcluster
Region=us-east-1
IAM=1
Profile=default
```

Profile 金鑰的值是您從 CLI AWS 登入資料中選擇的具名設定檔。本範例示範名為 default 之設定檔的憑證。

```
$ cat .aws/credentials
[default]
aws_access_key_id = ASIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

以下示範連線回應。

```
$ 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=>
```

搭配執行個體設定檔使用 DSN 連線

您可以使用 Amazon EC2 執行個體設定檔連線到 Amazon Redshift。執行個體設定檔必須具有呼叫 GetClusterCredentials 的權限。如需要使用的 DSN 屬性，請參閱以下範例。只有在 Host 不是 Amazon 提供的端點 (例如 examplecluster.abc123xyz789.us-west-2.redshift.amazonaws.com) 時，才需要 ClusterID 和 Region 參數。

```
[testinstanceprofile]
Driver=Default
Host=testcluster.example.com
Database=dev
DbUser=testuser
ClusterID=rsqltestcluster
Region=us-east-1
IAM=1
Instanceprofile=1
```

以下示範連線回應。

```
$ 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=>
```

搭配使用 DSN 連線與預設憑證提供者鏈結

若要使用預設登入資料提供者鏈結進行連線，請僅指定 IAM 屬性，Amazon Redshift RSQL 將嘗試按照適用於 Java 的 AWS SDK 中的使用登入資料中所述的順序取得 AWS 登入資料。鏈結中至少要有一個提供者必須具有 `GetClusterCredentials` 許可。這對於從 ECS 容器進行連線 (舉例來說) 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 中繼命令

Amazon Redshift RSQL 中繼命令會傳回有關資料庫或特定資料庫物件的資訊記錄。結果中可以包含各種資料行和中繼資料。其他命令會執行特定動作。這些命令的前面會加上反斜線。

\d[S+]

列出本機使用者建立的資料表、一般視觀表、晚期繫結視觀表及具體化視觀表。\dS 也會列出資料表和視觀表 (例如 \d)，但系統物件包含在傳回的記錄中。+ 會導致所有列出的物件產生額外的中繼資料資料行 `description`。以下示範執行命令所傳回的範例記錄。

```
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

描述資料表、視觀表或索引。包括資料行名稱和類型。其還會提供 diststyle、備份組態、建立日期 (2018 年 10 月之後建立的資料表) 和限制。例如，\dS+ sample 會傳回物件屬性。附加 S+ 會導致所傳回的記錄中包含額外資料行。

```
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)
```

資料表的分佈樣式 (簡稱 Diststyle) 可以是 KEY、AUTO、EVEN 或 ALL。

Backup 會指出在擷取快照時是否有備份資料表。有效值為 YES 或 NO。

Created 是建立資料表時的時間戳記。建立日期不適用於 2018 年 11 月之前建立的 Amazon Redshift 資料表。在此日期之前建立的資料表會顯示 n/a (不適用)。

Unique Constraints 會列出資料表上唯一的和主索引鍵的限制。

Foreign-key constraints 會列出資料表上的外部索引鍵限制。

\dC[+] [PATTERN]

列出轉換。包括來源類型、目標類型，以及轉換是否為隱含。

以下示範 \dC+ 的結果子集。

```
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]

顯示其他地方未顯示的物件描述。

\de

列出外部資料表。這包括 中的資料表 AWS Glue Data Catalog、Hive 中繼存放區和來自 Amazon RDS/Aurora MySQL、Amazon RDS/Aurora PostgreSQL 和 Amazon Redshift 資料共用資料表的聯合資料表。

\de NAME

描述外部資料表。

下列範例顯示 AWS Glue 外部資料表。

```
# \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 資料表。

```
# \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 外部資料表。

```
# \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]

列出各種類型的函數。例如，命令 \df 會傳回函數清單。結果包括名稱、傳回的資料類型、存取權限和其他中繼資料等屬性。函數類型可以包括觸發條件、預存程序、視窗函數和其他類型。當您附加 S+ 至命令時 (例如 \dfantS+)，其中會包括其他中繼資料資料行，例如 owner、security 和 access privileges。

\dL[S+] [PATTERN]

列出與資料庫相關聯之程序語言的相關資料。資訊中會包括名稱 (例如 plpgsql) 和其他中繼資料，其中包括其是否受信任、存取權限和描述。例如，範例呼叫是會列出語言及其屬性的 \dLS+。當您附加 S+ 至命令時，其中會包括其他中繼資料資料行，例如 call handler 和 access privileges。

範例結果：

```
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]

列出具體化視觀表。例如，\dmS+ 會列出具體化視觀表及其屬性。當您附加 S+ 至命令時，其中會包括其他中繼資料資料行。

\dn[S+] [PATTERN]

列出結構描述。當您附加 S+ 至命令時 (例如 \dnS+)，其中會包括其他中繼資料資料行，例如 description 和 access privileges。

\dp [PATTERN]

列出資料表、視觀表和序列存取權限。

\dt[S+] [PATTERN]

列出資料表。當您附加 S+ 至命令時 (例如 \dtS+)，其中會包括其他中繼資料資料行，例如本案例中的 description。

\du

列出資料庫的使用者。包括其名稱及其角色 (例如超級使用者) 和屬性。

\dv[S+] [PATTERN]

列出視觀表。包括結構描述、類型和擁有者資料。當您附加 S+ 至命令時 (例如 \dvS+)，其中會包括其他中繼資料資料行。

\H

開啟 HTML 輸出。這對於快速傳回格式化結果非常有用。例如，select * from sales; \H 會以 HTML 格式傳回銷售資料表的結果。若要切換回表格式結果，請使用 \q 或靜音。

\i

從檔案執行命令。例如，假設工作目錄中有 rsql_steps.sql，以下內容會執行檔案中的命令：\i rsql_steps.sql。

\l[+] [PATTERN]

列出資料庫。包括擁有者、編碼和其他資訊。

\q

結束或 \q 命令會登出資料庫工作階段並結束 RSQL。

\sv[+] VIEWNAME

顯示視觀表的定義。

\timing

例如，顯示查詢的執行時間。

\z [PATTERN]

與 \dp 相同的輸出。

\?

顯示說明資訊。選用參數會指定要解釋的項目。

\EXIT

登出所有資料庫工作階段並結束 Amazon Redshift RSQL。此外，您也可以指定選用的結束代碼。例如，\EXIT 15 會結束 Amazon Redshift RSQL 終端並傳回結束代碼 15。

以下範例示範從連線輸出以及從 RSQL 結束。

```
% 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 RSQL 用來儲存後續 SQL SELECT 陳述式所傳回之資料庫資訊的匯出檔案名稱。

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
```

主控台輸出

```
Rformat is off.
Target width is 1500.
```

```
Heading is set to: General Title
Titledashes is on.
(exported 40 rows)
```

\LOGON

連線至資料庫。您可以使用位置語法或連線字串形式指定連線參數。

命令語法如下：\logon {[DBNAME|- USERNAME|- HOST|- PORT|- [PASSWORD]] | conninfo}

DBNAME 是所連線的資料庫名稱。USERNAME 是連線時所用身分的使用者名稱。預設的 HOST 是 localhost。預設的 PORT 是 5439。

在 \LOGON 命令中指定主機名稱時，其會變成其他 \LOGON 命令的預設主機名稱。若要變更預設主機名稱，請在另外的 \LOGON 命令中指定新的 HOST。

來自 user1 之 \LOGON 命令的範例輸出如下。

```
(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 的範例輸出。

```
(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 命令的延伸。\REMARK 會將指定字串列印至輸出串流。\REMARK 會透過新增將輸出換行到不同行的能力來延伸 \echo。

以下範例示範命令的輸出。

```
(testcluster) user1@dev=# \remark 'hello//world'
hello
world
```

\RSET

命令 \rset 會設定命令參數和變數。\rset 同時具有互動式模式和批次模式。其不支援將選項作為 bash 選項，如 -x 或引數，例如 --<arg>。

其會設定變數，如下所示：

- ERRORLEVEL
- HEADING 和 RTITLE
- RFORMAT
- MAXERROR
- TITLEDASHES
- WIDTH

以下範例會指定標題。

```
\rset heading "Winter Sales Report"
```

如需更多如何使用 \rset 的範例，您可以在 Amazon Redshift RSQL 變數 主題中找到幾個。

\RUN

執行指定檔案中包含的 Amazon Redshift RSQL 指令碼。\RUN 會透過新增選項來擴展 \i 命令，以略過檔案中的標題行。

如果檔案名稱包含逗號、分號或空格，請以單引號將其括住。此外，如果檔案名稱後面有文字，請以引號將其括住。在 UNIX 中，檔案名稱會區分大小寫。在 Windows 中，檔案名稱不區分大小寫。

以下範例示範命令的輸出。

```
(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

\! 命令的別名。\OS 會執行以參數的形式傳遞的作業系統命令。執行命令後，控制項會回到 Amazon Redshift RSQL。例如，您可以執行以下命令來列印目前的系統日期時間，並返回 RSQL 終端：\os date。

```
(testcluster) user1@dev=# \os date
Tue Sep 7 20:47:54 UTC 2021
```

\GOTO

Amazon Redshift RSQL 的新命令。\GOTO 會略過所有中間命令，並在指定的 \LABEL 繼續處理。\LABEL 必須是前向參照。你不能跳到詞法上在 \GOTO 之前的 \LABEL。

下面顯示了範例輸出。

```
(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

Amazon Redshift RSQL 的新命令。\LABEL 會建立用於執行程式的進入點，以作為 \GOTO 命令的目標。

以下示範命令的範例輸出。

```
(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 和相關命令會有條件地執行輸入指令碼的某些部分。PSQL \if (\elif、\else、\endif) 命令的延伸。\IF 和 \ELSEIF 支援布林運算式，包括 AND、OR 和 NOT 條件。

以下示範命令的範例輸出。

```
(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
```

在您的分支邏輯中使用 ERRORCODE。

```
\if :'ERRORCODE' = '00000'
    \remark 'The statement was executed without error'
\else
    \remark :LAST_ERROR_MESSAGE
```

```
\endif
```

在 \IF 區塊內使用 \GOTO 來控制程式碼的執行方式。

Amazon Redshift RSQL 變數

某些關鍵字會作為 RSQL 中的變數。您可以將每個關鍵字設定為特定值，或重設該值。大多數關鍵字會使用具有互動式模式和批次模式的 \rset 來進行設定。命令可以用小寫或大寫來定義。

ACTIVITYCOUNT

指出上次提交的請求所影響的資料列數目。若為資料傳回請求，這是從資料庫傳回到 RSQL 的資料列數目。此值是 0 或正整數。最大值是 18,446,744,073,709,551,615。

特殊對待的變數 ACTIVITYCOUNT 類似於變數 ROW_COUNT。不過，ROW_COUNT 不會在 SELECT、COPY 或 UNLOAD 的命令完成時，向用戶端應用程式報告受影響的資料列計數。但 ACTIVITYCOUNT 會。

activitycount_01.sql：

```
select viewname, schemaname
from pg_views
where schemaname = 'not_existing_schema';
\if :ACTIVITYCOUNT = 0
\remark 'views do not exist'
\endif
```

主控台輸出：

```
viewname | schemaname
----------+------------
(0 rows)

views do not exist
```

ERRORLEVEL

指定錯誤的嚴重性等級。使用嚴重性等級可決定行動方式。如果尚未使用 ERRORLEVEL 命令，則其值預設會是 ON。

errorlevel_01.sql：

```
\rset errorlevel 42P01 severity 0

select * from tbl;

select 1 as col;

\echo exit
\quit
```

主控台輸出：

```
Errorlevel is on.
rsql: ERROR: relation "tbl" does not exist
(1 row)

col
1

exit
```

HEADING 和 RTITLE

可讓使用者指定顯示在報告頂端的標頭。RSET RTITLE 命令指定的標頭會自動包含用戶端電腦的目前系統日期。

rset_heading_rtitle_02.rsql content：

```
\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...
```

主控台輸出：

```
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 會終止任務處理的最大錯誤嚴重性等級。傳回代碼是 RSQL 在完成每項任務之後傳回給用戶端作業系統的整數值。傳回代碼的值會指出任務的完成狀態。如果指令碼包含的陳述式所產生的錯誤嚴重性等級大於指定的 maxerror 值，則 RSQL 會立即結束。因此，若要讓 RSQL 在錯誤嚴重性等級為 8 時結束，請使用 RSET MAXERROR 7。

maxerror_01.sql content：

```
\rset maxerror 0

select 1 as col;

\quit
```

主控台輸出：

```
Maxerror is default.
(1 row)

col
1
```

RFORMAT

可讓使用者指定是否要套用格式化命令的設定。

rset_rformat.rsql content：

```
\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...
```

主控台輸出：

```
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

取得受先前查詢影響的記錄數目。其通常用來檢查結果，如下面的程式碼片段：

```
SET result = ROW_COUNT;

IF result = 0
...
```

TITLEDASHES

此控制項可讓使用者指定是否要在針對 SQL 陳述式所傳回的資料行資料上方列印一行破折號字元。

範例：

```
\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;
```

主控台輸出：

```
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

將輸出格式設定為換行，並指定報告中每一行的目標寬度。如果沒有參數，則會傳回格式和目標寬度的目前設定。

rset_width_01.rsql content：

```
\echo Starting...
\rset width
\rset width 50
\rset width
\quit
\echo Finishing...
```

主控台輸出：

```
Starting...
Target width is 75.
Target width is 50.
Target width is 50.
Press any key to continue . . .
```

有參數的範例：

```
\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...
```

主控台輸出：

```
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 錯誤代碼

成功訊息、警告和例外狀況：

錯誤程式碼	錯誤類別	狀況名稱
00000	類別 00 — 成功完成	successful_completion
01000	類別 01 — 警告	warning
0100C	類別 01 — 警告	dynamic_result_sets_returned
01008	類別 01 — 警告	implicit_zero_bit_padding
01003	類別 01 — 警告	null_value_eliminated_in_set_function
01007	類別 01 — 警告	privilege_not_granted
01006	類別 01 — 警告	privilege_not_revoked
01004	類別 01 — 警告	string_data_right_truncation

錯誤程式碼	錯誤類別	狀況名稱
01P01	類別 01 — 警告	deprecated_feature
02000	類別 02 — 沒有資料	no_data
02001	類別 02 — 沒有資料	no_additional_dynamic_result_sets_returned
03000	類別 03 — SQL 陳述式尚未完成	sql_statement_not_yet_complete
08000	類別 08 — 連線例外狀況	connection_exception
08003	類別 08 — 連線例外狀況	connection_does_not_exist
08006	類別 08 — 連線例外狀況	connection_failure
08001	類別 08 — 連線例外狀況	sqlclient_unable_to_establish_sqlconnection
08004	類別 08 — 連線例外狀況	sqlserver_rejected_establishment_of_sqlconnection
08007	類別 08 — 連線例外狀況	transaction_resolution_unknown
08P01	類別 08 — 連線例外狀況	protocol_violation
09000	類別 09 — 觸發的動作例外狀況	triggered_action_exception
0A000	類別 0A — 功能不受支援	feature_not_supported
0A000	類別 0A — 功能不受支援	feature_not_supported
0B000	類別 0B — 無效的交易啟動	invalid_transaction_initiation
0F000	類別 0F — 定位器例外狀況	locator_exception
0F001	類別 0F — 定位器例外狀況	invalid_locator_specification
0L000	類別 0L — 無效的授與者	invalid_grantor
0LP01	類別 0L — 無效的授與者	invalid_grant_operation

錯誤程式碼	錯誤類別	狀況名稱
0P000	類別 0P — 無效的角色規格	invalid_role_specification
0Z000	類別 0Z — 診斷例外狀況	diagnostics_exception
0Z002	類別 0Z — 診斷例外狀況	stacked_diagnostics_accessed_without_active_handler
20000	類別 20 — 找不到案例	case_not_found
21000	類別 21 — 基數違規	cardinality_violation

資料例外狀況：

錯誤程式碼	錯誤類別	狀況名稱
22000	類別 22 — 資料例外狀況	data_exception
2202E	類別 22 — 資料例外狀況	array_subscript_error
22021	類別 22 — 資料例外狀況	character_not_in_repertoire
22008	類別 22 — 資料例外狀況	datetime_field_overflow
22012	類別 22 — 資料例外狀況	division_by_zero
22005	類別 01 — 警告	error_in_assignment
2200B	類別 01 — 警告	escape_character_conflict
22022	類別 01 — 警告	indicator_overflow
22015	類別 01 — 警告	interval_field_overflow
2201E	類別 01 — 警告	invalid_argument_for_logarithm
2201F	類別 01 — 警告	invalid_argument_for_power_function
2201G	類別 01 — 警告	invalid_argument_for_width_bucket_function

錯誤程式碼	錯誤類別	狀況名稱
22018	類別 01 — 警告	invalid_character_value_for_cast
22007	類別 01 — 警告	invalid_datetime_format
22019	類別 01 — 警告	invalid_escape_character
2200D	類別 01 — 警告	invalid_escape_octet
22025	類別 01 — 警告	invalid_escape_sequence
22P06	類別 01 — 警告	nonstandard_use_of_escape_character
22010	類別 01 — 警告	invalid_indicator_parameter_value
22023	類別 01 — 警告	invalid_parameter_value
2201B	類別 01 — 警告	invalid_regular_expression
22009	類別 01 — 警告	invalid_time_zone_displacement_value
2200C	類別 01 — 警告	invalid_use_of_escape_character
2200G	類別 01 — 警告	most_specific_type_mismatch
22004	類別 01 — 警告	null_value_not_allowed
22002	類別 01 — 警告	null_value_no_indicator_parameter
22003	類別 01 — 警告	numeric_value_out_of_range
22026	類別 01 — 警告	string_data_length_mismatch
22001	類別 01 — 警告	string_data_right_truncation
22011	類別 01 — 警告	substring_error
22027	類別 01 — 警告	trim_error
22024	類別 01 — 警告	unterminated_c_string

錯誤程式碼	錯誤類別	狀況名稱
2200F	類別 01 — 警告	zero_length_character_string
22P01	類別 01 — 警告	floating_point_exception
22P02	類別 01 — 警告	invalid_text_representation
22P03	類別 01 — 警告	invalid_binary_representation
22P04	類別 01 — 警告	bad_copy_file_format
22P05	類別 01 — 警告	untranslatable_character

完整性限制違規：

錯誤程式碼	錯誤類別	狀況名稱
23000	類別 23 — 完整性限制違規	integrity_constraint_violation
23001	類別 23 — 完整性限制違規	restrict_violation
23502	類別 23 — 完整性限制違規	not_null_violation
23503	類別 23 — 完整性限制違規	foreign_key_violation
23505	類別 23 — 完整性限制違規	unique_violation
23514	類別 23 — 完整性限制違規	check_violation
24000	類別 24 — 無效的游標狀態	invalid_cursor_state
01004	類別 01 — 警告	string_data_right_truncation
25000	類別 25 — 無效的交易狀態	invalid_transaction_state
25001	類別 25 — 無效的交易狀態	active_sql_transaction
25002	類別 25 — 無效的交易狀態	invalid_transaction_state

錯誤程式碼	錯誤類別	狀況名稱
25008	類別 25 — 無效的交易狀態	held_cursor_requires_same_isolation_level
25003	類別 25 — 無效的交易狀態	inappropriate_access_mode_for_branch_transaction
25004	類別 25 — 無效的交易狀態	inappropriate_isolation_level_for_branch_transaction
25005	類別 25 — 無效的交易狀態	no_active_sql_transaction_for_branch_transaction
25006	類別 25 — 無效的交易狀態	read_only_sql_transaction
25007	類別 25 — 無效的交易狀態	no_active_sql_transaction_for_branch_transaction
25P01	類別 25 — 無效的交易狀態	no_active_sql_transaction
25P02	類別 25 — 無效的交易狀態	in_failed_sql_transaction
26000	類別 26 — 無效的 SQL 陳述式名稱	invalid_sql_statement_name
28000	類別 28 — 無效的授權規格	invalid_authorization_specification
2B000	類別 2B — 仍存在相依權限描述項	dependent_privilege_descriptors_still_exist
2BP01	類別 2B — 仍存在相依權限描述項	dependent_objects_still_exist
2D000	類別 2D — 無效的交易終止	invalid_transaction_termination
2F000	類別 2F — SQL 常式例外狀況	sql_routine_exception
2F005	類別 2F — SQL 常式例外狀況	function_executed_no_return_statement
2F002	類別 2F — SQL 常式例外狀況	modifying_sql_data_not_permitted

錯誤程式碼	錯誤類別	狀況名稱
2F003	類別 2F — SQL 常式例外狀況	prohibited_sql_statement_attempted
2F004	類別 2F — SQL 常式例外狀況	reading_sql_data_not_permitted
34000	類別 34 — 無效的游標名稱	invalid_cursor_name
38000	類別 38 — 外部常式例外狀況	external_routine_exception
38001	類別 38 — 外部常式例外狀況	containing_sql_not_permitted
38002	類別 38 — 外部常式例外狀況	modifying_sql_data_not_permitted
38003	類別 38 — 外部常式例外狀況	prohibited_sql_statement_attempted
38004	類別 38 — 外部常式例外狀況	reading_sql_data_not_permitted
39000	類別 39 — 外部常式調用例外狀況	external_routine_invocation_exception
39001	類別 39 — 外部常式調用例外狀況	invalid_sqlstate_returned
39004	類別 39 — 外部常式調用例外狀況	null_value_not_allowed
39P01	類別 39 — 外部常式調用例外狀況	trigger_protocol_violated
39P02	類別 39 — 外部常式調用例外狀況	srf_protocol_violated
3D000	類別 3D — 無效的目錄名稱	invalid_catalog_name
3F000	類別 3F — 無效的結構描述名稱	invalid_schema_name
42000	類別 42 — 語法錯誤或存取規則違規	syntax_error_or_access_rule_violation

錯誤程式碼	錯誤類別	狀況名稱
42601	類別 42 — 語法錯誤或存取規則違規	syntax_error
42501	類別 42 — 語法錯誤或存取規則違規	insufficient_privilege
42846	類別 42 — 語法錯誤或存取規則違規	cannot_coerce
42803	類別 42 — 語法錯誤或存取規則違規	grouping_error
42830	類別 42 — 語法錯誤或存取規則違規	invalid_foreign_key
42602	類別 42 — 語法錯誤或存取規則違規	invalid_name
42622	類別 42 — 語法錯誤或存取規則違規	name_too_long
42939	類別 42 — 語法錯誤或存取規則違規	reserved_name
42804	類別 42 — 語法錯誤或存取規則違規	datatype_mismatch
42P18	類別 42 — 語法錯誤或存取規則違規	indeterminate_datatype
42809	類別 42 — 語法錯誤或存取規則違規	wrong_object_type
42703	類別 42 — 語法錯誤或存取規則違規	undefined_column
42883	類別 42 — 語法錯誤或存取規則違規	undefined_function

錯誤程式碼	錯誤類別	狀況名稱
42P01	類別 42 — 語法錯誤或存取規則違規	undefined_table
42P02	類別 42 — 語法錯誤或存取規則違規	undefined_parameter
42704	類別 42 — 語法錯誤或存取規則違規	undefined_object
42701	類別 42 — 語法錯誤或存取規則違規	duplicate_column
42P03	類別 42 — 語法錯誤或存取規則違規	duplicate_cursor
42P04	類別 42 — 語法錯誤或存取規則違規	duplicate_database
42723	類別 42 — 語法錯誤或存取規則違規	duplicate_function
42P05	類別 42 — 語法錯誤或存取規則違規	duplicate_prepared_statement
42P06	類別 42 — 語法錯誤或存取規則違規	duplicate_schema
42P07	類別 42 — 語法錯誤或存取規則違規	duplicate_table
42712	類別 42 — 語法錯誤或存取規則違規	duplicate_alias
42710	類別 42 — 語法錯誤或存取規則違規	duplicate_object
42702	類別 42 — 語法錯誤或存取規則違規	ambiguous_column

錯誤程式碼	錯誤類別	狀況名稱
42725	類別 42 — 語法錯誤或存取規則違規	ambiguous_function
42P08	類別 42 — 語法錯誤或存取規則違規	ambiguous_parameter
42P09	類別 42 — 語法錯誤或存取規則違規	ambiguous_alias
42P10	類別 42 — 語法錯誤或存取規則違規	invalid_column_reference
42611	類別 42 — 語法錯誤或存取規則違規	invalid_column_definition
42P11	類別 42 — 語法錯誤或存取規則違規	invalid_cursor_definition
42P12	類別 42 — 語法錯誤或存取規則違規	invalid_database_definition
42P13	類別 42 — 語法錯誤或存取規則違規	invalid_function_definition
42P14	類別 42 — 語法錯誤或存取規則違規	invalid_prepared_statement_definition
42P15	類別 42 — 語法錯誤或存取規則違規	invalid_schema_definition
42P16	類別 42 — 語法錯誤或存取規則違規	invalid_table_definition
42P17	類別 42 — 語法錯誤或存取規則違規	invalid_object_definition
44000	類別 44 — WITH CHECK OPTION 違規	with_check_option_violation

錯誤程式碼	錯誤類別	狀況名稱
53000	類別 53 — 資源不足	insufficient_resources
53100	類別 53 — 資源不足	disk_full
53200	類別 53 — 資源不足	out_of_memory
53300	類別 53 — 資源不足	too_many_connections
54000	類別 54 — 超過計劃限制	program_limit_exceeded
54001	類別 54 — 超過計劃限制	statement_too_complex
54011	類別 54 — 超過計劃限制	too_many_columns
54023	類別 54 — 超過計劃限制	too_many_arguments
55000	類別 55 — 物件未處於先決條件狀態	object_not_in_prerequisite_state
55006	類別 55 — 物件未處於先決條件狀態	object_in_use
55P02	類別 55 — 物件未處於先決條件狀態	cant_change_runtime_param
55P03	類別 55 — 物件未處於先決條件狀態	lock_not_available
57000	類別 57 — 操作員介入	operator_intervention
57014	類別 57 — 操作員介入	query_canceled
57P01	類別 57 — 操作員介入	admin_shutdown
57P02	類別 57 — 操作員介入	crash_shutdown
57P03	類別 57 — 操作員介入	cannot_connect_now

錯誤程式碼	錯誤類別	狀況名稱
58000	類別 58 — 系統錯誤 (PostgreSQL 外部的錯誤)	system_error
58030	類別 58 — 系統錯誤 (PostgreSQL 外部的錯誤)	io_error
58P01	類別 58 — 系統錯誤 (PostgreSQL 外部的錯誤)	undefined_file
58P02	類別 58 — 系統錯誤 (PostgreSQL 外部的錯誤)	duplicate_file
F0000	類別 F0 — 組態檔案錯誤	duplicate_file
F0001	類別 F0 — 組態檔案錯誤	lock_file_exists
P0000	類別 P0 — PL/pgSQL 錯誤	plpgsql_error
P0001	類別 P0 — PL/pgSQL 錯誤	raise_exception
P0002	類別 P0 — PL/pgSQL 錯誤	no_data_found
P0003	類別 P0 — PL/pgSQL 錯誤	too_many_rows
XX000	類別 XX — 內部錯誤	internal_error
XX001	類別 XX — 內部錯誤	data_corrupted
XX002	類別 XX — 內部錯誤	index_corrupted

Amazon Redshift RSQL 環境變數

Amazon Redshift RSQL 可以使用環境變數來選取預設的參數值。

RSPASSWORD

> **Important**
>
> 基於安全考量，不建議您使用此環境變數，因為某些作業系統會允許非系統管理使用者查看處理序環境變數。

設定 Amazon Redshift RSQL 在連線至 Amazon Redshift 時要使用的密碼。此環境變數需要 Amazon Redshift RSQL 1.0.4 及以上版本。

如果設定了 RSPASSWORD，RSQL 會優先使用 RSPASSWORD。如果未設定 RSPASSWORD，而且您使用 DSN 進行連線，則 RSQL 會從 DSN 檔案的參數中取得密碼。最後，如果未設定 RSPASSWORD，而且您沒有使用 DSN，則 RSQL 會在嘗試連線後提供密碼提示。

以下是設定 RSPASSWORD 的範例：

```
export RSPASSWORD=TestPassw0rd
```

使用身分驗證設定檔連線至 Amazon Redshift

如果您有許多 Amazon Redshift 連線，則可能難以管理所有連線的設定。每個 JDBC 或 ODBC 連線通常會使用特定的組態選項。透過使用身分驗證設定檔，您可以將連線選項儲存在一起。如此一來，使用者就可以選擇要用來連線的設定檔，而不必管理個別選項的設定。設定檔可以套用至各種案例和使用者類型。

在您建立身分驗證設定檔後，使用者便可以將已可使用的設定檔新增至連線字串。如此一來，其便可以使用每個角色和使用案例的正確設定來連線至 Amazon Redshift。

如需 Amazon Redshift API 資訊，請參閱 CreateAuthenticationProfile。

建立身分驗證設定檔

使用 AWS CLI，您可以使用 `create-authentication-profile`命令建立身分驗證設定檔。這會假設您有現有的 Amazon Redshift 叢集和現有的資料庫。您的憑證必須具有可連線至 Amazon Redshift 資料庫的許可，以及可擷取身分驗證設定檔的權限。您可以透過 JSON 字串的形式提供組態選項，也可以參考包含 JSON 字串的檔案。

```
create-authentication-profile --authentication-profile-name<value: String> --
authentication-profile-content<value: String>
```

以下範例會建立名為 ExampleProfileName 的設定檔。在這裡，您可以將定義叢集名稱和其他選項設定的金鑰和值新增為 JSON 字串。

```
create-authentication-profile --authentication-profile-name "ExampleProfileName"
 --authentication-profile-content "{\"AllowDBUserOverride\":\"1\",\"Client_ID
\":\"ExampleClientID\",\"App_ID\":\"ExampleAppID\",\"AutoCreate\":false,
\"enableFetchRingBuffer\":true,\"databaseMetadataCurrentDbOnly\":true}"
}
```

此命令會使用指定的 JSON 設定建立設定檔。系統會傳回以下內容，以指出您已建立設定檔。

```
{"AuthenticationProfileName": "ExampleProfileName",
"AuthenticationProfileContent": "{\"AllowDBUserOverride\":\"1\",
\"Client_ID\":\"ExampleClientID\",\"App_ID\":\"ExampleAppID\",
\"AutoCreate\":false,\"enableFetchRingBuffer\":true,
\"databaseMetadataCurrentDbOnly\":true}" }
```

建立身分驗證設定檔的限制和配額

每位客戶的配額為十 (10) 個身分驗證設定檔。

身分驗證設定檔可能會發生某些錯誤。例如，如果您使用現有名稱建立新的設定檔，或是您超過設定檔配額的話。如需詳細資訊，請參閱 CreateAuthenticationProfile。

您無法在身分驗證設定檔存放區中儲存 JDBC、ODBC 和 Python 連線字串的某些選項金鑰和值：

- AccessKeyID
- access_key_id
- SecretAccessKey
- secret_access_key_id
- PWD
- Password
- password

您無法在設定檔存放區中儲存 JDBC 或 ODBC 連線字串的金鑰或值 AuthProfile。若為 Python 連線，您無法儲存 auth_profile。

身分驗證設定檔存放在 Amazon DynamoDB 中並由 管理 AWS。

使用身分驗證設定檔連線

建立身分驗證設定檔之後，您可以包含設定檔名稱做為 JDBC 2.x 版的連線選項。 AuthProfile使用此連線選項會擷取已儲存的設定。

```
jdbc:redshift:iam://endpoint:port/database?AuthProfile=<Profile-
Name>&AccessKeyID=<Caller-Access-Key>&SecretAccessKey=<Caller-Secret-Key>
```

以下是 JDBC URL 字串範例。

```
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，以及身分驗證設定檔名稱。

您也可以使用分號分隔符號來分隔組態選項，例如以下範例 (其包含記錄的選項)。

```
jdbc:redshift:iam://my_redshift_end_point:5439/dev?LogLevel=6;LogPath=/
tmp;AuthProfile=my_profile;AccessKeyID="AKIAIOSFODNN7EXAMPLE";SecretAccessKey="wJalrXUtnFEMI/
K7MDENG/bPxRfiCYEXAMPLEKEY"
```

> **Note**
>
> 請勿將機密資訊新增至身分驗證設定檔。例如，請勿在身分驗證設定檔中儲存 AccessKeyID 或 SecretAccessKey 值。身分驗證設定檔存放區具有禁止儲存秘密金鑰的規則。如果您嘗試儲存與敏感資訊相關聯的金鑰和值，就會收到錯誤。

取得身分驗證設定檔

若要列出現有的身分驗證設定檔，請呼叫以下命令。

```
describe-authentication-profiles --authentication-profile-name <value: String>
```

以下範例示範所擷取的兩個設定檔。如果未指定設定檔名稱，則會傳回所有設定檔。

```
{ "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}" } ] }
```

針對 Amazon Redshift 中的連線問題進行疑難排解

如果從 SQL 用戶端工具連接至叢集時有問題，您可以檢查幾個項目來縮小問題的範圍。如果您使用 SSL 或伺服器憑證，對連線問題進行故障排除時，請先移除此複雜度。然後在找到解決方案之後加回來。如需詳細資訊，請參閱設定連線的安全選項。

如需 Amazon Redshift 功能中可能影響應用程式的行為變更相關資訊，請參閱 Amazon Redshift 中的行為變更。

> **Important**
>
> Amazon Redshift 已變更 SSL 憑證的管理方式。如果您無法使用 SSL 來連接，您可能需要更新目前的信任根 CA 憑證。如需詳細資訊，請參閱將 SSL 連線轉換為使用 ACM 憑證。

下一節提供連線問題的一些範例錯誤訊息及可能的解決方案。因為不同的 SQL 用戶端工具會提供不同的錯誤訊息，所以這不是完整的清單，但很適合作為問題疑難排解時的起點。

從 Amazon EC2 外部進行連線並發生防火牆逾時問題

執行長時間查詢 (例如 COPY 命令) 時，對資料庫的用戶端連線似乎懸置或逾時。在此情況下，您可能會發現 Amazon Redshift 主控台顯示查詢已完成，但用戶端工具本身仍似乎在執行查詢。取決於連線停止的時間，查詢的結果可能遺漏或不完整。

可能的解決方案

從 Amazon EC2 執行個體以外的機器連線到 Amazon Redshift 時，便會發生此問題。在此情況下，閒置連線會在閒置一段時間後，遭到中繼網路元件 (例如防火牆) 終止。當您從虛擬私有網路 (VPN) 或本機網路登入時，這是很典型的行為。

為了避免這些逾時，建議採取下列變更：

- 提高與 TCP/IP 逾時有關的用戶端系統值。請在您用來連接至叢集的電腦上進行這些變更。應該依您的用戶端和網路來調整逾時期間。如需詳細資訊，請參閱變更 TCP/IP 逾時設定。
- (選用) 在 DSN 層級上設定保持連線行為。如需詳細資訊，請參閱變更 DSN 逾時設定。

變更 TCP/IP 逾時設定

若要變更 TCP/IP 逾時設定，請根據您用於連接至叢集的作業系統來設定逾時設定。

- Linux — 如果用戶端在 Linux 上執行，請以根使用者身分執行下列命令，以變更目前工作階段的逾時設定：

```
/sbin/sysctl -w net.ipv4.tcp_keepalive_time=200 net.ipv4.tcp_keepalive_intvl=200
 net.ipv4.tcp_keepalive_probes=5
```

  若要保留設定，請使用下列值建立或修改 /etc/sysctl.conf 檔案，然後重新啟動系統。

```
net.ipv4.tcp_keepalive_time=200
net.ipv4.tcp_keepalive_intvl=200
net.ipv4.tcp_keepalive_probes=5
```

- Windows — 如果用戶端在 Windows 上執行，請在 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\ 下編輯下列登錄設定的值：
  - KeepAliveTime：30000
  - KeepAliveInterval：1000
  - TcpMaxDataRetransmissions：10

  這些設定使用 DWORD 資料類型。如果這些設定不存在於登錄路徑下，您可以建立設定並指定這些建議值。如需編輯 Windows 登錄的相關資訊，請參閱 Windows 文件。

  設定這些值之後，請重新啟動電腦以讓變更生效。

- Mac — 如果用戶端在 Mac 上執行，請執行下列命令，以變更目前工作階段的逾時設定：

```
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
```

若要保留設定，請使用下列值建立或修改 /etc/sysctl.conf 檔案：

```
net.inet.tcp.keepidle=200000
net.inet.tcp.keepintvl=200000
net.inet.tcp.keepinit=200000
net.inet.tcp.always_keepalive=1
```

重新啟動電腦，然後執行下列命令，以驗證是否已設定這些值。

```
sysctl net.inet.tcp.keepidle
sysctl net.inet.tcp.keepintvl
sysctl net.inet.tcp.keepinit
sysctl net.inet.tcp.always_keepalive
```

變更 DSN 逾時設定

您可以在 DSN 層級上設定保持連線行為 (若選擇這麼做)。在作法上是在 odbc.ini 檔案中新增或修改下列參數：

KeepAlivesCount

在認為連線已中斷之前可遺失的 TCP 保持連線封包數。

KeepAlivesIdle

驅動程式傳送 TCP 保持連線封包之前的閒置秒數。

KeepAlivesInterval

每一次 TCP 保持連線重新傳輸之間的秒數。

如果這些參數不存在，或值為 0，系統會使用指定給 TCP/IP 的保持連線參數，以決定 DSN 保持連線行為。在 Windows 上，您可以在 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\ 的登錄中找到 TCP/IP 參數。在 Linux 和 macOS 上，您可以在 sysctl.conf 檔案中找到 TCP/IP 參數。

連線遭拒或失敗

如果您的連線遭拒或失敗，您可能會收到類似下列其中一項的錯誤。

- 「無法對 *<endpoint>* 建立連線。」
- 「無法連接至伺服器：連線逾時。伺服器在主機 *'<endpoint>'* 上執行且在連接埠 *'<port>'* 上接受 TCP/IP 連線嗎？」
- 「連線遭拒。請檢查主機名稱和連接埠正確，且 postmaster 接受 TCP/IP 連線。」

可能的解決方案

一般來說，收到錯誤訊息指出無法建立連線時，表示存取叢集的許可有問題，或到達叢集的網路流量有問題。

若要從叢集所在網路外部的用戶端工具連線至叢集，請在叢集的安全群組中新增傳入規則。規則組態取決於 Amazon Redshift 叢集是否建立於虛擬私有雲端 (VPC)：

- 如果您在以 Amazon VPC 為基礎的虛擬私有雲端 (VPC) 中建立了 Amazon Redshift 叢集，請在 Amazon VPC 中於 VPC 安全群組內新增會指定用戶端 CIDR/IP 地址的傳入規則。如需為叢集設定 VPC 安全群組以及可公開存取選項的相關資訊，請參閱 VPC 中的 Redshift 資源。
- 如果在 VPC 外部建立了 Amazon Redshift 叢集，請在 Amazon Redshift 中將用戶端 CIDR/IP 地址新增至叢集安全群組。如需設定叢集安全群組的相關資訊，請參閱 Amazon Redshift 安全群組。

如果您嘗試從在 Amazon EC2 執行個體上執行的用戶端工具連線至叢集，您也會新增傳入規則。在此情況下，請在叢集安全群組中新增規則。此規則必須指定與用戶端工具的 Amazon EC2 執行個體相關聯的 Amazon EC2 安全群組。

在某些情況下，您的用戶端與伺服器之間可能有一層，例如防火牆。在這些情況下，請確定防火牆透過您為叢集設定的連接埠接受傳入連線。

用戶端和驅動程式不相容

如果您的用戶端和驅動器不相容，您可能會收到錯誤，指出「指定的 DSN 包含驅動器與應用程式之間的架構不符」。

可能的解決方案

當您嘗試連接並得到有關架構不符的錯誤時，這表示用戶端工具和驅動程式不相容。原因是它們的系統架構不相符。例如，假設您有 32 位元用戶端工具，但卻安裝 64 位元版本的驅動程式，就可能發生此錯誤。有時，64 位元用戶端工具可以使用 32 位元驅動程式，但 32 位元應用程式不能搭配 64 位元驅動程式一起使用。請確定驅動程式和用戶端工具使用相同版本的系統架構。

查詢似乎沒有回應且有時無法觸達叢集

您遇到查詢完成方面的問題，查詢似乎正在執行，但在 SQL 用戶端工具中無回應。有時，查詢無法出現在叢集，例如系統資料表或 Amazon Redshift 主控台。

可能的解決方案

由於封包捨棄，可能發生此問題。在兩個網際網路通訊協定 (IP) 主機之間的網路路徑中，最大傳輸單位 (MTU) 出現差異。MTU 大小決定網路連線上的一個乙太網路訊框中可傳輸的封包大小上限 (位元組)。在 中 AWS，某些 Amazon EC2 執行個體類型支援 1500 的 MTU （ 乙太網路 v2 訊框 ），而其他執行個體類型則支援 9001 的 MTU (TCP/IP 巨型訊框 ）。

為了避免因為 MTU 大小不同而發生問題，建議採取下列其中一個動作：

- 如果叢集使用 EC2-VPC 平台，請以傳回 `Destination Unreachable` 的傳入自訂「網際網路控制訊息通訊協定 (ICMP)」規則來設定 Amazon VPC 安全群組。此規則會藉以指示原始主機沿著網路路徑使用最低的 MTU 大小。如需此方法的詳細資訊，請參閱將安全群組設為允許 ICMP「無法觸達目標」。
- 如果叢集使用 EC2-Classic 平台，或您無法允許 ICMP 傳入規則，請停用 TCP/IP 巨型訊框，以便使用乙太網路 v2 訊框。如需此方法的詳細資訊，請參閱設定執行個體的 MTU。

將安全群組設為允許 ICMP「無法觸達目標」

在兩個主機之間的網路中，當 MTU 大小出現差異時，首先請確定網路設定並未禁止路徑 MTU 探索 (PMTUD)。PMTUD 可讓接收端主機向原始主機回應下列 ICMP 訊息：`Destination Unreachable: fragmentation needed and DF set (ICMP Type 3, Code 4)`。此訊息指示原始主機沿著網路路徑，使用最低的 MTU 大小來重新傳送請求。如果不這樣協議，可能會因為請求太大，使得接收端主機無法接受，而發生封包捨棄。如需此 ICMP 訊息的相關資訊，請前往網際網路工程任務組 (IETF) 網站上的 RFC792。

如果您沒有明確為 Amazon VPC 安全群組設定此 ICMP 傳入規則，則會封鎖 PMTUD。在 中 AWS，安全群組是虛擬防火牆，可指定執行個體傳入和傳出流量的規則。如需 Amazon Redshift 叢集安全群組的相關資訊，請參閱 Amazon Redshift 安全群組。對於使用 EC2-VPC 平台的叢集，Amazon Redshift 會使用 VPC 安全群組來允許或拒絕流量進入叢集。根據預設，安全群組會鎖定並拒絕所有傳入流量。如需如何為 EC2-Classic 或 EC2-VPC 執行個體設定傳入和傳出規則的相關資訊，請參閱《Amazon EC2 使用者指南》中的 EC2-Classic 與 VPC 中的執行個體差異。

如需如何將規則新增至 VPC 安全群組的相關資訊，請參閱VPC security groups (VPC 安全群組)。如需此規則中所需特定 PMTUD 設定的相關資訊，請參閱《Amazon EC2 使用者指南》中的路徑 MTU 探索。

設定執行個體的 MTU

在某些情況下，您的叢集可能會使用 EC2-Classic 平台，或您無法允許傳入流量的自訂 ICMP 規則。在這些情況下，我們建議您將連線到 Amazon Redshift 叢集之 EC2 執行個體的網路界面 (NIC) 上的 MTU 調整為 1500。此調整會停用 TCP/IP 巨型訊框，以確保連線始終使用相同的封包大小。不過，此選項會完全地降低執行個體的最大網路輸送量，而不只是 Amazon Redshift 的連線而已。如需詳細資訊，請參閱下列程序。

在 Microsoft Windows 作業系統上設定 MTU

如果用戶端在 Microsoft Windows 作業系統中執行，您可以使用 `netsh` 命令來檢閱並設定乙太網路卡的 MTU 值。

1. 執行下列命令來判斷目前的 MTU 值：

   ```
   netsh interface ipv4 show subinterfaces
   ```

2. 在輸出中檢閱 `MTU` 界面卡的 `Ethernet` 值。
3. 如果值不是 `1500`，請執行下列命令來設定此值：

   ```
   netsh interface ipv4 set subinterface "Ethernet" mtu=1500 store=persistent
   ```

   設定此值之後，請重新啟動電腦以讓變更生效。

在 Linux 作業系統上設定 MTU

如果用戶端在 Linux 作業系統中執行，您可以使用 `ip` 命令來檢閱並設定 MTU 值。

1. 執行下列命令來判斷目前的 MTU 值：

   ```
   $ ip link show eth0
   ```

2. 在輸出中檢閱 `mtu` 後面的值。
3. 如果值不是 `1500`，請執行下列命令來設定此值：

```
$ sudo ip link set dev eth0 mtu 1500
```

在 Mac 作業系統上設定 MTU

- 請遵循 macOS 支援網站上有關 的指示How to change the MTU for troubleshooting purposes。如需詳細資訊，請搜尋支援網站。

設定 JDBC 擷取大小參數

根據預設，Redshift JDBC 驅動程式會使用環形緩衝區來有效管理記憶體，並防止out-of-memory錯誤。擷取大小參數僅適用於明確停用環緩衝區的情況。如需詳細資訊，請檢閱連結。在此組態中，您應該設定擷取大小，以控制每個批次中要擷取的資料列數目。

何時使用擷取大小？

在下列情況下使用擷取大小參數：

- 您需要精細控制以資料列為基礎的批次
- 使用需要傳統擷取大小行為的舊版應用程式

設定擷取大小

停用環形緩衝區時，JDBC 驅動程式預設會一次收集查詢的所有結果。傳回大型結果集的查詢可能會耗用過多的記憶體。若要以批次而非一次全部擷取結果集，請在應用程式中設定 JDBC 擷取大小參數。

> **Note**
>
> ODBC 不支援擷取大小。

為求最佳效能，請將擷取大小設定為不會導致記憶體不足錯誤的最高值。較低的擷取大小值會造成更多伺服器來回行程，進而延長執行時間。伺服器會預留資源，包括 WLM 查詢位置和關聯的記憶體，直到用戶端擷取整個結果集或查詢取消為止。適當地調校擷取大小時，那些資源會更快速釋出，使得它們可供其他查詢使用。

> **Note**
>
> 如果您必須擷取大型資料集，建議使用 UNLOAD 陳述式來將資料傳輸至 Amazon S3。使用 UNLOAD 時，運算節點會平行運作，以加速資料的傳輸。

如需設定 JDBC 擷取大小參數的相關資訊，請前往 PostgreSQL 文件中的根據游標取得結果。

使用 Amazon Redshift 資料 API

Amazon Redshift 資料 API 可簡化存取 Amazon Redshift 資料倉儲的過程，讓您不需管理資料庫驅動器、連線、網路組態、資料緩衝、憑證等。您可以使用資料 API 操作搭配 AWS SDK 來執行 SQL 陳述式。如需資料 API 操作的詳細資訊，請參閱 Amazon Redshift 資料 API 參考。

資料 API 不需要與資料庫持續連線。而是提供安全的 HTTP 端點並與 AWS SDKs整合。您可以使用端點來執行 SQL 陳述式，而不需管理連線。對資料 API 的呼叫是非同步的。資料 API 可以使用存放在中的登入資料 AWS Secrets Manager 或臨時資料庫登入資料。不管是哪一種授權方法，都不需要在 API 呼叫中傳遞密碼。如需詳細資訊 AWS Secrets Manager，請參閱什麼是 AWS Secrets Manager？AWS Secrets Manager 《使用者指南》中的 。您也可以使用 AWS IAM Identity Center 進行授權。

使用資料 API，您可以使用 Web 服務型應用程式以程式設計方式存取 Amazon Redshift 資料 AWS Lambda，包括 Amazon SageMaker AI 筆記本和 AWS Cloud9。如需這些應用程式的詳細資訊，請參閱 AWS Lambda、Amazon SageMaker AI 和 AWS Cloud9。

若要進一步了解資料 API，請參閱 AWS 巨量資料部落格中的開始使用 Amazon Redshift 資料 API。

使用 Amazon Redshift 資料 API

在使用 Amazon Redshift 資料 API 之前，請先檢閱下列步驟：

1. 確定身為資料 API 呼叫者的您是否已獲得授權。如需授權的相關資訊，請參閱 授權 Amazon Redshift 資料 API 的存取。
2. 確定您打算使用來自 Secrets Manager 的身分驗證憑證、暫時憑證或是使用 AWS IAM Identity Center來呼叫資料 API。如需詳細資訊，請參閱在呼叫 Amazon Redshift 資料 API 時選擇資料庫身分驗證憑證。
3. 如果您使用 Secrets Manager 來取得驗證憑證，請設定機密。如需詳細資訊，請參閱在 中存放資料庫登入資料 AWS Secrets Manager。

4. 在呼叫資料 API 時，檢閱考量和限制。如需詳細資訊，請參閱呼叫 Amazon Redshift 資料 API 時的考量。
5. 從 AWS Command Line Interface (AWS CLI)、從您自己的程式碼呼叫資料 API，或使用 Amazon Redshift 主控台中的查詢編輯器。如需從 AWS CLI進行呼叫的範例，請參閱呼叫資料 API。

呼叫 Amazon Redshift 資料 API 時的考量

在呼叫資料 API 時，請考量下列事項：

- Amazon Redshift 資料 API 可以存取 Amazon Redshift 所佈建的叢集和 Redshift Serverless 工作群組中的資料庫。如需 Redshift 資料 API 可用 AWS 區域 位置的清單，請參閱 中針對 Redshift 資料 API 列出的端點Amazon Web Services 一般參考。
- 查詢的持續時間上限為 24 小時。
- 每個 Amazon Redshift 叢集的作用中查詢 (STARTED 和 SUBMITTED查詢) 數目上限為 500。
- 查詢結果大小上限為 500 MB (gzip 壓縮後)。如果呼叫傳回超過 500 MB 的回應資料，則呼叫會結束。
- 查詢結果的保留時間上限為 24 小時。
- 查詢陳述式大小上限為 200 KB。
- 資料 API 可用於查詢下列節點類型的所有無伺服器工作群組，以及佈建的單一節點和多節點叢集：
  - dc2.large
  - dc2.8xlarge
  - rg.large
  - rg.xlarge
  - rg.4xlarge
  - rg.12xlarge
  - ra3.large
  - ra3.xlplus
  - ra3.4xlarge
  - ra3.16xlarge
- 叢集必須位於以 Amazon VPC 服務為基礎的虛擬私有雲端 (VPC) 中。
- 根據預設，若使用者的 IAM 角色與 ExecuteStatement 或 BatchExecuteStatement API 操作的執行者相同，則使用者可以使用 CancelStatement、DescribeStatement、GetStatementResult、GetStatementResultV2

和 ListStatements API 操作來處理相同的陳述式。若要處理其他使用者的相同 SQL 陳述式，使用者必須能夠擔任執行了 SQL 陳述式之使用者的 IAM 角色。如需擔任角色的相關資訊，請參閱 授權 Amazon Redshift 資料 API 的存取。

- BatchExecuteStatement API 操作的 Sqls 參數中的 SQL 陳述式會依陣列的順序依序執行。在上一個陳述式完成之前，後續 SQL 陳述式不會啟動。根據預設，所有 SQL 陳述式都會執行為單一交易。如果任何 SQL 陳述式失敗，所有工作都會復原。使用 ExecutionMode 參數來控制交易行為：
  - TRANSACTION （預設） — 服務會以單一交易的形式執行所有 SQL 陳述式，並遞交或復原所有陳述式。
  - AUTO_COMMIT — 每個 SQL 陳述式都會個別遞交。一個陳述式的失敗不會影響其他陳述式。

  如需範例，請參閱 執行多個 SQL 陳述式。
- ExecuteStatement 或 BatchExecuteStatement API 操作中所使用的用戶端字符保留時間上限為 8 小時。
- 選用WaitTimeSeconds參數會在操作提交新查詢或針對進行中查詢時，讓資料 API 等待最多 30 秒。這可減少輪詢請求的數量，進而減少往返次數並降低延遲。如需詳細資訊，請參閱使用長輪詢減少 API 呼叫。
- 如果 Amazon Redshift 佈建叢集和 Redshift Serverless 工作群組是使用客戶受管金鑰進行加密，則 Redshift 會建立授權，允許 Redshift Data API 將金鑰用於其操作。如需詳細資訊，請參閱 AWS KMS 搭配 Amazon Redshift 資料 API 使用。
- 在限流請求之前，Redshift Data API 中的每個 API 都有每秒交易配額。如需配額的相關資訊，請參閱 Amazon Redshift Data API 的配額。如果請求的速率超過配額，則傳回附帶 HTTP 狀態碼：400 的 ThrottlingException。若要回應限流，請使用重試策略，如 AWS SDK 和工具參考指南中的重試行為中所述。針對某些 AWS SDK 中的限流錯誤，這個策略會自動實作。

> **Note**
>
> 在 中 AWS Step Functions，預設不會啟用重試。如果您需要在 Step Functions 狀態機器中呼叫 Redshift Data API，請在 Redshift Data API 呼叫中包含 ClientToken 等冪性參數。ClientToken 的值需要在重試之間持續存在。在 ExecuteStatement API 請求的下列範例片段中，表達式 States.ArrayGetItem(States.StringSplit($$.Execution.Id, ':'), 7) 會使用內部函數來擷取 $$.Execution.Id 的 UUID 部分，這對於狀態機器的每次執行而言都是唯一的。如需詳細資訊，請參閱 AWS Step Functions 開發人員指南中的內部函數。
>
> ```
> {
> ```

```
  "Database": "dev",
  "Sql": "select 1;",
  "ClusterIdentifier": "MyCluster",
  "ClientToken.$": "States.ArrayGetItem(States.StringSplit($$.Execution.Id,
 ':'), 7)"
}
```

在呼叫 Amazon Redshift 資料 API 時選擇資料庫身分驗證憑證

當您呼叫資料 API 時，您會對部分 API 操作使用下列其中一種身分驗證方法。每種方法都需要不同的參數組合。

AWS IAM Identity Center

資料 API 可透過 AWS IAM Identity Center中註冊的單一登入使用者存取。如需有關設定 IAM Identity Center 的步驟資訊，請參閱 使用資料 API 搭配可信身分傳播。

AWS Secrets Manager

使用此方法，提供存放在 AWS Secrets Manager 中具有 `username`和 `secret-arn`之秘密的 `password`。指定的機密包含用來連線至所指定 `database` 的憑證。當您連線至叢集時，您也會提供資料庫名稱，如果您提供叢集識別碼 (`dbClusterIdentifier`)，則其必須符合儲存在機密中的叢集識別碼。當您連線至無伺服器工作群組時，您也會提供資料庫名稱。如需詳細資訊，請參閱在中存放資料庫登入資料 AWS Secrets Manager。

使用此方法，您也可以提供指定資料所在 AWS 區域 位置`region`的值。

臨時憑證

使用此方法時，請選擇下列其中一個選項：

- 在連線至無伺服器工作群組時，請指定工作群組名稱和資料庫名稱。資料庫的使用者名稱會衍生自 IAM 身分。例如，`arn:iam::123456789012:user:foo` 的資料庫使用者名稱為 `IAM:foo`。此外，也需要用來呼叫 `redshift-serverless:GetCredentials` 操作的許可。
- 以 IAM 身分連線至叢集時，請指定叢集識別碼和資料庫名稱。資料庫的使用者名稱會衍生自 IAM 身分。例如，`arn:iam::123456789012:user:foo` 的資料庫使用者名稱為 `IAM:foo`。此外，也需要用來呼叫 `redshift:GetClusterCredentialsWithIAM` 操作的許可。
- 以資料庫使用者身分連線至叢集時，請指定叢集識別碼、資料庫名稱和資料庫使用者名稱。此外，也需要用來呼叫 `redshift:GetClusterCredentials` 操作的許可。如需有關在使用此方法進行連線時要如何加入資料庫群組的資訊，請參閱連線到叢集時加入資料庫群組。

使用此方法，您也可以提供指定資料所在 AWS 區域 位置`region`的值。

在呼叫 Amazon Redshift 資料 API 時映射 JDBC 資料類型

下表會將 Java Database Connectivity (JDBC) 資料類型對應至您在資料 API 呼叫中指定的資料類型。

JDBC 資料類型	資料 API 資料類型
`INTEGER, SMALLINT, BIGINT`	`LONG`
`FLOAT, REAL, DOUBLE`	`DOUBLE`
`DECIMAL`	`STRING`
`BOOLEAN, BIT`	`BOOLEAN`
`BLOB, BINARY, LONGVARBINARY`	`BLOB`
`VARBINARY`	`STRING`
`CLOB`	`STRING`
其他類型 (包含與日期和時間相關的類型)	`STRING`

字串值會傳遞至 Amazon Redshift 資料庫，並以隱含方式轉換為資料庫的資料類型。

> **Note**
>
> 目前，資料 API 不支援通用通用唯一識別碼 (UUID) 的陣列。

在呼叫 Amazon Redshift 資料 API 時執行含有參數的 SQL 陳述式

您可以透過對 SQL 陳述式的各個部分使用參數來呼叫資料 API 操作，以控制提交給資料庫引擎的 SQL 文字。具名參數可讓您靈活地傳遞參數，而不必以硬式編碼的方式將其寫入 SQL 文字內。其可協助您重複使用 SQL 文字，並避免 SQL 隱碼攻擊問題。

下列範例顯示 `execute-statement`或 `batch-execute-statement` AWS CLI 命令之`parameters`欄位的具名參數。

```
--parameters "[{\"name\": \"id\", \"value\": \"1\"},{\"name\": \"address\", \"value\":
 \"Seattle\"}]"
```

在使用具名參數時，請考量下列事項：

- 具名參數只能用來取代 SQL 陳述式中的值。
  - 您可以取代 INSERT 陳述式中的值，例如 `INSERT INTO mytable VALUES(:val1)`。

    具名參數可以是任何順序，並且可以在 SQL 文字中使用多次。前面範例中顯示的參數選項，1 和 `Seattle` 值會插入到資料表資料欄 `id` 和 `address`。在 SQL 文字中，您可以依下列方式指定具名參數：

    ```
    --sql "insert into mytable values (:id, :address)"
    ```
  - 您可以取代條件子句中的值，例如 `WHERE attr >= :val1`、`WHERE attr BETWEEN :val1 AND :val2` 和 `HAVING COUNT(attr) > :val`。
  - 您無法取代 SQL 陳述式中的資料欄名稱，例如 `SELECT column-name`、`ORDER BY column-name` 或 `GROUP BY column-name`。

    例如，下列 SELECT 陳述式會因為語法無效而失敗。

    ```
    --sql "SELECT :colname, FROM event" --parameters "[{\"name\": \"colname\", \"value
    \": \"eventname\"}]"
    ```

    如果使用錯誤的語法描述 (`describe-statement` 操作) 陳述式，則傳回的 `QueryString` 不會替代參數的資料欄名稱 (`"QueryString": "SELECT :colname, FROM event"`)，並且會回報錯誤 (錯誤：在 \"FROM\"\n Position: 12 或附近有語法錯誤)。
  - 您無法取代彙總函數中的資料欄名稱，例如 `COUNT(column-name)`、`AVG(column-name)` 或 `SUM(column-name)`。
  - 您無法取代 JOIN 子句中的資料欄名稱。
- 當 SQL 執行時，資料會以隱含方式轉換為資料類型。如需資料類型轉換的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的資料類型。
- 您無法將值設定為 NULL。資料 API 會將其解譯為常值字串 NULL。下列範例會將 `id` 取代為常值字串 `null`。不是 SQL NULL 值。

  ```
  --parameters "[{\"name\": \"id\", \"value\": \"null\"}]"
  ```

- 您無法設定零長度的值。資料 API SQL 陳述式會失敗。下列範例會嘗試使用零長度的值來設定 id，並導致 SQL 陳述式失敗。

```
--parameters "[{\"name\": \"id\", \"value\": \"\"}]"
```

- 您無法使用參數在 SQL 陳述式中設定資料表名稱。資料 API 會遵循 JDBC PreparedStatement 的規則。
- describe-statement 操作的輸出會傳回 SQL 陳述式的查詢參數。
- execute-statement 和 batch-execute-statement操作都支援具有參數的 SQL 陳述式。使用 時batch-execute-statement，參數會與批次中的所有 SQL 陳述式共用。每個 SQL 陳述式都可以參考提供的參數子集，但每個參數必須至少由一個 SQL 陳述式使用。

在呼叫 Amazon Redshift 資料 API 時執行含有等冪性字符的 SQL 陳述式

當您提出變動的 API 請求時，該請求一般會在操作的非同步工作流程完成之前傳回結果。即使請求已傳回結果，操作還是可能會在完成前就逾時或發生其他伺服器問題。這可能會讓您難以判斷請求是否成功，而且可能導致系統多次重試以確保操作能成功完成。但是，如果原始請求和後續的重試有成功，則操作會完成多次。這表示您可能會更新比預期數量還多的資源。

等冪性可確保 API 請求不會完成超過一次。使用等冪請求時，如果原始請求成功完成，則任何後續的重試都會成功完成，而不必執行任何進一步的動作。資料 API ExecuteStatement 和 BatchExecuteStatement 操作具有選用的 ClientToken 等冪參數。ClientToken 會在 8 小時後到期。

> **Important**
>
> 如果您從 AWS SDK 呼叫 ExecuteStatement和 BatchExecuteStatement操作，它會自動產生用戶端字符，以便在重試時使用。在這種情況下，我們不建議將 client-token 參數與 ExecuteStatement 和 BatchExecuteStatement 操作搭配使用。檢視 CloudTrail 日誌以查看 ClientToken。如需 CloudTrail 日誌範例，請參閱 Amazon Redshift 資料 API 範例。

下列execute-statement AWS CLI 命令說明冪等性的選用client-token參數。

```
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
```

下表顯示您可能會從等冪 API 請求得到的一些常見回應，並提供重試建議。

回應	建議	說明
200 (OK)	請勿重試	原始請求已成功完成。任何後續的重試都會成功傳回。
400 系列的回應碼	請勿重試	請求有下列方面的問題： • 其包含無效的參數或參數組合。 • 其使用您沒有許可的動作或資源。 • 其使用處於變更狀態過程的資源。 如果請求涉及處於變更狀態過程的資源，則重試請求有可能會成功。
500 系列的回應碼	重試	錯誤是由 AWS 伺服器端問題造成，通常是暫時性的。請使用適當的退避策略來重複請求。

如需有關 Amazon Redshift 回應碼的資訊，請參閱《Amazon Redshift API 參考》中的常見錯誤。

在呼叫 Amazon Redshift 資料 API 時執行含有 session reuse 的 SQL 陳述式

當您發出 API 請求來執行 SQL 陳述式時，SQL 執行所在的工作階段通常會在 SQL 完成時終止。為了讓工作階段在指定的秒數內保持作用中，資料 API `ExecuteStatement` 和 `BatchExecuteStatement` 操作具有選用的 `SessionKeepAliveSeconds` 參數。`SessionId` 回應欄位包含工作階段的識別，此識別可在後續 `ExecuteStatement` 和 `BatchExecuteStatement` 操作中使用。在後續呼叫中，您可以指定另一個 `SessionKeepAliveSeconds` 來變更閒置逾時時間。如果 `SessionKeepAliveSeconds` 未變更，則會保留初始閒置逾時設定。使用 session reuse 時，請考量下列事項：

- `SessionKeepAliveSeconds` 的最大值為 24 小時。
- 工作階段最長可持續 24 小時。24 小時過後，工作階段就會強制關閉，而進行中查詢也會終止。

- 每個 Amazon Redshift 叢集或 Redshift Serverless 工作群組的工作階段數目上限為 500 個。
- 您一次只能在一個工作階段中執行一個查詢。您需要等到查詢完成，才能在相同工作階段中執行下一個查詢。也就是說，您無法在提供的工作階段中平行執行查詢。
- 資料 API 無法將特定工作階段的查詢排入佇列。

若要擷取呼叫 ExecuteStatement 和 BatchExecuteStatement 操作所使用的 SessionId，請呼叫 DescribeStatement 和 ListStatements 操作。若要列出工作階段及其狀態，請使用 ListSessions操作。如需詳細資訊，請參閱列出工作階段。

下列範例示範如何使用 SessionKeepAliveSeconds 和 SessionId 參數讓工作階段保持作用中並重複使用。首先，呼叫 execute-statement AWS CLI 命令，並將選用session-keep-alive-seconds參數設定為 2。

```
aws redshift-data execute-statement
    --session-keep-alive-seconds 2
    --sql "select 1"
    --database dev
    --workgroup-name mywg
```

回應會包含工作階段識別碼。

```
{
    "WorkgroupName": "mywg",
    "CreatedAt": 1703022996.436,
    "Database": "dev",
    "DbUser": "awsuser",
    "Id": "07c5ffea-76d6-4786-b62c-4fe3ef529680",
    "SessionId": "5a254dc6-4fc2-4203-87a8-551155432ee4"
}
```

然後，使用從第一次呼叫SessionId傳回的 來呼叫 execute-statement AWS CLI 命令。選擇性地指定 session-keep-alive-seconds 參數並設定為 10，以變更閒置逾時值。

```
aws redshift-data execute-statement
    --sql "select 1"
    --session-id 5a254dc6-4fc2-4203-87a8-551155432ee4
    --session-keep-alive-seconds 10
```

擷取 SQL 陳述式的結果

您可以根據結果格式，使用不同的資料 API 操作來擷取 SQL 結果。當您呼叫 ExecuteStatement 和 BatchExecuteStatement 操作時，您可以指定 JSON 或 CSV 格式的結果。如未指定，預設值會是 JSON。若要擷取 JSON 結果，請使用 GetStatementResult 操作。若要擷取 CSV 結果，請使用 GetStatementResultV2 操作。

以 JSON 格式傳回的結果是包含每一欄相關中繼資料的記錄。每一筆紀錄都會是 JSON 格式。例如，GetStatementResult 的回應看起來如下：

```
{
   "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 格式傳回的結果是包含每一欄相關中繼資料的記錄。結果會以 1 MB 區塊為單位傳回，每個區塊可以儲存任意數量的 CSV 格式資料列。每個請求最多傳回 15 MB 的結果。如果結果大於 15 MB，則會傳回下一頁記號以繼續擷取結果。例如，GetStatementResultV2 的回應看起來如下：

```
{
    "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>
}
```

授權 Amazon Redshift 資料 API 的存取

若要存取資料 API，使用者必須獲得授權。您可以透過將受管政策 (即預先定義的 AWS Identity and Access Management (IAM) 政策) 新增至該使用者，來授權其存取資料 API。我們建議的最佳實務是，將許可政策附加到 IAM 角色，然後根據需要將其指派給使用者和群組。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。若要查看受管政策允許和拒絕的許可，請參閱 IAM 主控台 (https://console.aws.amazon.com/iam/)。

設定 IAM 許可權限

Amazon Redshift 會提供 `AmazonRedshiftDataFullAccess` 受管政策。此政策可提供 Amazon Redshift 資料 API 操作的完整存取權。此政策也允許對特定 Amazon Redshift 的存取範圍 AWS Secrets Manager，以及驗證和存取 Amazon Redshift 叢集或 Redshift Serverless 工作群組所需的 IAM API 操作。

您也可以建立自己的 IAM 政策，以允許存取特定資源。若要建立您的政策，請使用 `AmazonRedshiftDataFullAccess` 政策作為起始範本。建立政策後，將它新增給每一位需要資料 API 存取的使用者。

請考量與使用者相關聯之 IAM 政策的下列要求：

- 如果您使用 AWS Secrets Manager 進行身分驗證，請確認政策允許使用 `secretsmanager:GetSecretValue`動作來擷取以金鑰 標記的秘密`RedshiftDataFullAccess`。
- 如果您使用暫時憑證對叢集進行身分驗證，請確認政策允許對叢集中任何資料庫的資料庫使用者名稱 `redshift_data_api_user` 使用 `redshift:GetClusterCredentials` 動作。此使用者名稱必須已建立在資料庫中。
- 如果您使用暫時憑證向無伺服器工作群組進行身分驗證，請確認政策允許使用 `redshift-serverless:GetCredentials` 動作來擷取以索引鍵 `RedshiftDataFullAccess` 標記的工作群組。資料庫使用者會以 1：1 映射至 source AWS Identity and Access Management (IAM) 身分。例如，使用者 sample_user 會映射至資料庫使用者 `IAM:sample_user`，而 IAM 角色 sample_role 會映射至 `IAMR:sample_role`。如需 IAM 身分的相關資訊，請參閱《IAM 使用者指南》中的 IAM 身分 (使用者、使用者群組和角色)。
- IAM 動作 `redshift-data:GetStatementResult` 允許存取 `GetStatementResult` 和 `GetStatementResultV2` API 操作。

下列連結提供 IAM 使用者指南 AWS Identity and Access Management 中有關 的詳細資訊。

- 如需有關建立 IAM 角色的資訊，請參閱建立 IAM 角色。
- 如需有關建立 IAM 政策的資訊，請參閱建立 IAM 政策。
- 如需有關將 IAM 政策新增給使用者的資訊，請參閱新增和移除 IAM 身分許可。

在另一個帳戶擁有的叢集上執行查詢

若要在另一個帳戶擁有的叢集上執行查詢，擁有端帳戶必須提供資料 API 可在呼叫端帳戶中擔任的 IAM 角色。例如，假設帳戶 B 擁有帳戶 A 需要存取的叢集。帳戶 B 可將 AWS 受管政策連接至`AmazonRedshiftDataFullAccess`帳戶 B 的 IAM 角色。然後，帳戶 B 會使用信任政策來信任帳戶 A，如下所示：

JSON

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

```
                "AWS": [
                    "arn:aws:iam::111122223333:role/someRoleA"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

最後，帳戶 A 的 IAM 角色必須能夠擔任帳戶 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 IAM 角色，將資源限制在 中的 Redshift Serverless 工作群組和 Amazon Redshift 叢集 AWS 帳戶

您可以在身分型政策中指定資源 ARN，以控制對 AWS 帳戶中 Redshift Serverless 工作群組和 Amazon Redshift 叢集的存取。此範例示範如何建立政策，以便僅允許所指定 AWS 帳戶中的工作群組和叢集存取資料 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:*"
            ]
        }
    ]
}
```

設定 IAM 政策，以限制只有陳述式擁有者能夠存取 SQL 陳述式資訊

根據預設，Amazon Redshift 資料 API 會將呼叫 `ExecuteStatement` 和 `BatchExecuteStatement` 時使用的 IAM 角色視為 SQL 陳述式的擁有者。允許擔任該角色的任何人都可以存取 SQL 陳述式的相關資訊，包括其結果。若要限制只有特定擁有者的 IAM 角色工作階段才能存取 SQL 陳述式資訊，請新增條件 `redshift-data:statement-owner-iam-userid: "${aws:userid}"`。下列 IAM 政策會限制存取權。

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}"
                }
```

```
            }
        }
    ]
}
```

您可以使用條件 statement-owner-iam-userid 搭配 CancelStatement、DescribeStatement、GetStatementResult 和 ListStatements。如需詳細資訊，請參閱 Amazon Redshift 資料 API 定義的動作。

設定 IAM 政策以限制只有工作階段擁有者才能存取 SQL 結果

根據預設，Amazon Redshift 資料 API 會將呼叫 ExecuteStatement 和 BatchExecuteStatement 時使用的 IAM 角色，視為執行 SQL 陳述式的資料庫工作階段的擁有者。允許擔任該角色的任何人都可以將查詢提交至資料庫工作階段。若要限制只有特定擁有者的 IAM 角色工作階段才能存取，請新增條件 redshift-data:session-owner-iam-userid: "${aws:userid}"。下列 IAM 政策會限制存取權。

下列 IAM 政策僅允許工作階段擁有者取得陳述式結果。條件 session-owner-iam-userid 可用來限制只有指定的 userid 可存取資源。

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}"
                }
            }
        }
    ]
}
```

您可以使用條件 `session-owner-iam-userid` 搭配 `ExecuteStatement` 和 `BatchExecuteStatement`。如需詳細資訊，請參閱 Amazon Redshift 資料 API 定義的動作。

在 中存放資料庫登入資料 AWS Secrets Manager

當您呼叫資料 API 時，您可以透過在 AWS Secrets Manager中使用機密來傳遞叢集或無伺服器工作群組的憑證。若要這樣傳遞登入資料，請指定秘密的名稱或秘密的 Amazon Resource Name (ARN)。

若要使用 Secrets Manager 來儲存憑證，您需要 `SecretManagerReadWrite` 受管政策許可。如需最低許可的詳細資訊，請參閱AWS Secrets Manager 《 使用者指南》中的使用 Secrets Manager 建立和管理 AWS 秘密。

將憑證儲存在 Amazon Redshift 叢集的機密中

1. 使用 AWS Secrets Manager 主控台建立包含叢集登入資料的秘密：

   - 當您選擇儲存新的機密時，請選擇 Redshift 叢集的憑證。
   - 將使用者名稱 (資料庫使用者)、密碼和資料庫叢集 (叢集識別碼) 的值儲存在機密中。
   - 使用索引鍵 `RedshiftDataFullAccess` 標記機密。 AWS 受管政策`AmazonRedshiftDataFullAccess`僅允許`secretsmanager:GetSecretValue`以金鑰標記的秘密執行 動作`RedshiftDataFullAccess`。

   如需說明，請參閱《AWS Secrets Manager 使用者指南》中的建立基本秘密。
2. 使用 AWS Secrets Manager 主控台來檢視您建立之秘密的詳細資訊，或執行 `aws secretsmanager describe-secret` AWS CLI 命令。

   記下秘密的名稱和 ARN。您可以在呼叫資料 API 時使用這些資料。

將憑證儲存在無伺服器工作群組的機密中

1. 使用 AWS Secrets Manager AWS CLI 命令來存放秘密，其中包含無伺服器工作群組的登入資料：

   - 在檔案中建立您的秘密，例如名為 `mycreds.json` 的 JSON 檔案。在檔案中提供使用者名稱 (資料庫使用者) 和密碼的值。

   ```
   {
         "username": "myusername",
         "password": "mypassword"
   ```

```
}
```

- 將值儲存在機密中，並使用索引鍵 RedshiftDataFullAccess 標記機密。

```
aws secretsmanager create-secret --name MyRedshiftSecret  --tags
 Key="RedshiftDataFullAccess",Value="serverless" --secret-string file://
mycreds.json
```

以下將顯示輸出。

```
{
    "ARN":
 "arn:aws:secretsmanager:region:accountId:secret:MyRedshiftSecret-mvLHxf",
    "Name": "MyRedshiftSecret",
    "VersionId": "a1603925-e8ea-4739-9ae9-e509eEXAMPLE"
}
```

如需詳細資訊，請參閱《AWS Secrets Manager 使用者指南》中的使用 AWS CLI建立基本機密。

2. 使用 AWS Secrets Manager 主控台來檢視您建立之秘密的詳細資訊，或執行 aws secretsmanager describe-secret AWS CLI 命令。

記下秘密的名稱和 ARN。您可以在呼叫資料 API 時使用這些資料。

建立資料 API 的 Amazon VPC 端點 (AWS PrivateLink)

Amazon Virtual Private Cloud (Amazon VPC) 可讓您在虛擬私有雲端 (VPC) 中安全地啟動 AWS 資源，例如 AWS Amazon Redshift 叢集和應用程式。 在虛擬私有雲端 (VPCs) 和服務之間 AWS PrivateLink 提供私有連線。您可以使用 AWS PrivateLink建立 VPC 端點，以根據 Amazon VPC 連接到不同帳戶和 VPCs 的服務。如需詳細資訊 AWS PrivateLink，請參閱《Amazon Virtual Private Cloud 使用者指南》中的 VPC 端點服務 (AWS PrivateLink)。

您可以使用 Amazon VPC 端點呼叫資料 API。使用 Amazon VPC 端點可使 Amazon VPC 中的應用程式與資料 API 中的流量保持在 AWS 網路，而不使用公有 IP 地址。Amazon VPC 端點能協助您符合與限制公有網際網路連線相關的合規和法律需求。舉例來說，若使用 Amazon VPC 端點，就能讓在 Amazon EC2 執行個體上執行之應用程式和資料 API 的流量，只在包含兩者的 VPC 中傳送。

建立 Amazon VPC 端點之後就能開始使用，而不需要在應用程式中進行任何程式碼或組態變更。

建立資料 API 的 Amazon VPC 端點

1. 登入 AWS 管理主控台 ，並在 https：//https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 選擇 Endpoints (端點)，然後選擇 Create Endpoint (建立端點)。
3. 在 Create Endpoint (建立端點) 頁面上，針對 Service category (服務類別) 選擇 AWS services (服務)。針對服務名稱，選擇 redshift-data (`com.amazonaws.region.redshift-data`)。
4. 針對 VPC，選擇要在其中建立端點的 VPC。

   選擇包含進行資料 API 呼叫之應用程式的 VPC。
5. 針對子網路，選擇執行您應用程式之 AWS 服務所使用的每個可用區域 (AZ) 的子網路。

   若要建立 Amazon VPC 端點，請指定可存取端點的私有 IP 地址範圍。若要執行此作業，請選擇每個可用區域的子網路。這麼做會將 VPC 端點限制為每個可用區域專屬的私有 IP 地址範圍，並且也會在每個可用區域中建立 Amazon VPC 端點。
6. 針對 Enable DNS name (啟用 DNS 名稱)，選取 Enable for this endpoint (為此端點啟用)。

   私有 DNS 會將標準資料 API DNS 主機名稱 (`https://redshift-data.region.amazonaws.com`) 解析為與您 Amazon VPC 端點專用 DNS 主機名稱相關的私有 IP 地址。因此，您可以使用 AWS CLI AWS SDKs存取資料 API VPC 端點，而不需要進行任何程式碼或組態變更來更新資料 API 端點 URL。
7. 針對 Security group (安全群組)，選擇要與 Amazon VPC 端點建立關聯的安全群組。

   選擇允許存取執行您應用程式之 AWS 服務的安全群組。舉例來說，若有 Amazon EC2 執行個體在執行您的應用程式，請選擇要允許存取 Amazon EC2 執行個體的安全群組。安全群組能讓您控制 VPC 中，資源流向 Amazon VPC 端點的流量。
8. 選擇建立端點。

建立端點之後，請選擇 中的連結 AWS 管理主控台 以檢視端點詳細資訊。

端點 Details (詳細資訊) 標籤會顯示建立 Amazon VPC 端點時產生的 DNS 主機名稱。

您可以使用標準端點 (`redshift-data.region.amazonaws.com`) 或其中一個 VPC 專用端點，來在 Amazon VPC 中呼叫資料 API。標準資料 API 端點會自動路由至 Amazon VPC 端點。因為私有 DNS 主機名稱在 Amazon VPC 端點建立時已啟用，所以會發生此路由。

當您在資料 API 呼叫中使用 Amazon VPC 端點時，應用程式與資料 API 間的所有流量都會維持在包含兩者的 Amazon VPC 中。您可以使用 Amazon VPC 端點進行任何類型的資料 API 呼叫。如需呼叫資料 API 的詳細資訊，請參閱呼叫 Amazon Redshift 資料 API 時的考量。

連線到叢集時加入資料庫群組

資料庫群組是資料庫使用者的集合。您可以向群組授予資料庫權限。管理員可以設定 IAM 角色，以便在使用資料 API 執行 SQL 時將這些資料庫群組納入考量。如需資料庫群組的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的群組。

您可以設定資料 API 呼叫者的 IAM 角色，以便在資料 API 連線到叢集時，呼叫中指定的資料庫使用者會加入資料庫群組。只有在連線至已佈建的叢集時才支援此功能。連線至 Redshift Serverless 工作群組時則不支援。資料 API 呼叫者的 IAM 角色也必須允許 `redshift:JoinGroup` 動作。

請透過向 IAM 角色新增標籤來進行此設定。呼叫者 IAM 角色的管理員會新增標籤，標籤中包含資料庫群組清單的索引鍵 `RedshiftDbGroups` 和索引鍵值。此值是以冒號 (:) 分隔的資料庫群組名稱清單，總長度最多可有 256 個字元。您必須之前就已在連線的資料庫中定義好資料庫群組。如果在資料庫中找不到任何指定的群組，則會忽略該群組。例如，對於資料庫群組 `accounting` 和 `retail`，索引鍵-值為 `accounting:retail`。資料 API 會使用標籤索引鍵-值組 `{"Key":"RedshiftDbGroups","Value":"accounting:retail"}` 來判斷在呼叫資料 API 時，哪些資料庫群組會與所提供的資料庫使用者相關聯。

聯結資料庫群組

1. 登入 AWS 管理主控台 並開啟位於 https：//https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 在主控台導覽窗格中，選擇 Roles (角色)，然後選擇您要編輯的角色名稱。
3. 選擇標籤索引標籤，然後選擇管理標籤。
4. 選擇新增標籤，然後新增 RedshiftDbGroups 索引鍵和值 (此為 *`database-groups-colon-separated`* 清單)。
5. 選擇儲存變更。

   現在，當 IAM 主體 (附加了此 IAM 角色) 呼叫資料 API 時，指定的資料庫使用者便會加入 IAM 角色中指定的資料庫群組。

如需如何將標籤附加到主體 (包括 IAM 角色和 IAM 使用者) 的相關資訊，請參閱《IAM 使用者指南》中的標記 IAM 資源。

使用資料 API 搭配可信身分傳播

身為 Amazon Redshift 帳戶管理員，您可以將 Amazon Redshift 叢集或工作群組與 整合 AWS IAM Identity Center，這有助於透過單一登入來管理對 Amazon Redshift 的人力資源存取。如需詳細資訊，請參閱設定與 AWS Amazon Redshift 的 IAM Identity Center 整合。Amazon Redshift Data API 支援將 IAM Identity Center 使用者身分傳播到 Amazon Redshift 叢集或工作群組，以及其他服務 AWS Lake Formation，例如鏈結下。您可以使用資料 API 來設定和查詢，方法是遵循使用受信任身分傳播以程式設計方式存取 AWS 服務中的步驟。

當您使用來自身分增強型 IAM 角色工作階段的 IAM Identity Center 使用者身分呼叫資料 API 時，您只能存取使用相同 IAM Identity Center 使用者產生的陳述式和陳述式結果。例如，下列 AWS CLI 命令會呼叫 `execute-statement`操作，以執行具有信任身分傳播的 SQL 命令。

```
aws redshift-data execute-statement
--sql "select current_user;"
--cluster-id mycluster
--database dev
```

下列 AWS CLI 命令會呼叫 `batch-execute-statement`操作來執行兩個 SQL 命令。

```
aws redshift-data batch-execute-statement
--sqls  "select current_user;"  "select current_date;"
--cluster-id mycluster
--database dev
```

若要存取由身分增強型 IAM 角色工作階段所提交具有 `cancel-statement`、`describe-statement`、`get-statement-result` 和 `get-statement-result-v2` 的陳述式，IAM Identity Center 使用者和 IAM 角色必須符合用於執行 `execute-statement` 或 `batch-execute-statement` 的憑證。例如，下列 AWS CLI 命令會取得 SQL 陳述式的結果。

```
aws redshift-data get-statement-result
--id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

若要列出陳述式，則必須提供 `cluster-identifier` 或 `workgroup-name` 參數，以確保 IAM Identity Center 使用者只能存取指派給他們的 Amazon Redshift IAM Identity Center 應用程式。例如，下列 AWS CLI 命令會列出特定叢集的陳述式。

```
aws redshift-data list-statements
```

```
--cluster-identifier mycluster
```

您也可以調用資料 API 操作，以使用可信身分傳播來存取叢集或工作群組中的資料庫物件。這包括 `list-databases`、`list-schemas`、`list-tables` 和 `describe-table` 操作。

IAM Identity Center 使用者發出的 API 呼叫可以在 AWS CloudTrail中追蹤。CloudTrail 事件的 `onBehalfOf` 區段會顯示 IAM Identity Center 使用者 ID 和身分存放區 ARN。下列範例示範 CloudTrail 事件的程式碼片段，顯示 `onBehalfOf` 區段中包含 IAM Identity Center 使用者 ID `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111` 和身分存放區 ARN `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"
            }
```

您可以執行下列 SQL 命令來檢查 IAM Identity Center 使用者提交的查詢。在此範例中，Identity Center 中註冊的電子郵件為 `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;
```

呼叫資料 API

您可以呼叫資料 API 或 AWS CLI ，在您的叢集或無伺服器工作群組上執行 SQL 陳述式。在《Amazon Redshift 資料 API 參考》中，用來執行 SQL 陳述式的主要操作是 `ExecuteStatement` 和 `BatchExecuteStatement`。資料 API 支援 AWS SDK 支援的程式設計語言。如需這些語言的相關資訊，請參閱用來在 AWS上進行建置的工具。

若要查看呼叫資料 API 的程式碼範例，請參閱 GitHub 中的開始使用 Redshift 資料 API。此儲存庫具有使用 從 Amazon EC2 AWS Glue Data Catalog和 Amazon SageMaker 執行期 AWS Lambda 存取 Amazon Redshift 資料的範例。範例程式設計語言包括 Python、Go、Java 和 Javascript。

您可以使用 AWS CLI呼叫資料 API。

下列範例使用 AWS CLI 呼叫資料 API。若要執行範例，請編輯參數值以符合您的環境。許多範例會提供 `cluster-identifier` 來針對叢集執行。當您針對無伺服器工作群組執行時，請改為提供 `workgroup-name`。這些範例會示範一些資料 API 操作。如需詳細資訊，請參閱 AWS CLI 命令參考。

下列範例中的命令已經過分割和格式化，以方便您閱讀。並非所有參數和回應都會在所有範例中顯示。如需完整請求語法、請求參數、回應語法和回應元素的 API 定義，請參閱 Amazon Redshift 資料 API 參考。

將 SQL 陳述式傳遞至 Amazon Redshift 資料倉儲

本頁中的範例涵蓋將 SQL 陳述式傳遞至資料倉儲的不同方式

執行 SQL 陳述式

若要執行 SQL 陳述式，請使用 `aws redshift-data execute-statement` AWS CLI 命令。

或者，使用 `WaitTimeSeconds` 參數，讓陳述式在 API 傳回回應之前有額外的時間完成。當陳述式完成時，回應也包含 `Status`、 `RedshiftPid`和 `HasResultSet`。如需詳細資訊，請參閱使用長輪詢減少 API 呼叫。

下列 AWS CLI 命令會對叢集執行 SQL 陳述式，並傳回識別符以擷取結果。此範例會使用 AWS Secrets Manager 身分驗證方法。

```
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
```

以下是回應的範例。

```
{
    "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"
}
```

下列 AWS CLI 命令會對叢集執行 SQL 陳述式，並傳回識別符以擷取結果。此範例使用暫時憑證身分驗證方法。

```
aws redshift-data execute-statement
    --db-user myuser
    --cluster-identifier mycluster-test
    --database dev
    --sql "select * from stl_query limit 1"
```

以下是回應的範例。

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": 1598306924.632,
    "Database": "dev",
    "DbUser": "myuser",
    "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
}
```

下列 AWS CLI 命令會對無伺服器工作群組執行 SQL 陳述式，並傳回識別符以擷取結果。此範例使用暫時憑證身分驗證方法。

```
aws redshift-data execute-statement
    --database dev
    --workgroup-name myworkgroup
    --sql "select 1;"
```

以下是回應的範例。

```
{
 "CreatedAt": "2022-02-11T06:25:28.748000+00:00",
 "Database": "dev",
 "DbUser": "IAMR:RoleName",
 "Id": "89dd91f5-2d43-43d3-8461-f33aa093c41e",
 "WorkgroupName": "myworkgroup"
}
```

下列 AWS CLI 命令會對叢集執行 SQL 陳述式，並傳回識別符以擷取結果。此範例會使用 AWS Secrets Manager 身分驗證方法和等冪性字符。

```
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
```

以下是回應的範例。

```
{
    "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"
}
```

執行多個 SQL 陳述式

若要使用一個命令執行多個 SQL 陳述式，請使用 `aws redshift-data batch-execute-statement` AWS CLI 命令。

或者，使用 `WaitTimeSeconds` 參數，在 API 傳回回應之前，允許整個批次的額外時間完成。如需詳細資訊，請參閱使用長輪詢減少 API 呼叫。

下列 AWS CLI 命令會對叢集執行三個 SQL 陳述式，並傳回識別符來擷取結果。此範例使用暫時憑證身分驗證方法。

```
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"
```

以下是回應的範例。

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": 1598306924.632,
    "Database": "dev",
    "DbUser": "myuser",
    "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
}
```

下列 AWS CLI 命令會使用執行模式，在不同的交易中`AUTO_COMMIT`執行三個 SQL 陳述式。每個陳述式都是獨立遞交的 — 一個陳述式的失敗不會阻止剩餘的陳述式執行或影響已完成的陳述式。

```
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
```

以下是回應的範例。

```
{
    "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
```

```
}
```

使用參數執行陳述式

您可以搭配 execute-statement和 batch-execute-statement操作使用具名參數。使用時batch-execute-statement，參數會與批次中的所有 SQL 陳述式共用。每個 SQL 陳述式都可以參考提供的參數子集，但每個參數必須至少由一個 SQL 陳述式使用。

下列 AWS CLI 命令會對叢集執行 SQL 陳述式，並傳回識別符以擷取結果。此範例會使用 AWS Secrets Manager 身分驗證方法。SQL 文字有具名參數 distance。在此案例中，述詞中使用的距離是 5。在 SELECT 陳述式中，資料欄名稱的具名參數只能在述詞中使用。SQL 陳述式的具名參數值會於 parameters 選項中指定。

```
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
```

以下是回應的範例。

```
{
    "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"
}
```

下列範例會使用來自範例資料庫的 EVENT 資料表。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 EVENT 資料表。

如果資料庫中還沒有 EVENT 資料表，您可以使用資料 API 加以建立，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)"
```

下列命令會在 EVENT 資料表中插入一個資料列。

```
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\"}]"
```

下列命令會在 EVENT 資料表中插入第二個資料列。此範例示範下列操作：

- 名為 id 的參數會在 SQL 文字中使用四次。
- 在插入參數 starttime 時會自動套用隱含類型轉換。
- venueid 資料欄是轉換為 SMALLINT 資料類型的類型。
- 代表 DATE 資料類型的字元字串會隱含地轉換成 TIMESTAMP 資料類型。
- 您可以在 SQL 文字中使用註解。

```
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\"}]"
```

下列範例顯示所插入的兩個資料列：

```
 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
```

下列命令會在 WHERE 子句中使用具名參數來擷取 eventid 是 1 的資料列。

```
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\"}]"
```

執行下列命令以取得上一個 SQL 陳述式的 SQL 結果：

```
aws redshift-data get-statement-result --id 7529ad05-b905-4d71-9ec6-8b333836eb5a
```

提供下列結果：

```
{
    "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
}
```

下列 AWS CLI 命令會使用 對叢集執行兩個 SQL 陳述式，其中包含共用的具名參數batch-execute-statement。SQL 文字已命名參數 id和 name。第一個 SQL 陳述式使用 id 參數，第二個則使用 name 參數。

```
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\"}]"
```

以下是回應的範例。

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": 1598306924.632,
    "Database": "dev",
    "DbUser": "myuser",
    "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
}
```

下列 AWS CLI 命令會執行兩個 SQL 陳述式，其中兩個陳述式使用相同的參數。名為 的 參數id用於 INSERT 陳述式和 SELECT 陳述式。

```
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\"}]"
```

以下是回應的範例。

```
{
    "ClusterIdentifier": "mycluster-test",
    "CreatedAt": 1598306924.632,
    "Database": "dev",
    "DbUser": "myuser",
    "Id": "d9b6c0c9-0747-4bf4-b142-e8883122f766"
}
```

列出有關 SQL 陳述式的中繼資料

若要列出 SQL 陳述式的中繼資料，請使用 `aws redshift-data list-statements` AWS CLI 命令。執行此命令的授權取決於呼叫者的 IAM 許可。

下列 AWS CLI 命令會列出執行的 SQL 陳述式。

```
aws redshift-data list-statements
    --status ALL
```

以下是回應的範例。

```
{
    "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
        }
    ]
}
```

列出工作階段

若要列出您在過去 24 小時內建立的工作階段，請使用 `aws redshift-data list-sessions` AWS CLI 命令。根據預設，命令只會傳回狀態為 AVAILABLE或 的工作階段BUSY。您可以依工作階段狀態、運算目標（叢集或無伺服器工作群組）或資料庫篩選結果。您的 IAM 許可決定您是否可以執行此命令。如需工作階段的詳細資訊，請參閱在呼叫 Amazon Redshift 資料 API 時執行含有 session reuse 的 SQL 陳述式。

下列 AWS CLI 命令列出在叢集上執行的工作階段：

```
aws redshift-data list-sessions
   --cluster-identifier mycluster-test
```

以下是回應的範例：

```
{
```

```
    "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"
        }
    ]
}
```

下列 AWS CLI 命令會篩選CLOSED工作階段狀態，以列出無伺服器工作群組上的工作階段：

```
aws redshift-data list-sessions
    --workgroup-name myworkgroup
    --status CLOSED
```

以下是回應的範例：

```
{
    "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**
>
> 如果您使用身分增強的 IAM 角色工作階段，則必須提供 `cluster-identifier`或 `workgroup-name` 參數。使用此參數，您可以確保 AWS IAM Identity Center 使用者只能存取指派給他們的 Amazon Redshift IAM Identity Center 應用程式。如需詳細資訊，請參閱使用資料 API 搭配可信身分傳播。

描述有關 SQL 陳述式的中繼資料

若要取得 SQL 陳述式中繼資料的說明，請使用 `aws redshift-data describe-statement` AWS CLI 命令。您也可以使用此命令，藉由提供子陳述式 ID 來描述批次中的個別子陳述式。執行此命令的授權取決於呼叫者的 IAM 許可。

或者，若要減少輪詢請求的數量，請使用 `WaitTimeSeconds` 參數為在此 API 呼叫中更新操作狀態提供額外時間。如需詳細資訊，請參閱使用長輪詢減少 API 呼叫。

下列 AWS CLI 命令說明 SQL 陳述式。

```
aws redshift-data describe-statement
    --id d9b6c0c9-0747-4bf4-b142-e8883122f766
```

以下是回應的範例。

```
{
    "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
}
```

以下是在執行含有多個 SQL 陳述式的 batch-execute-statement 命令後的 describe-statement 回應範例。

```
{
    "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
}
```

下列 AWS CLI 命令透過提供子陳述式 ID 來描述批次中的特定子陳述式。

```
aws redshift-data describe-statement
    --id b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2
```

以下是回應的範例。父陳述式狀態為 ，STARTED因為批次尚未完成。SubStatements 陣列僅包含請求的子陳述式。

```
{
    "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
}
```

擷取 SQL 陳述式的結果

若要從執行的 SQL 陳述式擷取結果，請使用 `redshift-data get-statement-result`或 `redshift-data get-statement-result-v2` AWS CLI 命令。來自 `get-statement-result` 的結果為 JSON 格式。來自 `get-statement-result-v2` 的結果為 CSV 格式。您可以提供所收到的 `Id` 以回應 `execute-statement` 或 `batch-execute-statement`。由 `batch-execute-statement` 執行的 SQL 陳述式的 `Id` 值可以在 `describe-statement` 的結果中擷取，並且會加上由冒號和序號組成的字尾 (例如 `b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2`)。如果您使用 `batch-execute-statement` 執行多個 SQL 陳述式，則每個 SQL 陳述式都會有一個 `Id` 值，如 `describe-statement` 中所示。執行此命令的授權取決於呼叫者的 IAM 許可。

或者，若要減少輪詢請求的數量，請使用 `WaitTimeSeconds` 參數，讓陳述式在 API 傳回陳述式結果之前有額外的時間完成。如需詳細資訊，請參閱使用長輪詢減少 API 呼叫。

下列陳述式會傳回由 `ResultFormat` 預設為 `JSON` 的 `execute-statement` 所執行 SQL 陳述式的結果。若要擷取結果，請呼叫 `get-statement-result` 操作。

```
aws redshift-data get-statement-result
    --id d9b6c0c9-0747-4bf4-b142-e8883122f766
```

下列陳述式會傳回由 `batch-execute-statement` 執行的第二個 SQL 陳述式的結果。

```
aws redshift-data get-statement-result
    --id b2906c76-fa6e-4cdf-8c5f-4de1ff9b7652:2
```

以下是呼叫 `get-statement-result` 的回應範例，其中 SQL 結果會在回應的 `Records` 索引鍵中以 JSON 格式傳回。

```
{
    "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
}
```

下列範例顯示 `execute-statement` 執行的 SQL 陳述式，且傳回的結果為 JSON 格式。資料表 `testingtable` 有三個整數欄 (col1、col2、col3)，而且有三列的值 (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"
}
```

以下是呼叫 get-statement-result 的回應範例，其中 SQL 結果會在回應的 Records 索引鍵中以 JSON 格式傳回。

```
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
}
```

下列範例顯示 execute-statement 執行的 SQL 陳述式，且傳回的結果為 CSV 格式。資料表 testingtable 有三個整數欄 (col1、col2、col3)，而且有三列的值 (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"
}
```

以下是呼叫 get-statement-result-v2 的回應範例，其中 SQL 結果會在回應的 Records 索引鍵中以 CSV 格式傳回。資料列會以換行和新行 (\r\n) 符號分隔。Records 中傳回的第一列為欄標頭。以 CSV 格式傳回的結果會以 1 MB 為單位傳回，且每個區塊可以儲存最大 1MB 的任意列數。

```
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"
}
```

使用長輪詢減少 API 呼叫

根據預設，資料 API 操作會傳回立即回應。如果操作是針對進行中陳述式，您必須重複輪詢，直到陳述式達到結束狀態。使用長輪詢時，WaitTimeSeconds將 設為介於 1 到 30 之間的值。API 會延遲傳回同步回應，直到陳述式完成或WaitTimeSeconds到達，以先到者為準。

DescribeStatement、ExecuteStatement、BatchExecuteStatement、GetStatementResult 和 GetStatementResultV2 支援長輪詢。

當 操作使用 WaitTimeSeconds提交新陳述式或等待現有陳述式時：

- 如果陳述式在等待時間內完成，回應會根據操作包含終端機狀態或結果資料。
- 如果陳述式在等待時間過期時仍在執行，回應會傳回目前進行中的狀態。
- 如果陳述式在提出長輪詢請求之前已完成，API 會立即傳回終端機狀態。
- 如果省略 WaitTimeSeconds ，則會保留預設的非同步行為。

常用案例

以下是長時間輪詢可能有用的範例：

- 提交簡短查詢並等待完成 - ExecuteStatement搭配 使用WaitTimeSeconds。當查詢完成或等待時間過期時，回應會傳回陳述式 Status、 RedshiftPid和 HasResultSet。若要取得完整的陳述式中繼資料，請呼叫 DescribeStatement。如果 HasResultSet是 true，請呼叫 GetStatementResult來擷取結果。
- 長輪詢狀態 — 使用 ExecuteStatement提交查詢，然後使用 呼叫 DescribeStatement WaitTimeSeconds。當查詢完成時， DescribeStatement會傳回完整的陳述式中繼資料，包括狀態、持續時間和錯誤詳細資訊。
- 長輪詢結果 — 使用 ExecuteStatement提交查詢，然後使用 呼叫 GetStatementResult WaitTimeSeconds。當查詢完成時，會直接傳回結果資料。如果查詢在等待時間過期時仍在執行，ResourceNotFoundException則會傳回 ，表示尚無結果可用。
- 提交多個查詢並等待整個批次完成 — BatchExecuteStatement與 搭配使用WaitTimeSeconds。通話會保持開啟狀態，直到所有子陳述式完成，並傳回批次父系 ID 和整體狀態。

- 提交子陳述式的多個查詢和長輪詢狀態 — 使用 DescribeStatement或 GetStatementResult搭配子陳述式 ID 和 WaitTimeSeconds。一旦子陳述式完成，即會傳回 ，無需等待批次的其餘部分。

描述資料表

若要取得描述資料表的中繼資料，請使用 aws redshift-data describe-table AWS CLI 命令。

下列 AWS CLI 命令會對叢集執行 SQL 陳述式，並傳回描述資料表的中繼資料。此範例使用 AWS Secrets Manager 身分驗證方法。

```
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
```

以下是回應的範例。

```
{
    "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"
        }
    ]
}
```

下列 AWS CLI 命令會對描述資料表的叢集執行 SQL 陳述式。此範例使用暫時憑證身分驗證方法。

```
aws redshift-data describe-table
    --db-user myuser
    --cluster-identifier mycluster-test
    --database dev
    --schema information_schema
    --table sql_features
```

以下是回應的範例。

```
{
    "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"
        }
    ]
}
```

列出叢集中的資料庫

若要列出叢集中的資料庫，請使用 `aws redshift-data list-databases` AWS CLI 命令。

下列 AWS CLI 命令會對叢集執行 SQL 陳述式，以列出資料庫。此範例使用 AWS Secrets Manager 身分驗證方法。

```
aws redshift-data list-databases
```

```
    --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
    --cluster-identifier mycluster-test
    --database dev
```

以下是回應的範例。

```
{
    "Databases": [
        "dev"
    ]
}
```

下列 AWS CLI 命令會對叢集執行 SQL 陳述式，以列出資料庫。此範例使用暫時憑證身分驗證方法。

```
aws redshift-data list-databases
    --db-user myuser
    --cluster-identifier mycluster-test
    --database dev
```

以下是回應的範例。

```
{
    "Databases": [
        "dev"
    ]
}
```

列出資料庫中的結構描述

若要列出資料庫中的結構描述，請使用 `aws redshift-data list-schemas` AWS CLI 命令。

下列 AWS CLI 命令會對叢集執行 SQL 陳述式，以列出資料庫中的結構描述。此範例使用 AWS Secrets Manager 身分驗證方法。

```
aws redshift-data list-schemas
    --secret-arn arn:aws:secretsmanager:us-west-2:123456789012:secret:myuser-secret-
hKgPWn
    --cluster-identifier mycluster-test
```

```
    --database dev
```

以下是回應的範例。

```
{
    "Schemas": [
        "information_schema",
        "pg_catalog",
        "pg_internal",
        "public"
    ]
}
```

下列 AWS CLI 命令會對叢集執行 SQL 陳述式，以列出資料庫中的結構描述。此範例使用暫時憑證身分驗證方法。

```
aws redshift-data list-schemas
    --db-user mysuser
    --cluster-identifier mycluster-test
    --database dev
```

以下是回應的範例。

```
{
    "Schemas": [
        "information_schema",
        "pg_catalog",
        "pg_internal",
        "public"
    ]
}
```

列出資料庫中的資料表

若要列出資料庫中的資料表，請使用 `aws redshift-data list-tables` AWS CLI 命令。

下列 AWS CLI 命令會對叢集執行 SQL 陳述式，以列出資料庫中的資料表。此範例使用 AWS Secrets Manager 身分驗證方法。

```
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
```

以下是回應的範例。

```
{
    "Tables": [
        {
            "name": "sql_features",
            "schema": "information_schema",
            "type": "SYSTEM TABLE"
        },
        {
            "name": "sql_implementation_info",
            "schema": "information_schema",
            "type": "SYSTEM TABLE"
        }
}
```

下列 AWS CLI 命令會對叢集執行 SQL 陳述式，以列出資料庫中的資料表。此範例使用暫時憑證身分驗證方法。

```
aws redshift-data list-tables

     --db-user myuser
     --cluster-identifier mycluster-test
     --database dev
     --schema information_schema
```

以下是回應的範例。

```
{
    "Tables": [
        {
            "name": "sql_features",
```

```
            "schema": "information_schema",
            "type": "SYSTEM TABLE"
        },
        {
            "name": "sql_implementation_info",
            "schema": "information_schema",
            "type": "SYSTEM TABLE"
        }
    ]
}
```

針對 Amazon Redshift 資料 API 的問題進行故障診斷

請使用下列標題包含常見錯誤訊息的章節，協助針對您在使用資料 API 時所發生的問題進行故障診斷。

主題

- 查詢封包過大
- 資料庫回應超過大小上限

查詢封包過大

如果您看到錯誤，內容指出查詢的封包太大，這一般是因為針對資料列傳回的結果集太大。針對資料庫傳回結果集中的每個資料列，資料 API 的大小上限為每個資料列 64 KB。

若要解決此問題，請確定結果集的每個資料列都是 64 KB 或更小。

資料庫回應超過大小上限

如果您看到錯誤，內容指出資料庫回應超過大小限制，這一般是因為資料庫傳回的結果集大小太大。資料庫傳回的結果集中的資料 API 限制為 500 MB。

若要解決此問題，請確定對資料 API 的呼叫傳回 500 MB 或更少的資料。如果您需要傳回超過 500 MB，您可以使用查詢中的 `LIMIT`子句執行多個陳述式呼叫。

使用 Amazon EventBridge 來排程 Amazon Redshift 資料 API 操作

您可以建立會比對所選事件並將其路由到目標以採取行動的規則。您也可以使用規則對預定的排程採取行動。如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」。

若要使用 EventBridge 來排程資料 API 操作，相關聯的 IAM 角色必須信任 CloudWatch Events 的主體 (events.amazonaws.com)。此角色應該要有已附加之受管政策 `AmazonEventBridgeFullAccess` 的對等項目。其也應該要有由資料 API 管理的 `AmazonRedshiftDataFullAccess` 政策許可。您可以在 IAM 主控台上使用這些許可建立 IAM 角色。在 IAM 主控台上建立角色時，請選擇 CloudWatch Events AWS 的服務信任實體。在 EventBridge 目標的 `RoleArn` JSON 值中指定 IAM 角色。如需建立 IAM 角色的詳細資訊，請參閱《IAM 使用者指南》中的為 AWS 服務建立角色（主控台）。

您在 Amazon EventBridge 中所建立規則的 name 必須符合 `RedshiftDataParameters` 中的 `StatementName`。

下列範例顯示使用單一或多個 SQL 陳述式，以及使用 Amazon Redshift 叢集或 Amazon Redshift Serverless 工作群組作為資料倉儲來建立 EventBridge 規則的各種變化。

使用單一 SQL 陳述式和叢集來進行呼叫

下列範例使用 AWS CLI 建立 EventBridge 規則，用於對 Amazon Redshift 叢集執行 SQL 陳述式。

```
aws events put-rule
--name test-redshift-cluster-data
--schedule-expression "rate(1 minute)"
```

然後，會建立 EventBridge 目標以便依照規則中指定的排程來執行。

```
aws events put-targets
--cli-input-json file://data.json
```

data.json 輸入檔案如下。`Sql` JSON 索引鍵表示有單一 SQL 陳述式。`Arn` JSON 值包含叢集識別碼。`RoleArn` JSON 值包含用來執行 SQL 的 IAM 角色，如先前所述。

```
{
    "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
            }
        }
    ]
}
```

使用單一 SQL 陳述式和工作群組來進行呼叫

下列範例使用 AWS CLI 建立 EventBridge 規則，用於對 Amazon Redshift Serverless 工作群組執行 SQL 陳述式。

```
aws events put-rule
--name  test-redshift-serverless-workgroup-data
--schedule-expression "rate(1 minute)"
```

然後，會建立 EventBridge 目標以便依照規則中指定的排程來執行。

```
aws events put-targets
--cli-input-json file://data.json
```

data.json 輸入檔案如下。Sql JSON 索引鍵表示有單一 SQL 陳述式。Arn JSON 值包含工作群組名稱。RoleArn JSON 值包含用來執行 SQL 的 IAM 角色，如先前所述。

```
{
    "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
            }
        }
    ]
```

```
}
```

使用多個 SQL 陳述式和叢集進行呼叫

下列範例使用 AWS CLI 建立 EventBridge 規則，用於對 Amazon Redshift 叢集執行多個 SQL 陳述式。

```
aws events put-rule
--name  test-redshift-cluster-data
--schedule-expression "rate(1 minute)"
```

然後，會建立 EventBridge 目標以便依照規則中指定的排程來執行。

```
aws events put-targets
--cli-input-json file://data.json
```

data.json 輸入檔案如下。`Sqls` JSON 索引鍵表示有多個 SQL 陳述式。`Arn` JSON 值包含叢集識別碼。`RoleArn` JSON 值包含用來執行 SQL 的 IAM 角色，如先前所述。

```
{
    "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
            }
        }
    ]
}
```

使用多個 SQL 陳述式和工作群組進行呼叫

下列範例使用 AWS CLI 建立 EventBridge 規則，用於對 Amazon Redshift Serverless 工作群組執行多個 SQL 陳述式。

```
aws events put-rule
--name  test-redshift-serverless-workgroup-data
--schedule-expression "rate(1 minute)"
```

然後，會建立 EventBridge 目標以便依照規則中指定的排程來執行。

```
aws events put-targets
--cli-input-json file://data.json
```

data.json 輸入檔案如下。Sqls JSON 索引鍵表示有多個 SQL 陳述式。Arn JSON 值包含工作群組名稱。RoleArn JSON 值包含用來執行 SQL 的 IAM 角色，如先前所述。

```
{
    "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
            }
        }
    ]
}
```

監控資料 API

監控是維護資料 API 和其他 AWS 解決方案的可靠性、可用性和效能的重要部分。 AWS 提供下列監控工具來監看資料 API、在發生錯誤時回報，以及適時採取自動動作：

- Amazon EventBridge 可用來自動化您的 AWS 服務，並自動回應系統事件，例如應用程式可用性問題或資源變更。來自 AWS 服務的事件會以近乎即時的方式交付至 EventBridge。您可編寫簡單的規則，來指示您在意的事件，以及當事件符合規則時所要自動執行的動作。如需詳細資訊，請參閱「Amazon EventBridge 使用者指南」。

- AWS CloudTrail 會擷取由 AWS 您的帳戶發出或代表發出的 API 呼叫和相關事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。您可以識別呼叫的使用者和帳戶 AWS、進行呼叫的來源 IP 地址，以及呼叫的時間。若要進一步了解 Amazon Redshift 如何與 整合 AWS CloudTrail，請參閱使用 CloudTrail 記錄。如需 CloudTrail 的相關資訊，請參閱《AWS CloudTrail 使用者指南》。

主題

- 在 Amazon EventBridge 中監控 Amazon Redshift 資料 API 的事件

在 Amazon EventBridge 中監控 Amazon Redshift 資料 API 的事件

EventBridge 會從您自己的應用程式、軟體即服務 (SaaS) 應用程式和 AWS 服務提供即時資料串流，因此您可以在 EventBridge 中監控資料 API 事件。EventBridge 會將該資料路由到目標，例如 AWS Lambda 和 Amazon SNS。這些事件與 CloudWatch Events 中出現的事件相同，可提供近乎即時的系統事件串流，說明 AWS 資源的變更。事件會傳送至包含 Amazon Redshift 資料庫的帳戶。例如，如果您在另一個帳戶中擔任某個角色，事件便會傳送至該帳戶。如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 Amazon EventBridge 事件。

當 `ExecuteStatement` 或 `BatchExecuteStatement` API 操作將 `WithEvent` 選項設定為 `true` 時，系統就會傳送資料 API 事件。事件的 `state` 欄位會包含下列其中一個值：

- 已中止 — 使用者已停止執行查詢。
- 失敗 — 查詢執行失敗。
- 已完成 — 查詢已執行完成。

系統保證會提供事件。如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的來自 AWS 服務的事件。

資料 API 的已完成事件範例

下列範例顯示當 `ExecuteStatement` API 操作完成時的資料 API 事件。在此範例中，名為 `test.testtable` 的陳述式已執行完成。

```
{
    "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
    }
}
```

AWS KMS 搭配 Amazon Redshift 資料 API 使用

當您使用客戶受管金鑰加密 Amazon Redshift 叢集或 Redshift Serverless 工作群組時，Amazon Redshift Data API 會使用相同的客戶受管金鑰來存放和加密您的查詢和結果。

資料 API 預設會加密您的資料，以保護敏感資訊，例如查詢文字和查詢結果。它使用 擁有的 AWS KMS 加密金鑰 AWS 進行此保護。

當您保護敏感資料時，靜態資料的預設加密可降低營運開銷和複雜性。此方法可協助您建置符合嚴格加密合規和法規要求的安全應用程式。

在 中使用授予 AWS KMS

資料 API 需要授予才能使用客戶受管金鑰。

當您`BatchExecuteStatement`針對使用客戶受管金鑰加密的叢集呼叫 `ExecuteStatement`或時，Amazon Redshift 會透過傳送`CreateGrant`請求至 來代表您建立授予 AWS KMS。 AWS KMS 使用授予來授予資料 API 存取您帳戶中的 KMS 金鑰。

資料 API 需要授予 ，才能將客戶受管金鑰用於下列操作：

- 傳送`Encrypt`請求至 AWS KMS ，以使用客戶受管金鑰加密查詢中繼資料。
- 將`GenerateDataKey`請求傳送至 AWS KMS ，以產生由客戶受管金鑰加密的資料金鑰。
- 將`Decrypt`請求傳送至 AWS KMS 以解密加密的資料金鑰，讓它們可以加密您的資料。

您可以隨時撤銷授予的存取權，或移除客戶受管金鑰的 Amazon Redshift 存取權。如果您這麼做，資料 API 就無法再存取客戶受管金鑰加密的資料，這會影響依賴該資料的操作。例如，如果您在撤銷授予後嘗試擷取查詢結果或追蹤查詢狀態，資料 API 會傳回 `AccessDeniedException`。

客戶受管金鑰的金鑰政策

金鑰政策會控制客戶受管金鑰的存取權限。每個客戶受管金鑰都必須只有一個金鑰政策，其中包含決定誰可以使用金鑰及其使用方式的陳述式。在建立客戶受管金鑰時，可以指定金鑰政策。如需更多資訊，請參閱 AWS Key Management Service 開發人員指南中的客戶受管金鑰。

若要將客戶受管金鑰與資料 API 搭配使用，您必須先允許存取 Amazon Redshift。金鑰政策中必須允許下列 API 操作：

- `kms:CreateGrant`：新增客戶受管金鑰的授權。授予控制對指定 AWS KMS 金鑰的存取，允許存取 Amazon Redshift 所需的授予操作。如需詳細資訊，請參閱在 中使用授予 AWS KMS。

以下是金鑰政策範例：

```
"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":"*"
   }
]
```

資料 API 加密內容

加密內容是選用的一組金鑰/值對，其中包含有關資料的其他內容資訊。 AWS KMS 會使用加密內容作為額外的已驗證資料，以支援已驗證的加密。當您在加密資料的請求中包含加密內容時， 會將加密內容 AWS KMS 繫結至加密的資料。若要解密資料，您必須在請求中包含相同的加密內容。

資料 API 在所有已佈建叢集 AWS KMS 的密碼編譯操作中使用相同的三個加密內容金鑰/值對：

- `aws:redshift:arn` – 叢集的 Amazon Resource Name (ARN)
- `aws:redshift:createtime` – 您請求建立叢集時的時間戳記
- `serviceName` – `RedshiftDataAPI`

```
"EncryptionContextSubset": {
    "aws:redshift:arn": "arn:aws:redshift:us-east-1:123456789012:cluster:redshift-
cluster",
    "aws:redshift:createtime": "20250815T0000Z",
    "serviceName": "RedshiftDataAPI",
}
```

資料 API 在所有 AWS KMS 無伺服器工作群組的密碼編譯操作中使用兩個加密內容金鑰/值對：

- `aws:redshift-serverless:arn` – 命名空間的 Amazon Resource Name (ARN)
- `serviceName` – RedshiftDataAPI

```
"EncryptionContextSubset": {
    "aws:redshift-serverless:arn": "arn:aws:redshift-serverless:us-
east-1:123456789012:namespace:12345678-1234-1234-1234-123456789012",
    "serviceName": "RedshiftDataAPI"
}
```

如需加密的詳細資訊，請參閱 的密碼編譯詳細資訊簡介 AWS KMS。如需 Amazon Redshift 和 AWS KMS 整合的詳細資訊，請參閱 Amazon Redshift 如何使用 AWS KMS。

使用 Amazon Sagemaker Unified Studio 在 Amazon Redshift 和 SageMaker Lakehouse 中查詢資料庫

Amazon SageMaker Unified Studio 提供主控台外開發環境，並支援對 SageMaker Lakehouse、Amazon Redshift 和適用於 SQL 分析的 Amazon Athena 中的資料進行 SQL 分析。使用管理員的 URL 導覽至 Amazon SageMaker Unified Studio，並使用 SSO 或 AWS 登入資料登入。如需設定第一個專案的詳細資訊，請參閱《Amazon SageMaker Unified Studio 使用者指南》中的入門。

在 Amazon SageMaker Unified Studio 中，您可以使用查詢編輯器執行 Amazon Redshift 和 Amazon Athena 來進行 SQL 分析。使用查詢編輯器編寫和執行查詢、檢視結果，並與您的團隊共用您的工作。針對您 中的 Redshift 資料倉儲執行查詢 AWS 帳戶 （在相同 帳戶內和跨其他 AWS 帳戶)、使用相同界面為 Redshift 和 Athena 建置 SQL 查詢，並使用 Amazon Managed Workflows for Apache Airflow 排程 SQL 查詢。您也可以使用 Amazon Q 生成式 SQL 從自然語言生成 SQL。

Amazon Redshift 參數群組

在 Amazon Redshift 中，您可以將參數群組與您建立的每個叢集建立關聯。參數群組為參數的群組，這些參數會套用至您在叢集中建立的所有資料庫。這些參數用於設定查詢逾時和日期樣式等資料庫設定。啟動叢集時，必須將此叢集與參數群組建立關聯。如果稍後想要變更參數群組，您可以修改叢集並選擇不同的參數群組。

每個參數群組都有多個參數，用於設定資料庫的設定。可用的參數群組清單視參數群組所屬的參數群組系列而定。預設參數群組系列為 `redshift-2.0`。

Amazon Redshift 為每個參數群組系列提供一個預設的參數群組。預設參數群組的每個參數都有預設的數值，無法修改。預設參數群組名稱的格式為 `default.`*`parameter_group_family`*。例如，`redshift-2.0` 參數群組系列的預設參數群組為 `default.redshift-2.0`。

如果您想使用非預設參數群組的其他參數值，您必須建立自訂參數群組，然後將叢集與群組相關聯。自訂參數群組中的參數值一開始與預設參數群組中的相同。所有參數的初始 `source` 是 `engine-default`，因為值是由 Amazon Redshift 預設。變更參數值後，`source` 將變更為 `user`，表示數值經過修改，不同於預設值。

> **Note**
>
> Amazon Redshift 主控台不會顯示每個參數的 `source`。您必須使用 Amazon Redshift API AWS CLI、 或其中一個 AWS SDKs 來檢視 `source`。

如為您建立的參數群組，您可以隨時修改其參數值，或將所有參數值重設為預設值。也可將不同的參數群組與叢集相關聯。在某些情況下，您可能修改已與叢集相關聯的參數群組中的參數值，或將不同參數群組與叢集建立關聯。在這些情況下，您可能需要重新啟動叢集，更新的參數值才會生效。如果叢集失敗，而且 Amazon Redshift 重新啟動叢集，變更將在此時套用。如果叢集在維護期間重新啟動，則不會套用變更。如需詳細資訊，請參閱WLM 動態和靜態屬性。

預設參數值

> **Note**
>
> 截至 2025 年 1 月 10 日，`require_ssl` 參數的預設值為 true。如果您不希望叢集要求 SSL，您可以在建立叢集時使用自訂參數群組，或修改叢集，以在建立具有預設值的叢集之後將其與自訂參數群組建立關聯。

下表顯示預設參數值的概覽，包含各參數深入資訊的連結。這些是 `redshift-2.0` 參數群組系列的預設值。

參數名稱	Value	其他資訊
auto_analyze	true	《Amazon Redshift 資料庫開發人員指南》中的 auto_analyze
auto_mv	true	《Amazon Redshift 資料庫開發人員指南》中的自動具體化視觀表
datestyle	ISO、MDY	《Amazon Redshift 資料庫開發人員指南》中的 datestyle
enable_case_sensitive_identifier	false	《Amazon Redshift 資料庫開發人員指南》中的 enable_case_sensitive_identifier
enable_user_activity_logging	false	本指南中的 資料庫稽核記錄日誌
extra_float_digits	0	《Amazon Redshift 資料庫開發人員指南》中的 extra_float_digits
max_concurrency_scaling_clusters	1	《Amazon Redshift 資料庫開發人員指南》中的 max_concurrency_scaling_clusters
query_group	預設	《Amazon Redshift 資料庫開發人員指南》中的 query_group
require_ssl	true	本指南中的 設定連線的安全選項

參數名稱	Value	其他資訊
search_path	$user、public	《Amazon Redshift 資料庫開發人員指南》中的 search_path
statement_timeout	0	《Amazon Redshift 資料庫開發人員指南》中的 statement_timeout
wlm_json_configuration	[{"auto_wlm":true}]	本指南中的工作負載管理
use_fips_ssl	false	只在系統需要符合 FIPS 時啟用與 FIPS 相容的 SSL 模式。

> **Note**
>
> `max_cursor_result_set_size` 參數已棄用。如需游標結果集大小的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的游標限制條件。

您可在資料庫中使用 SET 命令暫時覆寫參數。SET 命令僅覆寫表示目前工作階段持續時間的參數。除了上表列出的參數，您也可設定資料庫中的 `wlm_query_slot_count` 來暫時調整槽計數。`wlm_query_slot_count` 參數不適用於參數群組中的組態。如需調整插槽計數的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 wlm_query_slot_count。如需暫時覆寫其他參數的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的修改伺服器組態。

工作負載管理

您可在 Amazon Redshift 中用工作負載管理 (WLM) 定義可用的查詢佇列數量，以及查詢如何路由至佇列進行處理。WLM 為參數群組組態的一部分。叢集將使用在其相關聯參數群組中指定的 WLM 組態。

建立參數群組時，預設的 WLM 組態將包含一個佇列，其最多可同時執行五個查詢。如果您想更有效地控制查詢處理功能，可新增其他佇列並設定其各自的 WLM 屬性。您所新增的每個佇列都會擁有相同的預設 WLM 組態，除非您設定其屬性。

新增額外的佇列時，組態中最後一個佇列為預設佇列。除非查詢依照 WLM 組態中的條件路由至其他佇列，否則皆由預設佇列處理。您可以為預設佇列指定模式和並行層級 (查詢槽)，但無法指定預設佇列的使用者群組或查詢群組。

如同其他參數，您也無法修改預設參數群組中的 WLM 組態。與預設參數群組相關聯的叢集永遠使用預設的 WLM 組態。若要修改 WLM 組態，請建立新的參數群組，然後將該參數群組與需要自訂 WLM 組態的任何叢集相關聯。

WLM 動態和靜態屬性

WLM 屬性分為動態和靜態兩種。您可在不重新啟動叢集的情形下，將動態屬性套用至資料庫，但靜態屬性需要重新啟動叢集才能讓變更生效。如需動態和靜態組態屬性的相關資訊，請參閱 WLM 動態和靜態組態屬性。

WLM 組態參數的屬性

您可以使用 Amazon Redshift 主控台、 AWS CLI、Amazon Redshift API 或其中一個 AWS SDKs 來設定 WLM。WLM 組態使用多項屬性定義佇列行為，例如跨佇列的記憶體配置、佇列中可同時執行的查詢數量等。

> **Note**
>
> 以下顯示屬性將同時在說明中列出其 Amazon Redshift 主控台名稱，以及對應的 JSON 屬性名稱。

下表摘要顯示屬性適用於自動 WLM 或是手動 WLM。

WLM 屬性	自動 WLM	手動 WLM
Auto WLM (自動 WLM)	是	是
啟用短期查詢加速	是	是
短期查詢最長執行時間	是	是
Priority	是	否
佇列類型	是	是
佇列名稱	是	是
並行擴展模式	是	是

WLM 屬性	自動 WLM	手動 WLM
並行	否	是
使用者群組	是	是
使用者群組萬用字元	是	是
查詢群組	是	是
查詢群組萬用字元	是	是
使用者角色	是	是
使用者角色萬用字元	是	是
Timeout (逾時)	否	已棄用
記憶體	否	是
查詢監控規則	是	是

下列清單說明您可以設定的 WLM 屬性。

Auto WLM (自動 WLM)

Auto WLM (自動 WLM) 設為 `true` 可啟用自動 WLM。自動 WLM 會將主體的並行和記憶體 (%) 的值設為 `Auto`。Amazon Redshift 可管理查詢並行和記憶體配置。預設值為 `true`。

JSON 屬性：`auto_wlm`

Enable short query acceleration (啟用短期查詢加速)

短期查詢加速 (SQA) 可排定讓短期執行的查詢優先於長期執行的查詢。SQA 會在專用佇列中執行短期查詢，所以 SQA 查詢不會被迫在佇列中排在長期查詢後面等待。SQA 可讓短期執行的查詢更快開始執行，使用者會更快看到結果。啟用 SQA 時，您也可以指定短期查詢的最長執行時間。若要啟用 SQA，請使用 `true`。預設值為 `false`。此設定適用於每個參數群組，而不是佇列。

JSON 屬性：`short_query_queue`

短期查詢最長執行時間

啟用 SQA 時，您可指定 0，讓 WLM 動態設定短期查詢的最長執行時間。或者，您也可以指定 1–20 秒 (以毫秒為單位) 的值。預設值為 0。

JSON 屬性：max_execution_time

優先順序

優先順序設定在佇列中執行之查詢的優先順序。若要設定優先順序，WLM mode (WLM 模式) 必須設定為 Auto WLM (自動 WLM)；也就是說，auto_wlm 必須是 true。優先順序值可以是 highest、high、normal、low 和 lowest。預設值為 normal。

JSON 屬性：priority

佇列類型

佇列類型指定佇列由 Auto WLM (自動 WLM) 或 Manual WLM (手動 WLM) 使用。將 queue_type 設定為 auto 或 manual。若沒有指定，則預設為 manual。

JSON 屬性：queue_type

佇列名稱

佇列的名稱。您可以根據商業需求設定佇列名稱。佇列名稱在 WLM 組態內必須是唯一的，最多可包含 64 個英數字元、底線或空格，且不能包含引號。例如，如果您有 ETL 查詢的佇列，則可以將其命名為 ETL queue。此名稱用於指標、系統資料表值和 Amazon Redshift 主控台中，以識別佇列。使用來自這些來源之名稱的查詢和報告必須能夠處理名稱的變更。先前，佇列名稱是由 Amazon Redshift 產生的。佇列的預設名稱是 Queue 1、Queue 2 到最後一個名為 Default queue 的佇列。

> **Important**
>
> 如果您變更佇列名稱，WLM 佇列指標的 QueueName 維度值 (例如 WLMQueueLength、WLMQueueWaitTime、WLMQueriesCompletedPerSecond、WLMQueryDurat 等) 也會變更。因此，如果您變更佇列的名稱，則可能需要變更您已設定的 CloudWatch 警報。

JSON 屬性：name

Concurrency Scaling mode (並行擴展模式)

若要啟用佇列的並行擴展，請將 Concurrency Scaling mode (並行擴展模式) 設定為 `auto`。當路由至佇列的查詢數量超過佇列設定的並行時，合格查詢將前往擴展叢集。當有可用的空位時，查詢會執行於主要叢集。預設值為 `off`。

JSON 屬性：`concurrency_scaling`

並行數量

手動 WLM 佇列中可同時執行的查詢數量。此屬性僅適用於手動 WLM。如果並行擴展已啟用，當佇列達到並行層級 (查詢槽) 時，合格查詢將前往擴展叢集。如果並行擴展未啟用，查詢會在佇列中等待，直到有空位為止。範圍介於 1 與 50 之間。

JSON 屬性：`query_concurrency`

User Groups (使用者群組)

逗號分隔的使用者群組名稱清單。使用者群組的成員在資料庫內執行查詢時，其查詢會路由至與其使用者群組相關聯的佇列。

JSON 屬性：`user_group`

User Group Wildcard (使用者群組萬用字元)

布林值類型，用以指示是否啟用使用者群組的萬用字元。如果為 0，表示萬用字元為停用；如果為 1，表示萬用字元為啟用。當萬用字元啟用時，您可以在執行查詢時使用 "*" 或 "?" 來指定多個使用者群組。如需詳細資訊，請參閱萬用字元。

JSON 屬性：`user_group_wild_card`

Query Groups (查詢群組)

逗號分隔的查詢群組清單。查詢群組的成員在資料庫內執行查詢時，其查詢會路由至與其查詢群組相關聯的佇列。

JSON 屬性：`query_group`

Query Group Wildcard (查詢群組萬用字元)

布林值類型，用以指示是否啟用查詢群組的萬用字元。如果為 0，表示萬用字元為停用；如果為 1，表示萬用字元為啟用。當萬用字元啟用時，您可以在執行查詢時使用 "*" 或 "?" 來指定多個查詢群組。如需詳細資訊，請參閱萬用字元。

JSON 屬性：query_group_wild_card

使用者角色

逗號分隔的使用者角色清單。具有該使用者角色的成員在資料庫內執行查詢時，其查詢會路由至與其使用者角色相關聯的佇列。如需使用者角色的相關資訊，請參閱角色型存取控制 (RBAC)。

JSON 屬性：user_role

使用者角色萬用字元

布林值類型，用以指示是否啟用查詢群組的萬用字元。如果為 0，表示萬用字元為停用；如果為 1，表示萬用字元為啟用。當萬用字元啟用時，您可以在執行查詢時使用 "*" 或 "?" 來指定多個查詢群組。如需詳細資訊，請參閱萬用字元。

JSON 屬性：user_role_wild_card

Timeout (ms) (逾時 (毫秒))

WLM 逾時 (max_execution_time) 已作廢。使用自動 WLM 時無法使用此功能。這時請改成使用 query_execution_time 建立查詢監控規則 (QMR)，以限制查詢的經歷執行時間。如需詳細資訊，請參閱 WLM 查詢監控規則。

查詢遭取消前所能執行的最長時間 (毫秒)。在某些情況下，唯讀查詢 (例如 SELECT 陳述式) 可能因為 WLM 逾時而被取消。在這些情況下，WLM 會嘗試根據 WLM 佇列指派規則，將查詢路由至下一個相符的佇列。如果查詢不符合任何其他的佇列定義，此查詢將取消；不會指派給預設佇列。如需詳細資訊，請參閱 WLM 查詢佇列跳轉。WLM 逾時不會套用到已達到 returning 狀態的查詢。若要檢視查詢的狀態，請參閱 STV_WLM_QUERY_STATE 系統資料表。

JSON 屬性：max_execution_time

Memory (%) (記憶體 (%))

配置給佇列的記憶體百分比。若您要為至少一個佇列指定記憶體百分比，您必須為所有其他佇列指定總共 100% 的記憶體。如果所有佇列合計的記憶體配置低於 100%，則未配置的記憶體由該服務管理。該服務可將未配置的記憶體，暫時提供給需要更多記憶體來處理的佇列。

JSON 屬性：memory_percent_to_use

查詢監控規則

您可以使用 WLM 查詢監控規則依照指定的條件或述詞持續監控查詢的 WLM 佇列。例如，您可監控經常使用過多系統資源的查詢，並在查詢超過指定的效能邊界時啟動指定動作。

> **Note**
>
> 如果您選擇以程式設計方式建立規則，強烈建議您使用主控台來產生 JSON，再加入參數群組定義中。

您需將查詢監控規則與特定的查詢佇列相關聯。每個佇列最多可以有 25 個規則，而全部佇列合計最多 25 規則。

JSON 屬性：`rules`

JSON 屬性階層：

```
rules
    rule_name
    predicate
        metric_name
        operator
        value
    action
        value
```

針對每個規則，您指定下列屬性：

- `rule_name` — 規則名稱在 WLM 組態內必須是唯一的。規則名稱最多為 32 個英數字元或底線，且不可含有空格和問號。
  - `predicate` — 每個規則最多可以有三個述詞。針對每個述詞，您指定下列屬性。
    - `metric_name` — 如需指標清單，請參閱《Amazon Redshift 資料庫開發人員指南》中的查詢監控指標。
    - `operator` — 運算子為 =、< 和 >。
    - `value` — 觸發動作之指定指標的閾值。
- `action` — 每個規則都與一個動作相關聯。有效的動作為：
  - `log`
  - hop (僅適用於手動 WLM)
  - `abort`
  - `change_query_priority` (僅適用於自動 WLM)

以下範例顯示名為 `rule_1`、含有兩個述詞和 hop 動作之 WLM 查詢監控規則的 JSON。

```
"rules": [
          {
            "rule_name": "rule_1",
            "predicate": [
              {
                "metric_name": "query_execution_time",
                "operator": ">",
                "value": 100000
              },
              {
                "metric_name": "query_blocks_read",
                "operator": ">",
                "value": 1000
              }
            ],
            "action": "hop"
          }
        ]
```

如需這些屬性和用於設定查詢佇列的策略的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的實作工作負載管理。

使用 設定 WLM 參數 AWS CLI

若要設定 WLM，您必須修改 `wlm_json_configuration` 參數。`wlm_json_configuration` 屬性值的大小上限為 8000 個字元。其值使用 JavaScript 物件標記法 (JSON) 格式。如果您使用 AWS CLI、Amazon Redshift API 或其中一個 AWS SDKs 來設定 WLM，請使用本節的其餘部分來了解如何建構 `wlm_json_configuration` 參數的 JSON 結構。

> **Note**
>
> 如果您使用 Amazon Redshift 主控台來設定 WLM，則不需要了解 JSON 格式，因為該主控台提供更簡單的方式來新增佇列及設定屬性。如需使用主控台來設定 WLM 的相關資訊，請參閱修改參數群組。

範例

以下範例是以自動 WLM 定義一個佇列的預設 WLM 組態。

```
{
   "auto_wlm": true
}
```

範例

以下範例為自訂的 WLM 組態，其定義一個並行層級 (查詢槽) 為 5 的手動 WLM 佇列。

```
{
   "query_concurrency":5
}
```

語法

預設的 WLM 組態非常簡單，只有一個佇列和一個屬性。您可以新增更多佇列，並為 JSON 結構的每個佇列設定多重屬性。以下語法代表用於設定有多重屬性之多重佇列的 JSON 結構：

```
[
   {
      "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",
                ...
             }
             ...

         ]"
   }
]
```

在上述範例中，開頭為 q1 的代表性屬性為第一個佇列陣列中的物件。這些物件中每個均有名稱/值對；name 和 value 共同設定了第一個佇列的 WLM 屬性。開頭為 q2 的代表性屬性為第二個佇列陣列中的物件。如果您需要更多佇列，可為其他的每個佇列新增額外的陣列，並設定各物件的屬性。

修改 WLM 組態時，即使您只需要變更佇列中的一個屬性，仍必須加入佇列的完整結構。因為整個 JSON 結構會與 wlm_json_configuration 參數值一起以字串傳入。

格式化 AWS CLI 命令

使用 AWS CLI時，wlm_json_configuration 參數需要特定格式。使用的格式需視用戶端作業系統而定。作業系統會使用不同方式來包圍 JSON 結構，以從命令列正確傳送。如需如何在 Linux、macOS 和 Windows 作業系統中建構適當命令的詳細資訊，請參閱以下各節。如需在 AWS CLI 中一般括住 JSON 資料結構差異的詳細資訊，請參閱AWS Command Line Interface 《 使用者指南》中的引用字串。

範例

下列命令範例為名為 example-parameter-group 的參數群組設定手動 WLM。該組態啟用短期查詢加速，短期查詢的最長執行時間設為 0，指示 WLM 動態設定數值。ApplyType 設定為 dynamic。此設定表示對參數中動態屬性進行的任何變更將立即套用，除非組態有任何其他靜態變更。該組態定義三個佇列，內容如下：

- 第一個佇列可讓使用者指定 report 為佇列中的標籤 (如 query_group 屬性中所指定)，以協助將查詢路由至該佇列。report* 標籤將啟用萬用字元搜尋，因此不需要精確指定要路由至佇列的查詢。例如，reports 和 reporting 皆符合此查詢群組。佇列將從所有佇列的總記憶體中分配到 25%，且最多可同時執行四個查詢。查詢限制的最長時間為 20000 毫秒 (ms)。模式設定為自動，因此當佇列的查詢空位已滿時，合格查詢將傳送到擴展叢集。
- 第二個佇列可讓屬於資料庫中 admin 或 dba 群組成員的使用者將其查詢路由至佇列，以進行處理。使用者群組不可用萬用字元搜尋，因此使用者必須與資料庫中的群組精確相符，其查詢才會路由至佇列。佇列將從所有佇列的總記憶體中分配到 40%，且最多可同時執行五個查詢。模式設為關閉，因此 admin 或 dba 群組成員傳送的所有查詢都會在主要叢集上執行。
- 組態中的最後一個佇列為預設佇列。此佇列將從所有佇列的總記憶體中分配到 35%，且一次最多可處理五個查詢。模式設為自動。

> **Note**
>
> 範例以多行顯示，做為示範之用。實際的命令不應包含分行符號。

```
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}
]'
```

以下是為自動 WLM 組態設定 WLM 查詢監控規則的範例。下列範例建立名為 example-monitoring-rules 的參數群組。此組態定義與先前範例中相同的三個佇列，但不再指定 query_concurrency 和 memory_percent_to_use。組態還新增了以下規則和查詢優先順序：

- 第一個佇列定義名為 rule_1 的規則。規則有兩個述詞：query_cpu_time > 10000000 和 query_blocks_read > 1000。規則動作為 log。此佇列的優先順序為 Normal。
- 第二個佇列定義名為 rule_2 的規則。規則有兩個述詞：query_execution_time > 600000000 和 scan_row_count > 1000000000。規則動作為 abort。此佇列的優先順序為 Highest。
- 組態中的最後一個佇列為預設佇列。此佇列的優先順序為 Low。

> **Note**
>
> 範例以多行顯示，做為示範之用。實際的命令不應包含分行符號。

```
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
} ]'
```

使用命令列 AWS CLI 中的 搭配 JSON 檔案來設定 WLM

您可以使用 AWS CLI 來修改 `wlm_json_configuration` 參數，並以 JSON 檔案的形式傳遞 `parameters` 引數的值。

```
aws redshift modify-cluster-parameter-group --parameter-group-name
 myclusterparaametergroup --parameters file://modify_pg.json
```

--parameters 的引數會儲存在 modify_pg.json 檔案中。檔案位置會以您的作業系統的格式指定。如需詳細資訊，請參閱從檔案載入參數。以下顯示 modify_pg.json JSON 檔案內容的範例。

```
[
    {
        "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 作業系統的命令列 AWS CLI 中使用 來設定 WLM 的規則

請遵循這些規則，在一行上執行具有參數的 AWS CLI 命令：

- 整個 JSON 結構必須包圍在單引號 (') 和一組括號 ([ ]) 內。
- 所有參數名稱和參數值必須包圍在雙引號 (") 內。
- ParameterValue 值內的整個巢狀結構必須包圍在雙引號 (") 和括號 ([ ]) 內。
- 巢狀結構內每個佇列的每個屬性和值必須包圍在大括號 ({ }) 內。
- 巢狀結構內的每個雙引號 (") 前必須使用反斜線 (\) 逸出字元。
- 如為名稱/值對，冒號 (:) 可分隔每個屬性與值。
- 每個名稱/值對用逗號 (,) 區隔。

- 多重佇列在佇列結束大括號 (}) 和下一個佇列起始大括號 ({) 之間用逗號 (,) 分隔。

在 Microsoft Windows 作業系統 AWS CLI 上使用 Windows PowerShell 設定 WLM 的規則

請遵循這些規則，在一行上執行具有參數的 AWS CLI 命令：

- 整個 JSON 結構必須包圍在單引號 (') 和一組括號 ([ ]) 內。
- 所有參數名稱和參數值必須包圍在雙引號 (") 內。
- `ParameterValue` 值內的整個巢狀結構必須包圍在雙引號 (") 和括號 ([ ]) 內。
- 巢狀結構內每個佇列的每個屬性和值必須包圍在大括號 ({ }) 內。
- 巢狀結構內的每個雙引號 (") 和其反斜線 (\) 逸出字元前必須使用反斜線 (\) 逸出字元。此需求意味著您將使用三個反斜線和一個雙引號，以確定用正確方式傳送屬性 (\\\")。
- 如為名稱/值對，冒號 (:) 可分隔每個屬性與值。
- 每個名稱/值對用逗號 (,) 區隔。
- 多重佇列在佇列結束大括號 (}) 和下一個佇列起始大括號 ({) 之間用逗號 (,) 分隔。

在 Windows 作業系統上使用命令提示設定 WLM 的規則

請遵循這些規則，在一行上執行具有參數的 AWS CLI 命令：

- 整個 JSON 結構必須包圍在雙引號 (") 和一組括號 ([ ]) 內。
- 所有參數名稱和參數值必須包圍在雙引號 (") 內。
- `ParameterValue` 值內的整個巢狀結構必須包圍在雙引號 (") 和括號 ([ ]) 內。
- 巢狀結構內每個佇列的每個屬性和值必須包圍在大括號 ({ }) 內。
- 巢狀結構內的每個雙引號 (") 和其反斜線 (\) 逸出字元前必須使用反斜線 (\) 逸出字元。此需求意味著您將使用三個反斜線和一個雙引號，以確定用正確方式傳送屬性 (\\\")。
- 如為名稱/值對，冒號 (:) 可分隔每個屬性與值。
- 每個名稱/值對用逗號 (,) 區隔。
- 多重佇列在佇列結束大括號 (}) 和下一個佇列起始大括號 ({) 之間用逗號 (,) 分隔。

建立參數群組

如果想要設定不同於預設參數群組的參數值，您可以建立自己的參數群組。

建立參數群組

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單中，選擇組態，然後選擇工作負載管理以顯示工作負載管理頁面。
3. 選擇 Create (建立) 以顯示 Create parameter group (建立參數群組) 視窗。
4. 針對 Parameter group name (參數群組名稱) 和 Description (描述) 輸入值。
5. 請選擇 Create (建立) 以建立參數群組。

修改參數群組

您可以檢視所有的參數群組，以查看參數值摘要和工作負載管理 (WLM) 組態。您可以修改參數來變更參數設定和 WLM 組態屬性。

> **Note**
>
> 您無法修改預設參數群組。

AWS 管理主控台

在主控台中，群組參數顯示於參數索引標籤中，而工作負載佇列則顯示於工作負載管理索引標籤中。

修改參數群組

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單中，選擇組態，然後選擇工作負載管理以顯示工作負載管理頁面。
3. 選擇您要修改的參數群組以顯示詳細資訊頁面及 Parameters (參數) 和 Workload management (工作負載管理) 標籤。
4. 選擇 Parameters (參數) 標籤以檢視目前的參數設定。
5. 選擇 Edit parameters (編輯參數) 以變更這些參數的設定：

   - `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

如需這些參數的相關資訊，請參閱 Amazon Redshift 參數群組。

6. 輸入您的變更，然後選擇 Save (儲存) 以更新參數群組。

修改參數群組的 WLM 組態

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單中，選擇組態，然後選擇工作負載管理以顯示工作負載管理頁面。
3. 選擇您要修改的參數群組以顯示詳細資訊頁面及 Parameters (參數) 和 Workload management (工作負載管理) 標籤。
4. 選擇 Workload management (工作負載管理) 標籤以檢視目前的 WLM 組態。
5. 選擇編輯工作負載佇列以編輯 WLM 組態。
6. (選用) 請選取 Enable short query acceleration (啟用短期查詢加速)，以啟用短期查詢加速 (SQA)。

   啟用 SQA 時，預設會將 Maximum run time for short queries (1 to 20 seconds) (短期查詢 (1 到 20 秒) 的最長執行時間) 設為 Dynamic (動態)。若要將最長執行時間設為固定值，請選擇 1–20 的值。
7. 執行下列其中一個動作來修改佇列組態：

- 選擇 Switch WLM mode (切換 WLM 模式)，以選擇 Auto WLM (自動 WLM) 或 Manual WLM (手動 WLM)。

  使用 Automatic WLM (自動 WLM) 時，Memory (記憶體) 和 Concurrency on main (主要叢集的並行) 值會設為 auto (自動)。
- 若要建立佇列，請選擇 Edit workload queues (編輯工作負載佇列)，再選擇 Add Queue (新增佇列)。
- 若要修改佇列，請變更資料表中的屬性值。依據佇列類型而定，屬性可能包括：
  - 可以變更 Queue name (佇列名稱)。
  - Memory (%) (記憶體 (%))
  - Concurrency on main (主要叢集的並行)
  - Concurrency scaling mode (並行擴展模式) 可以是 off (關閉) 或 auto (自動)
  - Timeout (ms) (逾時 (毫秒))
  - User groups (使用者群組)
  - Query groups (查詢群組)
  - 使用者角色

  如需這些屬性的相關資訊，請參閱WLM 組態參數的屬性。

  > **Important**
  >
  > 如果您變更佇列名稱，WLM 佇列指標的 QueueName 維度值 (例如 WLMQueueLength、WLMQueueWaitTime、WLMQueriesCompletedPerSecond、WLMQuer 等) 也會變更。因此，如果您變更佇列的名稱，則可能需要變更您已設定的 CloudWatch 警報。
- 若要變更佇列的順序，請選擇 Up (向上) 和 Down (向下) 箭頭按鈕。
- 若要刪除佇列，請在資料表中選擇佇列那一列的 Delete (刪除)。

8. (選用) 請選取 Defer dynamic changes until reboot (延遲動態變更直至重開機)，在下次重新開機之後將變更套用至叢集。

> **Note**
>
> 不論此設定如何，某些變更需要將叢集重新開機。如需詳細資訊，請參閱WLM 動態和靜態屬性。

9. 選擇 Save (儲存)。

AWS CLI

若要使用 設定 Amazon Redshift 參數 AWS CLI，請針對特定參數群組使用 modify-cluster-parameter-group 命令。在 `parameter-group-name` 中指定要修改的參數群組。您可用 `parameters` 參數 (用於 `modify-cluster-parameter-group` 命令) 指定參數群組中您想要修改之各參數的名稱/值對。

> **Note**
>
> 使用 AWS CLI設定 `wlm_json_configuration` 參數時有特殊考量。本節中的範例適用於除了 `wlm_json_configuration` 之外的所有參數。如需使用 `wlm_json_configuration`設定 的詳細資訊 AWS CLI，請參閱 工作負載管理。

修改參數值後，您必須重新啟動與修改過的參數群組相關聯的任何叢集。`applying` 的叢集狀態在套用值時會顯示 `ParameterApplyStatus`，套用結束後則顯示 `pending-reboot`。重新啟動後，叢集內的資料庫便會開始使用新的參數值。如需重新啟動叢集的相關資訊，請參閱重新啟動叢集。

> **Note**
>
> `wlm_json_configuration` 參數包含部分的動態屬性，因此不需為了套用變更而重新啟動相關聯的叢集。如需動態和靜態屬性的相關資訊，請參閱 WLM 動態和靜態屬性。

下列語法圖表顯示如何使用 `modify-cluster-parameter-group` 命令設定參數。您可指定 *`parameter_group_name`*，並將 *`parameter_name`* 和 *`parameter_value`* 換成要修改的實際參數和該參數的值。如果想同時修改一個以上的參數，請用空格分隔各個參數和值組。

```
aws redshift modify-cluster-parameter-group --parameter-group-
name parameter_group_name --parameters
 ParameterName=parameter_name,ParameterValue=parameter_value
```

下列範例說明如何設定 statement_timeout 參數群組的 enable_user_activity_logging 和 myclusterparametergroup 參數。

> **Note**
>
> 基於可讀性目的，此範例會顯示在幾行上，但實際上 AWS CLI 這是一行。

```
aws redshift modify-cluster-parameter-group
--parameter-group-name myclusterparametergroup
--parameters ParameterName=statement_timeout,ParameterValue=20000
 ParameterName=enable_user_activity_logging,ParameterValue=true
```

建立查詢監控規則

您可以使用 Amazon Redshift 主控台建立及修改 WLM 查詢監控規則。查詢監控規則是參數群組中之 WLM 組態參數的一部分。如果您修改查詢監控規則 (QMR)，則變更會自動進行，而不需要修改叢集。如需詳細資訊，請參閱 WLM 查詢監控規則。

建立規則時，您可以定義規則名稱、一或多個述詞和一個動作。

儲存含有規則的 WLM 組態時，您可以將規則定義的 JSON 程式碼視為 WLM 組態參數的一部分 JSON。

建立查詢監控規則

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽選單中，選擇組態，然後選擇工作負載管理以顯示工作負載管理頁面。
3. 選擇您要修改的參數群組以顯示詳細資訊頁面及 Parameters (參數) 和 Workload management (工作負載管理) 標籤。
4. 選擇工作負載管理索引標籤，然後選擇編輯工作負載佇列以編輯 WLM 組態。

5. 使用預先定義的範本新增規則，或從頭開始新增規則。

   若要使用預先定義的範本，執行以下作業：

   1. 選擇查詢監控規則群組中的從範本新增規則。規則範本清單即會出現。
   2. 選擇一個或多個規則範本。當您選擇 Save (儲存) 時，WLM 會為您選擇的每個範本建立一條規則。
   3. 輸入或確認該規則的值，包括 Rule names (規則名稱)、Predicates (述詞) 和 Actions (動作)。
   4. 選擇儲存。

   若要從頭開始新增規則，請執行下列作業：

   1. 若要新增其他述詞，請選擇 Add predicate (新增述詞)。每項規則最多可有三個述詞。如果符合所有述詞，WLM 會觸發相關聯的動作。
   2. 選擇 Action (動作)。每一個規則有一個動作。
   3. 選擇儲存。

Amazon Redshift 會產生 JSON 格式的 WLM 組態參數，並顯示於 JSON 區段。

刪除參數群組

您可以刪除已不需要且沒有任何相關聯叢集的參數群組。您只能刪除自訂參數群組。

刪除參數群組

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽選單中，選擇組態，然後選擇工作負載管理以顯示工作負載管理頁面。
3. 在 Parameter groups (參數群組) 中，選擇您要修改的參數群組。

   > **Note**
   >
   > 您無法刪除預設參數群組。

4. 選擇 Delete (刪除)，然後確認您要刪除該參數群組。

將 Amazon Redshift 與 AWS 合作夥伴整合

透過使用 Amazon Redshift，您可以在 Amazon Redshift 主控台上與 AWS 合作夥伴整合。從叢集詳細資訊頁面，您可以使用 AWS 合作夥伴應用程式加快資料加入 Amazon Redshift 資料倉儲的速度。您也可以將來自不同來源的資料與叢集中的現有資料聯結起來並進行分析。在完成與 Informatica 的整合之前，您必須先將合作夥伴的 IP 地址新增至傳入流量的允許清單。下列 AWS 合作夥伴可與 Amazon Redshift 整合：

- Datacoral
- Etleap
- Fivetran
- SnapLogic
- Stitch
- Upsolver
- Matillion (預覽)
- Sisense (預覽)
- Thoughtspot

AWS 合作夥伴可以使用 AWS CLI 或 Amazon Redshift API 操作與 Amazon Redshift 整合。如需詳細資訊，請參閱《AWS CLI 命令參考》和《Amazon Redshift API 參考》。

使用下列程序將叢集與 AWS 合作夥伴整合。

將 Amazon Redshift 叢集與 AWS 合作夥伴整合

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集。
3. 選擇您要整合的叢集。
4. 選擇新增合作夥伴整合。選擇合作夥伴頁面隨即開啟，其中包含可用 AWS 合作夥伴的詳細資訊。
5. 選擇 AWS 合作夥伴，然後選擇下一步。

   隨即顯示所選 AWS 合作夥伴的詳細資訊，以及您要整合之叢集的詳細資訊。叢集詳細資訊區段包含您在 AWS 合作夥伴網站上提供的資訊，例如叢集識別符、端點、資料庫名稱和使用者名稱（即資料庫使用者名稱）。此資訊會傳送給您選擇的合作夥伴。

6. 選擇新增合作夥伴以開啟 AWS 合作夥伴的網站。
7. 在合作夥伴的網站上設定與 Amazon Redshift 叢集的整合。在合作夥伴的網站上，您可以選取並設定載入到 Amazon Redshift 叢集的資料來源。您還可以定義其他的擷取、載入和轉換 (ELT) 轉換，以處理業務資料、將業務資料與其他資料集聯結，以及建置用於分析和報告的合併檢視。

您可以從叢集詳細資訊屬性索引標籤檢視和管理 AWS 合作夥伴整合。整合區段列出可用來連結至合作夥伴網站的 AWS 合作夥伴名稱、整合狀態、接收資料的資料庫，以及可能已更新叢集的上次成功連線。

可能的狀態值如下：

- 作用中 – AWS 合作夥伴可以連線至叢集並完成設定的任務。
- 非作用中 – AWS 合作夥伴整合不存在。
- 執行期失敗 – AWS 合作夥伴可以連線到叢集，但無法完成設定的任務。
- 連線失敗 – AWS 合作夥伴無法連線至叢集。

從 Amazon Redshift 刪除 AWS 合作夥伴整合後，資料會繼續流入您的叢集。請在合作夥伴的網站上完成刪除。

使用 AWS 合作夥伴載入資料

除了將合作夥伴與 Amazon Redshift 叢集整合之外，您還可以使用我們合作夥伴的資料載入工具，將資料從 30 多個來源移至 Amazon Redshift 叢集。在執行此操作之前，您必須將合作夥伴的 IP 地址 (如下所示) 新增至傳入規則的允許清單。如需將規則新增至 Amazon EC2 安全群組的詳細資訊，請參閱《Amazon EC2 使用者指南》中的授權執行個體的傳入流量。請注意，雖然 Information Data Loader 工具是免費的，但視您選擇的資料來源和目標而定，可能會收取資料傳入費用。

您可以從下列合作夥伴載入資料：

- 資訊學

透過合作夥伴將資料載入 Amazon Redshift 叢集

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表中，選擇 AWS 合作夥伴整合，然後選擇您要與叢集整合的合作夥伴。

3. 選擇完成 <partner-name> 整合。系統會將您重新導向至合作夥伴的整合網站。
4. 在合作夥伴的網站上輸入必要的詳細資訊，然後完成整合。

預留節點

在 中 AWS，您使用 Amazon Redshift 產生的費用是以運算節點為基礎。每個運算節點會以小時的費率來計費。小時費率會根據區域、節點類型以及節點是否接收隨需節點定價或預留節點定價之類的因素而定。

隨需節點定價是 Amazon Redshift 中最昂貴，但最具彈性的選項。透過隨需費率，只會對在執行叢集中擁有的運算節點向您收費。如果您將叢集關機或將其刪除，將不會針對曾在該叢集中的運算節點向您收費。只會對您使用的運算節點向您收費，僅此而已。針對每個運算節點而計費的小時費率會根據區域與節點類型之類的因素而有所不同。

預留節點定價較隨需定價便宜，因為運算節點是以打折的小時費率來計費。然而，若要接收這些打折的費率，您必須購買預留節點方案。當您購買方案時，即進行保留。保留會為您在保留期間預留的每個節點設定折扣費率。方案中的折扣費率會根據區域、節點類型、期間和付費選項之類的因素而有所不同。

您可以呼叫 `PurchaseReservedNodeOffering` API 作業或在 Amazon Redshift 主控台上選擇 Purchase reserved nodes (購買預留節點)，將節點指定為預留節點。購買預留節點時，您必須為適用的預留節點類型指定 AWS 區域、節點類型、期限、節點數量和方案類型。預留節點只能在指定區域中使用 AWS 。

此主題會討論預留節點方案為何與您可以購買的方式，以降低 Amazon Redshift 叢集的執行成本。此主題會討論費率以隨需或折扣形式的一般概念，讓您能夠了解定價的概念與定價影響計費的方式。如需特定費率的相關資訊，請前往 Amazon Redshift 定價。

預留節點優惠

如果想要讓 Amazon Redshift 叢集持續執行很長一段期間，您應考慮購買預留節點方案。這些方案可省下的金額大幅超過隨需定價，但您需要預留運算節點並承諾為那些節點支付一年或三年期的費用。

預留節點是一種計費概念，將嚴格用來判斷會使用何種費率來向您收取節點費用。預留節點實際上並不會為您建立任何節點。不論使用情形為何，都會向您收取預留節點的費用，也就是說，您必須為在保留期間預留的每個節點支付費用，不論您在執行叢集中是否有任何節點，而這些叢集是否適用折扣費率。

在專案的評估階段時，或是您正在構思概念驗證，隨需定價可以提供彈性，讓您只需為使用的量而隨需付費，並透過關閉或刪除叢集，即可隨時停止付費。在您確認生產環境的需求並開始實作階段，您應透過購買一或多個方案來考量預留運算節點。

任何方案都可適用於一或多個運算節點。您會在購買方案時指定要預留的運算節點數。您可以為多個運算節點選擇購買一個方案，或者可以選擇購買多個方案，並在每個方案中指定某個數量的運算節點。

例如，下列是為三個運算節點購買方案的方法，任一種皆為有效：

- 購買一個方案並指定三個運算節點。
- 購買兩個方案，並將一個運算節點指定給第一個方案，將兩個運算節點指定給第二個方案。
- 購買三個方案，並為每個方案指定一個運算節點。

比較預留節點方案間的定價

Amazon Redshift 提供方案的數種付費選項。您選擇的付費選項會影響付費排程以及針對保留向您收費所使用的折扣費率。您為保留所支付的預付費用越多，整體所省下的費用越多。

下列付款選項可用於方案。方案是按照隨需費率時所省下費用由少至多來排列。

> **Note**
>
> 無論您是否使用預留節點，都會針對在保留指定期間中的每小時以適用的小時費率向您收費。付費選項僅決定所適用的付費頻率與折扣。如需詳細資訊，請參閱預留節點優惠。

付款選項	付款排程	節省費用比較	持續時間	預付費用	經常性每月費用
不預付	保留期間的每月分期付款。無預付款項。	約隨需費率的 20% 折扣。	一年或三年	無	是
部分預付	保留期間的部分預付款和每月分期付款。	高達 41% 到 73% 折扣 (視期間而定)。	一年或三年	是	是
全額預付	保留的完整預付款。無每月費用.	高達 42% 到 76% 折扣 (視期間而定)。	一年或三年	是	無

特定選項和持續時間不一定都可供使用。

> **Note**
>
> 如果您之前購買過 Amazon Redshift 的重度使用方案，則類似的方案為部分預付方案。

預留節點運作的方式

透過預留節點方案，您會根據先前章節中所述的付費期限來付款。不論您是否已有執行中的叢集，或在擁有保留後啟動叢集，付費方式都是如此。

購買方案時，保留的狀態為 payment-pending (待付款)，直到保留進行處理為止。如果無法處理保留，狀態會顯示為 payment-failed (付款失敗) 且您可以再次嘗試該程序。在保留順利處理後，其狀態即會變成 active (作用中)。保留中的適用折扣費率在狀態變成 active (作用中) 前都不會加以套用。在保留期間過後，狀態會變更為 retired (淘汰)，但您可以基於查證目的繼續存取保留的相關資訊。在保留為 retired (淘汰) 後，叢集會繼續執行，且系統可能會以隨需費率向您收費，除非您有另一個將打折定價套用至節點的保留。

預留節點專屬於您購買方案的區域。如果您使用 Amazon Redshift 主控台購買方案，請選取您要購買方案 AWS 的區域，然後完成保留程序。如果您透過程式設計方式購買方案，會由您連接的 Amazon Redshift 端點來決定區域。如需 Amazon Redshift 區域的相關資訊，請前往《Amazon Web Services 一般參考》中的區域與端點。

為了確保折扣費率會在您啟動叢集時套用至所有節點，請確保您選取的區域、節點類型和節點數符合一或多個作用中的保留。否則，將針對不符合作用中保留的節點以隨需費率向您收費。

如果您在執行中叢集的節點數超出了您所預留的數量，將會以隨需費率向您收取那些其他節點的累積費用。此累積費用表示相同叢集內節點的計費費率可能會因您預留的節點數量而有所不同。如果您購買另一個方案來涵蓋那些其他節點，則在保留狀態變為 active (可用) 時即會為剩餘期間向那些節點套用折扣費率。

如果您將叢集的大小重新調整為另一個節點類型，且您沒有預留該類型的節點，將會以隨需費率向您收取費用。如果您想要為重新調整大小的叢集接收折扣費率，您可以購買另一個方案搭配新的節點類型。然而，您也繼續支付原本的保留直到其期間結束。如果您在期限過期前需要更改保留，可使用 AWS 主控台建立支援案例。

> **Note**
>
> 主控台會顯示已使用和未使用的預留節點計數。不過，主控台只會將目前使用者帳戶使用的節點數量顯示為已使用。如果同一付款人帳戶下有另一個使用者帳戶使用節點，主控台會將這些節點顯示為未使用。
>
> 範例
>
> - 付款人帳戶保留 20 個節點
> - 目前使用者帳戶使用六個節點

- 同一付款人帳戶下的另一個使用者帳戶也使用六個節點

在此範例中，主控台只會顯示六個已使用節點，以及十四個未使用節點。

預留節點和合併帳單

若購買帳戶是在單一合併帳單付款人帳戶下一組帳戶的一部分，則會共用預留節點的計費利益。所有子帳戶的小時用量每個月都會彙總到付款人的帳戶中。這通常對有不同功能團隊或群組的公司來說非常有用；然後，會套用標準預留節點邏輯計算帳單。如需詳細資訊，請參閱 AWS Billing 《使用者指南》中的合併帳單。

在使用預留執行個體的叢集上變更節點類型之前需要考量的事項

您可以使用調整大小功能（彈性或傳統）變更節點系列或節點大小，以變更叢集的節點類型。當您變更節點類型時，與叢集相關聯的預留節點購買不會自動變更。現有的保留會保持與原始節點類型繫結。您需要繼續支付這些保留的費用，直到其期限結束為止。同時，新類型的節點可能會以隨需費率計費。

因此，對於預留執行個體的變更，建議您先聯絡您的 AWS 客戶團隊或AWS 帳單支援，再變更使用預留節點的叢集節點類型。他們可以適當地引導您。

預留節點範例

此章節的使用案例會示範節點如何使用下列保留詳細資訊來根據隨需和折扣費率來累積費用：

- 區域：美國西部 (奧勒岡)
- 節點類型：ra3.xlplus
- 付款選項：無預付款
- 期間：一年期
- 預留節點數目：16

範例 1

您在美國西部 (奧勒岡) 區域擁有一個叢集，內含 20 個節點。

在此使用案例中，16 個節點會接收保留中的折扣費率，而叢集中其他 4 個節點會以隨需費率向您收費。

範例 2

您在美國西部 (奧勒岡) 區域擁有一個叢集，內含 12 個節點。

在此使用案例中，叢集中所有 12 個節點會從保留中接收折扣費率。然而，即使您目前沒有保留中剩餘預留節點適用的執行中叢集，您也仍需為它們支付費用。

範例 3

您在美國西部 (奧勒岡) 區域擁有一個叢集，內含 12 個節點。您使用此組態執行叢集數月的時間，接著您需要將節點新增至叢集。您會重新調整叢集大小，選擇相同的節點類型並指定節點總數 16。

在此使用案例中，會針對 16 個節點以折扣費率向您收費。因為您在叢集中擁有的節點數與您預留的節點數相等，因此在全年期中對您的收費向保持相同。

範例 4

您在美國西部 (奧勒岡) 區域擁有一個叢集，內含 16 個節點。您使用此組態執行叢集數月的時間，接著您需要新增節點。您會重新調整叢集大小，選擇相同的節點類型並指定節點總數 20。

在此使用案例中，在重新調整大小之前，會針對所有節點以折扣費率向您收費。在重新調整大小後，會在一年中剩下的時間針對 16 個節點以折扣費率向您收費，而針對您在叢集中新增的其他 4 個節點則會以隨需費率向您收費。

範例 5

您在美國西部 (奧勒岡) 區域擁有兩個叢集。其中一個叢集有 6 個節點，另一個叢集有 10 個節點。

在此使用案例中，因為兩個叢集中的節點總數與您預留的節點數相同，因此會以折扣費率針對所有節點向您收費。

範例 6

您在美國西部 (奧勒岡) 區域擁有兩個叢集。其中一個叢集有 4 個節點，另一個叢集有 6 個節點。

在此使用案例中，會針對您在執行中叢集中擁有的 10 個節點以折扣費率向您收費，而您也需為您預留的其他 6 個節點支付折扣費率 (即使您目前沒有它們所適用的執行中叢集)。

購買預留節點

您可以使用 AWS 管理主控台 或 AWS CLI 來購買預留節點方案，以及檢視目前和過去的保留。

AWS 管理主控台

購買預留節點

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ ：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表中，選擇叢集，然後選擇預留節點以顯示預留節點清單。
3. 選擇 Purchase reserved nodes (購買預留節點) 以顯示頁面，供您選擇要購買的節點屬性。
4. 輸入節點屬性，然後選擇 Purchase reserved nodes (購買預留節點)。

購買後，Reserved Node (預留節點) 清單會顯示您的預留節點以及各節點的詳細資訊，像是節點類型、節點數量、預留狀態。如需預留節點的相關資訊，請參閱預留節點運作的方式。

若要升級預留節點，請使用 AWS CLI。

您無法將所有節點類型轉換為預留節點，而且現有的預留節點也可能無法續約。這可能是因為節點類型已停用。請聯絡客戶支援以續約已停用的節點類型。

AWS CLI

使用 升級預留節點保留 AWS CLI

1. 取得方案的 ReservedNodeOfferingID 的清單，列出符合您的需要 (付款類型、期限、費用) 的方案。以下範例示範這個步驟。

```
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
                }
            ],
```

```
            "CurrencyCode": "USD",
            "OfferingType": "No Upfront",
            "ReservedNodeOfferingType": "Regular",
            "FixedPrice": 0.0
        }
    ]
}
```

2. 呼叫 accept-reserved-node-exchange，並提供要與上個步驟取得之 ReservedNodeOfferingID 一起交換的 DC1 預留節點 ID。

   以下範例示範這個步驟。

```
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
    }
}
```

您可以呼叫 describe-reserved-nodes 並檢查 Node type 的值，來確認交換完成。

Amazon Redshift 中的安全

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，該架構專為符合最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。共同責任模型 將此描述為雲端的安全和雲端內的安全：

- 雲端的安全性 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。 AWS 也為您提供可安全使用的服務。第三方稽核人員定期檢測及驗證安全的效率也是我們 AWS 合規計劃的一部分。若要了解適用於 Amazon Redshift 的合規計畫，請參閱合規計畫範圍內的AWS 服務。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的敏感度、您組織的需求和適用的法律及法規。

對 Amazon Redshift 資源的存取會以四個層級控制：

- 叢集管理 – 建立、設定和刪除叢集的功能是由授予與您 AWS 安全登入資料相關聯之使用者或帳戶的許可所控制。具有適當許可的使用者可以使用 AWS 管理主控台、 AWS Command Line Interface (CLI) 或 Amazon Redshift 應用程式介面 (API) 來管理其叢集。此存取透過使用 IAM 政策來管理。

  > **Important**
  >
  > Amazon Redshift 集合了用於管理許可、身分和安全存取的最佳實務。建議您在開始使用 Amazon Redshift 時熟悉這些內容。如需詳細資訊，請參閱Amazon Redshift 中的身分和存取管理。

- 叢集連線 – Amazon Redshift 安全群組會以無類別網域間路由 (CIDR) 格式指定授權連線至 Amazon Redshift 叢集的 AWS 執行個體。如需建立 Amazon Redshift、Amazon EC2 和 Amazon VPC 安全群組以及將它們與叢集建立關聯的詳細資訊，請參閱 Amazon Redshift 安全群組。
- 資料庫存取 — 存取資料庫物件 (例如資料表和檢視) 的功能是透過 Amazon Redshift 資料庫中的資料庫使用者帳戶控制。使用者只能存取其使用者帳戶已獲存取許可之資料庫中的資源。您可以透過使用 CREATE USER、CREATE GROUP、GRANT 和 REVOKE SQL 陳述式來建立這些 Amazon Redshift 使用者帳戶並管理許可。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的管理資料庫安全。
- 暫時資料庫登入資料和單一登入 — 除了使用 SQL 命令 (例如 CREATE USER 和 ALTER USER) 來建立和管理資料庫使用者，您可以使用自訂 Amazon Redshift JDBC 或 ODBC 驅動程式 (可在資料

庫登入程序中管理資料庫使用者的建立程序和暫時密碼)，來設定 SQL 用戶端。這些驅動程式會管理隨著資料庫登入的一部分建立使用者資料庫和暫時密碼的程序。

該驅動程式會根據 AWS Identity and Access Management (IAM) 身分驗證來驗證資料庫使用者。如果您已在 外部管理使用者身分 AWS，您可以使用 SAML 2.0 相容身分提供者 (IdP) 來管理對 Amazon Redshift 資源的存取。您可以使用 IAM 角色來設定 IdP AWS ，並允許聯合身分使用者產生臨時資料庫登入資料並登入 Amazon Redshift 資料庫。如需詳細資訊，請參閱使用 IAM 身分驗證產生資料庫使用者憑證。

本文件有助於您了解如何在使用 Amazon Redshift 時套用共同責任模型。下列主題說明如何將 Amazon Redshift 設定為符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 Amazon Redshift 資源。

主題

- Amazon Redshift 中的資料保護
- Amazon Redshift 中的身分和存取管理
- 使用 管理 Amazon Redshift 管理員密碼 AWS Secrets Manager
- 在 Amazon Redshift 中記錄和監控
- Amazon Redshift 的合規驗證
- Amazon Redshift 中的復原能力
- Amazon Redshift 的基礎設施安全
- Amazon Redshift 中的組態與漏洞分析

Amazon Redshift 中的資料保護

共同責任模型適用於 AWS Amazon Redshift 中的資料保護。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。 如需歐洲資料保護的詳細資訊，請參閱一般資料保護規則 (GDPR) 中心。

基於資料保護目的，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。

- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 設定 API 和使用者活動記錄 AWS CloudTrail。如需有關使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 追蹤。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-3。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用 Amazon Redshift 或使用主控台、API AWS CLI或 AWS SDKs的其他 AWS 服務 時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

資料加密

資料保護是指保護往返 Amazon Redshift 的傳輸中資料，以及儲存在 Amazon Redshift 資料中心內磁碟的靜態資料。您可以透過 SSL 或用戶端加密來保護傳輸中的資料。在 Amazon Redshift 中保護靜態資料有下列選項。

- 使用伺服器端加密 — 您可以請求 Amazon Redshift 來加密資料，再將它儲存至其資料中心內的磁碟，然後在您下載時解密物件。
- 使用用戶端加密 — 您可以在用戶端加密資料，再將加密的資料上傳至 Amazon Redshift。在這種情況下，您可以管理加密程序、加密金鑰和相關工具。

靜態加密

伺服器端加密與靜態資料加密有關；亦即，Amazon Redshift 會選擇性地在將資料寫入資料中心時將其加密，以及在您存取該資料時將其解密。只要您有驗證請求並具備存取許可，存取加密資料或未加密資料的方式並無不同。

Amazon Redshift 會透過加密保護靜態資料。您可以選擇性地使用進階加密標準 AES-256 來保護儲存在叢集內磁碟上的所有資料和 Amazon S3 中的所有備份。

若要管理用於加密和解密 Amazon Redshift 資源的金鑰，您可以使用 AWS Key Management Service (AWS KMS). AWS KMS combines 安全、高可用性的硬體和軟體，以提供針對雲端擴展的金鑰管理系統。使用 AWS KMS，您可以建立加密金鑰並定義控制如何使用這些金鑰的政策。 AWS KMS 支援 AWS CloudTrail，因此您可以稽核金鑰使用情況，以確認金鑰是否適當使用。您可以搭配 Amazon

Redshift 和支援的 AWS 服務使用 AWS KMS 金鑰。如需 支援的 服務清單 AWS KMS，請參閱《AWS Key Management Service 開發人員指南》中的AWS 服務使用方式 AWS KMS。

如果您選擇使用 管理佈建叢集或無伺服器命名空間的管理員密碼 AWS Secrets Manager，Amazon Redshift 也會接受使用 AWS Secrets Manager 來加密憑證的額外 AWS KMS 金鑰。此額外金鑰可以是自動產生的金鑰 AWS Secrets Manager，或您提供的自訂金鑰。

Amazon Redshift 查詢編輯器 v2 會安全地儲存輸入到查詢編輯器中的資訊，如下所示：

- 用來加密查詢編輯器 v2 資料的 KMS 金鑰 Amazon Resource Name (ARN)。
- 資料庫連線資訊。
- 檔案和資料夾的名稱與內容。

Amazon Redshift 查詢編輯器 v2 會搭配您的 KMS 金鑰或服務帳戶 KMS 金鑰，使用區塊層級加密來加密資訊。Amazon Redshift 資料的加密會由 Amazon Redshift 叢集屬性來控制。

主題

- Amazon Redshift 資料庫加密

Amazon Redshift 資料庫加密

在 Amazon Redshift 中，您的資料庫預設為加密狀態，以保護靜態資料。資料庫加密適用於叢集及其快照。

您可以修改未加密的叢集以使用 AWS Key Management Service (AWS KMS) 加密。若要這樣做，您可以使用 AWS擁有的金鑰或客戶受管金鑰。當您修改叢集以啟用 AWS KMS 加密時，Amazon Redshift 會自動將您的資料遷移至新的加密叢集。從加密叢集建立的快照也會加密。您也可以修改叢集和變更 Encrypt database (加密資料庫) 選項，將加密的叢集遷移到未加密的叢集。如需詳細資訊，請參閱變更叢集加密。

雖然您仍然可以在建立叢集後，將預設加密的叢集轉換為未加密，但建議您讓包含敏感資料的叢集保持加密狀態。此外，根據控管您的資料的指導方針或規範，可能會要求您使用加密。例如，支付卡產業資料安全標準 (PCI DSS)、沙賓法案 (SOX)、健康保險流通與責任法案 (HIPAA)和其他這類法規可提供用於處理特定資料類型的指導方針。

Amazon Redshift 使用加密金鑰的階層來加密資料庫。您可以使用 AWS Key Management Service (AWS KMS) 或硬體安全模組 (HSM) 來管理此階層中最上層的加密金鑰。Amazon Redshift 用於加密的程序會因您管理金鑰的方式而有所不同。Amazon Redshift 會自動與 整合 AWS KMS ，但不會與 HSM

整合。使用 HSM 時，您必須使用用戶端和伺服器憑證來設定 Amazon Redshift 與您的 HSM 之間的信任連線。

> **Important**
>
> 當您停用客戶自管 KMS 金鑰時，Amazon Redshift 可能會失去佈建叢集或無伺服器命名空間的 KMS 金鑰存取權。在上述情況下，Amazon Redshift 會備份 Amazon Redshift 資料倉儲，並使其進入 `inaccessible-kms-key` 狀態，為期 14 天。如果您在該期間內還原 KMS 金鑰，Amazon Redshift 將會還原存取權，且倉儲將會正常運作。如果 14 天期間結束後 KMS 金鑰仍未還原，Amazon Redshift 將會刪除資料倉儲。倉儲處於 `inaccessible-kms-key` 狀態時具有下列特性：
>
> - 您無法在資料倉儲上執行任何查詢。
> - 如果資料倉儲是資料共用的生產者倉儲，則您無法從取用者倉儲對其執行資料共用查詢。
> - 您無法建立跨區域快照副本。
>
> 如需還原已停用 KMS 金鑰的相關資訊，請參閱《AWS Key Management Service 開發人員指南》中的啟用和停用金鑰。如果倉儲的 KMS 金鑰已刪除，您可以在 `inaccessible-kms-key` 狀態的倉儲遭到刪除之前，使用備份建立新的資料倉儲。

可提升效能與可用性的加密程序改善

使用 RG 和 RA3 節點加密

更新 RG 和 RA3 節點的加密程序，讓體驗變得更好。加密程序進行期間，讀取和寫入查詢在執行時的效能比較不會因加密而受到影響了。此外，加密完成的速度也快了許多。已更新的程序步驟包括還原操作，以及將叢集中繼資料遷移至目標叢集的操作。改善的體驗適用於加密類型 AWS KMS，例如 。當您擁有 PB 規模的資料磁碟區時，操作已從幾週減少為幾天。

在加密叢集之前，如果您打算繼續執行資料庫工作負載，則可以透過新增具有彈性調整大小功能的節點來改善效能並加快程序的進行速度。加密進行期間中，您無法使用彈性調整大小功能，因此請在加密之前執行此操作。請注意，新增節點一般會導致成本上升。

使用其他節點類型進行加密

當您使用 DC2 節點加密叢集時，您無法執行寫入查詢，例如使用 RG 或 RA3 節點。您只能執行讀取查詢。

使用 RG 和 RA3 節點加密的用量備註

下列洞察和資源可協助您為加密做好準備並監控加密程序。

- 在啟動加密後執行查詢 — 啟動加密後，可在大約 15 分鐘內執行讀取和寫入操作。整個加密程序所需的完成時間取決於叢集上的資料量和工作負載層級。
- 加密需要多久時間？ — 加密資料所需的時間取決於幾個因素：這些因素包括執行的工作負載數量、使用的運算資源、節點數目及節點類型。建議您一開始先在測試環境中執行加密。根據經驗，如果您要處理數 PB 的資料磁碟區，則可能需要 1 至 3 天才能完成加密。
- 如何得知加密已完成？ - 在您啟用加密後，第一個快照完成即確認加密已完成。
- 復原加密 — 如果您需要復原加密操作，最好的方法是從在啟動加密前所製作的最新備份進行還原。您必須重新套用在上次備份後所做過的任何新更新 (更新/刪除/插入)。
- 執行資料表還原 — 請注意，您無法將資料表從未加密的叢集還原至已加密的叢集。
- 加密單一節點叢集 — 加密單一節點叢集有效能限制。其所需時間比加密多節點叢集還要久。
- 在加密後建立備份 — 當您加密叢集中的資料時，要等到整個叢集完成加密後，才會建立備份。此操作所需的時間會有所不同。視叢集大小而定，備份所需的時間從幾小時到幾天不等。加密完成後，會延遲一段時間才能建立備份。

  請注意，由於在加密程序進行期間會發生備份與還原操作，因此使用 BACKUP NO 所建立的任何資料表或具體化視觀表不會保留下來。如需詳細資訊，請參閱 CREATE TABLE 或 CREATE MATERIALIZED VIEW。

主題

- 使用 加密 AWS KMS
- 使用硬體安全模組進行加密
- 加密金鑰輪換
- 變更叢集加密
- 遷移至 HSM 加密的叢集
- 輪換加密金鑰

使用 加密 AWS KMS

當您選擇使用 Amazon Redshift AWS KMS 進行金鑰管理時，加密金鑰有四層階層。這些金鑰按層次順序列出為根金鑰、叢集加密金鑰 (CEK)、資料庫加密金鑰 (DEK) 和資料加密金鑰。

當您啟動叢集時，Amazon Redshift 會傳回 Amazon Redshift AWS KMS keys 或 AWS 您的帳戶已建立或具有使用許可的 清單 AWS KMS。您可以選取 KMS 金鑰作為您在加密階層中的根金鑰。

根據預設，Amazon Redshift 會選取自動產生的 AWS擁有金鑰，做為 AWS 您的帳戶在 Amazon Redshift 中使用的根金鑰。

如果您不想使用預設金鑰，您必須在 中分別擁有 （或建立） 客戶受管 KMS 金鑰， AWS KMS 才能在 Amazon Redshift 中啟動叢集。客戶受管金鑰可為您提供更多彈性，包含建立、輪換、停用、定義存取控制，以及稽核用於協助保護您資料之加密金鑰的能力。如需建立 KMS 金鑰的相關資訊，請參閱《AWS Key Management Service 開發人員指南》中的建立金鑰。

如果您想要使用來自另一個 AWS 帳戶的 AWS KMS 金鑰，您必須具有使用金鑰的許可，並在 Amazon Redshift 中指定其 Amazon Resource Name (ARN)。如需存取 中金鑰的詳細資訊 AWS KMS，請參閱《 AWS Key Management Service 開發人員指南》中的控制對金鑰的存取。

選擇根金鑰後，Amazon Redshift 會請求 AWS KMS 產生資料金鑰，並使用選取的根金鑰對其進行加密。此資料金鑰在 Amazon Redshift 中用作 CEK。 AWS KMS 會將加密的 CEK 匯出至 Amazon Redshift，其中，它會與 KMS 金鑰的授予和 CEK 的加密內容一併儲存在與叢集分開的網路磁碟內。只有加密的 CEK 會匯出至 Amazon Redshift；KMS 金鑰會保留在 AWS KMS中。Amazon Redshift 也會透過安全通道將加密的 CEK 傳遞到叢集，並將其載入到記憶體。然後，Amazon Redshift 會呼叫 AWS KMS 來解密 CEK，並將解密的 CEK 載入記憶體。如需授予、加密內容和其他 AWS KMS相關概念的詳細資訊，請參閱《 AWS Key Management Service 開發人員指南》中的概念。

接下來，Amazon Redshift 會隨機產生金鑰來用作 DEK，並在叢集中將它載入記憶體。解密的 CEK 可用來加密 DEK，然後透過安全管道從 Amazon Redshift 要在內部儲存的叢集，傳遞至與叢集不同網路中的磁碟。如同 CEK，加密和解密版本的 DEK 會在叢集中載入記憶體。接著使用 DEK 的解密版本來加密為資料庫中每個資料區塊隨機產生的個別加密金鑰。

當叢集重新啟動時，Amazon Redshift 會從內部存放且加密的 CEK 和 DEK 版本開始，將它們重新載入記憶體，然後再次呼叫 AWS KMS 以使用 KMS 金鑰解密 CEK，以便將其載入記憶體。然後使用解密的 CEK 來再次解密，而解密的 DEK 會載入記憶體，並視需要用來加密和解密資料區塊金鑰。

如需建立使用 AWS KMS 金鑰加密的 Amazon Redshift 叢集的詳細資訊，請參閱 建立叢集。

將 AWS KMS加密的快照複製到另一個快照 AWS 區域

AWS KMS 金鑰專屬於 AWS 區域。如果您想要啟用將 Amazon Redshift 快照從加密來源叢集複製到另一個叢集 AWS 區域，但想要將自己的 AWS KMS 金鑰用於目的地中的快照，則需要設定授予 Amazon Redshift 在目的地的帳戶中使用根金鑰 AWS 區域。此授權可讓 Amazon Redshift 在目的地

AWS 區域中加密快照。如果您希望目的地中的快照使用 AWS 區域擁有的金鑰加密，則不需要在目的地中設定任何授予 AWS 區域。如需跨區域快照複製的相關資訊，請參閱將快照複製到另一個 AWS 區域。

> **Note**
>
> 如果您啟用從加密叢集複製快照並 AWS KMS 用於根金鑰，則無法重新命名叢集，因為叢集名稱是加密內容的一部分。如果您必須重新命名叢集，您可以停用來源區域中快照的複製 AWS 、重新命名叢集，然後再次設定和啟用快照的複製。

為複製快照設定授予的程序如下所示。

1. 在目的地 AWS 區域中，執行下列動作來建立快照複製授予：
   - 如果您還沒有要使用的 AWS KMS 金鑰，請建立一個。如需建立 AWS KMS 金鑰的詳細資訊，請參閱《 AWS Key Management Service 開發人員指南》中的建立金鑰。
   - 指定快照複製授予的名稱。此名稱在 AWS 帳戶的該 AWS 區域中必須是唯一的。
   - 指定您要為其建立授予的 AWS KMS 金鑰 ID。如果您不指定金鑰 ID，該授予即會套用至您的預設金鑰。
2. 在來源 AWS 區域中，啟用快照的複製，並指定您在目的地 AWS 區域中建立的快照複製授予名稱。

只有在您使用 AWS CLI、Amazon Redshift API 或 SDKs 啟用快照複製時，才需要上述程序。如果是使用主控台，在啟用跨區域快照複製時，Amazon Redshift 會提供適當的工作流程來設定授予。如需使用主控台來為 AWS KMS加密叢集設定跨區域快照複製的相關資訊，請參閱為 AWS KMS加密的叢集設定跨區域快照複本。

在將快照複製到目的地 AWS 區域之前，Amazon Redshift 會使用來源區域中的根金鑰解密快照，AWS 並使用 Amazon Redshift 內部管理的隨機產生 RSA 金鑰暫時重新加密快照。然後，Amazon Redshift 透過安全通道將快照複製到目的地 AWS 區域，使用內部受管 RSA 金鑰解密快照，然後使用目的地區域中的根金鑰重新加密快照 AWS 。

使用硬體安全模組進行加密

如果您不使用 AWS KMS 進行金鑰管理，則可以使用硬體安全模組 (HSM) 搭配 Amazon Redshift 進行金鑰管理。

> **Important**
>
> DC2、RG 和 RA3 節點類型不支援 HSM 加密。

HSM 為可針對金鑰產生和管理提供直接控制的裝置。其會透過將金鑰管理與應用程式資料庫層分開以提供更好的安全性。Amazon Redshift 支援 AWS CloudHSM Classic 進行金鑰管理。當您使用 HSM 管理加密金鑰而非 時，加密程序會有所不同 AWS KMS。

> **Important**
>
> Amazon Redshift 僅支援 AWS CloudHSM Classic。我們不支援較新的 AWS CloudHSM 服務。
> AWS CloudHSM Classic 已向新客戶關閉。如需詳細資訊，請參閱 CloudHSM Classic Pricing。並非所有 AWS 區域都提供 AWS CloudHSM Classic。如需可用 AWS 區域的詳細資訊，請參閱AWS 區域表。

設定您的叢集使用 HSM 時，Amazon Redshift 會將請求傳送至 HSM 以產生和儲存要用作 CEK 的金鑰。不過，不同於 AWS KMS HSM 不會將 CEK 匯出至 Amazon Redshift。Amazon Redshift 會改為在叢集中隨機產生 DEK，並將它傳遞至 HSM 供 CEK 加密。HSM 會將已加密的 DEK 傳回給 Amazon Redshift，在此處使用隨機產生的內部根金鑰進行進一步加密，然後儲存在與叢集位於不同網路的磁碟內。Amazon Redshift 也會將 DEK 的解密版本載入到叢集的記憶體中，以便使用 DEK 來加密和解密資料區塊的個別金鑰。

如果叢集重新開機，Amazon Redshift 會使用內部根金鑰來解密內部儲存、雙重加密的 DEK，以讓在內部儲存的 DEK 回到 CEK 加密的狀態。然後將 CEK 加密的 DEK 傳遞至要解密的 HSM，並傳回 Amazon Redshift，在該處可將它再次載入記憶體，以搭配個別資料區塊金鑰使用。

設定 Amazon Redshift 與 HSM 之間的信任連線

選擇使用 HSM 管理您的叢集金鑰時，您必須在 Amazon Redshift 與您的 HSM 之間設定信任的網路連結。要這麼做需要對用戶端和伺服器憑證進行組態。信任的連線可用來於加密和解密操作期間，在 HSM 與 Amazon Redshift 之間傳遞加密金鑰。

Amazon Redshift 會從隨機產生的私有和公有金鑰對建立公有用戶端憑證。這些憑證會經過加密並在內部儲存。您可以在您的 HSM 中下載和註冊公有用戶端憑證，並將它指派至適用的 HSM 分割區。

您可以提供 Amazon Redshift 相關的 HSM IP 地址、HSM 分割區名稱、HSM 分割區密碼和公有 HSM 伺服器憑證 (其使用內部根金鑰進行加密)。Amazon Redshift 會完成組態程序，並驗證它可連線至

HSM。如果它無法連線，則會讓叢集處於 INCOMPATIBLE_HSM 狀態，並且不會建立叢集。在此情況下，您必須刪除不完整的叢集並重試。

> **Important**
>
> 修改您的叢集以使用不同的 HSM 分割區時，Amazon Redshift 會驗證它可以連線至新的分割區，但無法驗證存在有效的加密金鑰。使用新的分割區之前，您必須將金鑰複寫至新的分割區。如果叢集已重新啟動，並且 Amazon Redshift 找不到有效的金鑰，重新啟動會失敗。如需詳細資訊，請參閱跨 HSM 複寫金鑰。

初始組態之後，如果 Amazon Redshift 無法連線至 HSM，即會記錄事件。如需這些事件的相關資訊，請參閱 Amazon Redshift 事件通知。

加密金鑰輪換

在 Amazon Redshift 中，您可以輪換加密的叢集的加密金鑰。當您啟動金鑰輪換程序時，Amazon Redshift 會為指定叢集以及叢集的任何自動或手動快照輪換 CEK。Amazon Redshift 也會為指定叢集輪換 DEK，但是當快照儲存在 Amazon Simple Storage Service (Amazon S3) 內並使用現有 DEK 進行加密時，則無法輪換快照的 DEK。

在輪換進行中時，叢集會處於 ROTATING_KEYS 狀態直到完成為止，完成時叢集會回到 AVAILABLE 狀態。Amazon Redshift 會在金鑰輪換程序期間處理解密和重新加密。

> **Note**
>
> 您無法為沒有來源叢集的快照輪換金鑰。刪除叢集之前，請考慮其快照是否仰賴於金鑰輪換。

由於叢集金鑰輪換程序期間暫時無法使用，您應該只在您的資料需求或當您懷疑金鑰可能已遭到盜用時才輪換金鑰。做為最佳實務，您應該檢閱儲存的資料類型，並計畫輪換加密該資料之金鑰的頻率。輪換金鑰的頻率因對於資料安全性的公司政策，以及有關敏感資料和法規合規的任何產業標準而有所不同。確保您的計畫可平衡您的叢集對於可用性考量的安全性需求。

如需輪換金鑰的詳細資訊，請參閱 輪換加密金鑰。

變更叢集加密

您可以使用 AWS擁有的金鑰或客戶受管金鑰，將未加密的叢集修改為使用 AWS Key Management Service (AWS KMS) 加密。當您修改叢集以啟用 AWS KMS 加密時，Amazon Redshift 會自動將資料

遷移到新的加密叢集。您也可以使用 修改叢集 AWS CLI，但不能使用 ，將加密的叢集遷移至未加密的叢集 AWS 管理主控台。

在遷移操作期間，只能以唯讀模式使用您的叢集，而且叢集的狀態會顯示為 resizing (調整大小)。

如果您的叢集設定為啟用跨AWS 區域快照複本，您必須在變更加密之前停用它。如需詳細資訊，請參閱將快照複製到另一個 AWS 區域及為 AWS KMS加密的叢集設定跨區域快照複本。您無法經由修改叢集來啟用硬體安全模組 (HSM) 加密。反之，請建立新的 HSM 加密叢集，然後將您的資料遷移至新叢集。如需詳細資訊，請參閱遷移至 HSM 加密的叢集。

Amazon Redshift console

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後選擇您要修改加密的叢集。
3. 選擇 Properties (屬性)。
4. 在資料庫組態區段中，選擇編輯，然後選擇編輯加密。
5. 選擇其中一個加密選項，然後選擇儲存變更。

AWS CLI

若要修改未加密的叢集以使用 AWS KMS，請執行 `modify-cluster` CLI 命令並指定 `--encrypted`，如下所示。在預設情況下會使用預設 KMS 金鑰。若要指定客戶受管金鑰，請包含 `--kms-key-id` 選項。

```
aws redshift modify-cluster --cluster-identifier <value> --encrypted --kms-key-id
 <value>
```

若要從叢集移除加密，請執行下列 CLI 命令。

```
aws redshift modify-cluster --cluster-identifier <value> --no-encrypted
```

遷移至 HSM 加密的叢集

若要將未加密的叢集遷移至使用硬體安全模組 (HSM) 加密的叢集，請建立新的加密叢集，然後將您的資料移至新叢集。您無法經由修改叢集來遷移 HSM 加密的叢集。

若要從未加密的叢集遷移至 HSM 加密的叢集，請先從現有的來源叢集卸載您的資料。然後在新的目標叢集中，使用所選的加密設定將資料重新載入。如需啟動加密的叢集的相關資訊，請參閱Amazon Redshift 資料庫加密。

遷移程序期間，您的來源叢集可供唯讀查詢，直到最後一個步驟為止。最後一個步驟是重新命名目標和來源叢集，這會切換端點，使得所有流量會路由至新的目標叢集。在您重新命名後重新開機之前，目標叢集將無法使用。在資料傳送時，請暫停來源叢集上的所有資料載入和其他寫入操作。

準備遷移

1. 識別與 Amazon Redshift 互動的所有相依系統，例如商業智慧 (BI) 工具和擷取、轉換和載入 (ETL) 系統。
2. 識別驗證查詢以測試遷移。

   例如，您可以使用下列查詢來尋找使用者定義資料表的數目。

   ```
   select count(*)
   from pg_table_def
   where schemaname != 'pg_catalog';
   ```

   下列查詢會傳回所有使用者定義資料表的清單，和每個資料表中資料列的數目。

   ```
   select "table", tbl_rows
   from svv_table_info;
   ```
3. 選擇適合遷移的好時機。若要了解叢集使用率何時最低，請監控叢集指標，例如 CPU 使用率和資料庫連線數目。如需詳細資訊，請參閱檢視叢集效能資料。
4. 捨棄未使用的資料表。

   若要建立資料表的清單和查詢每個資料表的次數，請執行下列查詢。

   ```
   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. 啟動新的加密叢集。

   對目標叢集使用與來源叢集相同的連線埠號碼。如需啟動加密的叢集的相關資訊，請參閱Amazon Redshift 資料庫加密。

6. 設定卸載和載入程序。

   您可以使用 Amazon Redshift Unload/Copy Utility 來幫助您在叢集之間遷移資料。該公用程式會從來源叢集匯出資料至 Amazon S3 上的位置。資料會使用 加密 AWS KMS。然後該公用程式會將資料自動匯入至目標。您可以在遷移完成之後，選擇性地使用該公用程式來清理 Amazon S3。

7. 執行測試以驗證您的程序並估計寫入操作必須暫停的時間長度。

   在卸載和載入操作期間，透過暫停資料載入和其他寫入操作來保有資料一致性。使用您的其中一個最大的資料表，執行卸載和載入程序來幫助您估計時間。

8. 建立資料庫物件，例如結構描述、檢視和資料表。為了協助您產生必要的資料定義語言 (DDL) 陳述式，您可以在 AWS GitHub 儲存庫的 AdminViews 中使用指令碼。

遷移叢集

1. 停止來源叢集上的所有 ETL 處理。

   若要確認程序中沒有寫入操作，請使用 Amazon Redshift 管理主控台來監控寫入 IOPS。如需詳細資訊，請參閱檢視叢集效能資料。

2. 執行您稍早識別的驗證查詢，以在遷移之前收集未加密的來源叢集的相關資訊。

3. (選用) 建立一個工作負載管理 (WLM) 佇列以使用來源和目標叢集中最大可用的資源。例如，建立名為 `data_migrate` 的佇列，並設定具有 95% 記憶體和並行 4 的佇列。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的根據使用者群組和查詢群組將查詢路由至佇列。

4. 使用 `data_migrate` 佇列，執行 UnloadCopyUtility。

   使用 Amazon Redshift 主控台監控 UNLOAD 和 COPY 程序。

5. 再次執行驗證查詢，並驗證該結果符合來源叢集的結果。

6. 重新命名您的來源和目標叢集以交換端點。為了避免干擾，請在上班時間以外執行此操作。
7. 驗證您可以使用您的所有 SQL 用戶端 (例如 ETL 和報表工具) 連線至目標叢集。
8. 關閉未加密的來源叢集。

輪換加密金鑰

您可以使用下列程序，藉由使用 Amazon Redshift 主控台來輪換加密金鑰。

輪換叢集的加密金鑰

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後選擇您要更新加密金鑰的叢集。
3. 針對 Actions (動作)，選擇 Rotate encryption (輪換加密) 以顯示 Rotate encryption keys (輪換加密金鑰) 頁面。
4. 在 Rotate encryption keys (輪換加密金鑰) 頁面上，選擇 Rotate encryption keys (輪換加密金鑰)。

傳輸中加密

您可以設定您的環境以保護傳輸中的資料的機密性及完整性。

以下詳細資訊適用於在 Amazon Redshift 叢集與 SQL 用戶端之間透過 JDBC/ODBC 加密傳輸中的資料：

- 您可以從 SQL 用戶端工具透過 Java 資料庫連線 (JDBC) 和開放式資料庫連線 (ODBC) 來連線至 Amazon Redshift 叢集。
- Amazon Redshift 支援 Secure Sockets Layer (SSL) 連線，可將資料和伺服器憑證加密，以驗證用戶端所連線的伺服器憑證。用戶端會連線到 Amazon Redshift 叢集的領導節點。如需詳細資訊，請參閱設定連線的安全選項。
- 為了支援 SSL 連線，Amazon Redshift 會在每個叢集上建立並安裝 AWS Certificate Manager (ACM) 發行的憑證。如需詳細資訊，請參閱將 SSL 連線轉換為使用 ACM 憑證。
- 為了保護您的 AWS 雲端內傳輸中的資料，Amazon Redshift 使用硬體加速 SSL 與 Amazon S3 或 Amazon DynamoDB 通訊，以進行 COPY、UNLOAD、備份和還原操作。

以下詳細資訊適用於在 Amazon Redshift 叢集與 Amazon S3 或 DynamoDB 之間加密傳輸中的資料：

- Amazon Redshift 使用硬體加速的 SSL 來與 Amazon S3 或 DynamoDB 通訊，以進行 COPY、UNLOAD、備份和還原操作。
- Redshift Spectrum 使用由 (KMS) 管理的帳戶預設金鑰，支援 Amazon S3 伺服器端加密 AWS Key Management Service (SSE)。
- 您可以使用 Amazon S3 和 加密 Amazon Redshift 負載 AWS KMS。如需詳細資訊，請參閱使用 Amazon S3 加密 Amazon Redshift Loads 和 AWS KMS 。

下列詳細資訊適用於在 AWS CLI SDK 或 API 用戶端與 Amazon Redshift 端點之間傳輸中的資料的加密和簽署：

- Amazon Redshift 會提供 HTTPS 端點用於加密傳輸中資料。
- 為了保護向 Amazon Redshift 發出之 API 請求的完整性，必須由發起人簽署 API 呼叫。根據 Signature 第 4 版簽署程序 (Sigv4)，呼叫由 X.509 憑證或客戶的 AWS 私密存取金鑰進行簽署。如需詳細資訊，請參閱 AWS 一般參考 中的 Signature 第 4 版簽署程序。
- 使用 AWS CLI 或其中一個 AWS SDKs 向 提出請求 AWS。這些工具會自動使用您設定工具時指定的存取金鑰，替您簽署請求。

以下詳細資訊適用於在 Amazon Redshift 叢集與 Amazon Redshift 查詢編輯器 v2 之間加密傳輸中的資料：

- 資料會透過 TLS 加密的通道在查詢編輯器 v2 和 Amazon Redshift 叢集之間傳輸。

Amazon Redshift 的 VPC 加密控制

Amazon Redshift 支援 VPC 加密控制，這項安全功能可協助您針對區域中 VPCs內的所有流量強制執行傳輸中加密。本文件說明如何搭配 Amazon Redshift 叢集和無伺服器工作群組使用 VPC 加密控制。

VPC 加密控制提供集中式控制，以監控和強制執行 VPCs內傳輸中的加密。在強制模式下啟用時，它可確保在硬體層 （ 使用 AWS Nitro System) 或應用程式層 （ 使用 TLS/SSL) 加密所有網路流量。

Amazon Redshift 與 VPC 加密控制整合，協助您符合醫療保健 (HIPAA)、政府 (FedRAMP) 和金融 (PCI DSS) 等產業的合規要求。

VPC 加密控制如何與 Amazon Redshift 搭配使用

VPC 加密控制以兩種模式運作：

- 監控模式：提供流量加密狀態的可見性，並協助識別允許未加密流量的資源。

- 強制模式：防止建立或使用允許 VPC 內未加密流量的資源。所有流量都必須在硬體層 (Nitro 型執行個體） 或應用程式層 (TLS/SSL) 加密。

使用 VPC 加密控制的需求

執行個體類型需求

Amazon Redshift 需要 Nitro 型執行個體來支援 VPC 加密控制。所有現代 Redshift 執行個體類型都支援必要的加密功能。

SSL/TLS 要求

在強制執行模式下啟用 VPC 加密控制時，必須將 require_ssl 參數設定為 true，且無法停用。這可確保所有用戶端連線都使用加密的 TLS 連線。

遷移至 VPC 加密控制

對於現有叢集和工作群組

您無法在包含現有 Redshift 叢集或無伺服器工作群組的 VPC 上以強制執行模式啟用 VPC 加密控制。如果您有現有的叢集或工作群組，請參閱下列步驟以使用加密控制：

1. 建立現有叢集或命名空間的快照
2. 使用強制執行模式中啟用的 VPC 加密控制建立新的 VPC
3. 使用下列其中一個操作，從快照還原至新的 VPC：
    - 對於佈建的叢集：使用 `restore-from-cluster-snapshot`操作
    - 對於無伺服器：在您的工作群組上使用 `restore-from-snapshot`操作

在已啟用加密控制的 VPC 中建立新的叢集或工作群組時，需將 require_ssl 參數設為 true。

Amazon Redshift 需要 Nitro 型執行個體來支援 VPC 加密控制。所有現代 Redshift 執行個體類型都支援必要的加密功能。

SSL/TLS 要求

在強制執行模式下啟用 VPC 加密控制時，必須將 require_ssl 參數設定為 true，且無法停用。這可確保所有用戶端連線都使用加密的 TLS 連線。

考量和限制

在 Amazon Redshift 中使用 VPC 加密控制時，請考慮下列事項：

VPC 狀態限制

- 當 VPC 加密控制處於 `enforce-in-progress` 狀態時，會封鎖叢集和工作群組建立
- 您必須等到 VPC 達到 `enforce` 模式，才能建立新的資源

SSL 組態

- require_ssl 參數：必須一律`true`適用於在加密強制執行 VPCs 中建立的叢集和工作群組
- 在加密強制執行的 VPC 中建立叢集或工作群組後，就`require_ssl`無法在其生命週期內停用

區域可用性

在下列區域中，Amazon Redshift Serverless 的強制執行模式中無法使用此功能：

- 南美洲 (聖保羅)
- 歐洲 (蘇黎世)

金鑰管理

您可以設定您的環境以使用金鑰保護資料：

- Amazon Redshift 會自動與 key management. AWS KMS uses 信封加密的 AWS Key Management Service (AWS KMS) 整合。如需詳細資訊，請參閱信封加密。
- 在 中管理加密金鑰時 AWS KMS，Amazon Redshift 會使用以金鑰為基礎的四層架構進行加密。架構包含隨機產生的 AES-256 資料加密金鑰、資料庫金鑰、叢集金鑰和根金鑰。如需詳細資訊，請參閱 Amazon Redshift 如何使用 AWS KMS。
- 您可以在 AWS KMS中建立您自己的客戶受管金鑰。如需詳細資訊，請參閱建立 KMS 金鑰和 授權搭配 Amazon Redshift （佈建） 和 Amazon Redshift Serverless 使用 AWS KMS 客戶受管金鑰。
- 您也可以為新的 匯入自己的金鑰材料 AWS KMS keys。如需詳細資訊，請參閱在 AWS Key Management Service (AWS KMS) 中匯入金鑰材料。
- Amazon Redshift 支援在外部硬體安全模組 (HSM) 中加密金鑰的管理。HSM 可以是內部部署，也可以是 AWS CloudHSM。使用 HSM 時，您必須使用用戶端和伺服器憑證來設定 Amazon Redshift 與您的 HSM 之間的信任連線。Amazon Redshift 僅支援 AWS CloudHSM Classic 進行金鑰管理。

如需詳細資訊，請參閱使用硬體安全模組進行加密。如需 的資訊 AWS CloudHSM，請參閱什麼是 AWS CloudHSM？

- 您可以輪換加密的叢集的加密金鑰。如需詳細資訊，請參閱加密金鑰輪換。

授權搭配 Amazon Redshift （佈建） 和 Amazon Redshift Serverless 使用 AWS KMS 客戶受管金鑰

Amazon Redshift 與 AWS Key Management Service (AWS KMS) 整合，以加密靜態資料。當您使用客戶受管金鑰時，您可以完全控制金鑰政策、金鑰輪換，以及啟用或停用金鑰。如需客戶受管金鑰的詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的客戶受管金鑰。

當 Amazon Redshift 在密碼編譯操作中使用客戶受管金鑰時，它會代表建立或修改 Amazon Redshift 佈建叢集或 Amazon Redshift Serverless 資源的使用者。

所需的許可

若要建立和使用以客戶受管金鑰加密的 Amazon Redshift 佈建叢集或 Amazon Redshift Serverless 資源，使用者必須具有在客戶受管金鑰上呼叫下列操作的許可：

- `kms:CreateGrant`
- `kms:GenerateDataKey`
- `kms:Encrypt`
- `kms:Decrypt`
- `kms:DescribeKey`

客戶受管金鑰的擁有者或管理員可以在客戶受管金鑰的金鑰政策中為特定使用者新增這些許可，如果金鑰政策允許，則可以在 IAM 政策中新增這些許可。

Amazon Redshift 會透過 AWS KMS 授權存取客戶受管金鑰。請確定您的金鑰政策不會阻止 Amazon Redshift 在金鑰上建立或使用授予。例如，明確拒絕陳述式或允許陳述式的限制條件，可能會封鎖授予建立並導致操作失敗。

作為安全最佳實務，我們建議您僅提供使用案例所需的許可。例如，您可以使用 等條件索引鍵縮小`kms:CreateGrant`許可的範圍`kms:GrantIsForAWSResource`。

如需控制 AWS KMS 金鑰存取的詳細資訊，請參閱《 AWS Key Management Service 開發人員指南》中的金鑰政策 AWS KMS，以及允許其他帳戶中的使用者使用 KMS 金鑰。

資料字符化

字符化是出於資料安全目的，用不透明值替換實際值的程序。對安全敏感的應用程式會使用字符化，以字符替換敏感資料 (例如，個人身分識別資訊 (PII) 或受保護的醫療資訊 (PHI)) 來降低安全風險。去字符化會為授權使用者使用適當的安全政策將字符逆轉為實際值。

若要與第三方字符化服務整合，您可以使用您透過 AWS Lambda 建立的 Amazon Redshift 使用者定義函數 (UDF)。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 Lambda 使用者定義的函數。例如，請參閱 Protegrity。

Amazon Redshift 會將字符化請求傳送至透過 REST API 或預先定義的端點所存取的字符化伺服器。有兩個以上免費的 Lambda 函數會處理字符化和去字符化請求。對於此處理，您可以使用第三方字符化提供者所提供的 Lambda 函數。您也可以使用在 Amazon Redshift 中註冊為 Lambda UDF 的 Lambda 函數。

例如，假設您提交的查詢會在資料欄叫用字符化或去字符化 UDF。Amazon Redshift 叢集會對適用的引數資料列進行多工緩衝處理，並以批次方式將這些資料列平行傳送至 Lambda 函數。資料會在 Amazon Redshift 運算節點與 Lambda 之間另外以用戶端無法存取的隔離網路連線進行傳輸。Lambda 函數會將資料傳遞至字符化伺服器端點。字符化伺服器會根據需要對資料進行字符化或去字符化，並將其傳回。然後，Lambda 函數會將結果傳輸到 Amazon Redshift 叢集以進行進一步處理 (如有必要)，然後傳回查詢結果。

在 Amazon Redshift 中路由網際網路流量

您可以在 Amazon Redshift 中透過已知和私有網路路由來路由流量。本頁涵蓋如何在企業網路上以及相同 AWS 區域中的資源之間路由流量。

若要在 Amazon Redshift 與企業網路上的用戶端和應用程式之間路由流量：

- 在虛擬私有雲端 (VPC) 和公司網路之間設定私有連線。透過網際網路設定 IPsec VPN 連線或使用 Direct Connect 連線建立私有實體連線。 Direct Connect 可讓您直接從內部部署網路建立私有虛擬介面至 Amazon VPC，從而在網路和 VPC 之間提供私有、高頻寬的網路連線。您可使用多個虛擬界面，甚至能與多個 VPC 建立私有連線，同時保持網路隔離性。如需詳細資訊，請參閱什麼是 AWS Site-to-Site VPN？和什麼是 Direct Connect？

若要在 VPC 中的 Amazon Redshift 叢集與相同 AWS 區域中的 Amazon S3 儲存貯體之間路由流量：

- 設定 Amazon S3 私有 VPC 端點，以私密地存取來自 ETL 載入或卸載的 Amazon S3 資料。如需詳細資訊，請參閱 Amazon S3 的端點。

- 為 Amazon Redshift 叢集啟用「增強型 VPC 路由」，指定目標 Amazon S3 VPC 端點。然後 Amazon Redshift COPY、UNLOAD 或 CREATE LIBRARY 命令產生的流量會透過私有端點路由。如需詳細資訊，請參閱開啟增強型 VPC 路由。

Amazon Redshift 中的身分和存取管理

存取 Amazon Redshift 需要 AWS 可用於驗證請求的登入資料。這些登入資料必須具有存取 AWS 資源的許可，例如 Amazon Redshift 叢集。以下章節提供關於如何使用 AWS Identity and Access Management (IAM) 與 Amazon Redshift 的詳細資訊，藉由控制可存取的人員，協助保護您資源的安全：

- 使用身分進行身分驗證
- 存取控制

> **Important**
>
> 本主題集合了用於管理許可、身分和安全存取的最佳實務。建議您熟悉將 IAM 與 Amazon Redshift 搭配使用的最佳實務。這些實務包括使用用於套用許可的 IAM 角色。充分了解這些部分會可助您保持更安全的 Amazon Redshift 資料倉儲。

使用身分進行身分驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者、IAM 使用者或擔任 IAM 角色身分進行身分驗證。

您可以使用身分來源的登入資料，例如 AWS IAM Identity Center (IAM Identity Center)、單一登入身分驗證或 Google/Facebook 登入資料，以聯合身分的形式登入。如需有關登入的詳細資訊，請參閱《AWS 登入 使用者指南》中的如何登入您的 AWS 帳戶。

對於程式設計存取， AWS 提供 SDK 和 CLI 以密碼編譯方式簽署請求。如需詳細資訊，請參閱《IAM 使用者指南》中的 API 請求的AWS 第 4 版簽署程序。

AWS 帳戶 根使用者

當您建立 時 AWS 帳戶，您會從一個名為 AWS 帳戶 theroot 使用者的登入身分開始，該身分可完整存取所有 AWS 服務 和 資源。強烈建議不要使用根使用者來執行日常任務。有關需要根使用者憑證的任務，請參閱《IAM 使用者指南》中的需要根使用者憑證的任務。

IAM 使用者和群組

IAM 使用者https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html是一種身分具備單人或應用程式的特定許可權。建議以臨時憑證取代具備長期憑證的 IAM 使用者。如需詳細資訊，請參閱《IAM 使用者指南》中的要求人類使用者使用聯合身分提供者來 AWS 使用臨時憑證存取 。

IAM 群組會指定 IAM 使用者集合，使管理大量使用者的許可權更加輕鬆。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 使用者的使用案例。

IAM 角色

IAM 角色https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html的身分具有特定許可權，其可以提供臨時憑證。您可以透過從使用者切換到 IAM 角色 （主控台） 或呼叫 AWS CLI 或 AWS API 操作來擔任角色。如需詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

IAM 角色適用於聯合身分使用者存取、臨時 IAM 使用者許可、跨帳戶存取權與跨服務存取，以及在 Amazon EC2 執行的應用程式。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的快帳戶資源存取。

> **Note**
>
> Redshift 中的轉送存取工作階段 (FAS) 的有效期僅 12 個小時。此期間過後，任何使用 FAS 與其他服務整合的連線工作階段都必須重新建立。

預防跨服務混淆代理人

混淆代理人問題屬於安全性問題，其中沒有執行動作許可的實體可以強制具有更多許可的實體執行該動作。在 中 AWS，跨服務模擬可能會導致混淆代理人問題。在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了預防這種情況， AWS 提供的工具可協助您保護所有服務的資料，而這些服務主體已獲得您帳戶中資源的存取權。如需詳細資訊，請參閱《IAM 使用者指南》中的預防跨服務混淆代理人。

我們建議在資源型政策中使用 `aws:SourceArn`和 `aws:SourceAccount`全域條件內容金鑰，以限制 Amazon Redshift 為資源提供其他服務的許可。如果 `aws:SourceArn` 值不包含帳戶 ID (例如 Amazon Simple Storage Service (Amazon S3) 儲存貯體 ARN)，則必須使用這兩個全域條件內容索引鍵來限制許可。如果同時使用這兩個全域條件內容索引鍵，且 `aws:SourceArn` 值包含帳戶 ID，則在相同政策陳述式中使用 `aws:SourceAccount` 值和 `aws:SourceArn` 值中的帳戶時，必須使用相同的帳戶 ID。

下列範例顯示您可以套用的政策，以限制 Amazon Redshift 的混淆代理人問題。

```
{
  "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"
      }
    }
  }
}
```

下列範例顯示 Amazon Redshift Serverless 的政策。

```
{
  "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"
      }
    }
  }
```

```
}
```

存取控制

您可以持有效憑證來驗證請求，但還須具備許可，才能建立或存取 Amazon Redshift 資源。例如，您必須具備許可，才能建立 Amazon Redshift 叢集、建立快照、新增事件訂閱及其他動作。

以下章節描述如何管理 Amazon Redshift 的許可。我們建議您先閱讀概觀。

- 管理 Amazon Redshift 資源存取許可的概觀
- 將以身分為基礎的政策 (IAM 政策) 用於 Amazon Redshift

管理 Amazon Redshift 資源存取許可的概觀

每個 AWS 資源都由 AWS 帳戶擁有，而建立或存取資源的許可是由許可政策管理。帳戶管理員可以將許可政策連接到 IAM 身分 （即使用者、群組和角色），某些服務 （例如 AWS Lambda) 也支援將許可政策連接到 資源。

> **Note**
>
> 帳戶管理員 (或管理員使用者) 是具有管理員權限的使用者。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 最佳實務。

授予許可時，您會決定誰取得這些許可、取得哪些資源的許可，以及可以對那些資源進行的特定動作。

Amazon Redshift 資源和操作

Amazon Redshift 提供服務特有的資源、動作和條件內容索引鍵，可用於 IAM 許可政策。

Amazon Redshift、Amazon Redshift Serverless、Amazon Redshift 資料 API 和 Amazon Redshift 查詢編輯器 v2 的存取許可

當設定 存取控制 時，您要編寫可附加到 IAM 身分的許可政策 (身分型政策)。如需詳細的參考資訊，請參閱《服務授權參考》中的下列主題：

- 若為 Amazon Redshift，請參閱使用 `redshift:` 前置詞的 Amazon Redshift 的動作、資源和條件索引鍵。

- 若為 Amazon Redshift Serverless，請參閱使用 `redshift-serverless:` 前置詞的 Amazon Redshift Serverless 的動作、資源和條件索引鍵。
- 若為 Amazon Redshift 資料 API，請參閱使用 `redshift-data:` 前置詞的 Amazon Redshift 資料 API 的動作、資源和條件索引鍵。
- 對於 Amazon Redshift 查詢編輯器 v2，請參閱使用 `sqlworkbench:`字首的 AWS SQL Workbench (Amazon Redshift 查詢編輯器 v2) 的動作、資源和條件索引鍵。

  查詢編輯器 v2 包含非直接對應至 API 操作的僅限許可動作。《服務授權參考》中會使用 `[permission only]` 指出這些動作。

《服務授權參考》包含了可在 IAM 政策中使用哪些 API 操作的相關資訊。它還包含您可以授予許可 AWS 的資源，以及您可以包含用於精細存取控制的條件索引鍵。如需條件的相關資訊，請參閱使用 IAM 政策條件進行精細定義存取控制。

您可以在政策的 `Action` 欄位中指定動作、在政策的 `Resource` 欄位中指定資源值，以及在政策的 `Condition` 欄位中指定條件。若要指定 Amazon Redshift 的動作，請使用後接 API 操作名稱的 `redshift:` 前置詞 (例如，`redshift:CreateCluster`)。

了解資源所有權

資源擁有者是建立資源 AWS 的帳戶。也就是說，資源擁有者是驗證建立資源之請求的委託人實體（根帳戶、IAM 使用者或 IAM 角色） AWS 的帳戶。下列範例說明其如何運作：

- 如果您使用 AWS 帳戶的根帳戶登入資料來建立資料庫叢集， AWS 您的帳戶即為 Amazon Redshift 資源的擁有者。
- 如果您在 AWS 帳戶中建立具有建立 Amazon Redshift 資源許可的 IAM 角色，則任何可以擔任該角色的人都可以建立 Amazon Redshift 資源。您的 AWS 帳戶 (即角色所屬帳戶) 會擁有 Amazon Redshift 資源。
- 如果您在 AWS 帳戶中建立 IAM 使用者，並將建立 Amazon Redshift 資源的許可授予該使用者，則使用者可以建立 Amazon Redshift 資源。但是您的 AWS 帳戶 (即使用者所屬帳戶) 會擁有 Amazon Redshift 資源。在大多數情況下，不建議您使用此方法。建議您建立 IAM 角色並將許可附加到該角色，然後將角色指派給使用者。

管理 資源的存取

許可政策描述誰可以存取哪些資源。下一節說明可用來建立許可政策的選項。

> **Note**
>
> 本節著重討論如何在 Amazon Redshift 的環境中使用 IAM。它不提供 IAM 服務的詳細資訊。如需完整的 IAM 文件，請參閱《IAM 使用者指南》中的什麼是 IAM？。如需有關 IAM 政策語法和說明的資訊，請參閱《IAM 使用者指南》中的 AWS IAM 政策參考。

連接到 IAM 身分的政策稱為身分類型政策 (IAM 政策)，而連接到資源的政策參考資源類型政策。Amazon Redshift 僅支援以身分為基礎的政策 (IAM 政策)。

身分類型政策 (IAM 政策)

您可以透過將政策附加到 IAM 角色，然後將該角色指派給使用者或群組來指派許可。下面的範例政策包含用來為您的 AWS 帳戶建立、刪除、修改 Amazon Redshift 叢集和將叢集重新開機的許可。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"AllowManageClusters",
      "Effect":"Allow",
      "Action": [
        "redshift:CreateCluster",
        "redshift:DeleteCluster",
        "redshift:ModifyCluster",
        "redshift:RebootCluster"
      ],
      "Resource":"*"
    }
  ]
}
```

如需將以身分為基礎的政策與 Amazon Redshift 搭配使用的相關資訊，請參閱將以身分為基礎的政策 (IAM 政策) 用於 Amazon Redshift。如需使用者、群組、角色和許可的相關資訊，請參閱《IAM 使用者指南》中的身分 (使用者、群組和角色)。

資源型政策

其他服務 (例如 Amazon S3) 也支援以資源為基礎的許可政策。例如，您可以將政策連接至 S3 儲存貯體，以管理該儲存貯體的存取許可。Amazon Redshift 不支援資源型政策。

指定政策元素：動作、效果、資源和主體

針對每一個 Amazon Redshift 資源 (請參閱 Amazon Redshift 資源和操作)，服務會定義一組 API 操作 (請參閱動作)。Amazon Redshift 會定義一組您可在政策中指定的動作，以授予這些 API 操作的許可。執行一項 API 操作可能需要多個動作的許可。

以下是基本的政策元素：

- 資源 – 在政策中，您可以使用 Amazon Resource Name (ARN) 來識別要套用政策的資源。如需詳細資訊，請參閱Amazon Redshift 資源和操作。
- 動作：使用動作關鍵字識別您要允許或拒絕的資源操作。例如，`redshift:DescribeClusters` 許可會讓使用者獲得執行 Amazon Redshift `DescribeClusters` 操作的許可。
- 效果 - 您可以指定使用者要求特定動作時會有什麼效果；可為允許或拒絕。如果您未明確授予存取 (允許) 資源，則隱含地拒絕存取。您也可以明確拒絕資源存取，這樣做可確保使用者無法存取資源，即使不同政策授予存取也是一樣。
- 委託人：在以身分為基礎的政策 (IAM 政策) 中，政策所連接的使用者就是隱含委託人。對於資源型政策，您可以指定想要收到許可的使用者、帳戶、服務或其他實體 (僅適用於資源型政策)。Amazon Redshift 不支援資源型政策。

如需進一步了解有關 IAM 政策語法和說明的詳細資訊，請參閱《IAM 使用者指南》中的 AWS IAM 政策參考。

如需詳列所有 Amazon Redshift API 動作及適用資源的表格，請參閱 Amazon Redshift、Amazon Redshift Serverless、Amazon Redshift 資料 API 和 Amazon Redshift 查詢編輯器 v2 的存取許可。

在政策中指定條件

當您授予許可時，可以使用存取政策語言來指定政策應該何時生效的條件。例如，建議只在特定日期之後套用政策。如需以存取政策語言指定條件的相關資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。

若要識別許可政策適用的條件，請在您的 IAM 許可政策中包含 `Condition` 元素。例如，您可以建立允許使用者使用 `redshift:CreateCluster` 動作來建立叢集的政策，並且可以新增 `Condition`

元素，將該使用者限制在僅限於特定區域中建立叢集。如需詳細資訊，請參閱使用 IAM 政策條件進行精細定義存取控制。如需呈現所有條件索引鍵值以及 Amazon Redshift 動作和其套用之資源的清單，請參閱 Amazon Redshift、Amazon Redshift Serverless、Amazon Redshift 資料 API 和 Amazon Redshift 查詢編輯器 v2 的存取許可。

使用 IAM 政策條件進行精細定義存取控制

在 Amazon Redshift 中，您可以使用兩個條件索引鍵，根據那些資源的標籤來限制對資源的存取權限。下列是常見的 Amazon Redshift 條件索引鍵。

條件金鑰	說明
aws:RequestTag	每當使用者建立資源時，都會要求使用者包括標籤索引鍵 (名稱) 和值。如需詳細資訊，請參閱《IAM 使用者指南》中的 aws:RequestTag。
aws:ResourceTag	根據特定標籤索引鍵和值，限制使用者對資源的存取。如需詳細資訊，請參閱《IAM 使用者指南》中的 aws:ResourceTag。
aws:TagKeys	使用此鍵來將請求中的標籤鍵與您在政策中所指定的鍵進行比較。如需詳細資訊，請參閱《IAM 使用者指南》中的 aws:TagKeys。

如需標籤的詳細資訊，請參閱在 Amazon Redshift 中標記資源。

如需支援 redshift:RequestTag 和 redshift:ResourceTag 條件索引鍵的 API 動作清單，請參閱Amazon Redshift、Amazon Redshift Serverless、Amazon Redshift 資料 API 和 Amazon Redshift 查詢編輯器 v2 的存取許可。

下列條件索引鍵可以與 Amazon Redshift GetClusterCredentials 動作搭配使用。

條件金鑰	說明
redshift:DurationSeconds	限制可為持續時間指定的秒數。
redshift:DbName	限制可以指定的資料庫名稱。
redshift:DbUser	限制可以指定的資料庫使用者名稱。

範例 1：透過使用 aws:ResourceTag 條件金鑰限制存取

使用下列 IAM 政策，讓使用者僅針對 us-west-2 區域中具有標籤值為 environment之標籤的特定 AWS 帳戶修改 Amazon Redshift 叢集test。

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"
            }
        }
    }
}
```

範例 2：透過使用 aws:RequestTag 條件金鑰限制存取

使用下列 IAM 政策可讓使用者僅在建立叢集的命令包含名為 usage 的標籤和標籤值為 production 時建立 Amazon Redshift 叢集。aws:TagKeys 和 ForAllValues 修飾符的條件指定您在請求中只能指定金鑰 costcenter 和 usage。

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"
                ]
            }
        }
    }
}
```

將以身分為基礎的政策 (IAM 政策) 用於 Amazon Redshift

這個主題提供以身分為基礎的政策範例，在該政策中帳戶管理員可以將許可政策連接至 IAM 身分 (即使用者、群組和角色)。

> **Important**
>
> 建議您先檢閱簡介主題，其中說明基本概念及有何選項可供您管理對 Amazon Redshift 資源的存取。如需詳細資訊，請參閱管理 Amazon Redshift 資源存取許可的概觀。

以下顯示許可政策範例。此政策允許使用者建立、刪除、修改和重新啟動所有叢集，然後拒絕刪除或修改叢集識別符在 AWS 區域 `us-west-2`和 AWS 帳戶 `production`中以 開頭的任何叢集的許可`123456789012`。

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"
    }
  ]
}
```

此政策具有兩個陳述式：

- 第一個陳述式會將某使用者的許可授予給另一位使用者，讓該使用者得以建立、刪除、修改叢集和將它們重新開機。陳述式指定萬用字元 (*) 做為`Resource`值，以便政策套用至根 AWS 帳戶擁有的所有 Amazon Redshift 資源。
- 第二個陳述式會拒絕對叢集的刪除或修改許可。此陳述式會為包含萬用字元 (*) 的 `Resource` 值指定叢集 Amazon Resource Name (ARN)。因此，此陳述式適用於叢集識別符開頭為 的根 AWS 帳戶擁有的所有 Amazon Redshift 叢集`production`。

AWS Amazon Redshift 的 受管政策

AWS 透過提供由 建立和管理的獨立 IAM 政策，解決許多常見的使用案例 AWS。受管政策授與常見使用案例中必要的許可，讓您免於查詢需要哪些許可。如需詳細資訊，請參閱 IAM 使用者指南中的 AWS 受管政策。

您也能建立專屬的自訂 IAM 政策，以允許 Amazon Redshift API 操作與資源的相關許可。您可以將這些自訂政策附加至需要這些許可的 IAM 角色或群組。

下列各節說明 AWS 受管政策，您可以將這些政策連接到您帳戶中的使用者，並且專屬於 Amazon Redshift。

AWS 受管政策的 Amazon Redshift 更新

檢視自此服務開始追蹤 Amazon Redshift AWS 受管政策更新以來的詳細資訊。如需有關此頁面變更的自動提醒，請訂閱 Amazon Redshift 文件歷史記錄頁面上的 RSS 摘要。

變更	描述	日期
AmazonRedshiftFederatedAuthorization – 新政策	Amazon Redshift 新增了使用 Amazon Redshift 聯合授權執行查詢的新ease-of-use政策。	2025 年 11 月 21 日
AmazonRedshiftServiceLinkedRolePolicy – 更新現有政策	`lakeformation:GetDataAccess` 動作的許可已新增至受管政策。新增它可授予從中取得聯合目錄資訊的許可 AWS Lake Formation。 `glue:GetCatalog` 和 `glue:GetCatalogs` 動作的其他條件已新增至受管政策。	2025 年 3 月 13 日
AmazonRedshiftServiceLinkedRolePolicy – 更新現有政策	`glue:GetCatalog` 和 `glue:GetCatalogs` 動作的許可已新增至受管政策。新增這些項目會授予許可，以從 AWS Glue取得目錄資訊。	2024 年 12 月 3 日
AmazonRedshiftServiceLinkedRolePolicy – 更新現有政策	`servicequotas:GetServiceQuota` 動作的許可已新增至受管政策。這會提供存取配額或限制的許可。	2024 年 3 月 8 日
AmazonRedshiftQueryEditorV2FullAccess – 更新現有政策	`redshift-serverless:ListNamespaces` 和	2024 年 2 月 21 日

變更	描述	日期
	`redshift-serverless:ListWorkgroups` 動作的許可已新增至受管政策。新增這些項目會授予許可，以在 Amazon Redshift 資料倉儲中列出無伺服器命名空間和無伺服器工作群組。	
AmazonRedshiftQueryEditorV2NoSharing – 更新現有政策	`redshift-serverless:ListNamespaces` 和 `redshift-serverless:ListWorkgroups` 動作的許可已新增至受管政策。新增這些項目會授予許可，以在 Amazon Redshift 資料倉儲中列出無伺服器命名空間和無伺服器工作群組。	2024 年 2 月 21 日
AmazonRedshiftQueryEditorV2ReadSharing – 更新現有政策	`redshift-serverless:ListNamespaces` 和 `redshift-serverless:ListWorkgroups` 動作的許可已新增至受管政策。新增這些項目會授予許可，以在 Amazon Redshift 資料倉儲中列出無伺服器命名空間和無伺服器工作群組。	2024 年 2 月 21 日

變更	描述	日期
AmazonRedshiftQueryEditorV2ReadWriteSharing – 更新現有政策	`redshift-serverless:ListNamespaces` 和 `redshift-serverless:ListWorkgroups` 動作的許可已新增至受管政策。新增這些項目會授予許可，以在 Amazon Redshift 資料倉儲中列出無伺服器命名空間和無伺服器工作群組。	2024 年 2 月 21 日
AmazonRedshiftReadOnlyAccess – 更新現有政策	`redshift:ListRecommendations` 動作的許可已新增至受管政策。這會授予許可，以列出 Amazon Redshift Advisor 建議。	2024 年 2 月 7 日
AmazonRedshiftServiceLinkedRolePolicy – 更新現有政策	`ec2:AssignIpv6Addresses` 和 `ec2:UnassignIpv6Addresses` 動作的許可已新增至受管政策。新增這兩個動作可准許指派和取消指派 IP 地址。	2023 年 10 月 31 日
AmazonRedshiftQueryEditorV2NoSharing – 更新現有政策	`sqlworkbench:GetAutocompletionMetadata` 和 `sqlworkbench:GetAutocompletionResource` 動作的許可已新增至受管政策。新增這兩個動作可准許產生和擷取資料庫資訊，以便在編輯查詢時自動完成 SQL。	2023 年 8 月 16 日

變更	描述	日期
AmazonRedshiftQueryEditorV2ReadSharing – 更新現有政策	sqlworkbench:GetAutocompletionMetadata 和 sqlworkbench:GetAutocompletionResource 動作的許可已新增至受管政策。新增這兩個動作可准許產生和擷取資料庫資訊，以便在編輯查詢時自動完成 SQL。	2023 年 8 月 16 日
AmazonRedshiftQueryEditorV2ReadWriteSharing – 更新現有政策	sqlworkbench:GetAutocompletionMetadata 和 sqlworkbench:GetAutocompletionResource 動作的許可已新增至受管政策。新增這兩個動作可准許產生和擷取資料庫資訊，以便在編輯查詢時自動完成 SQL。	2023 年 8 月 16 日

變更	描述	日期
AmazonRedshiftServiceLinkedRolePolicy – 更新現有政策	在 上 AWS Secrets Manager 建立和管理秘密的動作許可會新增至 受管政策。已新增的許可如下： • secretsmanager:GetRandomPassword • secretsmanager:DescribeSecret • secretsmanager:PutSecretValue • secretsmanager:UpdateSecret • secretsmanager:UpdateSecretVersionStage • secretsmanager:RotateSecret • secretsmanager:DeleteSecret	2023 年 8 月 14 日

變更	描述	日期
AmazonRedshiftServiceLinkedRolePolicy – 更新現有政策	Amazon EC2 上用來建立和管理安全群組與路由規則的動作許可已從受管政策中移除。這些許可與建立子網路和 VPC 有關。已移除的許可如下： • ec2:AuthorizeSecurityGroupEgress • ec2:AuthorizeSecurityGroupIngress • ec2:UpdateSecurityGroupRuleDescriptionsEgress • ec2:ReplaceRouteTableAssociation • ec2:CreateRouteTable • ec2:AttachInternetGateway • ec2:UpdateSecurityGroupRuleDescriptionsIngress • ec2:AssociateRouteTable • ec2:RevokeSecurityGroupIngress • ec2:CreateRoute • ec2:CreateSecurityGroup • ec2:RevokeSecurityGroupEgress	2023 年 5 月 8 日

變更	描述	日期
	• ec2:ModifyVpcAttribute • ec2:CreateSubnet • ec2:CreateInternetGateway • ec2:CreateVpc 這些許可與 Purpose:RedshiftMigrateToVpc 資源標籤相關聯。這個標籤將許可的適用範圍限制在 Amazon EC2 Classic 到 Amazon EC2 VPC 的遷移任務。如需資源標籤的詳細資訊，請參閱使用標籤控制 AWS 資源的存取。	
AmazonRedshiftDataFullAccess – 更新現有政策	redshift:GetClusterCredentialsWithIAM 動作的許可已新增至受管政策。新增此項目可准許取得增強型臨時憑證，讓指定的 AWS 帳戶存取 Amazon Redshift 資料庫。	2023 年 4 月 7 日
AmazonRedshiftServiceLinkedRolePolicy – 更新現有政策	Amazon EC2 上用於建立和管理安全群組規則的動作許可已新增至受管政策。這些安全群組和規則特別與 Amazon Redshift aws:RequestTag/Redshift 資源標籤相關聯。這會將許可的適用範圍限制在特定的 Amazon Redshift 資源。	2023 年 4 月 6 日

變更	描述	日期
AmazonRedshiftQueryEditorV2NoSharing – 更新現有政策	sqlworkbench:GetSchemaInference 動作的許可已新增至受管政策。新增此項目可准許取得從檔案中推斷出的資料欄和資料類型。	2023 年 3 月 21 日
AmazonRedshiftQueryEditorV2ReadSharing – 更新現有政策	sqlworkbench:GetSchemaInference 動作的許可已新增至受管政策。新增此項目可准許取得從檔案中推斷出的資料欄和資料類型。	2023 年 3 月 21 日
AmazonRedshiftQueryEditorV2ReadWriteSharing – 更新現有政策	sqlworkbench:GetSchemaInference 動作的許可已新增至受管政策。新增此項目可准許取得從檔案中推斷出的資料欄和資料類型。	2023 年 3 月 21 日
AmazonRedshiftQueryEditorV2NoSharing – 更新現有政策	sqlworkbench:AssociateNotebookWithTab 動作的許可已新增至受管政策。新增此項目可准許建立和更新連結至使用者自有筆記本的標籤。	2023 年 2 月 2 日
AmazonRedshiftQueryEditorV2ReadSharing – 更新現有政策	sqlworkbench:AssociateNotebookWithTab 動作的許可已新增至受管政策。新增此項目可准許建立和更新連結至使用者自有筆記本或連結至與其共用之筆記本的標籤。	2023 年 2 月 2 日

變更	描述	日期
AmazonRedshiftQueryEditorV2ReadWriteSharing – 更新現有政策	`sqlworkbench:AssociateNotebookWithTab` 動作的許可已新增至受管政策。新增此項目可准許建立和更新連結至使用者自有筆記本或連結至與其共用之筆記本的標籤。	2023 年 2 月 2 日

變更	描述	日期
AmazonRedshiftQueryEditorV2NoSharing – 更新現有政策	為了授予使用筆記本的許可，Amazon Redshift 已新增下列動作的許可： • sqlworkbench:ListNotebooks • sqlworkbench:CreateNotebook • sqlworkbench:DuplicateNotebook • sqlworkbench:CreateNotebookFromVersion • sqlworkbench:ImportNotebook • sqlworkbench:GetNotebook • sqlworkbench:UpdateNotebook • sqlworkbench:DeleteNotebook • sqlworkbench:CreateNotebookCell • sqlworkbench:DeleteNotebookCell • sqlworkbench:UpdateNotebookCellContent • sqlworkbench:UpdateNotebookCellLayout	2022 年 10 月 17 日

變更	描述	日期
	• sqlworkbench:BatchGetNotebookCell • sqlworkbench:ListNotebookVersions • sqlworkbench:CreateNotebookVersion • sqlworkbench:GetNotebookVersion • sqlworkbench:DeleteNotebookVersion • sqlworkbench:RestoreNotebookVersion • sqlworkbench:ExportNotebook	

變更	描述	日期
AmazonRedshiftQueryEditorV2ReadSharing – 更新現有政策	為了授予使用筆記本的許可，Amazon Redshift 已新增下列動作的許可： • sqlworkbench:ListNotebooks • sqlworkbench:CreateNotebook • sqlworkbench:DuplicateNotebook • sqlworkbench:CreateNotebookFromVersion • sqlworkbench:ImportNotebook • sqlworkbench:GetNotebook • sqlworkbench:UpdateNotebook • sqlworkbench:DeleteNotebook • sqlworkbench:CreateNotebookCell • sqlworkbench:DeleteNotebookCell • sqlworkbench:UpdateNotebookCellContent • sqlworkbench:UpdateNotebookCellLayout	2022 年 10 月 17 日

變更	描述	日期
	• sqlworkbench:BatchGetNotebookCell • sqlworkbench:ListNotebookVersions • sqlworkbench:CreateNotebookVersion • sqlworkbench:GetNotebookVersion • sqlworkbench:DeleteNotebookVersion • sqlworkbench:RestoreNotebookVersion • sqlworkbench:ExportNotebook	

變更	描述	日期
AmazonRedshiftQueryEditorV2ReadWriteSharing – 更新現有政策	為了授予使用筆記本的許可，Amazon Redshift 已新增下列動作的許可： • sqlworkbench:ListNotebooks • sqlworkbench:CreateNotebook • sqlworkbench:DuplicateNotebook • sqlworkbench:CreateNotebookFromVersion • sqlworkbench:ImportNotebook • sqlworkbench:GetNotebook • sqlworkbench:UpdateNotebook • sqlworkbench:DeleteNotebook • sqlworkbench:CreateNotebookCell • sqlworkbench:DeleteNotebookCell • sqlworkbench:UpdateNotebookCellContent • sqlworkbench:UpdateNotebookCellLayout	2022 年 10 月 17 日

變更	描述	日期
	• sqlworkbench:BatchGetNotebookCell • sqlworkbench:ListNotebookVersions • sqlworkbench:CreateNotebookVersion • sqlworkbench:GetNotebookVersion • sqlworkbench:DeleteNotebookVersion • sqlworkbench:RestoreNotebookVersion • sqlworkbench:ExportNotebook	
AmazonRedshiftServiceLinkedRolePolicy – 更新現有政策	Amazon Redshift 已新增命名空間 AWS/Redshift 以允許將指標發佈到 CloudWatch。	2022 年 9 月 7 日
AmazonRedshiftQueryEditorV2NoSharing – 更新現有政策	Amazon Redshift 已新增動作 sqlworkbench:ListQueryExecutionHistory 和 sqlworkbench:GetQueryExecutionHistory 的許可。這可准許查看查詢歷史記錄。	2022 年 8 月 30 日
AmazonRedshiftQueryEditorV2ReadSharing – 更新現有政策	Amazon Redshift 已新增動作 sqlworkbench:ListQueryExecutionHistory 和 sqlworkbench:GetQueryExecutionHistory 的許可。這可准許查看查詢歷史記錄。	2022 年 8 月 30 日

變更	描述	日期
AmazonRedshiftQueryEditorV2ReadWriteSharing – 更新現有政策	Amazon Redshift 已新增動作 `sqlworkbench:ListQueryExecutionHistory` 和 `sqlworkbench:GetQueryExecutionHistory` 的許可。這可准許查看查詢歷史記錄。	2022 年 8 月 30 日
AmazonRedshiftFullAccess – 更新現有政策	Amazon Redshift Serverless 的許可已新增至現有的 AmazonRedshiftFullAccess 受管政策。	2022 年 7 月 22 日
AmazonRedshiftDataFullAccess – 更新現有政策	Amazon Redshift 已將標籤 `aws:ResourceTag/RedshiftDataFullAccess` 許可的 `redshift-serverless:GetCredentials` 預設範圍條件從 `StringEquals` 更新為 `StringLike`，以授予標記了標籤索引鍵 `RedshiftDataFullAccess` 和任何標籤值之資源的存取權。	2022 年 7 月 11 日
AmazonRedshiftDataFullAccess – 更新現有政策	Amazon Redshift 已新增新的許可，以允許使用 `redshift-serverless:GetCredentials` 獲得 Amazon Redshift Serverless 的臨時憑證。	2022 年 7 月 8 日

變更	描述	日期
AmazonRedshiftQueryEditorV2NoSharing – 更新現有政策	Amazon Redshift 已增加動作 `sqlworkbench:GetAccountSettings` 的許可。這可准許取得帳戶設定。	2022 年 6 月 15 日
AmazonRedshiftQueryEditorV2ReadSharing – 更新現有政策	Amazon Redshift 已增加動作 `sqlworkbench:GetAccountSettings` 的許可。這可准許取得帳戶設定。	2022 年 6 月 15 日
AmazonRedshiftQueryEditorV2ReadWriteSharing – 更新現有政策	Amazon Redshift 已增加動作 `sqlworkbench:GetAccountSettings` 的許可。這可准許取得帳戶設定。	2022 年 6 月 15 日
AmazonRedshiftServiceLinkedRolePolicy – 更新現有政策	為了能夠公開存取新的 Amazon Redshift Serverless 端點，Amazon Redshift 會將彈性 IP 地址配置並關聯到客戶帳戶中 VPC 端點的彈性網路介面。其會透過服務連結角色所提供的許可來執行此操作。為了實現這個使用案例，用來配置和釋放彈性 IP 地址的動作已新增至 Amazon Redshift Serverless 服務連結角色。	2022 年 5 月 26 日
AmazonRedshiftQueryEditorV2FullAccess – 更新現有政策	動作 `sqlworkbench:ListTaggedResources` 的許可。其具體的適用範圍是 Amazon Redshift 查詢編輯器 v2 資源。此政策更新提供了只能透過查詢編輯器 v2 呼叫 `tag:GetResources` 的權利。	2022 年 2 月 22 日

變更	描述	日期
AmazonRedshiftQueryEditorV2NoSharing – 更新現有政策	動作 `sqlworkbench:ListTaggedResources` 的許可。其具體的適用範圍是 Amazon Redshift 查詢編輯器 v2 資源。此政策更新提供了只能透過查詢編輯器 v2 呼叫 `tag:GetResources` 的權利。	2022 年 2 月 22 日
AmazonRedshiftQueryEditorV2ReadSharing – 更新現有政策	動作 `sqlworkbench:ListTaggedResources` 的許可。其具體的適用範圍是 Amazon Redshift 查詢編輯器 v2 資源。此政策更新提供了只能透過查詢編輯器 v2 呼叫 `tag:GetResources` 的權利。	2022 年 2 月 22 日
AmazonRedshiftQueryEditorV2ReadWriteSharing – 更新現有政策	動作 `sqlworkbench:ListTaggedResources` 的許可。其具體的適用範圍是 Amazon Redshift 查詢編輯器 v2 資源。此政策更新提供了只能透過查詢編輯器 v2 呼叫 `tag:GetResources` 的權利。	2022 年 2 月 22 日
AmazonRedshiftQueryEditorV2ReadSharing – 更新現有政策	`sqlworkbench:AssociateQueryWithTab` 動作的許可已新增至受管政策。新增此項目讓客戶得以建立會連結至與其共用之查詢的編輯器標籤。	2022 年 2 月 22 日

變更	描述	日期
AmazonRedshiftServiceLinkedRolePolicy – 更新現有政策	Amazon Redshift 已新增新動作的許可，以允許管理 Amazon Redshift 網路和 VPC 資源。	2021 年 11 月 22 日
AmazonRedshiftAllCommandsFullAccess – 新政策	Amazon Redshift 已新增一項新政策，以允許使用從 Amazon Redshift 主控台建立的 IAM 角色，並將其設定為叢集的預設值，以便執行 COPY (從 Amazon S3)、UNLOAD、CREATE EXTERNAL SCHEMA、CREATE EXTERNAL FUNCTION、CREATE MODEL 或 CREATE LIBRARY 命令。	2021 年 11 月 18 日
AmazonRedshiftServiceLinkedRolePolicy – 更新現有政策	Amazon Redshift 已經為新動作新增了許可，以允許管理 Amazon Redshift CloudWatch 日誌群組和日誌串流，包括稽核日誌匯出。	2021 年 11 月 15 日
AmazonRedshiftFullAccess – 更新現有政策	Amazon Redshift 已新增新的許可，以允許模型可解釋性、DynamoDB、Redshift Spectrum 和 Amazon RDS 聯合。	2021 年 10 月 7 日
AmazonRedshiftQueryEditorV2FullAccess – 新政策	Amazon Redshift 已新增一項新政策，以允許 Amazon Redshift 查詢編輯器 v2 的完整存取權。	2021 年 9 月 24 日

變更	描述	日期
AmazonRedshiftQueryEditorV2NoSharing – 新政策	Amazon Redshift 已新增一項新政策，以允許在不共用資源的情況下使用 Amazon Redshift 查詢編輯器 v2。	2021 年 9 月 24 日
AmazonRedshiftQueryEditorV2ReadSharing – 新政策	Amazon Redshift 已新增一項新政策，以允許在 Amazon Redshift 查詢編輯器 v2 內進行讀取共用。	2021 年 9 月 24 日
AmazonRedshiftQueryEditorV2ReadWriteSharing – 新政策	Amazon Redshift 已新增一項新政策，以允許在 Amazon Redshift 查詢編輯器 v2 內進行讀取和更新共用。	2021 年 9 月 24 日
AmazonRedshiftFullAccess – 更新現有政策	Amazon Redshift 已新增新的許可，以允許 `sagemaker:*Job*` 。	2021 年 8 月 18 日
AmazonRedshiftDataFullAccess – 更新現有政策	Amazon Redshift 已新增新的許可，以允許 `AuthorizeDataShare` 。	2021 年 8 月 12 日
AmazonRedshiftDataFullAccess – 更新現有政策	Amazon Redshift 已新增新的許可，以允許 `BatchExecuteStatement` 。	2021 年 7 月 27 日
Amazon Redshift 已開始追蹤變更	Amazon Redshift 開始追蹤其 AWS 受管政策的變更。	2021 年 7 月 27 日

AmazonRedshiftReadOnlyAccess

授予 AWS 帳戶所有 Amazon Redshift 資源的唯讀存取權。

您可以在 IAM 主控台上找到 AmazonRedshiftReadOnlyAccess 政策，並在《AWS 受管政策參考指南》中找到 AmazonRedshiftReadOnlyAccess。

AmazonRedshiftFullAccess

授予 AWS 帳戶所有 Amazon Redshift 資源的完整存取權。此外，這個政策還會授予所有 Amazon Redshift Serverless 資源的完整存取權。

您可以在 IAM 主控台上找到 AmazonRedshiftFullAccess 政策，並在《AWS 受管政策參考指南》中找到 AmazonRedshiftFullAccess。

AmazonRedshiftQueryEditor

授予 Amazon Redshift 主控台上查詢編輯器的完整存取權。

您可以在 IAM 主控台上找到 AmazonRedshiftQueryEditor 政策，並在《AWS 受管政策參考指南》中找到 AmazonRedshiftQueryEditor。

AmazonRedshiftDataFullAccess

授予 AWS 帳戶 Amazon Redshift Data API 操作和資源的完整存取權。

您可以在 IAM 主控台上找到 AmazonRedshiftDataFullAccess 政策，並在《AWS 受管政策參考指南》中找到 AmazonRedshiftDataFullAccess。

AmazonRedshiftQueryEditorV2FullAccess

授予 Amazon Redshift 查詢編輯器 v2 操作和資源的完整存取權。此政策也會授予其他必要服務的存取權。

您可以在 IAM 主控台上找到 AmazonRedshiftQueryEditorV2FullAccess 政策，並在《AWS 受管政策參考指南》中找到 AmazonRedshiftQueryEditorV2FullAccess。

AmazonRedshiftQueryEditorV2NoSharing

授予使用 Amazon Redshift 查詢編輯器 v2 的能力，而不共用資源。此政策也會授予其他必要服務的存取權。使用此政策的主體無法標記其資源 (例如查詢)，以便與相同 AWS 帳戶中的其他主體共用。

您可以在 IAM 主控台上找到 AmazonRedshiftQueryEditorV2NoSharing 政策，並在《AWS 受管政策參考指南》中找到 AmazonRedshiftQueryEditorV2NoSharing。

AmazonRedshiftQueryEditorV2ReadSharing

授予使用 Amazon Redshift 查詢編輯器 v2 的能力，並可有限度地共用資源。此政策也會授予其他必要服務的存取權。使用此政策的主體可以標記其資源 (例如查詢)，以便與相同 AWS 帳戶中的其他主體共用。獲得授予的主體可以讀取與其團隊共用的資源，但無法更新這些資源。

您可以在 IAM 主控台上找到 AmazonRedshiftQueryEditorV2ReadSharing 政策，並在《AWS 受管政策參考指南》中找到 AmazonRedshiftQueryEditorV2ReadSharing。

AmazonRedshiftQueryEditorV2ReadWriteSharing

授予使用 Amazon Redshift 查詢編輯器 v2 的能力，並可共用資源。此政策也會授予其他必要服務的存取權。使用此政策的主體可以標記其資源 (例如查詢)，以便與相同 AWS 帳戶中的其他主體共用。獲得授予的主體可以讀取和更新與其團隊共用的資源。

您可以在 IAM 主控台上找到 AmazonRedshiftQueryEditorV2ReadWriteSharing 政策，並在《AWS 受管政策參考指南》中找到 AmazonRedshiftQueryEditorV2ReadWriteSharing。

AmazonRedshiftServiceLinkedRolePolicy

您無法將 AmazonRedshiftServiceLinkedRolePolicy 附加至您的 IAM 實體。此政策會附加至服務連結角色，可讓 Amazon Redshift 存取帳戶資源。如需詳細資訊，請參閱為 Amazon Redshift 使用服務連結角色。

您可以在 IAM 主控台上找到 AmazonRedshiftServiceLinkedRolePolicy 政策，並在《AWS 受管政策參考指南》中找到 AmazonRedshiftServiceLinkedRolePolicy。

AmazonRedshiftAllCommandsFullAccess

授予相關能力，而能夠使用從 Amazon Redshift 主控台建立的 IAM 角色，並將其設定為叢集的預設值，以便執行 COPY (從 Amazon S3)、UNLOAD、CREATE EXTERNAL SCHEMA、CREATE EXTERNAL FUNCTION 和 CREATE MODEL 命令。此政策還會授予許可，以便為相關服務 (例如 Amazon S3、CloudWatch Logs、Amazon SageMaker AI 或 AWS Glue) 執行 SELECT 陳述式。

您可以在 IAM 主控台上找到 AmazonRedshiftAllCommandsFullAccess 政策，並在《AWS 受管政策參考指南》中找到 AmazonRedshiftAllCommandsFullAccess。

AmazonRedshiftFederatedAuthorization

此政策會合併對具有 Amazon Redshift 聯合許可的 Glue Data Catalog 資料庫執行查詢所需的 IAM 動作。這類查詢會經過 AWS Glue，因此需要取得目錄物件的動作來探索物件，以及建立、更新、重新命名和刪除動作來修改物件。請注意，資源是由 Amazon Redshift 管理，因此委託人還需要 Redshift 許可才能完成查詢。 `glue:FederateAuthorization`動作可讓 AWS Glue 將目錄物件的授權決策委派給 Amazon Redshift。

此政策允許主體對具有 Amazon Redshift 聯合許可的目錄執行查詢，但不允許向 Glue 註冊和取消註冊 Amazon Redshift AWS 命名空間。請參閱 Amazon Redshift 聯合許可設定的 IAM 政策要求文件。

您可以在 IAM 主控台上找到 AmazonRedshiftFederatedAuthorization 政策，並在 AWS 受管政策參考指南中找到 AmazonRedshiftFederatedAuthorization。

您也能建立專屬的自訂 IAM 政策，以允許 Amazon Redshift API 操作與資源的相關許可。您可以將這些自訂政策附加至需要這些許可的 IAM 角色或群組。

要使用 Redshift Spectrum 所需的許可

Amazon Redshift Spectrum 需要其他 AWS 服務的許可才能存取資源。如需有關 Redshift Spectrum IAM 政策中許可的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 Amazon Redshift Spectrum 的 IAM 政策。

要使用 Amazon Redshift 主控台所需的許可

若要讓使用者使用 Amazon Redshift 主控台，該使用者必須擁有一組最低許可，允許使用者描述其 AWS 帳戶的 Amazon Redshift 資源。這些許可也必須允許使用者描述其他相關資訊，包括 Amazon EC2 安全、Amazon CloudWatch、Amazon SNS 和網路資訊。

如果您建立比最基本必要許可更嚴格的 IAM 政策，則對於採取該 IAM 政策的使用者而言，主控台就無法如預期運作。為確保這些使用者仍可使用 Amazon Redshift 主控台，也請將 `AmazonRedshiftReadOnlyAccess` 受管政策附加至使用者。若要了解如何執行這項操作，請參閱 AWS Amazon Redshift 的 受管政策。

如需有關向使用者提供 Amazon Redshift 主控台上查詢編輯器存取權的資訊，請參閱要使用 Amazon Redshift 主控台查詢編輯器所需的許可。

對於僅呼叫 AWS CLI 或 Amazon Redshift API 的使用者，您不需要允許最低主控台許可。

要使用 Amazon Redshift 主控台查詢編輯器所需的許可

使用者若要使用 Amazon Redshift 查詢編輯器，就必須擁有一組包含 Amazon Redshift 和 Amazon Redshift 資料 API 操作的最基本許可。若要使用機密連線到資料庫，您也必須擁有 Secrets Manager 許可。

若要讓使用者存取 Amazon Redshift 主控台上的查詢編輯器，請連接 `AmazonRedshiftQueryEditor`和 `AmazonRedshiftReadOnlyAccess` AWS 受管政策。`AmazonRedshiftQueryEditor` 政策會讓使用者獲得僅擷取自己 SQL 陳述式結果的許可。也就是說，由`aws:userid`相同 提交的陳述式，如 `AmazonRedshiftQueryEditor` AWS 受管政策的本節所示。

```
{
    "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}"
        }
    }
}
```

若要允許使用者擷取有相同 IAM 角色之其他人的 SQL 陳述式結果，請建立您自己的政策，但不要有用來限制目前使用者存取權的條件。同時，請限制只有管理員能獲得變更政策的存取權。

要使用查詢編輯器 v2 所需的許可

若要讓使用者使用 Amazon Redshift 查詢編輯器 v2，該使用者必須擁有 Amazon Redshift、查詢編輯器 v2 操作和其他 AWS 服務的一組最低許可 AWS Key Management Service，例如 AWS Secrets Manager和標記服務。

若要讓使用者完整存取查詢編輯器 v2，請連接 `AmazonRedshiftQueryEditorV2FullAccess` AWS 受管政策。`AmazonRedshiftQueryEditorV2FullAccess` 政策會讓使用者獲得與同一團隊中的其他成員共用查詢編輯器 v2 資源 (例如查詢) 的許可。如需查詢編輯器 v2 資源存取權控制方式的詳細資訊，請參閱 IAM 主控台中查詢編輯器 v2 特定受管政策的定義。

有些 Amazon Redshift 查詢編輯器 v2 AWS 受管政策會在條件內使用 AWS 標籤來限制對 資源的存取。在查詢編輯器 v2 內，會以附加至主體 (IAM 角色) 的 IAM 政策中的標籤索引鍵和值 `"aws:ResourceTag/sqlworkbench-team": "${aws:PrincipalTag/sqlworkbench-team}"` 為基礎來共用查詢。相同 AWS 帳戶 中具有相同標籤值 (例如 `accounting-team`) 的主體，在查詢編輯器 v2 中會位於相同團隊。您一次只能與一個團隊相關聯。具有管理許可的使用者可以透過為所有團隊成員提供相同的 `sqlworkbench-team` 標籤值，以在 IAM 主控台中設定團隊。如果 IAM 使用者或 IAM 角色的 `sqlworkbench-team` 標籤值發生變更，則可能要等一段時間，共用資源才會反映此變更。如果資源 (例如查詢) 的標籤值發生變更，則一樣可能要等一段時間才會反映此變更。團隊成員還必須擁有 `tag:GetResources` 許可才能共用。

範例：新增 IAM 角色的 **accounting-team** 標籤

1. 登入 AWS 管理主控台 ，並在 https：//https://console.aws.amazon.com/iam/ 開啟 IAM 主控台。
2. 在主控台導覽窗格中，選擇 Roles (角色)，然後選擇您要編輯的角色名稱。
3. 選擇標籤索引標籤，然後選擇新增標籤。
4. 新增標籤索引鍵 sqlworkbench-team 和值 `accounting-team`。
5. 選擇儲存變更。

   現在，當 IAM 主體 (附加了此 IAM 角色) 與團隊共用查詢時，具有相同 `accounting-team` 標籤值的其他主體都可以檢視該查詢。

如需如何將標籤附加到主體 (包括 IAM 角色和 IAM 使用者) 的相關資訊，請參閱《IAM 使用者指南》中的標記 IAM 資源。

您也可以使用身分提供者 (IdP) 在工作階段層級設定團隊。這可讓使用相同 IAM 角色的多個使用者擁有不同的團隊。IAM 角色的信任政策必須允許 `sts:TagSession` 操作。如需詳細資訊，請參閱《IAM 使用者指南》中的新增工作階段標籤所需的許可。將主體標籤屬性新增至 IdP 提供的 SAML 聲明。

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:sqlworkbench-
team">
    <AttributeValue>accounting-team</AttributeValue>
</Attribute>
```

遵循身分提供者 (IdP) 的指示，將來自目錄的內容填入到 SAML 屬性。如需身分提供者 (IdP) 和 Amazon Redshift 的相關資訊，請參閱《IAM 使用者指南》中的使用 IAM 身分驗證產生資料庫使用者憑證和身分提供者和聯合。

`sqlworkbench:CreateNotebookVersion` 可准許在帳戶上取得筆記本儲存格的目前內容並建立筆記本版本。這意味著，在建立版本時，筆記本的目前內容會與版本的內容相同。之後，當目前的筆記本更新時，版本中的儲存格內容會保持不變。`sqlworkbench:GetNotebookVersion` 可准許取得筆記本的版本。在筆記本上沒有 `sqlworkbench:BatchGetNotebookCell` 許可但有 `sqlworkbench:CreateNotebookVersion` 和 `sqlworkbench:GetNotebookVersion` 許可的使用者可以存取版本中的筆記本儲存格。這位沒有 `sqlworkbench:BatchGetNotebookCell` 許可的使用者仍然可以透過先建立版本，然後取得所建立的這個版本，以擷取筆記本儲存格的內容。

要使用 Amazon Redshift 排程器所需的許可

使用 Amazon Redshift 排程器時，您要設定與 Amazon Redshift 排程器 (**scheduler.redshift.amazonaws.com**) 有信任關係的 IAM 角色，讓排程器代您取得許可。您也要將政策 (許可) 附加到您想排程之 Amazon Redshift API 操作的角色。

以下範例示範 JSON 格式的政策文件，以設定與 Amazon Redshift 排程器和 Amazon Redshift 的信任關係。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "scheduler.redshift.amazonaws.com",
                    "redshift.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

如需信任實體的詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可給 AWS 服務。

您也必須新增要排程之 Amazon Redshift 操作的許可。

為使用 `ResizeCluster` 操作的排程器，在您的 IAM 政策中新增類似以下的許可。是否需要加強政策限制，視環境而定。

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "redshift:ResizeCluster",
            "Resource": "*"
        }
    ]
}
```

如需為 Amazon Redshift 排程器建立角色的步驟，請參閱《IAM 使用者指南》中的為 AWS 服務建立角色（主控台）。在 IAM 主控台中建立角色時，選擇以下項目：

- 針對 Choose the service that will use this role (選擇將使用此角色的服務)：選擇 Redshift。
- 針對 Select your use case (選取您的使用案例)：選擇 Redshift - Scheduler (Redshift - 排程器)。
- 建立政策，或將政策附加到允許排程 Amazon Redshift 操作的角色。選擇 Create policy (建立政策) 或修改角色以連接政策。輸入要排程之操作的 JSON 政策。
- 建立角色之後，編輯 IAM 角色的 Trust Relationship (信任關係) 以納入服務 `redshift.amazonaws.com`。

您建立的 IAM 角色有信任的 `scheduler.redshift.amazonaws.com` 和 `redshift.amazonaws.com` 實體。且附加了允許受支援 Amazon Redshift API 動作的政策，例如 `"redshift:ResizeCluster"`。

要使用 Amazon EventBridge 排程器所需的許可

使用 Amazon EventBridge 排程器時，您要設定與 EventBridge 排程器 (**`events.amazonaws.com`**) 有信任關係的 IAM 角色，讓排程器代您取得許可。您也要將政策 (許可) 附加到您想排程之 Amazon Redshift 資料 API 操作的角色，以及附加 Amazon EventBridge 操作的政策。

當您使用主控台上的 Amazon Redshift 查詢編輯器建立已排程的查詢時，您會使用 EventBridge 排程器。

您可以建立 IAM 角色，以便在 IAM 主控台上執行已排程的查詢。在此 IAM 角色中，請附加 `AmazonEventBridgeFullAccess` 和 `AmazonRedshiftDataFullAccess`。

以下範例示範 JSON 格式的政策文件，以設定與 EventBridge 排程器的信任關係。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "events.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

如需信任實體的詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可給 AWS 服務。

如需為 EventBridge 排程器建立角色的步驟，請參閱《IAM 使用者指南》中的為 AWS 服務建立角色（主控台）。在 IAM 主控台中建立角色時，選擇以下項目：

- 針對選擇將使用此角色的服務：選擇 CloudWatch 事件。
- 針對選取您的使用案例：選擇 CloudWatch 事件。
- 附加以下許可政策：AmazonEventBridgeFullAccess 和 AmazonRedshiftDataFullAccess。

您建立的 IAM 角色有信任的 events.amazonaws.com 實體。且附加了允許受支援 Amazon Redshift 資料 API 動作的政策，例如 "redshift-data:*"。

要使用 Amazon Redshift 機器學習 (ML) 所需的許可

接下來，您可以找到要針對不同使用案例使用 Amazon Redshift 機器學習 (ML) 所需之許可的描述。

若要讓使用者能夠搭配 Amazon SageMaker AI 使用 Amazon Redshift ML，請使用比預設政策更嚴格的政策來建立 IAM 角色。您可以使用下列政策。您也可以修改此政策以因應您的需求。

下列政策顯示要使用 Amazon Redshift 的模型可解釋性來執行 SageMaker AI Autopilot 所需的許可。

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"
                    ]
                }
            }
        }
    ]
}
```

下列政策顯示可允許存取 Amazon DynamoDB、Redshift Spectrum 和 Amazon RDS 聯合的完整最基本許可。

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，若要使用 AWS KMS 金鑰進行加密，請將下列許可新增至政策。

```
{
    "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>"
    ]
}
```

若要允許 Amazon Redshift 和 SageMaker AI 擔任先前與其他服務互動的 IAM 角色，請將下列信任政策新增至該角色。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
```

```
            "redshift.amazonaws.com",
            "sagemaker.amazonaws.com",
            "forecast.amazonaws.com"
          ]
        },
        "Action": "sts:AssumeRole"
      }
    ]
}
```

在上文中，Amazon S3 儲存貯體 `redshift-downloads/redshift-ml/` 是用於其他步驟和範例之範例資料的儲存所在位置。如果您不需要從 Amazon S3 載入資料，則可以移除此儲存貯體。或者，也可以將其取代為您用來將資料載入 Amazon Redshift 的其他 Amazon S3 儲存貯體。

your-account-id、**your-role** 和 **your-s3-bucket** 值是您在 CREATE MODEL 命令中所指定的帳戶 ID、角色和儲存貯體。

或者，如果您指定要與 Amazon Redshift ML 搭配使用的 AWS KMS 金鑰，則可以使用範例政策的 AWS KMS 金鑰區段。**your-kms-key** 值是您用來作為 CREATE MODEL 命令一部分的金鑰。

當您為 hyperparameter 調校任務指定私有虛擬私有雲端 (VPC) 時，請新增下列許可。

```
{
            "Effect": "Allow",
            "Action": [
            "ec2:CreateNetworkInterface",
            "ec2:CreateNetworkInterfacePermission",
            "ec2:DeleteNetworkInterface",
            "ec2:DeleteNetworkInterfacePermission",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeVpcs",
            "ec2:DescribeDhcpOptions",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
            ]
}
```

若要使用模型解釋，請確定您有呼叫 SageMaker AI API 操作的許可。建議您使用 `AmazonSageMakerFullAccess` 受管政策。如果您想要建立具有更嚴格政策的 IAM 角色，請使用以下政策。

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": "*"
    }
  ]
}
```

如需 AmazonSageMakerFullAccess 受管政策的相關資訊，請參閱《Amazon SageMaker AI 開發人員指南》中的 AmazonSageMakerFullAccess。

如果您想要建立 Forecast 模型，建議您使用 AmazonForecastFullAccess 受管政策。如果您想要使用更嚴格的政策，請新增以下政策至 IAM 角色。

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": "*"
        }
    ]
}
```

如果您想要建立 Amazon Bedrock 模型，建議您使用 `AmazonBedrockFullAccess` 受管政策。如果您想要使用更嚴格的政策，請新增以下政策至 IAM 角色。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "bedrock:InvokeModel",
            "Resource": [
                "*",
                "arn:aws:bedrock:us-east-1::foundation-model/*"
            ]
        }
    ]
}
```

如需 Amazon Redshift ML 的詳細資訊，請參閱在 Amazon Redshift 中使用機器學習、CREATE MODEL 或 CREATE EXTERNAL MODEL。

串流擷取的許可

串流擷取可與兩個服務搭配運作。這兩個服務是 Kinesis Data Streams 和 Amazon MSK。

要搭配 Kinesis Data Streams 使用串流擷取所需的許可

您可以在開始從 Amazon Kinesis Data Streams 擷取串流中找到包含受管政策範例的程序。

要搭配 Amazon MSK 使用串流擷取所需的許可

您可以在開始從 Amazon Managed Streaming for Apache Kafka 擷取串流中找到包含受管政策範例的程序。

要使用資料共用 API 操作所需的許可

若要控制資料共用 API 操作的存取權，請使用 IAM 動作型政策。如需有關如何管理 IAM 政策的資訊，請參閱《IAM 使用者指南》中的管理 IAM 政策。

具體而言，假設生產者叢集管理員需要使用 `AuthorizeDataShare` 呼叫來為 AWS 帳戶外部的資料共用授權輸出。在此情況下，您需要設定 IAM 動作型政策以授予此許可。使用 `DeauthorizeDataShare` 呼叫以撤銷輸出。

使用 IAM 動作型政策時，您也可以在政策中指定 IAM 資源，例如 `DataShareARN`。以下內容會顯示 `DataShareARN` 的格式和範例。

```
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
```

您可以在 IAM 政策中指定資料共用名稱，以限制特定資料共用的 `AuthorizeDataShare` 存取權。

```
{
  "Statement": [
    {
      "Action": [
        "redshift:AuthorizeDataShare",
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-
e2e24359e9a8/SalesShare"
```

```
      ],
      "Effect": "Deny"
    }
  ]
}
```

您也可以將 IAM 政策的範圍限制在特定生產者叢集擁有的所有資料共用。若要這麼做，請使用萬用字元或星號來取代政策中的 **datashare-name** 值。保留叢集的 namespace-guid 值。

```
arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-e2e24359e9a8/
*
```

以下是可防止實體呼叫特定生產者叢集所擁有資料共用上 AuthorizeDataShare 的 IAM 政策。

```
{
  "Statement": [
    {
      "Action": [
        "redshift:AuthorizeDataShare",
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:555555555555:datashare:86b5169f-01dc-4a6f-9fbb-
e2e24359e9a8/*"
      ],
      "Effect": "Deny"
    }
  ]
}
```

DataShareARN 會根據資料共用名稱和擁有端叢集命名空間的全域唯一 ID (GUID) 來限制存取權。其會透過將名稱指定為星號來進行此操作。

GetClusterCredentials 的資源政策

若要使用具有 IAM 資料庫憑證的 JDBC 或 ODBC 連線連線到叢集資料庫，或以程式設計方式呼叫 GetClusterCredentials動作，您需要許可來呼叫可存取 dbuser 資源redshift:GetClusterCredentials的動作。

如果您使用 JDBC 或 ODBC 連線，而不是 server 和 port，您可以指定 cluster_id 和 region，但要這麼做，您的政策必須允許具有 redshift:DescribeClusters 資源存取的 cluster 動作。

如果您使用選用參數 Autocreate、DbGroups 和 DbName 來呼叫 GetClusterCredentials，請務必也允許動作並允許下表所列資源的存取權。

GetClusterCredentials 參數	Action	資源
Autocreate	redshift:CreateClusterUser	dbuser
DbGroups	redshift:JoinGroup	dbgroup
DbName	NA	dbname

如需資源的相關資訊，請參閱Amazon Redshift 資源和操作。

您也可以在政策包含下列條件：

- redshift:DurationSeconds
- redshift:DbName
- redshift:DbUser

> **Important**
>
> 對於 SAML SSO 整合，您可能需要使用 ${redshift:DbUser}變數指定 IAM 政策。在這些情況下，我們強烈建議使用條件陳述式，以確保發起人無法取得不符合其 AWS 使用者 ID 的使用者登入資料。例如 "StringEquals": {"aws:userid":"AIDIODR4TAW7CSEXAMPLE:${redshift:DbUser}"}"。請參閱 範例 8：使用 GetClusterCredentials 的 IAM 政策。如需條件的詳細資訊，請參閱在政策中指定條件

客戶管理政策範例

在本節中，您可以找到授予各種 Amazon Redshift 動作之許可的使用者政策範例。當您使用 Amazon Redshift API、 AWS SDKs或 時，這些政策會運作 AWS CLI。

> **Note**
>
> 所有範例皆使用美國西部 (奧勒岡) 區域 (us-west-2) 及虛構帳戶 ID。

範例 1：允許使用者對所有 Amazon Redshift 動作和資源的完整存取權

下列政策允許存取處理所有資源的所有 Amazon Redshift 動作。

JSON

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

`Action` 元素中的值 `redshift:*` 指出 Amazon Redshift 中的所有動作。

範例 2：拒絕使用者存取一組 Amazon Redshift 動作

預設會拒絕所有的許可。不過，有時您必須明確拒絕對特定動作或動作集的存取權。下列政策允許對所有 Amazon Redshift 動作的存取權，並明確拒絕對名稱開頭為 `Delete` 之任何 Amazon Redshift 動作的存取權。此政策適用於 `us-west-2` 中的所有 Amazon Redshift 資源。

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:*"
   }
  ]
}
```

範例 3：允許使用者管理叢集

下列政策允許使用者建立、刪除、修改所有叢集和將它們重新開機，然後拒絕刪除叢集名稱開頭為 `protected` 之任何叢集的許可。

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"
    }
  ]
}
```

範例 4：允許使用者授權和撤銷快照存取

下列政策允許使用者 (例如使用者 A) 來執行下列動作：

- 授予對透過名為 shared 之叢集建立的任何快照的存取權。
- 撤銷對透過 shared 叢集建立、快照名稱開頭為 revokable 之任何快照的快照存取權。

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"
    }
  ]
}
```

如果使用者 A 已允許使用者 B 存取快照，使用者 B 必須具備以下這類的政策以允許使用者 B 從快照還原叢集。下列政策允許使用者 B 從快照描述和還原，以及建立叢集。這些叢集名稱的開頭必須是 `from-other-account`。

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"
    }
  ]
}
```

範例 5：允許使用者複製叢集快照和從快照還原叢集

下列政策允許使用者複製透過名為 big-cluster-1 之叢集建立的任何快照，以及還原快照名稱開頭為 snapshot-for-restore 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"
    }
  ]
}
```

範例 6：允許使用者存取 Amazon Redshift，以及相關 AWS 服務的常見動作和資源

以下範例政策可讓您存取 Amazon Redshift、Amazon Simple Notification Service (Amazon SNS) 和 Amazon CloudWatch 的所有動作和資源。它也會允許在該帳戶下於所有相關 Amazon EC2 資源上的指定動作。

> **Note**
>
> 此範例政策中指定的 Amazon EC2 動作不支援資源層級許可。

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": [
        "*"
      ]
    }
  ]
}
```

範例 7：允許使用者使用 Amazon Redshift 主控台標記資源

以下範例政策可讓使用者使用 AWS Resource Groups透過 Amazon Redshift 主控台標記資源。此政策可附加至會叫用新的或原始 Amazon Redshift 主控台的使用者角色。如需標記的相關資訊，請參閱在 Amazon Redshift 中標記資源。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid":"TaggingPermissions",
            "Effect": "Allow",
            "Action": [
                "redshift:DeleteTags",
                "redshift:CreateTags",
                "redshift:DescribeTags",
                "tag:UntagResources",
                "tag:TagResources"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

範例 8：使用 GetClusterCredentials 的 IAM 政策

以下範例政策使用這些範例參數值：

- 區域: us-west-2
- AWS 帳戶： 123456789012
- 叢集名稱: examplecluster

以下策略啟用 GetCredentials、CreateClusterUser 和 JoinGroup 操作。政策使用條件索引鍵，只在 AWS 使用者 ID 符合 時，才允許 GetClusterCredentials和 CreateClusterUser動作"AIDIODR4TAW7CSEXAMPLE:${redshift:DbUser}@yourdomain.com"。僅會對 "testdb" 資料庫請求 IAM 存取權。此政策也允許使用者加入名為 "common_group" 的群組。

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"
        ]
      }
    ]
}
```

下面的範例顯示允許 IAM 角色呼叫 `GetClusterCredentials` 操作的政策。指定 Amazon Redshift `dbuser` 資源會將角色存取授予名為 `examplecluster` 的叢集上的資料庫使用者名稱 `temp_creds_user`。

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": {
    "Effect": "Allow",
    "Action": "redshift:GetClusterCredentials",
    "Resource": "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/
temp_creds_user"
  }
}
```

您可以使用萬用字元 (*) 來取代全部或一部分的叢集名稱、使用者名稱和資料庫群組名稱。以下範例允許指定帳戶中任何叢集任何開頭為 temp_ 的使用者名稱。

> **Important**
>
> 下列範例中的陳述式指定萬用字元 (*) 做為資源的部分值，因此政策允許任何開頭為指定字元的資源。在 IAM 政策中使用萬用字元可能會授予過多許可，根據最佳實務，建議使用適用於您商業應用程式最嚴苛的政策。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "redshift:GetClusterCredentials",
    "Resource": "arn:aws:redshift:us-west-2:123456789012:dbuser:*/temp_*"
  }
}
```

以下範例顯示允許 IAM 角色呼叫 GetClusterCredentials 操作的政策，且選項將自動建立新使用者並指定使用者於登入時加入的群組。"Resource": "*" 子句授予角色任何資源的存取權，包括叢集、資料庫使用者或使用者群組。

> **Important**
>
> 下列範例中的 陳述式指定萬用字元 (*) 做為指定動作的資源，以便政策允許存取任何叢集和資料庫使用者，並允許建立任何使用者。在 IAM 政策中使用萬用字元可能會授予過多許可，根據最佳實務，建議使用適用於您商業應用程式最嚴苛的政策。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
             "redshift:GetClusterCredentials",
             "redshift:CreateClusterUser",
  "redshift:JoinGroup"
            ],
    "Resource": "*"
  }
}
```

如需詳細資訊，請參閱 Amazon Redshift ARN 語法。

Amazon Redshift 的原生身分提供者 (IdP) 聯合

使用原生身分提供者聯合可以更輕鬆地管理 Amazon Redshift 的身分和許可，因為它可以運用您現有的身分提供者來簡化身分驗證和管理許可。它會透過將身分中繼資料從您的身分提供者共用至 Redshift 來達成此目的。對於此功能的第一個迭代，支援的身分提供者是 Microsoft Azure Active Directory (Azure AD)。

若要設定 Amazon Redshift 以便能夠從第三方身分提供者進行身分驗證，請向 Amazon Redshift 註冊身分提供者。這樣做可讓 Redshift 驗證身分提供者所定義的使用者和角色。因此，您就不必在第三方身分提供者和 Amazon Redshift 中執行精細的身分管理，因為身分資訊會共用。

如需有關使用從身分提供者 (IdP) 群組傳輸而來之工作階段角色的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 PG_GET_SESSION_ROLES。

原生身分提供者 (IdP) 聯合

若要完成身分提供者和 Amazon Redshift 之間的初步設定，請執行幾個步驟：首先，向身分提供者將 Amazon Redshift 註冊為第三方應用程式，以請求必要的 API 許可。然後，在身分提供者中建立使用者和群組。最後，使用會設定身分提供者專屬身分驗證參數的 SQL 陳述式向 Amazon Redshift 註冊身分提供者。在向 Redshift 註冊身分提供者時，指派命名空間以確保使用者和角色會正確分組。

透過向 Amazon Redshift 註冊的身分提供者，可以在 Redshift 與身分提供者之間設定通訊。然後，用戶端可以傳遞字符並以身分提供者實體的身分向 Redshift 進行驗證。Amazon Redshift 會使用 IdP 群組成員資格資訊以映射至 Redshift 角色。如果使用者先前不存在於 Redshift 中，系統會建立該使用者。系統會建立映射至身分提供者群組的角色 (如果它們不存在)。Amazon Redshift 管理員在角色上授予許可，使用者可以執行查詢和執行其他資料庫任務。

下列步驟概述使用者登入時，原生身分提供者聯合的運作方式：

1. 當使用者從用戶端使用原生 IdP 選項登入時，身分提供者字符會從用戶端傳送至驅動程式。
2. 使用者會進行身分驗證。如果該使用者還未存在於 Amazon Redshift 中，系統便會建立新使用者。Redshift 會將使用者的身分提供者群組映射至 Redshift 角色。
3. 系統會根據使用者的 Redshift 角色指派許可。這些許可會由管理員授予給使用者和角色。
4. 使用者可以查詢 Redshift。

桌面用戶端工具

如需如何使用原生身分提供者聯合透過 Power BI 連線到 Amazon Redshift 的指示，請參閱部落格文章整合 Amazon Redshift 原生 IdP 聯合與 Microsoft Azure Active Directory (AD) 和 Power BI。該文章中描述了使用 Azure AD 的 Amazon Redshift 原生 IdP 設定的逐步實作方式。其內容會詳細說明用來為 Power BI 桌面版或 Power BI 服務設定用戶端連線的步驟。這些步驟包括註冊應用程式、設定許可和設定憑證。

若要了解如何使用 Power BI 桌面版和 JDBC 用戶端 SQL Workbench/J 來整合 Amazon Redshift 原生 IdP 聯合，請觀看以下影片：

如需如何使用原生身分提供者聯合透過 SQL 用戶端 (具體來說是 DBeaver 或 SQL Workbench/J) 連線至 Amazon Redshift 的指示，請參閱部落格文章使用 SQL 用戶端將 Amazon Redshift 原生 IdP 聯合與 Microsoft Azure AD 進行整合。

限制

這些限制均適用：

- Amazon Redshift 驅動器支援下列版本以後的 `BrowserIdcAuthPlugin`：
  - Amazon Redshift JDBC 驅動器 v2.1.0.30
  - Amazon Redshift ODBC 驅動器 v2.1.3
  - Amazon Redshift Python 驅動器 v2.1.3
- Amazon Redshift 驅動器支援下列版本以後的 `IdpTokenAuthPlugin`：
  - Amazon Redshift JDBC 驅動器 v2.1.0.19
  - Amazon Redshift ODBC 驅動器 v2.0.0.9
  - Amazon Redshift Python 驅動器 v2.0.914
- 不支援增強型 VPC 路由 – 當您使用原生身分提供者聯合時，不支援增強型 VPC 路由。如需增強型 VPC 路由的詳細資訊，請參閱 Amazon Redshift 中的增強型 VPC 路由。
- AWS IAM Identity Center 快取 – AWS IAM Identity Center 快取工作階段資訊。當您嘗試透過 Redshift 查詢編輯器 v2 連線至 Redshift 資料庫時，這可能會導致無法預測的存取問題。這是因為查詢編輯器 v2 AWS 中的相關聯 IAM Identity Center 工作階段仍然有效，即使資料庫使用者從 AWS 主控台登出也是如此。快取會在一小時後過期，通常這樣就會修復所有問題。

在 Amazon Redshift 上設定身分提供者

本節說明用來設定身分提供者和 Amazon Redshift 以便為原生身分提供者聯合建立通訊的步驟。您需要身分提供者的作用中帳戶。在設定 Amazon Redshift 之前，您必須向身分提供者將 Redshift 註冊為應用程式，並授予管理員同意。

在 Amazon Redshift 中完成以下步驟：

1. 執行 SQL 陳述式以註冊識別提供者，包括 Azure 應用程式中繼資料的描述。若要在 Amazon Redshift 中建立身分提供者，請在取代 issuer、client_id、client_secret 和 audience 參數值後執行下列命令。這些參數是 Microsoft Azure AD 的特有參數。請將身分提供者名稱取代為您選擇的名稱，並以唯一名稱取代命名空間，以包含您身分提供者目錄中的使用者和角色。

```
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"]
}'
```

azure 類型表示提供者具體來說有利於與 Microsoft Azure AD 進行通訊。這是目前唯一支援的第三方身分提供者。

- issuer — 收到字符時要信任的發行者 ID。發行者會附加 tenant_id 的唯一識別碼。
- client_id — 向身分提供者註冊之應用程式的唯一公共識別碼。這可以稱為應用程式 ID。
- client_secret — 只有身分提供者和已註冊的應用程式才知道的機密識別碼或密碼。
- audience — 指派給 Azure 中應用程式的應用程式 ID。

您可以在建立身分提供者時設定參數以指定憑證、私有金鑰和私有金鑰密碼，而不是使用共用用戶端機密。

```
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"}';
```

私有金鑰密碼 client_pk_password 是選擇性的。

2. 選擇性：在 Amazon Redshift 中執行 SQL 命令以預先建立使用者和角色。這有助於事先授予許可。Amazon Redshift 中的角色名稱如下所示：<Namespace>:<GroupName on Azure AD>。例如，當您在 Microsoft Azure AD 中建立名為 rsgroup 的群組和名為 aad 的命名空間時，角色名稱會是 aad:rsgroup。Amazon Redshift 中的使用者和角色名稱，這些名稱是根據身分提供者命名空間中的這些使用者和群組成員資格定義的。

   角色和使用者的映射包括驗證其 external_id 值，以確保它是最新的。外部 ID 會映射至身分提供者中群組或使用者的識別碼。例如，角色的外部 ID 會映射至對應的 Azure AD 群組 ID。同樣地，每個使用者的外部 ID 會映射至身分提供者中的 ID。

   ```
   create role "aad:rsgroup";
   ```

3. 根據您的要求向角色授予相關許可。例如：

```
GRANT SELECT on all tables in schema public to role "aad:rsgroup";
```

4. 您也可以向特定使用者授予許可。

```
GRANT SELECT on table foo to aad:alice@example.com
```

請注意，聯合的外部使用者的角色成員資格只能在該使用者的工作階段中使用。這會對建立資料庫物件造成影響。例如，當聯合的外部使用者建立任何檢視或預存程序時，同一使用者便無法將這些物件的許可委派給其他使用者和角色。

命名空間的解釋

命名空間會將使用者或角色映射至特定身分提供者。例如，在 IAM AWS 中建立的使用者字首為 `iam:`。此前置詞可防止使用者名稱發生衝突，並可實現對於多個身分存放區的支援。如果使用 aad 命名空間註冊之身分來源的使用者 alice@example.com 登入，則系統會在 Redshift 中建立使用者 `aad:alice@example.com` (如果該使用者尚未存在)。請注意，使用者和角色命名空間的功能與 Amazon Redshift 叢集命名空間 (這是與叢集相關聯的唯一識別碼) 的功能不同。

自動為身分提供者建立 Amazon Redshift 角色

此功能可讓您根據身分提供者 (IdP) 的群組成員資格，在 Redshift 中自動建立角色。自動建立角色支援與原生 IdP 整合的 Azure Active Directory。

自動建立角色有幾個優點。當您自動建立角色時，Redshift 建立的角色會具有 IdP 中的群組成員資格，如此就能避免繁瑣的手動角色建立和維護工作。您也可以選擇篩選哪些群組要對應至 Redshift 角色。

運作方式

當您以 IdP 使用者身分登入 Redshift 時，下列事件會按順序發生：

1. Redshift 從 IdP 擷取您的群組成員資格。
2. Redshift 自動建立對應到這些群組的角色，且角色的格式為 *`idp_namespace`*`:`*`rolename`*。
3. Redshift 對您授予對應角色的許可。

每次使用者登入時，都會自動建立使用者所屬但不存在目錄內的每個群組。您可以選擇性地設定包含和排除篩選條件，以控制哪些 IdP 群組要建立 Redshift 角色。

設定自動建立角色

使用 CREATE IDENTITY PROVIDER 和 ALTER IDENTITY PROVIDER 命令來啟用並設定自動角色建立。

```
-- 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;
```

篩選群組

您可以選擇使用 INCLUDE 和 EXCLUDE 模式，篩選哪些 IdP 群組要對應至 Redshift 角色。當模式發生衝突時，EXCLUDE 優先於 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%';
```

範例

下列範例示範如何在不進行篩選的情況下開啟自動建立角色。

```
CREATE IDENTITY PROVIDER prod_idc TYPE azure ...
  AUTO_CREATE_ROLES TRUE;
```

下列範例包含開發群組並排除測試群組。

```
ALTER IDENTITY PROVIDER prod_idc
  AUTO_CREATE_ROLES TRUE
  INCLUDE GROUPS LIKE '%dev%'
  EXCLUDE GROUPS LIKE '%test%';
```

最佳實務

當您啟用角色的自動建立功能時，請考慮下列最佳實務：

- 使用 INCLUDE 和 EXCLUDE 篩選條件來控制哪些群組會取得角色。
- 定期稽核角色並清除未使用的角色。
- 利用 Redshift 角色階層來簡化許可管理。

將 Redshift 與 AWS IAM Identity Center 連線，以獲得單一登入體驗

您可以透過信任身分傳播來管理使用者和群組對 Amazon Redshift 資料倉儲的存取。

信任的身分傳播是一項 AWS IAM Identity Center 功能，連線的管理員 AWS 服務 可以使用此功能來授予和稽核服務資料的存取權。存取此資料是根據使用者屬性，例如群組關聯。設定信任的身分傳播需要連線的管理員 AWS 服務 與 IAM Identity Center 管理員之間的協同合作。如需詳細資訊，請參閱先決條件和考量。

若要說明一個end-to-end案例，您可以使用 Amazon Quick 儀表板或 Amazon Redshift 查詢編輯器 v2 來存取 Redshift。在這種情況下，存取權是以 AWS IAM Identity Center 群組為基礎。Redshift 可以判斷使用者是誰及其群組成員資格。 AWS IAM Identity Center 也可以透過 Okta 或 PingOne 等第三方身分提供者 (IdP) 來連接和管理身分。

在管理員設定 Redshift 和 IAM Identity Center AWS 之間的連線後，他們可以根據身分提供者群組設定精細存取，以授權使用者存取資料。

> **Important**
>
> 當您從 IAM Identity Center AWS 或連線身分提供者 (IdP) 目錄刪除使用者時，使用者不會自動從 Amazon Redshift 目錄中刪除。若要從 Amazon Redshift 目錄手動刪除使用者，請執行 DROP USER命令以完全刪除從 IAM Identity Center AWS 或 IdP 移除的使用者。如需如何捨棄使用者的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 DROP USER。

Redshift 與 IAM Identity Center AWS 整合的優勢

搭配 Redshift 使用 AWS IAM Identity Center 可透過下列方式讓您的組織受益：

- 中的儀表板作者 Amazon Quick 可以連線到 Redshift 資料來源，而無需重新輸入密碼或要求管理員設定具有複雜許可的 IAM 角色。
- AWS IAM Identity Center 為您的人力資源使用者提供集中位置 AWS。您可以直接在 AWS IAM Identity Center 中建立使用者和群組，或連接您在標準型身分提供者中管理的現有使用者和群組，例如 Okta、PingOne 或 Microsoft Entra ID (Azure AD)。 AWS IAM Identity Center 會將身分驗證導向使用者和群組的所選事實來源，並維護使用者和群組的目錄，以供 Redshift 存取。如需詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的管理身分來源和支援的身分提供者。
- 您可以使用簡單的自動探索和連線功能，與多個 Redshift 叢集和工作群組共用一個 AWS IAM Identity Center 執行個體。這可讓您快速新增叢集，而無需額外努力為每個叢集設定 AWS IAM Identity Center 連線，並確保所有叢集和工作群組都有一致的使用者、其屬性和群組檢視。請注意，組織的 AWS IAM Identity Center 執行個體必須與您連線的任何 Redshift 資料共用位於相同的區域。
- 由於使用者身分是已知且會隨著資料存取而被記錄，因此您可以更輕鬆地透過稽核中的使用者存取權來符合法規 AWS CloudTrail。

連接應用程式的管理員角色

以下是將分析應用程式連線至適用於 Redshift 的 AWS IAM Identity Center 受管應用程式的關鍵角色：

- 應用程式管理員 — 建立應用程式，並設定應用程式將啟用哪些服務來交換身分字符。此管理員也會指定哪些使用者或群組可存取應用程式。
- 資料管理員 — 設定對資料的精細存取權限。IAM Identity Center AWS 中的使用者和群組可以映射到特定許可。

透過 使用 IAM Identity Center 連線至 AWS Amazon Redshift Amazon Quick

以下說明如何使用 Quick 在連線至 Redshift 時向 Redshift 進行身分驗證，並透過 IAM Identity Center AWS 管理存取：授權從 Quick 到 Amazon Redshift 叢集的連線。這些步驟也適用於 Amazon Redshift Serverless。

透過 Amazon Redshift 查詢編輯器 v2 使用 AWS IAM Identity Center 連線至 Amazon Redshift

完成設定 IAM Identity Center AWS 與 Redshift 連線的步驟後，使用者可以透過以 AWS IAM Identity Center 為基礎的命名空間字首身分，存取資料庫中的資料庫和適當的物件。如需有關使用查詢編輯器 v2 登入連線至 Redshift 資料庫的詳細資訊，請參閱 使用查詢編輯器 v2 查詢資料庫。

跨多個 使用 AWS IAM Identity Center AWS 區域

Amazon Redshift AWS 支援多個 IAM Identity Center AWS 區域。您可以將 AWS IAM Identity Center 從主要區域擴展 AWS 區域 到其他區域，透過接近使用者和可靠性來改善效能。在 IAM Identity Center AWS 中新增新區域時，您可以在新區域中建立 Redshift IAM Identity Center 應用程式，而無需從主要區域複寫身分。您可以在新的區域中使用 AWS IAM Identity Center 設定 Amazon Redshift 聯合許可，您可以在其中啟用資料列層級、資料欄層級和遮罩控制項。如需在多個區域中開始使用 AWS IAM Identity Center 的詳細資訊，請參閱《AWS IAM Identity Center 使用者指南》中的管理多個 中的 AWS 區域 AWS IAM Identity Center。

設定與 AWS Amazon Redshift 的 IAM Identity Center 整合

您的 Amazon Redshift 叢集管理員或 Amazon Redshift Serverless 管理員必須執行數個步驟，將 Redshift 設定為已啟用 AWS IAM Identity Center 的應用程式。這可讓 Redshift 自動探索並連線至 AWS IAM Identity Center，以接收登入和使用者目錄服務。之後，當 Redshift 管理員建立叢集或工作群組時，他們可以讓新的資料倉儲使用 AWS IAM Identity Center 來管理資料庫存取。

將 Redshift 啟用為 AWS IAM Identity Center 受管應用程式的重點是，您可以控制來自 IAM Identity Center AWS 內部的使用者和群組許可，或來自與其整合的第三方身分提供者。當您的資料庫使用者登入 Redshift 資料庫時，例如分析師或資料科學家，它會在 IAM Identity Center AWS 中檢查其群組，並將這些群組與 Redshift 中的角色名稱相符。以這種方式定義 Redshift 資料庫角色名稱的群組，可以存取一組用於銷售分析的資料表。下列各節展示如何設定此項。

先決條件

以下是整合 AWS IAM Identity Center 與 Amazon Redshift 的先決條件：

- 帳戶組態 – 如果您計劃有跨帳戶使用案例，或如果您在不同具有相同 AWS AWS IAM Identity Center 執行個體的帳戶中使用 Redshift 叢集，則必須在 AWS 組織的管理帳戶中設定 IAM Identity Center。這包括設定您的身分識別來源。如需詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的入門、員工身分識別和支援的身分提供者。您必須確定已在 IAM Identity Center AWS 中建立使用者或群組，或從身分來源同步使用者和群組，才能將他們指派給 Redshift 中的資料。

> **Note**
>
> 您可以選擇使用 IAM Identity Center AWS 的帳戶執行個體，前提是 Redshift 和 AWS IAM Identity Center 位於同一個帳戶中。當您建立和設定 Redshift 叢集或工作群組時，可以使用小工具建立此執行個體。

- 設定受信任的字符發行者 — 在某些情況下，您可能需要使用受信任的字符簽發者，這是一個可以發出和驗證信任字符的實體。設定 IAM Identity Center AWS 整合的 Redshift 管理員可以選取信任的權杖發行者並新增必要的屬性以完成組態之前，必須先執行初步步驟。這可能包括將外部身分提供者設定為可信任的字符發行者，並在 IAM Identity Center AWS 主控台中新增其屬性。若要完成這些步驟，請參閱使用具有受信任權杖頒發者的應用程式。

  > **Note**
  >
  > 並非所有外部連線都需要設定受信任的字符發行者。使用 Amazon Redshift 查詢編輯器 v2 連線到 Redshift 資料庫，並不需要受信任的字符發行者組態。但可以套用於第三方應用程式，例如透過您的身分提供者進行驗證的儀表板或自訂應用程式。

- 設定 IAM 角色 — 必須設定後續提及權限的區段。您必須依據 IAM 最佳實務新增許可權。下列程序會詳細說明特定權限。
- 增強型 VPC 路由的網路組態 – 如果開啟增強型 VPC 路由，您的 VPC 必須能夠連接 AWS IAM Identity Center 服務。介面 VPC 端點是提供此連線的建議方式。

  > **Note**
  >
  > 在使用者登入之前設定此連線。如需搭配 IAM Identity Center 使用增強型 VPC AWS 路由的詳細資訊，請參閱 搭配增強型 VPC AWS 路由使用 IAM Identity Center 身分驗證。

如需詳細資訊，請參閱 AWS IAM Identity Center 入門。

設定您的身分提供者以使用 AWS IAM Identity Center

控制使用者和群組身分管理的第一步是連線到 AWS IAM Identity Center，並設定您的身分提供者。您可以使用 AWS IAM Identity Center 本身做為身分提供者，也可以連接第三方身分存放區，例如 Okta。如需有關設定連線和設定身分提供者的詳細資訊，請參閱《AWS IAM Identity Center 使用者指南》中的連線至外部身分提供者。在此程序結束時，請確定您已將一小部分使用者和群組新增至 AWS IAM Identity Center，以供測試之用。

管理許可

Redshift/AWS IAM Identity Center 應用程式生命週期管理所需的許可

您必須建立 IAM 身分，讓 Redshift 管理員用來設定 Redshift 以搭配 IAM Identity Center AWS 使用。通常您會建立具有許可的 IAM 角色，並視需要將其指派給其他身分。此角色必須具有列出的許可，才能執行下列動作。

建立 Redshift/AWS IAM Identity Center 應用程式

- `sso:PutApplicationAssignmentConfiguration`— 用於安全性。
- `sso:CreateApplication` – 用來建立 AWS IAM Identity Center 應用程式。
- `sso:PutApplicationAuthenticationMethod`— 授予 Redshift 驗證存取權。
- `sso:PutApplicationGrant`— 用於變更受信任的字符發行者資訊。
- `sso:PutApplicationAccessScope` – 適用於 Redshift AWS IAM Identity Center 應用程式設定。這包括適用於 Amazon S3 Access Grants 的 AWS Lake Formation 和 。
- `redshift:CreateRedshiftIdcApplication` – 用來建立 Redshift AWS IAM Identity Center 應用程式。

描述 Redshift/AWS IAM Identity Center 應用程式

- `sso:GetApplicationGrant`— 用於列出受信任的字符發行者資訊。
- `sso:ListApplicationAccessScopes` - 讓 Redshift AWS IAM Identity Center 應用程式設定列出下游整合，例如針對 AWS Lake Formation 和 S3 存取授權。
- `redshift:DescribeRedshiftIdcApplications` – 用來描述現有的 AWS IAM Identity Center 應用程式。

變更 Redshift/AWS IAM Identity Center 應用程式

- `redshift:ModifyRedshiftIdcApplication`— 用於變更現有的 Redshift 應用程式。
- `sso:UpdateApplication` – 用來更新 AWS IAM Identity Center 應用程式。
- `sso:GetApplicationGrant` - 取得信任權杖頒發者資訊。
- `sso:ListApplicationAccessScopes` - 用於 Redshift AWS IAM Identity Center 應用程式設定。
- `sso:DeleteApplicationGrant` – 刪除信任權杖發行者資訊。
- `sso:PutApplicationGrant`— 用於變更受信任的字符發行者資訊。

- sso:PutApplicationAccessScope – 適用於 Redshift AWS IAM Identity Center 應用程式設定。這包括適用於 Amazon S3 Access Grants 的 AWS Lake Formation 和 。
- sso:DeleteApplicationAccessScope - 用於刪除 Redshift AWS IAM Identity Center 應用程式設定。這包括適用於 Amazon S3 Access Grants 的 AWS Lake Formation 和 。

刪除 Redshift/AWS IAM Identity Center 應用程式

- sso:DeleteApplication – 用來刪除 AWS IAM Identity Center 應用程式。
- redshift:DeleteRedshiftIdcApplication – 提供刪除現有 Redshift AWS IAM Identity Center 應用程式的能力。

Redshift/查詢編輯器 v2 應用程式生命週期管理所需的許可

您必須建立 IAM 身分，讓 Redshift 管理員用來設定 Redshift 以搭配 IAM Identity Center AWS 使用。通常您會建立具有許可的 IAM 角色，並視需要將其指派給其他身分。此角色必須具有列出的許可，才能執行下列動作。

建立查詢編輯器 v2 應用程式

- redshift:CreateQev2IdcApplication - 用於建立 QEV2 應用程式。
- sso:CreateApplication - 提供建立 AWS IAM Identity Center 應用程式的功能。
- sso:PutApplicationAuthenticationMethod— 授予 Redshift 驗證存取權。
- sso:PutApplicationGrant— 用於變更受信任的字符發行者資訊。
- sso:PutApplicationAccessScope – 適用於 Redshift AWS IAM Identity Center 應用程式設定。這包括查詢編輯器第 2 版。
- sso:PutApplicationAssignmentConfiguration— 用於安全性。

說明查詢編輯器 v2 應用程式

- redshift:DescribeQev2IdcApplications – 用來描述 AWS IAM Identity Center QEV2 應用程式。

變更查詢編輯器 v2 應用程式

- redshift:ModifyQev2IdcApplication - 用於變更 AWS IAM Identity Center QEV2 應用程式。

- `sso:UpdateApplication` – 用於變更 AWS IAM Identity Center QEV2 應用程式。

刪除查詢編輯器 v2 應用程式

- `redshift:DeleteQev2IdcApplication` - 用於刪除 QEV2 應用程式。
- `sso:DeleteApplication` - 用於刪除 QEV2 應用程式。

> **Note**
>
> 在 Amazon Redshift SDK 中，無法使用下列 API：
>
> - CreateQev2IdcApplication
> - DescribeQev2IdcApplications
> - ModifyQev2IdcApplication
> - DeleteQev2IdcApplication
>
> 這些動作專用於在 AWS 主控台中執行與 Redshift QEV2 AWS 的 IAM Identity Center 整合。如需詳細資訊，請參閱 Amazon Redshift 定義的動作。

資料庫管理員在主控台中連線新資源所需的許可

在建立過程中，需要這些權限才能連接新的佈建叢集或 Amazon Redshift Serverless 工作群組。如果您具有這些許可，主控台中就會顯示選項，供您選擇連線至適用於 Redshift 的 AWS IAM Identity Center 受管應用程式。

- `redshift:DescribeRedshiftIdcApplications`
- `sso:ListApplicationAccessScopes`
- `sso:GetApplicationAccessScope`
- `sso:GetApplicationGrant`

我們建議的最佳實務是，將許可政策附加到 IAM 角色，然後根據需要將其指派給使用者和群組。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。

使用 IAM Identity Center 將 Redshift AWS 設定為 AWS 受管應用程式

AWS 在 IAM Identity Center 可以管理 Amazon Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組的身分之前，Redshift 管理員必須完成讓 Redshift 成為 IAM Identity Center AWS 受管應用程式的步驟：

1. 在 Amazon Redshift 或 Amazon Redshift Serverless 主控台功能表中選取 AWS IAM Identity Center 整合，然後選取連線至 AWS IAM Identity Center。您可以從該處逐步執行一系列選擇，以填入 AWS IAM Identity Center 整合的屬性。
2. 為 Redshift AWS 的 IAM Identity Center 受管應用程式選擇顯示名稱和唯一名稱。
3. 指定組織的命名空間。這通常是組織名稱的縮寫版。它會新增為 Redshift 資料庫中 AWS IAM Identity Center 受管使用者和角色的字首。
4. 選取要使用的 IAM 角色。此 IAM 角色應與用於 Redshift 的其他角色分開，建議您不要將其用於其他用途。需要的特定政策許可如下所示：

   - `sso:DescribeApplication`— 在目錄中建立身分提供者 (IdP) 項目時需要。
   - `sso:DescribeInstance`— 用於手動建立 IdP 聯合角色或使用者。
5. 設定用戶端連線和受信任的字符發行者。藉由設定與外部身分提供者的關係，設定受信任的字符發行者有助於信任的身分傳播。身分傳播可讓使用者登入一個應用程式的同時，存取另一個應用程式中的特定資料。這可讓使用者更順暢地從不同位置收集資料。在此步驟中，您可以在主控台為每個受信任的字符發行者設定屬性。這些屬性包括名稱和受眾聲明 (或 aud claim)，您可能必須從工具或服務的組態屬性中取得這些聲明。您可能還需要提供第三方工具的 JSON Web 字符 (JWT)應用程序名稱。

   > **Note**
   >
   > 根據字符類型，每個第三方工具或服務的 `aud claim` 需求可能會有所不同，字符類型可以是身分提供者發行的存取字符或另一種類型 (例如 ID 字符)。每個廠商可以有所不同。當您實作信任的身分傳播並與 Redshift 整合時，需要為第三方工具傳送給 AWS的字符類型提供正確的 aud 值。查看您的工具或服務廠商的建議。

   如需有關可信身分傳播的詳細資訊，請參閱《AWS IAM Identity Center 使用者指南》中的可信身分傳播概觀。

Redshift 管理員完成這些步驟並儲存組態後， AWS IAM Identity Center 屬性就會出現在 Redshift 主控台中。您也可以查詢系統檢視 SVV_IDENTITY_PROVIDERS，以驗證應用程式的屬性。其中包括應用

程式名稱和命名空間。您可以使用命名空間做為與應用程式相關聯之 Redshift 資料庫物件的字首。完成這些任務可讓 Redshift 成為已啟用 AWS IAM Identity Center 的應用程式。主控台內的屬性包括整合狀態。當整合完成時，會顯示已啟用。完成此程序後，就可以在每個新叢集上啟用 AWS IAM Identity Center 整合。

設定之後，您可以選擇使用者或群組索引標籤，然後選擇指派，在 Redshift AWS 中包含來自 IAM Identity Center 的使用者和群組。

為新的 AWS Amazon Redshift 叢集或 Amazon Redshift Serverless 工作群組啟用 IAM Identity Center 整合

您的資料庫管理員會設定新的 Redshift 資源，以便與 IAM Identity Center AWS 保持一致，讓登入和資料存取更容易。這是建立已佈建叢集或 Serverless 工作群組的步驟的一部分來執行。具有建立 Redshift 資源許可的任何人都可以執行這些 AWS IAM Identity Center 整合任務。當您建立佈建叢集時，您可以從 Amazon Redshift 主控台中選擇建立叢集開始。以下步驟說明如何為資料庫啟用 AWS IAM Identity Center 管理。(不包括建立叢集的所有步驟。)

1. 在建立叢集步驟中的 IAM Identity Center 整合區段中選擇啟用 <您的叢集名稱>。
2. 啟用整合時，處理程序依照步驟。您可以在主控台選擇啟用 IAM Identity Center 整合以執行此操作。
3. 對於新的叢集或工作群組，請使用 SQL 命令在 Redshift 中建立資料庫角色。以下是命令：

   ```
   CREATE ROLE <idcnamespace:rolename>;
   ```

   命名空間和角色名稱如下所示：

   - IAM Identity Center 命名空間字首 – 這是您在設定 IAM Identity Center 和 Redshift AWS 之間的連線時定義的命名空間。
   - 角色名稱 – 此 Redshift 資料庫角色必須符合 IAM Identity Center AWS 中的群組名稱。

   Redshift 會與 AWS IAM Identity Center 連線，並擷取建立資料庫角色並將其對應至 IAM Identity Center AWS 群組所需的資訊。

請注意，建立新的資料倉儲時，針對 AWS IAM Identity Center 整合指定的 IAM 角色會自動附加到佈建叢集或 Amazon Redshift Serverless 工作群組。完成輸入所需的叢集中繼資料並建立資源後，您可以在 屬性中檢查 AWS IAM Identity Center 整合的狀態。如果您在 AWS IAM Identity Center 中的群組名稱有空格，則必須在建立相符角色時使用 SQL 中的引號。

啟用 Redshift 資料庫並建立角色之後，您就可以使用 Amazon Redshift 查詢編輯器 v2 或 Amazon Quick連線到資料庫。接下來章節中會進一步提供詳細資訊。

使用 API 設定預設值 **RedshiftIdcApplication**

設定作業由您的身分管理員執行。使用 API，您可以建立並填入 RedshiftIdcApplication，代表 IAM Identity Center AWS 中的 Redshift 應用程式。

1. 若要開始，您可以建立使用者，並將其新增至 IAM Identity Center AWS 中的群組。您可以在 IAM Identity Center AWS 的 AWS 主控台中執行此操作。
2. 呼叫 create-redshift-idc-application建立 AWS IAM Identity Center 應用程式，使其與 Redshift 用量相容。您可以填入必要的值來建立應用程式。顯示名稱是指要在 AWS IAM Identity Center 儀表板上顯示的名稱。IAM 角色 ARN 是具有 AWS IAM Identity Center 許可的 ARN，也可由 Redshift 擔任。

```
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'
```

下列範例顯示從呼叫 create-redshift-idc-application 傳回的範例 RedshiftIdcApplication 回應。

```
"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` 將特定群組或個別使用者指派給 IAM Identity Center AWS 中的受管應用程式。透過這樣做，您可以指定要透過 IAM Identity Center AWS 管理的群組。如果資料庫管理員在 Redshift 中建立資料庫角色，IAM Identity Center AWS 中的群組名稱會對應至 Redshift 中的角色名稱。那些角色可以控制資料庫中的權限。如需詳細資訊，請參閱 IAM Identity Center AWS 主控台中的指派使用者存取應用程式。
4. 啟用應用程式後，從 IAM Identity Center 呼叫`create-cluster`並包含 Redshift AWS 受管應用程式 ARN。這樣做會將叢集與 IAM Identity Center AWS 中的受管應用程式建立關聯。

將 AWS IAM Identity Center 應用程式與現有叢集或工作群組建立關聯

如果您要為現有的叢集或工作群組啟用 AWS IAM Identity Center 整合，可藉由執行 SQL 命令來進行此操作。您也可以執行 SQL 命令來變更整合的設定。如需詳細資訊，請參閱 ALTER IDENTITY PROVIDER。

您也可以捨棄現有的身分提供者。以下範例顯示 CASCADE 如何刪除附加至身分提供者的使用者和角色。

```
DROP IDENTITY PROVIDER
<provider_name> [ CASCADE ]
```

設定使用者權限

管理員會根據使用者的身分屬性和群組成員資格，在其身分提供者或 IAM Identity Center AWS 中直接設定各種資源的許可。例如，身分提供者管理員可以將資料庫工程師新增至適合其角色的群組。此群組名稱會對應至 Redshift 資料庫角色名稱。此角色可提供或限制 Redshift 中特定資料表或視觀表的存取權。

搭配增強型 VPC AWS 路由使用 IAM Identity Center 身分驗證

開啟增強型 VPC 路由時，Redshift 會透過您的虛擬私有雲端 (VPC) 路由流量，而不是透過網際網路路由流量。這包括 Redshift 用來驗證 IAM Identity Center AWS 使用者的呼叫。無論使用者以互動方式登入，或是連線到透過受信任身分傳播取得的 AWS IAM Identity Center 權杖，這都會適用。若要讓這些連線成功，Redshift 必須能夠從您的 VPC AWS 連線到 IAM Identity Center。介面 VPC 端點是提供此連線的建議方式。它們可讓您的佈建叢集或 Amazon Redshift Serverless 工作群組透過 AWS 網路到達 AWS IAM Identity Center。

> **Important**
>
> 這些要求適用於透過 AWS IAM Identity Center 驗證的任何連線。IAM 使用者和本機資料庫使用者透過不同的機制進行身分驗證，不受影響。如果已關閉增強型 VPC 路由，則不需要這些端點。

> **Note**
>
> 變更增強型 VPC 路由會重新啟動佈建叢集。如需開啟增強型 VPC 路由的詳細資訊，請參閱開啟增強型 VPC 路由。

介面 VPC 端點

在叢集或工作群組所在的 VPC 中建立下列界面 VPC 端點 (AWS PrivateLink)。兩個端點都是必要的。如果端點遺失或無法連線， AWS IAM Identity Center 登入會失敗。

`com.amazonaws.region.sso-oauth`

驗證使用者在登入時提供的 AWS IAM Identity Center 存取權杖，並將其交換為您的 Redshift 工作階段範圍的權杖。

`com.amazonaws.region.identitystore`

從 AWS IAM Identity Center 身分存放區解析使用者及其 IAM Identity Center AWS 群組成員資格。

如需有關建立介面 VPC 端點的資訊，請參閱《Amazon VPC 使用者指南》中的建立 VPC 端點。

建立端點之前，請確認已針對叢集或工作群組執行的 VPC 開啟 `DNS hostnames`和 `DNS resolution` 屬性。私有 DNS 名稱取決於兩個屬性。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的 VPC 的 DNS 屬性。

增強型 VPC 路由要求您的佈建叢集或 Redshift Serverless 工作群組不可公開存取。

當您建立每個端點時，請執行下列動作：

- 開啟私有 DNS 名稱 – Redshift 會使用其標準 DNS 名稱連線至這些服務。如果沒有私有 DNS 名稱，這些名稱會繼續解析為公有地址，而且呼叫不會路由到您的端點。

- 在 AWS IAM Identity Center 區域中建立端點 – AWS IAM Identity Center 是一種區域服務。在 IAM Identity Center AWS 執行個體執行的區域中建立這些端點。如果您的叢集或工作群組在不同區域中執行，而且您不使用 AWS IAM Identity Center 多區域複寫，請使用跨區域端點選項，讓端點到達 IAM Identity Center AWS 區域。
- 選擇相同可用區域中的子網路 – 選取叢集或工作群組使用的可用區域中的子網路。
- 允許傳入 HTTPS – 連接到每個端點的安全群組必須允許 TCP 連接埠 443 上來自叢集或工作群組子網路的傳入流量。
- 保留預設端點政策 – 預設政策允許完整存取。如果您的組織需要限制性端點政策，請依據委託人而非動作清單來設定範圍。然後驗證登入是否仍然成功。

使用 AWS 網路防火牆

AWS Network Firewall 是選用的。如果您使用 AWS Network Firewall 篩選傳出流量，而不是使用介面 VPC 端點，防火牆政策必須允許 AWS IAM Identity Center 服務網域。否則，當 Redshift 無法連線時，登入會失敗。將下列網域新增至防火牆的允許清單，將*##*取代為您的 IAM Identity Center AWS 區域：

`oidc.`*`region`*`.amazonaws.com`

驗證 AWS IAM Identity Center 存取權杖，並將其交換為您的 Redshift 工作階段範圍的權杖。

`.sso.`*`region`*`.amazonaws.com`

連接 AWS IAM Identity Center 服務以解析 AWS IAM Identity Center 應用程式及其指派。

`identitystore.`*`region`*`.amazonaws.com`

解析使用者及其 AWS IAM Identity Center 群組成員資格。

如果您的防火牆使用deny-by-default政策，它還必須允許 Redshift 在增強型 VPC 路由下用於一般操作的網域，例如 Amazon S3，以及如果您查詢資料湖，則為 AWS Glue。如需詳細資訊，請參閱 Amazon Redshift 中的增強型 VPC 路由。

使用介面 VPC 端點，如中所述介面 VPC 端點， 會將此流量保留在 AWS 網路上，且不需要防火牆允許清單。

自動建立 的 Amazon Redshift 角色 AWS IAM Identity Center

此功能是與 的整合 AWS IAM Identity Center ，可讓您根據群組成員資格在 Redshift 中自動建立角色。

自動建立角色有幾個優點。當您自動建立角色時，Redshift 建立的角色會具有 IdP 中的群組成員資格，如此就能避免繁瑣的手動角色建立和維護工作。您也可以選擇使用包含和排除模式，以篩選哪些群組要對應至 Redshift 角色。

運作方式

當您以 IdP 使用者身分登入 Redshift 時，下列事件會按順序發生：

1. Redshift 從 IdP 擷取您的群組成員資格。
2. Redshift 自動建立對應到這些群組的角色，且角色的格式為 *idp_namespace*:*rolename*。
3. Redshift 對您授予對應角色的許可。

每次使用者登入時，都會自動建立使用者所屬但不存在目錄內的每個群組。您可以選擇性地設定包含和排除篩選條件，以控制哪些 IdP 群組要建立 Redshift 角色。

設定自動建立角色

使用 CREATE IDENTITY PROVIDER 和 ALTER IDENTITY PROVIDER 命令來啟用並設定自動角色建立。

```
-- 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;
```

篩選群組

您可以選擇使用 INCLUDE 和 EXCLUDE 模式，篩選哪些 IdP 群組要對應至 Redshift 角色。當模式發生衝突時，EXCLUDE 優先於 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%';
```

範例

下列範例示範如何在不進行篩選的情況下開啟自動建立角色。

```
CREATE IDENTITY PROVIDER prod_idc TYPE AWSIDC  ...
  AUTO_CREATE_ROLES TRUE;
```

下列範例包含開發群組並排除測試群組。

```
ALTER IDENTITY PROVIDER prod_idc
  AUTO_CREATE_ROLES TRUE
  INCLUDE GROUPS LIKE '%dev%'
  EXCLUDE GROUPS LIKE '%test%';
```

最佳實務

當您啟用角色的自動建立功能時，請考慮下列最佳實務：

- 使用 INCLUDE 和 EXCLUDE 篩選條件來控制哪些群組會取得角色。
- 定期稽核角色並清除未使用的角色。
- 利用 Redshift 角色階層來簡化許可管理。

Amazon Redshift 與 Amazon S3 存取授權整合

使用與 Amazon S3 存取授權的整合，即可順利地傳播您的 IAM Identity Center 身分，以控制對 Amazon S3 資料的存取權。此整合可讓您根據 IAM Identity Center 使用者和群組授權 Amazon S3 資料存取。

如需有關 Amazon S3 存取授權的詳細資訊，請參閱使用 S3 存取授權管理存取權。

使用 Amazon S3 存取授權可為您的應用程式帶來以下優點：

- 根據 IAM Identity Center 身分精細控制 Amazon S3 資料的存取權。
- 在 Amazon Redshift 和 Amazon S3 之間集中管理 IAM Identity Center 身分。
- 您可以避免針對 Amazon S3 存取管理個別 IAM 許可。

運作方式

若要將您的應用程式與 Amazon S3 存取授權整合，請執行下列操作：

- 首先，您將 Amazon Redshift 設定為使用 AWS 管理主控台 或 與 Amazon S3 Access Grants 整合 AWS CLI。
- 接著，具有 IdC 管理員權限的使用者會使用 Amazon S3 存取授權服務，對特定 IdC 使用者/群組授予 Amazon S3 儲存貯體或字首存取權。如需詳細資訊，請參閱在 S3 存取授權中使用授權。
- 當通過 Redshift 身分驗證的 IdC 使用者執行存取 S3 的查詢時 (例如 COPY、UNLOAD 或 Spectrum 操作)，Amazon Redshift 會從 Amazon S3 存取授權服務擷取範圍限定於該 IdC 身分的臨時 S3 存取憑證。
- 然後 Amazon Redshift 會使用擷取的臨時憑證來存取該查詢的授權 Amazon S3 位置。

設定與 Amazon S3 存取授權的整合

若要設定 Amazon Redshift 與 Amazon S3 存取授權的整合，請執行下列操作：

主題

- 使用 設定與 Amazon S3 Access Grants 的整合 AWS 管理主控台
- 使用 啟用與 Amazon S3 Access Grants 的整合 AWS CLI

使用 設定與 Amazon S3 Access Grants 的整合 AWS 管理主控台

1. 開啟 Amazon Redshift 主控台。
2. 從叢集窗格中選擇您的叢集。
3. 在叢集詳細資訊頁面的身分提供者整合區段中，啟用與 S3 存取授權服務的整合。

> **Note**
>
> 如果您未設定 IAM Identity Center，則不會顯示身分提供者整合區段。如需詳細資訊，請參閱啟用 AWS IAM Identity Center。

使用 啟用與 Amazon S3 Access Grants 的整合 AWS CLI

1. 若要建立新的 Amazon Redshift IdC 應用程式同時啟用 S3 整合，請執行下列操作：

```
aws redshift create-redshift-idc-application <other parameters>
  --service-integrations '[ {"S3AccessGrants": [{"ReadWriteAccess":
 {"Authorization": "Enabled"}}]} ]'
```

2. 若要修改現有的應用程式以啟用 S3 存取授權整合，請執行下列操作：

```
aws redshift modify-redshift-idc-application <other parameters>
  --service-integrations '[ {"S3AccessGrants": [{"ReadWriteAccess":
 {"Authorization": "Enabled"}}]} ]'
```

3. 若要修改現有的應用程式以停用 S3 存取授權整合，請執行下列操作：

```
aws redshift modify-redshift-idc-application <other parameters>
  --service-integrations '[ {"S3AccessGrants": [{"ReadWriteAccess":
 {"Authorization": "Disabled"}}]} ]'
```

使用 S3 存取授權整合

設定 S3 存取授權整合之後，存取 S3 資料的查詢 (例如 COPY、UNLOAD 或 Spectrum 查詢) 會使用 IdC 身分進行授權。未使用 IdC 驗證身分的使用者也可以執行這些查詢，但這些使用者帳戶不會利用 IdC 提供的集中式管理。

下列範例示範使用 S3 存取授權整合執行的查詢：

```
COPY table FROM 's3://mybucket/data';  // -- Redshift uses IdC identity
UNLOAD ('SELECT * FROM table') TO 's3://mybucket/unloaded/'    // -- Redshift uses IdC
 identity
```

透過 查詢資料 AWS Lake Formation

使用 AWS Lake Formation 可讓您更輕鬆地集中管理和保護資料湖，並提供資料存取。透過 IAM Identity Center AWS 將身分傳播設定為 Lake Formation，Redshift 可讓管理員根據組織的身分提供者 (IdP) 群組，允許精細存取 Amazon S3 資料湖。這些群組是透過 AWS IAM Identity Center 進行管理。本節說明如何設定幾個使用案例、從資料湖進行查詢，以及從資料共用進行查詢，以示範如何使用 AWS IAM Identity Center 搭配 Redshift 連線至 Lake Formation 管理的資源。

使用 AWS IAM Identity Center 和 Redshift 連線查詢資料湖

這些步驟涵蓋的使用案例，您使用連接至 Redshift AWS 的 IAM Identity Center 來查詢由 Lake Formation 管理的資料湖。

先決條件

此程序有幾個先決條件步驟：

1. AWS 必須設定 IAM Identity Center 以支援 Redshift 的身分驗證和身分管理。您可以從主控台啟用 AWS IAM Identity Center，然後選取身分提供者 (IdP) 來源。之後，請將一組 IdP 使用者與 IAM Identity Center AWS 同步。您也必須依照本文件先前詳述的步驟，設定 AWS IAM Identity Center 和 Redshift 之間的連線。
2. 在組態步驟中建立新的 Amazon Redshift 叢集，並透過 AWS IAM Identity Center 啟用身分管理。
3. 為 Lake Formation 建立受管 AWS IAM Identity Center 應用程式並進行設定。接下來設定 IAM Identity Center AWS 和 Redshift 之間的連線。步驟如下：
   a. 在 中 AWS CLI，使用 `modify-redshift-idc-application`命令來啟用 Lake Formation 服務與 Redshift 的 AWS IAM Identity Center 受管應用程式整合。此呼叫包含 `service-integrations` 參數，該參數設定為啟用 Lake Formation 授權的組態字串值。
   b. 使用 `create-lake-formation-identity-center-configuration` 命令設定 Lake Formation。這會為 Lake Formation AWS 建立 IAM Identity Center 應用程式，該應用程式會顯示在 AWS IAM Identity Center 入口網站中。管理員必須設定 `--cli-input-json` 引數，其值是所有 CLI API AWS 呼叫使用標準格式的 JSON 檔案路徑。您可以包含下列各值：
      - `CatalogId`— Lake Formation 目錄 ID。
      - `InstanceArn` – AWS IAM Identity Center 執行個體 ARN 值。

管理員完成必要條件組態之後，資料庫管理員可以建立外部結構描述以查詢資料湖。

1. 管理員建立外部結構描述 — Redshift 資料庫管理員使用下列 SQL 陳述式連線到資料庫，並建立外部結構描述：

```
CREATE EXTERNAL SCHEMA if not exists my_external_schema from DATA CATALOG database
 'my_lf_integrated_db' catalog_id '12345678901234';
```

   請注意，在此情況下不需要指定 IAM 角色，因為存取是透過 IAM Identity Center AWS 管理。
2. 管理員授予許可 – 管理員授予 IAM Identity Center AWS 群組的使用量，該群組授予 Redshift 資源的許可。這是透過執行 SQL 陳述式來完成，如下所示：

```
GRANT USAGE ON SCHEMA "my_external_schema" to "MYCO:sales";
```

然後，管理員會根據組織的需求，使用 CLI 授予物件的 Lake Formation AWS 許可：

```
aws lakeformation grant-permissions ...
```

3. 使用者執行查詢 - 此時，屬於銷售群組 (僅為示意) 的 AWS IAM Identity Center 使用者可以透過查詢編輯器 v2 登入 Redshift 資料庫。然後，他們可以執行存取外部結構描述中資料表的查詢，如以下範例所示：

```
SELECT * from my_external_schema.table1;
```

使用 AWS IAM Identity Center 和 Redshift 連線連線至資料共用

透過 IAM Identity Center 管理存取時，您可以從不同的 Redshift AWS 資料倉儲存取資料共用。若要這麼做，您可以執行查詢來設定外部資料庫。在完成這些步驟之前，假設您已在 Redshift 和 IAM Identity Center AWS 之間設定連線，而且已建立 AWS Lake Formation 應用程式，如先前程序中詳述。

1. 建立外部資料庫 — 管理員建立外部資料庫以進行資料共用，並透過其 ARN 參考該資料庫。以下是展示如何執行此操作的範例：

```
CREATE DATABASE "redshift_external_db" FROM ARN 'arn:aws:glue:us-
east-1:123456789012:database/redshift_external_db-iad' WITH NO DATA CATALOG SCHEMA;
```

在此使用案例中，如果您使用 AWS IAM Identity Center 搭配 Redshift 進行身分管理，則不包含 IAM 角色。

2. 管理員設定許可：建立資料庫後，管理員會將用量授予 IAM Identity Center AWS 群組。這會授予 Redshift 資源的權限：

```
GRANT USAGE ON DATABASE "my_external_db" to "MYCO:sales";
```

系統管理員還會使用 AWS CLI 授予物件的 Lake Formation 權限：

```
aws lakeformation grant-permissions ...
```

3. 使用者執行查詢 — 來自銷售群的使用者可以根據指派的權限，查詢資料庫中的資料表：

```
select * from redshift_external_db.public.employees;
```

如需有關授予資料湖權限以及授予資料共用權限的詳細資訊，請參閱將權限授予使用者和群組。如需將使用權授予結構描述或資料庫的詳細資訊，請參閱 GRANT。

使用信任的字符發行者將您的應用程式或工具與 OAuth 整合

您可以將功能新增至您建立的用戶端工具，以透過 IAM Identity Center AWS 連線連線至 Redshift。如果您已設定 Redshift 與 AWS IAM Identity Center 整合，請使用本節中詳述的屬性來設定連線。

使用 IAM Identity Center 連線至 Redshift AWS 的身分驗證外掛程式

您可以使用下列驅動程式外掛程式，使用 AWS IAM Identity Center 連線至 Amazon Redshift：

- `BrowserIdcAuthPlugin` – 此外掛程式有助於與 IAM Identity Center AWS 進行無縫single-sign-on整合。它會建立瀏覽器視窗，讓使用者使用其公司身分提供者中定義的使用者憑證登入。
- `IdpTokenAuthPlugin` – 想要自行管理身分驗證流程的應用程式應該使用此外掛程式，而不是讓 Amazon Redshift 驅動程式開啟瀏覽器視窗以進行 AWS IAM Identity Center 身分驗證。它接受來自與 AWS IAM Identity Center 連線之任何 Web 身分提供者的 IAM Identity Center 付費存取字符或 OpenID Connect (OIDC) JSON Web 字符 AWS (JWT)，例如 Okta、PingOne 和 Microsoft Entra ID (Azure AD)。用戶端應用程式負責產生此必要的存取權杖/JWT。

使用 **`BrowserIdcAuthPlugin`** 進行身分驗證

根據您的 Amazon Redshift 驅動器，使用以下外掛程式名稱來使用 `BrowserIdcAuthPlugin` 進行連線。

驅動程式	連線選項關鍵字	Value	備註
JDBC	`plugin_name`	com.amazon.redshift.plugin.BrowserIdcAuthPlugin	連線時，您必須輸入外掛程式的完整類別名稱。
ODBC	`plugin_name`	BrowserIdcAuthPlugin	
Python	`credentials_provider`	BrowserIdcAuthPlugin	Python 驅動器沒有可用的 `plugin_na`

驅動程式	連線選項關鍵字	Value	備註
			me 選項。請改用 credentials_provider 。

BrowserIdcAuthPlugin 外掛程式有以下其他連線選項：

選項名稱	是否為必要？	說明	範例
idc_region	必要	IAM Identity Center AWS 區域 AWS 執行個體所在的 。	us-east-1
issuer_url	必要	AWS IAM Identity Center 伺服器的執行個體端點。您可以使用 IAM Identity Center AWS 主控台找到此值。	https://identitycenter.amazonaws.com/ssoins-g5j2k70sn4yc5nsc
listen_port	選用	Amazon Redshift 驅動程式用來透過瀏覽器重新導向從 AWS IAM Identity Center 接收auth_code 回應的連接埠。	7890
idc_client_display_name	選用	在 AWS IAM Identity Center 的單一登入同意快顯視窗中， AWS IAM Identity Center 用戶端用於應用程式的名稱。	Amazon Redshift 驅動器
idp_response_timeout	選用	Redshift 驅動器等待身分驗證流程完成的	60

選項名稱	是否為必要？	說明	範例
		時間長度，以秒為單位。	

您必須在您建立並用來連線之工具的連線屬性中輸入這些值。如需詳細資訊，請參閱各驅動器的連線選項文件：

- JDBC 驅動器 2.x 版組態的選項
- ODBC 驅動器選項
- Amazon Redshift Python 連接器的組態選項

使用 **IdpTokenAuthPlugin** 進行身分驗證

根據您的 Amazon Redshift 驅動器，使用以下外掛程式名稱來使用 IdpTokenAuthPlugin 進行連線。

驅動程式	連線選項關鍵字	Value	備註
JDBC	plugin_name	com.amazon.redshift.plugin.IdpTokenAuthPlugin	連線時，您必須輸入外掛程式的完整類別名稱。
ODBC	plugin_name	IdpTokenAuthPlugin	
Python	credentials_provider	IdpTokenAuthPlugin	Python 驅動器沒有可用的 plugin_name 選項。請改用 credentials_provider 。

IdpTokenAuthPlugin 外掛程式有以下其他連線選項：

選項名稱	是否為必要？	說明
token	必要	IAM Identity Center AWS 提供存取字符或 OpenID Connect (OIDC) JSON Web Token (JWT)，由與 AWS IAM Identity Center 連線的 Web 身分提供者提供。您的應用程式必須使用 IAM Identity Center 或與 IAM Identity Center AWS 連線 AWS 的身分提供者來驗證應用程式使用者，以產生此字符。
token_type	必要	用於 `IdpTokenAuthPlugin` 的權杖類型。可能的值如下： • ACCESS_TOKEN – 如果您使用 AWS IAM Identity Center 提供的存取權杖，請輸入此項目。 • EXT_JWT – 如果您使用與 IAM Identity Center 連線的 Web 型身分提供者提供的 OpenID Connect (OIDC) JSON Web Token AWS (JWT)，請輸入此項目。

您必須在您建立並用來連線之工具的連線屬性中輸入這些值。如需詳細資訊，請參閱各驅動器的連線選項文件：

- JDBC 驅動器 2.x 版組態的選項
- ODBC 驅動器選項
- Amazon Redshift Python 連接器的組態選項

對 Amazon Redshift 查詢編輯器 v2 的連線進行故障診斷

此清單詳細說明了經常發生的錯誤，並可協助您使用 IAM Identity Center 身分，使用查詢編輯器 v2 AWS 連線至 Redshift 資料庫。

- 錯誤：Connection Issue: No Identity center session information available. (連線問題：沒有可用的 Identity Center 工作階段資訊。) - 當此錯誤發生時，請檢查您瀏覽器的安全和隱私權設定。這些瀏覽器設定，特別是針對安全 Cookie 的設定 (例如 Firefox 的 Total Cookie Protection 功能)，可能會導致 Amazon Redshift 查詢編輯器 v2 對 Redshift 資料庫的連線嘗試遭到封鎖。請依照您瀏覽器的詳細修復步驟進行：
  - Firefox - 目前預設為封鎖第三方 Cookie。按一下瀏覽器網址列中的盾牌圖示，並將切換開關切換至關閉，以關閉查詢編輯器 v2 的增強型追蹤保護。
  - Chrome incognito 模式 - 根據預設，Chrome Incognito 模式會封鎖第三方 Cookie。按一下網址列中的眼睛圖示，允許查詢編輯器 v2 使用第三方 Cookie。您變更設定以允許 Cookie 之後，網址列上可能不會顯示眼睛圖示。
  - Safari - 在 Mac 上，開啟 Safari app。選擇設定，然後選擇進階。切換至關閉：封鎖所有 Cookie。
  - Edge - 選擇設定，然後選擇 Cookie 和網站許可。然後選取管理和刪除 Cookie 和網站資料，然後關閉封鎖第三方 Cookie。

  如果您在變更設定後嘗試連線，並繼續收到錯誤訊息連線問題：沒有可用的 Identity Center 工作階段資訊，建議您重新整理與 IAM Identity Center AWS 的連線。若要這樣做，請在 Redshift 資料庫執行個體上按一下滑鼠右鍵，然後選擇重新整理。新視窗隨即出現，您可以用來進行身分驗證。
- 錯誤：Connection issue: Identity center session expired or invalid. (連線問題：Identity Center 工作階段已過期或無效。) – 將 Redshift 佈建叢集或 Serverless AWS 工作群組與 IAM Identity Center 整合之後，當使用者嘗試從查詢編輯器 v2 連線至 Redshift 資料庫時，可能會收到此錯誤。成功連線後也可能發生此錯誤。在這種情況下，我們建議您重新驗證身分。若要這樣做，請在 Redshift 資料庫執行個體上按一下滑鼠右鍵，然後選擇重新整理。新視窗隨即出現，您可以用來進行身分驗證。
- 錯誤：Invalid scope. (無效的範圍。) User credentials are not authorized to connect to Redshift. (使用者憑證未經授權無法連線至 Redshift。) – 將 Redshift 佈建叢集或 Serverless 工作群組與 IAM Identity Center AWS 整合以進行身分管理後，當使用者嘗試從查詢編輯器 v2 連線至 Redshift 資料庫時，可能會收到此錯誤。在此情況下，管理員必須透過 Redshift AWS 主控台將使用者指派給 Redshift AWS IAM Identity Center 應用程式，才能透過 IAM Identity Center 成功連接和驗證使用者，以存取正確的資源。此操作會在 IAM Identity Center 連線下完成。之後，使用者可以在一小時後建立成功連線，這是 AWS IAM Identity Center 工作階段快取的限制。

- 錯誤：Databases couldn't be listed. (無法列出資料庫。) FATAL: Failed query when cluster is auto paused. (嚴重：叢集自動暫停時查詢失敗。) – 當 Amazon Redshift Serverless 資料庫處於閒置狀態，而不是處理任何工作負載時，它可以在您使用 IAM Identity Center AWS 身分連線時保持暫停。若要修正此問題，請使用另一種身分驗證方法登入，以繼續執行無伺服器工作群組。然後使用 IAM Identity Center AWS 身分連線到資料庫。
- 錯誤：An error occurred during the attempt to federate with AWS IAM Identity Center. (嘗試與 AWS IAM Identity Center 聯合時發生錯誤。) Amazon Redshift 管理員必須使用 Redshift AWS 主控台刪除並重新建立 IAM Identity Center QEV2 應用程式。– 刪除與查詢編輯器 v2 相關聯的 AWS IAM Identity Center 應用程式執行個體時，通常會發生此錯誤。若要修正此問題，Amazon Redshift 管理員必須刪除並重新建立 IAM Identity Center 的 Redshift AWS 和查詢編輯器 v2 應用程式。此操作可以在 Redshift 主控台上或使用 https://docs.aws.amazon.com/cli/latest/reference/redshift/delete-redshift-idc-application.html CLI 命令執行。

使用 Amazon Redshift 的服務連結角色

Amazon Redshift 使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 Amazon Redshift 的一種特殊 IAM 角色類型。服務連結角色是由 Amazon Redshift 預先定義，並包含服務代表 Amazon Redshift 叢集呼叫 AWS 服務所需的所有許可。

服務連結角色可讓設定 Amazon Redshift 變得更為簡單，因為您不必手動新增必要的許可。角色已連結到 Amazon Redshift 使用案例並且具備預先定義的許可。只有 Amazon Redshift 可以擔任此角色，且只有服務連結角色可以使用預先定義的許可政策。Amazon Redshift 會在您第一次建立叢集或 Redshift 受管 VPC 端點時，在您的帳戶中建立服務連結角色。只有在您刪除帳戶中的所有 Amazon Redshift 叢集或 Redshift 受管 VPC 端點之後，才可以刪除服務連結角色。因為您不會不小心移除存取資源所需的許可，這可保護您的 Amazon Redshift 資源。

Amazon Redshift 在所有提供服務的區域中支援使用服務連結的角色。如需詳細資訊，請參閱 AWS 區域與端點。

如需關於支援服務連結角色的其他服務資訊，請參閱《可搭配 IAM 運作的AWS 服務》，尋找 Service-Linked Role (服務連結角色) 欄中顯示為 Yes (是) 的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

Amazon Redshift 的服務連結角色許可

Amazon Redshift 使用名為 AWSServiceRoleForRedshift 的服務連結角色 – 允許 Amazon Redshift 代表您呼叫 AWS 服務。此服務連結角色連接至下列受管政

策：AmazonRedshiftServiceLinkedRolePolicy。如需此政策的更新，請參閱 Amazon Redshift 的AWS受管 (預先定義) 政策。

AWSServiceRoleForRedshift 服務連結角色只會信任由 **redshift.amazonaws.com** 擔任該角色。

AWSServiceRoleForRedshift 服務連結角色許可政策會允許 Amazon Redshift 在所有相關資源上完成下列動作：

- 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

網路資源的許可

下列許可允許在 Amazon EC2 上執行動作，以建立和管理安全群組規則。這些安全群組和規則特別與 Amazon Redshift aws:RequestTag/Redshift 資源標籤相關聯。這會將許可的適用範圍限制在特定的 Amazon Redshift 資源。

- ec2:CreateSecurityGroup
- ec2:AuthorizeSecurityGroupEgress
- ec2:AuthorizeSecurityGroupIngress
- ec2:RevokeSecurityGroupEgress
- ec2:RevokeSecurityGroupIngress
- ec2:ModifySecurityGroupRules
- ec2:DeleteSecurityGroup

服務配額的許可

下列許可允許呼叫者取得服務配額。

servicequotas:GetServiceQuota

下列 JSON 片段顯示服務配額的動作和資源範圍。

```
{
   "Sid": "ServiceQuotasToCheckCustomerLimits",
   "Effect": "Allow",
   "Action": [
      "servicequotas:GetServiceQuota"
   ],
   "Resource": [
      "arn:aws:servicequotas:*:*:ec2/L-0263D0A3",
      "arn:aws:servicequotas:*:*:vpc/L-29B6F2EB"
   ]
}
```

配額代碼如下：

- L-0263D0A3 - EC2-VPC 彈性 IP 的配額代碼。
- L-29B6F2EB - 每個 VPC 的介面 VPC 端點的配額代碼。

如需更多相關資訊，請參閱 AWS Service Quotas。

稽核記錄的動作

列出 logs 字首的動作與稽核記錄和相關功能有關。具體而言是日誌群組和日誌串流的建立與管理。

- logs:CreateLogGroup
- logs:PutRetentionPolicy
- logs:CreateLogStream
- logs:PutLogEvents
- logs:DescribeLogStreams
- logs:GetLogEvents

下列 JSON 會向 Amazon Redshift 顯示動作和資源範圍，以進行稽核記錄。

```
[
    {
        "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:*"
        ]
    }
]
```

如需服務連結角色及其用途的詳細資訊 AWS，請參閱使用服務連結角色。如需 Amazon Redshift 的特定動作和其他 IAM 資源的相關資訊，請參閱 Amazon Redshift 的動作、資源和條件索引鍵。

使用 管理管理員登入資料的動作 AWS Secrets Manager

列出 `secretsmanager` 字首的動作與使用 Amazon Redshift 來管理管理員憑證有關。這些動作可讓 Amazon Redshift 使用 AWS Secrets Manager 來建立和管理管理員登入資料秘密。

下列 JSON 向 Amazon Redshift 顯示用於管理管理員憑證的動作和資源範圍 AWS Secrets Manager。

```
[
    {
        "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": "*"
    }
]
```

將叢集和無伺服器命名空間註冊到 的動作 AWS Glue Data Catalog

字glue首 列出的動作與存取 中的目錄有關，該目錄 AWS Glue Data Catalog 來自註冊佈建的叢集或無伺服器命名空間。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 Amazon Redshift 的 Apache Iceberg 相容性。

下列 JSON 會向 Amazon Redshift 顯示用於存取 AWS Glue Data Catalog中目錄的動作和資源範圍：

```
[
    {
        "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 許可具有條件 glue:EnabledForRedshiftAutoDiscovery:true，這表示 Amazon Redshift 會授予 IAM 存取權以自動探索目錄。若要選擇退出，請新增 AWS Glue 帳戶層級資源政策，以選擇性地拒絕服務連結角色存取目錄。由於服務連結角色在政策中已有明確的允許動作，因此選擇退出政策需要明確拒絕該動作。請參考下列範例，範例中的其他政策拒絕 Amazon Redshift 的自動探索：

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>/*"
            ]
        }
}
```

允許 IAM 實體建立 AWSServiceRoleForRedshift 服務連結角色

```
{
    "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"}}
}
```

允許 IAM 實體刪除 AWSServiceRoleForRedshift 服務連結角色

將以下政策陳述式新增到該 IAM 實體的許可中：

```
{
    "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 受管政策來提供 Amazon Redshift 的完整存取權。

建立 Amazon Redshift 的服務連結角色

您不需要手動建立 AWSServiceRoleForRedshift 服務連結角色。Amazon Redshift 會為您建立服務連結角色。如果已從您的帳戶刪除 AWSServiceRoleForRedshift 服務連結角色，Amazon Redshift 會在您啟動新的 Amazon Redshift 叢集時建立角色。

> **Important**
>
> Amazon Redshift 自 2017 年 9 月 18 日開始支援服務連結角色，若您在這之前就有使用 Amazon Redshift 服務，則 Amazon Redshift 會在您的帳戶中建立 AWSServiceRoleForRedshift 角色。若要進一步了解，請參閱顯示在我的 IAM 帳戶中的新角色。

編輯 Amazon Redshift 的服務連結角色

Amazon Redshift 不允許您編輯 AWSServiceRoleForRedshift 服務連結角色。因為可能有各種實體會參考服務連結角色，所以您無法在建立角色之後變更其名稱。不過，您可以使用 IAM 主控台、 AWS Command Line Interface (AWS CLI) 或 IAM API 編輯角色的描述。如需詳細資訊，請參閱《IAM 使用者指南》中的修改角色。

刪除 Amazon Redshift 的服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。

您必須先關閉和刪除帳戶中的任何叢集，才能刪除帳戶的服務連結角色。如需詳細資訊，請參閱關閉及刪除叢集。

您可以使用 IAM 主控台 AWS CLI、 或 IAM API 來刪除服務連結角色。如需詳細資訊，請參閱《IAM 使用者指南》中的刪除服務連結角色。

使用 IAM 身分驗證產生資料庫使用者憑證

您可依 AWS Identity and Access Management (IAM) 許可政策所授予的許可產生暫時性資料庫憑證，以管理使用者對 Amazon Redshift 資料庫的存取。

Amazon Redshift 資料庫使用者通常會使用資料庫使用者名稱和密碼來登入資料庫。但是，您不必維護 Amazon Redshift 資料庫中的使用者名稱和密碼。另一種方法是，您可以將系統設定為允許使用者建立使用者登入資料，並根據其 IAM 登入資料登入資料庫。

Amazon Redshift 提供 GetClusterCredentials API 操作，用於產生暫存資料庫使用者憑證。
您可以使用 Amazon Redshift JDBC 或 ODBC 驅動程式來設定 SQL 用戶端，以便管理呼叫 `GetClusterCredentials` 操作的程序。其作法是擷取資料庫使用者憑證，然後在 SQL 用戶端和 Amazon Redshift 資料庫之間建立連線。但您也可以使用資料庫應用程式，以程式設計的方式呼叫 `GetClusterCredentials` 操作，擷取資料庫使用者登入資料，然後連線至資料庫。

如果您已在外部管理使用者身分 AWS，您可以使用符合安全性聲明標記語言 (SAML) 2.0 的身分提供者 (IdP) 來管理對 Amazon Redshift 資源的存取。您需設定 IdP，允許聯合身分使用者存取 IAM 角色。您可利用該 IAM 角色產生暫存資料庫憑證，並登入 Amazon Redshift 資料庫。

您的 SQL 用戶端需要許可，才能為您呼叫 `GetClusterCredentials` 操作。若要管理這些許可，您需建立 IAM 角色並連接 IAM 許可政策，以授予或限制對 `GetClusterCredentials` 操作和相關動作的存取。我們建議的最佳實務是，將許可政策附加到 IAM 角色，然後根據需要將其指派給使用者和群組。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。

政策也會授予或限制特定資源的存取，例如 Amazon Redshift 叢集、資料庫、資料庫使用者名稱和使用者群組名稱。

> **Note**
>
> 我們建議使用 Amazon Redshift JDBC 或 ODBC 驅動程式來管理呼叫 `GetClusterCredentials` 操作及登入資料庫的程序。為簡化程序，我們假設您在本主題中都透過 JDBC 或 ODBC 驅動程式使用 SQL 用戶端。
> 如需有關使用 `GetClusterCredentials` 操作或平行 `get-cluster-credentials` CLI 命令的具體細節和範例，請參閱 GetClusterCredentials 和 get-cluster-credentials。

為了集中管理身分驗證和授權，Amazon Redshift 支援使用 IAM 進行資料庫身分驗證，可讓您透過企業聯合進行使用者身分驗證。您可以使用現有的身分 AWS Directory Service、企業使用者目錄或 Web 身分提供者，而不是建立使用者。這些稱為聯合身分使用者。當透過 IdP 請求存取時， AWS 會將角色指派給聯合身分使用者。

若要為組織使用者或用戶端應用程式提供同盟存取以呼叫 Amazon Redshift API 操作，您也可以使用具有 SAML 2.0 支援的 JDBC 或 ODBC 驅動程式，向組織 IdP 請求身分驗證。在這種情況下，組織的使用者無法直接存取 Amazon Redshift。

如需詳細資訊，請參閱《IAM 使用者指南》中的身分提供者與聯合。

建立暫時性 IAM 憑證

您將在本節中了解如何設定系統以產生暫時性 IAM 型資料庫使用者登入資料，並使用新的登入資料來登入資料庫。

高階的程序流程如下：

1. 步驟 1：建立適用於 IAM 單一登入存取的 IAM 角色

   (選用) 您可整合 IAM 身分驗證和第三方身分提供者 (IdP)，對存取 Amazon Redshift 資料庫的使用者進行驗證。
2. 步驟 2：設定 IdP 的 SAML 聲明

   (選用) 若要利用 IdP 使用 IAM 身分驗證，您需要在 IdP 應用程式中定義宣告規則，將組織中的使用者或群組映射到 IAM 角色。您也可選擇性加入屬性元素，以設定 `GetClusterCredentials` 參數。

3. 步驟 3：建立具有呼叫 GetClusterCredentialsWithIAM 或 GetClusterCredentials 許可的 IAM 角色 GetClusterCredentials

   SQL 用戶端應用程式在呼叫 `GetClusterCredentials` 操作時會擔任使用者。如果已建立用於身分提供者存取的 IAM 角色，您可為該角色新增必要的許可。

4. 步驟 4：建立資料庫使用者和資料庫群組

   (選用) 根據預設，如果使用者名稱不存在，`GetClusterCredentials` 會傳回登入資料以建立新的使用者。您也可以選擇指定使用者登入時會加入的使用者群組。根據預設，資料庫使用者將加入 PUBLIC 群組。

5. 步驟 5：設定 JDBC 或 ODBC 連線使用 IAM 登入資料

   若要連線至 Amazon Redshift 資料庫，請將 SQL 用戶端設定為使用 Amazon Redshift JDBC 或 ODBC 驅動程式。

步驟 1：建立適用於 IAM 單一登入存取的 IAM 角色

如果您未使用身分提供者進行單一登入存取，可略過此步驟。

如果您已在 外部管理使用者身分 AWS，您可以透過整合 IAM 身分驗證和第三方 SAML-2.0 身分提供者 (IdP) 來驗證使用者對 Amazon Redshift 資料庫的存取。

如需詳細資訊，請參閱《IAM 使用者指南》中的身分提供者與聯合。

在使用 Amazon Redshift IdP 身分驗證之前，請先建立 AWS SAML 身分提供者。您可以在 IAM 主控台中建立 IdP，以通知 AWS IdP 及其組態。這樣做會在 AWS 您的帳戶和 IdP 之間建立信任。如需建立角色的步驟，請參閱《IAM 使用者指南》中的為 SAML 2.0 聯合建立角色 (主控台)。

步驟 2：設定 IdP 的 SAML 聲明

建立 IAM 身分後，請在 IdP 應用程式中定義宣告規則，以將組織中的使用者或群組映射到 IAM 角色。如需詳細資訊，請參閱《IAM 使用者指南》中的的為身分驗證回應設定 SAML 聲明。

如果您選擇使用選用的 `GetClusterCredentials` 參數 `DbUser`、`AutoCreate`、和 `DbGroups`，則有兩個選項。您可以使用 JDBC 或 ODBC 連線來設定參數值，或將 SAML 屬性元素新增至 IdP 以設定數值。如需 `DbUser`、`AutoCreate` 和 `DbGroups` 參數的詳細資訊，請參閱 步驟 5：設定 JDBC 或 ODBC 連線使用 IAM 登入資料。

> **Note**
>
> 如果您使用 IAM 政策變數 `${redshift:DbUser}`，如 GetClusterCredentials 的資源政策中所述，則會以 API 操作的請求內容所擷取的值取代 `DbUser` 的值。Amazon Redshift 驅動程式使用連線 URL 提供的 `DbUser` 變數的值，而非提供作為 SAML 屬性的值。
> 為了協助保護此組態的安全，建議您在 IAM 政策中使用條件，以使用 `RoleSessionName` 來驗證 `DbUser` 值。您可以在範例 8：使用 GetClusterCredentials 的 IAM 政策中找到如何使用 IAM 政策來設定條件的範例。

若要設定 IdP 以設定 `DbUser`、`AutoCreate` 和 `DbGroups` 參數，請包含下列 `Attribute` 元素：

- Name 屬性設定為 "https://redshift.amazon.com/SAML/Attributes/DbUser" 的 `Attribute` 元素

  將 `AttributeValue` 設定為即將連線至 Amazon Redshift 資料庫的使用者名稱。

  `AttributeValue` 元素中的值長度必須為小寫，開頭需為英文字母，只能包含英數字元、底線 (「_」)、加號 (「+」)、點號 (「.」)、At (「@」) 或連字號 (「-」)，且不得超過 128 個字元。使用者名稱的通常為使用者 ID (例如，bobsmith) 或電子郵件地址 (例如，bobsmith@example.com)。該值不能包含空格 (例如，Bob Smith 之類的使用者顯示名稱)。

  ```
  <Attribute Name="https://redshift.amazon.com/SAML/Attributes/DbUser">
      <AttributeValue>user-name</AttributeValue>
  </Attribute>
  ```

- 名稱屬性設定為「https://redshift.amazon.com/SAML/Attributes/AutoCreate」的屬性元素

  如果尚不存在，將 AttributeValue 元素設定為 true，即可建立新的資料庫使用者。將 AttributeValue 設定為 false，指定資料庫使用者必須存在於 Amazon Redshift 資料庫中。

  ```
  <Attribute Name="https://redshift.amazon.com/SAML/Attributes/AutoCreate">
      <AttributeValue>true</AttributeValue>
  </Attribute>
  ```

- Name 屬性設定為 "https://redshift.amazon.com/SAML/Attributes/DbGroups" 的 `Attribute` 元素

  此元素包含一或多個 `AttributeValue` 元素。將每個 `AttributeValue` 元素設定為 `DbUser` 在連線至 Amazon Redshift 資料庫之工作階段期間將加入的資料庫群組名稱。

  ```
  <Attribute Name="https://redshift.amazon.com/SAML/Attributes/DbGroups">
  ```

```
    <AttributeValue>group1</AttributeValue>
    <AttributeValue>group2</AttributeValue>
    <AttributeValue>group3</AttributeValue>
</Attribute>
```

步驟 3：建立具有呼叫 GetClusterCredentialsWithIAM 或 GetClusterCredentials 許可的 IAM 角色 GetClusterCredentials

您的 SQL 用戶端需要授權，才能代表您呼叫 GetClusterCredentialsWithIAM或 GetClusterCredentials操作。若要提供授權，您應建立使用者或角色，並附加授予必要許可的政策。這兩個操作都可用來取得叢集登入資料，但其身分驗證方法不同。GetClusterCredentialsWithIAM使用 IAM 角色，自動建立映射至角色的資料庫使用者，這有助於管理 IAM 角色層級的許可，同時為資料庫中指定的使用者名稱GetClusterCredentials提供登入資料。

建立具有呼叫 GetClusterCredentialsWithIAM 許可的 IAM 角色

1. 使用 IAM 服務建立使用者或角色。也可使用現有的使用者或角色，例如，如果已建立用於身分提供者存取的 IAM 角色，您可將必要的 IAM 政策連接至該角色。
2. 連接具有呼叫 redshift:GetClusterCredentialsWithIAM 操作許可的許可政策。下列政策範例顯示允許特定叢集和資料庫、叢集中的任何資料庫，以及任何叢集中的任何資料庫操作的選項。

```
{
    "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": "*"
        }
    ]
}
```

建立具有呼叫 GetClusterCredentials 之許可的 IAM 角色

1. 使用 IAM 服務建立使用者或角色。也可使用現有的使用者或角色，例如，如果已建立用於身分提供者存取的 IAM 角色，您可將必要的 IAM 政策連接至該角色。
2. 連接具有呼叫 `redshift:GetClusterCredentials` 操作許可的許可政策。根據指定的選用參數而定，您也能在政策中允許或限制其他動作和資源：

   - 若要允許 SQL 用戶端擷取叢集 ID、 AWS 區域和連接埠，請包含使用 Redshift 叢集資源呼叫 `redshift:DescribeClusters`操作的許可。
   - 如果使用 AutoCreate 選項，請在 dbuser 資源中包含呼叫 `redshift:CreateClusterUser` 的許可。以下 Amazon Resource Name (ARN) 會指定 Amazon Redshift dbuser。*cluster-name* 將 *account-id*、 *region*和 取代為您的 AWS 區域、帳戶和叢集的值。針對 *dbuser-name*，指定用來登入叢集資料庫的使用者名稱。

     ```
     arn:aws:redshift:region:account-id:dbuser:cluster-name/dbuser-name
     ```

   - (選用) 新增 ARN 以指定 Amazon Redshift dbname 資源，格式如下。*cluster-name* 將 *account-id*、 *region*和 取代為您的 AWS 區域、帳戶和叢集的值。針對 *database-name*，指定使用者將登入的資料庫名稱。

     ```
     arn:aws:redshift:region:account-id:dbname:cluster-name/database-name
     ```

   - 如果您使用 DbGroups 選項，請包含對 Amazon Redshift dbgroup 資源呼叫 `redshift:JoinGroup` 操作的許可，格式如下。*cluster-name* 將 *account-id*、

*region*和 取代為您的 AWS 區域、帳戶和叢集的值。針對 *dbgroup-name*，指定使用者登入時要加入的使用者群組名稱。

```
arn:aws:redshift:region:account-id:dbgroup:cluster-name/dbgroup-name
```

如需詳細資訊和範例，請參閱 GetClusterCredentials 的資源政策。

步驟 4：建立資料庫使用者和資料庫群組

您可選擇性建立資料庫使用者，以用來登入叢集資料庫。如果您為現有的使用者建立暫時性使用者登入資料，您可停用使用者的密碼，以強制使用者用暫時性密碼登入。或者，您也可使用 GetClusterCredentials Autocreate 選項自動建立新的資料庫使用者。

您可建立資料庫使用者群組並加入允許 IAM 資料庫使用者於登入時加入的許可。呼叫 GetClusterCredentials 操作時，您可指定新使用者於登入時加入的使用者群組名稱清單。這些群組成員資格僅適用於使用指定請求產生之登入資料所建立的工作階段。

建立資料庫使用者和資料庫群組

1. 登入 Amazon Redshift 資料庫並使用 CREATE USER 建立資料庫使用者，或使用 ALTER USER 來變更現有的使用者。
2. 選擇性地指定 PASSWORD DISABLE 選項，防止使用者使用密碼。停用使用者密碼後，使用者只能使用暫時性憑證登入。如果未停用密碼，使用者可用其密碼或暫時性憑證登入。您無法停用超級使用者的密碼。

   如果使用者想要與 AWS 外部互動，則需要程式設計存取 AWS 管理主控台。授予程式設計存取權的方式取決於正在存取的使用者類型 AWS。

   若要授予使用者程式設計存取權，請選擇下列其中一個選項。

哪個使用者需要程式設計存取權？	到	根據
IAM	（建議）使用主控台登入資料做為臨時登入資料，以簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》

哪個使用者需要程式設計存取權？	到	根據
		中的登入以進行 AWS 本機開發。 • AWS SDKs，請參閱 AWS SDKs 和工具參考指南中的登入以進行 AWS 本機開發。
人力資源身分 (IAM Identity Center 中管理的使用者)	使用臨時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的設定 AWS CLI 要使用 AWS IAM Identity Center的 。 • AWS SDKs、工具和 AWS APIs，請參閱 AWS SDK 和工具參考指南中的 SDKsIAM Identity Center 身分驗證。
IAM	使用臨時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	遵循《IAM 使用者指南》中將臨時登入資料與 AWS 資源搭配使用的指示。

哪個使用者需要程式設計存取權？	到	根據
IAM	(不建議使用) 使用長期登入資料來簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的使用 IAM 使用者憑證進行身分驗證。 • AWS SDKs和工具，請參閱 AWS SDKs和工具參考指南中的使用長期憑證進行身分驗證。 • 對於 AWS APIs，請參閱《IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。

以下範例會建立停用密碼的使用者。

```
create user temp_creds_user password disable;
```

以下範例會停用現有使用者的密碼。

```
alter user temp_creds_user password disable;
```

3. 使用 CREATE GROUP 建立資料庫使用者群組。
4. 使用 GRANT 命令定義群組的存取權限。

步驟 5：設定 JDBC 或 ODBC 連線使用 IAM 登入資料

您可以使用 Amazon Redshift JDBC 或 ODBC 驅動程序設定您的 SQL 用戶端。此驅動程式會管理建立資料庫使用者憑證，然後在 SQL 用戶端與 Amazon Redshift 資料庫之間建立連線的程序。

如果您使用身分提供者進行身分驗證，請指定登入資料供應商外掛程式的名稱。Amazon Redshift JDBC 和 ODBC 驅動程式包含下列以 SAML 為基礎之身分提供者的外掛程式：

- Active Directory Federation Services (AD FS)
- PingOne
- Okta
- Microsoft Azure AD

  如需將 Microsoft Azure AD 設定為身分提供者的步驟，請參閱 設定 JDBC 或 ODBC 單一登入身分驗證。

設定 JDBC 連線使用 IAM 登入資料

1. 從 設定適用於 Amazon Redshift 的 JDBC 驅動器 2.x 版的連線 頁面下載最新的 Amazon Redshift JDBC 驅動程式。
2. 使用下列其中一種格式的 IAM 登入資料選項建立 JDBC URL。若要使用 IAM 身分驗證，請將 `iam:` 新增至 Amazon Redshift JDBC URL 的 `jdbc:redshift:` 後方，如下列範例所示。

   ```
   jdbc:redshift:iam://
   ```

   新增 `cluster-name`、`region` 和 `account-id`。JDBC 驅動程式使用您的 IAM 帳戶資訊和叢集名稱來擷取叢集 ID 和 AWS 區域。若要這樣做，您的使用者或角色必須具備為指定叢集呼叫 `redshift:DescribeClusters` 操作的許可。如果您的使用者或角色沒有呼叫 `redshift:DescribeClusters`操作的許可，請包含叢集 ID、 AWS 區域和連接埠，如下列範例所示。連接埠號碼是選用的。

   ```
   jdbc:redshift:iam://examplecluster.abc123xyz789.us-
   west-2.redshift.amazonaws.com:5439/dev
   ```

3. 新增 JDBC 選項，以提供 IAM 登入資料。您可使用不同的 JDBC 選項組合來提供 IAM 登入資料。如需詳細資訊，請參閱用於建立資料庫使用者憑證的 JDBC 和 ODBC 選項。

   以下 URL 指定使用者的 AccessKeyID 和 SecretAccessKey。

   ```
   jdbc:redshift:iam://examplecluster:us-west-2/dev?
   AccessKeyID=AKIAIOSFODNN7EXAMPLE&SecretAccessKey=wJalrXUtnFEMI/K7MDENG/
   bPxRfiCYEXAMPLEKEY
   ```

以下範例指定包含 IAM 登入資料的命名設定檔。

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?Profile=user2
```

4. 新增 JDBC 驅動程式用來呼叫 `GetClusterCredentials` API 操作的 JDBC 選項。如果您用程式設計的方式呼叫 `GetClusterCredentials` API 操作，請勿加入這些選項。

   下列範例包含 JDBC `GetClusterCredentials` 選項。

```
jdbc:redshift:iam://examplecluster:us-west-2/dev?
plugin_name=com.amazon.redshift.plugin.AzureCredentialsProvider&UID=user&PWD=password&idp_t
```

設定 ODBC 連線使用 IAM 登入資料

在以下程序中，您只能尋找設定 IAM 身分驗證的步驟。如需使用標準身分驗證 (使用資料庫使用者名稱和密碼) 的步驟，請參閱 設定 Amazon Redshift ODBC 驅動程式 2.x 版連線。

1. 安裝並設定您的作業系統適用的 Amazon Redshift OBDC 驅動程式。如需詳細資訊，請參閱 設定 Amazon Redshift ODBC 驅動程式 2.x 版連線 頁面。

   > **Important**
   >
   > Amazon Redshift ODBC 驅動程式必須為 1.3.6.1000 版或更新版本。

2. 依照您作業系統適用的步驟，設定連線設定。
3. 在 Microsoft Windows 作業系統中，存取 Amazon Redshift ODBC Driver DSN Setup 視窗。

   a. 在 Connection Settings (連線設定) 下，輸入下列資訊：

      - Data Source Name (資料來源名稱)
      - Server (伺服器) (選用)
      - Port (連接埠) (選用)
      - 資料庫

      如果您的使用者或角色具有呼叫 `redshift:DescribeClusters` 操作的許可，則只需要資料來源名稱和資料庫。Amazon Redshift 會使用 ClusterId 和區域，透過呼叫 `DescribeCluster` 操作來取得伺服器和連接埠。

如果您的使用者或角色不具有呼叫 `redshift:DescribeClusters` 操作的許可，請指定伺服器和連接埠。

b. 在 Authentication (身分驗證) 下方，選擇 Auth Type (驗證類型) 的值。

針對每個驗證類型，輸入下列的值：

AWS 設定檔

輸入下列資訊：

- ClusterID (ClusterID)
- 區域
- Profile name (設定檔名稱)

  在包含 ODBC AWS 連線選項值的組態檔案中，輸入設定檔的名稱。如需詳細資訊，請參閱使用組態設定檔。

(選用) 提供 ODBC 驅動程式用來呼叫 `GetClusterCredentials` API 操作之選項的詳細資訊：

- DbUser (DbUser)
- User AutoCreate (使用者 AutoCreate)
- DbGroups (DbGroups)

  如需詳細資訊，請參閱用於建立資料庫使用者憑證的 JDBC 和 ODBC 選項。

IAM 登入資料

輸入下列資訊：

- ClusterID (ClusterID)
- 區域
- AccessKeyID (AccessKeyID) 和 SecretAccessKey (SecretAccessKey)

  IAM 資料庫身分驗證所設定之 IAM 角色或使用者的存取金鑰 ID 和私密存取金鑰。

- SessionToken (SessionToken)

  SessionToken (SessionToken) 對於擁有暫時性登入資料的 IAM 角色來說為必填。如需詳細資訊，請參閱暫時性安全登入資料。

提供 ODBC 驅動程式用來呼叫 `GetClusterCredentials` API 操作之選項的詳細資訊：

- DbUser (DbUser) (必填)
- User AutoCreate (使用者 AutoCreate) (選用)
- DbGroups (DbGroups) (選用)

  如需詳細資訊，請參閱用於建立資料庫使用者憑證的 JDBC 和 ODBC 選項。

身分提供者：AD FS

針對使用 AD FS 的 Windows 整合式身分驗證，請讓 User (使用者) 和 Password (密碼) 保持空白。

提供 IdP 詳細資訊：

- IdP Host (IdP 主機)

  企業身分提供商主機的名稱。該名稱不得包含任何斜線 (/)。

- IdP Port (IdP 連接埠) (選用)

  身分提供者使用的連接埠。預設為 443。

- Preferred Role (偏好的角色)

  IAM 角色的 Amazon Resource Name (ARN)，來自 SAML 聲明中 `Role` 屬性的多值 `AttributeValue` 元素。若要尋找偏好角色適用的值，請洽詢 IdP 管理員。如需詳細資訊，請參閱步驟 2：設定 IdP 的 SAML 聲明。

(選用) 提供 ODBC 驅動程式用來呼叫 `GetClusterCredentials` API 操作之選項的詳細資訊：

- DbUser (DbUser)
- User AutoCreate (使用者 AutoCreate)
- DbGroups (DbGroups)

如需詳細資訊，請參閱用於建立資料庫使用者憑證的 JDBC 和 ODBC 選項。

身分提供者：PingFederate

針對 User (使用者) 和 Password (密碼)，輸入 IdP 的使用者名稱和密碼。

提供 IdP 詳細資訊：

- IdP Host (IdP 主機)

  企業身分提供商主機的名稱。該名稱不得包含任何斜線 (/)。

- IdP Port (IdP 連接埠) (選用)

  身分提供者使用的連接埠。預設為 443。

- Preferred Role (偏好的角色)

  IAM 角色的 Amazon Resource Name (ARN)，來自 SAML 聲明中 `Role` 屬性的多值 `AttributeValue` 元素。若要尋找偏好角色適用的值，請洽詢 IdP 管理員。如需詳細資訊，請參閱步驟 2：設定 IdP 的 SAML 聲明。

(選用) 提供 ODBC 驅動程式用來呼叫 `GetClusterCredentials` API 操作之選項的詳細資訊：

- DbUser (DbUser)
- User AutoCreate (使用者 AutoCreate)
- DbGroups (DbGroups)

如需詳細資訊，請參閱用於建立資料庫使用者憑證的 JDBC 和 ODBC 選項。

身分提供者：Okta

針對 User (使用者) 和 Password (密碼)，輸入 IdP 的使用者名稱和密碼。

提供 IdP 詳細資訊：

- IdP Host (IdP 主機)

  企業身分提供商主機的名稱。該名稱不得包含任何斜線 (/)。

- IdP Port (IdP 連接埠)

  Okta 不會使用此值。

- Preferred Role (偏好的角色)

  IAM 角色的 Amazon Resource Name (ARN)，來自 SAML 聲明中 `Role` 屬性的 `AttributeValue` 元素。若要尋找偏好角色適用的值，請洽詢 IdP 管理員。如需詳細資訊，請參閱步驟 2：設定 IdP 的 SAML 聲明。

- Okta App ID (Okta 應用程式 ID)

  Okta 應用程式的 ID。Okta 應用程式嵌入連結中「amazon_aws」後方的 App ID 值。請洽詢 IdP 管理員，以取得此值。

(選用) 提供 ODBC 驅動程式用來呼叫 `GetClusterCredentials` API 操作之選項的詳細資訊：

- DbUser (DbUser)
- User AutoCreate (使用者 AutoCreate)
- DbGroups (DbGroups)

如需詳細資訊，請參閱用於建立資料庫使用者憑證的 JDBC 和 ODBC 選項。

身分提供者：Azure AD

針對 User (使用者) 和 Password (密碼)，輸入 IdP 的使用者名稱和密碼。

針對叢集 ID 和區域，輸入您的 Amazon Redshift 叢集的叢集 ID 和 AWS 區域。

針對資料庫，輸入您為 Amazon Redshift 叢集建立的資料庫。

提供 IdP 詳細資訊：

- IdP Tenant (IdP 租用戶)

  用於 Azure AD 的租用戶。

- Azure Client Secret (Azure 用戶端密碼)

  Azure 中 Amazon Redshift 企業應用程式的用戶端密碼。

- Azure Client ID (Azure 用戶端 ID)

  Azure 中 Amazon Redshift 企業應用程式的用戶端 ID (應用程式 ID)。

(選用) 提供 ODBC 驅動程式用來呼叫 `GetClusterCredentials` API 操作之選項的詳細資訊：

- DbUser (DbUser)
- User AutoCreate (使用者 AutoCreate)
- DbGroups (DbGroups)

如需詳細資訊，請參閱用於建立資料庫使用者憑證的 JDBC 和 ODBC 選項。

提供 IAM 登入資料的選項

若要提供 JDBC 或 ODBC 連線的 IAM 登入資料，請選擇下列選項之一。

- AWS profile

  做為以 JDBC 或 ODBC 設定形式提供登入資料值的替代方案，您可將值放入命名設定檔。如需詳細資訊，請參閱使用組態設定檔。
- IAM 登入資料

  以 JDBC 或 ODBC 設定形式提供 AccessKeyID、SecretAccessKey 和 (選用) SessionToken 的值。SessionToken 只對擁有暫時性登入資料的 IAM 角色為必填。如需詳細資訊，請參閱提供 IAM 登入資料的 JDBC 和 ODBC 選項。
- 身分提供者聯合

  當您使用身分提供者聯合，讓來自身分提供者的使用者向 Amazon Redshift 進行驗證時，請指定憑證供應商外掛程式的名稱。如需詳細資訊，請參閱憑證提供者外掛程式。

  Amazon Redshift JDBC 和 ODBC 驅動程式包含下列以 SAML 為基礎之聯合身分憑證供應商的外掛程式：

  - Microsoft Active Identity Federation Services (AD FS)
  - PingOne
  - Okta
  - Microsoft Azure Active Directory (Azure AD)

  您可用 JDBC 或 ODBC 設定形式或使用設定檔提供外掛程式名稱和相關值。如需詳細資訊，請參閱JDBC 驅動器 2.x 版組態的選項。

如需詳細資訊，請參閱步驟 5：設定 JDBC 或 ODBC 連線使用 IAM 登入資料。

使用組態設定檔

您可以在 AWS 組態檔案中的具名設定檔中提供 IAM 登入資料選項和`GetClusterCredentials`選項做為設定。若要提供設定檔名稱，請使用設定檔 JDBC 選項。組態儲存於主目錄內名為 `.aws` 之資料夾中名為 `config` 的檔案或名為 `credentials` 的檔案中。

針對具有 Amazon Redshift JDBC 或 ODBC 驅動程式之以 SAM 為基礎的憑證供應商外掛程式，您可以使用先前在憑證提供者外掛程式中所述的設定。如果 plugin_name 未使用，則會忽略其他選項。

下列範例顯示具有兩個設定檔的 ~/.aws/credentials 檔案：

```
[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==
```

若要使用 user2 範例的登入資料，請在 JDBC URL 中指定 Profile=user2。

如需使用設定檔的詳細資訊，請參閱《 使用者指南》中的組態和登入資料檔案設定。 AWS Command Line Interface

如需為 JDBC 驅動程式使用設定檔的相關資訊，請參閱指定設定檔。

如需為 ODBC 驅動程式使用設定檔的相關資訊，請參閱身分驗證方法。

提供 IAM 登入資料的 JDBC 和 ODBC 選項

下表列出提供 IAM 登入資料的 JDBC 和 ODBC 選項。

選項	說明
Iam	僅使用於 ODBC 連線字串。設為 1 以使用 IAM 身分驗證。
AccessKeyID SecretAccessKey	IAM 資料庫身分驗證所設定之 IAM 角色或使用者的存取金鑰 ID 和私密存取金鑰。SessionToken　僅需用於具有暫時憑證的 IAM 角色。SessionToken 不適用於使用者。如需詳細資訊，請參閱暫時性安全登入資料。

選項	說明
SessionToken	
plugin_name	實作登入資料供應商之類別的完全合格名稱。Amazon Redshift JDBC 驅動程式包含以 SAML 為基礎的憑證供應商外掛程式。如果您提供 plugin_name ，您也可以提供其他相關選項。如需詳細資訊，請參閱憑證提供者外掛程式。
Profile	AWS 登入資料或組態檔案中包含 JDBC 連線選項值的設定檔名稱。如需詳細資訊，請參閱使用組態設定檔。

用於建立資料庫使用者憑證的 JDBC 和 ODBC 選項

若要使用 Amazon Redshift JDBC 或 ODBC 驅動程式建立資料庫使用者憑證，請以 JDBC 或 ODBC 選項提供資料庫使用者名稱。如果資料庫使用者不存在，您也可選擇性讓驅動程式建立新的資料庫使用者，此外也可指定使用者於登入時加入的資料庫使用者群組清單。

如果您使用身分提供者 (IdP)，請洽詢 IdP 管理員，以決定這些選項的正確值。IdP 管理員也可設定您的 IdP，以提供這些選項，如此您便不用透過 JDBC 或 ODBC 選項提供。如需詳細資訊，請參閱步驟 2：設定 IdP 的 SAML 聲明。

> **Note**
>
> 如果您使用 IAM 政策變數 ${redshift:DbUser}，如 GetClusterCredentials 的資源政策中所述，則會以 API 操作的請求內容所擷取的值取代 DbUser 的值。Amazon Redshift 驅動程式使用連線 URL 提供的 DbUser 變數的值，而非提供作為 SAML 屬性的值。
> 為了協助保護此組態的安全，建議您在 IAM 政策中使用條件，配合 RoleSessionName 來驗證 DbUser 值。您可以在範例 8：使用 GetClusterCredentials 的 IAM 政策中找到如何使用 IAM 政策來設定條件的範例。

下表列出提供用於建立資料庫使用者登入資料的選項。

選項	說明
DbUser	資料庫使用者的名稱。如果資料庫中有名稱為 DbUser 的使用者，暫時性使用者登入資料與現有使用者會有相同的許可。如果資料庫中不存在 DbUser，且 AutoCreate 為

選項	說明
	true，將建立名為 DbUser 的新使用者。您也可選擇性停用現有使用者的密碼。如需詳細資訊，請參閱 ALTER_USER
AutoCreate	如果資料庫使用者不存在，指定 `true` 將建立資料庫使用者，並將其命名為 DbUser 指定的名稱。預設值為 false。
`DbGroups`	資料庫使用者在目前工作階段中加入之一或多個現有資料庫群組名稱的逗號分隔清單。根據預設，新使用者只會加入 PUBLIC。

憑證提供者外掛程式

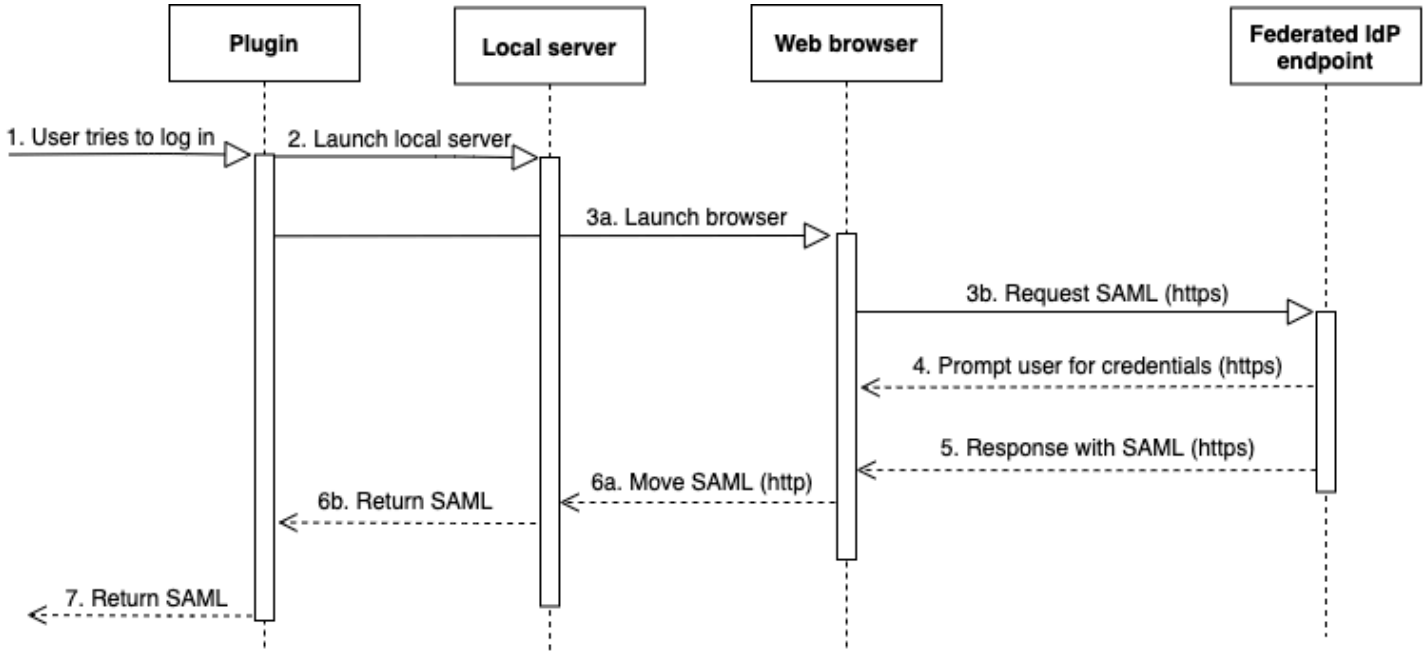
Amazon Redshift 使用憑證供應商外掛程式進行單一登入身分驗證。

為了支援單一登入身分驗證，Amazon Redshift 提供適用於 Microsoft Azure Active Directory 的 Azure AD 外掛程式。如需如何設定此外掛程式的資訊，請參閱設定 JDBC 或 ODBC 單一登入身分驗證。

多重要素驗證
多重要素驗證

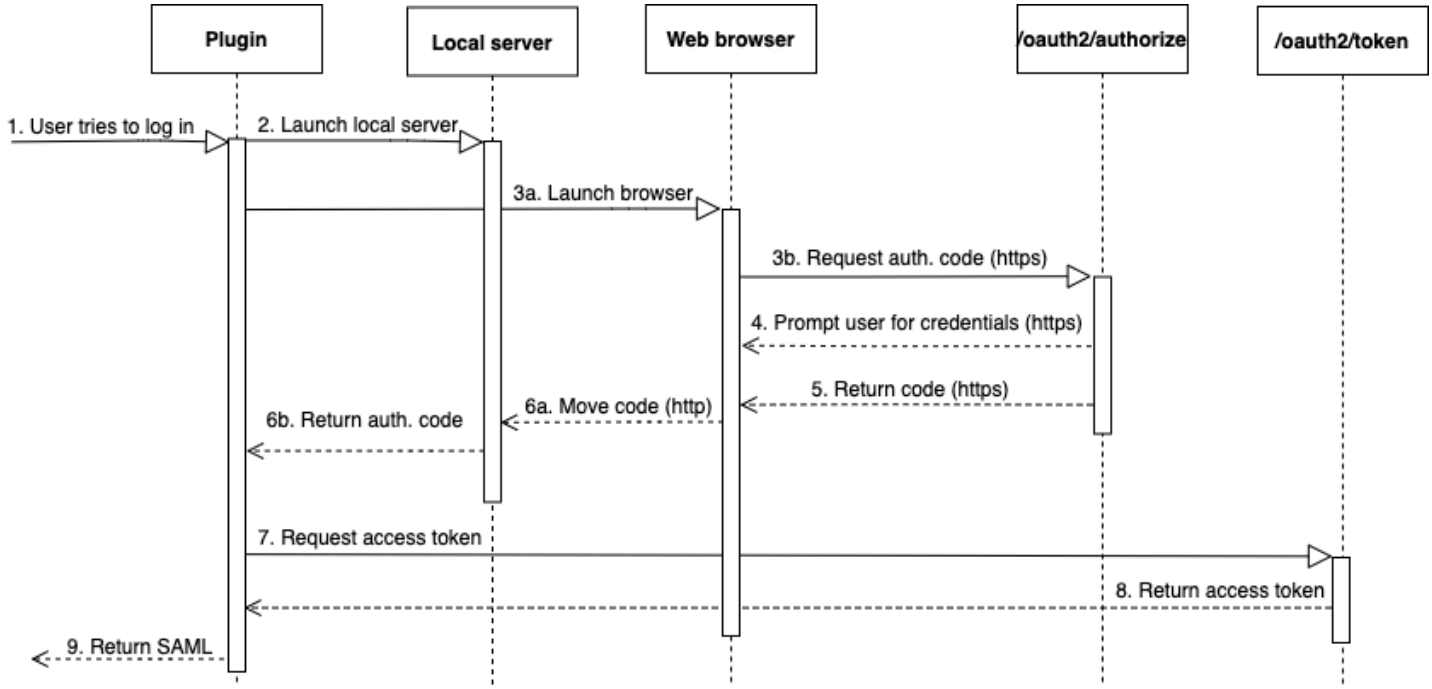
為了支援多重要素驗證 (MFA)，Amazon Redshift 提供瀏覽器型的外掛程式。使用適用於 Okta、PingOne 的瀏覽器 SAML 外掛程式，以及適用於 Microsoft Azure Active Directory 的瀏覽器 Azure AD 外掛程式。

使用瀏覽器 SAML 外掛程式時的 OAuth 身分驗證流程如下：



1. 使用者嘗試登入。
2. 外掛程式會啟動本機伺服器來接聽 localhost 上的傳入連線。
3. 外掛程式會啟動 web 瀏覽器，從指定的單一登入 URL 聯合身分識別供應商端點透過 HTTPS 請求 SAML 回應。
4. web 瀏覽器會遵循連結，並提示使用者輸入登入資料。
5. 在使用者驗證並授予同意後，聯合身分提供者端點會透過 HTTPS 將 SAML 回應傳回由 `redirect_uri` 指定的 URI。
6. web 瀏覽器會將回應訊息與 SAML 回應移至指定的 `redirect_uri`。
7. 本機伺服器會接受傳入的連線，而外掛程式會擷取 SAML 回應並將其傳遞給 Amazon Redshift。

使用瀏覽器 Azure AD 外掛程式時，SAML 身分驗證的流程如下：



1. 使用者嘗試登入。
2. 外掛程式會啟動本機伺服器來接聽 localhost 上的傳入連線。
3. 外掛程式會啟動 web 瀏覽器，以從 Azure AD `oauth2/authorize` 端點請求授權碼。
4. web 瀏覽器會透過 HTTPS 遵循所產生的連結，並提示使用者輸入登入資料。連結是使用組態屬性 (例如租用戶和 client_id) 所產生的。
5. 在使用者進行驗證並授予同意後，Azure AD `oauth2/authorize` 端點會傳回，並透過 HTTPS 將包含授權碼的回應傳送至指定 `redirect_uri`。
6. web 瀏覽器會將回應訊息與 SAML 回應移至指定的 `redirect_uri`。
7. 本機伺服器會接受傳入的連線和外掛程式請求並擷取授權碼，然後將 POST 請求傳送至 Azure AD `oauth2/token` 端點。
8. Azure AD `oauth2/token` 端點會將包含存取權杖的回應傳回指定 `redirect_uri`。
9. 外掛程式會擷取 SAML 回應並將其傳遞給 Amazon Redshift。

請參閱以下章節：

- Active Directory Federation Services (AD FS)

  如需詳細資訊，請參閱設定 JDBC 或 ODBC 單一登入身分驗證。
- PingOne (Ping)

Ping 只有在使用 Forms 身分驗證的預先決定 PingOne IdP 轉接器時受到支援。

如需詳細資訊，請參閱設定 JDBC 或 ODBC 單一登入身分驗證。

- Okta

  Okta 只有在使用 Okta 提供用於 AWS 管理主控台的應用程式時受到支援。

  如需詳細資訊，請參閱設定 JDBC 或 ODBC 單一登入身分驗證。

- Microsoft Azure Active Directory

  如需詳細資訊，請參閱設定 JDBC 或 ODBC 單一登入身分驗證。

外掛程式選項
外掛程式選項

若要使用以 SAML 為基礎的憑證供應商外掛程式，請使用 JDBC 或 ODBC 選項或在命名設定檔中指定下列選項。如果未指定 `plugin_name`，則會忽略其他選項。

選項	說明

`plugin_name`	針對 JDBC，為實作登入資料供應商的類別名稱。請指定下列其中一項：
	• 針對 Active Directory Federation Services
	`com.amazon.redshift.plugin.AdfsCredentialsProvider`
	• 針對 Okta
	`com.amazon.redshift.plugin.OktaCredentialsProvider`
	• 針對 PingFederate
	`com.amazon.redshift.plugin.PingCredentialsProvider`
	• 針對 Microsoft Azure Active Directory
	`com.amazon.redshift.plugin.AzureCredentialsProvider`
	• 針對 SAML MFA

選項	說明

	`com.amazon.redshift.plugin.BrowserSamlCredentialsProvider`
	• 針對 Microsoft Azure Active Directory 單一登入與 MFA
	`com.amazon.redshift.plugin.BrowserAzureCredentialsProvider`
	針對 ODBC，請指定下列其中一項：
	• 針對 Active Directory Federation Services：`adfs`
	• 針對 Okta：`okta`
	• 針對 PingFederate：`ping`
	• 針對 Microsoft Azure Active Directory：`azure`
	• 針對 SAML MFA：`browser saml`
	• 針對 Microsoft Azure Active Directory 單一登入與 MFA： `browser azure ad`

`idp_host`	企業身分提供商主機的名稱。該名稱不得包含任何斜線 (「/」)。針對 Okta 身分提供者，`idp_host` 的值應以 `.okta.com` 結尾。
`idp_port`	身分提供者使用的連接埠。預設為 443。Okta 會忽略此連接埠。
`preferred_role`	Amazon Resource Name (ARN) 角色，來自 SAML 聲明中 `Role` 屬性的 `AttributeValue` 元素。若要尋找偏好角色適用的值，請洽詢 IdP 管理員。如需詳細資訊，請參閱步驟 2：設定 IdP 的 SAML 聲明。
`user`	企業使用者名稱，包括網域 (如適用)。例如，在 Active Directory 中，網域名稱為必填，使用格式為網域\使用者名稱。
password	企業使用者的密碼。我們建議不要使用此選項。請改用 SQL 用戶端提供密碼。
`app_id`	Okta 應用程式的 ID。僅搭配 Okta 使用。Okta 應用程式嵌入連結中 `app_id` 後方的 `amazon_aws` 值。若要取得此值，請洽詢 IdP 管理員。以下為應用程式嵌入連結的範例：`https://example.okta.com/home/amazon_aws/0oa2hylwrpM8UGehd1t7/272`

選項	說明
idp_tenant	用於 Azure AD 的租用戶。僅搭配 Azure 使用。
client_id	Azure AD 中 Amazon Redshift 企業應用程式的用戶端 ID。僅搭配 Azure 使用。

使用 Amazon Redshift CLI 或 API 產生 IAM 身分的資料庫憑證

若要以程式設計方式產生臨時資料庫使用者憑證，Amazon Redshift 為 AWS Command Line Interface (AWS CLI) 和 GetClusterCredentials API 操作提供 get-cluster-credentials GetClusterCredentials 命令。或者，您可以使用 Amazon Redshift JDBC 或 ODBC 驅動程式來設定 SQL 用戶端，以管理呼叫 GetClusterCredentials 操作、擷取資料庫使用者憑證，以及在 SQL 用戶端和 Amazon Redshift 資料庫之間建立連接的程序。如需詳細資訊，請參閱用於建立資料庫使用者憑證的 JDBC 和 ODBC 選項。

> **Note**
>
> 我們建議使用 Amazon Redshift JDBC 或 ODBC 驅動程式產生資料庫使用者憑證。

您可在本節中找到程式設計的步驟，以用來呼叫 GetClusterCredentials 操作或 get-cluster-credentials 命令、擷取資料庫使用者登入資料，以及連線至資料庫。

產生及使用暫時性資料庫登入資料

1. 建立或修改具有必要許可的使用者或角色。如需 IAM 許可的相關資訊，請參閱 步驟 3：建立具有呼叫 GetClusterCredentialsWithIAM 或 GetClusterCredentials 許可的 IAM 角色 GetClusterCredentials。
2. 以您在先前步驟中授權的使用者或角色身分，執行 get-cluster-credentials CLI 命令或呼叫 GetClusterCredentials API 操作，然後提供下列值：
   - 叢集識別碼 — 包含資料庫之叢集的名稱。
   - 資料庫使用者名稱 — 現有或新資料庫使用者的名稱。
     - 如果資料庫中不存在使用者，且 AutoCreate 為 true，將建立停用 PASSWORD 的新使用者。
     - 如果使用者不存在，且 AutoCreate 為 false，則請求失敗。

    - 針對此範例，資料庫使用者名稱為 temp_creds_user。
  - 自動建立 — (選用) 如果資料庫使用者名稱不存在，則建立新使用者。
  - 資料庫名稱 — (選用) 使用者授權登入之資料庫的名稱。如果未指定資料庫名稱，使用者將可登入任何的叢集資料庫。
  - 資料庫群組 — (選用) 現有資料庫使用者群組的清單。成功登入後，資料庫使用者將新增至指定的使用者群組。如果未指定群組，使用者只會具有 PUBLIC 許可。使用者群組名稱必須符合附加至使用者或角色之 IAM 政策中指定的 dbgroup 資源 ARN。
  - 過期時間 — (選用) 暫時性憑證過期前經過的時間 (以秒計)。您可指定 900 秒 (15 分鐘) 至 3600 秒 (60 分鐘) 之間的值。預設為 900 秒。
3. Amazon Redshift 會確定使用者具備用指定資源呼叫 GetClusterCredentials 操作的許可。
4. Amazon Redshift 將傳回暫時性密碼和資料庫使用者名稱。

   下列範例使用 Amazon Redshift CLI 為名為 temp_creds_user 的現有使用者產生暫時性資料庫憑證。

   ```
   aws redshift get-cluster-credentials --cluster-identifier examplecluster --db-user
    temp_creds_user --db-name exampledb --duration-seconds 3600
   ```

   結果如下所示。

   ```
   {
     "DbUser": "IAM:temp_creds_user",
     "Expiration": "2016-12-08T21:12:53Z",
     "DbPassword": "EXAMPLEjArE3hcnQj8zt4XQj9Xtma8oxYEM8OyxpDHwXVPyJYBDm/
   gqX2Eeaq6P3DgTzgPg=="
   }
   ```

   下列範例使用 Amazon Redshift CLI 與 autocreate 為新使用者產生暫時性資料庫憑證，並將使用者加入到 example_group 群組。

   ```
   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
   ```

   結果如下所示。

   ```
   {
   ```

```
  "DbUser": "IAMA:temp_creds_user:example_group",
  "Expiration": "2016-12-08T21:12:53Z",
  "DbPassword": "EXAMPLEjArE3hcnQj8zt4XQj9Xtma8oxYEM8OyxpDHwXVPyJYBDm/
gqX2Eeaq6P3DgTzgPg=="
}
```

5. 建立與 Amazon Redshift 叢集的 Secure Socket Layer (SSL) 身分驗證連線，並用來自 `GetClusterCredentials` 回應中的使用者名稱和密碼傳送登入請求。請在使用者名稱中加入 `IAM:` 或 `IAMA:` 字首，例如，`IAM:temp_creds_user` 或 `IAMA:temp_creds_user`。

   > **Important**
   >
   > 設定 SQL 用戶端要求 SSL。否則，如果您的 SQL 用戶端自動嘗試使用 SSL 連線，發生任何類型的故障時其將切換為非 SSL。在此情況下，第一個連線嘗試可能會因為登入資料過期或失效而失敗，而第二個連線嘗試可能會因連線不是 SSL 而失敗。此時，可能會跳過第一個錯誤訊息。如需使用 SSL 連接至叢集的相關資訊，請參閱 設定連線的安全選項。

6. 如果連線不使用 SSL，連線嘗試將失敗。
7. 叢集會傳送 `authentication` 請求至 SQL 用戶端。
8. 接著 SQL 用戶端會傳送暫時性密碼至叢集。
9. 如果密碼為有效且尚未過期，叢集將完成連線。

設定 JDBC 或 ODBC 單一登入身分驗證

您可以利用外部身分提供者 (IdP) 來驗證使用者身分，並授權使用者存取 Amazon Redshift 叢集，以簡化使用者管理和增強安全性。如此即可在多項服務之間集中管理使用者、啟用角色型存取控制和稽核功能。常見的使用案例包括簡化各種使用者群組的身分驗證、強制執行一致的存取政策，以及符合監管要求。

以下頁面將引導您設定與 Redshift 叢集的 IdP 整合。如需將 AWS 設定為 IdP 服務提供者的詳細資訊，請參閱《IAM 使用者指南》中的使用依賴方信任設定 SAML 2.0 IdP 和新增宣告。

AD FS

本教學課程說明如何使用 AD FS 作為身分提供者 (IdP) 來存取您的 Amazon Redshift 叢集。

步驟 1：設定 AD FS 和 AWS 您的帳戶彼此信任

下列程序說明如何設定信任關係。

1. 建立或使用現有的 Amazon Redshift 叢集，以供 AD FS 使用者連線。若要設定連線，則需要此叢集的特定屬性，例如叢集識別碼。如需詳細資訊，請參閱建立叢集。
2. 設定 AD FS 以控制 Microsoft 管理主控台上的 Amazon Redshift 存取：
    1. 選擇 ADFS 2.0，然後選擇 Add Relying Party Trust (新增信賴方信任)。在 Add Relying Party Trust Wizard (新增信賴方信任精靈) 頁面上，選擇 Start (開始)。
    2. 在 Select Data Source (選取資料來源) 頁面上，選擇 Import data about the relying party published online or on a local network (匯入關於在線上或本機網路上發佈信賴方的資料)。
    3. 針對 Federation metadata address (host name or URL) (聯合中繼資料地址 (主機名稱或 URL))，輸入 **`https://signin.aws.amazon.com/saml-metadata.xml`**。中繼資料 XML 檔案是標準 SAML 中繼資料文件，描述 AWS 為依賴方。
    4. 在 Specify Display Name (指定顯示名稱) 頁面上，輸入 Display name (顯示名稱) 的值。
    5. 在 Choose Issuance Authorization Rules (選擇發行授權規則) 頁面上，選擇允許或拒絕所有使用者存取此信賴方的發行授權規則。
    6. 在 Ready to Add Trust (準備新增信任) 頁面上檢閱您的設定。
    7. 在 Finish (完成) 頁面上，選擇 Open the Edit Claim Rules dialog for this relying party trust when the wizard closes (當精靈關閉時，開啟此信賴方信任的「編輯宣告規則」對話方塊)。
    8. 在內容 (按一下滑鼠右鍵) 功能表上，選擇 Relying Party Trusts (依賴方信任)。
    9. 針對您的信賴方，開啟內容 (按一下滑鼠右鍵) 功能表，然後選擇 Edit Claim Rules (編輯宣告規則)。在 Edit Claim Rules (編輯宣告規則) 頁面上，選擇 Add Rule (新增規則)。
    10. 在 Claim rule template (宣告規則範本) 中，選擇 Transform an Incoming Claim (轉換傳入宣告)，然後在 Edit Rule – NameId (編輯規則 – NameId) 頁面上執行下列動作：
        - 針對 Claim rule name (宣告規則名稱)，輸入 NameId。
        - 針對 Incoming claim name (傳入宣告名稱)，選擇 Windows Account Name (Windows 帳戶名稱)。
        - 針對 Outgoing claim name (傳出宣告名稱)，選擇 Name ID (名稱 ID)。
        - 針對 Outgoing name ID format (傳出名稱 ID 格式)，選擇 Persistent Identifier (持久性標識符)。
        - 選擇 Pass through all claim values (傳遞所有宣告值)。
    11. 在 Edit Claim Rules (編輯宣告規則) 頁面上，選擇 Add Rule (新增規則)。在 Select Rule Template (選取規則範本) 頁面上，針對 Claim rule template (宣告規則範本)，選擇 Send LDAP Attributes as Claims (將 LDAP 屬性傳送為宣告)。
    12. 在 Configure Rule (設定規則) 頁面上，執行下列作業：

- 針對 Claim rule name (宣告規則名稱)，輸入 RoleSessionName。
- 針對 Attribute store (屬性存放區)，選擇 Active Directory。
- 針對 LDAP Attribute (LDAP 屬性)，選擇 Email Addresses (電子郵件地址)。
- 針對 傳出宣告類型，選擇 https://aws.amazon.com/SAML/Attributes/RoleSessionName。

13. 在 Edit Claim Rules (編輯宣告規則) 頁面上，選擇 Add Rule (新增規則)。在 Select Rule Template (選取規則範本) 頁面上，針對 Claim rule template (宣告規則範本)，選擇 Send Claims Using a Custom Rule (使用自訂規則傳送宣告)。
14. 在 Edit Rule – Get AD Groups (編輯規則 – 取得 AD 群組) 頁面上，針對 Claim rule name (宣告規則名稱)，輸入 Get AD Groups (取得 AD 群組)。
15. 針對 Custom rule (自訂規則)，輸入下列內容。

```
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 Edit Claim Rules (編輯宣告規則) 頁面上，選擇 Add Rule (新增規則)。在 Select Rule Template (選取規則範本) 頁面上，針對 Claim rule template (宣告規則範本)，選擇 Send Claims Using a Custom Rule (使用自訂規則傳送宣告)。
17. 在 Edit Rule – Roles (編輯規則 – 角色) 頁面上，針對 Claim rule name (宣告規則名稱)，輸入 Roles (角色)。
18. 針對 Custom rule (自訂規則)，輸入下列內容。

```
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-"));
```

請記下 SAML 供應商的 ARN，以及要扮演的角色。在此範例中，`arn:aws:iam:123456789012:saml-provider/ADFS` 是 SAML 供應商的 ARN，`arn:aws:iam:123456789012:role/ADFS-` 是角色的 ARN。

3. 請確定您已下載 `federationmetadata.xml` 檔案。檢查文件內容沒有無效的字元。這是您在設定與 AWS的信任關係時，使用的中繼資料檔案。

4. 在 IAM 主控台上建立 IAM SAML 身分提供者。您所提供的中繼資料文件，是您在設定 Azure 企業應用程式時儲存的聯合中繼資料 XML 檔案。如需詳細步驟，請參閱《IAM 使用者指南》中的建立和管理 IAM 身分提供者 (主控台)。
5. 在 IAM 主控台上建立 SAML 2.0 聯合的 IAM 角色。如需詳細步驟，請參閱《IAM 使用者指南》中的為 SAML 建立角色。
6. 建立 IAM 政策，您可將其附加至您在 IAM 主控台上為 SAML 2.0 聯合建立的 IAM 角色。如需詳細步驟，請參閱《IAM 使用者指南》中的建立 IAM 政策 (主控台)。如需 Azure AD 範例，請參閱設定 JDBC 或 ODBC 單一登入身分驗證。

步驟 2：設定 JDBC 或 ODBC 以對 AD FS 進行身分驗證

JDBC

下列程序說明如何設定 JDBC 與 AD FS 的關係。

- 將資料庫用戶端設定為使用 AD FS 單一登入，透過 JDBC 連線到您的叢集。

  您可使用任何採用 JDBC 驅動程式的用戶端使用 AD FS 單一登入進行連線，或使用 Java 之類的語言透過指令碼進行連線。如需安裝和組態資訊，請參閱 設定適用於 Amazon Redshift 的 JDBC 驅動器 2.x 版的連線。

  例如，您可以使用 SQLWorkbench/J 作為用戶端。當您設定 SQLWorkbench /J 時，您的資料庫 URL 會使用下列格式。

  ```
  jdbc:redshift:iam://cluster-identifier:us-west-1/dev
  ```

  如果您使用 SQLWorkbench /J 作為用戶端，請執行下列步驟：

  a. 啟動 SQL Workbench/J。在選取連線設定檔頁面中，新增設定檔群組，例如 **ADFS**。
  b. 針對 Connection Profile (連線設定檔)，輸入您的連線設定檔名稱，例如 **ADFS**。
  c. 選擇 Manage Drivers (管理驅動程式)，然後選擇 Amazon Redshift。選擇程式庫旁邊的開啟資料夾圖示，然後選擇適當的 JDBC .jar 檔案。
  d. 在 Select Connection Profile (選取連線設定檔) 頁面上，將資訊新增至連線設定檔，如下所示：

     - 針對 User (使用者)，輸入您的 AD FS 使用者名稱。這是您用於單一登入之 帳戶的使用者名稱，該帳戶具有您嘗試驗證之叢集的許可。

- 針對 Password (密碼)，輸入您的 AD FS 密碼。
- 針對 Drivers (驅動程式)，選擇 Amazon Redshift (com.amazon.redshift.jdbc.Driver)。
- 針對 URL，輸入 **jdbc:redshift:iam://*your-cluster-identifier*:*your-cluster-region*/*your-database-name***。

e. 選擇延伸屬性。針對 plugin_name，輸入 **com.amazon.redshift.plugin.AdfsCredentialsProvider**。此值會指定驅動程式使用 AD FS 單一登入做為身分驗證方法。

ODBC

設定 ODBC 以對 AD FS 進行身分驗證

- 將資料庫用戶端設定為使用 AD FS 單一登入，透過 ODBC 連線到您的叢集。

  Amazon Redshift 會提供適用於 Linux、Windows 和 macOS 作業系統的 ODBC 驅動程式。安裝 ODBC 驅動程式之前，查明 SQL 用戶端工具是 32 位元或 64 位元。安裝符合 SQL 用戶端工具需求的 ODBC 驅動程式。

  在 Windows 上，於 Amazon Redshift ODBC Driver DSN Setup (Amazon Redshift ODBC 驅動程式 DSN 設定) 頁面的 Connection Settings (連線設定) 之下，輸入下列資訊：

  - 針對 Data source name (資料來源名稱)，輸入 ***your-DSN***。這會指定作為 ODBC 設定檔名稱的資料來源名稱。
  - 針對驗證類型，選擇身分提供者：SAML。這是 ODBC 驅動程式使用 AD FS 單一登入來進行驗證的身分驗證方法。
  - 針對 Cluster ID (叢集 ID)，輸入 ***your-cluster-identifier***。
  - 針對 Region (區域)，輸入 ***your-cluster-region***。
  - 針對 Database (資料庫)，輸入 ***your-database-name***。
  - 針對 User (使用者)，輸入 ***your-adfs-username***。這是您用於單一登入之 AD FS 帳戶的使用者名稱，該帳戶具有您嘗試驗證之叢集的許可。僅在 Auth Type (驗證類型) 為Identity Provider: SAML (身分提供者：SAML) 的情況下使用此項目。
  - 針對 Password (密碼)，輸入 ***your-adfs-password***。僅在 Auth Type (驗證類型) 為Identity Provider: SAML (身分提供者：SAML) 的情況下使用此項目。

  在 macOS 和 Linux 上，編輯 `odbc.ini` 檔案，如下所示：

> **Note**
>
> 所有項目都不區分大小寫。

- 針對 clusterid，輸入 ***your-cluster-identifier***。這是所建立 Amazon Redshift 叢集的名稱。
- 針對 region (區域)，輸入 ***your-cluster-region***。這是所建立 Amazon Redshift 叢集 AWS 的區域。
- 針對 database (資料庫)，輸入 ***your-database-name***。這是您嘗試在 Amazon Redshift 叢集上存取的資料庫名稱。
- 針對 locale，輸入 **en-us**。這是錯誤訊息的顯示語言。
- 針對 iam，輸入 **1**。此值指定驅動程式使用 IAM 登入資料進行身分驗證。
- 針對 plugin_name，執行下列其中一項作業：
  - 若為具有 MFA 的 AD FS 單一登入組態，請輸入 **BrowserSAML**。這是 ODBC 驅動程式用來驗證 AD FS 的身分驗證方法。
  - 若為 AD FS 單一登入組態，請輸入 **ADFS**。這是 ODBC 驅動程式使用 Azure AD 單一登入來進行驗證的身分驗證方法。
- 針對 uid，輸入 ***your-adfs-username***。這是您用於單一登入之 Microsoft Azure 帳戶的使用者名稱，該帳戶具有您嘗試驗證之叢集的許可。僅在 plugin_name 為 ADFS 的情況下使用此項目。
- 針對 pwd，輸入 ***your-adfs-password***。僅在 plugin_name 為 ADFS 的情況下使用此項目。

在 macOS 和 Linux 上，同時編輯設定檔設定以新增下列匯出：

```
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 作為身分提供者 (IdP) 來存取您的 Amazon Redshift 叢集。本教學課程說明如何使用 Azure 作為身分提供者 (IdP) 來存取您的 Amazon Redshift 叢集。

觀看以下影片，了解如何搭配 Microsoft Azure AD 單一登入來聯合 Amazon Redshift 存取：搭配 Microsoft Azure AD 單一登入來聯合 Amazon Redshift 存取。

步驟 1：設定 Azure 和 AWS 您的帳戶彼此信任

下列程序說明如何設定信任關係。

設定 Azure AD 和 AWS 您的帳戶彼此信任

1. 建立或使用現有的 Amazon Redshift 叢集，以供 Azure AD 使用者連線。若要設定連線，則需要此叢集的特定屬性，例如叢集識別碼。如需詳細資訊，請參閱建立叢集。
2. 在 Microsoft Azure 入口網站 AWS 上設定用於 的 Azure Active Directory、群組和使用者。
3. 在 Microsoft Azure 入口網站上將 Amazon Redshift 新增為企業應用程式，以用於 AWS 主控台的單一登入和 Amazon Redshift 的聯合登入。選擇 Enterprise application (企業應用程式)。
4. 選擇 +New application (新增應用程式)。新增應用程式頁面隨即顯示。
5. 在搜尋欄位中搜尋 **AWS**。
6. 選擇 Amazon Web Services (AWS)，然後選擇新增。此操作會建立 AWS 應用程式。
7. 在 Manage (管理) 下方，選擇 Single sign-on (單一登入)。
8. 選擇 SAML。Amazon Web Services (AWS) | 以 SAML 為基礎的登入頁面隨即顯示。
9. 選擇 Yes (是)，繼續進行「使用 SAML 頁面設定單一登入」。此頁面會顯示預先設定的單一登入相關屬性清單。
10. 在 Basic SAML Configuration (基本 SAML 組態) 中，選擇編輯圖示並選擇 Save (儲存)。
11. 設定超過一個應用程式時，請提供識別碼值。例如，輸入 ***https://signin.aws.amazon.com/saml#2***。請注意，從第二個應用程式開始，請使用此格式搭配 # 符號來指定唯一的 SPN 值。
12. 在 User Attributes and Claims (使用者屬性和宣告) 區段中選擇編輯圖示。

    根據預設，唯一使用者標識符 (UID)、角色、RoleSessionName 和 SessionDuration 宣告都是預先設定的。
13. 選擇 + Add new claim (新增宣告) 來新增資料庫使用者的宣告。

    對於名稱，輸入 **DbUser**。

針對 Namespace (命名空間)，輸入 **`https://redshift.amazon.com/SAML/Attributes`**。

針對 Source (來源)，選擇 Attribute (屬性)。

針對 Source attribute (來源屬性)，選擇 user.userprincipalname。然後選擇 Save (儲存)。

14. 選擇 + Add new claim (新增宣告) 來新增 AutoCreate 的宣告。

對於名稱，輸入 **`AutoCreate`**。

針對 Namespace (命名空間)，輸入 **`https://redshift.amazon.com/SAML/Attributes`**。

針對 Source (來源)，選擇 Attribute (屬性)。

針對 Source attribute (來源屬性)，選擇 “true”。然後選擇 Save (儲存)。

在此，*`123456789012`* 是您的 AWS 帳戶，*`AzureSSO`* 是您建立的 IAM 角色，而 *`AzureADProvider`* 是 IAM 提供者。

宣告名稱	Value
唯一使用者識別碼 (名稱 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

宣告名稱	Value
https://redshift.amazon.com/SAML/Attributes/DbUser	user.userprincipalname

15. 在 App Registration (應用程式註冊) > ***your-application-name*** > Authentication (身分驗證) 下方，新增 Mobile And Desktop Application (行動和桌面應用程式)。將 URL 指定為 http://localhost/redshift。
16. 在 SAML Signing Certificate (SAML 簽署憑證) 區段中，選擇 Download (下載) 以下載並儲存聯合中繼資料 XML 檔案，以便在建立 IAM SAML 身分提供者時使用。此檔案用於建立單一登入聯合身分。
17. 在 IAM 主控台上建立 IAM SAML 身分提供者。您所提供的中繼資料文件，是您在設定 Azure 企業應用程式時儲存的聯合中繼資料 XML 檔案。如需詳細步驟，請參閱《IAM 使用者指南》中的建立和管理 IAM 身分提供者 (主控台)。
18. 在 IAM 主控台上建立 SAML 2.0 聯合的 IAM 角色。如需詳細步驟，請參閱《IAM 使用者指南》中的為 SAML 建立角色。
19. 建立 IAM 政策，您可將其附加至您在 IAM 主控台上為 SAML 2.0 聯合建立的 IAM 角色。如需詳細步驟，請參閱《IAM 使用者指南》中的建立 IAM 政策 (主控台)。

    為您的環境修改下列政策 (JSON 格式)：

    - 將叢集 AWS 的區域替換為 *us-west-1*。
    - 將 AWS 您的帳戶替換為 *123456789012*。
    - 以您的叢集識別碼 (或 * 代表所有叢集) 代替 *cluster-identifier*。
    - 以您的資料庫 (或 * 代表所有資料庫) 代替 *dev*。
    - 以 IAM 角色的唯一識別碼代替 *AROAJ2UCCR6DPCEXAMPLE*。
    - 以您的租用戶或公司電子郵件網域代替 *example.com*。
    - 以您計劃為其指派使用者的資料庫群組代替 *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": "*"
        }
    ]
}
```

此政策會授與許可，如下所示：

- 第一個區段會授與 GetClusterCredentials API 操作的許可，以取得所指定叢集的暫時性登入資料。在此範例中，資源是 *cluster-identifier*，其具有資料庫 *dev*、位於帳戶 *123456789012*、並位於 AWS 區域 *us-west-1*。${redshift:DbUser} 子句只允許符合 Azure AD 中指定 DbUser 值的使用者連線。

- 條件子句會強制只有特定使用者會取得臨時登入資料。這些是依據貴公司電子郵件網域中的電子郵件地址所識別的 IAM 帳戶中，具備角色唯一 ID *AROAJ2UCCR6DPCEXAMPLE* 所指定角色的使用者。如需唯一 ID 的相關資訊，請參閱《IAM 使用者指南》中的唯一 ID。

  您所設定的 IdP (在此案例中為 Azure AD) 會決定條件子句的撰寫方式。如果您的員工電子郵件是 `johndoe@example.com`，請先將 `${redshift:DbUser}` 設定為符合員工使用者名稱 `johndoe` 的超級欄位。然後將 AWS SAML `RoleSessionName` 欄位設定為符合員工電子郵件 `johndoe@example.com` 的超級欄位，讓此條件發揮作用。當您採取這種方法時，請考慮下列事項：

  - 如果您將 `${redshift:DbUser}` 設定為員工的電子郵件，然後移除範例 JSON 中符合 `RoleSessionName` 的 `@example.com`。
  - 如果您將 `RoleSessionId` 設定為員工的使用者名稱，然後移除範例中符合 `RoleSessionName` 的 `@example.com`。
  - 在範例 JSON 中，`${redshift:DbUser}` 和 `RoleSessionName` 均設定為員工的電子郵件。此範例 JSON 會使用 Amazon Redshift 資料庫使用者名稱搭配 `@example.com`，讓使用者登入以存取叢集。
- 第二個區段會授與許可，以在指定的叢集中建立 `dbuser` 名稱。在此範例 JSON 中，其會將建立限制於 `${redshift:DbUser}`。
- 第三個區段會授與許可，以指定使用者可以加入的 `dbgroup`。在此範例 JSON 中，使用者可以加入指定叢集中的 `my_dbgroup` 群組。
- 第四個區段會授予使用者可對所有資源執行動作的許可。在此範例 JSON 中，它允許使用者呼叫 `redshift:DescribeClusters` 以取得叢集資訊，例如叢集端點、 AWS 區域和連接埠。此外，還可允許使用者呼叫 `iam:ListRoles`，以檢查使用者可以擔任哪些角色。

步驟 2：設定 JDBC 或 ODBC 以對 Azure 進行身分驗證

JDBC

設定 JDBC 以對 Microsoft Azure AD 進行身分驗證

- 將資料庫用戶端設定為使用 Azure AD 單一登入，透過 JDBC 連線到您的叢集。

  您可使用任何採用 JDBC 驅動程式的用戶端使用 Azure AD 單一登入進行連線，或使用 Java 之類的語言透過指令碼進行連線。如需安裝和組態資訊，請參閱 設定適用於 Amazon Redshift 的 JDBC 驅動器 2.x 版的連線。

例如，您可以使用 SQLWorkbench/J 作為用戶端。當您設定 SQLWorkbench /J 時，您的資料庫 URL 會使用下列格式。

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

如果您使用 SQLWorkbench /J 作為用戶端，請執行下列步驟：

a. 啟動 SQL Workbench/J。在選取連線設定檔頁面上，新增名為 **AzureAuth** 的設定檔群組。

b. 針對 Connection Profile (連線設定檔)，輸入 **Azure**。

c. 選擇 Manage Drivers (管理驅動程式)，然後選擇 Amazon Redshift。選擇程式庫旁邊的開啟資料夾圖示，然後選擇適當的 JDBC .jar 檔案。

d. 在 Select Connection Profile (選取連線設定檔) 頁面上，將資訊新增至連線設定檔，如下所示：

- 針對 User (使用者)，輸入您的 Microsoft Azure 使用者名稱。這是您用於單一登入之 Microsoft Azure 帳戶的使用者名稱，該帳戶具有您嘗試驗證之叢集的許可。
- 針對 Password (密碼)，輸入您的 Microsoft Azure 密碼。
- 針對 Drivers (驅動程式)，選擇 Amazon Redshift (com.amazon.redshift.jdbc.Driver)。
- 針對 URL，輸入 **jdbc:redshift:iam://*your-cluster-identifier*:*your-cluster-region*/*your-database-name***。

e. 選擇 Extended Properties (擴充屬性) 以將其他資訊新增至連線屬性，如下所述。

針對 Azure AD 單一登入組態新增其他資訊，如下所示：

- 針對 plugin_name，輸入 **com.amazon.redshift.plugin.AzureCredentialsProvider**。此值會指定驅動程式使用 Azure AD 單一登入做為身分驗證方法。
- 針對 idp_tenant，輸入 ***your-idp-tenant***。僅用於 Microsoft Azure AD。這是您在 Azure AD 上設定的公司租用戶名稱。此值可以是租用戶名稱或租用戶唯一 ID (含連字號)。
- 針對 client_secret，輸入 ***your-azure-redshift-application-client-secret***。僅用於 Microsoft Azure AD。這是您在設定 Azure 單一登入組態時所建立之 Amazon Redshift 應用程式的用戶端秘密。這只適用於 com.amazon.redshift.plugin.AzureCredentialsProvider 外掛程式。

- 針對 client_id，輸入 ***your-azure-redshift-application-client-id***。僅用於 Microsoft Azure AD。這是您在設定 Azure 單一登入組態時所建立之 Amazon Redshift 應用程式的用戶端 ID (含連字號)。

針對具有 MFA 的 Azure AD 單一登入組態，將其他資訊新增至連線屬性，如下所示：

- 針對 plugin_name，輸入 **com.amazon.redshift.plugin.BrowserAzureCredentialsProvider**。此值會指定驅動程式使用具有 MFA 的 Azure AD 單一登入做為身分驗證方法。
- 針對 idp_tenant，輸入 ***your-idp-tenant***。僅用於 Microsoft Azure AD。這是您在 Azure AD 上設定的公司租用戶名稱。此值可以是租用戶名稱或租用戶唯一 ID (含連字號)。
- 針對 client_id，輸入 ***your-azure-redshift-application-client-id***。此選項僅用於 Microsoft Azure AD。這是您在設定具有 MFA 的 Azure AD 單一登入組態時所建立之 Amazon Redshift 應用程式的用戶端 ID (含連字號)。
- 針對 listen_port，輸入 ***your-listen-port***。這是本機伺服器接聽的連接埠。預設值為 7890。
- 針對 idp_response_timeout，輸入 ***the-number-of-seconds***。這是 IdP 伺服器傳回回應時，在逾時之前等待的秒數。秒數的下限必須為 10。如果建立連線所花的時間超過此臨界值，連線會中止。

ODBC

設定 ODBC 以對 Microsoft Azure AD 進行身分驗證

- 將資料庫用戶端設定為使用 Azure AD 單一登入，透過 ODBC 連線到您的叢集。

  Amazon Redshift 會提供適用於 Linux、Windows 和 macOS 作業系統的 ODBC 驅動程式。安裝 ODBC 驅動程式之前，查明 SQL 用戶端工具是 32 位元或 64 位元。安裝符合 SQL 用戶端工具需求的 ODBC 驅動程式。

  在 Windows 上，於 Amazon Redshift ODBC Driver DSN Setup (Amazon Redshift ODBC 驅動程式 DSN 設定) 頁面的 Connection Settings (連線設定) 之下，輸入下列資訊：

  - 針對 Data source name (資料來源名稱)，輸入 ***your-DSN***。這會指定作為 ODBC 設定檔名稱的資料來源名稱。

- 針對 Azure AD 單一登入組態的驗證類型，請選擇 **Identity Provider: Azure AD**。這是 ODBC 驅動程式使用 Azure 單一登入來進行驗證的身分驗證方法。
- 針對具有 MFA 的 Azure AD 單一登入組態的驗證類型，請選擇 **Identity Provider: Browser Azure AD**。這是 ODBC 驅動程式使用具有 MFA 之 Azure 單一登入來進行驗證的身分驗證方法。
- 針對 Cluster ID (叢集 ID)，輸入 ***your-cluster-identifier***。
- 針對 Region (區域)，輸入 ***your-cluster-region***。
- 針對 Database (資料庫)，輸入 ***your-database-name***。
- 針對 User (使用者)，輸入 ***your-azure-username***。這是您用於單一登入之 Microsoft Azure 帳戶的使用者名稱，該帳戶具有您嘗試驗證之叢集的許可。僅在 Auth Type (驗證類型) 為Identity Provider: Azure AD (身分提供者：Azure AD) 的情況下使用此項目。
- 針對 Password (密碼)，輸入 ***your-azure-password***。僅在 Auth Type (驗證類型) 為Identity Provider: Azure AD (身分提供者：Azure AD) 的情況下使用此項目。
- 針對 IdP Tenant (IdP 租用戶)，輸入 ***your-idp-tenant***。這是您在 IdP (Azure) 上設定的公司租用戶名稱。此值可以是租用戶名稱或租用戶唯一 ID (含連字號)。
- 針對 Azure Client Secret (Azure 用戶端密碼)，輸入 ***your-azure-redshift-application-client-secret***。這是您在設定 Azure 單一登入組態時所建立之 Amazon Redshift 應用程式的用戶端秘密。
- 針對 Azure Client ID (Azure 用戶端 ID)，輸入 ***your-azure-redshift-application-client-id***。這是您在設定 Azure 單一登入組態時所建立之 Amazon Redshift 應用程式的用戶端 ID (含連字號)。
- 針對 Listen Port (接聽連接埠)，輸入 ***your-listen-port***。這是本機伺服器接聽的預設接聽連接埠。預設值為 7890。這僅適用於瀏覽器 Azure AD 外掛程式。
- 針對 Response Timeout (回應逾時)，輸入 ***the-number-of-seconds***。這是 IdP 伺服器傳回回應時，在逾時之前等待的秒數。秒數的下限必須為 10。如果建立連線所花的時間超過此臨界值，連線會中止。此選項僅適用於瀏覽器 Azure AD 外掛程式。

在 macOS 和 Linux 上，編輯 `odbc.ini` 檔案，如下所示：

> **Note**
>
> 所有項目都不區分大小寫。

- 針對 clusterid，輸入 ***your-cluster-identifier***。這是所建立 Amazon Redshift 叢集的名稱。
- 針對 region (區域)，輸入 ***your-cluster-region***。這是所建立 Amazon Redshift 叢集 AWS 的區域。
- 針對 database (資料庫)，輸入 ***your-database-name***。這是您嘗試在 Amazon Redshift 叢集上存取的資料庫名稱。
- 針對 locale，輸入 **en-us**。這是錯誤訊息的顯示語言。
- 針對 iam，輸入 **1**。此值指定驅動程式使用 IAM 登入資料進行身分驗證。
- 如需 Azure AD 單一登入組態的 plugin_name，請輸入 **AzureAD**。這會指定驅動程式使用 Azure 單一登入作為身分驗證方法。
- 如需具有 MFA 的 Azure AD 單一登入組態的 plugin_name，請輸入 **BrowserAzureAD**。這會指定驅動程式使用具有 MFA 的 Azure 單一登入做為身分驗證方法。
- 針對 uid，輸入 ***your-azure-username***。這是您用於單一登入之 Microsoft Azure 帳戶的使用者名稱，該帳戶具有您嘗試驗證之叢集的許可。僅在 plugin_name 為 AzureAD 的情況下使用此項目。
- 針對 pwd，輸入 ***your-azure-password***。僅在 plugin_name 為 AzureAD 的情況下使用此項目。
- 針對 idp_tenant，輸入 ***your-idp-tenant***。這是您在 IdP (Azure) 上設定的公司租用戶名稱。此值可以是租用戶名稱或租用戶唯一 ID (含連字號)。
- 針對 client_secret，輸入 ***your-azure-redshift-application-client-secret***。這是您在設定 Azure 單一登入組態時所建立之 Amazon Redshift 應用程式的用戶端秘密。
- 針對 client_id，輸入 ***your-azure-redshift-application-client-id***。這是您在設定 Azure 單一登入組態時所建立之 Amazon Redshift 應用程式的用戶端 ID (含連字號)。
- 針對 listen_port，輸入 ***your-listen-port***。這是本機伺服器接聽的連接埠。預設值為 7890。這適用於瀏覽器 Azure AD 外掛程式。
- 針對 idp_response_timeout，輸入 ***the-number-of-seconds***。這是等待 Azure 回應的指定時間期間 (秒)。此選項適用於瀏覽器 Azure AD 外掛程式。

在 macOS 和 Linux 上，同時編輯設定檔設定以新增下列匯出：

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

疑難排解

若要對瀏覽器 Azure AD 外掛程式的問題進行故障診斷，請考慮下列事項：

- 若要使用瀏覽器 Azure AD 外掛程式，您必須設定請求中指定的回覆 URL，以符合為您的應用程式設定的回覆 URL。瀏覽至 Microsoft Azure 入口網站上的 Set up Single Sign-On with SAML (使用 SAML 設定單一登入) 頁面。然後檢查回覆網址 是否設定為 http://localhost/redshift/。
- 如果您收到 IdP 租用戶錯誤，請確認 IdP 租用戶名稱與您最初用來設定 Microsoft Azure 中使用中目錄的網域名稱相符。

  在 Windows 上，導覽至 Amazon Redshift ODBC DSN 設定頁面的連線設定區段。接著檢查您在 IdP (Azure) 上設定的公司租用戶名稱是否與您最初用來在 Microsoft Azure 中設定使用中目錄的網域名稱相符。

  在 macOS 和 Linux 上，找到這個 odbc.ini 檔案。接著檢查您在 IdP (Azure) 上設定的公司租用戶名稱是否與您最初用來在 Microsoft Azure 中設定使用中目錄的網域名稱相符。
- 如果您收到請求中指定的回覆 URL 與為應用程式設定的回覆 URL 不相符的錯誤，請確認重新導向 URI 與回覆 URL 相同。

  瀏覽至 Microsoft Azure 入口網站上應用程式的 App registration (應用程式註冊) 頁面。然後檢查重新導向 URI 與回覆 URL 是否相符。
- 如果您收到未經授權的回應：未經授權的錯誤，請確認您已完成行動裝置和桌面應用程式組態。

  瀏覽至 Microsoft Azure 入口網站上應用程式的 App registration (應用程式註冊) 頁面。然後瀏覽至 Authentication (驗證)，並檢查您已將行動裝置和桌面應用程式設定為使用 http://localhost/redshift/ 做為重新導向 URI。

Ping 身分

您可以使用 Ping 身分做為身分提供者 (IdP) 來存取您的 Amazon Redshift 叢集。本教學課程說明如何使用 Ping Identity 作為身分提供者 (IdP) 來存取您的 Amazon Redshift 叢集。

步驟 1：設定 Ping Identity 和 AWS 您的帳戶彼此信任

下列程序說明如何使用 PingOne 入口網站設定信任關係。

設定 Ping Identity 和 AWS 您的帳戶彼此信任

1. 建立或使用現有的 Amazon Redshift 叢集，以供 Ping Identity 使用者連線。若要設定連線，則需要此叢集的特定屬性，例如叢集識別碼。如需詳細資訊，請參閱建立叢集。
2. 在 PingOne 入口網站上將 Amazon Redshift 新增為新的 SAML 應用程式。如需詳細步驟，請參閱 Ping 身分文件。
    1. 前往 My Applications (我的應用程式)。
    2. 在 Add Application (新增應用程式) 下方，選擇 New SAML Application (新增 SAML 應用程式)。
    3. 針對 Application Name (應用程式名稱)，輸入 **`Amazon Redshift`**。
    4. 針對 Protocol Version (通訊協定版本)，選擇 SAML v2.0。
    5. 針對 Category (類別)，選擇 ***`your-application-category`***。
    6. 針對 Assertion Consumer Service (ACS) (聲明消費者服務 (ACS))，輸入 ***`your-redshift-local-host-url`***。這是 SAML 聲明會重新導向至的本機主機與連接埠。
    7. 針對 Entity ID (實體 ID)，輸入 `urn:amazon:webservices`。
    8. 針對 Signing (簽署)，選擇 Sign Assertion (簽署聲明)。
    9. 在 SSO Attribute Mapping (SSO 屬性映射) 區段中，建立如下表所示的宣告。

應用程式屬性	常值的身分橋接屬性
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」屬性中的群組包含 @directory 字首。若要移除此項目，請在 Identity

應用程式屬性	常值的身分橋接屬性
	Bridge 中輸入 memberOf。在函數中，選擇 ExtractByRegularExpression。在運算式中，輸入 (.*)[\@](?:.*)。

3. 針對 Group Access (群組存取)，視需要設定下列群組存取：
   - 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. 檢閱您的設定並進行變更 (如有必要)。
5. 使用 Initiate Single Sign-On (SSO) URL (啟動單一登入 (SSO) URL) 做為瀏覽器 SAML 外掛程式的登入 URL。
6. 在 IAM 主控台上建立 IAM SAML 身分提供者。您所提供的中繼資料文件，是您在設定 Ping 身分時儲存的聯合中繼資料 XML 檔案。如需詳細步驟，請參閱《IAM 使用者指南》中的建立和管理 IAM 身分提供者 (主控台)。
7. 在 IAM 主控台上建立 SAML 2.0 聯合的 IAM 角色。如需詳細步驟，請參閱《IAM 使用者指南》中的為 SAML 建立角色。
8. 建立 IAM 政策，您可將其附加至您在 IAM 主控台上為 SAML 2.0 聯合建立的 IAM 角色。如需詳細步驟，請參閱《IAM 使用者指南》中的建立 IAM 政策 (主控台)。如需 Azure AD 範例，請參閱設定 JDBC 或 ODBC 單一登入身分驗證。

步驟 2：設定 JDBC 或 ODBC 以對 Ping Identity 進行身分驗證

JDBC

設定 JDBC 以對 Ping 身分進行驗證

- 將資料庫用戶端設定為使用 Ping Identity 單一登入，透過 JDBC 連線到您的叢集。

  您可使用任何採用 JDBC 驅動程式的用戶端使用 Ping Identity 單一登入進行連線，或使用 Java 之類的語言透過指令碼進行連線。如需安裝和組態資訊，請參閱 設定適用於 Amazon Redshift 的 JDBC 驅動器 2.x 版的連線。

例如，您可以使用 SQLWorkbench/J 作為用戶端。當您設定 SQLWorkbench /J 時，您的資料庫 URL 會使用下列格式。

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

如果您使用 SQLWorkbench /J 作為用戶端，請執行下列步驟：

a. 啟動 SQL Workbench/J。在選取連線設定檔頁面中，新增設定檔群組，例如 **Ping**。

b. 針對 Connection Profile (連線設定檔)，輸入 ***your-connection-profile-name***，例如 **Ping**。

c. 選擇 Manage Drivers (管理驅動程式)，然後選擇 Amazon Redshift。選擇程式庫旁邊的開啟資料夾圖示，然後選擇適當的 JDBC .jar 檔案。

d. 在 Select Connection Profile (選取連線設定檔) 頁面上，將資訊新增至連線設定檔，如下所示：

- 針對 User (使用者)，輸入您的 PingOne 使用者名稱。這是您用於單一登入之 PingOne 帳戶的使用者名稱，該帳戶具有您嘗試驗證之叢集的許可。
- 針對 Password (密碼)，輸入您的 PingOne 密碼。
- 針對 Drivers (驅動程式)，選擇 Amazon Redshift (com.amazon.redshift.jdbc.Driver)。
- 針對 URL，輸入 **jdbc:redshift:iam://*your-cluster-identifier*:*your-cluster-region*/*your-database-name***。

e. 選擇 Extended Properties (延伸屬性)，然後執行下列任一項：

- 針對 login_url，輸入 ***your-ping-sso-login-url***。此值會指定 URL 使用單一登入做為登入的身分驗證。
- 若為 Ping 身分，請針對 plugin_name 輸入 **com.amazon.redshift.plugin.PingCredentialsProvider**。此值指定驅動程式使用 Ping Identity 單一登入作為身分驗證方法。
- 若為具有單一登入的 Ping Identity，請針對 plugin_name 輸入 **com.amazon.redshift.plugin.BrowserSamlCredentialsProvider**。此值會指定驅動程式將 Ping 身分 PingOne 與單一登入搭配使用，以做為身分驗證方法。

ODBC

設定 ODBC 以對 Ping 身分進行身分驗證

- 將資料庫用戶端設定為使用 Ping Identity PingOne 單一登入，透過 ODBC 連線到您的叢集。

  Amazon Redshift 會提供適用於 Linux、Windows 和 macOS 作業系統的 ODBC 驅動程式。安裝 ODBC 驅動程式之前，查明 SQL 用戶端工具是 32 位元或 64 位元。安裝符合 SQL 用戶端工具需求的 ODBC 驅動程式。

  在 Windows 上，於 Amazon Redshift ODBC Driver DSN Setup (Amazon Redshift ODBC 驅動程式 DSN 設定) 頁面的 Connection Settings (連線設定) 之下，輸入下列資訊：

  - 針對 Data source name (資料來源名稱)，輸入 ***your-DSN***。這會指定作為 ODBC 設定檔名稱的資料來源名稱。
  - 針對 Auth Type (驗證類型)，執行下列其中一項操作：
    - 針對 Ping 身分組態，選擇 Identity Provider: Ping Federate (身分提供者：Ping 聯合)。這是 ODBC 驅動程式使用 Ping Identity 單一登入用來驗證的身分驗證方法。
    - 針對具有單一登入的 Ping Identity 組態，選擇身分提供者：瀏覽器 SAML。這是 ODBC 驅動程式透過具有單一登入的 Ping Identity 用來進行身分驗證的身分驗證方法。
  - 針對 Cluster ID (叢集 ID)，輸入 ***your-cluster-identifier***。
  - 針對 Region (區域)，輸入 ***your-cluster-region***。
  - 針對 Database (資料庫)，輸入 ***your-database-name***。
  - 針對 User (使用者)，輸入 ***your-ping-username***。這是您用於單一登入之 PingOne FS 帳戶的使用者名稱，該帳戶具有您嘗試驗證之叢集的許可。僅在 Auth Type (驗證類型) 為 Identity Provider: PingFederate (身分提供者：PingFederate) 的情況下使用此項目。
  - 針對 Password (密碼)，輸入 ***your-ping-password***。僅在 Auth Type (驗證類型) 為 Identity Provider: PingFederate (身分提供者：PingFederate) 的情況下使用此項目。
  - 針對 Listen Port (接聽連接埠)，輸入 ***your-listen-port***。這是本機伺服器接聽的連接埠。預設值為 7890。這僅適用於瀏覽器 SAML 外掛程式。
  - 針對 Response Timeout (回應逾時)，輸入 ***the-number-of-seconds***。這是 IdP 伺服器傳回回應時，在逾時之前等待的秒數。秒數的下限必須為 10。如果建立連線所花的時間超過此臨界值，連線會中止。這僅適用於瀏覽器 SAML 外掛程式。
  - 針對 Login URL (登入 URL)，輸入 ***your-login-url***。這僅適用於瀏覽器 SAML 外掛程式。

在 macOS 和 Linux 上，編輯 `odbc.ini` 檔案，如下所示：

> **Note**
>
> 所有項目都不區分大小寫。

- 針對 clusterid，輸入 ***your-cluster-identifier***。這是所建立 Amazon Redshift 叢集的名稱。
- 針對 region (區域)，輸入 ***your-cluster-region***。這是所建立 Amazon Redshift 叢集 AWS 的區域。
- 針對 database (資料庫)，輸入 ***your-database-name***。這是您嘗試在 Amazon Redshift 叢集上存取的資料庫名稱。
- 針對 locale，輸入 **en-us**。這是錯誤訊息的顯示語言。
- 針對 iam，輸入 **1**。此值指定驅動程式使用 IAM 登入資料進行身分驗證。
- 針對 plugin_name，執行下列其中一項作業：
  - 針對 Ping 身分組態，輸入 **BrowserSAML**。這是 ODBC 驅動程式用來驗證具有 MFA 之 Ping 身分 SSO 的身分驗證方法。
  - 對於具有單一登入的 Ping Identity 組態，請輸入 **Ping**。這是 ODBC 驅動程式透過具有單一登入的 Ping Identity 用來進行身分驗證的身分驗證方法。
- 針對 uid，輸入 ***your-ping-username***。這是您用於單一登入之 Microsoft Azure 帳戶的使用者名稱，該帳戶具有您嘗試驗證之叢集的許可。僅在 plugin_name 為 Ping 的情況下使用此項目。
- 針對 pwd，輸入 ***your-ping-password***。僅在 plugin_name 為 Ping 的情況下使用此項目。
- 針對 login_url，輸入 ***your-login-url***。這是會傳回 SAML 回應的初始單一登入 URL。這僅適用於瀏覽器 SAML 外掛程式。
- 針對 idp_response_timeout，輸入 ***the-number-of-seconds***。這是等待 PingOne 身分回應的指定時間期間 (秒)。這僅適用於瀏覽器 SAML 外掛程式。
- 針對 listen_port，輸入 ***your-listen-port***。這是本機伺服器接聽的連接埠。預設值為 7890。這僅適用於瀏覽器 SAML 外掛程式。

在 macOS 和 Linux 上，同時編輯設定檔設定以新增下列匯出：

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

Okta

您可以使用 Okta 做為身分提供者 (IdP) 來存取您的 Amazon Redshift 叢集。本教學課程說明如何使用 Okta 作為身分提供者 (IdP) 來存取您的 Amazon Redshift 叢集。

步驟 1：設定 Okta 和 AWS 您的帳戶彼此信任

下列程序說明如何設定信任關係。

設定 Okta 和 AWS 您的帳戶彼此信任

1. 建立或使用現有的 Amazon Redshift 叢集，以供 Okta 使用者連線。若要設定連線，則需要此叢集的特定屬性，例如叢集識別碼。如需詳細資訊，請參閱建立叢集。
2. 在 Okta 入口網站上將 Amazon Redshift 新增為新的應用程式。如需詳細步驟，請參閱 Okta 文件。
   - 選擇 Add Application (新增應用程式)。
   - 在 Add Application (新增應用程式) 下方，選擇 Create New App (建立新的應用程式)。
   - 在 Create a New Add Application Integration (建立新的新增應用程式整合) 頁面上，針對 Platform (平台) 選擇 Web。
   - 針對 Sign on method (登入方法)，選擇 SAML v2.0。
   - 在 General Settings (一般設定) 頁面上，針對 App name (應用程式名稱) 輸入 ***your-redshift-saml-sso-name***。這是您的應用程式名稱。
   - 在 SAML Settings (SAML 設定) 頁面上，針對 Single sign on URL (單一登入 URL)，輸入 ***your-redshift-local-host-url***。這是 SAML 聲明會重新導向至的本機主機與連接埠，例如 `http://localhost:7890/redshift/`。
3. 使用單一登入 URL 值做為收件人 URL 和目標 URL。
4. 針對 Signing (簽署)，選擇 Sign Assertion (簽署聲明)。
5. 針對對象 URI (SP 實體 ID)，針對宣告輸入 **`urn:amazon:webservices`**，如下表所示。
6. 在進階設定區段中，針對 SAML 發行者 ID 輸入 ***your-Identity-Provider-Issuer-ID***，您可以在檢視設定指示區段中找到此值。

7. 在 Attribute Statements (屬性陳述式) 區段中，建立如下表所示的宣告。

宣告名稱	Value
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 App Embed Link (應用程式內嵌連結) 區段中，尋找您可以用來做為瀏覽器 SAML 外掛程式登入 URL 的 URL。
9. 在 IAM 主控台上建立 IAM SAML 身分提供者。您所提供的中繼資料文件，是您在設定 Okta 時儲存的聯合中繼資料 XML 檔案。如需詳細步驟，請參閱《IAM 使用者指南》中的建立和管理 IAM 身分提供者 (主控台)。
10. 在 IAM 主控台上建立 SAML 2.0 聯合的 IAM 角色。如需詳細步驟，請參閱《IAM 使用者指南》中的為 SAML 建立角色。
11. 建立 IAM 政策，您可將其附加至您在 IAM 主控台上為 SAML 2.0 聯合建立的 IAM 角色。如需詳細步驟，請參閱《IAM 使用者指南》中的建立 IAM 政策 (主控台)。如需 Azure AD 範例，請參閱設定 JDBC 或 ODBC 單一登入身分驗證。

步驟 2：設定 JDBC 或 ODBC 以對 Okta 進行身分驗證

JDBC

設定 JDBC 以對 Okta 進行身分驗證

- 將資料庫用戶端設定為使用 Okta 單一登入，透過 JDBC 連線到您的叢集。

您可使用任何採用 JDBC 驅動程式的用戶端使用 Okta 單一登入進行連線，或使用 Java 之類的語言透過指令碼進行連線。如需安裝和組態資訊，請參閱 設定適用於 Amazon Redshift 的 JDBC 驅動器 2.x 版的連線。

例如，您可以使用 SQLWorkbench/J 作為用戶端。當您設定 SQLWorkbench /J 時，您的資料庫 URL 會使用下列格式。

```
jdbc:redshift:iam://cluster-identifier:us-west-1/dev
```

如果您使用 SQLWorkbench /J 作為用戶端，請執行下列步驟：

a. 啟動 SQL Workbench/J。在選取連線設定檔頁面中，新增設定檔群組，例如 **Okta**。

b. 針對 Connection Profile (連線設定檔)，輸入 ***your-connection-profile-name***，例如 **Okta**。

c. 選擇 Manage Drivers (管理驅動程式)，然後選擇 Amazon Redshift。選擇程式庫旁邊的開啟資料夾圖示，然後選擇適當的 JDBC .jar 檔案。

d. 在 Select Connection Profile (選取連線設定檔) 頁面上，將資訊新增至連線設定檔，如下所示：

- 在 User (使用者) 輸入您的 Okta 使用者名稱。這是您用於單一登入之 Okta 帳戶的使用者名稱，該帳戶具有您嘗試驗證之叢集的許可。
- 在 Password (密碼) 輸入您的 Okta 密碼。
- 針對 Drivers (驅動程式)，選擇 Amazon Redshift (com.amazon.redshift.jdbc.Driver)。
- 針對 URL，輸入 **jdbc:redshift:iam://*your-cluster-identifier*:*your-cluster-region*/*your-database-name***。

e. 選擇 Extended Properties (延伸屬性)，然後執行下列任一項：

- 針對 login_url，輸入 ***your-okta-sso-login-url***。此值會指定 URL 使用單一登入做為登入 Okta 的身分驗證。
- 針對 Okta 單一登入，請為 plugin_name 輸入 **com.amazon.redshift.plugin.OktaCredentialsProvider**。此值指定驅動程式使用 Okta 單一登入作為身分驗證方法。
- 針對具有 MFA 的 Okta 單一登入，請為 plugin_name 輸入 **com.amazon.redshift.plugin.BrowserSamlCredentialsProvider**。此值會指定驅動程式使用具有 MFA 的 Okta 單一登入做為身分驗證方法。

ODBC

設定 ODBC 以對 Okta 進行身分驗證

- 將資料庫用戶端設定為使用 Okta 單一登入，透過 ODBC 連線到您的叢集。

  Amazon Redshift 會提供適用於 Linux、Windows 和 macOS 作業系統的 ODBC 驅動程式。安裝 ODBC 驅動程式之前，查明 SQL 用戶端工具是 32 位元或 64 位元。安裝符合 SQL 用戶端工具需求的 ODBC 驅動程式。

  在 Windows 上，於 Amazon Redshift ODBC Driver DSN Setup (Amazon Redshift ODBC 驅動程式 DSN 設定) 頁面的 Connection Settings (連線設定) 之下，輸入下列資訊：

  - 針對 Data source name (資料來源名稱)，輸入 ***your-DSN***。這會指定作為 ODBC 設定檔名稱的資料來源名稱。
  - 針對 Auth Type (驗證類型)，執行下列其中一項操作：
    - 針對 Okta 單一登入組態，請選擇 **Identity Provider: Okta**。這是 ODBC 驅動程式使用 Okta 單一登入來進行驗證的身分驗證方法。
    - 針對具有 MFA 的 Okta 單一登入組態，請選擇 **Identity Provider: Browser SAML**。這是 ODBC 驅動程式使用具有 MFA 之 Okta 單一登入來進行驗證的身分驗證方法。
  - 針對 Cluster ID (叢集 ID)，輸入 ***your-cluster-identifier***。
  - 針對 Region (區域)，輸入 ***your-cluster-region***。
  - 針對 Database (資料庫)，輸入 ***your-database-name***。
  - 針對 User (使用者)，輸入 ***your-okta-username***。這是您用於單一登入之 Okta 帳戶的使用者名稱，該帳戶具有您嘗試驗證之叢集的許可。僅在 Auth Type (驗證類型) 為Identity Provider: Okta (身分提供者：Okta) 的情況下使用此項目。
  - 針對 Password (密碼)，輸入 ***your-okta-password***。僅在 Auth Type (驗證類型) 為Identity Provider: Okta (身分提供者：Okta) 的情況下使用此項目。

  在 macOS 和 Linux 上，編輯 `odbc.ini` 檔案，如下所示：

  > **Note**
  >
  > 所有項目都不區分大小寫。

- 針對 clusterid，輸入 ***your-cluster-identifier***。這是所建立 Amazon Redshift 叢集的名稱。
- 針對 region (區域)，輸入 ***your-cluster-region***。這是所建立 Amazon Redshift 叢集 AWS 的區域。
- 針對 database (資料庫)，輸入 ***your-database-name***。這是您嘗試在 Amazon Redshift 叢集上存取的資料庫名稱。
- 針對 locale，輸入 **en-us**。這是錯誤訊息的顯示語言。
- 針對 iam，輸入 **1**。此值指定驅動程式使用 IAM 登入資料進行身分驗證。
- 針對 plugin_name，執行下列其中一項作業：
  - 若為具有 MFA 的 Okta 單一登入組態，請輸入 **BrowserSAML**。這是 ODBC 驅動程式用來驗證具有 MFA 之 Okta 單一登入的身分驗證方法。
  - 針對 Okta 單一登入組態，請輸入 **Okta**。這是 ODBC 驅動程式使用 Okta 單一登入來進行驗證的身分驗證方法。
- 針對 uid，輸入 ***your-okta-username***。這是您用於單一登入之 Okta 帳戶的使用者名稱，該帳戶具有您嘗試驗證之叢集的許可。僅在 plugin_name 為 Okta 的情況下使用此項目。
- 針對 pwd，輸入 ***your-okta-password***。僅在 plugin_name 為 Okta 的情況下使用此項目。
- 針對 login_url，輸入 ***your-login-url***。這是會傳回 SAML 回應的初始單一登入 URL。這僅適用於瀏覽器 SAML 外掛程式。
- 針對 idp_response_timeout，輸入 ***the-number-of-seconds***。這是等待 PingOne 回應的指定時間期間 (秒)。這僅適用於瀏覽器 SAML 外掛程式。
- 針對 listen_port，輸入 ***your-listen-port***。這是本機伺服器接聽的連接埠。預設值為 7890。這僅適用於瀏覽器 SAML 外掛程式。

在 macOS 和 Linux 上，同時編輯設定檔設定以新增下列匯出：

```
export ODBCINI=/opt/amazon/redshift/Setup/odbc.ini
```

```
export ODBCINSTINI=/opt/amazon/redshift/Setup/odbcinst.ini
```

授權 Amazon Redshift 代表您存取 AWS 服務

有些 Amazon Redshift 功能需要 Amazon Redshift 代表您存取其他 AWS 服務。例如，COPY 和 UNLOAD 命令可在使用 Amazon S3 儲存貯體的 Amazon Redshift 叢集上載入或卸載資料。CREATE EXTERNAL FUNCTION 命令可以使用純量 AWS Lambda 使用者定義函數 (UDF) 叫用 Lambda 函數。Amazon Redshift Spectrum 可以使用 Amazon Athena 或 中的資料目錄 AWS Glue。若要讓 Amazon Redshift 叢集代替您執行作業，您需為叢集提供安全憑證。提供安全憑證的建議方式為指定 AWS Identity and Access Management (IAM) 角色。對於 COPY 和 UNLOAD，您可以提供臨時憑證。

如果使用者想要與 AWS 外部互動，則需要程式設計存取 AWS 管理主控台。授予程式設計存取權的方式取決於正在存取的使用者類型 AWS。

若要授予使用者程式設計存取權，請選擇下列其中一個選項。

哪個使用者需要程式設計存取權？	到	根據
IAM	（建議）使用主控台登入資料做為臨時登入資料，以簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface《使用者指南》中的登入以進行 AWS 本機開發。 • AWS SDKs，請參閱 AWS SDKs 和工具參考指南中的登入以進行 AWS 本機開發。
人力資源身分 (IAM Identity Center 中管理的使用者)	使用臨時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface《使用者指南》中的設定 AWS CLI 要使用 AWS IAM Identity Center的 。

哪個使用者需要程式設計存取權？	到	根據
		• AWS SDKs、工具和 AWS APIs，請參閱 AWS SDK 和工具參考指南中的 SDKsIAM Identity Center 身分驗證。
IAM	使用臨時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	遵循《IAM 使用者指南》中將臨時登入資料與 AWS 資源搭配使用的指示。
IAM	(不建議使用) 使用長期登入資料來簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的使用 IAM 使用者憑證進行身分驗證。 • AWS SDKs和工具，請參閱 AWS SDKs和工具參考指南中的使用長期憑證進行身分驗證。 • 對於 AWS APIs，請參閱《IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。

接下來，了解如何建立具有適當許可的 IAM 角色來存取其他 AWS 服務。您也需要在執行 Amazon Redshift 命令時將角色與您的叢集相關聯，並指定角色的 Amazon Resource Name (ARN)。如需詳細資訊，請參閱使用 IAM 角色授權 COPY、UNLOAD、CREATE EXTERNAL FUNCTION 和 CREATE EXTERNAL SCHEMA 操作。

此外，超級使用者可以將 ASSUMEROLE 權限授予特定使用者和群組，以提供 COPY 和 UNLOAD 操作角色的存取權。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 GRANT。

建立 IAM 角色以允許 Amazon Redshift 叢集存取 AWS 服務

建立 IAM 角色以允許 Amazon Redshift 叢集存取 AWS 服務

若要建立 IAM 角色以允許 Amazon Redshift 叢集代表您與其他 AWS 服務進行通訊，請採取下列步驟。本節中使用的值是範例，您可以根據需要選擇值。

建立 IAM 角色以允許 Amazon Redshift 存取 AWS 服務

1. 開啟 IAM 主控台。
2. 在導覽窗格中，選擇 Roles (角色)。
3. 選擇 Create Role (建立角色)。
4. 選擇 AWS 服務，然後選擇 Redshift。
5. 在選取您的使用案例下，選擇Redshift – 可自訂，然後選擇下一步：許可。連接許可政策頁面隨即出現。
6. 舉例來說，若要使用 COPY 存取 Amazon S3，您可以使用 **AmazonS3ReadOnlyAccess** 和 append。若要使用 COPY 或 UNLOAD 存取 Amazon S3，我們建議您建立受管政策，以便相應地限制對所需儲存貯體和字首的存取。對於讀取和寫入操作，我們建議您強制執行最低權限並僅限用於 Amazon Redshift 所需的 Amazon S3 儲存貯體和索引鍵字首。

   如需為 CREATE EXTERNAL FUNCTION 命令調用 Lambda 函數的存取權，請新增 **AWSLambdaRole**。

   對於 Redshift Spectrum，除了 Amazon S3 存取之外，另請新增 **AWSGlueConsoleFullAccess** 或 **AmazonAthenaFullAccess**。

   選擇下一步：標籤。
7. Add tags (新增標籤) 頁面隨即出現。您可以選擇性地新增標籤。選擇下一步：檢閱。
8. 針對 Role name (角色名稱)，輸入您的角色名稱，例如 **RedshiftCopyUnload**。選擇建立角色。
9. 新角色適用於叢集上使用該角色的所有使用者。若要限制只有特定叢集上的特定使用者或特定區域中的叢集可存取，請編輯角色的信任關係。如需詳細資訊，請參閱限制對 IAM 角色的存取。
10. 將角色與叢集相關聯。您可在建立叢集或新增角色至現有叢集時將 IAM 角色與叢集相關聯。如需詳細資訊，請參閱將 IAM 角色與叢集建立關聯。

> **Note**
>
> 若要限制對特定資料的存取權，請使用授予所需最低權限的 IAM 角色。

限制對 IAM 角色的存取

根據預設，Amazon Redshift 叢集上可用的 IAM 角色適用於該叢集上所有的使用者。但您可以選擇將 IAM 角色限制為只有特定叢集上的特定 Amazon Redshift 資料庫使用者或特定區域可存取。

若只要允許特定資料庫使用者使用 IAM 角色，請遵循以下步驟。

識別可存取 IAM 角色的特定資料庫使用者

1. 識別 Amazon Redshift 叢集中資料庫使用者的 Amazon Resource Name (ARN)。資料庫使用者的 ARN 格式如下：`arn:aws:redshift:region:account-id:dbuser:cluster-name/user-name`。

   對於 Amazon Redshift Serverless，請使用以下 ARN 格式。`arn:aws:redshift:region:account-id:dbuser:serverless-account-id-workgroup-id/user-name`
2. 開啟 IAM 主控台。
3. 在導覽窗格中，選擇 Roles (角色)。
4. 選擇您想要限制只有特定 Amazon Redshift 資料庫使用者可存取的 IAM 角色。
5. 選擇 Trust Relationships (信任關係) 索引標籤，然後選擇 Edit Trust Relationship (編輯信任關係)。允許 Amazon Redshift AWS 代表您存取其他服務的新 IAM 角色具有信任關係，如下所示：

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
```

```
    }
  ]
}
```

6. 新增條件至信任關係的 sts:AssumeRole 動作區段，將 sts:ExternalId 欄位限制在您所指定的值。包含您想要授予角色存取之各資料庫使用者的 ARN。外部 ID 可以是任何唯一的字串。

   例如，以下的信任關係指定只有 user1 區域中 user2 叢集上的 my-cluster 和 us-west-2 資料庫使用者有使用此 IAM 角色的許可。

   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. 選擇 Update Trust Policy (更新信任政策)。

將 IAM 角色限制為 AWS 區域

您可以將 IAM 角色限制為只能在特定區域中存取 AWS 。根據預設，Amazon Redshift 的 IAM 角色不受限於任何單一區域。

若要限制區域可使用的 IAM 角色，請遵循以下步驟。

識別 IAM 角色的允許區域

1. 在 https：//https://console.aws.amazon.com/ 開啟 IAM 主控台。
2. 在導覽窗格中，選擇角色。
3. 選擇您想要針對特定區域修改的角色。
4. 選擇 Trust Relationships (信任關係) 索引標籤，然後選擇 Edit Trust Relationship (編輯信任關係)。允許 Amazon Redshift AWS 代表您存取其他服務的新 IAM 角色具有信任關係，如下所示：

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "Service": "redshift.amazonaws.com"
         },
         "Action": "sts:AssumeRole"
       }
     ]
   }
   ```

5. 搭配您想要允許使用角色的特定區域清單，修改 `Service` 的 `Principal` 清單。`Service` 清單中的各區域必須使用下列格式：`redshift.region.amazonaws.com`。

   例如，以下經編輯後的信任關係僅允許在 `us-east-1` 和 `us-west-2` 區域使用 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. 選取 Update Trust Policy (更新信任政策)

在 Amazon Redshift 中鏈結 IAM 角色

當您將角色連接到叢集時，您的叢集可以擔任該角色來 AWS Lambda 代表您存取 Amazon S3 AWS Glue、Amazon Athena 和 。如果連接至叢集的角色無法存取必要資源，您可鏈結其他角色，該角色甚至可以來自其他帳戶。您的叢集接著會暫時擔任鏈結的角色，以存取資料。您也可以用鏈結角色的方式來授予跨帳戶存取。鏈結中的每個角色都會擔任鏈結中的下一個角色，直到叢集擔任鏈結尾端的角色為止。您可建立關聯的 IAM 角色數量上限需符合配額。如需詳細資訊，請參閱 中的配額「Amazon Redshift 存取其他服務的叢集 IAM 角色 AWS 」Amazon Redshift 物件的配額。

> **Note**
>
> 您必須指定 IAM 角色，鏈結才能正常運作。

例如，假設 A 公司想要存取屬於 B 公司的 Amazon S3 儲存貯體中的資料。A 公司為名為 的 Amazon Redshift 建立 AWS 服務角色，RoleA並將其連接到其叢集。B 公司應建立一個名為 RoleB 的角色，允許其存取 B 公司儲存貯體內的資料。若要存取 B 公司儲存貯體內的資料，A 公司應使用鏈結 iam_role 和 RoleA 的RoleB 參數執行 COPY 命令。在 COPY 操作期間，RoleA 會暫時擔任 RoleB，以存取 Amazon S3 儲存貯體。

若要鏈結角色，您應建立角色間的信任關係。擔任其他角色的角色 (例如 RoleA) 必須擁有許可政策，允許其擔任下一個鏈結的角色 (例如 RoleB)。另一方面，要傳送許可的角色 (RoleB) 必須擁有信任政策，允許其將許可傳送給上一個鏈結的角色 (RoleA)。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 角色。

鏈結中的第一個角色必須為連接至叢集的角色。第一個角色，以及擔任鏈結中下一個角色的每個後續角色，必須擁有加入特定陳述式的政策。此陳述式對 Allow動作和 sts:AssumeRole 元素中下一個角色的 Amazon Resource Name (ARN) 有 Resource 的效果。在我們的範例中，RoleA 具有下列許可政策，允許其擔任 AWS 帳戶 210987654321 的 RoleB。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1487639602000",
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/RoleB"
        }
    ]
}
```

傳遞給另一個角色的角色必須與擔任該角色的角色或擁有該角色 AWS 的帳戶建立信任關係。在我們的範例中，RoleB 擁有下列與 RoleA 建立信任關係的信任政策。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/RoleA"
      }
    }
  ]
}
```

下列信任政策會與 、RoleA AWS 帳戶 的擁有者建立信任關係123456789012。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      }
    }
  ]
}
```

> **Note**
>
> 若要將角色鏈結授權限制為特定使用者，請定義條件。如需詳細資訊，請參閱限制對 IAM 角色的存取。

執行 UNLOAD、COPY、CREATE EXTERNAL FUNCTION 或 CREATE EXTERNAL SCHEMA 命令時，您應將逗號分隔的角色 ARN 清單加入到 iam_role 參數內，以鏈結角色。以下顯示 iam_role 參數中鏈結角色的語法。

```
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**
>
> 整個角色鏈結用單引號括住，且不得包含空格。

在下列範例中，RoleA 連接到屬於 AWS 帳戶 123456789012 的叢集。RoleB (屬於帳戶 210987654321) 具備許可存取名為 s3://companyb/redshift/ 的儲存貯體。下列範例鏈結 RoleA 和 RoleB，將資料 UNLOAD (卸載) 到 s3://companyb/redshift/ 儲存貯體。

```
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';
```

以下範例使用 COPY 命令載入在前一範例中卸載的資料。

```
copy venue
from 's3://companyb/redshift/venue_pipe_'
iam_role 'arn:aws:iam::123456789012:role/RoleA,arn:aws:iam::210987654321:role/RoleB';
```

在以下範例中，CREATE EXTERNAL SCHEMA 使用鏈結的角色來擔任 RoleB 角色。

```
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';
```

在以下範例中，CREATE EXTERNAL FUNCTION 使用鏈結的角色來擔任 RoleB 角色。

```
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 角色授權 COPY、UNLOAD、CREATE EXTERNAL FUNCTION 和 CREATE EXTERNAL SCHEMA 操作

您可以用 COPY 命令將資料載入 (或匯入) Amazon Redshift，也能使用 UNLOAD 命令從 Amazon Redshift 卸載 (或匯出) 資料。您可以使用 CREATE EXTERNAL FUNCTION 命令來建立從中叫用函數的使用者定義函數 AWS Lambda。

使用 Amazon Redshift Spectrum 時，您可以使用 CREATE EXTERNAL SCHEMA 命令指定 Amazon S3 儲存貯體的位置 (其中包含您資料)。執行 COPY、UNLOAD 或 CREATE EXTERNAL SCHEMA 命令時，您必須提供安全憑證。這些憑證會授權 Amazon Redshift 叢集對您的目標目的地 (例如 Amazon S3 儲存貯體) 讀取或寫入資料。

執行CREATE EXTERNAL FUNCTION時，您可以使用 IAM 角色參數提供安全憑證。這些登入資料授權您的 Amazon Redshift 叢集從中叫用 Lambda 函數 AWS Lambda。提供安全登入資料的偏好方法是

指定 AWS Identity and Access Management (IAM) 角色。對於 COPY 和 UNLOAD，您可以提供臨時憑證。如需有關建立 IAM 角色的詳細資訊，請參閱 授權 Amazon Redshift 代表您存取 AWS 服務。

如果使用者想要與 AWS 外部互動，則需要程式設計存取 AWS 管理主控台。授予程式設計存取權的方式取決於正在存取的使用者類型 AWS。

若要授予使用者程式設計存取權，請選擇下列其中一個選項。

哪個使用者需要程式設計存取權？	到	根據
IAM	（建議）使用主控台登入資料做為臨時登入資料，以簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《使用者指南》中的登入以進行 AWS 本機開發。 • AWS SDKs請參閱 AWS SDKs和工具參考指南中的登入以進行 AWS 本機開發。
人力資源身分 (IAM Identity Center 中管理的使用者)	使用臨時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《使用者指南》中的設定 AWS CLI 要使用 AWS IAM Identity Center的 。 • AWS SDKs、工具和 AWS APIs，請參閱 AWS SDK 和工具參考指南中的 SDKsIAM Identity Center 身分驗證。
IAM	使用臨時登入資料簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	遵循《IAM 使用者指南》中將臨時登入資料與 AWS 資源搭配使用的指示。

哪個使用者需要程式設計存取權？	到	根據
IAM	(不建議使用) 使用長期登入資料來簽署對 AWS CLI、 AWS SDKs程式設計請求。 AWS APIs	請依照您要使用的介面所提供的指示操作。 • 如需 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的使用 IAM 使用者憑證進行身分驗證。 • AWS SDKs和工具，請參閱 AWS SDKs和工具參考指南中的使用長期憑證進行身分驗證。 • 對於 AWS APIs，請參閱《 IAM 使用者指南》中的管理 IAM 使用者的存取金鑰。

使用 IAM 角色的步驟如下：

- 建立要用於 Amazon Redshift 叢集的 IAM 角色。
- 將 IAM 角色與叢集相關聯。
- 呼叫 COPY、UNLOAD、CREATE EXTERNAL SCHEMA 或 CREATE EXTERNAL FUNCTION 命令時，請加入 IAM 角色的 ARN。

將 IAM 角色與叢集建立關聯

建立 IAM 角色來授權 Amazon Redshift 代表您存取其他 AWS 服務後，您必須將該角色與 Amazon Redshift 叢集相關聯。您必須執行這項操作，才能使用該角色來載入或卸載資料。

將 IAM 角色與叢集建立關聯所需的許可

若要將 IAM 角色與叢集相關聯，使用者必須擁有該 IAM 角色的 `iam:PassRole` 許可。管理員可以運用此許可，限制使用者可將哪些 IAM 角色與 Amazon Redshift 叢集相關聯。我們建議的最佳實務是，將許可政策連接到 IAM 角色，然後根據需要將其指派給使用者和群組。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。

以下範例顯示可連接到使用者的 IAM 政策，以允許使用者採取下列動作：

- 取得該使用者帳戶所擁有之所有 Amazon Redshift 叢集的詳細資訊。
- 將任何三個 IAM 角色與兩個 Amazon Redshift 叢集中的任一個相關聯。

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"
             ]
        }
    ]
}
```

使用者取得適當的許可後，該使用者可以將 IAM 角色與 Amazon Redshift 叢集相關聯。然後，IAM 角色就能使用 COPY 或 UNLOAD 命令，或是其他 Amazon Redshift 命令。

如需 IAM 政策的相關資訊，請參閱《IAM 使用者指南》中的 IAM 政策概觀。

管理 IAM 角色與叢集的關聯

您可以在建立叢集時將 IAM 角色與 Amazon Redshift 叢集相關聯。您也可以修改現有的叢集，並新增或移除一或多個 IAM 角色關聯。

請注意以下事項：

- 您可建立關聯的 IAM 角色數量上限需符合配額。
- IAM 角色可以與多個 Amazon Redshift 叢集相關聯。
- 只有在 IAM 角色和叢集都屬於相同 AWS 帳戶時，IAM 角色才能與 Amazon Redshift 叢集建立關聯。

您可在主控台中使用下列程序管理叢集的 IAM 角色關聯。

管理 IAM 角色關聯

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後選擇您要更新的叢集。
3. 針對 Actions (動作)，選擇 Manage IAM roles (管理 IAM 角色) 以顯示與叢集相關聯的目前 IAM 角色清單。
4. 在 Manage IAM roles (管理 IAM 角色) 頁面上，選擇要新增的可用 IAM 角色，然後選擇 Add IAM role (新增 IAM 角色)。
5. 選擇 Done (完成) 以儲存變更。

您可以使用 AWS CLI 下列方法，透過 管理叢集的 IAM 角色關聯。

若要在建立叢集時將 IAM 角色與叢集相關聯，請用 `--iam-role-arns` 命令的 `create-cluster` 參數指定 IAM 角色的 Amazon Resource Name (ARN)。您在呼叫 `create-cluster` 命令時可新增的 IAM 角色數量上限需符合配額。

IAM 角色與 Amazon Redshift 叢集的相關聯和取消關聯是非同步的程序。您可呼叫 `describe-clusters` 命令，以取得所有 IAM 角色叢集關聯的狀態。

以下範例將兩個 IAM 角色與名為 my-redshift-cluster 的新建立叢集相關聯。

```
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"
```

若要將 IAM 角色與現有的 Amazon Redshift 叢集建立關聯，請在 modify-cluster-iam-roles 命令的 --add-iam-roles 參數中，指定 IAM 角色的 Amazon Resource Name (ARN)。您在呼叫 modify-cluster-iam-roles 命令時可新增的 IAM 角色數量上限需符合配額。

以下範例將一個 IAM 角色與名為 my-redshift-cluster 的現有叢集相關聯。

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier "my-redshift-cluster" \
    --add-iam-roles "arn:aws:iam::123456789012:role/RedshiftCopyUnload"
```

若要將 IAM 角色與叢集取消關聯，請用 modify-cluster-iam-roles 命令的 --remove-iam-roles 參數指定 IAM 角色的 ARN。modify-cluster-iam-roles您在呼叫 modify-cluster-iam-roles 命令時可移除的 IAM 角色數量上限需符合配額。

下列範例會從名為 的叢集中移除123456789012 AWS 帳戶 IAM 角色的關聯my-redshift-cluster。

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier "my-redshift-cluster" \
    --remove-iam-roles "arn:aws:iam::123456789012:role/RedshiftCopyUnload"
```

使用 列出叢集的 IAM 角色關聯 AWS CLI

若要列出與 Amazon Redshift 叢集相關聯的所有 IAM 角色，以及 IAM 角色關聯的狀態，請呼叫 describe-clusters 命令。IamRoles 清單會傳回與叢集關聯之每個 IAM 角色的 ARN，如下列輸出範例所示。

已經與叢集相關聯的角色狀態會顯示為 in-sync。正在與叢集相關聯的角色狀態會顯示為 adding。正在與叢集取消關聯的角色狀態會顯示為 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"
                }
            ],
            ...
        }
    ]
}
```

如需使用 的詳細資訊 AWS CLI，請參閱 AWS CLI 使用者指南。

建立 IAM 角色做為 Amazon Redshift 的預設角色

當您透過 Redshift 主控台建立 IAM 角色時，Amazon Redshift 會以程式設計方式在您的 中建立角色，AWS 帳戶 並自動將現有的 AWS 受管政策連接到這些角色。此方法意味著您可以保留在 Redshift 主控台內，而不必切換到 IAM 主控台來建立角色。若要對在 Amazon Redshift 主控台中建立的現有 IAM 角色進行更精細的許可控制，您可以將自訂的受管政策連接到 IAM 角色。

在主控台中建立的 IAM 角色

當您使用 Amazon Redshift 主控台建立 IAM 角色時，Amazon Redshift 會追蹤透過主控台建立的所有 IAM 角色。Amazon Redshift 會預先選取最新的預設 IAM 角色，以建立所有新叢集和從快照還原叢集。

您可以透過主控台建立 IAM 角色，該角色有權執行 SQL 命令的政策。這些命令包括 COPY、UNLOAD、CREATE EXTERNAL FUNCTION、CREATE EXTERNAL TABLE、CREATE EXTERNAL SCHEMA、CREATE MODEL 或 CREATE LIBRARY。或者，您可以透過建立自訂政策並將其連接到 IAM 角色，進而更精細地控制使用者對 AWS 資源的存取權限。

使用主控台建立 IAM 角色並將其設定為叢集預設值時，您不必提供 IAM 角色的 Amazon Resource Name (ARN) 即可執行驗證和授權。

您透過主控台為叢集建立的 IAM 角色會自動附加 `AmazonRedshiftAllCommandsFullAccess` 受管政策。此 IAM 角色可讓 Amazon Redshift 複製、卸載、查詢和分析 IAM 帳戶中 AWS 資源的資料。受管政策可讓您存取 COPY、UNLOAD、CREATE EXTERNAL FUNCTION、CREATE EXTERNAL SCHEMA、CREATE MODEL 和 CREATE LIBRARY 操作。此政策也會授予許可，以執行相關 AWS 服務的 SELECT 陳述式，例如 Amazon S3、Amazon CloudWatch Logs、Amazon SageMaker AI 和 AWS Glue。

CREATE EXTERNAL FUNCTION、CREATE EXTERNAL SCHEMA、CREATE MODEL 和 CREATE LIBRARY 命令有 `default` 關鍵字。對於這些命令的這個關鍵字，Amazon Redshift 會使用設定為預設值且在命令執行時與叢集關聯的 IAM 角色。您可以執行 DEFAULT_IAM_ROLE 命令來檢查目前已連接到叢集的 IAM 角色。

若要控制建立的 IAM 角色的存取權限，並將其設定為 Redshift 叢集的預設值，請使用 ASSUMEROLE 權限。此存取控制適用於資料庫使用者和群組執行命令 (例如前面列出的命令)。將 ASSUMEROLE 權限授予 IAM 角色的使用者或群組之後，使用者或群組就可以在執行這些命令時擔任該角色。透過使用 ASSUMEROLE 權限，您可以視需要授予適當命令的存取權。

使用 Amazon Redshift 主控台，您可以執行以下操作：

- 建立 IAM 角色做為預設角色
- 從叢集移除 IAM 角色
- 將 IAM 角色與叢集建立關聯
- 將 IAM 角色設定為預設角色
- 讓 IAM 角色不再是叢集的預設角色

AmazonRedshiftAllCommandsFullAccess 受管政策的許可

下列範例顯示 AmazonRedshiftAllCommandsFullAccess 受管政策中的許可，這些許可允許針對設定為叢集預設的 IAM 角色執行某些動作。連接了許可政策的 IAM 角色授權使用者或群組可以執行和不能執行的操作。有了這些許可，您可以從 Amazon S3 執行 COPY 命令、執行 UNLOAD，然後使用 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*/*"
            ]
}
```

下列範例顯示 AmazonRedshiftAllCommandsFullAccess 受管政策中的許可，這些許可允許針對設定為叢集預設的 IAM 角色執行某些動作。連接了許可政策的 IAM 角色授權使用者或群組可以執行和不能執行的操作。根據下列許可，您可以執行 CREATE EXTERNAL FUNCTION 命令。

```
{
    "Action": [
        "lambda:InvokeFunction"
    ],
    "Resource": "arn:aws:lambda:*:*:function:*redshift*"
}
```

下列範例顯示 AmazonRedshiftAllCommandsFullAccess 受管政策中的許可，這些許可允許針對設定為叢集預設的 IAM 角色執行某些動作。連接了許可政策的 IAM 角色授權使用者或群組可以執行和不能執行的操作。有了下列許可，您就可以執行 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*"
            ]
}
```

下列範例顯示 AmazonRedshiftAllCommandsFullAccess 受管政策中的許可，這些許可允許針對設定為叢集預設的 IAM 角色執行某些動作。連接了許可政策的 IAM 角色授權使用者或群組可以執行和不能執行的操作。有了下列許可，您就可以使用聯合查詢執行 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"
                }
            }
},
```

使用主控台管理為叢集建立的 IAM 角色

若要建立、修改和移除從 Amazon Redshift 主控台建立的 IAM 角色，請使用主控台中的叢集區段。

建立 IAM 角色做為預設角色

在主控台上，您可以為叢集建立自動連接 AmazonRedshiftAllCommandsFullAccess 政策的 IAM 角色。您建立的新 IAM 角色可讓 Amazon Redshift 複製、載入、查詢和分析 IAM 帳戶中 Amazon 資源中的資料。

只能將一個 IAM 角色設定為叢集的預設角色。如果在目前將現有 IAM 角色指派為預設值時，建立另一個 IAM 角色做為叢集預設角色，則新的 IAM 角色會取代另一個角色作為預設角色。

建立新叢集和設定為新叢集預設值的 IAM 角色

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單上，選擇叢集。 AWS 區域 系統會列出目前 中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。
3. 選擇 Create cluster (建立叢集) 以建立叢集。
4. 遵循主控台頁面說明以輸入 Cluster configuration (叢集組態) 的屬性。如需此步驟的相關資訊，請參閱 建立叢集。
5. (選用) 選擇載入範例資料將範例資料集載入 Amazon Redshift 叢集，以開始使用查詢編輯器查詢資料。

   如果您在防火牆後，則資料庫連接埠必須是接受傳入連線的開放連接埠。
6. 遵循主控台頁面說明以輸入資料庫組態的屬性。
7. 在叢集許可下，從管理 IAM 角色中選擇建立 IAM 角色。
8. 選擇下列其中一種方法來指定 IAM 角色要存取的 Amazon S3 儲存貯體：
   - 選擇不使用其他 Amazon S3 儲存貯體來建立 IAM 角色，而無需指定特定的 Amazon S3 儲存貯體。
   - 選擇任何 Amazon S3 儲存貯體，允許擁有 Amazon Redshift 叢集存取權的使用者也可以存取您的任何 Amazon S3 儲存貯體及其中的 AWS 帳戶內容。
   - 選擇特定的 Amazon S3 儲存貯體，以指定要建立的 IAM 角色具有存取許可的一或多個 Amazon S3 儲存貯體。然後從資料表中選擇一或多個 Amazon S3 儲存貯體。
9. 選擇建立 IAM 角色作為預設值。Amazon Redshift 會自動建立 IAM 角色，並將其設定為叢集的預設角色。
10. 選擇 Create cluster (建立叢集) 以建立叢集。該叢集可能需要幾分鐘方可使用。

從叢集移除 IAM 角色

您可以從叢集移除一或多個 IAM 角色。

從叢集移除 IAM 角色

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單上，選擇叢集。 AWS 區域 系統會列出目前 中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。
3. 選擇您要從中移除 IAM 角色的叢集。
4. 在叢集許可下，選擇一或多個要從叢集移除的 IAM 角色。
5. 在管理 IAM 角色中，選擇移除 IAM 角色。

將 IAM 角色與叢集建立關聯

您可以將一或多個 IAM 角色與叢集建立關聯。

將 IAM 角色與叢集產生關聯

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單上，選擇叢集。 AWS 區域 系統會列出目前 中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。
3. 選擇您要與 IAM 角色建立關聯的叢集。
4. 在叢集許可下，選擇一或多個要與叢集建立關聯的 IAM 角色。
5. 從管理 IAM 角色中，選擇關聯 IAM 角色。
6. 選擇一個或多個 IAM 角色來與您的叢集建立關聯。
7. 選擇關聯 IAM 角色。

將 IAM 角色設定為預設角色

您可以將 IAM 角色設定為叢集的預設角色。

將 IAM 角色設為叢集的預設角色

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單上，選擇叢集。 AWS 區域 系統會列出目前 中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。

3. 選擇您要設定預設 IAM 角色的叢集。
4. 在叢集許可下，從關聯的 IAM 角色中，選擇要設為叢集預設的 IAM 角色。
5. 在設定預設值下，選擇設為預設。
6. 出現提示時，選擇設定預設值以確認將指定的 IAM 角色設為預設角色。

讓 IAM 角色不再是叢集的預設角色

您可以讓 IAM 角色不再做為叢集的預設角色。

清除 IAM 角色不再做為叢集預設的預設角色

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽選單上，選擇叢集。 AWS 區域 系統會列出目前 中您帳戶的叢集。每個叢集的屬性子集會在清單中分欄顯示。
3. 選擇您要與 IAM 角色建立關聯的叢集。
4. 在叢集許可下，從關聯的 IAM 角色中選擇預設 IAM 角色。
5. 在設定預設值下，選擇清除預設值
6. 出現提示時，選擇清除預設值以確認清除指定的 IAM 角色不再做為預設角色。

使用 管理在叢集上建立的 IAM 角色 AWS CLI

您可以使用 AWS CLI管理在叢集上建立的 IAM 角色。

建立 Amazon Redshift 叢集並將 IAM 角色設定為預設角色

若要使用 IAM 角色建立 Amazon Redshift 叢集，將其設定為叢集的預設值，請使用 `aws redshift create-cluster` AWS CLI 命令。

下列 AWS CLI 命令會建立 Amazon Redshift 叢集和名為 myrole1 的 IAM 角色。 AWS CLI 命令也會將 myrole1 設定為叢集的預設值。

```
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'
```

下列程式碼片段為回應範例。

```
{
    "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"
            }
        ]
        ...
    }
}
```

將一或多個 IAM 角色新增至 Amazon Redshift 叢集

若要新增與叢集相關聯的一或多個 IAM 角色，請使用 `aws redshift modify-cluster-iam-roles` AWS CLI 命令。

下列 AWS CLI 命令會將 `myrole3`和 `myrole4` 新增至叢集。

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier mycluster \
    --add-iam-roles 'arn:aws:iam::012345678910:role/myrole3'
 'arn:aws:iam::012345678910:role/myrole4'
```

下列程式碼片段為回應範例。

```
{
    "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"
            }
        ],
        ...
    }
}
```

從 Amazon Redshift 叢集移除一或多個 IAM 角色

若要移除與叢集相關聯的一或多個 IAM 角色，請使用 `aws redshift modify-cluster-iam-roles` AWS CLI 命令。

下列 AWS CLI 命令會從叢集移除 `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'
```

下列程式碼片段為回應範例。

```
{
    "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"
            }
        ],
        ...
    }
}
```

將關聯的 IAM 角色設定為叢集的預設角色

若要將關聯的 IAM 角色設定為叢集的預設值，請使用 `aws redshift modify-cluster-iam-roles` AWS CLI 命令。

下列 AWS CLI 命令會將 `myrole2`設定為叢集的預設值。

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier mycluster \
    --default-iam-role-arn 'arn:aws:iam::012345678910:role/myrole2'
```

下列程式碼片段為回應範例。

```
{
    "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"
            }
        ],
        ...
    }
}
```

將未關聯的 IAM 角色設定為叢集的預設角色

若要將未關聯的 IAM 角色設定為叢集的預設值，請使用 `aws redshift modify-cluster-iam-roles` AWS CLI 命令。

下列 AWS CLI 命令會`myrole2`新增至 Amazon Redshift 叢集，並將其設定為叢集的預設值。

```
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'
```

下列程式碼片段為回應範例。

```
{
    "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"
            }
        ],
        ...
    }
}
```

從快照還原叢集並將 IAM 角色設定為其預設值

從快照還原叢集時，您可以關聯現有的 IAM 角色，也可以建立新的 IAM 角色並將其設定為叢集的預設角色。

若要從快照還原 Amazon Redshift 叢集，並將 IAM 角色設定為叢集預設值，請使用 `aws redshift restore-from-cluster-snapshot` AWS CLI 命令。

下列 AWS CLI 命令會從快照還原叢集，並將 `myrole2`設定為叢集的預設值。

```
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'
```

下列程式碼片段為回應範例。

```
{
    "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"
```

```
            }
        ],
        ...
    }
}
```

使用聯合身分來管理 Amazon Redshift 對本機資源和 Amazon Redshift Spectrum 外部資料表的存取。

在 中使用聯合身分 AWS 搭配 提供的登入資料，`GetDatabaseCredentials`可以簡化對本機資料和外部資料的授權和存取。在本教學課程中，我們會示範如何透過 AWS 聯合身分提供 資源的存取權，而不是使用特定的 IAM 角色。

目前，為了讓使用者能夠存取 Amazon S3 中的外部資料，您可以使用許可政策中定義的許可來建立 IAM 角色。然後，具有連接角色的使用者可以存取外部資料。這是可行的，但如果您想提供精細的規則，例如讓特定使用者無法使用特定的欄，您可能必須對外部結構描述進行其他設定。

使用 提供的登入資料，聯合身分`GetDatabaseCredentials`可以使用更易於指定 AWS Glue 和變更的精細 IAM 規則來提供對 和 Redshift Spectrum 資源的存取。這可讓您更輕鬆地應用符合業務規則的存取權限。

使用聯合憑證的好處如下：

- 您無需為 Redshift Spectrum 管理叢集連接的 IAM 角色。
- 叢集管理員可以建立具有不同 IAM 內容的消費者可存取的外部結構描述。舉例來說，這對於在資料表上執行欄篩選非常有用，其中不同的消費者查詢相同的外部結構描述，並在傳回的記錄中取得不同的欄位。
- 您可以使用具有 IAM 許可的使用者來查詢 Amazon Redshift，而不是僅使用某個角色。

準備身分以使用聯合身分登入

使用聯合身分登入之前，您必須先執行幾個初步步驟。這些說明假設您有一個現有的 Redshift Spectrum 外部結構描述，該結構描述參考儲存在 Amazon S3 儲存貯體中的資料檔案，且該儲存貯體與您的 Amazon Redshift 叢集或 Amazon Redshift Serverless 資料倉儲位於同一帳戶中。

1. 建立 IAM 身分。這可以是使用者或 IAM 角色。使用 IAM 支援的任何名稱。
2. 將許可政策連接至身分。指定下列任一項：

    - `redshift:GetClusterCredentialsWithIAM` (適用於 Amazon Redshift 佈建的叢集)

- `redshift-serverless:GetCredentials` (適用於 Amazon Redshift Serverless)

您可以使用 IAM 主控台透過政策編輯器新增許可。

IAM 身分還需要許可才能存取外部資料。直接新增下列 AWS 受管政策，以授予 Amazon S3 的存取權：

- `AmazonS3ReadOnlyAccess`
- `AWSGlueConsoleFullAccess`

如果您使用 AWS Glue 準備外部資料，則需要最後一個受管政策。如需授予 Amazon Redshift Spectrum 存取權的步驟相關資訊，請參閱為 Amazon Redshift 建立 IAM 角色，這是 Amazon Redshift 和 Redshift Spectrum 入門指南的一部分。它顯示了新增 IAM 政策以存取 Redshift Spectrum 的步驟。

3. 設定您的 SQL 用戶端以連接到 Amazon Redshift。使用 Amazon Redshift JDBC 驅動器，並將使用者的憑證新增至工具的憑證屬性。像 SQL Workbench/J 這樣的用戶端非常適合此目的。設定下列用戶端連線延伸屬性：

   - AccessKeyID – 您的存取金鑰識別碼。
   - SecretAccessKey – 您的私密存取金鑰。(如果不使用加密，請注意傳輸金鑰的安全風險。)
   - SessionToken – IAM 角色的一組暫時憑證。
   - groupFederation - 如果您要為已佈建的叢集設定聯合身分，則設定為 `true`。如果您使用的是 Amazon Redshift Serverless，請勿設定此參數。
   - LogLevel - 整數日誌層級值。這是選用的。

4. 將 URL 設定為在 Amazon Redshift 或 Amazon Redshift Serverless 主控台中找到的 JDBC 端點。將您的 URL 結構描述替換為 jdbc:redshift:iam: 並使用以下格式：

   - Amazon Redshift 佈建叢集的格式：`jdbc:redshift:iam://<cluster_id>.<unique_suffix>.<region>.redshift.amazonaws.com:<port>/<database_name>`

     範例：`jdbc:redshift:iam://test1.12345abcdefg.us-east-1.redshift.amazonaws.com:5439/dev`
   - Amazon Redshift Serverless 的格式：`jdbc:redshift:iam://<workgroup-name>.<account-number>.<aws-region>.redshift-serverless.amazonaws.com:5439:<port>/<database_name>`

範例：jdbc:redshift:iam://default.123456789012.us-east-1.redshift-serverless.amazonaws.com:5439/dev

在您第一次使用 IAM 身分連線到資料庫之後，Amazon Redshift 會自動建立具有相同名稱的 Amazon Redshift 身分，並為使用者加上 IAM: 字首或為 IAM 角色加上 IAMR: 字首。本主題中的其餘步驟顯示使用者的範例。

如果未自動建立 Redshift 使用者，您可以使用管理員帳戶執行 CREATE USER 陳述式，並以格式 IAM:<user name> 指定使用者名稱來建立一個使用者。

5. 身為 Amazon Redshift 叢集管理員，請授予 Redshift 使用者存取外部結構描述所需的許可。

```
GRANT ALL ON SCHEMA my_schema to "IAM:my_user";
```

若要授予 Redshift 使用者在外部結構描述中建立資料表的能力，他們必須是結構描述擁有者。例如：

```
ALTER SCHEMA my_schema owner to "IAM:my_user";
```

6. 若要驗證組態，請在授予許可之後，使用 SQL 用戶端以使用者身分執行查詢。此查詢範例會從外部資料表擷取資料。

```
SELECT * FROM my_schema.my_table;
```

開始將身分和授權傳播到 Redshift Spectrum

若要傳遞聯合身分來查詢外部資料表，請將 SESSION 設定為 CREATE EXTERNAL SCHEMA 的 IAM_ROLE 查詢參數值。下列步驟說明如何設定及運用 SESSION 來授權對外部結構描述進行查詢。

1. 建立本機資料表和外部資料表。使用 AWS Glue 編目的外部資料表可以用於此目的。
2. 使用您的 IAM 身分連接到 Amazon Redshift。如上一節所述，當身分連接到 Amazon Redshift 時，便會建立一個 Redshift 資料庫使用者。如果先前不存在，則會建立使用者。如果使用者是新使用者，管理員必須授予他們在 Amazon Redshift 中執行任務 (例如查詢和建立資料表) 的許可。
3. 使用您的管理員帳戶連接到 Redshift。執行命令以使用 SESSION 值建立外部結構描述。

```
create external schema spectrum_schema from data catalog
database '<my_external_database>'
```

```
region '<my_region>'
iam_role 'SESSION'
catalog_id '<my_catalog_id>';
```

請注意，在這種情況下會設定 catalog_id。這是與功能一起新增的設定，因為 SESSION 會取代特定角色。

在此範例中，查詢中的值模擬真實值的顯示方式。

```
create external schema spectrum_schema from data catalog
database 'spectrum_db'
region 'us-east-1'
iam_role 'SESSION'
catalog_id '123456789012'
```

在此情況下catalog_id的值是 AWS 您的帳戶 ID。

4. 使用您在步驟 2 中連接的 IAM 身分，執行查詢以存取外部資料。例如：

```
select * from spectrum_schema.table1;
```

在這種情況下，table1 可以是 Amazon S3 儲存貯體中檔案中的 JSON 格式資料。

5. 如果您已經擁有使用叢集連接 IAM 角色的外部結構描述 (指向外部資料庫或結構描述)，則可以取代現有結構描述並使用這些步驟中詳述的聯合身分，或建立新的結構描述。

SESSION 表示聯合身分憑證用於查詢外部結構描述。當您使用 SESSION 查詢參數時，請務必設定 catalog_id。這是必要的，因為它指向用於結構描述的資料目錄。先前，catalog_id 是從指派給 iam_role 的值中擷取。當您以這種方式設定身分識別和授權傳播時，例如，透過使用聯合憑證查詢外部結構描述，就不需要透過 IAM 角色進行授權。

使用須知

一個常見的連線錯誤如下：IAM 擷取臨時憑證時發生錯誤：無法使用提供的解組器取消封送異常回應。此錯誤是具有舊版 JDBC 驅動器的結果。聯合身分所需的最低驅動程式版本為 2.1.0.9。您可以從下載 Amazon Redshift JDBC 驅動器 2.x 版取得 JDBC 驅動器。

其他資源

這些連結提供管理外部資料存取權的其他資訊。

- 您仍然可以使用 IAM 角色存取 Redshift Spectrum 資料。如需詳細資訊，請參閱授權 Amazon Redshift 代表您存取 AWS 服務。
- 當您使用 AWS Lake Formation管理對外部資料表的存取權時，您可以使用 Redshift Spectrum 搭配聯合 IAM 身分來查詢。您不再需要為 Redshift Spectrum 管理叢集連接的 IAM 角色，就能查詢向 AWS Lake Formation註冊的資料。如需詳細資訊，請參閱將 AWS Lake Formation 與 Amazon Redshift Spectrum 搭配使用。

使用 管理 Amazon Redshift 管理員密碼 AWS Secrets Manager

Amazon Redshift 可與 整合 AWS Secrets Manager ，以在加密的秘密內產生和管理管理員憑證。使用 時 AWS Secrets Manager，您可以將管理員密碼取代為 API 呼叫，以程式設計方式在需要時擷取秘密。使用機密而非硬式編碼憑證可降低這些憑證遭到洩露或遭到入侵的風險。如需 的詳細資訊 AWS Secrets Manager，請參閱AWS Secrets Manager 《 使用者指南》 。

您可以在執行下列其中一個操作 AWS Secrets Manager 時，指定 Amazon Redshift 使用 管理您的管理員密碼：

- 建立佈建叢集或無伺服器命名空間
- 編輯、更新或修改佈建叢集或無伺服器命名空間的管理員憑證
- 從快照還原叢集或無伺服器命名空間

當您指定 Amazon Redshift 管理 中的管理員密碼時 AWS Secrets Manager，Amazon Redshift 會產生密碼並將其存放在 Secrets Manager 中。您可以直接在 中存取秘密 AWS Secrets Manager ，以擷取管理員使用者的登入資料。或者，如果您需要從另一個 AWS 帳戶存取秘密，您可以指定客戶受管金鑰來加密秘密。您也可以使用 AWS Secrets Manager 提供的 KMS 金鑰。

Amazon Redshift 會管理機密的設定，並依預設每 30 天輪換一次密碼。您可以隨時手動輪換機密。如果您刪除管理秘密的佈建叢集或無伺服器命名空間 AWS Secrets Manager，則也會刪除秘密及其相關聯的中繼資料。

若要使用秘密管理的登入資料連線至叢集或無伺服器命名空間，您可以使用 AWS Secrets Manager Secrets Manager 主控台或 Secrets Manager API 呼叫從 擷取`GetSecretValue`秘密。如需詳細資訊，請參閱AWS Secrets Manager 《 使用者指南》中的從 擷取秘密 AWS Secrets Manager，並使用 AWS Secrets Manager 秘密中的登入資料連線至 SQL 資料庫。

AWS Secrets Manager 整合所需的許可

使用者必須具有必要的權限，才能執行與 AWS Secrets Manager 整合相關的操作。建立 IAM 政策，授予對其需要的指定資源執行特定 API 操作的許可。然後，將這些政策附加至需要這些許可的 IAM 許可集或角色。如需詳細資訊，請參閱Amazon Redshift 中的身分和存取管理。

指定 Amazon Redshift 在 中管理管理員密碼的使用者 AWS Secrets Manager 必須具有執行下列操作的許可：

- `secretsmanager:CreateSecret`
- `secretsmanager:RotateSecret`
- `secretsmanager:DescribeSecret`
- `secretsmanager:UpdateSecret`
- `secretsmanager:DeleteSecret`
- `secretsmanager:GetRandomPassword`
- `secretsmanager:TagResource`

如果使用者想要在佈建叢集的 `MasterPasswordSecretKmsKeyId` 參數中傳遞 KMS 金鑰，或在無伺服器命名空間的 `AdminPasswordSecretKmsKeyId` 參數中傳遞 KMS 金鑰，除了上面所列的許可外，其還需要下列許可。

- `kms:Decrypt`
- `kms:GenerateDataKey`
- `kms:CreateGrant`
- `kms:RetireGrant`

管理員密碼機密輪換

根據預設，Amazon Redshift 會每 30 天自動輪換您的機密，以確保您的憑證不會長時間保持不變。當 Amazon Redshift 輪換管理員密碼秘密時， 會 AWS Secrets Manager 更新現有的秘密以包含新的管理員密碼。Amazon Redshift 會變更叢集的管理員密碼，以符合已更新之機密中的密碼。

您可以使用 AWS Secrets Manager立即輪換機密，而無需等待排程的輪換。如需輪換機密的相關資訊，請參閱《AWS Secrets Manager 使用者指南》中的輪換 AWS Secrets Manager 機密。

搭配使用 AWS Secrets Manager 與 Amazon Redshift 的考量事項

使用 AWS Secrets Manager 管理佈建叢集或無伺服器命名空間的管理員登入資料時，請考慮下列事項：

- 當您暫停由其管理員登入資料管理的叢集時 AWS Secrets Manager，將不會刪除叢集的秘密，您將繼續支付秘密的費用。只有刪除叢集時才會刪除機密。
- 如果您的叢集在 Amazon Redshift 嘗試輪換其附加的機密時暫停了，輪換會失敗。在這種情況下，即使您繼續叢集，Amazon Redshift 還是會停止自動輪換，而且不會嘗試再次輪換。您必須使用 `secretsmanager:RotateSecret` API 呼叫重新啟動自動輪換排程，才能繼續讓 AWS Secrets Manager 自動輪換機密。
- 如果在 Amazon Redshift 嘗試輪換其附加的機密時，您的無伺服器命名空間沒有相關聯的工作群組，則輪換會失敗，且不會嘗試再次輪換，即使在您附加工作群組之後也是如此。您必須使用 `secretsmanager:RotateSecret` API 呼叫重新啟動自動輪換排程，才能繼續讓 AWS Secrets Manager 自動輪換機密。

在 Amazon Redshift 中擷取機密的 Amazon Resource Name (ARN)

您可以使用 Amazon Redshift 主控台，檢視由 AWS Secrets Manager 管理之任何機密的 Amazon Resource Name (ARN)。取得秘密的 ARN 後，您可以使用 檢視秘密的詳細資訊，以及秘密中的加密資料 AWS Secrets Manager。如需使用 ARN 擷取機密的相關資訊，請參閱《AWS Secrets Manager 使用者指南》中的擷取機密。

檢視 Amazon Redshift 佈建叢集之機密的詳細資訊

請透過下列程序，使用 Amazon Redshift 主控台檢視叢集機密的 Amazon Resource Name (ARN)：

1. 登入 AWS 管理主控台 並開啟 Amazon Redshift 主控台。
2. 在叢集概觀窗格中，選擇您要檢視其機密的叢集。
3. 選擇屬性索引標籤。
4. 在管理員憑證 ARN 下檢視機密的 ARN。此 ARN 是秘密的識別符，您可以在 中使用它 AWS Secrets Manager 來檢視秘密的詳細資訊。

檢視 Amazon Redshift Serverless 命名空間之機密的詳細資訊

請透過下列程序，使用 Amazon Redshift 主控台檢視無伺服器命名空間機密的 Amazon Resource Name (ARN)：

1. 登入 AWS 管理主控台 並開啟 Amazon Redshift 主控台。
2. 從佈建叢集儀表板中，選擇頁面右上角的前往無伺服器。
3. 在 Serverless 儀表板中，捲動至命名空間/工作群組窗格，然後選擇您要檢視其機密的命名空間。
4. 在一般資訊窗格中，於管理員憑證 ARN 下檢視機密的 ARN。此 ARN 是秘密的識別符，您可以在 中使用它 AWS Secrets Manager 來檢視秘密的詳細資訊。

建立資料庫連線憑證的機密

您可以建立 Secrets Manager 機密，以儲存用來連線至 Amazon Redshift 佈建叢集或 Redshift Serverless 命名空間和工作群組的憑證。您在 Amazon Redshift 查詢編輯器 v2 中排程查詢時，也可以使用此機密。

使用 Secrets Manager 主控台在 Amazon Redshift 佈建叢集中建立資料庫的機密

1. 開啟 Secrets Manager 主控台 (https://console.aws.amazon.com/secretsmanager/)。
2. 瀏覽至機密清單，並選擇儲存新機密。
3. 選擇 Amazon Redshift 資料倉儲的憑證。在建立機密的步驟中輸入您的資訊，如下所示：
   - 在使用者名稱的憑證中，輸入資料倉儲管理使用者的名稱。
   - 在密碼的憑證中，輸入使用者名稱的密碼。
   - 針對加密金鑰，選擇您的加密金鑰。
   - 針對資料倉儲，選擇包含您的資料的 Amazon Redshift 佈建叢集。
   - 針對機密名稱，輸入機密的名稱。
   - 針對說明，輸入機密的說明。
   - 針對標籤，輸入包含 **`Redshift`** 這個字的標籤索引鍵。當您嘗試使用 Amazon Redshift 查詢編輯器 v2 連線至資料倉儲時，須有此標籤索引鍵才能列出機密。機密必須具有開頭為字串 **`Redshift`** 的標籤索引鍵，您才能在管理主控台的 AWS Secrets Manager 下列出機密。
4. 繼續進行數個步驟並輸入有關機密的資訊，直到您在檢閱步驟中儲存變更為止。

   憑證、引擎、主機、連接埠和叢集識別碼的特定值都會儲存在機密中。此外，機密會加上具有標籤索引鍵 `Redshift` 的標籤。

使用 Redshift Serverless 主控台在 Redshift Serverless 命名空間中建立資料庫的機密

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 選擇 Redshift 無伺服器並瀏覽至命名空間組態。
3. 選擇要為其建立機密憑證的命名空間。
4. 開啟動作、編輯管理員憑證。
5. 針對管理員密碼，選擇在 AWS Secrets Manager中管理管理員憑證。
6. 選擇儲存變更，以儲存您所做的變更。

確認出現密碼變更成功的訊息。您也可以在 Secrets Manager 主控台中檢視機密。您可以使用此秘密，透過連線方法，在 Redshift Serverless 主控台和 Amazon Redshift 查詢編輯器 v2 中 AWS Secrets Manager 連線至工作群組中的資料庫。機密必須具有開頭為字串 "Redshift" 的標籤索引鍵，該機密才能在查詢編輯器 v2 Web 應用程式中列出。秘密必須具有以字串開頭的標籤金鑰**`Redshift`**，才能在 管理主控台 AWS Secrets Manager 的 下列出秘密。

使用 Secrets Manager 主控台在 Redshift Serverless 命名空間中建立資料庫的機密

1. 開啟 Secrets Manager 主控台 (https://console.aws.amazon.com/secretsmanager/)。
2. 瀏覽至機密清單，並選擇儲存新機密。
3. 選擇 Amazon Redshift 資料倉儲的憑證。在建立機密的步驟中輸入您的資訊，如下所示：
   - 在使用者名稱的憑證中，輸入資料倉儲管理使用者的名稱。
   - 在密碼的憑證中，輸入使用者名稱的密碼。
   - 針對加密金鑰，選擇您的加密金鑰。
   - 針對資料倉儲，選擇包含您的資料的 Redshift Serverless 命名空間。
   - 針對機密名稱，輸入機密的名稱。
   - 針對說明，輸入機密的說明。
   - 針對標籤，輸入包含 **`Redshift`** 這個字的標籤索引鍵。當您嘗試使用 Amazon Redshift 查詢編輯器 v2 連線至資料倉儲時，須有此標籤索引鍵才能列出機密。機密必須具有開頭為字串 **`Redshift`** 的標籤索引鍵，您才能在管理主控台的 AWS Secrets Manager 下列出機密。
4. 繼續進行數個步驟並輸入有關機密的資訊，直到您在檢閱步驟中儲存變更為止。

   憑證、資料庫名稱、主機、連接埠、命名空間和引擎的特定值都會儲存在機密中。此外，機密會加上具有標籤索引鍵 `Redshift` 的標籤。

使用 在 Redshift Serverless 命名空間中建立資料庫的秘密 AWS CLI

您可以使用 AWS CLI 來建立秘密。其中一種方法是使用 AWS CloudShell 執行 Secrets Manager AWS CLI 命令，如下所示。您必須有適當許可才能執行下列程序中顯示的 AWS CLI 命令。

1. 在 AWS 主控台上，開啟 AWS CloudShell 命令提示字元。如需詳細資訊 AWS CloudShell，請參閱AWS CloudShell 《 使用者指南》中的什麼是 AWS CloudShell 。
2. 例如，針對機密 `MyTestSecret` 輸入 Secrets Manager 命令，以儲存用來連線至資料庫或排程 Amazon Redshift 查詢編輯器 v2 查詢的機密。將命令中的下列值取代為您環境的值：

   - *admin* 是資料倉儲的管理員使用者名稱。
   - *passw0rd* 是管理員的密碼。
   - *dev* 是資料倉儲中的初始資料庫名稱。
   - *region* 是 AWS 區域 包含資料倉儲的 。例如 `us-east-1`。
   - *123456789012* 是 AWS 帳戶。
   - *namespace-id* 是類似 `c3928f0e-c889-4d2b-97a5-5738324d5d3e` 的命名空間識別碼。您可以在 Amazon Redshift 主控台詳細資訊頁面上找到無伺服器命名空間的此識別碼。

```
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\"}]"
```

在 Amazon Redshift 中記錄和監控

監控是維護 Amazon Redshift 和 AWS 解決方案可靠性、可用性和效能的重要部分。您可以從 AWS 解決方案的所有部分收集監控資料，以便在發生多點失敗時更輕鬆地偵錯。 AWS 提供數種工具來監控 Amazon Redshift 資源並回應潛在事件：

Amazon CloudWatch 警示

使用 Amazon CloudWatch 警示，您可在自己指定的一段時間內監看單一指標。如果指標超過指定的閾值，則會將通知傳送至 Amazon SNS 主題或 AWS Auto Scaling 政策。CloudWatch 警示不會

因為處於特定狀態而調用動作。必須是狀態已變更並維持了所指定的時間長度，才會呼叫動作。如需詳細資訊，請參閱建立 警示。如需指標的清單，請參閱 Amazon Redshift 中的效能資料。

AWS CloudTrail 日誌

CloudTrail 提供由使用者、IAM 角色或 Amazon Redshift 中的 AWS 服務所採取的 API 操作記錄。您可以利用 CloudTrail 所收集的資訊來判斷向 Amazon Redshift 發出的請求，以及發出請求的 IP 地址、人員、時間和其他詳細資訊。如需詳細資訊，請參閱使用 CloudTrail 進行記錄。

資料庫稽核記錄日誌

Amazon Redshift 會記錄您資料庫中連線和使用者活動的相關資訊。這些日誌可協助您基於安全和故障診斷之目的監控資料庫，此程序稱為資料庫稽核。日誌可以儲存在：

- Amazon S3 儲存貯體 — 這會讓負責監控資料庫中活動的使用者能夠存取資料安全功能。
- Amazon CloudWatch — 您可以使用 CloudWatch 內建的功能 (例如視覺化功能和動作設定) 來檢視稽核記錄資料。

> **Note**
>
> SYS_CONNECTION_LOG 會收集 Amazon Redshift Serverless 的連線日誌資料。請注意，當您收集 Amazon Redshift Serverless 的稽核記錄資料時，並無法將資料傳送至日誌檔，只能傳送至 CloudWatch。

主題

- Amazon Redshift 日誌
- 稽核日誌和 Amazon CloudWatch
- 啟用稽核記錄
- 安全記錄

Amazon Redshift 日誌

Amazon Redshift 會將資訊記錄在下列日誌檔中：

- 連線日誌 — 記錄身分驗證嘗試、連線和中斷連線。
- 使用者日誌 — 記錄對資料庫使用者定義所做之變更的相關資訊。

- 使用者活動日誌 — 在每個查詢於資料庫上執行之前先行記錄。

連線和使用者日誌主要用於安全目的。您可以使用連線日誌，來監控正在連線至資料庫的使用者相關資訊，以及相關的連線資訊。此資訊可能是使用者的 IP 地址、使用者何時提出請求、使用者使用哪種類型的身分驗證等等。您可以利用使用者日誌，來監控對資料庫使用者之定義所做的變更。

使用者活動日誌主要用於故障診斷目的。它會追蹤使用者和系統在資料庫中執行之查詢類型的相關資訊。

連線日誌和使用者日誌對應至儲存在資料庫中系統資料表的資訊。您可以使用系統資料表來取得相同資訊，但是日誌檔提供更簡單的擷取和檢閱機制。日誌檔依賴 Amazon S3 許可而非資料庫許可，針對資料表執行查詢。此外，藉由檢視日誌檔中的資訊，而非查詢系統資料表，您可以減少與資料庫互動的任何影響。

> **Note**
>
> 日誌檔不如系統日誌資料表 STL_USERLOG 和 STL_CONNECTION_LOG 一樣新。比最新記錄舊的記錄 (但不含最新記錄) 會複製到日誌檔。

> **Note**
>
> 若為 Amazon Redshift Serverless，SYS_CONNECTION_LOG 會收集連線日誌資料。當您收集 Amazon Redshift Serverless 的稽核記錄資料時，並無法將資料傳送至日誌檔，只能傳送至 CloudWatch。

連線日誌

記錄身分驗證嘗試以及連線和中斷連線。下表說明連線日誌中的資訊。如需這些欄位的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 STL_CONNECTION_LOG。如需 Amazon Redshift Serverless 所收集連線日誌資料的相關資訊，請參閱 SYS_CONNECTION_LOG。

欄名稱	說明
事件	連線或身分驗證事件。
recordtime	發生事件的時間。

欄名稱	說明
remotehost	遠端主機的名稱或 IP 地址。
remoteport	遠端主機的連接埠號碼。
pid	與陳述式相關聯的處理程序 ID。
dbname	資料庫名稱。
使用者名稱	使用者名稱。
authmethod	身分驗證方法。
持續時間	連線的持續時間，以微秒為單位。
sslversion	Secure Sockets Layer (SSL) 版本。
sslcipher	SSL 密碼。
mtu	最大傳輸單位 (MTU)。
sslcompression	SSL 壓縮類型。
sslexpansion	SSL 擴展類型。
iamauthguid	AWS CloudTrail 請求的 AWS Identity and Access Management (IAM) 身分驗證 ID。這是用來建立指定連線所用憑證之 GetClusterCredentials API 呼叫的識別碼。
application_name	工作階段應用程式的初始名稱或已更新名稱。
os_version	連線到 Amazon Redshift 叢集之用戶端機器上的作業系統版本。
driver_version	從第三方 SQL 用戶端工具連線到 Amazon Redshift 叢集的 ODBC 或 JDBC 驅動器版本。
plugin_name	連接至您的 Amazon Redshift 叢集時使用的外掛程式名稱。

欄名稱	說明
protocol_version	Amazon Redshift 驅動程式在建立與伺服器的連線時使用的內部通訊協定版本。
sessionid	目前工作階段的全域唯一識別碼。
壓縮	用於連線的壓縮演算法。

使用者日誌

記錄資料庫使用者之下列變更的詳細資訊：

- 建立使用者
- 捨棄使用者
- 更改使用者 (重新命名)
- 更改使用者 (更改屬性)

欄名稱	說明
userid	受到變更影響之使用者的 ID。
使用者名稱	受到變更影響之使用者的使用者名稱。
oldusername	若為重新命名動作，這是原始使用者名稱。若是任何其他動作，此欄位空白。
動作	發生的動作。有效值： • Alter • 建立 • Drop • 重新命名
usecreatedb	若為 true (1)，表示使用者具備建立資料庫許可。
usesuper	若為 true (1)，表示使用者是超級使用者。

欄名稱	說明
usecatupd	若為 true (1)，表示使用者可以更新系統目錄。
valuntil	密碼到期日。
pid	處理程序 ID。
xid	交易 ID。
recordtime	查詢開始的時間，以 UTC 表示。

查詢 SYS_USERLOG 系統檢視可尋找有關使用者變更的其他資訊。此檢視包含來自 Amazon Redshift Serverless 的日誌資料。

使用者活動日誌

在每個查詢於資料庫上執行之前先行記錄。

欄名稱	說明
recordtime	發生事件的時間。
db	資料庫名稱。
user	使用者名稱。
pid	與陳述式相關聯的處理程序 ID。
userid	使用者 ID。
xid	交易 ID。
query	日誌的字首：後面跟著查詢的文字，包括換行字元。

稽核日誌和 Amazon CloudWatch

根據預設，Amazon Redshift 不會開啟稽核記錄。當您在叢集上開啟記錄時，Amazon Redshift 會將日誌匯出至 Amazon CloudWatch，或建立日誌並將其上傳至 Amazon S3，擷取從啟用稽核記錄起到目前為止的資料。每個記錄更新都會接續先前的日誌。

目標為 CloudWatch 或 Amazon S3 的稽核記錄是選擇性程序。目標為系統資料表的記錄則非選用，而且會自動發生。如需目標為系統資料表之記錄的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的系統資料表參考。

使用 、Amazon Redshift API 參考或 AWS Command Line Interface () 一起啟用連線日誌 AWS 管理主控台、使用者日誌和使用者活動日誌AWS CLI。對於使用者活動日誌，您也須啟用 `enable_user_activity_logging` 資料庫參數。如果您只啟用稽核記錄功能，但未啟用相關聯參數，則資料庫稽核日誌將只記錄連線日誌和使用者日誌的日誌資訊，但不記錄使用者活動日誌的日誌資訊。預設不會啟用 `enable_user_activity_logging` 參數 (`false`)。您可以將其設為 `true` 來啟用使用者活動日誌。如需詳細資訊，請參閱Amazon Redshift 參數群組。

當您啟用目標為 CloudWatch 的記錄時，Amazon Redshift 會將叢集連線、使用者和使用者活動日誌資料匯出到 Amazon CloudWatch Logs 日誌群組。日誌資料在結構描述方面不會變更。CloudWatch 專為監控應用程式而建置，您可以使用它來執行即時分析或為其進行設定以採取動作。您可以使用 Amazon CloudWatch Logs 將日誌記錄儲存在耐用的儲存體中。

若不要在 Amazon S3 中存放日誌檔，建議的替代方法是使用 CloudWatch 來檢視日誌。其不需要太多的組態設定，並可能適合您的監控要求，特別是如果您已經使用它來監控其他服務和應用程式的話。

Amazon CloudWatch 中的日誌群組和日誌事件

選取要匯出的 Amazon Redshift 日誌後，您就可以在 Amazon CloudWatch Logs 中監控日誌事件。系統會自動在下列字首下，為 Amazon Redshift Serverless 建立新的日誌群組，其中 `log_type` 代表日誌類型。

```
/aws/redshift/cluster/<cluster_name>/<log_type>
```

例如，如果您選擇匯出連線日誌，則日誌資料會儲存在下列日誌群組中。

```
/aws/redshift/cluster/cluster1/connectionlog
```

日誌事件會使用日誌串流匯出至日誌群組。若要在日誌事件內搜尋無伺服器端點的資訊，請使用 Amazon CloudWatch Logs 主控台、 AWS CLI或 Amazon CloudWatch Logs API。如需有關搜尋和篩選日誌資料的資訊，請參閱使用篩選條件從日誌事件建立指標。

在 CloudWatch 中，您可以使用能提供精細度和彈性的查詢語法來搜尋日誌資料。如需詳細資訊，請參閱 CloudWatch Logs Insights 查詢語法。

遷移到 Amazon CloudWatch 稽核記錄

無論何時，如果您要將日誌傳送到 Amazon S3，而且您變更了組態 (例如，將日誌傳送到 CloudWatch)，留在 Amazon S3 中的日誌不會受到影響。您仍可在日誌資料所在的 Amazon S3 儲存貯體中查詢日誌資料。

Amazon S3 中的日誌檔案

Amazon S3 中 Amazon Redshift 日誌檔案的數目和大小極度取決於叢集中的活動。如果您有作用中叢集，正在產生大量日誌，則 Amazon Redshift 可能更頻繁地產生日誌檔案。對於相同類型的活動，您可能具有一系列日誌檔案，例如在同一小時內具有多個連線日誌。

當 Amazon Redshift 使用 Amazon S3 來儲存日誌時，您在 Amazon S3 中使用的儲存會產生費用。在設定目標為 Amazon S3 的記錄之前，請計劃您需要儲存日誌檔案多長時間。這包括根據您的稽核需求，決定何時可刪除或封存日誌檔案。您建立的計劃極度取決於您儲存的資料類型，例如受制於合規或法規需求的資料。如需 Amazon S3 定價的相關資訊，請前往 Amazon Simple Storage Service (Amazon S3) 定價。

啟用目標為 Amazon S3 的記錄時的限制

稽核記錄有下列限制：

- 您只能使用 Amazon S3 受管金鑰 (SSE-S3) 加密 (AES-256)。
- Amazon S3 儲存貯體必須關閉 S3 物件鎖定功能。

Amazon Redshift 稽核記錄的儲存貯體許可

當您開啟目標為 Amazon S3 的記錄功能時，Amazon Redshift 會收集記錄資訊，並將其上傳到儲存在 Amazon S3 的日誌檔案。您可以使用現有的儲存貯體，或建立新的儲存貯體。Amazon Redshift 需要儲存貯體的以下 IAM 許可：

- `s3:GetBucketAcl` 服務對 Amazon S3 儲存貯體需有讀取許可，才能識別儲存貯體擁有者。
- `s3:PutObject` 服務需有放置物件許可，才能上傳物件。此外，開啟記錄功能的使用者或 IAM 角色還必須具有 Amazon S3 儲存貯體的 `s3:PutObject` 許可。每次上傳日誌，服務就會判斷目前儲存貯體擁有者是否符合啟用記錄時的儲存貯體擁有者。如果這些擁有者不相符，您會收到錯誤。

當您啟用稽核記錄時，如果您選取建立新儲存貯體的選項，系統便會為其套用正確的許可。不過，如果您在 Amazon S3 建立自己的儲存貯體，或使用現有的儲存貯體，則請務必新增包含儲存貯體名稱的儲

存貯體政策。日誌會使用服務主體憑證來傳送。對於大多數 AWS 區域，您可以新增 Redshift 服務主體名稱，https：//*redshift.amazonaws.com*。

儲存貯體政策會使用以下格式。*ServiceName* 和 *BucketName* 是您自己值的預留位置。此外，還要在儲存貯體政策中指定相關聯的動作和資源。

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/*"
            ]
        }
    ]
}
```

下列範例是美國東部 (維吉尼亞北部) 區域的儲存貯體政策，以及名為 AuditLogs 的儲存貯體。

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/*"
            ]
        }
    ]
}
```

預設未啟用的區域 (也稱為「選擇加入」區域) 需要區域特定的服務主體名稱。對於這些區域，服務主體名稱會包含 `redshift.region.amazonaws.com` 格式的區域。例如，對於亞太區域 (香港) 區域，名稱是 *redshift.ap-east-1.amazonaws.com*。如需預設未啟用的區域清單，請參閱《AWS 一般參考》中的管理 AWS 區域。

> **Note**
>
> 區域特定服務主體名稱會對應到叢集所在的區域。

日誌檔案的最佳實務

當 Redshift 將日誌檔案上傳到 Amazon S3 時，大型檔案可以分段上傳。如果分段上傳不成功，檔案的這些分段可能會留在 Amazon S3 儲存貯體中。這可能會導致額外的儲存成本，因此請務必了解分段上傳失敗時會發生什麼情況。如需有關稽核日誌分段上傳的詳細說明，請參閱使用分段上傳來上傳和複製物件和中止分段上傳。

如需建立 S3 儲存貯體和新增儲存貯體政策的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的建立一般用途儲存貯體和 Amazon S3 的儲存貯體政策。

Amazon Redshift 稽核記錄的儲存貯體結構

依預設，Amazon Redshift 會使用下列儲存貯體和物件結構來組織 Amazon S3 儲存貯體中的日誌檔案：

AWSLogs/*AccountID*/*ServiceName*/*Region*/*Year*/*Month*/*Day*/*AccountID_ServiceName_Region*

範例為：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 Amazon S3 金鑰字首，請將字首置於金鑰開頭處。

例如，如果您指定 myprefix 的字首：myprefix/AWSLogs/123456789012/redshift/us-east-1/2013/10/29/123456789012_redshift_us-east-1_mycluster_userlog_2013-10-29T18:01.gz

Amazon S3 金鑰字首不能超過 512 個字元。它不能包含空格 ( )、雙引號 (“)、單引號 (‘)、反斜線 (\)。也有一些不允許的特殊字元和控制字元。這些字元的十六進位碼如下：

- x00 到 x20
- x22
- x27
- x5c
- x7f 或更大

Amazon S3 中稽核記錄的考量事項

Amazon Redshift 稽核記錄可能會由於下列原因而中斷：

- Amazon Redshift 沒有將日誌上傳至 Amazon S3 儲存貯體的許可。驗證儲存貯體是使用正確 IAM 政策設定的。如需詳細資訊，請參閱Amazon Redshift 稽核記錄的儲存貯體許可。
- 儲存貯體擁有者已變更。當 Amazon Redshift 上傳日誌時，其會驗證儲存貯體擁有者是否與啟用記錄時的儲存貯體擁有者相同。如果儲存貯體擁有者已變更，則 Amazon Redshift 無法上傳日誌，直到您設定另一個用於稽核記錄的儲存貯體為止。
- 找不到儲存貯體。如果在 Amazon S3 中刪除該儲存貯體，Amazon Redshift 便無法上傳日誌。您必須重建儲存貯體，或設定 Amazon Redshift 以將日誌上傳至不同的儲存貯體。

使用 的 API 呼叫 AWS CloudTrail

Amazon Redshift 已與 服務整合 AWS CloudTrail，此服務提供由 Amazon Redshift AWS 中的使用者、角色或服務所採取之動作的記錄。CloudTrail 會將 Amazon Redshift 的所有 API 呼叫擷取為事件。如需 Amazon Redshift 與 整合的詳細資訊 AWS CloudTrail，請參閱使用 CloudTrail 記錄。

您可以獨立於或除了 Amazon Redshift 資料庫稽核記錄之外使用 CloudTrail。

若要進一步了解 CloudTrail，請參閱《AWS CloudTrail 使用者指南》https://docs.aws.amazon.com/awscloudtrail/latest/userguide/。

啟用稽核記錄

設定 Amazon Redshift 以匯出稽核日誌資料。日誌可以匯出到 CloudWatch，或以檔案形式匯出至 Amazon S3 儲存貯體。

使用主控台啟用稽核記錄日誌

主控台步驟

啟用叢集的稽核記錄

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇叢集，然後選擇您要更新的叢集。
3. 選擇屬性索引標籤。在資料庫組態面板上，選擇編輯，然後選擇編輯稽核記錄。
4. 在編輯稽核記錄頁面上，選擇開啟，然後選取 S3 儲存貯體或 CloudWatch。建議您使用 CloudWatch，因為其管理簡單，並且具有實用的資料視覺化功能。
5. 選擇要匯出的日誌。
6. 若要儲存您的選擇，請選擇儲存變更。

安全記錄

當 Amazon Redshift 記錄參考一或多個 AWS Glue Data Catalog 檢視的查詢時，Amazon Redshift 會在記錄有關該查詢的中繼資料時自動遮罩特定系統資料表中的欄位和檢視資料欄。

安全日誌遮罩適用於 Amazon Redshift 在執行符合遮罩條件的查詢時，產生的所有系統資料表和檢視項目。下表列出已套用安全記錄、文字加上 `******` 遮罩，以及數字加上 `-1` 遮罩的系統檢視和欄。用於遮罩文字的星號數目與原始文字中的字元數相符，最多 6 個字元。超過 6 個字元的字串仍會顯示為 6 個星號。

系統資料表	敏感的欄
SYS_EXTERNAL_QUERY_DETAIL	欄：source_type、total_partitions、qualified_partitions、scanned_files、returned_rows、returned_bytes、file_format、file_location、external_query_text、warning_message。

系統資料表	敏感的欄
SYS_EXTERNAL_QUERY_ERROR	欄：file_location、rowid、column_name、original_value、modified_value、trigger、action、action_value、error_code。
SYS_QUERY_DETAIL	欄：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	欄：returned_rows、returned_bytes。
STL_AGGR	欄：rows、bytes、tbl、type。
STL_BCAST	欄：rows、bytes、packets。
STL_DDLTEXT	欄：label、text。
STL_DELETE	欄：rows、tbl。
STL_DIST	欄：rows、bytes、packets。
STL_ERROR	欄：file、linenum、context、error。
STL_EXPLAIN	欄：plannode、info。
STL_FILE_SCAN	欄：name、line、bytes。
STL_HASH	欄：rows、bytes、tbl、est_rows。
STL_HASHJOIN	欄：rows、tbl、num_parts、join_type。
STL_INSERT	欄：rows、tbl。
STL_LIMIT	欄：rows。
STL_MERGE	欄：rows。
STL_MERGEJOIN	欄：rows、tbl。

系統資料表	敏感的欄
STL_NESTLOOP	欄：rows、tbl。
STL_PARSE	欄：rows。
STL_PLAN_INFO	欄：startupcost、totalcost、rows、bytes。
STL_PROJECT	欄：rows、tbl。
STL_QUERY	欄：querytxt。
STL_QUERY_METRICS	欄：max_rows、rows、max_blocks_read、blocks_read、max_blocks_to_disk、blocks_to_disk、max_query_scan_size、query_scan_size。
STL_QUERYTEXT	欄：text。
STL_RETURN	欄：rows、bytes。
STL_S3CLIENT	欄：bucket、key、transfer_size、data_size。
STL_S3CLIENT_ERROR	欄：bucket、key、error、transfer_size。
STL_儲存	欄：rows、bytes、tbl。
STL_SCAN	欄：rows、bytes、fetches、type、tbl、rows_pre_filter、rows_pre_user_filter、perm_table_name、scanned_mega_value。
STL_SORT	欄：rows、bytes、tbl。
STL_SSHCLIENT_ERROR	欄：ssh_username、endpoint、command、error。
STL_TR_CONFLICT	欄：table_id。
STL_UNDONE	欄：table_id。
STL_UNIQUE	欄：rows、type、bytes。

系統資料表	敏感的欄
STL_UTILITYTEXT	欄：label、text。
STL_WINDOW	欄：rows。
STV_BLOCKLIST	欄：col、tbl、num_values、minvalue、maxvalue。
STV_EXEC_STATE	欄：rows、bytes、label。
STV_INFLIGHT	欄：label、text。
STV_LOCKS	欄：table_id。
STV_QUERY_METRICS	欄：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	欄：table_id、table_name。
STV_TBL_PERM	欄：id、name、rows、sorted_rows、temp、block_count、query_scan_size。
STV_TBL_TRANS	欄：id、rows、size。
SVCS_EXPLAIN	欄：plannode、info。
SVCS_PLAN_INFO	欄：rows、bytes。
SVCS_QUERY_SUMMARY	欄：step、rows、bytes、rate_row、rate_byte、label、rows_pre_filter。
SVCS_S3LIST	欄：bucket、prefix、retrieved_files、max_file_size、avg_file_size。
SVCS_S3LOG	欄：message。
SVCS_S3PARTITION_SUMMARY	欄：total_partitions、qualified_partitions、min_assigned_partitions、max_assigned_partitions、avg_assigned_partitions。

系統資料表	敏感的欄
SVCS_S3QUERY_SUMMARY	欄：external_table_name、file_format、s3_scanned_rows、s3_scanned_bytes、s3query_returned_rows、s3query_returned_bytes。
SVL_QUERY_METRICS	欄：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	欄：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	欄：rows、bytes、label、rows_pre_filter。
SVL_QUERY_SUMMARY	欄：rows、bytes、rows_pre_filter。
SVL_S3LIST	欄：bucket、prefix、retrieved_files、max_file_size、avg_file_size。
SVL_S3LOG	欄：message。
SVL_S3PARTITION	欄：rows、bytes、label、rows_pre_filter。
SVL_S3PARTITION_SUMMARY	欄：total_partitions、qualified_partitions、min_assigned_partitions、max_assigned_partitions、avg_assigned_partitions。
SVL_S3QUERY	欄：external_table_name、file_format、s3_scanned_rows、s3_scanned_bytes、s3query_returned_rows、s3query_returned_bytes、files。
SVL_S3QUERY_SUMMARY	欄：external_table_name、file_format、s3_scanned_rows、s3_scanned_bytes、s3query_returned_rows、s3query_returned_bytes。
SVL_S3RETRIES	欄：file_size、location、message。
SVL_SPECTRUM_SCAN_ERROR	欄：location、rowid、colname、original_value、modified_value、trigger、action、action_value、error_code。

系統資料表	敏感的欄
SVL_STATEMENTTEXT	欄：type、text。
SVL_STORED_PROC_CALL	欄：querytxt。
SVL_STORED_PROC_MESSAGES	欄：message、linenum、querytext。
SVL_UDF_LOG	欄：message、funcname。
SVV_DISKUSAGE	欄：name、col、tbl、blocknum、num_values、minvalue、maxvalue。
SVV_QUERY_STATE	欄：rows、bytes、label。
SVV_TABLE_INFO	欄：table_id、table。
SVV_TRANSACTIONS	欄：relation。

如需系統資料表和檢視的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的系統資料表和檢視參考。如需有關 Amazon Redshift 的動態遮罩查詢結果功能的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的動態資料遮罩。如需 AWS Glue Data Catalog 使用 Amazon Redshift 在 中建立檢視的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的AWS Glue Data Catalog 檢視。

使用 CloudTrail 進行記錄

Amazon Redshift、資料共用、Amazon Redshift Serverless、Amazon Redshift Data API 和查詢編輯器 v2 都已與 整合 AWS CloudTrail。CloudTrail 是一種服務，可提供使用者、角色或 AWS 服務在 Amazon Redshift 中採取之動作的記錄。CloudTrail 會將 Amazon Redshift 的所有 API 呼叫擷取為事件。擷取的呼叫包括從 Redshift 主控台進行的呼叫，以及針對 Redshift 操作的程式碼呼叫。

如果您建立 CloudTrail 追蹤，就可以持續傳送 CloudTrail 事件至 Amazon S3 儲存貯體，包括 Redshift 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台中的事件歷史記錄檢視最新事件。使用由 CloudTrail 收集的資訊，您就可以判斷特定事情。這些事情包括向 Redshift 提出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

您可以獨立於或除了 Amazon Redshift 資料庫稽核記錄之外使用 CloudTrail。

若要進一步了解 CloudTrail，請參閱AWS CloudTrail 《使用者指南》。

CloudTrail 中的資訊

當您建立 AWS 帳戶時，會在您的帳戶中開啟 CloudTrail。活動發生時，該活動會記錄在 CloudTrail 事件中，以及事件歷史記錄中的其他服務 AWS 事件。您可以在 AWS 帳戶中檢視、搜尋和下載最近的事件。如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》中的使用 CloudTrail 事件歷史記錄檢視事件。

若要持續記錄您 AWS 帳戶中的事件，包括 Redshift 的事件，請建立追蹤。CloudTrail 會使用「追蹤」將日誌檔案傳送至 Amazon S3 儲存貯體。根據預設，當您在主控台建立線索時，線索會套用到所有 AWS 區域。線索會記錄 AWS 分割區中所有區域的事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》 中的下列主題：

- 建立追蹤的概觀
- CloudTrail 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 從多個區域接收 CloudTrail 日誌檔案，以及從多個帳戶接收 CloudTrail 日誌檔案

CloudTrail 會記錄所有 Amazon Redshift、Amazon Redshift Serverless、資料 API、資料共用和查詢編輯器 v2 動作。例如，對 `AuthorizeDatashare`、`CreateNamespace`、`ExecuteStatement` 以及 `CreateConnection` 動作發出的呼叫會在 CloudTrail 日誌檔案中產生項目。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或使用者憑證提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》中的 CloudTrail userIdentity 元素。

日誌檔案項目

追蹤是一種組態，允許事件以日誌檔案的形式交付至您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌專案。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時

間、請求參數等資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

Amazon Redshift 資料共用範例

以下範例顯示的 CloudTrail 日誌項目會說明 `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 範例

Amazon Redshift Serverless 已與 整合 AWS CloudTrail ，以提供在 Amazon Redshift Serverless 中所採取動作的記錄。CloudTrail 會將 Amazon Redshift Serverless 的所有 API 呼叫擷取為事件。如需 Amazon Redshift Serverless 功能的詳細資訊，請參閱 Amazon Redshift Serverless 功能概觀。

以下範例顯示的是展示 `CreateNamespace` 動作的 CloudTrail 日誌項目。

```
{
    "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 資料 API 範例

以下範例顯示的是展示 ExecuteStatement 動作的 CloudTrail 日誌項目。

```
{
    "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"
}
```

以下範例顯示的是展示 ExecuteStatement 動作、顯示用於等冪性之 clientToken 的 CloudTrail 日誌項目。

```
{
    "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 查詢編輯器 v2 範例

以下範例顯示的是展示 CreateConnection 動作的 CloudTrail 日誌項目。


```
{
    "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 日誌中的 Amazon Redshift IDs

當 Amazon Redshift 為您呼叫其他 AWS 服務時，系統會使用屬於 Amazon Redshift 的帳戶 ID 記錄呼叫。而不會使用您的帳戶 ID 來記錄呼叫。例如，假設 Amazon Redshift call AWS Key Management Service (AWS KMS) 操作，例如 `CreateGrant`、`Encrypt`、 `Decrypt`和 `RetireGrant`來管理叢集上的加密。在此情況下， AWS CloudTrail 會使用 Amazon Redshift 帳戶 ID 記錄呼叫。

呼叫其他 AWS 服務時，Amazon Redshift 會使用下表中的帳戶 ID。

區域	區域	帳戶 ID
美國東部 (維吉尼亞北部) 區域	us-east-1	368064434614
美國東部 (俄亥俄) 區域	us-east-2	790247189693
美國西部 (加利佛尼亞北部) 區域	us-west-1	703715109447
美國西部 (奧勒岡) 區域	us-west-2	473191095985
非洲 (開普敦) 區域	af-south-1	420376844563
亞太地區 (香港) 區域	ap-east-1	651179539253
亞太區域 (海德拉巴)	ap-south-2	297058826802
亞太區域 (雅加達)	ap-southeast-3	623197973179
亞太區域 (馬來西亞) 區域	ap-southeast-5	590184011157
亞太區域 (墨爾本) 區域	ap-southeast-4	945512339897
亞太地區 (孟買) 區域	ap-south-1	408097707231
亞太區域 (紐西蘭)	ap-southeast-6	730335362089
亞太 (大阪) 區域	ap-northeast-3	398671365691
亞太 (首爾) 區域	ap-northeast-2	713597048934
亞太地區 (新加坡) 區域	ap-southeast-1	960118270566
亞太地區 (雪梨) 區域	ap-southeast-2	485979073181
亞太區域 (台北)	ap-east-2	211125526206
亞太區域 (泰國) 區域	ap-southeast-7	767397930036
亞太地區 (東京) 區域	ap-northeast-1	615915377779
加拿大 (中部) 區域	ca-central-1	764870610256

區域	區域	帳戶 ID
加拿大西部 (卡加利) 區域	ca-west-1	830903446466
歐洲 (法蘭克福) 區域	eu-central-1	434091160558
歐洲 (愛爾蘭) 區域	eu-west-1	246478207311
歐洲 (倫敦) 區域	eu-west-2	885798887673
Europe (Milan) Region	eu-south-1	041313461515
歐洲 (巴黎) 區域	eu-west-3	694668203235
歐洲 (西班牙) 區域	eu-south-2	028811157404
歐洲 (斯德哥爾摩) 區域	eu-north-1	553461782468
歐洲 (蘇黎世) 區域	eu-central-2	668912161003
以色列 (特拉維夫) 區域	il-central-1	901883065212
墨西哥 (中部) 區域	mx-central-1	058264411980
Middle East (Bahrain) Region	me-south-1	051362938876
中東 (阿拉伯聯合大公國) 區域	me-central-1	595013617770
南美洲 (聖保羅) 區域	sa-east-1	392442076723

下列範例顯示 Amazon Redshift 呼叫的 AWS KMS Decrypt 操作的 CloudTrail 日誌項目。

```
{

    "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"
```

```
}
```

系統資料表與 S3 資料表整合

使用系統資料表與 Amazon S3 Tables 整合，將 Amazon Redshift 系統資料表資料保留超過 7 天的叢集內保留期。這可協助您符合合規、稽核和可觀測性要求。當您啟用此功能時，Redshift 會以 Apache Iceberg 格式將系統資料表日誌寫入 S3 Tables，並管理分割、壓縮和快照維護。只要您需要進行分析和稽核，您就可以保留此資料。由於資料是以開放的 Iceberg 格式儲存，因此您可以使用 Redshift、Amazon Athena 或任何其他 Iceberg 相容引擎來查詢資料。您也可以將 AI 代理程式技能用於自然語言查詢 （請參閱 GitHub 網站上的AWS 客服人員工具組系統資料表技能)。

當您透過主控台、Redshift CLI 或 SDK 啟用此功能時，Redshift 會立即開始從您選取的系統資料表，以固定頻率將新資料列寫入 S3 Tables。您可以選擇要發佈的系統資料表；S3 Tables 會強制執行您設定的保留期間。

這支援下列使用案例：

- 符合合規和稽核要求。保留查詢、連線和變更歷史記錄數月或數年。
- 從一處監控您的機群。整合來自您帳戶和區域中多個資料倉儲的監控資料，並分析整個機群的活動。
- 移除自訂擷取、轉換和載入 (ETL) 管道的需求。停止建置和維護將系統資料表資料匯出至 S3 的任務，以延長保留時間。
- 調查過去的事件。查詢超過 7 天叢集內時段的歷史資料，以進行根本原因分析、事件後檢閱，以及監控工作負載隨著時間的演變和效能趨勢。

此功能適用於支援 Redshift 和 S3RA33 和 RG 執行個體和 Amazon Redshift Serverless。 https://aws.amazon.com/redshift/redshift-serverless/ AWS 在本主題中，資料倉儲是指佈建叢集或 Redshift Serverless 工作群組。

主題

- 運作方式
- 啟用與 S3 Tables 的系統資料表整合
- 支援的系統資料表
- 新增至每個資料表的中繼資料欄
- 設定保留
- 資料控管

- 交付和生命週期行為
- 最佳實務
- 帳單
- 考量和限制

運作方式

啟用此功能並選取一或多個支援的系統資料表後，Redshift 會將新完成的記錄從這些系統資料表寫入 S3 資料表。資料會以固定頻率分批交付。

系統資料表存在於 AWS 您的帳戶中，但服務受管： AWS 擁有並操作資料交付和資料表管理，包括下列項目：

- 在帳戶中建立 S3 資料表儲存貯體、命名空間和 S3 資料表。
- 定義和發展資料表結構描述。
- 將Apache Iceberg格式的資料寫入 S3 資料表。
- 壓縮資料和維護快照（由 S3 Tables 自動處理）。

無需建置或維護基礎設施，也不會影響工作負載。

您可以設定透過 S3 Tables 記錄過期來保留資料的時間。如需詳細資訊，請參閱the section called “設定保留”。

由於資料表是由 管理 AWS，因此您無法修改或刪除 Redshift 交付的資料列。交付的資料是不可變的。您只控制對資料表的讀取存取權。如需詳細資訊，請參閱the section called “資料控管”。

只有完成的活動才會交付至 S3 Tables。資料倉儲中的系統資料表可以顯示進行中的活動，但只會複製已達到最終狀態的記錄。例如，已交付已完成、已中止或取消的查詢，但在達到最終狀態之前，仍會交付仍在執行中的查詢。

啟用與 S3 Tables 的系統資料表整合

若要啟用與 S3 Tables 的系統資料表整合，請使用具有新 日誌目的地類型的現有 Redshift APIs`s3table`。啟用時，Redshift 會在您的帳戶`aws-redshift`中建立名為 的 S3 資料表儲存貯體。您可以使用與其他日誌目的地相同的操作來管理功能：適用於佈建叢集的 `disable-logging`、、`enable-logging`和 `describe-logging-status`；適用於 Redshift Serverless 的 `update-namespace`和 `get-namespace`。

許可

啟用、修改或停用功能的委託人必須具有下列許可：

- `redshift:EnableLogging`（適用於佈建叢集）或 `redshift-serverless:UpdateNamespace`（適用於 Redshift Serverless)。
- 建立和設定 S3 資料表儲存貯體的許可。啟用主體需要下列 S3 Tables 許可：
  - `s3tables:CreateTableBucket`
  - `s3tables:PutTableBucketEncryption`
  - `s3tables:PutTableBucketPolicy`

  當您啟用此功能時，Redshift 會使用觸發操作的委託人身分來建立和設定 S3 資料表儲存貯體。

建立儲存貯體之後，Redshift 會使用 Redshift 和 S3 Tables 之間的服務信任關係，在其中建立命名空間和資料表。啟用主體不需要建立命名空間或資料表的許可。

部署選項

您可以透過兩種模式之一來設定交付。指定的資料倉儲一次使用一種模式。

部署選項	說明
每個倉儲	Redshift 會將每個資料倉儲的系統資料表資料寫入自己的一組 S3 資料表。資料表中的每一列都來自單一資料倉儲。用於倉儲之間的實體隔離。
合併	Redshift 會將相同帳戶和區域內多個資料倉儲的系統資料表資料寫入一組共用的 S3 資料表。來自不同倉儲的資料列會依 `warehouse_namespace_arn` 和 `warehouse_name` 資料欄區分。將此用於集中式、跨倉儲分析。

部署模型會決定如何在 S3 Tables 中組織資料。在每一倉儲模型中，S3 Tables 命名空間名稱包含資料倉儲的唯一識別符，因此每個倉儲的資料會實際分成自己的命名空間。在合併模型中，S3 Tables 命名空間名稱包含 AWS 帳戶號碼，該帳戶和區域中所有倉儲的資料會一起存放在相同的命名空間和資料表中。

單一 AWS 帳戶和單一 AWS 區域支援這兩個選項。若要跨區域或帳戶分析資料，請在查詢時合併每個區域或帳戶資料表的結果。對於跨帳戶存取，您可以使用 AWS Glue Data Catalog 共用。如需詳細資訊，請參閱《 AWS Glue 開發人員指南》中的授予跨帳戶存取權。

> **Note**
>
> 如果您將資料倉儲從一個部署模式切換到另一個部署模式，則先前交付的資料會保留在其現有的資料表中，並根據其設定的保留來保留。Redshift 開始將新資料交付至新目標，而不會回填歷史資料。

使用主控台

設定與 S3 Tables 整合的系統資料表

1. 開啟 Redshift 主控台，然後在導覽窗格中，在 Amazon Redshift 或 Amazon Redshift Serverless 下，選擇系統資料表整合。您也可以從叢集或命名空間詳細資訊頁面選擇整合索引標籤，或選擇動作、整合、設定系統資料表整合來設定 。
2. 選擇建立系統資料表整合。
3. 選取已佈建的叢集或 Redshift Serverless 工作群組。
4. 選擇要發佈的系統資料表。選取個別SYS_*檢視，或選擇選取所有支援的系統資料表，以發佈所有目前和未來的支援檢視。如果您選取全部，則未來新增的新檢視會自動包含在內，而不需要變更組態。
5. 選取部署模型：
   - 每個資料倉儲每個系統資料表的個別 S3 資料表，將此倉儲的資料保留在其自己的一組資料表中。
   - 跨資料倉儲的每個系統資料表共用 S3 資料表，將帳戶中多個倉儲的資料合併到一組共用資料表。
6. 或者，選擇加密金鑰。根據預設，資料會使用 Amazon S3-managed金鑰 (SSE-S3) 加密。若要使用您自己的 AWS KMS 金鑰，請在此處選取它。如果您使用自己的 AWS KMS 金鑰，金鑰政策必須授予 Redshift 和 S3 Tables 存取金鑰的權限。如需詳細資訊，請參閱the section called “設定加密”。
7. 儲存您的變更。Redshift 開始將選取的系統資料表發佈至 S3 Tables，並繼續定期新增記錄。

若要停止發佈系統資料表，請返回設定並將其移除。已發佈的資料會保留，直到您透過 S3 Tables 設定過期為止。

您可以從叢集或命名空間詳細資訊頁面檢閱整合的狀態。展開檢視 S3 資料表映射，以查看每個系統資料表與其對應 S3 資料表名稱和命名空間之間的映射。您也可以從 S3 Tables 主控台檢視發佈的資料。

使用 AWS CLI

下列參數會設定 S3 Tables 發佈：

參數	適用對象	說明
`--log-destination-type`	兩者	設定為 `s3table`以發佈至 S3 Tables。其他值為 `s3`和 `cloudwatch` 。
`--log-exports`	佈建	要發佈的系統資料表，做為`SYS_*`檢視名稱清單，或 `all`。
`--s3-table-names`	無伺服器	要發佈的系統資料表，做為`SYS_*`檢視名稱清單，或 `all`。
`--s3-table-action`	無伺服器	`Enable` 或 `Disable` S3 Tables 發佈。
`--s3-table-granularity`	兩者	資料表範圍。佈建： `cluster`（預設）或 `account`。無伺服器： `namespace` （預設）或 `account`。
`--s3-table-kms-key-id`	兩者	用於加密的選用 AWS KMS 金鑰 ARN 或 ID。預設為 Amazon S3-managed金鑰 (SSE-S3)。

佈建叢集

啟用交付。 `--log-exports` 接受`all`或以空格分隔的支援系統資料表清單，並接受 `--s3-table-granularity` `cluster`（每個倉儲）或 `account`（合併）。

```
aws redshift enable-logging \
    --cluster-identifier my-redshift-cluster \
    --log-destination-type s3table \
    --log-exports all \
    --s3-table-granularity account
```

僅啟用特定系統資料表的交付：

```
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
```

檢查交付狀態。

```
aws redshift describe-logging-status \
    --cluster-identifier my-redshift-cluster
```

停用所有系統資料表的交付，或使用 的特定資料表的交付`--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 設定每個命名空間的功能`update-namespace`。系統資料表名稱會在 `--s3-table-names` (而非 `--log-exports`) 上傳遞。

啟用交付。

```
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
```

停用特定系統資料表 （ 或 `all`) 的交付：

```
aws redshift-serverless update-namespace \
    --namespace-name my-namespace \
    --log-destination-type s3table \
    --s3-table-action Disable \
    --s3-table-names sys_stream_scan_states
```

檢查交付狀態。

```
aws redshift-serverless get-namespace \
```

```
    --namespace-name my-namespace
```

若要在捨棄 S3 資料表儲存貯體或資料表後繼續交付，請再次執行啟用命令。

向 註冊 AWS Glue Data Catalog

在使用 Redshift、Athena 或其他分析服務查詢保留的資料之前，服務受管 S3 資料表儲存貯體必須與整合 AWS Glue Data Catalog。這是每個帳戶和區域的一次性步驟。如果您已將 S3 Tables 與 整合 AWS Glue Data Catalog，則不需要其他動作。

如需整合說明，請參閱《 AWS Glue 開發人員指南》中的將 Amazon S3 資料表與 整合 AWS Glue Data Catalog。

設定加密

根據預設，傳送到 S3 Tables 的資料會使用 Amazon S3-managed金鑰 (SSE-S3) 加密。若要改用 AWS KMS 客戶受管金鑰，請在啟用交付`--s3-table-kms-key-id`時以 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
```

如果您使用客戶受管金鑰，其金鑰政策必須允許 Redshift 系統資料表整合服務主體在寫入系統資料表時產生資料金鑰，以及 S3 Tables 維護服務主體在資料表維護期間（例如壓縮）使用金鑰。將下列陳述式新增至金鑰政策，以您自己的值取代 *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/*"
        }
      }
    }
  ]
}
```

如果金鑰政策未授予這些許可，則 Redshift 無法寫入加密的 S3 資料表，且交付失敗。

支援的系統資料表

您可以選擇下列任何SYS_*監控檢視來與 S3 Tables 整合。如需每個檢視中資料欄的說明，請選擇檢視名稱。

- 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**
>
> 下列檢視需要修補程式 P203 或更新版本：SYS_CHILD_QUERY_TEXT、SYS_COPY_REPLACEMENTS、SYS_EXTERNAL_QUERY_ERROR、和 SYS_UNLOAD_DETAIL。如果您在執行 P203 之前修補程式的資料倉儲上啟用這些檢視，則會使用正確的結構描述建立資料表，但在資料倉儲更新為 P203 或更新版本之前不會包含任何資料。

新增至每個資料表的中繼資料欄

每個 S3 資料表都包含來源SYS_*檢視中的所有資料欄，以及 Redshift 新增的下列五個中繼資料資料欄，用於識別每一列的來源及其交付時間。

資料行	Type	Description
`warehouse_account_id`	string	擁有來源資料倉儲 AWS 的帳戶。
`warehouse_region_name`	string	來源資料倉儲執行所在的 AWS 區域。
`warehouse_namespace_arn`	string	來源資料倉儲命名空間的 ARN。這是穩定的唯一識別符，在重新命名和重新建立之間是不可變的。
`warehouse_name`	string	來源資料倉儲的人類可讀取名稱（叢集名稱或工作群組名稱）。
`s3_tables_ingestion_time`	timestamp(6)，UTC	Redshift 將資料列遞交至 S3 Tables 的時間。這是交付時間，而不是基礎事件發生的時間。

設定保留

Redshift 不會為其交付至 S3 Tables 的資料定義保留政策。您可以設定透過 S3 Tables 設定記錄過期政策來保留資料的時間。您可以指定天數（例如 5 天、100 天或 365 天），而 S3 Tables 會自動移除超過設定持續時間的資料。

如果您未設定過期政策，資料會無限期保留。

當您停用整合或移除系統資料表時，Redshift 會停止寫入新資料，但先前交付的資料會保留在 S3 Tables 中，並繼續遵守您設定的任何過期政策。

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的設定 S3 資料表儲存貯體的記錄過期。Amazon S3

資料控管

您可以管理對保留系統資料表資料的存取；Redshift 不會代表您套用讀取許可，也不會遮罩資料。

- 存取控制。將 S3 資料表儲存貯體與 整合後 AWS Glue Data Catalog，您可以使用 AWS Identity and Access Management(IAM) 或 來管理讀取存取AWS Lake Formation。在合併部署中，您可以限制每個倉儲的讀取存取。
- 目錄編製和探索。 AWS Glue Data Catalog 整合提供集中位置來探索保留的資料表，並套用精細的許可 （例如，使用 Lake Formation 進行資料表層級、資料欄層級或資料列層級的存取）。

交付和生命週期行為

- 交付頻率。資料會以固定頻率分批交付。
- 完全交付一次。每個記錄都會傳送一次；重新啟用或重新新增系統資料表不會建立重複項目。
- 啟用、停用、重新啟用。當功能 （或特定系統資料表） 停用時，該期間不會擷取任何資料。停用 會停止寫入，但不會移除 S3 資料表或先前交付的資料。重新啟用會繼續交付，而不會填補差距。
- 從您的選擇中移除系統資料表。Redshift 會停止交付該系統資料表的新資料。先前交付的資料會保留在 S3 Tables 中，並受限於您設定的任何記錄過期政策。重新新增系統資料表會繼續交付，而不會複製資料。
- 使用傳輸中的資料刪除資料倉儲。如果您在交付過程中刪除資料倉儲，且資料倉儲的加密金鑰與用於 S3 Tables 的金鑰不同，則 Redshift 可能會在刪除後在資料倉儲的金鑰上使用短暫的 AWS KMS 授予，以完成任何剩餘資料的交付。

檢查交付狀態

您可以隨時檢閱目前的組態和每個系統資料表的最新交付時間：

- 佈建的叢集。 `describe-logging-status` （或 `describe-cluster`) 會傳回作用中的系統資料表、S3 資料表命名空間、精細程度，以及每個系統資料表的上次擷取時間。
- Redshift Serverless。 會在命名空間的 S3 資料表發佈狀態中`get-namespace`傳回相同的資訊。

Redshift Serverless

交付不會讓工作群組保持喚醒或耗用您的運算。在工作群組暫停之前，Redshift 會確保交付任何待定批次。

最佳實務

1. 評估您的資料敏感度。了解資料倉儲中存放哪些類型的資料，以及系統資料表是否包含敏感資訊。有些系統資料表（例如 SYS_QUERY_TEXT 和 SYS_PROCEDURE_MESSAGES) 可以從查詢和預存程序擷取常值。
2. 選擇部署模型。如果您的資料倉儲存放敏感資訊，請考慮使用每個倉儲的部署模型，讓每個資料倉儲的系統資料表資料保持實體隔離。如果您的資料不敏感，而且您想要執行跨倉儲查詢，而不合併多個資料表的結果，請使用合併部署模型。
3. 選取您需要的系統資料表。檢閱支援的系統資料表清單，並選擇符合您合規、稽核或可觀測性需求的資料表。您不需要啟用所有系統資料表。
4. 在啟用功能之前設定加密。如果您想要使用客戶受管 AWS KMS 金鑰，請在第一次啟用此功能時指定它。您無法在建立 S3 Tables 之後變更金鑰。若要變更金鑰，請停用此功能、使用 S3 Tables API 捨棄 S3 Tables（這會永久刪除保留的資料），然後使用新金鑰重新啟用此功能。
5. 根據業務需求設定保留。根據您的合規、稽核或操作需求，在資料表層級直接在 S3 資料表中設定記錄過期政策。不同的資料表可以有不同的保留期間。如果您未設定過期政策，資料會無限期保留。若要監控系統資料表的儲存用量，請參閱《Amazon S3 使用者指南》中的 Amazon S3 Tables CloudWatch 指標。 Amazon S3

帳單

將系統資料表資料寫入 S3 Tables 是免費的。您需要支付保留資料的標準 S3 Tables 儲存和維護費用，以及根據該引擎的定價，用於讀取資料的查詢引擎的費用。如需詳細資訊，請參閱 Amazon S3 Tables 定價。

考量和限制

- 在單一 AWS 帳戶和單一 AWS 區域中支援。若要分析跨帳戶或區域的資料，請在查詢時合併結果。
- 資料倉儲一次使用一個部署模式（每個倉儲或合併）。
- 切換部署模式、停用和重新啟用，或移除和重新新增系統資料表不會回填歷史資料。
- 交付的資料是不可變的。您無法透過 Redshift 修改或刪除個別資料列。
- 您可以捨棄此功能建立的 S3 資料表，但這樣做會永久移除這些資料表中的所有保留資料並停止交付。Redshift 不會自動重新建立捨棄的資料表。若要繼續交付，您必須重新啟用此功能，該功能會建立新的資料表，並開始交付未來的新資料。先前交付的資料不會還原。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的刪除 S3 資料表。 Amazon S3
- 交付是以批次為基礎（以固定頻率寫入資料）。

- 從 Redshift 查詢需要 S3 資料表儲存貯體與 整合 AWS Glue Data Catalog。

Amazon Redshift 的合規驗證

在多個合規計畫中，第三方稽核人員會評估 Amazon Redshift 的安全性和 AWS 合規性。這些計劃包括 SOC、PCI、FedRAMP、HIPAA 等等。

如需特定合規計劃範圍內 AWS 的服務清單，請參閱AWS 合規計劃範圍內的服務。如需一般資訊，請參閱 AWS 合規計劃。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱在 AWS Artifact 中下載報告。

使用 Amazon Redshift 時的合規責任會取決於資料的敏感度、組織的合規目標，以及適用法律和法規。如果您使用 Amazon Redshift 符合 HIPAA、PCI 或 FedRAMP 等標準， AWS 會提供資源來協助：

- 安全與合規快速入門指南，討論部署以安全與合規為中心之基準環境的架構考量和步驟 AWS。
- HIPAA 安全與合規架構白皮書說明公司如何使用 AWS 來建立符合 HIPAA 規範的應用程式。
- 可能適用於您產業和據點的AWS 合規資源、工作手冊和指南。
- AWS Config AWS 服務可以評估資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub CSPM AWS 服務提供 內安全狀態的全方位檢視 AWS ，可協助您檢查是否符合安全產業標準和最佳實務。Security Hub CSPM 使用安全控制來評估資源組態和安全標準，以協助您遵守各種合規架構。如需使用 Security Hub CSPM 評估 Amazon Redshift 資源的詳細資訊，請參閱AWS Security Hub 《 使用者指南》中的 Amazon Redshift 控制項。

下列合規和安全文件涵蓋 Amazon Redshift，可透過 隨需提供 AWS Artifact。如需詳細資訊，請參閱AWS Artifact。

- Cloud Computing Compliance Controls Catalogue (C5)
- ISO 27001:2013 適用性聲明 (SoA)
- ISO 27001:2013 認證
- ISO 27017:2015 適用性聲明 (SoA)
- ISO 27017:2015 認證
- ISO 27018:2015 適用性聲明 (SoA)
- ISO 27018:2014 認證

- ISO 9001:2015 認證
- PCI DSS 合規聲明文件 (AOC) 與責任摘要
- 服務組織控制 (SOC) 1 報告
- 服務組織控制 (SOC) 2 報告
- 服務組織控制 (SOC) 2 報告，針對機密性

Amazon Redshift 中的復原能力

AWS 全球基礎設施是以 AWS 區域和可用區域 (AZs) 為基礎建置。 AWS 區域提供多個實體隔離且隔離的可用區域，這些可用區域以低延遲、高輸送量和高備援聯網連接。透過可用區域，您所設計與操作的應用程式和資料庫，就能夠在可用區域之間自動容錯移轉，而不會發生中斷。可用區域的可用性、容錯能力和擴展性都比傳統單一或多個資料中心的基礎設施高。

幾乎所有 AWS 區域都有多個可用區域和資料中心。您可以在相同區域中的多個可用區域間部署應用程式，以提供容錯能力與低延遲效能。

若要將叢集移至另一個可用區域，而不遺失任何資料或應用程式變更，您可以為叢集設定重新定位。透過重新定位，您可以在叢集上發生服務中斷時繼續操作，將所受影響降到最低。開啟叢集重新定位時，Amazon Redshift 可能會在某些情況下選擇將叢集重新定位。如需 Amazon Redshift 中重新定位的相關資訊，請參閱重新定位叢集。

在可用區域發生意外事件的故障案例中，您可以設定多個可用區域 (多可用區) 部署，以確保 Amazon Redshift 資料倉儲能夠繼續運作。Amazon Redshift 會在可透過單一端點存取的兩個可用區域中部署相等的運算資源。如果整個可用區域發生故障，便可使用第二個可用區域中的剩餘運算資源繼續處理工作負載。如需異地同步備份部署的相關資訊，請參閱多可用區域部署。

如需 AWS 區域和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。如需使用 Amazon Redshift 進行災難復原的詳細資訊，請參閱使用 Amazon Redshift 實作災難復原。

.

Amazon Redshift 的基礎設施安全

Amazon Redshift 是受管服務，受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及如何 AWS 保護基礎設施的資訊，請參閱AWS 雲端安全。若要使用基礎設施安全的最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取 Amazon Redshift。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

網路隔離

以 Amazon VPC 服務為基礎的虛擬私有雲端 (VPC) 是您在 AWS 雲端中以邏輯方式隔離的私有網路。您可以執行下列步驟，在 VPC 內部署 Amazon Redshift 叢集或 Redshift Serverless 工作群組：

- 在 區域中建立 VPC AWS 。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的什麼是 Amazon VPC？。
- 建立兩個或多個私有 VPC 子網路。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 VPC 和子網路。
- 部署 Amazon Redshift 叢集或 Redshift Serverless 工作群組。如需詳細資訊，請參閱 Redshift 資源的子網路 或 工作群組和命名空間 。

依預設會在佈建時鎖定 Amazon Redshift 叢集。若要允許來自 Amazon Redshift 用戶端的傳入網路流量，請將 VPC 安全群組與 Amazon Redshift 叢集相關聯。如需詳細資訊，請參閱Redshift 資源的子網路。

若要僅允許往返特定 IP 位址範圍的流量，請更新您的 VPC 的安全群組。一個例子是僅允許往返您的企業網路的流量。

設定與 Amazon Redshift 叢集加上標籤之子網路相關聯的網路存取控制清單時，請確保將個別 AWS 區域的 S3 CIDR 範圍新增至傳入和傳出規則的允許清單。這樣做可讓您執行以 S3 為基礎的操作，例如 Redshift Spectrum、COPY 和 UNLOAD，而不會造成中斷。

以下範例命令會在 us-east-1 區域中剖析 Amazon S3 所使用的所有 IPv4 地址的 JSON 回應。

```
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
```

如需如何取得特定區域之 S3 IP 範圍的指示，請參閱 AWS IP 地址範圍。

Amazon Redshift 支援將叢集部署至專用租用 VPC。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的專用預留執行個體。

Amazon Redshift 安全群組

當您佈建 Amazon Redshift 叢集時，依預設它會遭鎖定，以便沒有人可以存取它。若要授予其他使用者 Amazon Redshift 叢集的傳入存取，您可使叢集與安全群組建立關聯。如果您是在 EC2-VPC 平台上，則可以使用現有的 Amazon VPC 安全群組，或定義新的安全群組，然後使它與叢集建立關聯。如需在 EC2-VPC 平台上管理叢集的相關資訊，請參閱VPC 中的 Redshift 資源。

介面 VPC 端點

您可以在虛擬私有雲端 (VPC) 中使用介面 VPC 端點 (AWS PrivateLink) 直接連線至 Amazon Redshift 和 Amazon Redshift Serverless API 服務，而非透過網際網路進行連線。如需 Amazon Redshift API 動作的相關資訊，請參閱《Amazon Redshift API 參考》中的動作。如需 Redshift Serverless API 動作的相關資訊，請參閱《Amazon Redshift Serverless API 參考》中的動作。如需 AWS PrivateLink的相關資訊，請參閱《Amazon VPC 使用者指南》中的界面 VPC 端點 (AWS PrivateLink)。請注意，叢集或工作群組的 JDBC/ODBC 連線不屬於 Amazon Redshift API 服務的一部分。

當您使用界面 VPC 端點時，VPC 與 Amazon Redshift 或 Redshift Serverless 之間的通訊會完全在 AWS 網路中執行，這可提供更高的安全性。每個 VPC 端點皆會由一個或多個具私有 IP 地址彈性網路界面來表示，而該界面位於 VPC 子網路中。如需彈性網路界面的詳細資訊，請參閱 Amazon EC2 使用者指南中的彈性網路界面。

界面 VPC 端點會將 VPC 直接連線到 Amazon Redshift。它不使用網際網路閘道、網路位址轉譯 (NAT) 裝置、虛擬私有網路 (VPN) 連線或 Direct Connect 連線。VPC 中的執行個體不需要公有 IP 地址，就能與 Amazon Redshift API 進行通訊。

若要透過 VPC 使用 Amazon Redshift 或 Redshift Serverless，您有兩個選項。一個是從 VPC 內部的執行個體進行連線。另一個方法是使用 Site-to-Site VPN 選項 或 將私有網路連接到 VPC Direct Connect。如需 Site-to-Site VPN 選項的詳細資訊，請參閱《Amazon VPC 使用者指南》中的 VPN 連線。如需有關 Direct Connect的資訊，請參閱《Direct Connect 使用者指南》中的建立連線。

您可以使用 AWS 管理主控台 或 AWS Command Line Interface (AWS CLI) 命令建立介面 VPC 端點以連線至 Amazon Redshift。如需詳細資訊，請參閱建立界面端點。

在您建立界面 VPC 端點後，您可以為該端點啟用私有 DNS 主機名稱。若您這樣做，預設端點如下所示：

- Amazon Redshift 佈建：`https://redshift.Region.amazonaws.com`
- Amazon Redshift Serverless：`https://redshift-serverless.Region.amazonaws.com`

如果您未啟用私有 DNS 主機名稱，Amazon VPC 會透過下列格式提供 DNS 端點名稱供您使用。

- Amazon Redshift 佈建：`VPC_endpoint_ID.redshift.Region.vpce.amazonaws.com`
- Amazon Redshift Serverless：`VPC_endpoint_ID.redshift-serverless.Region.vpce.amazonaws.com`

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的介面 VPC 端點 (AWS PrivateLink)。

Amazon Redshift 和 Redshift Serverless 支援會在 VPC 內呼叫所有 Amazon Redshift API 操作和 Redshift Serverless API 操作。

可以將 VPC 端點政策附接至某個 VPC 端點，以控制 AWS Identity and Access Management (IAM) 主體的存取權。您也可以將安全群組與 VPC 端點建立關聯，以根據網路流量的來源和目的地來控制傳入和傳出存取。範例是 IP 地址範圍。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的使用 VPC 端點控制服務的存取。

Amazon Redshift 的 VPC 端點政策

您可以為 Amazon Redshift 的 VPC 端點建立政策，以指定下列各項：

- 可執行或不可執行動作的主體
- 可執行的動作
- 可在其中執行動作的資源

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的使用 VPC 端點控制對服務的存取。

接下來，您可以找到 VPC 端點政策的範例。

Amazon Redshift 佈建端點政策範例

接下來，您可以找到 Amazon Redshift 佈建的 VPC 端點政策範例。

範例：拒絕來自指定 AWS 帳戶的所有存取的 VPC 端點政策

下列 VPC 端點政策會拒絕 AWS 帳戶使用此端點對資源*123456789012*的所有存取。

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

範例：可用來只向指定 IAM 角色允許 VPC 存取的 VPC 端點政策

下列 VPC 端點政策僅允許完整存取 AWS 帳戶 *123456789012* *redshiftrole*中的 IAM 角色。所有其他 IAM 主體均無法存取該端點。

```
   {
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:role/redshiftrole"
                ]
            }
        }]
}
```

這只是範例。在大多數使用案例中，建議您附加特定動作的許可，以縮小許可範圍。

範例：可用來只向指定 IAM 主體 (使用者) 允許 VPC 存取的 VPC 端點政策

下列 VPC 端點政策僅允許 AWS 帳戶 *123456789012* *redshiftadmin*中的 IAM 使用者完整存取。所有其他 IAM 主體均無法存取該端點。

```
   {
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:user/redshiftadmin"
                ]
            }
        }]
}
```

這只是範例。在大多數使用案例中，建議您先附加許可給某個角色，再指派給使用者。此外，也建議您使用特定動作來縮小許可範圍。

範例：可用來允許唯讀 Amazon Redshift 操作的 VPC 端點政策

下列 VPC 端點政策僅允許 AWS 帳戶*123456789012*執行指定的 Amazon Redshift 動作。

指定的動作會為 Amazon Redshift 提供等效的唯讀存取權。拒絕指定的帳戶存取在該 VPC 上的所有其他動作。此外，也會拒絕所有其他帳戶的任何存取。如需 Amazon Redshift 動作的清單，請參閱《IAM 使用者指南》中的 Amazon Redshift 的動作、資源和條件金鑰。

```
  {
    "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"
                ]
            }
        }
    ]
}
```

範例：拒絕指定叢集存取的 VPC 端點政策

下列 VPC 端點政策會允許所有帳戶和主體擁有完整存取權。同時，它會拒絕 AWS 帳戶*123456789012*對叢集 ID 為 的 Amazon Redshift 叢集上執行動作的任何存取*my-redshift-cluster*。仍然允許其他不支援叢集資源層級許可的 Amazon Redshift 動作。如需 Amazon Redshift 動作及其對應資源類型的清單，請參閱《IAM 使用者指南》中的 Amazon Redshift 的動作、資源和條件金鑰。

```
 {
    "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 端點政策範例

接下來，您可以找到 Redshift Serverless 的 VPC 端點政策範例。

範例：允許唯讀 Redshift Serverless 操作的 VPC 端點政策

下列 VPC 端點政策僅允許 AWS 帳戶*123456789012*執行指定的 Redshift Serverless 動作。

指定的動作會為 Redshift Serverless 提供同等的唯讀存取權。拒絕指定的帳戶存取在該 VPC 上的所有其他動作。此外，也會拒絕所有其他帳戶的任何存取。如需 Redshift Serverless 動作的清單，請參閱《IAM 使用者指南》中的 Redshift Serverless 的動作、資源和條件索引鍵。

```
 {
   "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"
                ]
            }
        }
    ]
}
```

範例：拒絕指定工作群組進行存取的 VPC 端點政策

下列 VPC 端點政策會允許所有帳戶和主體擁有完整存取權。同時，它會拒絕 AWS 帳戶*123456789012*對工作群組 ID 為 的 Amazon Redshift 工作群組上執行動作的任何存取*my-redshift-workgroup*。仍然允許其他不支援工作群組的資源層級許可的 Amazon Redshift 動作。如需 Redshift Serverless 動作及其對應資源類型的清單，請參閱《IAM 使用者指南》中的 Redshift Serverless 的動作、資源和條件索引鍵。

```
{
   "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 中的組態與漏洞分析

AWS 處理基本安全任務，例如訪客作業系統 (OS) 和資料庫修補、防火牆組態和災難復原 (DR)。這些程序已由認證的第三方進行檢閱。如需詳細資訊，請參閱 Amazon Redshift 的合規驗證、共同責任模式與安全、身分和合規的最佳實務。

Amazon Redshift 會自動套用升級並修補資料倉儲，這樣您就能專注於應用程式本身，而不是管理。修補和升級會在可設定的維護時間期間套用。如需詳細資訊，請參閱維護時段。

Amazon Redshift 查詢編輯器 v2 是 AWS受管應用程式。所有修補程式和更新都會由 AWS 視需要套用。

使用身分增強 IAM 角色工作階段連線至 Redshift

您可以使用 IAM Identity Center 來提供 Amazon Redshift 叢集和無伺服器工作群組的聯合存取。此方法可讓使用者使用其 Identity Center 登入資料進行身分驗證。

Amazon Redshift 提供 `GetIdentityCenterAuthToken` API 操作，以產生包含使用者身分資訊的授權字符。這些 APIs適用於佈建叢集和無伺服器工作群組。字符可讓您使用現有的 Identity Center 設定，無縫存取 Amazon Redshift 資料庫的單一登入。

先決條件

將 Identity Center 身分驗證與 Amazon Redshift 搭配使用之前，請確定您有下列項目：

- Identity Center 設定：您的帳戶必須使用使用者身分和適當的應用程式指派來設定 IAM Identity Center。如需設定說明，請參閱設定 IAM Identity Center。

  > **Important**
  >
  > 如果您想要連線至 Redshift，您必須使用 redshift：connect 範圍。

- 身分增強憑證：您的應用程式必須使用包含內嵌使用者身分資訊的身分增強憑證。如需詳細資訊，請參閱使用身分增強型 IAM 角色工作階段。
- IAM 許可：您的 IAM 角色或使用者必須具有呼叫 `GetIdentityCenterAuthToken` API 和存取指定叢集或工作群組的許可。必要許可：
  - 對於佈建的叢集：在叢集 ARN `redshift:GetIdentityCenterAuthToken`上（格式：`arn:aws:redshift:region:account:cluster:cluster-name`) ARNs
  - 對於無伺服器工作群組：在工作群組 ARNs `redshift-serverless:GetIdentityCenterAuthToken`上（格式：`arn:aws:redshift-serverless:region:account:workgroup/workgroup-name`)
- 相容驅動程式：使用支援 Identity Center 授權字符的 Amazon Redshift JDBC 或 ODBC 驅動程式：
  - JDBC 驅動程式：請參閱安裝和設定 Amazon Redshift JDBC 驅動程式 2.x 版
  - ODBC 驅動程式：請參閱安裝和設定 Amazon Redshift ODBC 驅動程式 2.0 版
- 增強型 VPC 路由的網路組態：如果開啟增強型 VPC 路由，您的 VPC 必須能夠連線到 Identity Center 服務。介面 VPC 端點是提供此連線的建議方式。在使用者登入之前設定此連線。如需 Identity Center 增強型 VPC 路由需求的詳細資訊，請參閱 搭配增強型 VPC AWS 路由使用 IAM Identity Center 身分驗證。

Identity Center 身分驗證的運作方式

Amazon Redshift 的身分中心身分驗證使用以下工作流程：

1. 您的應用程式會使用包含內嵌使用者身分資訊的身分增強憑證來呼叫 `GetIdentityCenterAuthToken` API。
2. Amazon Redshift 會驗證 Identity Center 身分，並產生範圍限定於特定叢集或工作群組的加密授權字符。請參閱範例 IAM 政策。
3. 您的應用程式使用此權杖連線到指定的 Amazon Redshift 叢集或工作群組。
4. Amazon Redshift 資料平面會驗證字符，並根據 Identity Center 應用程式內的 Identity Center 使用者許可授予存取權。

> **Important**
>
> 此 API 需要身分增強憑證。如需詳細資訊，請參閱使用身分增強型 IAM 角色工作階段。
> 如果您在沒有身分增強憑證的情況下呼叫 API，您會收到`UnsupportedOperationFault`錯誤。

GetIdentityCenterAuthToken API 操作

Amazon Redshift 提供兩個不同的 `GetIdentityCenterAuthToken` API 操作：一個用於佈建的叢集，另一個用於無伺服器工作群組。這兩個操作的名稱都相同，但接受不同的參數，取決於目標資源類型。

已佈建叢集的 GetIdentityCenterAuthToken

對於佈建的 Amazon Redshift 叢集，請使用 Amazon Redshift 服務中的 `GetIdentityCenterAuthToken` API 來產生授權字符。

請求語法

```
{
   "ClusterIds": [ "string" ]
}
```

請求參數

ClusterIds

有權存取字符的 Amazon Redshift 叢集識別符清單。字符只能用來驗證此清單中指定的叢集。

類型：字串陣列

長度限制：最少 1 個項目。最多 20 個項目。

必要：是

CLI 範例

範例：取得單一叢集的授權字符

```
aws redshift get-identity-center-auth-token \
    --cluster-ids my-redshift-cluster
```

範例：取得多個叢集的授權字符

```
aws redshift get-identity-center-auth-token \
    --cluster-ids my-cluster-1 my-cluster-2
```

適用於無伺服器工作群組的 GetIdentityCenterAuthToken

對於 Amazon Redshift Serverless 工作群組，請使用 Amazon Redshift Serverless 服務中的 `GetIdentityCenterAuthToken` API 來產生授權字符。

請求語法

```
{
   "WorkgroupNames": [ "string" ]
}
```

請求參數

WorkgroupNames

有權存取字符的 Amazon Redshift Serverless 工作群組名稱清單。字符只能用來驗證此清單中指定的工作群組。

類型：字串陣列

長度限制：最少 1 個項目。最多 20 個項目。

必要：是

CLI 範例

範例：取得單一工作群組的授權字符

```
aws redshift-serverless get-identity-center-auth-token \
    --workgroup-names my-workgroup
```

範例：取得多個工作群組的授權字符

```
aws redshift-serverless get-identity-center-auth-token \
    --workgroup-names workgroup-1 workgroup-2
```

回應語法

兩個 APIs都會傳回相同的回應結構：

```
{
   "AuthorizedToken": "string",
   "ExpirationTime": "timestamp"
}
```

回應參數

AuthorizedToken

加密的授權字符，其中包含使用者身分資訊和授權叢集或工作群組的清單。此字符應視為敏感資料。

類型：字串

ExpirationTime

字符過期的日期和時間，以 UTC 為單位。字符自產生時起 1 小時內有效。

類型：Timestamp

回應範例

```
{
    "AuthorizedToken":
 "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ1c2VyQGV4YW1wbGUuY29tIiwiaWF0IjoxNjQwOTk1MjAw
    "ExpirationTime": "2024-01-01T12:00:00Z"
}
```

驅動程式整合

Amazon Redshift 驅動程式透過直接字符用量支援 Identity Center 身分驗證：

直接字符用量

呼叫 `GetIdentityCenterAuthToken` API 以取得字符之後，請使用 `IdpTokenAuthPlugin`搭配`SUBJECT_TOKEN`字符類型。

連線組態：

```
plugin_name = com.amazon.redshift.plugin.IdpTokenAuthPlugin
token_type = SUBJECT_TOKEN
token = {encrypted_token_from_api_response}
```

如需 Identity Center 身分驗證外掛程式和驅動程式組態的詳細資訊，請參閱連線至 Amazon Redshift 叢集。

Java 程式碼範例

使用 Identity Center 身分驗證連線的範例 Java 程式碼：

```
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 政策要求

若要搭配 Amazon Redshift 使用 Identity Center 身分驗證，除了連線至 Amazon Redshift 叢集和工作群組所需的標準許可之外，還需要特定的 IAM 許可。

API 許可

對於佈建的叢集，增強型 IAM 角色工作階段必須具有：

- redshift:GetIdentityCenterAuthToken 叢集 ARNs上的 （格式：arn:aws:redshift:region:account:cluster:cluster-name)

對於無伺服器工作群組，增強型 IAM 角色工作階段必須具有：

- redshift-serverless:GetIdentityCenterAuthToken 工作群組 ARNs上的 （格式：arn:aws:redshift-serverless:region:account:workgroup/workgroup-name)

範例 IAM 政策

已佈建叢集的範例政策：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "redshift:GetIdentityCenterAuthToken"
            ],
            "Resource": [
                "arn:aws:redshift:us-east-1:123456789012:cluster:my-cluster"
            ]
        }
    ]
}
```

無伺服器工作群組的範例政策：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "redshift-serverless:GetIdentityCenterAuthToken"
            ],
            "Resource": [
                "arn:aws:redshift-serverless:us-east-1:123456789012:workgroup/my-
workgroup"
            ]
```

```
        }
    ]
}
```

多個資源的政策範例：

```
{
    "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/*"
            ]
        }
    ]
}
```

區域可用性

Identity Center 身分驗證可在下列 AWS 區域使用：

- 商業區域：所有支援的 Amazon Redshift 區域
- AWS GovCloud：適用於 us-gov-east-1 和 us-gov-west-1
- 中國區域：適用於 cn-north-1 和 cn-northwest-1

> **Note**
>
> 功能可用性在初始推出期間可能會有所不同。

故障診斷 Identity Center 身分驗證

使用下列各節來協助對 Identity Center 身分驗證的問題進行疑難排解。

主題

- API 錯誤
- 連線錯誤

API 錯誤

`GetIdentityCenterAuthToken` API 可能會傳回下列錯誤：

- UnsupportedOperationFault – 在沒有身分增強憑證的情況下提出請求，或目前組態不支援 Identity Center 身分傳播。
- ClusterNotFoundFault (redshift：GetIdentityCenterAuthToken) – 一或多個指定的叢集識別符不存在，或者您沒有存取它們的許可。
- WorkgroupNotFoundFault (redshift-serverless：GetIdentityCenterAuthToken) – 一或多個指定的工作群組名稱不存在，或者您沒有存取這些名稱的許可。
- InvalidParameterValueException – 請求參數無效，例如叢集或工作群組的空清單，或清單中超過 20 個項目。

連線錯誤

使用 Identity Center 權杖連線至 Amazon Redshift 時，您可能會遇到下列錯誤：

- 權杖憑證已過期或無效 – 使用 `GetIdentityCenterAuthToken` API 產生新的權杖並更新您的連線組態。
- 無效字符 – 驗證字符格式，並確保為正確的叢集或工作群組產生字符格式。
- TYPE_INVALID_TOKEN – 請確定您在連線參數`token_type = SUBJECT_TOKEN`中使用 。
- 不支援身分提供者類型 "awsidc" – 確認您的 Amazon Redshift 叢集或工作群組支援身分中心身分驗證並更新至相容的版本。

- AWS IdC 身分提供者不存在 – 透過 AWS 主控台設定 Amazon Redshift 資源的 Identity Center 整合。
- 無效的範圍。使用者的登入資料未獲授權連線至 Amazon Redshift – 確認使用者在 Identity Center 中具有適當的許可，並且已針對正確的資源產生權杖。
- 使用增強型 VPC 路由的連線失敗 – 開啟增強型 VPC 路由時，Amazon Redshift 會透過您的 VPC 到達 Identity Center 服務。確認您的 VPC 具有適用於這些服務的介面 VPC 端點。確認每個端點的私有 DNS 名稱都已開啟。確認端點安全群組允許 TCP 連接埠 443 上的傳入流量。如需 Identity Center 增強型 VPC 路由需求的詳細資訊，請參閱 搭配增強型 VPC AWS 路由使用 IAM Identity Center 身分驗證。

聯網任務

您可以執行聯網任務，例如自訂 Redshift 資料庫的連線。基於安全考量或其他目的，您可能會希望利用這種方式來控制流量。您也可以執行 DNS 相關任務，例如設定 Redshift 資源的自訂網域名稱。如果您有 Amazon Redshift 佈建叢集或使用 Amazon Redshift Serverless 工作群組，則可以使用這些組態任務。

主題

- 自訂網域名稱來進行用戶端連線
- Redshift 受管 VPC 端點
- VPC 中的 Redshift 資源
- 使用 Redshift 增強型 VPC 路由控制網路流量

自訂網域名稱來進行用戶端連線

您可以為 Amazon Redshift 叢集和 Amazon Redshift Serverless 工作群組建立自訂網域名稱 (也稱為自訂 URL)。其為易讀的 DNS 記錄，可將 SQL 用戶端連線路由到您的端點。您可以隨時為現有叢集或工作群組設定此項目。其提供了數種優勢：

- 自訂網域名稱比預設 URL 更簡單的字串，通常包括叢集名稱或工作群組名稱和區域。其更容易召回和使用。
- 例如，在容錯移轉案例中，您可以快速將流量路由至新叢集或工作群組。這會讓用戶端不必在重新連線時變更組態。連線可以集中地重新路由，而且中斷情況最少。
- 您可以避免在連線 URL 中共用私人資訊，例如伺服器名稱。您可以將其隱藏在自訂 URL 中。

當您使用 CNAME 設定自訂網域名稱時，Amazon Redshift 不會產生任何額外費用。如果您建立新的網域名稱，DNS 供應商可能會向您收取相關費用，但金額通常很小。

註冊網域名稱

設定自訂網域名稱包含幾項任務：其中包括向 DNS 提供者註冊網域名稱以及建立憑證。執行這些工作之後，您可以在 Amazon Redshift 主控台或 Amazon Redshift Serverless 主控台中設定自訂網域名稱，或使用 AWS CLI 命令進行設定。

您必須有已註冊的網際網路網域名稱，才能在 Amazon Redshift 中設定自訂網域名稱。您可以使用 Route 53 或第三方網域註冊提供商來註冊網際網路網域。您會在 Amazon Redshift 主控台之外完成這些任務。想要完成剩餘程序以便建立自訂網域，先決條件是要有已註冊的網域。。

> **Note**
>
> 若您正在使用佈建的叢集，在執行相關步驟以設定自訂網域名稱之前，您必須先啟用重新定位。如需詳細資訊，請參閱重新定位叢集。Amazon Redshift Serverless 不需要此步驟。

自訂網域名稱通常包括根網域和子網域，例如 `mycluster.example.com`。若要進行設定，請執行下列步驟：

為自訂網域名稱建立 DNS CNAME 項目

1. 註冊根網域，例如 `example.com`。您可以選擇使用現有網域。自訂名稱可能會受限於特定字元限制或其他命名驗證。如需使用 Route 53 註冊網域的相關資訊，請參閱註冊新網域。
2. 針對您的叢集或工作群組，新增會將自訂網域指向 Redshift 端點的 DNS CNAME 記錄。您可以在叢集或工作群組的屬性、Redshift 主控台或 Amazon Redshift Serverless 主控台尋找端點。在一般資訊底下的叢集或工作群組屬性中複製可用的 JDBC URL。URL 顯示如下：

   - 對於 Amazon Redshift 叢集：`redshift-cluster-sample.abc123456.us-east-1.redshift.amazonaws.com`
   - 對於 Amazon Redshift Serverless 工作群組：`endpoint-name.012345678901.us-east-1-dev.redshift-serverless-dev.amazonaws.com`

   如果 URL 具有 JDBC 字首，請將其移除。

   > **Note**
   >
   > DNS 記錄不一定都可供使用，因為每個名稱都必須是唯一的，並且可在您的組織中使用。

限制

為自訂網域建立 CNAME 記錄有幾項限制：

- 不支援為相同佈建的叢集或 Amazon Redshift Serverless 工作群組建立多個自訂網域名稱。您僅能為一個 CNAME 記錄建立關聯性。
- 不支援使 CNAME 記錄與多個叢集或工作群組產生關聯。每個 Redshift 資源的 CNAME 必須是唯一的。

在註冊網域並建立 CNAME 記錄後，您就要選取新的憑證或現有憑證。您可以使用下列方式執行此步驟 AWS Certificate Manager：

我們建議您建立符合受管續約資格的 DNS 驗證憑證，此憑證適用於 AWS Certificate Manager。受管續約的意思是 ACM 會自動續約您的憑證，或在即將到期時傳送電子郵件通知給您。如需詳細資訊，請參閱 ACM 憑證的受管續約。

請求網域名稱的憑證

Amazon Redshift 或 Amazon Redshift Serverless 需要經過驗證的 Secure Sockets Layer (SSL) 憑證，才能讓自訂端點確保通訊安全並驗證網域名稱的擁有權。您可以使用 AWS Certificate Manager 帳戶搭配 AWS KMS key 進行安全憑證管理。安全驗證包括完整的主機名稱驗證 (sslmode=verify-full)。

只有在選擇 DNS 驗證 (而非電子郵件驗證) 時，Amazon Redshift 才會管理憑證的續約。如果您使用電子郵件驗證，您還是可以使用憑證，但必須在憑證到期前自行續約。建議您為憑證選擇 DNS 驗證。您可以在 AWS Certificate Manager中監控所匯入憑證的到期日期。

向 ACM 請求網域名稱的憑證

1. 登入 AWS 管理主控台 並開啟位於 https：//https://console.aws.amazon.com/acm/ 的 ACM 主控台。
2. 選擇 Request a certificate (請求憑證)。
3. 在網域名稱欄位中輸入您的自訂網域名稱。

   > **Note**
   >
   > 除了憑證網域外，您還可以指定許多字首，以便將單一憑證用於多個自訂網域記錄。若要說明，您可以搭配相同憑證來使用其他記錄，例如 `one.example.com`、`two.example.com` 或萬用字元 DNS 記錄如 `*.example.com`。

4. 選擇 Review and request (檢閱和請求)。
5. 選擇 Confirm and request (確認和請求)。

6. 針對有效的請求，在 ACM 發出憑證之前，網際網路網域的註冊擁有者必須先同意請求。完成這些步驟後，請確定狀態在 ACM 主控台中顯示為已發行。

設定自訂網域

您可以使用 Amazon Redshift 或Amazon Redshift Serverless 主控台來建立自訂網域 URL。如果您尚未設定，在一般資訊底下，自訂網域名稱屬性會顯示為破折號 (–)。建立 CNAME 記錄和憑證之後，為叢集或工作群組的自訂網域名稱建立關聯。

若要建立自訂網域關聯，您必須有下列 IAM 許可：

- `redshift:CreateCustomDomainAssociation` — 您可以透過新增特定叢集的 ARN 來限制特定叢集的許可。
- `redshiftServerless:CreateCustomDomainAssociation` — 您可以透過新增 ARN 來限制特定工作群組的許可。
- `acm:DescribeCertificate`

我們建議的最佳實務是，將許可政策附加到 IAM 角色，然後根據需要將其指派給使用者和群組。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。

您可以透過執行下列步驟來指派自訂網域名稱。

1. 在 Amazon Redshift 主控台選擇叢集，或在 Amazon Redshift Serverless 主控台選擇工作群組，然後在動作功能表下選擇建立自訂網域名稱。隨即會出現對話方塊。
2. 輸入自訂網域名稱。
3. 從 選取 ACM 憑證 AWS Certificate Manager 的 ARN。 確認變更。根據您建立憑證所採取步驟中的指引，我們建議您選擇符合受管續約資格的 DNS 驗證憑證 AWS Certificate Manager。
4. 在叢集屬性中確認自訂網域名稱和自訂網域憑證 ARN 已填入您的項目。其中也會列出自訂網域憑證到期日。

設定好自訂網域後，使用 `sslmode=verify-full` 只會對新的自訂網域起作用。其不會對預設端點起作用。但是您仍然可以透過使用其他 ssl 模式 (例如 `sslmode=verify-ca`) 來連線到預設端點。

> **Note**
>
> 提醒您，叢集重新定位不是設定其他 Redshift 網路功能的先決條件。您不需要開啟此功能即可啟用下列項目：

- 從跨帳戶或跨區域 VPC 連線至 Redshift – 您可以從一個 AWS 虛擬私有雲端 (VPC) 連線至包含 Redshift 資料庫的另一個。這樣可以更輕鬆地管理，例如：從不同的帳戶或 VPC 進行用戶端存取，而不必提供本機 VPC 存取權限來連線至資料庫的身分識別。如需詳細資訊，請參閱從其他帳戶或區域中的 Redshift VPC 端點連線至 Amazon Redshift Serverless。
- 設定自訂網域名稱 — 您可以建立自訂網域名稱 (如本主題所述)，讓端點名稱更相關且簡單。

連線至您的 Amazon Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組

若要使用自訂網域名稱進行連線，必須有佈建叢集的下列 IAM 許可：`redshift:DescribeCustomDomainAssociations`。對於 Amazon Redshift Serverless 伺服器，您不需新增許可。

我們建議的最佳實務是，將許可政策附加到 IAM 角色，然後根據需要將其指派給使用者和群組。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。

在主控台完成建立 CNAME 並將其指派給叢集的步驟後，您可以在 SQL 用戶端的連線屬性中提供自訂 URL。請注意，建立 CNAME 記錄之後，系統會立即因為 DNS 傳播而發生延遲。

1. 開啟 SQL 用戶端。例如，您可以使用 SQL/Workbench J。開啟連線的屬性，並新增連線字串的自訂網域名稱。例如 `jdbc:redshift://mycluster.example.com:5439/dev?sslmode=verify-full`。在此範例中，`dev` 會指定預設資料庫。
2. 新增資料庫使用者的使用者名稱和密碼。
3. 測試連線。根據向資料庫使用者授予的許可或向所指派的 Amazon Redshift 資料庫角色授予的許可，您查詢資料庫資源 (例如特定資料表) 的能力會有所不同。

   請注意，如果在 VPC 中，您可能必須將叢集或工作群組設定為可公開存取，才能連線到叢集或工作群組。您可以在網路屬性中變更此設定。

> **Note**
>
> JDBC、ODBC 和 Python 驅動器支援透過自訂網域名稱的連線。

重新命名已指派自訂網域的叢集

> **Note**
>
> 這一系列步驟不適用於 Amazon Redshift Serverless 工作群組。您無法變更工作群組名稱。

若要重新命名具有自訂網域名稱的叢集，就必須有 `acm:DescribeCertificate` IAM 許可。

1. 前往 Amazon Redshift 主控台，然後選擇要變更其名稱的叢集。選擇編輯以編輯叢集屬性。
2. 叢集叢集識別碼。您也可以變更叢集的其他屬性。接著選擇 Save changes (儲存變更)。
3. 重新命名叢集之後，您必須更新 DNS 記錄，將自訂網域的 CNAME 項目變更為指向更新後的 Amazon Redshift 端點。

說明自訂網域關聯

使用本節中的命令可取得與特定佈建叢集或 Amazon Redshift Serverless 工作群組相關聯的自訂網域名稱清單。

您需要具備下列許可：

- 針對已佈建的叢集：`redshift:DescribeCustomDomainAssociations`
- 對於 Amazon Redshift Serverless 工作群組：`redshiftServerless:ListCnameAssociations`

我們建議的最佳實務是，將許可政策附加到 IAM 角色，然後根據需要將其指派給使用者和群組。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。

以下是列出指定 Amazon Redshift 叢集之自訂網域名稱的範例命令：

```
aws redshift describe-custom-domain-associations --custom-domain-name customdomainname
```

您可以在啟用自訂網域名稱時執行此命令，以判斷與叢集相關聯的自訂網域名稱。如需用來描述自訂網域關聯之 CLI 命令的詳細資訊，請參閱 describe-custom-domain-associations。

同樣的，以下顯示範例命令以列出指定 Amazon Redshift Serverless 工作群組的自訂網域名稱。有幾種不同的方法可以做到這一點。您只能提供自訂網域名稱：

```
aws redshift-serverless list-custom-domain-associations --custom-domain-
name customdomainname
```

您也可以透過僅提供憑證 ARN 來取得關聯：

```
aws redshift-serverless list-custom-domain-associations --custom-domain-certificate-
arn certificatearn
```

當啟用自訂網域名稱以決定與工作群組相關聯的自訂網域名稱時，您可以執行這些命令。您也可以執行命令來取得自訂網域關聯的內容。若要這麼做，您必須提供自訂網域名稱和工作群組名稱做為參數。系統會傳回憑證 ARN、工作群組名稱和自訂網域的憑證到期時間：

```
aws redshift-serverless get-custom-domain-association --workgroup-name workgroupname --
custom-domain-name customdomainname
```

如需適用於 Amazon Redshift Serverless 的 CLI 參考命令的詳細資訊，請參閱 redshift-serverless。

將自訂網域與不同的憑證建立關聯

若要變更自訂網域名稱的憑證關聯，需要下列 IAM 許可：

- `redshift:ModifyCustomDomainAssociation`
- `acm:DescribeCertificate`

我們建議的最佳實務是，將許可政策附加到 IAM 角色，然後根據需要將其指派給使用者和群組。如需詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。

使用下列命令以利用不同憑證與自訂網域建立關聯。`--custom-domain-name` 和 `custom-domain-certificate-arn` 引數是強制性的。新憑證的 ARN 必須與現有 ARN 不同。

```
aws redshift modify-custom-domain-association --cluster-id redshiftcluster --custom-
domain-name customdomainname --custom-domain-certificate-arn certificatearn
```

以下範例顯示如何為自訂網域與 Amazon Redshift Serverless 工作群組的不同憑證建立關聯。

```
aws redshift-serverless modify-custom-domain-association --workgroup-
name redshiftworkgroup --custom-domain-name customdomainname --custom-domain-
certificate-arn certificatearn
```

連線到叢集之前，最多會延遲 30 秒。部分延遲發生在 Amazon Redshift 叢集更新其屬性時，並且在 DNS 更新時也會有一些額外的延遲。如需 API 和每個屬性設定的相關資訊，請參閱 ModifyCustomDomainAssociation。

刪除自訂網域

若要刪除自訂網域名稱，使用者必須具有下列動作的許可：

- 針對已佈建的叢集：`redshift:DeleteCustomDomainAssociation`
- 對於 Amazon Redshift Serverless 工作群組：`redshiftServerless:DeleteCustomDomainAssociation`

在主控台上

您可以透過選取動作按鈕並選擇刪除自訂網域名稱來刪除自訂網域名稱。執行此操作之後，您仍然可以卜透過更新工具以使用主控台列出的端點來連接伺服器。

使用 CLI 指令

以下範例顯示如何刪除自訂網域名稱。刪除作業需要您提供叢集的現有自訂網域名稱。

```
aws redshift delete-custom-domain-association --cluster-id redshiftcluster --custom-
domain-name customdomainname
```

以下範例顯示如何刪除 Amazon Redshift Serverless 工作群組的自訂網域名稱。自訂網域名稱是必要的參數。

```
aws redshift-serverless delete-custom-domain-association --workgroup-name workgroupname
 --custom-domain-name customdomainname
```

如需詳細資訊，請參閱 DeleteCustomDomainAssociation。

Redshift 受管 VPC 端點

根據預設，Amazon Redshift 叢集或 Amazon Redshift Serverless 工作群組會佈建在虛擬私有雲端 (VPC) 中。當您允許公開存取，或設定網際網路閘道、NAT 裝置或 AWS Direct Connect 連線來將流量路由時，便可從其他 VPC 或子網路存取 VPC。您也可以設定 Redshift 受管 VPC 端點（由 提供支援）來存取叢集或工作群組 AWS PrivateLink。

您可以將 Redshift 受管 VPC 端點設定為包含叢集或工作群組的 VPC 與用戶端工具執行所在的 VPC 之間的私有連線。如果叢集或工作群組位於其他帳戶中，則帳戶擁有者 (授予者) 必須對連線帳戶 (被授予者) 授予存取權。透過此方式，您不需使用公有 IP 位址或透過網際網路來路由流量，就可以存取資料倉儲。

以下是允許使用 Redshift 受管 VPC 端點進行存取的常見原因：

- AWS 帳戶 A 想要允許 AWS 帳戶 B 中的 VPC 存取叢集或工作群組。
- AWS 帳戶 A 想要允許帳戶 AWS A 中的 VPC 存取叢集或工作群組。
- AWS 帳戶 A 想要允許 AWS 帳戶 A 內 VPC 中的不同子網路存取叢集或工作群組。

設定 Redshift 受管 VPC 端點存取其他帳戶中的叢集和工作群組的工作流程如下：

1. 擁有者帳戶將存取授權授予其他帳戶，並指定被授予者的 AWS 帳戶 ID 和 VPC 識別碼 (或所有 VPC)。
2. 被授予者帳戶收到通知，指出其已擁有建立 Redshift 管理的 VPC 端點的許可。
3. 被授予者帳戶建立 Redshift 管理的 VPC 端點。
4. 被授予者帳戶會使用 Redshift 受管 VPC 端點來存取擁有者帳戶的叢集或工作群組。

您可以使用 Amazon Redshift 主控台 AWS CLI、 或 Amazon Redshift API 來執行此操作。

使用 Redshift 管理的 VPC 端點時的考量

> **Note**
>
> 若要建立或修改 Redshift 受管 VPC 端點，除了 AWS 受管政策 `ec2:ModifyVpcEndpoint` 中指定的其他許可之外，還需要 許可`ec2:CreateVpcEndpoint`或 IAM 政策中的 `AmazonRedshiftFullAccess`。

在使用 Redshift 管理的 VPC 端點時，請記住以下事項：

- 如果您使用的是佈建叢集，它必須具有 RG 或 RA3 節點類型。Amazon Redshift Serverless 工作群組也可用於設定 VPC 端點。
- 對於佈建叢集，請確定已啟用叢集重新放置或多可用區域的叢集。如需開啟叢集重新定位要求的相關資訊，請參閱重新定位叢集。如需啟用多可用區域的相關資訊，請參閱 在建立新叢集時設定多可用區。

- 確定要透過此安全群組存取的叢集或工作群組可在有效的連接埠範圍 5431 到 5455 和 8191 到 8215 內使用。預設值為 5439。
- 您可以修改與現有 Redshift 管理的 VPC 端點相關聯的 VPC 安全群組。若要修改其他設定，請刪除目前的 Redshift 管理的 VPC 端點並建立新端點。
- 您可建立的 Redshift 管理的 VPC 端點數目受限於 VPC 端點的配額。
- Redshift 管理的 VPC 端點無法從網際網路存取。在路由表和安全群組許可的情況下，Redshift 受管 VPC 端點只能在佈建端點的 VPC 內存取，或是從與佈建端點的 VPC 對等的任何 VPC 內存取。
- 您無法使用 Amazon VPC 主控台來管理 Redshift 管理的 VPC 端點。
- 當您為佈建叢集建立 Redshift 受管 VPC 端點時，您選擇的 VPC 必須具有子網路群組。若要建立子網路群組，請參閱 建立叢集子網路群組。
- 如果可用區域關閉，Amazon Redshift 不會在另一個可用區域中建立新的彈性網路介面。在這種情況下，您可能需要建立新的端點。

如需有關配額和命名限制的資訊，請參閱 Amazon Redshift 中的配額和限制。

如需定價的詳細資訊，請參閱 AWS PrivateLink 定價。

授予 VPC 的存取權

若您要存取叢集或工作群組的 VPC 位於其他 AWS 帳戶，請務必從擁有者 (授予者) 的帳戶對其授權。

允許另一個 AWS 帳戶中的 VPC 存取您的叢集或工作群組

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽選單上，選擇叢集。針對 Amazon Redshift Serverless，選擇 Serverless 儀表板。
3. 對於您要允許存取的叢集，選擇該叢集名稱來檢視詳細資訊。選擇叢集的屬性索引標籤。

   已授予的帳戶區段會顯示可存取您叢集的帳戶和對應的 VPC。對於 Amazon Redshift Serverless 工作群組，請選擇工作群組。已授予的帳戶位於資料存取索引標籤下。
4. 選擇授予存取權以顯示表單，供您輸入要新增帳戶的被授予者資訊。
5. 對於 AWS 帳戶 ID，輸入您要授予存取權之帳戶的 ID。您可以授予存取權給指定帳戶中的特定 VPC 或所有 VPC。
6. 選擇授予存取權以授予存取權。

建立 Redshift 管理的 VPC 端點

如果您擁有叢集或工作群組，或是您已獲授予該叢集的存取權可進行管理，則您可以為其建立 Redshift 受管 VPC 端點。

建立 Redshift 管理的 VPC 端點

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇組態。

   組態頁面會顯示已建立的 Redshift 管理的 VPC 端點。若要檢視端點的詳細資訊，請選擇其名稱。對於 Amazon Redshift Serverless，當您選擇工作群組時，VPC 端點位於資料存取索引標籤下。
3. 選擇建立端點以顯示表單，供您輸入要新增之端點的相關資訊。
4. 輸入端點名稱、12 位數 AWS 帳戶 ID、端點所在的虛擬私有雲端 (VPC)、子網路和 VPC 安全群組的值。

   子網路中的子網路會定義 Amazon Redshift 部署端點所在的子網路和 IP 位址。Amazon Redshift 會選擇具有 IP 地址可供與該端點相關聯之網路介面使用的子網路。

   VPC 安全群組中的安全群組規則會定義您為端點授權之傳入流量的連接埠、通訊協定和來源。您允許透過安全群組或工作負載執行所在的 CIDR 範圍來存取選取的連接埠。
5. 選擇建立端點以建立端點。

建立端點之後，您可以透過 Redshift 受管 VPC 端點的組態設定中，端點 URL 所顯示的 URL 來存取叢集或工作群組。

VPC 中的 Redshift 資源

您可以在以 Amazon VPC 服務為基礎的 EC2-VPC 平台上，於 VPC 中啟動 Amazon Redshift 叢集或 Amazon Redshift Serverless 工作群組。如需詳細資訊，請參閱使用 EC2 建立叢集。

> **Note**
>
> 不支援啟動叢集和無伺服器工作群組進入專用的租用 VPC。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的專用執行個體。

在 VPC 中佈建資源時，您必須執行下列操作：

- 提供 VPC 資訊。

  當您在 VPC 中建立佈建叢集時，您必須建立叢集子網路群組以提供 VPC 資訊。此資訊包括 VPC ID 和 VPC 中的子網路清單。當您啟動叢集時，需要提供子網路群組，讓 Redshift 可以在 VPC 的其中一個子網路中佈建該群組。使用 Amazon Redshift Serverless 時的程序類似。您會直接將子網路指派至 Serverless 工作群組。但是在無伺服器的情況下，您不會建立子網路群組。如需在 Amazon Redshift 中建立子網路群組的相關資訊，請參閱 Redshift 資源的子網路。如需設定 VPC 的相關資訊，請參閱《Amazon VPC 入門指南》中的 Amazon VPC 入門。

- 您也可以選擇設定可存取性選項。

  Amazon Redshift 中的佈建叢集和無伺服器工作群組預設為私有。如果您將佈建叢集或無伺服器工作群組設定為可公開存取，則 Amazon Redshift 會針對外部 IP 位址使用彈性 IP 位址。彈性 IP 地址是靜態 IP 地址。其可讓您變更基礎組態，而不會影響用戶端用於連線的 IP 位址。這個方法對於諸如故障後復原等情況很有幫助。是否建立彈性 IP 地址取決於您的可用區域重新定位設定。您有兩個選項：

  1. 如果您已開啟可用區域重新定位，且想要啟用公開存取，則不要指定彈性 IP 地址。系統會指派由 Amazon Redshift 管理的彈性 IP 地址。其會與您的 AWS 帳戶相關聯。
  2. 如果您已關閉可用區域重新定位，且想要啟用公開存取，則可以在啟動 Amazon Redshift 叢集或工作群組之前，選擇在 Amazon EC2 中為 VPC 建立彈性 IP 位址。如果您不建立 IP 地址，Amazon Redshift 會提供已設定的彈性 IP 地址以用於 VPC。此彈性 IP 地址由 Amazon Redshift 管理，不會與 AWS 您的帳戶建立關聯。

  如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的彈性 IP 位址。

  在某些情況下，您的 VPC 中可能有可公開存取的叢集，而您想要從 VPC 內使用私有 IP 位址連線至該叢集。若是如此，請將下列 VPC 參數設定為 `true`：

  - `DNS resolution`
  - `DNS hostnames`

  請注意，使用 Amazon Redshift Serverless 時，您無法以此方式連線。

  假設您的 VPC 中有可公開存取的佈建叢集，但未在 VPC 中將這些參數設定為 `true`。在這些情況下，從 VPC 內部建立的連線會解析為資源的彈性 IP 位址，而不是私有 IP 位址。建議您在從 VPC 內部連線時，將這些參數設為 `true`，並讓可公開存取的叢集使用私有 IP 地址。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 將 DNS 用於您的 VPC。

> **Note**
>
> 如果您的 VPC 中有可公開存取的現有叢集，來自 VPC 內部的連線會繼續使用彈性 IP 位址來連線至叢集，直到您調整其大小為止 (如果是佈建叢集)。即使已設定上述參數，也會發生此情況。任何新建立的叢集都會遵守新的行為，也就是在從相同 VPC 內部連線至可公開存取的叢集時，使用私有 IP 位址。

彈性 IP 位址是用來存取 VPC 外部資源的外部 IP 位址。對於佈建叢集，此與 Amazon Redshift 主控台節點 IP 位址底下顯示的公有 IP 位址和私有 IP 位址無關。不論叢集是否可公開存取，公有和私有叢集節點 IP 位址都會顯示。這些位址只能在特定情況下，用來設定遠端主機上的傳入規則。當您從 Amazon EC2 執行個體或其他使用安全殼層 (SSH) 連線的遠端主機載入資料時，就會發生這些情況。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的步驟 1：擷取叢集公有金鑰和叢集節點 IP 地址。

> **Note**
>
> 節點 IP 位址不適用於 Redshift Serverless 工作群組。

您可以在建立叢集或從快照還原叢集時，選擇將佈建叢集和彈性 IP 位址建立關聯。在某些情況下，您可能想要將叢集與彈性 IP 地址相關聯，或是變更與叢集相關聯的彈性 IP 地址。若要在建立叢集後連接彈性 IP 地址，請先更新叢集，使其無法公開存取，然後將其同時設為可公開存取，並在同一個操作中新增靜態 IP 地址。

如需如何將佈建叢集或 Amazon Redshift Serverless 工作群組設為公開存取，並指派彈性 IP 位址的詳細資訊，請參閱具有預設或自訂安全群組組態的公開存取性。

- 與 VPC 安全群組建立關聯

  您使用 VPC 安全群組授予傳入存取。如需詳細資訊，請參閱設定 Amazon Redshift 叢集的安全群組通訊設定，其中會提供在用戶端與佈建叢集或 Amazon Redshift Serverless 工作群組之間設定傳入和傳出規則的指引。另一個可協助您了解安全群組的資源是《Amazon VPC 使用者指南》中的 VPC 中的安全

在 VPC 中還原佈建叢集或 Serverless 工作群組的快照

VPC 中叢集或 Serverless 工作群組的快照只能在 VPC 中還原，不能在 VPC 外還原。您可以在相同 VPC 中還原，或在帳戶的其他 VPC 中還原。如需快照的相關資訊，請參閱 Amazon Redshift 快照和備份。

在 VPC 中建立 Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組

以下是在虛擬私有雲端 (VPC) 中部署叢集或工作群組的一般步驟。

在 VPC 中建立叢集或 Serverless 工作群組

1. 設定 VPC - 您可以在帳戶的預設 VPC 中 (如果帳戶有預設 VPC)，或是在您建立的 VPC 中建立 Redshift 資源。如需詳細資訊，請參閱使用 EC2 建立叢集。若要建立 VPC，請參閱《Amazon VPC 使用者指南》中的 VPC 的子網路。請記下 VPC 識別碼、子網路和子網路的可用區域。當您啟動叢集或工作群組時，會需要此資訊。

   > **Note**
   >
   > 您至少必須在 VPC 中定義一個子網路，才能在下一個步驟將其新增至子網路群組。如需在您的 VPC 中新增子網路的相關資訊，請參閱《Amazon VPC 使用者指南》中的在您的 VPC 中新增子網路。

2. 建立 Amazon Redshift 叢集子網路群組，以指定您的 Amazon Redshift 叢集可用於 VPC 的子網路。對於 Redshift Serverless，您不會建立子網路群組，而是在建立工作群組時將子網路集合指派至該工作群組。您可以在建立工作群組時，於 Serverless 儀表板中執行此操作。

   您可以使用 Amazon Redshift 主控台或以程式設計的方式建立子網路群組。如需詳細資訊，請參閱Redshift 資源的子網路。

3. 在您要與叢集或工作群組建立關聯的 VPC 安全群組中，授權傳入連線的存取權。您可以允許 VPC 外部的用戶端 (在公有網際網路上) 連線至叢集。若要執行此操作，請將叢集與授予傳入存取權的 VPC 安全群組建立關聯。如需詳細資訊，請參閱設 Amazon Redshift 叢集或 Amazon Redshift Serverless 工作群組的安全群組通訊設定。

4. 依照下列步驟在 Redshift 佈建主控台或工作群組中，或在 Amazon Redshift Serverless 主控台中建立叢集。在網路與安全中，指定您設定的虛擬私有雲端 (VPC)、叢集子網路群組和 VPC 安全群組。

   如需顯示建立佈建資料倉儲叢集之更詳細步驟的逐步解說，請參閱《Amazon Redshift 入門指南》中的開始使用 Amazon Redshift 佈建資料倉儲。如需建立 Amazon Redshift Serverless 的詳細資訊，請參閱《Amazon Redshift 入門指南》中的開始使用 Amazon Redshift 資料倉儲。

您可以依照《入門》中的步驟上傳範例資料並執行範例查詢，藉此測試叢集或工作群組。如需詳細資訊，請參閱《Amazon Redshift 入門指南》中的開始使用 Amazon Redshift Serverless 資料倉儲。

VPC security groups (VPC 安全群組)

當您佈建 Amazon Redshift 叢集或 Amazon Redshift Serverless 工作群組時，根據預設存取權會受到限制，不讓任何人存取。若要對其他使用者授予傳入存取權，您可以將其與安全群組建立關聯。如果您是在 EC2-VPC 平台上，則可以使用現有的 Amazon VPC 安全群組，或定義新的安全群組。然後將其與叢集或工作群組建立關聯，如下所述。如果您是在 EC2-Classic 平台上，則會定義安全群組，並使其與叢集或工作群組建立關聯。如需在 EC2-Classic 平台上使用安全群組的詳細資訊，請參閱 Amazon Redshift 安全群組。

VPC 安全群組包含一組規則，用來控制對 VPC 上執行個體 (例如您的叢集) 的存取。個別規則設定存取是依據 IP 地址的範圍或依據其他 VPC 安全群組。當您將 VPC 安全群組與叢集或工作群組建立關聯時，VPC 安全群組中定義的規則即會控制存取權。

您在 EC2-VPC 平台上佈建的每個叢集都有一個或多個 Amazon VPC 安全群組與其相關聯。Amazon VPC 會提供名為預設的 VPC 安全群組，系統會在您建立 VPC 時自動建立此群組。若您未在建立 Redshift 資源時指定其他 VPC 安全群組，您在 VPC 中啟動的每個叢集都會自動與預設 VPC 安全群組建立關聯。您可以在建立叢集時將 VPC 安全群組與叢集建立關聯，也可以在稍後修改叢集，將 VPC 安全群組與其建立關聯。

以下螢幕擷取畫面說明預設 VPC 安全群組的預設規則。

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

您可以視需要變更預設 VPC 安全群組的規則。

如果預設 VPC 安全群組夠您使用，則您無需再建立更多 VPC 安全群組。不過，您可以選擇性地建立額外的 VPC 安全群組，以妥善管理傳入存取權。例如，假設您是在 Amazon Redshift 叢集或 Serverless 工作群組上執行服務，而且為您的客戶提供幾個不同的服務層級。如果不想讓所有服務層級具有相同的存取，則您可能需要建立個別的 VPC 安全群組，每個服務層級一個。然後，您可將這些 VPC 安全群組與您的叢集或工作群組建立關聯。

您可以為 VPC 建立多達 100 個 VPC 安全群組，且可以將 VPC 安全群組與多個叢集和工作群組建立關聯。不過請注意，您可以與叢集或工作群組建立關聯的 VPC 安全群組數量有所限制。

Amazon Redshift 會立即套用對 VPC 安全群組所做的變更。因此，如果您已將 VPC 安全群組與叢集建立關聯，則已更新之 VPC 安全群組中的傳入叢集存取規則會立即套用。

您可以在 https://console.aws.amazon.com/vpc/ 建立和修改 VPC 安全群組。您也可以使用 AWS CLI、Amazon EC2 CLI 和 ，以程式設計方式管理 VPC 安全群組 AWS Tools for Windows PowerShell。如需使用 VPC 安全群組的相關資訊，請參閱《Amazon VPC 使用者指南》中的 VPC 的安全群組。

設 Amazon Redshift 叢集或 Amazon Redshift Serverless 工作群組的安全群組通訊設定

本主題可協助您設定安全群組，以適當地路由和接收網路流量。幾個常見的使用案例如下：

- 您開啟 Amazon Redshift 叢集或 Amazon Redshift Serverless 工作群組的公開存取性，但其未接收流量。為此，您必須設定傳入規則，以允許流量從網際網路抵達該項目。
- 您的叢集或工作群組無法公開存取，而且您使用 Redshift 預先設定的預設 VPC 安全群組來允許傳入流量。但是您必須使用預設安全群組以外的安全群組，而且這個自訂安全群組不允許傳入流量。您必須將其設定為允許通訊。

下列各節可協助您為每個使用案例選擇正確的回應，並向您說明如何根據您的要求來設定網路流量。您可以選擇性地使用這些步驟來設定來自其他私有安全群組的通訊。

> **Note**
>
> 大多數情況下，系統不會在 Amazon Redshift 中自動設定網路流量設定。這是因為這些設定在精細層級上可能會有所不同 (具體取決於流量來源是網際網路還是私有安全群組)，以及因為安全要求有所不同。

具有預設或自訂安全群組組態的公開存取性

如果您要建立叢集或工作群組，或您已經有叢集或工作群組，請執行下列組態步驟，以將其設為可公開存取。當您選擇預設安全群組或自訂安全群組時，都請這麼做：

1. 尋找網路設定：
   - 對於 Amazon Redshift 佈建叢集，請選擇屬性索引標籤，然後在網路和安全設定下選取叢集的 VPC。

- 對於 Amazon Redshift Serverless 工作群組，請選擇工作群組組態。從清單中選擇工作群組。然後，在資料存取下的網路和安全面板中，選擇編輯。

2. 設定 VPC 的網際網路閘道和路由表。您可以透過依名稱選擇 VPC 來啟動組態。其會開啟 VPC 儀表板。若要從網際網路連線到可公開存取的叢集或工作群組，就必須將網際網路閘道附加至路由表。您可以透過在 VPC 儀表板中選擇路由表來進行此設定。確認已為網際網路閘道的目標設定來源 0.0.0.0/0 或公開 IP CIDR。路由表必須與叢集所在的 VPC 相關聯。如需為 VPC 設定網際網路存取的相關資訊 (如此處所述)，請參閱 Amazon VPC 文件中的啟用網際網路存取。如需設定路由表的相關資訊，請參閱設定路由表。
3. 在設定網際網路閘道和路由表之後，請返回 Redshift 的網路設定。透過先選擇安全群組，再選擇傳入規則來開啟傳入存取。選擇 Edit inbound Rules (編輯傳入規則)。
4. 根據您的要求，為一或多個傳入規則選擇通訊協定和連接埠，以允許來自用戶端的流量。針對 RG 或 RA3 叢集，選取範圍為 5431-5455 或 8191-8215 的連接埠。完成時，請儲存每個規則。
5. 編輯可公開存取設定來加以啟用。您可以從叢集或工作群組的動作功能表執行此操作。

當您開啟可公開存取設定時，Redshift 會建立彈性 IP 地址。這是與 AWS 您的帳戶相關聯的靜態 IP 地址。在 VPC 外的用戶端可以使用此地址進行連線。

如需設定安全群組的相關資訊，請參閱 Amazon Redshift 安全群組。

您可以透過與用戶端連線來測試規則，如果您要連線到 Amazon Redshift Serverless，請執行以下操作。完成網路組態後，請與您的用戶端工具連線，例如 Amazon Redshift RSQL。使用您的 Amazon Redshift Serverless 網域作為主機，輸入以下內容：

```
rsql -h workgroup-name.account-id.region.amazonaws.com -U admin -d dev -p 5439
```

具有預設或自訂安全群組組態的私有存取性

如果在與叢集或工作群組通訊時未透過網際網路，這種方式便稱為可私有存取。如果您在建立安全群組時選擇了預設安全群組，安全群組中便會包含下列預設通訊規則：

- 一個傳入規則，允許指派給安全群組的所有資源所傳來的流量。
- 一個傳出規則，允許所有傳出流量。此規則的目的地是 0.0.0.0/0。在無類別域間路由 (CIDR) 標記法中，其代表所有可能的 IP 地址。

您可以透過選取叢集或工作群組的安全群組，在主控台中檢視規則。

如果您的叢集或工作群組和用戶端都使用預設安全群組，則不需要設定任何額外的組態來允許網路流量。但是，如果您刪除或變更 Redshift 或用戶端的預設安全群組中的任何規則，此做法就不再適用。在此情況下，您必須設定規則以允許傳入和傳出通訊。常見的安全群組組態如下：

- 針對用戶端 Amazon EC2 執行個體：
  - 傳入規則，允許用戶端的 IP 地址。
  - 傳出規則，允許針對 Redshift 用途所提供之所有子網路的 IP 地址範圍 (CIDR 區塊)。或者，您也可以指定 0.0.0.0/0，這是所有 IP 地址範圍。
- 針對您的 Redshift 叢集或工作群組：
  - 傳入規則，允許用戶端安全群組。
  - 傳出規則，允許流向 0.0.0.0/0 的流量。一般來說，傳出規則會允許所有傳出流量。您可以選擇性地新增傳出規則，允許流量流向用戶端安全群組。在這個選擇性案例中，不一定要有傳出規則，因為每個請求的回應流量都可以抵達執行個體。如需有關請求和回應行為的詳細資訊，請參閱《Amazon VPC 使用者指南》中的安全群組。

如果您變更針對 Redshift 用途指定的任何子網路或安全群組的組態，則可能需要相應地變更流量規則，以確保通訊能持續開啟。如需建立傳入和傳出規則的相關資訊，請參閱《Amazon VPC 使用者指南》中的 VPC CIDR 區塊。如需從用戶端連線至 Amazon Redshift 的相關資訊，請參閱在 Amazon Redshift 中設定連線。

Redshift 資源的子網路

如果要在虛擬私有雲端 (VPC) 中建立佈建叢集，則需建立子網路群組。每個 VPC 可以有一或多個子網路，也就是 VPC 內的 IP 位址子網路，這些子網路可讓您根據安全和操作的需要將資源分組。當您建立佈建叢集時，您會建立子網路群組以在 VPC 中指定一組子網路。在佈建叢集儀表板中，您可以在組態下尋找和編輯叢集子網路群組。在佈建叢集的初始組態設定期間，您會指定子網路群組，而 Amazon Redshift 會在其中一個子網路中建立叢集。如需 VPC 服務的詳細資訊，請參閱 Amazon VPC 產品詳細資訊頁面。

Amazon Redshift Serverless 工作群組的子網路組態類似於佈建叢集，但步驟略有不同。當您建立和設定 Serverless 工作群組時，您會指定工作群組的子網路，並將其新增至清單。您可以在 Serverless 儀表板中選取工作群組屬性，以檢視現有工作群組的子網路。這些子網路會在網路和安全屬性中提供。如需詳細資訊，請參閱使用命名空間建立工作群組。

如需建立 VPC 的詳細資訊，請前往 Amazon VPC 使用者指南文件。

為佈建叢集建立子網路群組，或為 Serverless 工作群組選擇子網路後，即可移除先前新增的子網路或新增更多子網路。您可以使用主控台或使用 API 操作進行這些變更。如需有關佈建叢集的 API 操作的詳細資訊，請參閱 ModifyClusterSubnetGroup。如需了解 Serverless 工作群組的 API 操作，請參閱 UpdateWorkgroup。

您可在子網路群組的其中一個子網路上佈建叢集。叢集子網路群組可讓您指定 Virtual Private Cloud (VPC) 中的子網路集合。

> **Warning**
>
> 在進行傳統調整大小、暫停和恢復、多可用區域容錯移轉或其他事件等叢集維護操作期間，您的佈建運算節點可能會移至 Amazon Redshift 叢集子網路群組內的另一個子網路。請注意，子網路群組中的所有子網路都必須具有相同的網路 ACL 傳入和傳出規則，以及相同的路由表路由。這可確保往返 Amazon Redshift 運算資源的連線，使其能夠在此類維護事件之後以最佳方式進行通訊和運作。避免將具有不同網路 ACL 或路由表組態的子網路新增至同一個 Amazon Redshift 叢集子網路群組。
>
> 如需設定子網路的詳細資訊，請參閱《Amazon VPC 使用者指南》中的 VPC 的子網路。如需 Redshift 多可用區域部署的詳細資訊，請參閱《Redshift 管理指南》中的 多可用區域部署。《Redshift 管理指南》中也涵蓋了 調整叢集大小。

建立叢集子網路群組

下列程序將逐步說明如何為佈建叢集建立子網路群組。您至少必須定義一個叢集子網路群組，才能在 VPC 中佈建叢集。

為佈建叢集建立叢集子網路群組

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ : // 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇組態，然後選擇子網路群組。子網路群組清單即會出現。
3. 選擇 Create cluster subnet group (建立叢集子網路群組) 以顯示建立頁面。
4. 輸入子網路群組的資訊，包括要新增的子網路。
5. 選擇 Create cluster subnet group (建立叢集子網路群組)，使用您選擇的子網路建立群組。

> **Note**
>
> 如需有關如何建立具有子網路集合的 Amazon Redshift Serverless 工作群組的資訊，請參閱《Amazon VPC 使用者指南》中的使用命名空間建立工作群組或建立子網路。

修改叢集子網路群組

建立子網路群組之後，您可以在 Amazon Redshift 主控台上修改其資訊。下列程序會逐步說明如何修改佈建叢集的子網路群組。

修改佈建叢集的叢集子網路群組

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇組態，然後選擇子網路群組。子網路群組清單即會出現。
3. 選擇要修改的子網路群組。
4. 針對 Actions (動作)，選擇 Modify (修改) 以顯示子網路群組的詳細資訊。
5. 更新子網路群組的資訊。
6. 選擇 Save (儲存) 以修改群組。

若要在某些情況下變更或移除子網路，您需要進行額外的步驟。例如，這篇 AWS 知識中心文章：如何將 Amazon Redshift 佈建叢集移至不同的子網路？便有內容涵蓋如何移動叢集的使用案例。

刪除佈建叢集的叢集子網路群組

當您使用完叢集子網路群組後，您應刪除該群組以將其清除。下列程序將引導您逐步完成刪除佈建叢集的子網路群組的步驟。

刪除叢集子網路群組

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇組態，然後選擇子網路群組。子網路群組清單即會出現。
3. 選擇要刪除的子網路群組，再選擇 Delete (刪除)。

> **Note**
>
> 您無法刪除叢集正在使用的叢集子網路群組。

封鎖對 VPC 和子網路的公開存取

VPC 封鎖公開存取 (BPA) 是一項集中式安全功能，可用來封鎖您在 中擁有VPCs 和子網路中的資源，使其 AWS 區域 無法連線網際網路，或透過網際網路閘道和僅限輸出網際網路閘道從網際網路連線。如果您在 中開啟此功能，預設 AWS 帳戶會影響 Amazon Redshift 使用的任何 VPC 或子網路。這表示，Amazon Redshift 會封鎖所有公開操作。

當您開啟 VPC BPA 並想要透過公有網際網路使用 Amazon Redshift API 時，您必須新增排除項目，才能針對 VPC 或子網路使用 Amazon EC2 API。排除項目可以有下列其中一種模式：

- 雙向：允許進出排除 VPC 和子網路的所有網際網路流量。
- 僅輸出：允許來自排除 VPC 和子網路的傳出網際網路流量。已封鎖對排除 VPC 和子網路的傳入網際網路流量。這僅適用於 BPA 設為雙向時。

VPC BPA 排除項目會將 VPC 內的整個 VPC 或特定子網路指定為可公開存取。在該界限內的網路介面是否具有公有網際網路的路由和存取權方面，該介面會遵守一般 VPC 網路控制，例如安全群組、路由表和網路 ACL。如需新增排除項目的詳細資訊，請參閱《Amazon VPC 使用者指南》中的建立和刪除排除項目。

佈建叢集

子網路群組是來自相同 VPC 的子網路組合。如果佈建叢集的子網路群組位於已開啟 VPC BPA 的帳戶中，則下列功能會遭到封鎖：

- 建立公有叢集
- 還原公有叢集
- 將私有叢集修改為公有
- 在群組內至少有一個公有叢集時，將開啟 VPC BPA 的子網路新增至子網路群組

無伺服器叢集

Redshift Serverless 不使用子網路群組。而是每個叢集都有自己的一組子網路。如果工作群組位於已開啟 VPC BPA 的帳戶中，則下列功能會遭到封鎖：

- 建立公有存取工作群組
- 將私有工作群組修改為公有
- 在工作群組為公有時，將開啟 VPC BPA 的子網路新增至工作群組

使用 Redshift 增強型 VPC 路由控制網路流量

當您使用 Amazon Redshift 增強型 VPC 路由時，Amazon Redshift 會根據 Amazon VPC 服務來強制叢集與資料儲存庫之間所有的 COPY 與 UNLOAD 流量通過您的虛擬私有雲端 (VPC)。使用增強型 VPC 路由時，您可以使用標準 VPC 功能，例如 VPC 安全群組、網路存取控制清單 (ACL)、VPC 端點、VPC 端點政策、網際網路閘道，以及網域名稱系統 (DNS) 伺服器，如《Amazon VPC 使用者指南》中所述。您可以使用這些功能來控制 Amazon Redshift 叢集與其他資源之間的資料流程。當您使用增強型 VPC 路由以透過 VPC 路由流量時，亦可使用 VPC 流程日誌來監控 COPY 與 UNLOAD 流量。

Amazon Redshift 叢集和 Amazon Redshift Serverless 工作群組都支援增強型 VPC 路由。您無法搭配 Redshift Spectrum 使用增強型 VPC 路由。如需詳細資訊，請參閱使用增強型 VPC 路由查詢資料湖資料表。

如果未開啟增強型 VPC 路由，Amazon Redshift 會透過網際網路路由流量，包括網路中其他服務的流量 AWS 。

> **Important**
>
> 由於增強型 VPC 路由會影響 Amazon Redshift 存取其他資源的方式，COPY 與 UNLOAD 命令可能會失敗，除非您正確設定 VPC。您必須在叢集的 VPC 與資料資源之間特別建立網路路徑，如下所述。

當您在已開啟增強型 VPC 路由的叢集上執行 COPY 或 UNLOAD 命令時，您的 VPC 會使用「最嚴格」(或最獨特) 可用的網路路徑，將流量路由至指定的資源。

例如，您可以在 VPC 中設定下列路徑：

- VPC 端點 – 對於與叢集或工作群組位於相同 AWS 區域中的 Amazon S3 儲存貯體流量，您可以建立 VPC 端點，將流量直接導向儲存貯體。當您使用 VPC 端點時，可以連接端點政策以管理 Amazon S3 的存取。如需搭配 Redshift 使用端點的相關資訊，請參閱 使用 VPC 端點控制資料庫流量。如果您使用 Lake Formation，您可以在 Lake Formation 和介面 VPC 端點 () 的 AWS VPC 與 Lake Formation 之間建立私有連線的詳細資訊。 AWSAWS PrivateLink

> **Note**
>
> 當您搭配 Amazon S3 VPC Gateway 端點使用 Redshift VPC 端點時，您必須在 Redshift 中啟用增強型 VPC 路由。如需詳細資訊，請參閱 Amazon S3 的閘道端點。

- NAT 閘道 – 您可以連接到另一個 AWS 區域中的 Amazon S3 儲存貯體，也可以連接到網路中的 AWS 其他服務。您也可以存取 AWS 網路外部的主機執行個體。若要這麼做，請設定網路位址轉譯 (NAT) 閘道，如《Amazon VPC 使用者指南》中所述。
- 網際網路閘道 — 若要連線至 VPC 外的 AWS 服務，您可以將網際網路閘道附加至您的 VPC 子網路，如《Amazon VPC 使用者指南》中所述。若要使用網際網路閘道，您的叢集或工作群組必須可公開存取，以便讓其他服務與其進行通訊。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 VPC 端點。

> **Note**
>
> 增強型 VPC 路由也會影響 Redshift 在使用者登入時對 IAM Identity Center AWS 進行的呼叫。如果您的叢集或工作群組使用 AWS IAM Identity Center 身分驗證，您必須為這些服務建立介面 VPC 端點。否則，登入會失敗。如需搭配 IAM Identity Center 使用增強型 VPC AWS 路由的詳細資訊，請參閱 搭配增強型 VPC AWS 路由使用 IAM Identity Center 身分驗證。

使用增強型 VPC 路由無需額外付費。某些操作可能需要額外的數據傳輸費。這些操作包括 UNLOAD 到不同區域中的 Amazon S3 AWS 。從 Amazon EMR 或 Secure Shell (SSH) 透過公有 IP 地址 COPY。如需定價的相關資訊，請參閱 Amazon EC2 定價。

主題

- 使用 VPC 端點控制資料庫流量
- 開啟增強型 VPC 路由
- 使用增強型 VPC 路由查詢資料湖資料表

使用 VPC 端點控制資料庫流量

您可以使用 VPC 端點，在 VPC 中的 Amazon Redshift 叢集或 Serverless 工作群組與 Amazon Simple Storage Service (Amazon S3) 之間建立受管連線。當您這麼做時，資料庫與 Amazon S3 上的資料之

間的 COPY 與 UNLOAD 流量會停留在 Amazon VPC 中。您可以將端點政策連接至您的端點，更緊密地管理您資料的存取。例如，您可以將政策新增至您的 VPC 端點，僅允許資料上傳至您帳戶中特定的 Amazon S3 儲存貯體。

若要使用 VPC 端點，請為資料倉儲所在的 VPC 建立 VPC 端點，然後開啟增強型 VPC 路由。您可以在建立叢集或工作群組時開啟增強型 VPC 路由，也可以修改 VPC 中的叢集或工作群組以使用增強型 VPC 路由。

VPC 端點使用路由表來控制 VPC 中的叢集或工作群組與 Amazon S3 之間流量的路由。與指定之路由表關聯之子網路中的所有叢集和工作群組會自動使用該端點來存取服務。

您的 VPC 使用最具體或最嚴格且匹配您流量的路由，以判斷如何路由流量。例如，假設您的路由表中有路由處理指向網際網路閘道和 Amazon S3 端點的所有網際網路流量 (0.0.0.0/0)。在此情況下，端點路由會優先處理所有以 Amazon S3 為目的地的流量。這是因為 Amazon S3 服務的 IP 地址比 0.0.0.0/0 更明確。在此範例中，所有其他網際網路流量會流向網際網路閘道，包括以其他 AWS 區域區域中的 Amazon S3 儲存貯體為目的地的流量。

如需建立端點的相關資訊，請參閱《Amazon VPC 使用者指南》中的建立 VPC 端點。

您使用端點政策，控制從您的叢集或工作群組對於存有您資料檔案之 Amazon S3 儲存貯體的存取。如需更精確的控制，您可以選擇連接自訂端點政策。如需詳細資訊，請參閱《AWS PrivateLink 指南》中的使用端點政策控制對服務的存取。

> **Note**
>
> AWS Database Migration Service (AWS DMS) 是一種雲端服務，可讓您遷移關聯式資料庫、資料倉儲和其他類型的資料存放區。它可以連接到任何 AWS 來源或目標資料庫，包括已啟用 VPC 的 Amazon Redshift 資料庫，並具有一些組態限制。支援 Amazon VPC 端點可讓您更輕鬆地 AWS DMS 維護end-to-end網路安全。如需搭配 Redshift 使用的詳細資訊 AWS DMS，請參閱AWS Database Migration Service 《 使用者指南》中的將 VPC 端點設定為 AWS DMS 來源和目標端點。

使用端點無需額外付費。需支付標準數據傳輸與資源使用費。如需有關定價的詳細資訊，請參閱 Amazon EC2 定價。

開啟增強型 VPC 路由

您可以在建立或修改叢集時，以及建立或修改 Amazon Redshift Serverless 工作群組時，開啟增強型 VPC 路由。

如要使用增強型 VPC 路由，您的叢集或 Serverless 工作群組必須符合以下需求與限制條件：

- 您的叢集必須在 VPC 中。

  如果您連接 Amazon S3 VPC 端點，VPC 端點僅用於存取相同區域中的 Amazon S3 儲存貯 AWS 體。若要存取另一個 AWS 區域中的儲存貯體（不使用 VPC 端點）或存取其他服務 AWS ，請讓您的叢集或 Serverless 工作群組可公開存取，或使用網路位址轉譯 (NAT) 閘道。如需詳細資訊，請參閱在 VPC 中建立 Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組。
- 您必須在 VPC 中啟用網域名稱服務 (DNS) 解析。或者，如果您使用自己的 DNS 伺服器，請確定對 Amazon S3 提出的 DNS 請求可正確解析為 AWS所維護的 IP 地址。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 使用 DNS 與您的 VPC 搭配。
- 您的 VPC 必須啟用 DNS 主機名稱。DNS 主機名稱預設為啟用。
- 您的 VPC 端點政策必須允許存取 Amazon Redshift 中的 COPY、UNLOAD 或 CREATE LIBRARY 呼叫所使用的任何 Amazon S3 儲存貯體，包括存取任何涉及的資訊清單檔案。對於來自遠端主機的 COPY，您的端點政策必須允許存取每個主機機器。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 COPY、UNLOAD 和 CREATE LIBRARY 的 IAM 許可。

開啟佈建叢集的增強型 VPC 路由

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ ：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇佈建叢集儀表板，然後選擇建立叢集並輸入叢集詳細資訊屬性。
3. 若要顯示 Additional configurations (其他組態) 區段，請選擇關閉 Use defaults (使用預設值)。
4. 導覽至網路和安全區段。
5. 若要開啟增強型 VPC 路由，請選擇開啟以強制透過 VPC 傳輸叢集流量。
6. 選擇 Create cluster (建立叢集) 以建立叢集。該叢集可能需要幾分鐘方可使用。

開啟 Amazon Redshift Serverless 的增強型 VPC 路由

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/ ：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇 Serverless 儀表板，然後選擇建立工作群組並輸入工作群組的屬性。
3. 導覽至網路和安全區段。
4. 選取開啟增強型 VPC 路由以透過 VPC 來路由網路流量。
5. 選擇下一步並完成工作群組屬性的輸入，直到您建立工作群組為止。

使用增強型 VPC 路由查詢資料湖資料表

開啟增強型 VPC 路由時，Amazon Redshift 會透過您的 VPC 路由流量。資料湖資料表將其資料存放在 Amazon S3 中，其中繼資料存放在 中 AWS Glue Data Catalog，兩者都在 VPC 之外。若要讓資料湖查詢成功，您的叢集或工作群組必須能夠連接 Amazon S3 和 AWS Glue。如果您的資料湖資料表是由 AWS Lake Formation 管理，它也必須能夠連線到 Lake Formation。增強型 VPC 路由會影響 Amazon Redshift 存取這些外部資源的方式，因此除非您正確設定 VPC，否則查詢可能會失敗。

在叢集或工作群組執行的 VPC 和子網路中建立下列 VPC 端點，並將它們與適當的路由表建立關聯。整合的資料湖查詢引擎和 Amazon Redshift Spectrum 都需要這些端點：

- Amazon S3 閘道端點 – 透過 AWS 網路為叢集或工作群組提供路由至 Amazon S3，以讀取資料湖資料檔案。如需步驟，請參閱《Amazon VPC 使用者指南》中的 Amazon S3 的閘道端點。
- AWS Glue 介面端點 – 為您的叢集或工作群組提供通往 的路由， AWS Glue Data Catalog 以解析資料湖結構描述和資料表。如需步驟，請參閱《Amazon VPC 使用者指南》中的建立介面端點。
- AWS Lake Formation 介面端點 – 只有在資料湖資料表由 AWS Lake Formation 管理時才需要。為您的叢集或工作群組提供 Lake Formation 的路由。如需步驟，請參閱《Amazon VPC 使用者指南》中的建立介面端點。

無論您如何查詢外部資料表，端點組態都是相同的，但流量路由方式會因查詢引擎而有所不同。下列各節說明 RG 佈建叢集和 Amazon Redshift Serverless 上整合資料湖查詢引擎的行為，以及 RA3 和 DC2 佈建叢集上的 Redshift Spectrum 的行為。

適用於 RG 佈建叢集和 Amazon Redshift Serverless 的整合式資料湖查詢引擎

RG 佈建叢集和 Amazon Redshift Serverless 包含整合式資料湖查詢引擎。此引擎會在您 VPC 內的叢集或工作群組自己的運算資源上執行。開啟增強型 VPC 路由並備妥必要的 VPC 端點時，存取 Amazon S3 並 AWS Glue 來自您的 VPC 內運算，並流經這些 VPC 端點。此流量會保留在您的 VPC 界限和 AWS 網路上。此存取權是根據連接至叢集或工作群組的 IAM 角色進行授權。若要進一步管理此流量，您可以修改 IAM 角色和連接到 Amazon S3 儲存貯體的政策。

適用於 RA3 和 DC2 佈建叢集的 Amazon Redshift Spectrum

Redshift Spectrum 會在 Amazon Redshift 擁有但位於 VPC 外部的 AWS 受管資源上執行。因此，即使開啟增強型 VPC 路由，Redshift Spectrum 傳送至 Amazon S3 的流量也不會通過您的 VPC，但叢集仍需要先前所述的 Amazon S3 和 AWS Glue VPC 端點來執行資料湖查詢。

Redshift Spectrum 和 Amazon S3 之間的流量會透過 AWS 私有網路安全地路由至 VPC 外部。傳輸中的流量是使用 Amazon Signature 第 4 版通訊協定 (SIGv4) 來簽署，並使用 HTTPS 來加密。此流量

是根據連接到 Amazon Redshift 叢集的 IAM 角色而獲得授權。若要進一步管理 Redshift Spectrum 流量，您可以修改叢集的 IAM 角色和已連接到 Amazon S3 儲存貯體的政策。

資料湖查詢的許可政策組態

在 Amazon S3 中查詢資料湖資料表時，請考慮下列事項：

- Amazon S3 儲存貯體存取政策和 IAM 角色
- 擔任 IAM 角色的許可
- 記錄和稽核 Amazon S3 存取
- 存取 AWS Glue 或 Amazon Athena

Amazon S3 儲存貯體存取政策和 IAM 角色

您可以使用連接到儲存貯體的儲存貯體政策，以及使用連接到佈建叢集的 IAM 角色，來控制 Amazon S3 儲存貯體中資料的存取權。

如果 Amazon S3 儲存貯體使用的儲存貯體政策僅限於存取指定的 VPC 端點，佈建叢集上的 Redshift Spectrum 便無法存取該儲存貯體中儲存的資料。反之，請使用儲存貯體政策，限制只能存取特定委託人，例如特定 AWS 帳戶或特定使用者。啟用增強型 VPC 路由時，RG 佈建叢集和 Amazon Redshift Serverless 上的整合資料湖查詢引擎可以存取此類儲存貯體，因為其 Amazon S3 存取會流經您的 VPC 端點。

對於獲授權存取儲存貯體的 IAM 角色，請使用只允許由 Amazon Redshift 服務主體擔任該角色的信任關係。連接到叢集時，該角色只能用於 Amazon Redshift 的內容中，不能在叢集外共用。如需詳細資訊，請參閱限制對 IAM 角色的存取。服務控制政策 (SCP) 也可用於進一步限制角色，請參閱《AWS Organizations 使用者指南》中的防止 IAM 使用者和角色進行指定變更，且指定的管理員角色除外。

> **Note**
>
> 若要使用 Redshift Spectrum，則不能有任何會阻止使用 Amazon S3 預先簽章 URL 的 IAM 政策。Amazon Redshift Spectrum 產生的預先簽章 URL 有效期為 1 小時，如此 Amazon Redshift 就有足夠的時間從 Amazon S3 儲存貯體載入所有檔案。Redshift Spectrum 掃描的每個檔案都會產生唯一的預先簽章 URL。對於包含 `s3:signatureAge` 動作的儲存貯體政策，請務必將值設定為至少 3,600,000 毫秒。

下列儲存貯體政策範例允許存取 AWS 帳戶 擁有的指定儲存貯體`123456789012`。

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/*"
            ]
        }
    ]
}
```

擔任 IAM 角色的許可

連接到叢集的角色應該具有只允許 Amazon Redshift 服務擔任該角色的信任關係，如下所示。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "redshift.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
```

```
}
```

如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 Redshift Spectrum 的 IAM 政策。

記錄和稽核 Amazon S3 存取

使用 Amazon Redshift 增強型 VPC 路由的一項好處是所有 COPY 和 UNLOAD 流量都記錄在 VPC 流程日誌中。源自 Redshift Spectrum 到 Amazon S3 的流量不會通過 VPC，所以沒有記錄在 VPC 流程日誌中。當 Redshift Spectrum 存取 Amazon S3 中的資料時，它會在 AWS 帳戶內容和個別角色權限中執行這些操作。您可以在 AWS CloudTrail 和 Amazon S3 中使用伺服器存取日誌記錄，以記錄和稽核 Amazon S3。

確定 S3 IP 範圍已新增至您的允許清單。若要進一步了解所需的 S3 IP 範圍，請參閱網路隔離。

AWS CloudTrail 日誌

若要追蹤對 Amazon S3 中的物件的所有存取，包括 Redshift Spectrum 存取，請對 Amazon S3 物件啟用 CloudTrail 記錄。

您可以使用 CloudTrail 來檢視、搜尋、下載、封存、分析和回應整個 AWS 基礎設施的帳戶活動。如需詳細資訊，請參閱 CloudTrail 入門。

根據預設，CloudTrail 只會追蹤儲存貯體層級的動作。若要追蹤物件層級的動作 (例如 `GetObject`)，請對每一個已記錄的儲存貯體啟用資料和管理事件。

Amazon S3 伺服器存取記錄

伺服器存取記錄日誌，應儲存貯體要求，提出的詳細記錄。存取記錄資訊在安全與存取稽核中相當實用。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的如何啟用伺服器存取記錄。

如需詳細資訊，請參閱 AWS 安全部落格文章如何使用儲存貯體政策和套用Defense-in-Depth以協助保護您的 Amazon S3 資料。

存取 AWS Glue 或 Amazon Athena

Redshift Spectrum 和整合的資料湖查詢引擎都會存取 AWS Glue 或 Athena 中的資料目錄。另一個選項是使用專用的 Hive 中繼存放區來處理資料目錄。

到達 的建議方法是為 AWS Glue Data Catalog 建立介面 VPC 端點 (AWS PrivateLink) AWS Glue。當您使用 VPC 介面端點時，VPC 和 之間的通訊 AWS Glue 會在 AWS 網路中路由。如需詳細資訊，請參閱建立界面端點。

或者，若要連接 AWS Glue 和 Athena 的公有端點，請使用網際網路閘道或 NAT 閘道設定您的 VPC，並設定您的 VPC 安全群組以允許這些公有端點的傳出流量。您可以在 VPC 中設定下列路徑：

- 網際網路閘道 – 若要連線至 VPC 以外的 AWS 服務，您可以將網際網路閘道連接至 VPC 子網路，如 Amazon VPC 使用者指南中所述。若要使用網際網路閘道，佈建叢集必須有公有 IP 位址，才能讓其他服務與其進行通訊。
- NAT 閘道 – 若要連線至另一個 AWS 區域中的 Amazon S3 儲存貯體或網路中的 AWS 其他服務，請設定網路位址轉譯 (NAT) 閘道，如 Amazon VPC 使用者指南中所述。您也可以使用此組態來存取 AWS 網路外的主機執行個體。

Amazon Redshift 事件

Amazon Redshift 會追蹤叢集事件，並在 AWS 您的帳戶中保留相關資訊數週。對於每個事件，Amazon Redshift 會報告相關的資訊，如事件發生的日期、描述、事件來源 (例如，叢集、參數群組或快照)，以及來源 ID。

Amazon Redshift 會事先提供某些事件的通知。這些事件的事件類別為 `pending`。例如，如果叢集的其中一個節點要硬體更新，我們會傳送事先通知。如同其他 Amazon Redshift 事件一樣，您也可以訂閱待定事件。如需詳細資訊，請參閱Amazon Redshift 叢集事件通知訂閱。

您可以使用 Amazon Redshift 管理主控台、Amazon Redshift API 或 AWS SDKs來取得事件資訊。您可以取得事件清單或套用篩選條件，例如事件持續時間或開始與結束日期，以取得特定期間的事件資訊。

您可以取得特定來源類型所產生的事件，例如叢集事件或參數群組事件。「來源」資料欄會顯示觸發指定動作的資源名稱和資源類型。

您可以建立指定一組事件篩選條件的 Amazon Redshift 事件通知訂閱。發生符合篩選條件的事件時，Amazon Redshift 便會使用 Amazon Simple Notification Service 來主動通知您已發生事件。

如需依照來源類型與類別分組的 Amazon Redshift 事件清單，請參閱 the section called “佈建叢集事件通知”。

Amazon Redshift 叢集事件通知訂閱

Amazon Redshift 會使用 Amazon Simple Notification Service (Amazon SNS) 來傳達 Amazon Redshift 事件的通知。建立 Amazon Redshift 事件訂閱，即可啟用通知。您可以在指定的叢集、快照、安全群組或參數群組發生事件時收到通知。建立訂閱最簡單的方式是使用 Amazon SNS 主控台。如需建立和訂閱 Amazon SNS 主題的詳細資訊，請參閱 Amazon SNS 入門。

在 Amazon Redshift 訂閱中，您需要指定 Amazon Redshift 事件的一組篩選條件和 Amazon SNS 主題。每當發生符合篩選條件的事件時，Amazon Redshift 就會向 Amazon SNS 主題發佈通知訊息。

然後，Amazon SNS 會將訊息傳送給任何已有該主題之 Amazon SNS 訂閱的 Amazon SNS 取用者。傳送給 Amazon SNS 消費者的訊息可以是 AWS 區域 Amazon SNS 支援的任何形式，例如電子郵件、文字訊息或對 HTTP 端點的呼叫。例如，所有區域均支援電子郵件通知，但簡訊通知只能針對美國東部 (維吉尼亞北部) 來建立。

> **Note**
>
> 目前，您只能建立 Amazon SNS 標準主題的事件訂閱 (而非 Amazon SNS FIFO 主題的事件訂閱)。如需詳細資訊，請參閱《Amazon Simple Notification Service 開發人員指南》中的 Amazon SNS 事件來源。

當您建立事件通知訂閱時，您需要指定一或多個事件篩選條件。只要發生符合所有篩選條件的事件，Amazon Redshift 就會透過訂閱傳送通知。篩選條件包含來源類型 (例如叢集或快照)、來源 ID (例如叢集或快照的名稱)、事件類別 (例如監控或安全性) 和事件嚴重性 (例如 INFO 或 ERROR)。

如果您使用 CLI 或 API 建立事件通知訂閱，您必須建立 Amazon Simple Notification Service 主題，並使用 Amazon SNS 主控台或 Amazon SNS API 訂閱該主題。您也將必須保留主題的 Amazon Resource Name (ARN)，因為在提交 CLI 命令或 API 動作時會用到它。

您可以透過在 No中將已啟用選項按鈕設定為 ， AWS 管理主控台 或使用 Amazon Redshift CLI 或 API 將 `Enabled` 參數設定為 ，輕鬆關閉通知`false`而不刪除訂閱。

Amazon Redshift 事件訂閱可指定這些事件條件：

- 來源類型，其值包括叢集、快照、參數群組和安全群組。
- 資源的來源 ID，例如 `my-cluster-1` 或 `my-snapshot-20130823`。ID 所屬的資源必須與事件訂閱位於相同的 AWS 區域。
- 事件類別，其值包括組態、管理、監控、安全與待定。
- 事件嚴重性，其值包括 INFO 或 ERROR。

事件條件可單獨指定，但您必須先指定來源類型，才能在主控台中指定來源 ID。例如，您可指定事件類別，但不需指定來源類型、來源 ID 或嚴重性。雖然您可以為非來源類型所指定類型的資源指定來源 ID，但將不會針對這些資源傳送事件通知。例如，如果您指定來源類型為叢集，ID 為安全群組，而該安全群組發生的事件均與來源類型篩選條件不符，將不會傳送這些事件的通知。

發生符合訂閱指定之所有條件的任何事件時，Amazon Redshift 便會傳送通知。傳回之事件集合的一些範例：

- 訂閱指定來源類型為叢集，來源 ID 為 my-cluster-1，類別為監控，嚴重性為 ERROR。訂閱將針對來自 my-cluster-1、嚴重性為 ERROR 的監控事件傳送通知。
- 訂閱指定來源類型為叢集，類別為組態，嚴重性為 INFO。訂閱將針對 AWS 帳戶中任何 Amazon Redshift 叢集嚴重性為 INFO 的組態事件傳送通知。

- 訂閱指定類別為組態，嚴重性為 INFO。訂閱會從 AWS 帳戶中任何 Amazon Redshift 資源傳送嚴重性為 INFO 的組態事件通知。
- 訂閱指定嚴重性為 ERROR。訂閱會從 AWS 帳戶中任何 Amazon Redshift 資源傳送嚴重性為 ERROR 的所有事件通知。

如果刪除或移除名稱在現有訂閱中做為來源 ID 參考的物件，訂閱將保持作用中，但不會傳送該物件的任何事件。如果之後建立新物件，且物件使用訂閱來源 ID 中所參考的相同名稱，訂閱將開始傳送新物件的事件通知。

Amazon Redshift 會將事件通知發佈至以 Amazon Resource Name (ARN) 識別的 Amazon SNS 主題。使用 Amazon Redshift 主控台建立事件訂閱時，您可指定現有的 Amazon SNS 主題，或要求主控台在建立訂閱時建立主題。

傳送至 Amazon SNS 主題的所有 Amazon Redshift 事件通知，接著會傳送給所有訂閱該主題的 Amazon SNS 取用者。使用 Amazon SNS 主控台來變更 Amazon SNS 主題，例如新增或移除取用者對主題的訂閱。

以下幾節會列出您可以收到通知的所有類別和事件。亦提供有關訂閱和使用 Amazon Redshift 事件訂閱的資訊。

建立事件通知訂閱

您可以建立 Amazon Simple Notification Service (Amazon SNS) 事件通知訂閱，在指定的 Amazon Redshift 叢集、快照、安全群組或參數群組發生事件時傳送通知。這些通知將傳送至 SNS 主題，接著再傳送訊息給訂閱該主題的任何 SNS 消費者。

傳送給消費者的 SNS 訊息可以是 AWS 區域 Amazon SNS 支援的任何通知形式，例如電子郵件、文字訊息或對 HTTP 端點的呼叫。例如，所有區域均支援電子郵件通知，但簡訊通知只能針對美國東部 (維吉尼亞北部) 來建立。如需詳細資訊，請參閱Amazon Redshift 佈建叢集事件通知。

建立事件訂閱

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇事件。
3. 選擇 Event subscription (事件訂閱) 標籤，然後選擇 Create event subscriptions (建立事件訂閱)。
4. 輸入事件訂閱的屬性，例如名稱、來源類型、類別和嚴重性。您也可以啟用 Amazon SNS 主題來接收事件通知。

5. 選擇 Create event subscriptions (建立事件訂閱) 以建立訂閱。

Amazon Redshift 佈建叢集事件通知

本節說明各 Amazon Redshift 來源類型的事件 ID 和類別。

叢集來源類型的類別和事件

下表顯示來源類型為叢集時的事件類別和事件清單。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
Configuration	REDSHIFT-EVENT-1000	INFO	參數群組 [參數群組名稱] 於 [時間] 更新。如果您只變更動態參數，則現在正在修改相關聯的叢集。如果您變更靜態參數，則會在您重新啟動相關聯的叢集時套用所有更新，包括動態參數。
Configuration	REDSHIFT-EVENT-1001	INFO	您的 Amazon Redshift 叢集 [叢集名稱] 於 [時間] 修改為使用參數群組 [參數群組名稱]。
Configuration	REDSHIFT-EVENT-1500	ERROR	Amazon VPC [VPC 名稱] 不存在。您對叢集 [叢集名稱] 的組態變更尚未套用。請造訪 AWS 管理主控台 以修正問題。
Configuration	REDSHIFT-EVENT-1501	ERROR	您為 Amazon VPC [VPC 名稱] 指定的客戶子網路 [子網路名稱] 不存在或無效。您對叢集 [叢集名稱] 的組態變更尚未套用。請造訪 AWS 管理主控台 以修正問題。
Configuration	REDSHIFT-EVENT-1502	ERROR	叢集子網路群組 [子網路群組名稱] 中的子網路沒有可用的 IP 地址。無法建立叢集 [叢集名稱]。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
Configuration	REDSHIFT-EVENT-1503	ERROR	Amazon VPC [VPC 名稱] 未連接網際網路閘道。您對叢集 [叢集名稱] 的組態變更尚未套用。請造訪 AWS 管理主控台 以修正問題。
Configuration	REDSHIFT-EVENT-1504	ERROR	叢集 [叢集名稱] 的 HSM 無法連線。
Configuration	REDSHIFT-EVENT-1505	ERROR	叢集 [叢集名稱] 的 HSM 無法註冊。請嘗試不同的組態。
Configuration	REDSHIFT-EVENT-1506	ERROR	Amazon Redshift 已超過您帳戶的彈性網路界面限制。刪除最多 【彈性網路介面的數量上限】 個彈性網路介面，或請求增加 EC2 每個 AWS 區域的網路介面數量限制。
Configuration	REDSHIFT-EVENT-1509	ERROR	無法建立 Amazon Redshift 叢集 [叢集名稱]，因為您的帳戶已達到 VPC 端點限制。刪除未使用的 VPC 端點，或請求提高 VPC 端點的限制。 如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 VPC 端點。
Configuration	REDSHIFT-EVENT-1510	ERROR	我們偵測到在 Amazon Redshift 叢集 [叢集名稱] 上載入範例資料的嘗試失敗。若要載入範例資料，請先設定 VPC 以便能夠存取 Amazon S3 儲存貯體，然後再建立新叢集並載入範例資料。 如需詳細資訊，請參閱《Amazon Redshift 管理指南》中的啟用增強型 VPC 路由。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
Configuration	REDSHIFT-EVENT-1511	ERROR	無法建立 Amazon Redshift 叢集 [叢集名稱]，因為您已超過帳戶的彈性 IP 地址限制。使用 Amazon EC2 刪除未使用的彈性 IP 地址或請求提高限制。
管理	REDSHIFT-EVENT-2000	INFO	您的 Amazon Redshift 叢集 [叢集名稱] 已建立且已就緒可供使用。
管理	REDSHIFT-EVENT-2001	INFO	您的 Amazon Redshift 叢集 [叢集名稱] 於 [時間] 刪除。最後的快照 [已/未] 儲存。
管理	REDSHIFT-EVENT-2002	INFO	叢集 [叢集名稱] 的 VPC 安全群組已於 [UTC 時間] 更新。
管理	REDSHIFT-EVENT-2003	INFO	叢集 [叢集名稱] 於 [UTC 時間] 開始維護。
管理	REDSHIFT-EVENT-2004	INFO	叢集 [叢集名稱] 於 [UTC 時間] 完成維護。
管理	REDSHIFT-EVENT-2006	INFO	叢集 [叢集名稱] 於 [UTC 時間] 開始調整大小。叢集為唯讀模式。
管理	REDSHIFT-EVENT-2007	INFO	已確認叢集 [叢集名稱] 的調整大小請求。
管理	REDSHIFT-EVENT-2008	INFO	建立新 Amazon Redshift 叢集 [叢集名稱] 快照 [快照名稱] 的還原操作於 [時間] 開始。若要監控還原流程，請造訪 AWS 管理主控台。
管理	REDSHIFT-EVENT-2013	INFO	您的 Amazon Redshift 叢集 [叢集名稱] 於 [時間] 重新命名。
管理	REDSHIFT-EVENT-2014	INFO	已收到 Amazon Redshift 叢集 [叢集名稱] 的資料表還原要求。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
管理	REDSHIFT-EVENT-2015	INFO	Amazon Redshift 叢集 [叢集名稱] 於 [時間] 取消資料表還原。
管理	REDSHIFT-EVENT-2016	INFO	您的 Amazon Redshift 叢集 [叢集名稱] 於 [時間] 開始替換。
管理	REDSHIFT-EVENT-2017	INFO	您的 Amazon Redshift 叢集 [叢集名稱] 已於 [時間] 開始客戶所啟動的維護。叢集在維護期間可能無法使用。
管理	REDSHIFT-EVENT-2018	INFO	您的 Amazon Redshift 叢集 [叢集名稱] 已於 [時間] 完成客戶所啟動的維護。
管理	REDSHIFT-EVENT-2019	ERROR	您的 Amazon Redshift 叢集 [叢集名稱] 於 [時間] 執行客戶所啟動的維護失敗。正在將叢集返回其原始狀態。
管理	REDSHIFT-EVENT-2020	INFO	您的 Amazon Redshift 叢集 [叢集名稱] 追蹤已從 [舊追蹤] 修改為 [新追蹤]。
管理	REDSHIFT-EVENT-2021	ERROR	從容量集區取得容量時，Amazon Redshift 叢集 [叢集名稱] 的 [操作] 未成功。我們正在設法取得容量，但現在，我們已取消您的請求。刪除此叢集，稍後再試。
管理	REDSHIFT-EVENT-2022	ERROR	從容量集區取得容量時，Amazon Redshift 叢集 [叢集名稱] 的 [操作] 未成功。我們正在設法取得容量，但現在，我們已取消您的請求。[替代可用區域] 中有可用容量。請刪除此叢集，並在替代可用區域中重試。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
管理	REDSHIFT-EVENT-2023	ERROR	我們偵測到您的單一節點 Amazon Redshift 叢集 [叢集名稱] 發生硬體故障，這可能已導致查詢失敗，或叢集的可用性不穩定。從容量集區取得容量時，取代叢集未成功。您將需要從快照還原新叢集。刪除此叢集，選取最新可用的快照，然後從該快照還原新叢集。這將會自動佈建在狀況良好的硬體上。
管理	REDSHIFT-EVENT-2024	ERROR	我們偵測到您的單一節點 Amazon Redshift 叢集 [叢集名稱] 發生硬體故障，這可能已導致查詢失敗，或叢集的可用性不穩定。從容量集區取得容量時，取代叢集未成功。可用區域 [替代可用區域] 中有可用容量。刪除此叢集，選取最新可用的快照，然後從該快照還原新叢集。這將會自動佈建在狀況良好的硬體上。
管理	REDSHIFT-EVENT-3011	INFO	Amazon Redshift 叢集「[叢集名稱]」的彈性調整大小於 [時間] 開始。我們在調整大小期間會保持資料庫連線。在此操作期間，部分查詢和連線可能終止或逾時。
管理	REDSHIFT-EVENT-3012	INFO	我們收到叢集 '[叢集名稱]' 在 [時間] 開始的彈性調整大小請求。調整大小開始時，我們會提供事件通知。
待定	REDSHIFT-EVENT-2025	INFO	叢集 [cluster name] 的資料庫將在 [start time] 和 [end time] 之間更新。將無法存取您的叢集。請妥善規劃。
待定	REDSHIFT-EVENT-2026	INFO	叢集 [cluster name] 將在 [start time] 和 [end time] 之間更新。將無法存取您的叢集。請妥善規劃。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
監控	REDSHIFT-EVENT-2050	INFO	在 Amazon Redshift 叢集 [叢集名稱] 上偵測到硬體問題。已於 [時間] 啟動替換要求。
監控	REDSHIFT-EVENT-3000	INFO	您的 Amazon Redshift 叢集 [叢集名稱] 於 [時間] 重新開機。
監控	REDSHIFT-EVENT-3001	INFO	您的 Amazon Redshift 叢集 [叢集名稱] 上的某一節點於 [時間] 自動替換，叢集目前正常運作中。
監控	REDSHIFT-EVENT-3002	INFO	您的 Amazon Redshift 叢集 [叢集名稱] 已完成調整規模，叢集目前可供讀寫。該規模調整於 [時間] 啟動，並花了 [小時數] 小時完成。
監控	REDSHIFT-EVENT-3003	INFO	已成功從快照 [快照名稱] 建立 Amazon Redshift 叢集 [叢集名稱]，且叢集目前可供使用。
監控	REDSHIFT-EVENT-3007	INFO	您的 Amazon Redshift 快照 【快照名稱】 已在 【時間】 成功從 【來源 AWS 區域】 複製到 【目的地 AWS 區域】。
監控	REDSHIFT-EVENT-3008	INFO	Amazon Redshift 叢集 [叢集名稱] 於 [時間] 開始資料表還原。
監控	REDSHIFT-EVENT-3009	INFO	Amazon Redshift 叢集 [叢集名稱] 於 [時間] 成功完成資料表還原。
監控	REDSHIFT-EVENT-3010	ERROR	Amazon Redshift 叢集 [叢集名稱] 於 [時間] 執行資料表還原失敗。
監控	REDSHIFT-EVENT-3013	ERROR	由於 [原因]，對 Amazon Redshift 叢集 [叢集名稱] 所要求的彈性調整大小操作於 [時間] 失敗。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
監控	REDSHIFT-EVENT-3014	INFO	Amazon Redshift 叢集 [叢集名稱] 於 [時間] 重新開機。
待定	REDSHIFT-EVENT-3015	INFO	我們已排程叢集 [cluster name] 在 [start time in UTC] 與 [end time in UTC] 之間重新啟動以進行維護。您的叢集在此期間將無法存取。我們建議您規劃中斷，以避免發生任何阻礙。
監控	REDSHIFT-EVENT-3016	INFO	叢集將重新啟動。
監控	REDSHIFT-EVENT-3500	ERROR	您的 Amazon Redshift 叢集 [叢集名稱] 調整大小失敗。幾分鐘後會自動重試調整規模。
監控	REDSHIFT-EVENT-3501	ERROR	從快照 [快照名稱] 建立 Amazon Redshift 叢集 [叢集名稱] 的還原操作於 [時間] 執行失敗。請重試操作。
監控	REDSHIFT-EVENT-3504	ERROR	Amazon S3 儲存貯體 [儲存貯體名稱] 無法用於叢集 [叢集名稱] 的記錄。
監控	REDSHIFT-EVENT-3505	ERROR	Amazon S3 儲存貯體 [儲存貯體名稱] 未擁有叢集 [叢集名稱] 的正確 IAM 政策。
監控	REDSHIFT-EVENT-3506	ERROR	Amazon S3 儲存貯體 [儲存貯體名稱] 不存在。叢集 [叢集名稱] 無法繼續記錄。
監控	REDSHIFT-EVENT-3507	ERROR	Amazon Redshift 叢集 [叢集名稱] 無法用 EIP [IP 地址] 建立。該 EIP 已被使用。
監控	REDSHIFT-EVENT-3508	ERROR	Amazon Redshift 叢集 [叢集名稱] 無法用 EIP [IP 地址] 建立。找不到該 EIP。
監控	REDSHIFT-EVENT-3509	ERROR	跨區域快照複本未針對叢集 [叢集名稱] 啟用。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
監控	REDSHIFT-EVENT-3510	ERROR	Amazon Redshift 叢集 [叢集名稱] 於 [時間] 啟動資料表還原失敗。原因：[原因]。
監控	REDSHIFT-EVENT-3511	ERROR	Amazon Redshift 叢集 [叢集名稱] 於 [時間] 執行資料表還原失敗。
監控	REDSHIFT-EVENT-3512	ERROR	Amazon Redshift 叢集 [叢集名稱] 因硬體問題而失敗。正自動從 [時間] 建立的最新快照 [快照名稱] 還原叢集。
監控	REDSHIFT-EVENT-3513	ERROR	Amazon Redshift 叢集 [叢集名稱] 因硬體問題而失敗。正自動從 [時間] 建立的最新快照 [快照名稱] 還原叢集。在此時間後進行的任何資料庫變更將需要重新提交。
監控	REDSHIFT-EVENT-3514	ERROR	Amazon Redshift 叢集 [叢集名稱] 因硬體問題而失敗。叢集目前為硬體故障狀態。請刪除叢集，並從 [時間] 建立的最新快照 [快照名稱] 還原。
監控	REDSHIFT-EVENT-3515	ERROR	Amazon Redshift 叢集 [叢集名稱] 因硬體問題而失敗。叢集目前為硬體故障狀態。請刪除叢集，並從 [時間] 建立的最新快照 [快照名稱] 還原。在此時間後進行的任何資料庫變更將需要重新提交。
監控	REDSHIFT-EVENT-3516	ERROR	Amazon Redshift 叢集 [叢集名稱] 因硬體問題而故障，且該叢集沒有備份。叢集目前為硬體故障狀態，可以刪除。
監控	REDSHIFT-EVENT-3519	INFO	叢集 [叢集名稱] 於 [時間] 開始重新啟動。
監控	REDSHIFT-EVENT-3520	INFO	叢集 [叢集名稱] 於 [時間] 完成重新啟動。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
監控	REDSHIFT-EVENT-3521	INFO	我們在叢集「[叢集名稱]」上偵測到連線問題。已於 [時間] 啟動自動診斷檢查。
監控	REDSHIFT-EVENT-3522	INFO	「[叢集名稱]」叢集的復原動作於 [時間] 失敗。Amazon Redshift 團隊正在尋找解決方案。
監控	REDSHIFT-EVENT-3533	ERROR	'[叢集名稱]' 上的叢集調整大小已在 [時間] 取消。已取消操作，因為 [原因]。[需要的動作]。
監控	REDSHIFT-EVENT-3534	INFO	Amazon Redshift 叢集「[叢集名稱]」的彈性調整大小在 [時間] 完成。當我們傳輸資料時，叢集現在可用於讀取和寫入操作。完成資料傳輸前，部分查詢可能需要較長的時間才會完成。
監控	REDSHIFT-EVENT-3537	INFO	叢集「[叢集名稱]」資料傳輸已於 [UTC 時間] 完成。
監控	REDSHIFT-EVENT-3600	INFO	對 Amazon RedShift 叢集 '[叢集名稱]' 要求的調整大小操作已在過去取消。轉返在 [時間] 完成。
待定	REDSHIFT-EVENT-3601	INFO	叢集 [cluster name] 上的節點將在 [start time] 和 [end time] 之間進行取代。您無法延遲此維護。請妥善規劃。
待定	REDSHIFT-EVENT-3602	INFO	已排程在 [start time] 和 [end time] 之間取代叢集 [cluster name] 上的節點。將無法存取您的叢集。請妥善規劃。
管理	REDSHIFT-EVENT-3603	INFO	從快照 [快照名稱] 建立叢集 [叢集名稱] 的還原操作因內部錯誤而失敗。叢集目前不相容還原狀態，因此可以刪除。請嘗試以不同組態，將快照還原成叢集。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
管理	REDSHIFT-EVENT-3614	INFO	排程動作 [已排程的動作名稱] 已在 [UTC 時間] 建立。第一次呼叫已排程在 [UTC 時間] 進行。
管理	REDSHIFT-EVENT-3615	INFO	排程動作 [已排程的動作名稱] 已排程在 [UTC 時間] 進行。
監控	REDSHIFT-EVENT-3616	INFO	排程動作 [已排程的動作名稱] 已在 [UTC 時間] 完成，其狀態為 'SUCCEEDED' (成功)。
監控	REDSHIFT-EVENT-3617	ERROR	排程動作 [已排程的動作名稱] 已在 [UTC 時間] 因延遲而跳過。
監控	REDSHIFT-EVENT-3618	INFO	叢集 [叢集名稱] 暫停操作已在 [UTC 時間] 開始。已開始暫停
監控	REDSHIFT-EVENT-3619	INFO	Amazon Redshift 叢集 [叢集名稱] 已在 [UTC 時間] 成功暫停。
管理	REDSHIFT-EVENT-3626	INFO	排程動作 [已排程的動作名稱] 已在 [UTC 時間] 修改。第一次呼叫已排程在 [UTC 時間] 進行。
管理	REDSHIFT-EVENT-3627	INFO	排程動作 [已排程的動作名稱] 已在 [UTC 時間] 遭到刪除。
監控	REDSHIFT-EVENT-3628	ERROR	排程動作 [已排程的動作名稱] 已在 [UTC 時間] 完成，其狀態為 'FAILED' (失敗)。
監控	REDSHIFT-EVENT-3629	INFO	已初始化叢集 [cluster name] 的內部容錯移轉請求。
監控	REDSHIFT-EVENT-3630	INFO	Amazon Redshift 已成功將 Amazon Redshift 叢集 [cluster name] 從 [availability-zone] 重新放置到 [availability-zone] 以進行復原。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
管理	REDSHIFT-EVENT-3631	INFO	Amazon Redshift [cluster name] 已收到您的重新放置請求。當可用區域重新定位完成時，Amazon Redshift 會傳送事件通知。
監控	REDSHIFT-EVENT-3632	INFO	Amazon Redshift 叢集 [cluster name] 已成功從 [availability-zone] 重新放置到 [availability-zone]。您現在可以使用該叢集了。
監控	REDSHIFT-EVENT-3658	ERROR	Redshift 叢集 [叢集 ID] 的 EC2-Classic 至 EC2-VPC 遷移失敗。
監控	REDSHIFT-EVENT-3659	INFO	Redshift 叢集 [叢集 ID] 的 EC2-Classic 至 EC2-VPC 遷移成功。
監控	REDSHIFT-EVENT-3660	INFO	叢集目前為硬體故障狀態。請刪除 EC2-Classic 叢集，並從 [UTC 時間] 建立的最新快照 [快照名稱] 還原至 EC2-VPC 叢集。
管理	REDSHIFT-EVENT-3666	INFO	Amazon Redshift 多可用區域叢集 [cluster name] 在 [time in UTC] 偵測到失敗並觸發自動復原。
管理	REDSHIFT-EVENT-3667	INFO	Amazon Redshift 多可用區域叢集 [cluster name] 已於 [time in UTC] 成功復原，並且可在 [first availability zone] 中使用。另一個可用區域中的次要運算很快就可使用。
監控	REDSHIFT-EVENT-3668	ERROR	Amazon Redshift 多可用區域叢集 [cluster name] 無法在 [time in UTC] 復原。
管理	REDSHIFT-EVENT-3669	INFO	Amazon Redshift 多可用區域叢集 [cluster name] 已於 [time in UTC] 成功復原，並且可搭配 [first availability zone] 和 [second availability-zone] 的運算資源使用。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
管理	REDSHIFT-EVENT-3670	INFO	Amazon Redshift 叢集 [cluster name] 的維護已於 [time in UTC] 完成，並且可搭配 [first availability zone] 中的運算資源使用。另一個可用區域中的次要運算很快就可使用。
管理	REDSHIFT-EVENT-3671	INFO	Amazon Redshift 叢集 [cluster name] 上的調整大小已於 [time in UTC] 完成，並且可在 [first availability zone] 中使用。另一個可用區域中的次要運算很快就可使用。
管理	REDSHIFT-EVENT-3672	INFO	Amazon Redshift 多可用區域叢集 [cluster name] 於 [time in UTC] 在 [second availability-zone] 中偵測到失敗並觸發自動復原。
管理	REDSHIFT-EVENT-3673	INFO	為 Amazon Redshift 叢集 [cluster name] 啟用多可用區域的操作已在 [time in UTC] 開始。
管理	REDSHIFT-EVENT-3674	INFO	為 Amazon Redshift 叢集 [cluster name] 啟用多可用區域的操作已在 [time in UTC] 成功完成。
監控	REDSHIFT-EVENT-3675	ERROR	為 Amazon Redshift 叢集 [cluster name] 啟用多可用區域的操作已在 [time in UTC] 失敗。
管理	REDSHIFT-EVENT-3676	INFO	停用 Amazon Redshift 多可用區域叢集 [cluster name] 之多可用區域的操作已在 [time in UTC] 開始。
管理	REDSHIFT-EVENT-3677	INFO	停用 Amazon Redshift 叢集 [cluster name] 之多可用區域的操作已在 [time in UTC] 成功完成。
監控	REDSHIFT-EVENT-3678	ERROR	停用 Amazon Redshift 叢集 [cluster name] 之多可用區域的操作已在 [time in UTC] 失敗。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
Configuration	REDSHIFT-EVENT-3679	INFO	已成功修改 Amazon Redshift 叢集 [cluster name] 的連接埠。
Configuration	REDSHIFT-EVENT-3680	ERROR	Amazon Redshift 無法建立叢集 [叢集名稱]，因為無法存取此操作所需的服務連結角色 (SLR)。請嘗試從 Amazon Redshift 主控台再次建立。Amazon Redshift 會自動建立 SLR。
監控	REDSHIFT-EVENT-3684	ERROR	您的 Amazon S3 儲存貯體 【儲存貯體名稱】 已使用未知或無法存取的 AWS KMS 金鑰加密。請修改您 Amazon S3 儲存貯體的加密。
管理	REDSHIFT-EVENT-3685	ERROR	叢集 [cluster name] 上的還原操作失敗，因為沒有足夠的可用磁碟空間。正在恢復操作。請嘗試還原至使用不同組態的叢集。
管理	REDSHIFT-EVENT-3686	ERROR	叢集 [cluster name] 上的調整大小操作失敗，因為沒有足夠的可用磁碟空間。正在恢復操作。請嘗試調整為使用不同組態的叢集。
管理	REDSHIFT-EVENT-3687	INFO	針對 Amazon Redshift 叢集 [cluster name] 請求的調整大小操作已在 [time in UTC] 完成。
安全	REDSHIFT-EVENT-3688	ERROR	Amazon Redshift 叢集 [cluster name] 的加密金鑰輪換未能完成。
資料共用	REDSHIFT-EVENT-3689	INFO	將 Amazon Redshift 命名空間 <cluster name> 註冊至 Glue Data Catalog 帳戶 <account id> 的操作已在 <time in UTC> 開始。
資料共用	REDSHIFT-EVENT-3690	INFO	將 Amazon Redshift 命名空間 <cluster name> 註冊至 Glue Data Catalog 帳戶 <account id> 的操作已在 <time in UTC> 成功完成。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
資料共用	REDSHIFT-EVENT-3691	INFO	將 Amazon Redshift 命名空間 <cluster name> 註冊至 Glue Data Catalog 帳戶 <account id> 的操作已在 <time in UTC> 失敗。
資料共用	REDSHIFT-EVENT-3692	INFO	將 Amazon Redshift 命名空間 <cluster name> 從 Glue Data Catalog 帳戶 <account id> 取消註冊的操作已在 <time in UTC> 開始。
資料共用	REDSHIFT-EVENT-3693	INFO	將 Amazon Redshift 命名空間 <cluster name> 從 Glue Data Catalog 帳戶 <account id> 取消註冊的操作已在 <time in UTC> 成功完成。
資料共用	REDSHIFT-EVENT-3694	INFO	將 Amazon Redshift 命名空間 <cluster name> 從 Glue Data Catalog 帳戶 <account id> 取消註冊的操作已在 <time in UTC> 失敗。
安全	REDSHIFT-EVENT-4000	INFO	您 Amazon Redshift 叢集 [叢集名稱] 的管理員憑證已於 [時間] 更新。
安全	REDSHIFT-EVENT-4001	INFO	安全群組 [安全群組名稱] 於 [時間] 修改。變更將自動套用到所有相關聯的叢集。
安全	REDSHIFT-EVENT-4005	ERROR	叢集 【叢集名稱】 具有無效的 AWS KMS key 許可。您的叢集將進入 "INACCESSIBLE_KMS_KEY" 狀態，且您將無法再存取它。
安全	REDSHIFT-EVENT-4006	INFO	叢集 【叢集名稱】 現在已還原至其先前的狀態，因為已還原 AWS KMS key 許可。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
安全	REDSHIFT-EVENT-4500	ERROR	找不到您為 Amazon Redshift 叢集 [cluster name] 提供的一或多個安全群組。如果有任何有效的安全群組，這些群組會套用到叢集。未套用任何無效的安全群組。如果您要修改現有叢集，但沒有任何有效的安全群組，則叢集會保留其原始安全群組。如果您要建立新叢集，但沒有任何有效的安全群組，則叢集建立會失敗。在任何一種情況下，請使用有效的安全群組重試請求。如需在 Amazon Redshift 中管理安全群組的詳細資訊，請前往《Amazon Redshift 管理指南》中的 VPC security groups (VPC 安全群組)。
安全	REDSHIFT-EVENT-4501	ERROR	找不到叢集安全群組 [叢集安全群組名稱] 中指定的 安全群組 [安全群組名稱]。無法完成授權。
安全	REDSHIFT-EVENT-4502	ERROR	由於並行活動，Amazon Redshift 叢集 [叢集名稱] 的管理員憑證無法在 [時間] 更新。請允許目前的工作負載以完成或減少作用中的工作負載，然後重試該操作。
安全	REDSHIFT-EVENT-4503	ERROR	Amazon Redshift 無法存取叢集的秘密 [叢集名稱]。
安全	REDSHIFT-EVENT-4504	ERROR	Amazon Redshift 無法存取用來加密叢集 [叢集名稱] 的管理員登入資料密碼的 KMS 金鑰 [KMS 金鑰]。
安全	REDSHIFT-EVENT-4505	ERROR	Amazon Redshift 無法輪換叢集 [叢集名稱] 的密碼，因為叢集上有正在進行的作業。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
安全	REDSHIFT-EVENT-4506	ERROR	您的 Amazon Redshift 叢集 [叢集名稱] 已暫停。Amazon Redshift 無法輪換已暫停叢集的秘密。

參數群組來源類型的類別和事件

下表顯示當參數群組為來源類型時的事件類別和事件清單。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
Configuration	REDSHIFT-EVENT-1002	INFO	參數 [參數名稱] 於 [時間] 從 [數值] 更新為 [數值]。
Configuration	REDSHIFT-EVENT-1003	INFO	已建立 [群組名稱] 叢集參數群組。
Configuration	REDSHIFT-EVENT-1004	INFO	已刪除 [群組名稱] 叢集參數群組。
Configuration	REDSHIFT-EVENT-1005	INFO	叢集參數群組 [名稱] 於 [時間] 更新。如果您只變更動態參數，則現在正在修改相關聯的叢集。如果您變更靜態參數，則會在您重新啟動相關聯的叢集時套用所有更新，包括動態參數。

安全群組來源類型的類別和事件

下表顯示當安全群組為來源類型時的事件類別和事件清單。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
安全	REDSHIFT-EVENT-4002	INFO	已建立 [群組名稱] 叢集安全群組。
安全	REDSHIFT-EVENT-4003	INFO	已刪除 [群組名稱] 叢集安全群組。
安全	REDSHIFT-EVENT-4004	INFO	叢集安全群組 [群組名稱] 於 [時間] 變更。變更將自動套用到所有相關聯的叢集。

快照來源類型的類別和事件

下表顯示當快照為來源類型時的事件類別和事件清單。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
管理	REDSHIFT-EVENT-2009	INFO	Amazon Redshift 叢集 [叢集名稱] 的使用者快照 [快照名稱] 於 [時間] 開始。若要監控快照流程，請造訪 AWS 管理主控台。
管理	REDSHIFT-EVENT-2010	INFO	Amazon Redshift 叢集 [叢集名稱] 的使用者快照 [快照名稱] 於 [時間] 取消。
管理	REDSHIFT-EVENT-2011	INFO	Amazon Redshift 叢集 [叢集名稱] 的使用者快照 [快照名稱] 於 [時間] 刪除。
管理	REDSHIFT-EVENT-2012	INFO	Amazon Redshift 叢集 [叢集名稱] 最後的快照 [快照名稱] 於 [時間] 開始。
監控	REDSHIFT-EVENT-3004	INFO	Amazon Redshift 叢集 [叢集名稱] 的使用者快照 [快照名稱] 於 [時間] 成功完成。
監控	REDSHIFT-EVENT-3005	INFO	Amazon Redshift 叢集 [叢集名稱] 最後的快照 [快照名稱] 於 [時間] 成功完成。

Amazon Redshift 類別	事件 ID	事件嚴重性	說明
監控	REDSHIFT-EVENT-3006	INFO	Amazon Redshift 叢集 [叢集名稱] 最後的快照 [快照名稱] 於 [時間] 取消。
監控	REDSHIFT-EVENT-3502	ERROR	Amazon Redshift 叢集 [叢集名稱] 最後的快照 [快照名稱] 於 [時間] 失敗。團隊正在調查問題。請造訪 AWS 管理主控台 以重試操作。
監控	REDSHIFT-EVENT-3503	ERROR	Amazon Redshift 叢集 [叢集名稱] 的使用者快照 [快照名稱] 於 [時間] 失敗。團隊正在調查問題。請造訪 AWS 管理主控台 以重試 操作。

使用 Amazon EventBridge 的 Amazon Redshift Serverless 事件通知

Amazon Redshift Serverless 會使用 Amazon EventBridge 來管理事件通知，讓您隨時掌握關於資料倉儲變化的最新資訊。Amazon EventBridge 是無伺服器事件匯流排服務，可讓您用於將應用程式與來自各種來源的資料互相連線。在這種情況下，事件來源是 Amazon Redshift。事件 (環境中的受監控變更) 會自動從 Amazon Redshift 資料倉儲傳送至 EventBridge。系統會以接近即時的速度傳送事件。

EventBridge 的功能包括提供環境讓您編寫事件規則，它可以指定要針對特定事件採取的動作。您也可以設定目標，這些目標是 EventBridge 可以向其傳送事件的資源。目標可以包含 API 目的地、Amazon CloudWatch 日誌群組和其他項目。如需規則的相關資訊，請參閱 Amazon EventBridge 規則。如需目標的相關資訊，請參閱 Amazon EventBridge 目標。

事件可以分為嚴重性和類別兩種。可用的篩選條件如下：

- 資源篩選 — 根據事件所關聯的資源來接收訊息。資源包括工作群組和快照等項目。
- 時段篩選 — 將範圍限定在特定時段內的事件。
- 類別篩選 — 接收指定類別中所有事件的事件通知。

下表包含 Amazon Redshift Serverless 事件，以及其他中繼資料：

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
RateChange	REDSHIFT-SERVERLESS-EVENT-1001	INFO	工作群組基礎 RPU 變更已於 <UTC 時間> 成功完成。
RateChange	REDSHIFT-SERVERLESS-EVENT-1002	ERROR	工作群組基礎 RPU 變更無法於 <UTC 時間> 完成。
監控	REDSHIFT-SERVERLESS-EVENT-1003	INFO	Amazon Redshift 資料倉儲 <端點名稱> 上的軟體已於 <UTC 時間> 更新。
Configuration	REDSHIFT-SERVERLESS-EVENT-1011	ERROR	Amazon Redshift Serverless 無法建立工作群組 [工作群組名稱]，因為無法存取此操作所需的服務連結角色 (SLR)。請嘗試在 Amazon Redshift 主控台上再次建立。Amazon Redshift 會自動建立 SLR。
監控	REDSHIFT-SERVERLESS-EVENT-1029	ERROR	工作群組基本 RPU 變更無法在 [time in UTC] 完成，因為沒有足夠的可用磁碟空間。請使用不同的組態再試一次。
監控	REDSHIFT-SERVERLESS-EVENT-1500	ERROR	無法建立或更新工作群組 <工作群組名稱>，因為您已超過帳戶

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
			的彈性 IP 地址限制。使用 Amazon EC2 刪除未使用的彈性 IP 地址或請求提高限制。
監控	REDSHIFT-SERVERLESS-EVENT-1501	ERROR	子網路 <子網路 ID> 沒有可用的 IP 地址。這會讓下列查詢類型無法在工作群組 <工作群組名稱> 上成功執行：EMR、聯合查詢、從 Amazon EC2 進行的 COPY/UNLOAD。若要修正此問題，請刪除 ENI 以在子網路中釋放 IP。
監控	REDSHIFT-SERVERLESS-EVENT-1502	ERROR	子網路 <子網路 ID> 沒有可用的 IP 地址。這會讓 Amazon EMR、Redshift 聯合查詢、Redshift COPY/UNLOAD、Redshift ML 查詢類型無法在工作群組 <工作群組名稱> 上成功執行。若要修正此問題，請刪除未使用的彈性網路界面 (ENI) 以在子網路中釋放 IP。

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
管理	REDSHIFT-SERVERLESS-EVENT-1008	INFO	您的 Amazon Redshift 工作群組 <工作群組名稱> 已建立並可供使用。
管理	REDSHIFT-SERVERLESS-EVENT-1009	INFO	您的 Amazon Redshift 工作群組 <工作群組名稱> 已於 <UTC 時間> 刪除。
監控	REDSHIFT-SERVERLESS-EVENT-1000	INFO	快照 <快照名稱> 已於 <UTC 時間> 成功完成。
管理	REDSHIFT-SERVERLESS-EVENT-1004	INFO	已於 <UTC 時間> 成功完成從命名空間 <命名空間名稱> 上的快照還原。
管理	REDSHIFT-SERVERLESS-EVENT-1005	ERROR	從命名空間 <命名空間名稱> 上的快照還原的操作已於 <UTC 時間> 失敗。
管理	REDSHIFT-SERVERLESS-EVENT-1006	INFO	已於 <UTC 時間> 成功完成從命名空間 <命名空間名稱> 上的復原點還原。
管理	REDSHIFT-SERVERLESS-EVENT-1007	INFO	從命名空間 <命名空間名稱> 上的復原點還原的操作已於 <UTC 時間> 失敗。

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
安全	REDSHIFT-SERVERLESS-EVENT-1012	ERROR	<namespace name>Amazon Redshift 無法存取您的命名空間 <namespace name> 的秘密。
安全	REDSHIFT-SERVERLESS-EVENT-1013	ERROR	Amazon Redshift 無法存取用來加密命名空間<namespace name>之管理員登入資料秘密的 KMS 金鑰。
安全	REDSHIFT-SERVERLESS-EVENT-1014	ERROR	Amazon Redshift 無法輪換命名空間<namespace name>的秘密，因為工作群組正在進行操作。
安全	REDSHIFT-SERVERLESS-EVENT-1015	ERROR	您的命名空間<namespace name>沒有附加工作群組。Amazon Redshift 只能輪換已連接工作群組之命名空間的秘密。
安全	REDSHIFT-SERVERLESS-EVENT-1016	INFO	針對您的命名空間<namespace name>在<time in UTC>更新的管理員憑證。

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
安全	REDSHIFT-SERVERLESS-EVENT-1030	INFO	將 Amazon Redshift 命名空間 <namespace name> 註冊至 Glue Data Catalog 帳戶 <account id> 的操作已在 <time in UTC> 開始。
安全	REDSHIFT-SERVERLESS-EVENT-1031	INFO	將 Amazon Redshift 命名空間 <namespace name> 註冊至 Glue Data Catalog 帳戶 <account id> 的操作已在 <time in UTC> 成功完成。
安全	REDSHIFT-SERVERLESS-EVENT-1032	INFO	將 Amazon Redshift 命名空間 <namespace name> 註冊至 Glue Data Catalog 帳戶 <account id> 的操作已在 <time in UTC> 失敗。
安全	REDSHIFT-SERVERLESS-EVENT-1033	INFO	將 Amazon Redshift 命名空間 <namespace name> 從 Glue Data Catalog 帳戶 <account id> 取消註冊的操作已在 <time in UTC> 開始。

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
安全	REDSHIFT-SERVERLESS-EVENT-1034	INFO	將 Amazon Redshift 命名空間 <namespace name> 從 Glue Data Catalog 帳戶 <account id> 取消註冊的操作已在 <time in UTC> 成功完成。
安全	REDSHIFT-SERVERLESS-EVENT-1035	INFO	將 Amazon Redshift 命名空間 <namespace name> 從 Glue Data Catalog 帳戶 <account id> 取消註冊的操作已在 <time in UTC> 失敗。
管理	REDSHIFT-SERVERLESS-EVENT-1036	ERROR	Redshift Serverless 工作群組 <workgroup name> 上客戶啟動的追蹤更新失敗。將工作群組恢復其原始追蹤。

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
管理	REDSHIFT-SERVERLESS-EVENT-1037	ERROR	Redshift Serverless 工作群組 <workgroup name> 上的追蹤更新失敗。Amazon Redshift Serverless 工作群組的追蹤更新失敗，因為工作群組當時忙碌中。因此，工作群組已恢復其原始追蹤。若要重試追蹤更新，請等待工作群組上活動較少的時間，然後再次嘗試更新追蹤。
管理	REDSHIFT-SERVERLESS-EVENT-1038	INFO	您的 Amazon Redshift 工作群組 <workgroup name> 追蹤已修改。追蹤變更已完成。
命名空間	REDSHIFT-SERVERLESS-EVENT-1039	ERROR	命名空間 <命名空間名稱> 具有無效的 AWS KMS key 許可。您的命名空間將進入 "INACCESSIBLE_KMS_KEY" 狀態，且您將無法再存取它。

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
命名空間	REDSHIFT-SERVERLESS-EVENT-1040	INFO	命名空間 <命名空間名稱> 現在已還原至其先前的狀態，因為已還原 AWS KMS key 許可。
命名空間	REDSHIFT-SERVERLESS-EVENT-1045	INFO	Redshift Serverless 命名空間 <namespace name> 上的資料共用 <datashare name> 因無伺服器還原而遭到捨棄。
命名空間	REDSHIFT-SERVERLESS-EVENT-1046	INFO	由於無伺服器還原，已撤銷 Redshift Serverless 命名空間 <namespace name> 上資料共用 <datashare name> 的消費者 <消費者原則 id>。
命名空間	REDSHIFT-SERVERLESS-EVENT-1047	INFO	由於無伺服器還原，Redshift Serverless 命名空間 <namespace name> 上資料共用 <datashare name> 的公有可存取性已變更。

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
安全	REDSHIFT-SERVERLESS-EVENT-1048	ERROR	由於內部問題，您請求退出命名空間 <namespace name> 的 Redshift-Secrets Manager 整合失敗。手動重試選擇退出操作以解決此問題。
安全	REDSHIFT-SERVERLESS-EVENT-1049	ERROR	由於內部問題，您無法成功完成選擇加入命名空間 <namespace name> Redshift-Secrets Manager 整合的請求。手動輪換與 Redshift Serverless 命名空間相關聯的秘密，以解決此問題。
安全	REDSHIFT-SERVERLESS-EVENT-1050	ERROR	由於內部問題，您在命名空間 <namespace name> 上重設管理員登入資料的請求失敗。重試 操作以重設管理員使用者登入資料。

使用 Amazon EventBridge 的零 ETL 整合事件通知

零 ETL 整合使用 Amazon EventBridge 來管理事件通知，讓您隨時掌握整合變更的最新資訊。Amazon EventBridge 是無伺服器事件匯流排服務，可讓您用於將應用程式與來自各種來源的資料互相連線。在

這種情況下，事件來源是 Amazon Redshift。事件 (環境中的受監控變更) 會自動從 Amazon Redshift 資料倉儲傳送至 EventBridge。系統會以接近即時的速度傳送事件。

EventBridge 會提供環境讓您編寫事件規則，它可以指定要針對特定事件採取的動作。您也可以設定目標，這些目標是 EventBridge 可以向其傳送事件的資源。目標可以包含 API 目的地、Amazon CloudWatch 日誌群組和其他項目。如需規則的相關資訊，請參閱 Amazon EventBridge 規則。如需目標的相關資訊，請參閱 Amazon EventBridge 目標。

事件可以分為嚴重性和類別兩種。可用的篩選條件如下：

- 資源篩選 — 根據事件所關聯的資源來接收訊息。資源包括工作群組或快照。
- 時段篩選 — 將範圍限定在特定時段內的事件。
- 類別篩選 — 接收指定類別中所有事件的事件通知。

下表包含零 ETL 整合事件，以及其他中繼資料：

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
監控	REDSHIFT-INTEGRATION-EVENT-0000	INFO	您的零 ETL 整合 <整合名稱> 已建立，並在 UTC> 的 <時間變為 ACTIVE。
監控	REDSHIFT-INTEGRATION-EVENT-0001	INFO	您的零 ETL 整合 <整合名稱> 已在 UTC> 的 <時間刪除。
監控	REDSHIFT-INTEGRATION-EVENT-0002	INFO	您的零 ETL 整合 <整合名稱> 已在 UTC> 的 <時間啟動刪除。
監控	REDSHIFT-INTEGRATION-EVENT-0003	INFO	您的零 ETL 整合 <整合名稱> 正在將交易資料同步至 Amazon Redshift 資料倉儲。

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
監控	REDSHIFT-INTEGRATION-EVENT-0004	WARNING	Amazon Redshift 無法複寫資料表，因為資料表缺少主索引鍵。將主索引鍵（主索引鍵）新增至來源資料表，資料表會自動重新同步。
監控	REDSHIFT-INTEGRATION-EVENT-0005	WARNING	Amazon Redshift 無法複寫資料表，因為一或多個資料欄使用不支援的資料類型。修改您的整合以在篩選條件中排除此資料表，或從來源資料表捨棄資料欄，並執行 'ALTER DATABASE <Redshift 資料庫名稱> INTEGRATION REFRESH TABLE <結構描述名稱>.<table name>' 以同步此資料表。
監控	REDSHIFT-INTEGRATION-EVENT-0006	ERROR	無法建立整合。請刪除並重新建立整合。如果錯誤仍然存在，請聯絡 AWS Support。

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
監控	REDSHIFT-INTEGRATION-EVENT-0007	ERROR	由於內部錯誤，無法載入資料。請刪除並重新建立整合。如果錯誤仍然存在，請聯絡 AWS Support。
監控	REDSHIFT-INTEGRATION-EVENT-0008	ERROR	授權失敗，因為已從來源資料庫叢集撤銷許可。如果您使用客戶受管金鑰 (CMK) 來加密整合，請檢查 KMS 金鑰許可。然後，刪除並重新建立整合。
監控	REDSHIFT-INTEGRATION-EVENT-0009	ERROR	無法將資料傳送至 Amazon Redshift，因為資料表和結構描述的數目超過 Amazon Redshift 的限制。請刪除並重新建立整合。
監控	REDSHIFT-INTEGRATION-EVENT-0012	ERROR	已在目的地無伺服器命名空間上叫用從復原點還原。請刪除並重新建立整合。
監控	REDSHIFT-INTEGRATION-EVENT-0013	INFO	您的零 ETL 整合 < 整合名稱> 現在為 ACTIVE。

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
監控	REDSHIFT-INTEGRATION-EVENT-0014	ERROR	您的整合 <integration name> 處於 FAILED 狀態，因為由於內部錯誤，我們無法對其進行修改。請刪除並重新建立整合。如果錯誤仍然存在，請聯絡 AWS Support。
作業	REDSHIFT-INTEGRATION-EVENT-0016	INFO	您的零 ETL 整合 <integration name> 正在處理修改請求。
作業	REDSHIFT-INTEGRATION-EVENT-0017	INFO	已套用您對零 ETL 整合 <整合名稱> 的修改。
作業	REDSHIFT-INTEGRATION-EVENT-0018	WARNING	目標 Amazon Redshift 叢集正在暫停。等待叢集暫停，然後繼續叢集以繼續串流資料。
作業	REDSHIFT-INTEGRATION-EVENT-0019	WARNING	目標 Amazon Redshift 叢集已暫停。必須繼續叢集，才能繼續串流資料。
作業	REDSHIFT-INTEGRATION-EVENT-0020	WARNING	目標 Amazon Redshift 叢集正在繼續。等待叢集處於作用中狀態，以繼續串流資料。

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
監控	REDSHIFT-INTEGRATION-EVENT-0025	ERROR	在目的地無伺服器命名空間上調用無伺服器還原，並且您的整合已停用，因為它在還原的快照中不可用。還原可用整合時建立的快照，或刪除整合。
組態	REDSHIFT-INTEGRATION-EVENT-1000	ERROR	來源 Aurora 資料庫叢集上的一或多個參數設定錯誤。請修正參數群組，並重新啟動叢集以套用變更，然後重新建立整合。
組態	REDSHIFT-INTEGRATION-EVENT-1001	ERROR	整合失敗，因為 enable_case_sensitive_identifier 參數的值不正確。將來源 Aurora 資料庫叢集的值設為 true，然後刪除並重新建立整合。
組態	REDSHIFT-INTEGRATION-EVENT-1002	ERROR	整合失敗，因為 cdc_insert_enabled 參數的值不正確。將來源 Aurora 資料庫叢集的值設為 true，然後刪除並重新建立整合。

Amazon Redshift 類別	外部事件 ID	事件嚴重性	訊息描述
組態	REDSHIFT-INTEGRATION-EVENT-1003	ERROR	來源資料庫叢集參數群組中的 binlog_format 參數必須設定為 ROW。請修正參數群組，並重新啟動叢集以套用變更，然後重新建立整合。
組態	REDSHIFT-INTEGRATION-EVENT-1004	ERROR	無法載入資料，因為已啟用 binlog_transaction_compression 叢集參數。將參數值設定為 OFF，並重新啟動寫入器執行個體以套用變更，然後重新建立整合。
組態	REDSHIFT-INTEGRATION-EVENT-1005	ERROR	無法載入資料，因為 binlog_row_value_options 叢集參數設定為不支援的 PARTIAL_JSON。請修正參數群組，並重新啟動寫入器執行個體以套用變更，然後重新建立整合。
組態	REDSHIFT-INTEGRATION-EVENT-1006	WARNING	無法剖析整合篩選條件。請修正篩選條件語法。

Amazon Redshift 中的配額和限制

Amazon Redshift 具有配額，可限制每個 AWS 區域 AWS 在帳戶中使用數個資源。每個配額都有預設值，而且有些配額可以調整。

Amazon Redshift 物件的配額

Amazon Redshift 的配額會限制數種物件類型的使用。每個類型都有預設值。

配額名稱	AWS 預設值	可調整	說明
AWS 您可以授權每個快照還原快照的 帳戶	20	否	每個快照您可以授權還原快照 AWS 的帳戶數目上限。
AWS 您可以授權 還原每個快照的 帳戶 AWS KMS key	100	否	您可以授權每個 KMS 金鑰還原快照 AWS 的帳戶數目上限。換言之，如果您有 10 個以單一 KMS 金鑰加密的快照，那麼您可以授權 10 個 AWS 帳戶還原每個快照，或者是合計最多 100 個帳戶、每個快照不超過 20 個帳戶的其他組合。
Amazon Redshift 存取其他服務的叢集 IAM 角色 AWS	50[1]	否	您可以與叢集建立關聯的 IAM 角色數量上限，以授權 Amazon Redshift AWS 為擁有叢集和 IAM 角色的使用者存取其他服務。 [1]以下配額為 10 AWS 區域：us-iso-east-1、us-iso-west-1、us-isob-east-1。
所有使用者定義手動 WLM 佇列的並行層級 (查詢槽)	50	否	手動工作負載管理定義之所有使用者定義佇列的查詢槽上限。

配額名稱	AWS 預設值	可調整	說明
並行擴展叢集	10	是	並行擴展叢集的數目上限。
叢集中的 DC2 節點	128	是	您可以配置給叢集的 DC2 節點數目上限。如需每個節點類型之節點限制的相關資訊，請參閱 Amazon Redshift 中的叢集和節點。
事件訂閱	20	是	目前 AWS 區域中此帳戶的事件訂閱數量上限。
節點	200	是	目前 AWS 區域中此帳戶所有資料庫執行個體的節點數量上限。
參數群組	20	否	目前 AWS 區域中此帳戶的參數群組數目上限。
叢集中的 RG 節點	128	是	您可以配置給叢集的 RG 節點數量上限。如需每個節點類型之節點限制的相關資訊，請參閱 Amazon Redshift 中的叢集和節點。
叢集中的 RA3 節點	128	是	您可以配置給叢集的 RA3 節點數目上限。如需每個節點類型之節點限制的相關資訊，請參閱 Amazon Redshift 中的叢集和節點。
已連線至叢集的 RedShift 受管 VPC 端點	30	是	您可以連線至叢集的 RedShift 受管 VPC 端點數目上限。如需 RedShift 受管 VPC 端點的相關資訊，請參閱Redshift 受管 VPC 端點。
透過 RedShift 受管 VPC 端點存取之叢集的被授予者	10	是	叢集擁有者可授權其為叢集建立 RedShift 受管 VPC 端點的被授予者數目上限。如需 RedShift 受管 VPC 端點的相關資訊，請參閱Redshift 受管 VPC 端點。

配額名稱	AWS 預設值	可調整	說明
每個授權的 RedShift 受管 VPC 端點數目	10	是	每個授權可建立的 RedShift 受管 VPC 端點數目上限。如需 RedShift 受管 VPC 端點的相關資訊，請參閱Redshift 受管 VPC 端點。
預留節點	200	是	目前 AWS 區域中此帳戶的預留節點數目上限。
各叢集中每個資料庫中的結構描述	9,900	否	各叢集中可在每個資料庫中建立的結構描述數目上限。然而，pg_temp_*　結構描述不會計入此配額。
Security groups (安全群組)	20	是	目前 AWS 區域中此帳戶的安全群組數量上限。
使用 COPY 載入時的單一資料列大小	64	否	使用 COPY 命令載入時的單一資料列大小上限 (單位 MB)。這是 COPY 命令可以掃描的單一未剖析記錄的大小上限；資料列中的個別 SUPER 物件獨立限制為 16 MB。
快照	700	是	目前 AWS 區域中此帳戶的使用者快照數目上限。
子網路群組	50	是	目前 AWS 區域中此帳戶的子網路群組數量上限。
子網路群組中的子網路	20	是	子網路群組的子網路數目上限。
具有單一節點rg.large集的叢集節點類型的資料表	9,900	否	具有單節點叢集之 rg.large 叢集節點類型的資料表數目上限。此限制包括永久資料表、暫存資料表、資料共用資料表和具體化視觀表。外部資料表會計為暫存資料表。暫存資料表包括使用者定義的暫存資料表，以及 Amazon Redshift 在查詢處理或系統維護期間建立的暫存資料表。檢視和系統暫存資料表不包含在此限制中。

配額名稱	AWS 預設值	可調整	說明
具有多節點rg.large集的叢集節點類型的資料表	20,000	否	具有多節點叢集之rg.large節點類型的資料表數目上限。此限制包括永久資料表、暫存資料表、資料共用資料表和具體化視觀表。外部資料表會計為暫存資料表。暫存資料表包括使用者定義的暫存資料表，以及 Amazon Redshift 在查詢處理或系統維護期間建立的暫存資料表。檢視和系統暫存資料表不包含在此限制中。
rg.xlarge 叢集節點類型的資料表。	20,000	否	rg.xlarge 叢集節點類型的資料表數目上限。此限制包括永久資料表、暫存資料表、資料共用資料表和具體化視觀表。外部資料表會計為暫存資料表。暫存資料表包括使用者定義的暫存資料表，以及 Amazon Redshift 在查詢處理或系統維護期間建立的暫存資料表。檢視和系統暫存資料表不包含在此限制中。
rg.4xlarge 叢集節點類型的資料表	200,000	否	rg.4xlarge 叢集節點類型的資料表數目上限。此限制包括永久資料表、暫存資料表、資料共用資料表和具體化視觀表。外部資料表會計為暫存資料表。暫存資料表包括使用者定義的暫存資料表，以及 Amazon Redshift 在查詢處理或系統維護期間建立的暫存資料表。檢視和系統暫存資料表不包含在此限制中。
rg.12xlarge 叢集節點類型的資料表	200,000	否	rg.12xlarge 叢集節點類型的資料表數目上限。此限制包括永久資料表、暫存資料表、資料共用資料表和具體化視觀表。外部資料表會計為暫存資料表。暫存資料表包括使用者定義的暫存資料表，以及 Amazon Redshift 在查詢處理或系統維護期間建立的暫存資料表。檢視和系統暫存資料表不包含在此限制中。
ra3.large 叢集節點類型的資料表	9,900	否	ra3.large 叢集節點類型的資料表數目上限。此限制包括永久資料表、暫存資料表、資料共用資料表和具體化視觀表。外部資料表會計為暫存資料表。暫存資料表包括使用者定義的暫存資料表，以及 Amazon Redshift 在查詢處理或系統維護期間建立的暫存資料表。檢視和系統暫存資料表不包含在此限制中。

配額名稱	AWS 預設值	可調整	說明
具有單節點叢集之 ra3.xlplus 叢集節點類型的資料表。	9,900	否	具有單節點叢集之 ra3.xlplus 叢集節點類型的資料表數目上限。此限制包括永久資料表、暫存資料表、資料共用資料表和具體化視觀表。外部資料表會計為暫存資料表。暫存資料表包括使用者定義的暫存資料表，以及 Amazon Redshift 在查詢處理或系統維護期間建立的暫存資料表。檢視和系統暫存資料表不包含在此限制中。
具有多節點叢集之 ra3.xlplus 叢集節點類型的資料表。	20,000	否	具有多節點叢集之 ra3.xlplus 叢集節點類型的資料表數目上限。此限制包括永久資料表、暫存資料表、資料共用資料表和具體化視觀表。外部資料表會計為暫存資料表。暫存資料表包括使用者定義的暫存資料表，以及 Amazon Redshift 在查詢處理或系統維護期間建立的暫存資料表。檢視和系統暫存資料表不包含在此限制中。
ra3.4xlarge 叢集節點類型的資料表	200,000	否	ra3.4xlarge 叢集節點類型的資料表數目上限。此限制包括永久資料表、暫存資料表、資料共用資料表和具體化視觀表。外部資料表會計為暫存資料表。暫存資料表包括使用者定義的暫存資料表，以及 Amazon Redshift 在查詢處理或系統維護期間建立的暫存資料表。檢視和系統暫存資料表不包含在此限制中。
ra3.16xlarge 叢集節點類型的資料表	200,000	否	ra3.16xlarge 叢集節點類型的資料表數目上限。此限制包括永久資料表、暫存資料表、資料共用資料表和具體化視觀表。外部資料表會計為暫存資料表。暫存資料表包括使用者定義的暫存資料表，以及 Amazon Redshift 在查詢處理或系統維護期間建立的暫存資料表。檢視和系統暫存資料表不包含在此限制中。

配額名稱	AWS 預設值	可調整	說明
dc2.large 叢集節點類型的資料表	9,900	否	dc2.large 叢集節點類型的資料表數目上限。此限制包括永久資料表、暫存資料表、資料共用資料表和具體化視觀表。外部資料表會計為暫存資料表。暫存資料表包括使用者定義的暫存資料表，以及 Amazon Redshift 在查詢處理或系統維護期間建立的暫存資料表。檢視和系統暫存資料表不包含在此限制中。
dc2.8xlarge 叢集節點類型的資料表	200,000	否	dc2.8xlarge 叢集節點類型的資料表數目上限。此限制包括永久資料表、暫存資料表、資料共用資料表和具體化視觀表。外部資料表會計為暫存資料表。暫存資料表包括使用者定義的暫存資料表，以及 Amazon Redshift 在查詢處理或系統維護期間建立的暫存資料表。檢視和系統暫存資料表不包含在此限制中。
資料庫數目	60	否	Amazon Redshift 叢集中允許的資料庫計數上限。這會排除從資料共用建立的資料庫。
閒置或非作用中工作階段的逾時	4 小時	否	此設定適用於叢集。如需有關為使用者設定閒置工作階段逾時值的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 ALTER USER。使用者設定的優先順序高於叢集設定。
閒置交易的逾時	6 小時	否	開啟中交易的非作用中期間上限，超過之後，Amazon Redshift 就會結束與交易相關聯的工作階段。此設定的優先順序高於任何使用者定義的閒置逾時設定。其適用於叢集。
資料庫中的預存程序	10,000	否	預存程序的數目上限。若要了解更多限制，請參閱預存程序支援的限制和差異。
RG 節點的連線數目上限	2,000	否	與 RG 叢集的連線數上限。這是適用於所有 RG 節點類型的硬性限制，無法增加。

配額名稱	AWS 預設值	可調整	說明
RA3 節點的連線數上限	2,000	否	RA3 叢集的連線數目上限。這是適用於所有 RA3 節點類型的硬性限制，無法增加。
DC2 節點的連線數上限	各有不同	否	與 dc2.large 叢集的連線數上限為 500。與 dc2.8xlarge 叢集的連線數上限為 2000。
叢集中的 Amazon Redshift 角色數目	1,000	是	您可以為每個叢集建立的 Amazon Redshift 角色數目上限。如需角色型存取控制 (RBAC) 角色的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的角色型存取控制 (RBAC)。

Amazon Redshift Serverless 物件的配額

Amazon Redshift 的配額會限制 Amazon Redshift Serverless 命名空間中數個物件類型的使用。每個類型都有預設值。

配額名稱	AWS 預設值	可調整	說明
資料庫數目	100	否	Amazon Redshift Serverless 命名空間中允許的資料庫計數上限。這會排除從資料共用建立的資料庫。
結構描述數目	9,900	否	Amazon Redshift Serverless 命名空間中允許的結構描述計數上限。
資料表數目	200,000	否	Amazon Redshift Serverless 命名空間中允許的資料表計數上限。
使用 4 個 Redshift 處理單元 (RPU) 之	最多 32 TB	否	具有 4 個 RPU 的資料倉儲支援的最大 RMS 儲存空間。在 32 TB 限制內，Amazon Redshift 會在內部使用數百 MB 的儲存空間。

配額名稱	AWS 預設值	可調整	說明
資料倉儲的 Redshift 受管儲存單位			
使用 8、16 或 24 個 Redshift 處理單元 (RPU) 之資料倉儲的 Redshift 受管儲存單位	最多 128 TB	否	具有 8、16 或 24 個 RPU 的資料倉儲支援的最大 RMS 儲存空間。在 128 TB 限制內，Amazon Redshift 會在內部使用數百 MB 的儲存空間。
閒置或非作用中工作階段的逾時	1 小時	否	如需有關為使用者設定閒置工作階段逾時值的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 ALTER USER。會優先採用使用者設定。
執行中查詢的逾時	86,399 秒 (24 小時)	否	執行中的查詢在遭到 Amazon Redshift 結束前的執行時間上限。
閒置交易的逾時	6 小時	否	開啟中交易的非作用中期間上限，超過之後，Amazon Redshift Serverless 就會結束與交易相關聯的工作階段。此設定的優先順序高於任何使用者定義的閒置逾時設定。
連線數目上限	2000	否	允許連線至工作群組的連線數目上限。
工作群組數目	25	是	支援的工作群組數目。
命名空間數目	25	是	支援的命名空間數目。

配額名稱	AWS 預設值	可調整	說明
工作群組中的 Amazon Redshift 角色數目	1,000	是	您可以為每個工作群組建立的 Amazon Redshift 角色數目上限。如需角色型存取控制 (RBAC) 角色的相關資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的角色型存取控制 (RBAC)。
每個 的基本 Redshift 處理單元 RPUs) AWS 帳戶	3,200 個 RPU，或前六個月最大彙總基本 RPU 的 1.5 倍，以兩者中較大者為準。	是	Redshift Serverless 基礎 RPUs AWS 帳戶 的配額是前六個月 3，200 個 RPUs 或最大彙總基礎 RPUs 1.5 倍。 .

如需 Amazon Redshift Serverless 計費如何受逾時組態影響的相關資訊，請參閱 Amazon Redshift Serverless 的帳單。

Amazon Redshift Data API 的配額

Amazon Redshift 的配額限制了 Redshift Data API 的使用。每個類型都有預設值。如需 Amazon Redshift Data API 的詳細資訊，請參閱 使用 Amazon Redshift 資料 API。

配額名稱	AWS 預設值	可調整	說明
`BatchExecuteStatement` API 每秒交易數 (TPS)	20	否	您每秒可以提出而不受限制的操作請求數量上限。
`CancelStatement`	3	否	您每秒可以提出而不受限制的操作請求數量上限。

配額名稱	AWS 預設值	可調整	說明
API 每秒交易數 (TPS)			
DescribeStatement API 每秒交易數 (TPS)	100	否	您每秒可以提出而不受限制的操作請求數量上限。
DescribeTable API 每秒交易數 (TPS)	3	否	您每秒可以提出而不受限制的操作請求數量上限。
ExecuteStatement API 每秒交易數 (TPS)	30	否	您每秒可以提出而不受限制的操作請求數量上限。
GetStatementResult API 每秒交易數 (TPS)	20	否	您每秒可以提出而不受限制的操作請求數量上限。
ListDatabases API 每秒交易數 (TPS)	3	否	您每秒可以提出而不受限制的操作請求數量上限。
ListSchemas API 每秒交易數 (TPS)	3	否	您每秒可以提出而不受限制的操作請求數量上限。

配額名稱	AWS 預設值	可調整	說明
`ListStatements` API 每秒交易數 (TPS)	3	否	您每秒可以提出而不受限制的操作請求數量上限。
`ListTables` API 每秒交易數 (TPS)	3	否	您每秒可以提出而不受限制的操作請求數量上限。

查詢編輯器 v2 物件的配額

Amazon Redshift 的配額會限制 Amazon Redshift 查詢編輯器 v2 中數個物件類型的使用。每個類型都有預設值。

配額名稱	AWS 預設值	可調整	說明
連線	500	是	目前區域中，您可以在此帳戶中使用查詢編輯器 v2 建立的連線數目上限。
每個帳戶的作用中主體	50	是	在目前區域中，此帳戶中可使用查詢編輯器 v2 的同步主體數目上限。
已儲存的查詢	2,500	是	目前區域中，您可以在此帳戶中使用查詢編輯器 v2 建立的已儲存查詢數目上限。
查詢版本	20	是	目前區域中，您可以在此帳戶中使用查詢編輯器 v2 為每一查詢建立的版本數目上限。
已儲存的圖表	500	是	目前區域中，您可以在此帳戶中使用查詢編輯器 v2 建立的已儲存圖表數目上限。

配額名稱	AWS 預設值	可調整	說明
每一查詢擷取的資料大小	100	否	目前區域中，可在此帳戶中由查詢編輯器 v2 為每一查詢擷取的資料大小上限 (以 MB 為單位)。
每 100 列擷取的資料大小	5	否	在目前區域中，此帳戶中由查詢編輯器 v2 擷取的每一查詢結果頁面 (100 列) 的資料大小上限 (以 MB 為單位)。
並行連線數目上限	3	否	每一使用者的資料庫連線數目上限 (包括隔離的工作階段)。此值可由查詢編輯器 v2 管理員在帳戶設定中設定為 1–10。如果達到管理員所設的限制，請考慮在執行 SQL 時使用共用工作階段而非隔離的工作階段。如需連線的相關資訊，請參閱開啟查詢編輯器 v2。如需設定限制的相關資訊，請參閱帳戶設定。

Amazon Redshift Spectrum 物件的配額和限制

Amazon Redshift Spectrum 具有以下配額和限制：

- 使用 時，每個 AWS 帳戶的資料庫數目上限 AWS Glue Data Catalog。如需此值，請參閱《Amazon Web Services 一般參考》中的 AWS Glue 服務配額。
- 使用 AWS Glue Data Catalog時，每個資料庫的資料表數目上限。如需此值，請參閱《Amazon Web Services 一般參考》中的 AWS Glue 服務配額。
- 使用 AWS Glue Data Catalog時，每個資料保的分割區數目上限。如需此值，請參閱《Amazon Web Services 一般參考》中的 AWS Glue 服務配額。
- 使用 時，每個 AWS 帳戶的分割區數量上限 AWS Glue Data Catalog。如需此值，請參閱《Amazon Web Services 一般參考》中的 AWS Glue 服務配額。
- 使用 時外部資料表的欄數上限 AWS Glue Data Catalog，啟用虛擬欄時為 1，597，未啟用虛擬欄時為 1，600。
- 使用 時，ION 或 JSON 檔案中字串值的大小上限為 AWS Glue Data Catalog 16 KB。如果達到此限制，則可以截斷字串。
- 您可以使用單一 ALTER TABLE 陳述式新增最多 100 個分割區。

- 所有 S3 資料都必須位於與 Amazon Redshift 叢集相同的 AWS 區域。
- ION 和 JSON 中的時間戳記必須使用 ISO8601 格式。
- 不支援 ORC 檔案的外部壓縮。
- Text、OpenCSV 和 Regex SERDE 不支援大於 '\177' 的八進位分隔符號。
- 您可以在分割欄指定述詞，以避免讀取所有分割區。

  例如，下列述詞會篩選資料欄 `ship_dtm`，但不會將篩選條件套用至分割區資料欄 `ship_yyyymm`：

  ```
  WHERE ship_dtm > '2018-04-01'.
  ```

  若要略過不需要的分割區，您需要新增述詞 `WHERE ship_yyyymm = '201804'`。此述詞會將讀取操作限於分割區 `\ship_yyyymm=201804\`。

這些限制不適用於 Apache Hive 中繼存放區。

命名限制條件

下表說明 Amazon Redshift 中的命名限制。

叢集識別碼	• 叢集識別碼必須僅包含小寫字母。 • 他們必須包含 1–63 個英數字元或連字號。 • 第一個字元必須是字母。 • 不能以一個連字號結尾或是連續包含兩個連字號。 • 它在 AWS 帳戶的所有叢集中必須是唯一的。
資料庫名稱	• 資料庫名稱必須包含 1–64 個英數字元。 • 它只能包含小寫字母。 • 它不能是保留字。如需保留字清單，請參閱《Amazon Redshift 資料庫開發人員指南》中的保留字。

RedShift 受管 VPC 端點的端點名稱	• 端點名稱必須包含 1–30 個字元。 • 有效字元為 A-Z、a-z、0-9 和連字號 (-)。 • 第一個字元必須是字母。 • 名稱不能包含兩個連續連字號或以連字號結尾。
管理員使用者名稱	• 管理員使用者名稱必須僅包含小寫字母。 • 其必須包含 1–128 個英數字元。 • 第一個字元必須是字母。 • 它不能是保留字。如需保留字清單，請參閱《Amazon Redshift 資料庫開發人員指南》中的保留字。
管理員密碼	• 管理員密碼必須包含 8–64 個字元。 • 必須包含至少一個大寫字母。 • 必須包含至少一個小寫字母。 • 它必須包含一個數字。 • 它可以使用任何 ASCII 字元 (ASCII 碼 33–126)，但 ' (單引號)、" (雙引號)、\、/ 或 @ 除外。
參數群組名稱	• 參數群組名稱必須包含 1–255 個英數字元或連字號。 • 它只能包含小寫字元。 • 第一個字元必須是字母。 • 不能以一個連字號結尾或是連續包含兩個連字號。

叢集安全群組名稱	• 叢集安全群組名稱必須包含 255 個以內的英數字元或連字號。 • 它只能包含小寫字元。 • 它不得為 **Default**。 • 它對於您 AWS 帳戶建立的所有安全群組必須是唯一的。
子網路群組名稱	• 子網路安全群組名稱必須包含 255 個以內的英數字元或連字號。 • 它只能包含小寫字元。 • 它不得為 **Default**。 • 它對於您 AWS 帳戶建立的所有子網路群組必須是唯一的。
叢集快照識別碼	• 叢集快照識別碼必須包含 255 個以內的英數字元或連字號。 • 它只能包含小寫字元。 • 它不得為 **Default**。 • 它對於您 AWS 帳戶建立的所有快照識別符必須是唯一的。

在 Amazon Redshift 中標記資源

在 中 AWS，標籤是使用者定義的標籤，由索引鍵/值對組成。Amazon Redshift 支援標記功能，以讓您一目了然地查看資源的中繼資料，並根據成本分配來對帳單報告進行分類。若要使用標籤進行成本分配，您必須先在 AWS 帳單與成本管理 服務中啟用這些標籤。如需設定和使用標籤供計費使用的相關資訊，請參閱將成本分配標籤用於自訂帳單報告和設定每月成本分配報告。

在 Amazon Redshift 中的資源不須標籤，但它們可協助提供內容。您可能希望使用與成本中心、專案名稱和其他與資源之相關資訊的中繼資料來為資源加上標籤。例如，假設您想要追蹤哪些資源屬於測試環境和生產環境。您可以建立名為 `environment` 的索引鍵並提供值 `test` 或 `production` 來辨識每個環境中使用的資源。如果您在其他 AWS 服務中使用標記或具有業務的標準類別，我們建議您為 Amazon Redshift 中的資源建立相同的鍵值對，以保持一致性。

在您重新調整叢集的大小後與在相同區域內復原叢集的快照後，系統會保留資源的標籤。然而，如果您將快照複製到另一個區域，系統就不會保留標籤，因此您必須在該新區域中重新建立標籤。如果您刪除資源，任何關聯的標籤也會遭到刪除。

每個資源皆有一個標籤集，此為指派給該資源之一或多個標籤的集合。每個資源每個標籤集最多可擁有 50 個標籤。在您建立資源時與資源建立後，您可以新增標籤。您可以將標籤新增至 Amazon Redshift 中的以下資源類型：

- CIDR/IP
- 叢集
- 叢集安全群組
- 叢集安全群組傳入規則
- Amazon EC2 安全群組
- 硬體安全模組 (HSM) 連線
- HSM 用戶端憑證
- 參數群組
- 快照
- 子網路群組
- 整合 (零 ETL 整合)

若要從 Amazon Redshift 主控台使用標記功能，使用者可以附加 AWS 受管政策 `AmazonRedshiftFullAccess`。如需可附加至 Amazon Redshift 主控台使用者、具有有限標記許可

的 IAM 政策範例，請參閱範例 7：允許使用者使用 Amazon Redshift 主控台標記資源。如需標記的詳細資訊，請參閱什麼是 AWS 資源群組？。

標記需求

標籤均擁有以下要求：

- 索引鍵字首不能是 aws:。
- 索引鍵在標籤集內必須是唯一的。
- 索引鍵必須介於 1 到 128 個允許的字元之間。
- 值必須介於 0 到 256 個允許的字元之間。
- 值在每個標籤集中不需要是唯一的。
- 索引鍵和值的允許字元為 Unicode 字母、數字、空格和下列任何符號：_ . : / = + - @。
- 索引鍵和值會區分大小寫。

管理資源標籤

下列程序將逐步引導您使用資源的標籤。

管理 Amazon Redshift 資源上的標籤

1. 登入 AWS 管理主控台 ，並在 https://console.aws.amazon.com/redshiftv2/：// 開啟 Amazon Redshift 主控台。
2. 在導覽功能表上，選擇組態，然後選擇管理標籤。
3. 輸入您選擇的資源，並選擇要新增、修改或刪除的標籤。然後選擇 Manage tags of the resources that you chose (管理所選資源的標籤)。

   您可標記的資源包括叢集、參數群組、子網路群組、HSM 用戶端憑證、HSM 連線和快照。
4. 在管理標籤導覽窗格上，選擇檢閱和套用標籤變更，然後選擇套用以儲存變更。

AWS Backup 與 Amazon Redshift 整合

AWS Backup 是一種全受管服務，可協助您集中和自動化跨 AWS 服務、雲端和內部部署的資料保護。

使用 AWS Backup for Amazon Redshift，您可以在單一位置設定資料保護政策並監控不同 Amazon Redshift 資源的活動。您還可以在 Amazon Redshift 佈建叢集和無伺服器命名空間上建立和儲存快照。這可讓您自動化並合併之前必須單獨執行的備份任務，而不必進行任何手動程序。

> **Note**
>
> RG 或 RA3 佈建叢集和 Amazon Redshift Serverless 工作群組不支援無備份資料表。在 RG 或 RA3 叢集或無伺服器工作群組中標記為無備份的資料表會被視為永久資料表，在拍攝快照時一律會備份，並在從快照還原時一律還原。為了避免無備份資料表的快照成本，請在拍攝快照之前截斷這些資料表。

備份或復原點代表資源的內容，例如 Amazon Redshift 叢集，在指定的時間。 AWS Backup 會將備份儲存在備份文件庫中，您可以根據業務需求進行組織。復原點和備份這兩個詞可互換使用。如需詳細資訊 AWS Backup，請參閱《 AWS Backup 開發人員指南》中的備份建立、維護和還原。

Amazon Redshift 原生與 整合 AWS Backup。這可讓您定義備份計畫並將 Amazon Redshift 資源指派給備份計畫。 AWS Backup 會自動建立 Amazon Redshift 手動快照，並將這些快照安全地存放在您在備份計畫中指定的備份文件庫中。如需關於保存庫的詳細資訊，請參閱《AWS Backup 開發人員指南》中的備份保存庫。在備份計劃中，您可以定義備份頻率、備份時段、生命週期或備份保存庫。如需關於備份計畫的資訊，請參閱《AWS Backup 開發人員指南》中的備份計畫。

如需不使用 AWS Backup建立和還原 Amazon Redshift Serverless 快照的相關資訊，請參閱 快照和復原點。如需不使用 建立和還原 Amazon Redshift 佈建叢集快照的相關資訊， AWS Backup 請參閱 Amazon Redshift 快照和備份。

主題

- 搭配使用 AWS Backup 與 Amazon Redshift 的考量事項
- 搭配使用 AWS Backup 與 Amazon Redshift 的限制
- AWS Backup 使用 Amazon Redshift 管理

搭配使用 AWS Backup 與 Amazon Redshift 的考量事項

以下是 AWS Backup 搭配 Amazon Redshift 使用 的考量事項：

- AWS Backup 適用於 Amazon Redshift 的 適用於同時在相同 AWS Backup 和 Amazon Redshift 中的 AWS 區域。如需 AWS Backup 可用 的詳細資訊，請參閱 中的 Amazon Redshift 端點和配額AWS 一般參考。
- 若要開始使用 AWS Backup，請確認您已符合所有先決條件。如需詳細資訊，請參閱《AWS Backup 開發人員指南》中的先決條件。
- 肯定選擇加入 AWS Backup 服務。選擇加入選項適用於特定帳戶和 AWS 區域。如果您想要在多個區域中透過特定帳戶使用備份，您必須使用該帳戶選擇加入每個個別區域。如需詳細資訊，請參閱《AWS Backup 開發人員指南》中的選擇使用 AWS Backup管理服務。
- AWS Backup Amazon Redshift 的 整合僅支援佈建叢集和無伺服器命名空間的手動快照。
- 使用 AWS Backup 管理快照設定後，您就無法繼續使用 Amazon Redshift 管理手動快照設定。反之，您可以繼續使用 AWS Backup 計劃來管理設定。如需詳細資訊，請參閱《AWS Backup 開發人員指南》中的備份計畫。
- 將整個資料倉儲快照還原至無伺服器命名空間是一種破壞性變更。當您將資料倉儲快照還原至目標命名空間時，該命名空間中所有先前存在的資料都會遺失。這僅適用於還原資料倉儲快照。將單一資料表快照還原至命名空間並不會刪除現有資料。
- 若要將快照還原至佈建叢集，您需要具有 `RestoreFromClusterSnapshot` 許可的 IAM 政策。若要將快照還原至無伺服器命名空間，您需要具有 `RestoreFromSnapshot` 許可的 IAM 政策。這些許可適用於目標資料倉儲類型，不適用於來源快照類型。例如，若要將叢集快照還原至命名空間，您需要 `RestoreFromSnapshot` 許可，而不是 `RestoreFromClusterSnapshot`。如需管理 IAM 政策的詳細資訊，請參閱 Amazon Redshift 中的身分和存取管理。

搭配使用 AWS Backup 與 Amazon Redshift 的限制

以下是 AWS Backup 搭配 Amazon Redshift 使用 的限制：

- 您無法使用 AWS Backup 管理 Amazon Redshift 自動化快照。若要管理自動快照，請使用標籤。如需有關標記資源的資訊，請參閱在 Amazon Redshift 中標記資源。
- 從快照還原單一資料表時，您無法從佈建叢集快照還原至無伺服器命名空間，反之亦然。您可以在任何組態中還原整個快照。例如，您可以將佈建叢集快照中的所有資料庫還原至無伺服器命名空間，但無法將單一資料表從相同的快照還原至相同的命名空間。

AWS Backup 使用 Amazon Redshift 管理

若要保護 Amazon Redshift 資料倉儲上的資源，您可以使用 AWS Backup 主控台，或以程式設計方式使用 AWS Backup API 或 AWS Command Line Interface (AWS CLI)。當您需要復原資源時，您可以使用 AWS Backup 主控台或 AWS CLI 來尋找和復原所需的資源。如需詳細資訊，請參閱AWS Command Line Interface。

使用 AWS Backup for Amazon Redshift 時，您可以執行下列動作：

- 建立會自動啟動 Amazon Redshift 快照的定期備份。定期備份對於滿足您的長期資料保留需求非常有用。如需詳細資訊，請參閱《AWS Backup 開發人員指南》中的 Amazon Redshift 備份。
- 集中設定備份計劃，將備份排程與保留自動化。
- 將佈建叢集或無伺服器命名空間還原至您選擇的已儲存備份。您可以選擇從快照或單一資料表還原所有資料。您必須設定備份資源的頻率。如需有關還原佈建叢集快照的資訊，請參閱《AWS Backup 開發人員指南》中的還原 Amazon Redshift 叢集。如需有關還原無伺服器命名空間快照的資訊，請參閱《AWS Backup 開發人員指南》中的 Amazon Redshift Serverless 還原。

適用於 Amazon Redshift 的叢集版本

Amazon Redshift 會定期發行叢集版本。您的 Amazon Redshift 叢集會在系統維護時段進行修補。修補程式的時間取決於您的 AWS 區域 和維護時段設定。您可以從 Amazon Redshift 檢視或變更維護時段設定。如需維護的相關資訊，請參閱叢集維護。

您可以在 Amazon Redshift 主控台上，於叢集詳細資訊的維護索引標籤上檢視您叢集的叢集版本。或者，您也可以在 SQL 命令的輸出中查看叢集版本：

```
SELECT version();
```

> **Note**
>
> 影響 Amazon Redshift 行為重大更新會隨著 Amazon Redshift 演進而推出。若要掌握這些變更、採取動作，並防止工作負載可能中斷的情形發生，請參閱 Amazon Redshift 中的行為變更。

主題

- Amazon Redshift 修補程式 205
- Amazon Redshift 修補程式 204
- Amazon Redshift 修補程式 203
- Amazon Redshift 修補程式 202
- Amazon Redshift 修補程式 201
- Amazon Redshift 修補程式 200
- Amazon Redshift 修補程式 199
- Amazon Redshift 修補程式 198
- Amazon Redshift 修補程式 197
- Amazon Redshift 修補程式 196
- Amazon Redshift 修補程式 195
- Amazon Redshift 修補程式 194
- Amazon Redshift 修補程式 193
- Amazon Redshift 修補程式 192

- Amazon Redshift 修補程式 191
- Amazon Redshift 修補程式 190
- Amazon Redshift 修補程式 189
- Amazon Redshift 修補程式 188
- Amazon Redshift 修補程式 187
- Amazon Redshift 修補程式 186
- Amazon Redshift 修補程式 185
- Amazon Redshift 修補程式 184
- Amazon Redshift 修補程式 183
- Amazon Redshift 修補程式 182
- Amazon Redshift 修補程式 181
- Amazon Redshift 修補程式 180
- Amazon Redshift 修補程式 179
- Amazon Redshift 修補程式 178
- Amazon Redshift 修補程式 177
- Amazon Redshift 修補程式 176
- Amazon Redshift 修補程式 175
- Amazon Redshift 修補程式 174
- Amazon Redshift 修補程式 173
- Amazon Redshift 修補程式 172
- Amazon Redshift 修補程式 171
- Amazon Redshift 修補程式 170
- Amazon Redshift 修補程式 169
- Amazon Redshift 修補程式 168

Amazon Redshift 修補程式 205

此修補程式中的叢集版本：

- 1.0.477953 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 10 月 6 日發行

- 1.0.434008 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 9 月 16 日發行

此修補程式中的新功能和改善

- Iceberg MV – 存放在 Iceberg 中的具體化視觀表提供增量重新整理和完整重新計算。
- 系統資料表現在包含 `external_user_id`資料欄，可記錄與每個查詢相關聯的耐久 IAM 或 IAM Identity Center 身分，讓每個使用者能夠跨叢集生命週期事件進行篩選。
- 外部資料表上的 Amazon Redshift Spectrum 查詢現在支援讀取 gzip 和 bzip2 壓縮的文字/CSV 檔案。
- 新增對 COPY 和 UNLOAD 的 IAM_ROLE 'SESSION' 關鍵字的支援，允許客戶使用工作階段繫結的 FAS 登入資料進行 Amazon S3 存取，而不是指定明確的 IAM 角色 ARN。
- 已新增 `ALTER DATABASE ... INTEGRATION REFRESH REMEDIABLE TABLES`，這會重新同步標記為隨需重複日期警告的零 ETL 資料表，選擇性地範圍為結構描述。
- 新增零 ETL 整合的 `IntegrationLatestSourceCommit` Amazon CloudWatch 指標，報告觀察到的最新來源遞交時間，讓您可以區分閒置來源與停滯的管道。
- Amazon Redshift 現在可讓管理員使用 `ALTER SYSTEM SET passwd_min_length = N`(8–32) 提高資料庫使用者密碼所需的長度下限。
- 修正 Azure 身分提供者上設定自動建立角色的不正確參數錯誤`ALTER IDENTITY PROVIDER ... ENABLE`失敗的問題。
- Amazon Redshift 現在支援 TLS 連線的後量子混合金鑰交換 (`X25519MLKEM768`)。
- Amazon Redshift 現在支援聯合許可目錄資料庫的`DEBUG`許可，讓全域身分、IAM Identity Center (IdC) 群組或消費者帳戶的管理員檢視未修訂的安全日誌記錄，以取得精細存取控制 (FGAC) 保護的資料。
- Amazon Redshift 現在支援具有中繼資料安全性的範圍許可。
- Iceberg 資料表現在可以使用 NOT NULL 資料欄限制條件建立。
- Amazon Redshift 起始的 API 呼叫的 Amazon CloudTrail 稽核事件現在包含與系統資料表相互關聯的`queryUuid`欄位。
- 為了保持一致性，在系統資料表和檢視`query_uuid`中將資料`user_query_uuid`欄重新命名為 。
- 在高查詢並行下，改善具有零 ETL 整合之叢集的查詢效能和可用性。
- 現在，對具有大量資料欄和少數資料欄層級許可的資料表進行許可檢查的速度更快，可加速中繼資料查詢和列出資料表權限的用戶端工具。
- 針對在彙總中使用 Lambda UDFs查詢增強穩定性。

- 自動膨脹移除現在可讓超級區塊收縮以符合其即時標頭集，不再需要透過超級區塊清空 (SBV) 回收膨脹。
- 改善了自動資料表排序，以減少具有大量資料欄之資料表的儲存成長。
- 修正在叢集重新啟動之前，可能停止自動清空在叢集上執行的問題。
- 修正使用自動工作負載管理並行擴展，在叢集的清空操作期間可能導致叢集重新啟動的問題。
- 已修正可能導致包含字串常值之遞迴 CTEs 類型推論錯誤的錯誤。
- 已修正聯合 PostgreSQL 查詢可能在無回應的遠端資料庫上懸置，並忽略取消請求的問題。
- 已修正記憶體壓力下，COPY 負載狀態管理中無兩次所造成的罕見叢集重新啟動。
- 修正在 `LIKE`子句中指定長模式時，`SHOW`探索命令找不到結果的問題。
- 已修正在已達到資料庫角色數量上限的叢集上建立或發佈資料共用可能會失敗的問題。
- 取消`DROP DATABASE`命令現在會以錯誤結束命令，而不是導致非預期的叢集重新啟動。
- 改善在使用並行擴展時多次查詢 Iceberg 資料表，然後在本機資料表上查詢之預存程序的可靠性。
- 改善啟用並行擴展時，將 Iceberg 游標與非 Iceberg 查詢混合的工作階段可靠性。
- 新增讓 fdisk fetch-block IPC 等待可取消的功能，讓遺失的 ACK 成為可刪除的查詢，而不是可停滯的懸置。
- 改善了線上彈性調整大小期間使用的內部程序的可靠性，使程序更確定/強大。

Amazon Redshift 修補程式 204

此修補程式中的叢集版本：

- 1.0.436211 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 9 月 22 日發行
- 1.0.436211 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 9 月 10 日發行
- 1.0.416217 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 8 月 25 日發行
- 1.0.394035 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 8 月 11 日發行

此修補程式中的新功能和改善

- 新增對單一節點 rg.large 叢集的支援。這是建立單一節點 rg.large 叢集所需的最低叢集版本。

- 將 query_uuid 資料欄新增至系統資料表和檢視，提供全域唯一查詢識別符，該識別符會保留在叢集暫停/繼續週期中。
- 已將 user_query_id 新增至 stl_udf_log 資料表，以移除 sys_ 檢視定義中 stl_user_query_map 聯結的相依性。
- 針對零 ETL 整合新增兩個新的 Amazon CloudWatch 指標 IntegrationLatestDetectedChange 和 IntegrationLatestAppliedChange，以允許監控複寫運作狀態。
- 新增零 ETL 整合資料庫上 CREATE DATABASE 和 ALTER DATABASE 的 AUTO_REMEDIATION 選項。設定 AUTO_REMEDIATION 時，Amazon Redshift 會自動重新同步存在複寫問題的資料表。詳細資訊會顯示在 svv_integration_table_state.reason 中。
- 新增支援使用 Amazon Redshift 原生引擎讀取、建立和寫入 Apache Iceberg v3 資料表。這包括對所有先前支援之資料類型的預設值、資料列歷程記錄和刪除向量的支援。此版本不包含對變體、地理和幾何資料類型的支援，並將在未來的修補程式中宣布。
- 將 Iceberg 資料表升級至 v3 後，Iceberg `timestamptz`類型會映射至 Amazon Redshift `TIMESTAMPTZ`類型。在 Iceberg v2 資料表中， `timestamptz` 會對應至 Amazon Redshift `TIMESTAMP`類型。使用 v3，這表示您的查詢會根據其時區輸出時間戳記。
- 新增對超過執行個體類型結果集大小上限`CURSOR`的 的支援。當游標的結果集超過可用的磁碟容量時，Amazon Redshift 會自動將溢位資料溢出到 Amazon S3。向 AWS Support 請求時，每個叢集都可以啟用此功能。
- 已將新的 bin 封裝指標資料欄新增至`stl_fasma_data_cache_stats`系統資料表，以監控快取活動。
- 參考延遲繫結檢視 (LBVs) 的查詢現在會在取用者叢集上進行剖析和語意分析。這會消除 LBV 處理中的生產者端瓶頸。
- Amazon Redshift ML 現在會在對 Amazon SageMaker 提出`CreateAutoMLJob`和`CreateProcessingJob`請求時啟用容器間流量加密選項。這符合已套用至 的設定，`BuildTrainingRequest`並防止安全掃描器警示。
- 改善高頻率工作負載的零 ETL 整合效能，以減少具有許多並行更新之資料表的複寫延遲。
- 零 ETL 整合現在會將超過目標資料欄長度的來源資料正確報告為無法重試的錯誤。
- 改善叢集重新啟動後的零 ETL 整合穩定性。
- 現在，當零 ETL 整合上的資料表篩選條件變更從結構描述中移除所有資料表時，會自動清除空白結構描述。
- 改善在具有作用中零 ETL 整合的資料庫上嘗試 DROP DATABASE 時的錯誤訊息。
- DynamoDB 零 ETL 整合現在會在初始植入和持續複寫期間遵守 ACCEPTINVCHARS 和 TRUNCATECOLUMNS 選項。當來源包含無效的 UTF-8 字元時，資料表會繼續複寫。

- 改善具有零 ETL 整合之叢集的查詢效能，尤其是複寫許多資料庫或資料表的工作負載。
- 改善使用暫存資料表的工作負載效能、降低目錄存取額外負荷並改善整體輸送量。
- SHOW 探索命令，例如 SHOW TABLES、SHOW SCHEMAS、SHOW COLUMNS 和 SHOW GRANTS，現在最多可傳回 133，500 個物件，因此結構描述和具有非常大物件計數的資料庫上的中繼資料清單會成功，而不是失敗，資料列限制錯誤為 10，000。
- 當 DROP USER 命令因為使用者仍擁有物件或保留權限而遭到拒絕時，錯誤訊息現在會命名特定封鎖物件，並報告還有多少其他相依性。
- Amazon Redshift 現在會在領導節點上快取 IAM 角色登入資料。這可減少 COPY、UNLOAD、Amazon Redshift Spectrum 和 Lambda UDF 查詢的擷取開銷。常用的登入資料會在查詢之間重複使用，從而縮短查詢啟動時間。
- AWS IAM Identity Center 身分驗證現在可在啟用增強型 VPC 路由的叢集和工作群組上運作。
- 改善 RG 節點類型的磁碟運作狀態偵測，以防止在正常磁碟 I/O 下運作的叢集上進行不必要的自動硬體修復。
- 透過略過後端程序中的額外複本來增強資料共用效能。
- 擴展消費者大小支援多倉儲寫入查詢。
- 增強 Lambda 使用者定義函數的穩定性，聯結中沒有引數。
- 修正述詞包含 IDF 時 MDDL 排序索引鍵運算的問題。
- 修正在 BEGIN READ ONLY 交易內執行時，外部資料表、資料共用物件和跨資料庫參考的查詢因「交易為唯讀」而失敗的問題。
- 修正在相同資料表上與自動資料表最佳化同時執行的 COPY 或 INSERT 可能導致叢集重新啟動的罕見情況。
- 已修正罕見的編碼不相符錯誤，可能導致透過在較舊叢集版本上建立的資料共用存取的資料表上的自動資料表最佳化失敗。
- 已修正在資料分佈偏斜的資料表上自動清空可能導致叢集重新啟動的罕見情況。
- 修正 Auto COPY 任務可能停止進行進度並封鎖相同叢集上其他 Auto COPY 操作的問題。
- 已修正 COPY 命令在查詢清除期間因競爭條件而可能導致叢集重新啟動的罕見條件。
- 已修正 svl_s3query_summary.s3query_returned_rows 可能針對外部資料表上具有 LIMIT 子句的查詢報告多一列的問題。
- 已修正在具有零 ETL 整合且未設定 Auto COPY 任務的叢集上，在引導期間可能發生的罕見叢集重新啟動。
- 修正將多字元分隔符號與 ESCAPE 選項搭配使用時 COPY 欄位剖析的問題。

- 修正當透過 IAM Identity Center 連線的使用者使用 CREDENTIALS 參數提供明確的暫時登入資料時，COPY 和 UNLOAD 命令失敗的問題。
- 修正 SHOW CONSTRAINTS 中可能導致叢集重新啟動的問題。
- 修正授予角色的 CONNECT 許可不允許映射群組中的 AWS IAM Identity Center 使用者在已啟用 Amazon Redshift 聯合許可的資料庫上連線的問題。
- 修正在 SUPER 陣列上使用 UNNEST 的查詢錯誤地無法在並行擴展叢集上執行的問題。
- 修正將 ARRAY_SORT 函數套用至巢狀 SUPER 結構時，可能會失敗並出現路徑導覽錯誤的問題。
- 修正深度巢狀 PartiQL 或 SUPER 表達式可能導致叢集重新啟動的問題。
- 修正當格式字串指定的精確度大於 38 時，TO_NUMBER 函數可能會失敗的問題。
- 修正加密金鑰輪換可能會損毀移植資料表金鑰的問題，這可能會導致資料存取失敗。
- 已修正儲存操作中可能導致查詢停止或節點無回應的死結條件。
- 已修正在磁碟區塊管理期間可能導致節點重新啟動的罕見當機問題。
- 已修正在區塊配置期間out-of-memory而導致的當機，這可能會導致節點重新啟動。
- 已修正在區塊複寫指派期間，可能導致節點重新啟動的磁碟完整條件損毀。

Amazon Redshift 修補程式 203

此修補程式中的叢集版本：

- 1.0.417950 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 8 月 26 日發行
- 1.0.417950 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 8 月 25 日發行
- 1.0.400034 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 8 月 17 日發行
- 1.0.384821 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 8 月 4 日發行
- 1.0.377293 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 7 月 25 日發行
- 1.0.365190 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 7 月 21 日發行
- 1.0.347559 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 7 月 6 日發行

此修補程式中的新功能和改善

- 已將新資料欄新增至 stl_alter_dist_sort，以指出在主叢集、資料共用取用者或並行擴展叢集上執行的 ALTER DISTKEY 和 ALTER SORTKEY 操作。
- 新增 SHOW GRANTS FOR GROUP 語法，以探索授予特定使用者群組的許可。
- 透過新增檢查點的精細成功狀態來增強零 ETL 整合可觀測性，從而對系統資料表中的複寫進度進行更精細的監控。
- 引進在零 ETL 整合中正常處理來源資料表結構描述中不支援的 UTF-8 字元和 DDL 中繼資料，以防止整合失敗。
- Amazon Redshift 現在支援並行擴展，以重新整理 Amazon Kinesis Data Streams (KDS) 連線的串流具體化視觀表 (MV)。
- 從修補程式 203 開始，Amazon Redshift 開始強制執行對現有 Python UDFs終止支援。在修補程式 198 中已棄用建立新的 Python UDFs，且所有 Python UDF 支援都會在 2026 年 6 月 30 日之後結束。將您現有的 Python UDFs 遷移至 Lambda UDFs，以確保持續運作。如需遷移指引，請參閱 AWS 大數據部落格上的 Amazon Redshift Python 使用者定義函數將於 2026 年 6 月 30 日之後終止支援，以及 Amazon Redshift Lambda 使用者定義函數的最佳實務。如需行為變更的詳細資訊，請參閱 純量 Python UDF 將於 2026 年 6 月 30 日之後終止支援。
- 當資料表具有主索引鍵時，從 Amazon Redshift Spectrum 載入 COPY、具有明確資料欄編碼或非空白資料表的資料表現在符合自動 DISTKEY 指派的資格，從而改善主索引鍵上聯結的查詢效能。
- 更新資料共用資料表上的 ALTER TABLE ... ALTER DISTKEY 錯誤訊息，以顯示面向客戶的資料表名稱。
- 減少叢集上具有大量資料表的自動清空背景活動，改善先前經歷過多維護活動的叢集穩定性。
- 在某些情況下，透過將子查詢與不相符的篩選條件、彙總和視窗函數結合來最佳化查詢效能。
- 改善資料表重新同步後，在具有 Amazon DynamoDB 歷史記錄模式的零 ETL 資料表上，時間歷程查詢的時間戳記準確性。
- 在具有特定主索引鍵組態的資料表上處理 CDC 檢查點時，零 ETL 整合不會再進入重新同步迴圈。
- 改善無效頻道篩選條件表達式的零 ETL 整合錯誤報告。
- 原始查詢文字不再使用安全記錄遮罩（適用於針對外部 AWS Lake Formation 檢視或具有聯合 Amazon Redshift 許可的資料庫的查詢）。
- 將大型資料表的 ANALYZE 效能提升高達 30%。
- 使用 AWS Lambda 使用者定義函數搭配 ROLLUP 彙總來增強查詢的效能。
- 透過略過雜湊聯結外部資料表上不必要的資料掃描，增強 ORDER BY with LIMIT 子句的查詢效能。

- 減少背景樹狀目錄清理操作期間的記憶體耗用，消除了整個節點當機機機機群。透過實作倖存者計數的批次耗盡來限制運算節點樹狀目錄的記憶體佔用，防止高中繼資料流失下的記憶體無限制成長。樹狀目錄清除路徑現在會在背景樹狀目錄重播完成時鎖定。
- 改善 Amazon S3 讀取失敗後的故障復原，消除節點當機。慢速樹疊代運算現在會在 Amazon S3 讀取失敗後正確重設，確保後續的讀取重試正確，而不是無提示地傳回過時或不完整的資料。
- 已修正以下問題：調整叢集調整大小後，使用 ALTER TABLE ... ALTER DISTSTYLE AUTO 修改的資料表可能會無限期保持在 sys_restore_state 的待定狀態。
- 修正 SYS_QUERY_TEXT 系統檢視可能會在區塊邊界顯示缺少空格或錯誤分割單字的查詢文字的問題。
- 修正啟動失敗，因為 Amazon Redshift Catalog 中繼資料不一致，導致 2023 年 8 月至 2024 年 3 月之間建立之叢集的零 ETL 整合。
- 修正 SHOW GRANTS 錯誤顯示授予 identity_type "role" 使用者群組的許可的問題。
- 已修正在工作階段暫停處理期間查詢可能無限期停止回應的問題。
- 修正 SYSLOG ACCESS UNRESTRICTED 的非超級使用者只能在 sys_session_history 中查看自己的工作階段資料，而不是所有使用者的工作階段如預期的問題。
- 已修正以下問題：資料庫或使用者連線限制設為 1000 或更高，導致連線數量不正確。
- 修正查詢可能會在記憶體壓力下無限期停止回應的問題。
- 修正 Amazon Redshift Serverless 端點在暫停操作期間可能會短暫無法使用的問題。
- 修正具有特定參數組合的 COPY 命令在查詢編譯期間可能會意外失敗的問題。
- 修正使用 APPROXIMATE PERCENTILE_DISC 搭配 GROUP BY 的查詢可能失敗並發生編譯錯誤的問題。
- 修正使用 COLLATE 表達式的查詢可能傳回不正確資料欄值的錯誤結果問題。
- 修正具有 SUPER 類型值的 LEAST 和 GREATEST 函數可能導致查詢失敗的問題。
- 修正 REGEXP_REPLACE 中錯誤的結果問題，其中具有擷取群組參考的替換字串可能會產生截斷的輸出。
- 已修正中止重新複寫讀取期間潛在的資料損毀。已解決當重新複寫讀取在傳輸中中止時可能發生的雙無條件，以防止可能的記憶體損毀或當機。

Amazon Redshift 修補程式 202

此修補程式中的叢集版本：

- 1.0.369760 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 7 月 29 日發行
- 1.0.377292 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 7 月 25 日發行
- 1.0.377292 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 7 月 25 日發行
- 1.0.358853 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 7 月 21 日發行
- 1.0.358853 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 7 月 14 日發行
- 1.0.343555 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 6 月 29 日發行
- 1.0.340728 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 6 月 22 日發行
- 1.0.331549 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 6 月 16 日發行
- 1.0.323213 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 6 月 3 日發行
- 1.0.314775 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 5 月 28 日發行

此修補程式中的新功能和改善

- 新增對 RG 節點類型的支援 (rg.large 和 rg.12xlarge 多節點）。這是遷移至 rg.large 和 rg.12xlarge 所需的最低叢集版本。
- 新增支援超級使用者捨棄處於 Lakehouse 類型停用狀態的 IAM Identity Center 身分提供者
- STL_QUERY 和 STL_QUERYTEXT 系統資料表中標示為「JIT ANALYZE」的表面內部維護查詢。這些查詢由 Amazon Redshift 自動執行，以收集外部資料表統計資料並保持查詢計劃最佳化。不需要客戶動作。
- 我們已最佳化 Kinesis 串流身分驗證，以減少 `DescribeStreamSummary` API 呼叫。這將避免為具有許多 Kinesis 串流 MVs Redshift 客戶達到 Kinesis API 配額。如需詳細資訊，請參閱 Redshift 串流文件一節。
- 針對具有 GROUP BY、ORDER BY 和 LIMIT 子句的查詢使用 Top K Aggregation 來增強查詢執行效能，從而減少彙總期間處理的資料量。

- 將複合篩選條件述詞 (OR、IN、 NOT IN) 推送至 AWS Glue GetPartitions API，以增強分割 Glue Data Catalog 資料表上查詢的效能，減少擷取的分割區數量。
- 使用 Lambda 使用者定義函數搭配 Window 函數增強查詢的效能。
- 在雜湊聯結的聯結篩選條件中使用 Lambda 使用者定義函數，增強特定查詢的效能。
- 針對在聯結中使用 Lambda UDFs查詢增強穩定性。
- 建立 AWS Lambda 使用者定義函數時，允許使用 ARN 做為 Lambda 的名稱。
- 透過最佳化 COALESCE 和 CASE WHEN 表達式的記憶體使用量來改善查詢效能。
- 將進階篩選套用至包含評估一次之子計畫的查詢，其結果會在其餘執行期間重複使用為常數。
- 使用 ST_Transform 改善第一次執行查詢的效能
- 透過接受包含重複資料列索引的位置刪除檔案來改善 Apache Iceberg 相容性，這些索引可由 AWS Glue 壓縮產生。
- 在刪除檔案參考中接受 s3a：// 路徑，以改善 Apache Iceberg 相容性。
- 改善處理未正確設定資料表時 Spectrum 分割區層級擷取錯誤的處理
- 改善透過使用在資料庫間調用巢狀 SQL UDFs 之預存程序的穩定性
- 改善預存程序上不支援操作的錯誤訊息，例如從資料共用新增或移除它們。
- 修正使用 EVR 時，無法偵測客戶 VPC 資訊的錯誤。此資訊僅適用於訓練本機 Redshift ML 模型，不會影響任何其他 Redshift 功能。這不會阻止使用現有的已訓練 Redshift ML 模型。
- 修正當內部查詢 IDs計數器接近其最大值時，叢集暫停和繼續後，系統資料表中可能會出現重複查詢 ID 的問題。
- 修正 STL 系統資料表上的查詢在多可用區容錯移轉後，因為內部快取不一致而可能傳回不完整結果的問題
- 修正延伸資料列描述中的「isAutoIncremented」欄位可能傳回 IDENTITY 資料欄不正確值的問題
- 修正以下問題：指定 NULL 輸入時，ARRAY_CONTAINS 和 ARRAY_POSITION 函數未如預期傳回 NULL。
- 修正在向量化掃描期間具有多個合併篩選條件述詞的查詢可能會意外失敗的罕見問題。
- 已修正並行 COPY 操作在資料載入過程中可能導致叢集不穩定的罕見問題
- 修正自動維護操作如清空和分析可能會在程序關閉期間造成叢集不穩定的問題
- 修正在動態關閉並重新開啟工作負載管理 (WLM) 短期查詢加速之後，查詢可能會失敗的問題
- 已修正使用歷史記錄模式之零 ETL 資料表上的時間歷程查詢問題。
- 修正在並行用戶端關閉期間，零 ETL 整合可能會變得沒有回應的問題
- 修正零 ETL 資料表共用來源碎片可能會觸發不必要的重新同步的問題

- 修正當內部原因欄位超過允許的長度時，零 ETL 整合資料表狀態更新可能會失敗的問題
- 修正在 Aurora PostgreSQL 分割資料表上的資料欄變更期間，零 ETL 整合發生錯誤的問題

Amazon Redshift 修補程式 201

此修補程式中的叢集版本：

- 1.0.331388 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 6 月 16 日發行
- 1.0.331388 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 6 月 15 日發行
- 1.0.326225 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 6 月 8 日發行
- 1.0.319797 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 6 月 3 日發行
- 1.0.313685 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 5 月 27 日發行
- 1.0.300094 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 5 月 12 日發行
- 1.0.298697 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 5 月 7 日發行
- 1.0.292316 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 5 月 4 日發行
- 1.0.280974 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 2026 年 4 月 22 日發行

此修補程式中的新功能和改善

- 新增對 RG 節點類型的支援 (rg.xlarge 和 rg.4xlarge 多節點）。這是遷移至 rg.xlarge 和 rg.4xlarge 所需的最低叢集版本。
- 新增對 Iceberg 資料表上 ALTERs支援。
- 新增支援寫入掛載在 AwsDataCatalog 下的 Iceberg 資料表。
- 新增使用 USE 命令連線至資料庫時對 SHOW GRANTS 命令的支援。

- 新增對具有聯合許可之身分的 SHOW GRANTS 和 SHOW COLUMN GRANTS 命令的支援。
- 新增了對具體化視觀表自動重新整理的支援，作為在 Amazon Redshift Serverless 上具有高載容量的使用者工作負載。
- 允許 AutoWLM 將更多叢集記憶體指派給寫入寬資料表的查詢。
- Amazon Redshift 改善了 SVV_ATTACHED_MASKING_POLICY 系統檢視的查詢效能，減少具有大量許可和動態資料遮罩政策附件之執行個體的執行時間。
- 從 2.8.x 將 librdkafka 程式庫升級至 2.13.x 版，以提供最新的錯誤修正和效能增強功能。
- 在彙總的 HAVING 子句中使用 Lambda 使用者定義函數增強查詢的效能。
- 使用 Lambda 使用者定義函數搭配視窗函數增強查詢的效能。
- 增強 Lambda 使用者定義函數的穩定性，不含引數。
- 透過消除不必要的資料掃描，在使用 ORDER BY 和 LIMIT 子句時增強了資料湖資料表查詢執行效能。
- 改善零 ETL 整合的錯誤訊息，為不支援的限制、不支援的 DDL 陳述式、日期精確度損失、多位元組字元長度不相符和略過的來源資料表提供更清楚的詳細資訊。
- 透過在複寫狀態檢查期間正常處理登入資料擷取失敗，改善零 ETL 整合彈性。
- 改善執行並行複寫任務時零 ETL 整合的效能。
- 改善了具有經常更新資料列之零 ETL 資料表的時間歷程查詢結果，以確保每個歷史資料列版本反映其個別刪除時間戳記。
- 修正 AI 驅動擴展無法正確調整主叢集大小的一些邊緣案例。
- 修正以下問題：當工作負載管理 (WLM) 移出並重試查詢時，Amazon CloudWatch 指標可能會報告重複的失敗查詢計數。
- 修正 COPY 與 Redshift 資料表複製可能產生分佈樣式不相符的問題。
- 修正分割區排序結尾列計算不正確的問題，這些問題可能會影響範圍限制掃描結果。
- 修正當資料表同時捨棄時，自動清空操作可能會失敗的問題。
- 修正 VACUUM 在取得關係時可能使用過時資料表版本的問題，這可能會導致錯誤。
- 確保清空交易映射在高載叢集上正確清除。
- 修正透過資料共用存取時，零 ETL 複寫資料表上的查詢可能會失敗的問題。
- 修正零 ETL 整合資料複寫中可能導致叢集重新啟動的問題。
- 修正在更新合併操作期間，與 DynamoDB 歷史記錄模式的零 ETL 整合可能會失敗的問題。
- 修正零 ETL 整合資料表在 Amazon Redshift 主控台中可能顯示不正確資料列計數的問題。
- 修正在零 ETL 整合操作期間，Amazon Redshift Serverless 工作群組無法暫停的問題。

- 修正空的 Avro 檔案導致 COPY 擷取失敗的問題。現在會略過空的 Avro 檔案，並且繼續擷取而不會發生錯誤。
- 修正使用與位置無關的超級區塊時，epsilon 樹狀結構重播期間的過時區塊標頭檢查總和問題。

Amazon Redshift 修補程式 200

此修補程式中的叢集版本：

- 1.0.314426 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 5 月 27 日發行
- 1.0.293472 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 5 月 7 日發行
- 1.0.293472 – CURRENT 追蹤 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 4 月 30 日發行
- 1.0.266483 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 4 月 9 日發行
- 1.0.257586 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 3 月 27 日發行

此修補程式中的新功能和改善

- 修正使用多維資料配置排序索引鍵的資料表上 INSERT 操作可能會失敗的問題。
- 改善多維資料配置 (MDDL) 排序索引鍵的 Amazon Redshift Advisor 建議，適用於可受益於工作負載型排序的資料表。
- 增強型 Redshift Serverless AI 驅動的擴展和最佳化搭配改善的磁碟使用訊號，實現更準確且回應靈敏的擴展決策。
- 增強型 Redshift Serverless AI 驅動的擴展和最佳化，可在您將佈建叢集快照遷移至無伺服器命名空間時智慧地判斷初始容量。
- 已修正在取用者上快取的身分字符過期時，使用 IAM Identity Center 身分傳播的資料共用查詢可能會失敗的問題。
- 修正 IAM Identity Center 群組同步可能導致後續背景維護操作失敗的問題。
- 修正使用不同的應用程式 ARN 更新 IAM Identity Center 身分提供者組態可能會導致叢集重新啟動的問題。
- 修正使用 IAM Identity Center 身分傳播的資料共用查詢在特定 AWS 區域中可能會失敗的問題。

- 透過確保一致傳播身分提供者組態變更，改善 IAM Identity Center 整合可靠性。
- 透過將一些關聯的 NOT EXISTS 子查詢裝飾為反聯結來最佳化查詢計畫。
- 修正了當目標資料表不存在時，Apache Iceberg 資料表上的 DROP TABLE IF EXISTS 可能不正確地報告查詢失敗而非成功的問題。
- 當使用者嘗試在目錄資料表上執行 ANALYZE 時，新增適當的錯誤訊息。
- Spectrum COPY 現已正式推出，移除 Spectrum 型 COPY 操作暫時資料表的相依性。
- 當持久的持久性任務無法跟上速度時，透過調節 S3 通知佇列讀取來改善 Auto COPY 可靠性，從而減少對 S3 列出操作協調 COPY JOB 狀態的需求。
- 並行擴展現在通常可用於零 ETL 資料表的讀取和寫入操作。
- 修正零 ETL CDC 複寫中的競爭條件，當並行磁碟遞交干擾未遞交的資料表狀態追蹤時，可能導致叢集重新啟動。
- 透過使用資料表和用戶端狀態的最新擷取位置來篩選已套用的資料，改善零 ETL CDC 擷取可靠性。
- 修正零 ETL 資料表重新同步中的資料可靠性問題，其中舊的、已套用的檢查點可以在資料表失敗後重新擷取，方法是在重新同步生命週期中正確追蹤檢查點下限和重新同步狀態。
- 已改善具有更明確失敗原因的零 ETL 錯誤訊息，包括超過資料欄長度的詳細資訊和無效的 UTF-8 字元錯誤。
- 增強了 SYS_LUDF_DETAIL 系統檢視，以微秒為單位報告持續時間，而不是毫秒，以提高可觀測性。
- 修正在零 ETL 資料庫中錯誤允許自動 COPY 任務建立的問題。
- 透過將其他內建函數分類為 RLS 安全，為先前遇到效能降低的常見查詢模式提供述詞下推，改善資料列層級安全性 (RLS) 保護資料表的查詢效能。
- 新增對 Apache Iceberg 資料表上 DELETE、UPDATE 和 MERGE 操作的支援。
- 新增對位置無關超級區塊的支援，無論資料量為何，都能啟用可預測的叢集重新啟動時間。
- 啟用 GUID 型磁碟快取，移除超級區塊位置上磁碟快取的相依性，並啟用可預測的叢集重新啟動時間，無論資料磁碟區為何。
- 改善了自動資料表分佈最佳化：具有單一主索引鍵的資料表現在有資格從 ALL 分佈自動轉換為 KEY 分佈，從而改善查詢效能並減少儲存體額外負荷。
- 修正清空和並行 DML 操作存取相同的儲存物件時，VACUUM 操作可能因衝突而失敗或產生不正確結果的問題。
- 修正處理特殊分割區鏈時，VACUUM FULL 可能因記憶體存取out-of-bounds而在特定資料表上失敗的問題。
- 修正 ALTER TABLE OWNER 操作可能不正確地允許變更物件擁有權給內部引導使用者的問題。

- 已修正並行 VACUUM 操作和臨時資料表重複使用可能導致叢集不穩定的競爭條件。
- 修正當處理過的分割區數量超過內部限制時，大型資料表上的 VACUUM FULL 或 VACUUM SORT 可能會失敗並出現內部錯誤的問題。
- 修正從遠端叢集彙總統計資料時捨棄資料表時，背景分割區統計資料收集可能會失敗的問題。
- 透過避免不必要的資料重新整理，減少對排序資料表進行清空所需的時間和資源，改善 VACUUM SORT 操作的效能。
- 修正清空操作期間不正確的內部分割區資料存取可能導致非預期錯誤或叢集不穩定的問題。
- 修正 ALTER TABLE 操作可能會錯誤地將資料表擁有權變更為非特殊權限使用者的問題。擁有者在 ALTER 操作期間所做的變更現在已正確限制為僅限授權使用者。
- 已修正在清空操作期間進行內部限制檢查時，可能會對具有大量分割區的資料表造成不必要的查詢失敗的罕見問題。
- 修正以下問題：背景統計資料收集可能會在遇到已刪除的資料表時失敗或產生錯誤，從而改善叢集穩定性。
- 修正清空操作可能使用過時資料表中繼資料的問題，在清空具有並行結構描述變更的資料表時，可能會導致不正確的行為。
- 發佈了新的查詢編譯增強功能，以提高 CURRENT 維護軌道上 Amazon Redshift 修補程式 P198 中新查詢（第一次執行查詢）的效能。現在，此功能也會在 TRAILING 賽道上從 Amazon Redshift 修補程式 P200 開始的所有版本上啟用。

Amazon Redshift 修補程式 199

此修補程式中的叢集版本：

- 1.0.266409 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 4 月 20 日發行
- 1.0.252420 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 3 月 25 日發行
- 1.0.232773 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 3 月 5 日發行

此修補程式中的新功能和改善

- 發佈了新的查詢編譯增強功能，以提高 Redshift CURRENT 維護軌道中新查詢（第一次執行查詢）的效能。TRAILING 追蹤將在即將發行的修補程式版本中遵循。

- 為生產者上的新承授者啟用了使用 Redshift 聯合許可的消費者對資料欄層級授予的支援。
- Amazon Redshift 聯合許可支援透過延遲繫結檢視 (LBVs) 查詢外部資料表。
- ALTER EXTERNAL SCHEMA 現在可與 KINESIS 外部結構描述類型搭配使用，以變更現有外部結構描述的 IAM_ROLE 和 REGION。
- 新增對 Redshift RA3 和無伺服器資料倉儲上 NO-BACKUP 資料表的還原支援。由於這些執行個體類型不支援 NO-BACKUP 功能，因此標記為 NO-BACKUP 的資料表一律會包含在快照中，現在可正確還原為永久資料表
- Amazon Redshift 現在支援在使用 IAM 角色或 IAM 使用者連線時寫入外部資料表。先前，IAM 角色型身分驗證僅支援外部資料表上的讀取操作。在此更新中，Redshift 會在將資料寫入外部資料表的 Amazon S3 時正確使用您的 IAM 工作階段憑證，以啟用完整的讀取和寫入存取。
- 對於自動重新整理 Kafka 串流 MVs，我們現在使用更有效率的啟發式來判斷是否有要擷取的新記錄。這將減少從 Kafka 代理程式傳輸以進行此檢查的資料量，並減少啟動後續 MV 重新整理時的延遲。
- 升級備份和還原操作以使用與 FIPS 相容的 SHA384 加密，取代舊版 MD5 雜湊
- 針對結合 Lambda 使用者定義函數和 UNNEST 陳述式的查詢增強效能
- 針對系統資料表和檢視使用 Lambda 使用者定義函數的查詢增強效能
- 透過消除不必要的資料掃描，提高使用 ORDER BY 和 LIMIT 子句時的查詢執行效能
- 增強 Redshift Serverless AI 驅動的擴展和最佳化，具有廣泛的支援 RPUs，現在支援至少 8 RPUs
- Amazon Redshift 具體化視觀表 (MVs) 的 Auto REFRESH 查詢現在視為使用者查詢，而不是背景自律程序。Auto REFRESH 查詢現在會以與其他使用者查詢相同的優先順序執行，且不再由自律背景程序延遲。MV Auto REFRESH 行為變更功能僅適用於修補程式版本 P198 及更新版本的 CURRENT 軌道上的 Amazon Redshift 佈建叢集。它目前在 Serverless 上已停用。
- 改善自動任務的排程，以更優先考慮使用者關鍵任務，例如串流擷取、自動複製 (COPY JOB) 和具體化檢視自動重新整理。
- 修正 pg_last_query_id() 錯誤地傳回重寫查詢 ID 而非多可用區域叢集的使用者查詢 ID 的問題
- 修正 SHOW GRANTS 無法處理單一部分物件參考的問題。
- 修正 DROP TABLE 因同時更新 pg_class_extended 而與目錄重建工作者的時間戳記自動工作者衝突的問題，方法是在 DROP TABLE 期間略過刪除，並允許時間戳記工作者在下次執行時清除。
- 修正資料庫、結構描述或資料表名稱包含/排除關鍵字時的零 ETL 通道篩選條件剖析問題
- 已修正清空操作與零 ETL 資料表操作同時執行時可能發生的競爭條件
- 修正 KMS 加密內容變更時，零 ETL 整合未正確嘗試重新同步操作而非轉換為錯誤狀態的問題
- 修正在資料複寫作用中時，叢集重新啟動期間可能發生的零 ETL 整合競爭條件

- 修正 SHOW TABLES 命令在資料表擁有者遭到捨棄時傳回錯誤的問題

Amazon Redshift 修補程式 198

此修補程式中的叢集版本：

- 1.0.239402 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 3 月 18 日發行
- 1.0.227967 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 2 月 25 日發行
- 1.0.224252 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 2 月 19 日發行
- 1.0.219674 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 2 月 11 日發行
- 1.0.212832 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 2 月 10 日發行
- 1.0.204436 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 2 月 4 日發行

此修補程式中的新功能和改善

- 發佈了新的查詢編譯增強功能，以提高 Redshift CURRENT 維護軌道中新查詢 （第一次執行查詢）的效能。TRAILING 追蹤將在即將發行的修補程式版本中遵循。
- 在比較資料欄與常數值時套用適當的定序規則，以改善 INTERSECT 和 EXCEPT 查詢。
- 允許自動重新整理手動建立的具體化視觀表，以便在並行擴展叢集上執行。
- 透過啟用主要叢集與並行擴展叢集之間的直接連線、減少延遲並消除連線輸送量限制，來改善連線效能。
- 改善零 ETL 整合的錯誤分類，讓您更輕鬆地識別根本原因和偵錯問題。
- 從 SYS_QUERY_HISTORY 移除錯誤訊息，以成功執行查詢，提供更乾淨的查詢歷史記錄。
- 改善 SYS_QUERY_HISTORY 中爆量查詢的編譯時間記錄準確性，提供更好的效能洞察。
- 新增對陣列函數的支援：ARRAY_CONTAINS、ARRAY_POSITION、ARRAYS_OVERLAP、ARRAY_INTERSECTION、ARRAY_S和 ARRAY_POSITIONS。

- 新增 GET_NUMBER_ATTRIBUTES 函數的支援，該函數會傳回 SUPER 物件中的屬性數目。
- 啟用失敗交易的閒置交易逾時，以改善資源管理。
- 新增對 TEMPLATE 物件類型的支援，可讓您定義可重複使用的格式參數，無需為每個操作手動指定這些參數。
- 改善在通用資料表表達式 (CTE) 上調用 Lambda 使用者定義函數 (LUDFs) 的查詢效能。 CTEs
- 在 CASE 表達式的 THEN 子句中，改善叫用 Lambda 使用者定義函數 (LUDFs) 的查詢效能。
- 新 Python 使用者定義函數的建立已棄用。現有的 Python UDFs將支援至 2026 年 6 月 30 日。遷移至 Lambda 使用者定義函數 (LUDFs) 以獲得持續支援。
- 使用新的系統資料表 SYS_LUDFs) 的可觀測性。此資料表會記錄查詢中使用的 LUDFs資訊和指標。
- Amazon Redshift 具體化視觀表 (MVs) 的 Auto REFRESH 查詢現在視為使用者查詢，而不是背景自律程序。Auto REFRESH 查詢現在會以與其他使用者查詢相同的優先順序執行，且不再由自律背景程序延遲。MV Auto REFRESH 行為變更功能僅適用於修補程式版本 P198 及更新版本的 CURRENT 軌道上的 Amazon Redshift 佈建叢集。它目前在 Serverless 上已停用。

Amazon Redshift 修補程式 197

此修補程式中的叢集版本：

- 1.0.237947 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 3 月 10 日發行
- 1.0.198462 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 1 月 28 日發行
- 1.0.198462 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 1 月 21 日發行
- 1.0.194394 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 1 月 14 日發行
- 1.0.187928 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 1 月 9 日發行
- 1.0.179517 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 12 月 16 日發行
- 1.0.176066 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 12 月 9 日發行
- 1.0.166219 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 11 月 21 日發行

此修補程式中的新功能和改善

- Autonomics 現在會考慮整個網格的工作負載，以最佳化清空、分析和 ATO。
- 從多個 Amazon Redshift 資料倉儲建立和重新整理具體化檢視。
- 新增支援在 SUPER 資料類型中存放最多 16，000，000 個位元組的字串常值
- 新增 GET_NUMBER_ATTRIBUTES(super_object) 函數，其會傳回 SUPER 物件中包含的屬性總數。
- 資料表還原操作會強制執行版本修補程式相容性限制。嘗試還原資料表時，只能在執行與建立備份時相同的修補程式版本的叢集上使用備份，比備份低一個修補程式版本，或比備份高的任何修補程式版本。
- 增強 LIKE 運算子，以便在使用 CHAR 資料類型時正確處理模式的結尾空格。
- 引進了在儲存格階層上運作的 4 個新空間 H3 函數：H3_Resolution、H3_ToParent、H3_ToChildren 和 H3_IsValid。
- 啟用 Amazon Redshift 聯合許可，可讓您定義一次資料許可，並自動在 中的所有倉儲中強制執行它們，以簡化跨多個 Redshift 資料倉儲的許可管理 AWS 帳戶。
- 在 Amazon Redshift Federated Permissions Catalog 物件上啟用物件層級授予、精細的資料欄層級存取控制、範圍許可授予和資料庫層級授予。
- 在 Amazon Redshift 聯合許可目錄上啟用 CREATE、ALTER、ATTACH、DETACH 和 DROP RLS 政策操作。
- 允許在 Amazon Redshift 聯合許可目錄中開啟或關閉關係的資料列層級安全性。
- 在 Amazon Redshift 聯合許可目錄物件上啟用 CREATE、ALTER、ATTACH、DETACH 和 DROP 遮罩政策操作。
- 讓 SHOW POLICIES 在 Amazon Redshift Federated Permissions Catalog 物件中顯示連線資料庫和關聯上的 RLS 和遮罩政策。
- 讓資料庫使用者擔任 IAM 角色，以查詢 Amazon Redshift 聯合許可目錄物件。
- 增強的 SHOW GRANTS 命令，可支援相同 Redshift 叢集中的跨資料庫授予可見性。
- 新增對 SHOW COLUMN GRANTS 命令的支援，以簡化資料欄層級存取控制的探索。
- 新增對 SHOW PROCEDURES 命令的支援，以簡化預存程序中繼資料探索的探索。
- 新增對 SHOW FUNCTIONS 命令的支援，以簡化使用者定義函數中繼資料的探索。
- 推出 SHOW PARAMETERS 命令，以顯示函數和預存程序的參數中繼資料。
- 引進 SHOW CONSTRAINTS 命令，以顯示函數和預存程序的限制中繼資料。

- 新增對 Amazon Redshift 聯合許可目錄和相同 Redshift 叢集中跨資料庫的 CREATE/CREATE OR REPLACE/DROP VIEW 操作的支援。
- 改善某些 INTERSECT 或 EXCEPT 查詢中的常數字串定序處理。
- 增強的 SHOW TABLES 可在缺少 FAS 登入資料進行 AWS Glue Data Catalog 存取時顯示警告訊息，而不是失敗。
- 增強的 SHOW COLUMNS 命令具有額外的中繼資料資料欄，用於排序索引鍵、分佈樣式、編碼和定序資訊。
- 新增 grantor_name 欄至 SHOW GRANTS 輸出，並在所有 SHOW GRANTS 結果集中包含標準化的 database_name。
- 具有額外資料欄的增強型 SHOW TABLES 命令：table_owner、last_altered_time、last_modified_time、dist_style 和 table_sub_type。
- Identity Center 身分驗證支援：Amazon Redshift 現在支援透過新的 `GetIdentityCenterAuthToken` API 操作進行 IAM Identity Center 身分驗證。此功能可讓您使用現有的 Identity Center 身分，聯合存取 Amazon Redshift 叢集和無伺服器工作群組。主要優點包括使用 Identity Center 登入資料進行單一登入存取、使用加密身分驗證字符增強安全性、全面稽核記錄以確保合規性，以及與現有 Identity Center 設定的無縫整合。API 操作會產生範圍限定於特定叢集識別符或工作群組名稱的身分驗證字符。要求包括具有內嵌 Identity Center 身分資訊的身分增強憑證，以及 API 存取的適當 IAM 許可。

Amazon Redshift 修補程式 196

此修補程式中的叢集版本：

- 1.0.179872 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2026 年 1 月 8 日發行
- 1.0.163480 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 11 月 20 日發行
- 1.0.160807 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 11 月 14 日發行
- 1.0.155905 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 11 月 11 日發行

此修補程式中的新功能和改善

- 透過將關聯 IN 和 EXISTS 子查詢裝飾為半聯結，最佳化關聯 IN 和 EXISTS 子查詢的計劃。
- 延伸 SHOW TABLE 命令，以包含每個資料欄的定序資訊。

Amazon Redshift 修補程式 195

此修補程式中的叢集版本：

- 1.0.172541 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 12 月 8 日發行
- 1.0.162991 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 11 月 20 日發行
- 1.0.151179 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 11 月 4 日發行
- 1.0.148180 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 10 月 31 日發行
- 1.0.142459 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 10 月 24 日發行

此修補程式中的新功能和改善

- 新增使用 CREATE TABLE 、CREATE TABLE AS SELECT、INSERT INTO、SQL 命令寫入 Iceberg 資料表的支援。
- 在具有不區分大小寫定序的資料庫中新增對 SUPER 資料類型的支援。
- 將 IANA 時區資料庫版本更新至 2025b。
- 新增對 TZDB_VERSION 函式的支援。此函式會顯示目前使用的 IANA 時區資料庫版本。
- Amazon Redshift 現在會強制執行修補程式相容性限制。在修補程式版本上建立備份時，若修補程式版本比目前叢集的修補程式還新一版以上，則不再允許該備份執行資料表還原操作。
- 延伸 UNNEST 功能以支援 JOIN 表達式。先前 UNNEST 僅限於單一資料表參考。
- 新增對 JIT (即時) ANALYZE 功能的支援。JIT ANALYZE 會自動、智慧且有效率地收集 Amazon Redshift 查詢最佳化工具產生高效率查詢執行計畫所需的最低限度統計資料，讓客戶能夠為 Apache Iceberg out-of-box(OOB) 查詢和工作負載提供最佳價格效能。

Amazon Redshift 修補程式 194

此修補程式中的叢集版本：

- 1.0.148169 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 11 月 5 日發行
- 1.0.138443 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 10 月 18 日發行
- 1.0.129791 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 10 月 8 日發行

此修補程式中的新功能和改善

- Amazon Redshift 將不再讓存取其中 `AuthorizedColumns` 欄位包含反斜線的 AWS Lake Formation管控關係的查詢失敗。
- 新增了針對 SUPER 欄支援 COPY 命令中的 NULL AS 選項，可讓您指定自訂字串來代表 NULL 值。
- 新增了搭配日期和時間戳記使用時，EXTRACT 和 DATE_PART 函式中 ISOWEEK 和 ISOYEAR 日期部分的支援。
- 修正了群組變數上的 PartiQL 導覽，以確保正確剖析和評估 GROUP BY 查詢中的巢狀欄位存取。
- 新增了使用 SHOW CONSTRAINTS 命令檢視主索引鍵和外部索引鍵限制的支援。
- 新增了從資料共用具體化視觀表建立的本機具體化視觀表的支援。
- 改善從不同資料表擷取資料並在聯結條件下調用 Lambda 使用者定義函式的查詢效能。
- 在資料共用 MVs上新增對 MVs支援
- DROP DATABASE 命令現在使用背景工作處理程序，在認可之後移除領導節點目錄的目錄。使用工作處理程序可避免發生與目錄的目錄清除相關的少見錯誤。

Amazon Redshift 修補程式 193

此修補程式中的叢集版本：

- 1.0.136890 – TRAILING Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 10 月 17 日發行

- 1.0.127211 – CURRENT Track Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 – 已於 2025 年 9 月 30 日發行

此修補程式中的新功能和改善

- 新增了支援使用 CREATE TABLE LIKE 建立資料表，以從遠端資料庫複製一個上層資料表 (同時包括在相同叢集內和不同叢集之間)。
- 新增了 ST_GeomFromGeoJSON 過載，其會從 SUPER 資料類型建立 GeoJSON 格式的幾何，並支援最高 1 MB 的幾何。
- 變更了預設後援選項，以防自動工作負載管理 (WLM) 發生組態設定錯誤。預設後援選項現在是自動 WLM，而不是手動 WLM。
- 採用最新的 IANA 時區資料庫修補程式更新了時區計算。此變更會更改特定時區和時段的日期和時間轉換運作方式。如需詳細資訊，請參閱 Amazon Redshift 使用最新的 IANA 時區資料庫。
- 新增了對多維資料佈局 (MDDL) 排序的支援。MDDL 排序索引鍵會根據實際查詢篩選條件動態排序資料，以加速查詢效能。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的多維資料佈局排序。

Amazon Redshift 修補程式 192

此修補程式中的叢集版本：

- 1.0.124422 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 9 月 22 日發行
- 1.0.122914 - 後續 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 9 月 16 日發行
- 1.0.121035 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 8 月 22 日發行
- 1.0.119650 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 8 月 7 日發行
- 1.0.118447 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 7 月 24 日發行

此修補程式中的新功能和改善

- 修正在定序的不區分大小寫欄和常數字串欄上的 JOIN 選取不相符的雜湊函式的問題。

Amazon Redshift 修補程式 191

此修補程式中的叢集版本：

- 1.0.117891 - 後續 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 7 月 29 日發行
- 1.0.117891 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 7 月 22 日發行
- 1.0.117458 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 7 月 15 日發行
- 1.0.116415 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 6 月 26 日發行
- 1.0.116263 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 6 月 25 日發行
- 1.0.115865 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 6 月 18 日發行

此修補程式中的新功能和改善

- 新增在 SELECT 陳述式的 FROM 子句中支援 UNNEST 操作，讓客戶能夠將陣列元素轉換為個別列。
- 改善在一般資料表表達式 (CTE) 和遞迴 CTE 內使用 SUPER 資料類型時 PartiQL 解除巢狀化語法的錯誤處理，特別是在解析有相同名稱的資料表別名時。
- 透過支援大型資料表的增量自動分析，改善自動分析的效能。
- Amazon Redshift 現在支援巢狀具體化視觀表 (MV) 的階層式重新整理。
- Amazon Redshift 現在支援自動重新整理 Apache Iceberg 資料表上定義的具體化視觀表 (MV)。
- 新增 Amazon RDS for Oracle 與 Amazon Redshift 的零 ETL 整合支援。
- 新增 Amazon RDS for PostgreSQL 與 Amazon Redshift 的零 ETL 整合支援。

Amazon Redshift 修補程式 190

此修補程式中的叢集版本：

- 1.0.115140 - 後續 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 6 月 16 日發行
- 1.0.115564 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 6 月 14 日發行
- 1.0.113436 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 5 月 15 日發行
- 1.0.112895 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 5 月 7 日發行

此修補程式中的新功能和改善

- 新增改良的模型，用於 Amazon Redshift Serverless AI 驅動的擴展和最佳化。
- 新增在啟用增強型 VPC 路由的情況下使用 Amazon Redshift 時，支援自動複製。
- 新增支援在不同資料表之間並行執行多個自動資料表最佳化操作。
- 修正串流擷取中，排入佇列的自動重新整理查詢會觸發同樣排入佇列的 UNDO 查詢，導致鎖定爭用的少見情況。
- 新增支援與 Amazon Redshift 的 Oracle Database@AWS 零 ETL 整合。

Amazon Redshift 修補程式 189

此修補程式中的叢集版本：

- 1.0.112790 - 後續 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 5 月 14 日發行
- 1.0.112790 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 5 月 6 日發行
- 1.0.112086 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 4 月 29 日發行
- 1.0.111040 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 4 月 16 日發行

- 1.0.109768 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 3 月 22 日發行
- 1.0.109284 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 3 月 13 日發行
- 1.0.108971 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 3 月 7 日發行

此修補程式中的新功能和改善

- 新增支援使用 SUPER、GEOMETRY 和 GEOGRAPHY 資料類型的寫入查詢進行並行擴展。
- 修正在使用 CREATE TABLE AS 建立資料表期間，允許 SUPER、GEOMETRY 或 GEOGRAPHY 欄設定為 DISTKEY 或 SORTKEY 欄的問題。
- 新增 TRY_CAST 函式的支援。
- 可支援在不同資料表中並行執行多個清空命令。

Amazon Redshift 修補程式 188

此修補程式中的叢集版本：

- 1.0.109616 - 後續 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 3 月 27 日發行
- 1.0.109616 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 3 月 24 日發行
- 1.0.108470 - 後續 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 3 月 13 日發行
- 1.0.108470 - 目前 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 3 月 11 日發行
- 1.0.108950 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 3 月 10 日發行
- 1.0.108790 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 3 月 9 日發行
- 1.0.108470 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 3 月 4 日發行
- 1.0.107910 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 2 月 20 日發行
- 1.0.107360 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 2 月 13 日發行

- 1.0.106767 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 2 月 5 日發行

此修補程式中的新功能和改善

- 新增支援自動掛載 Amazon S3 Tables 目錄，讓您更容易在 Amazon S3 Tables 中管理的 Apache Iceberg 資料表上執行查詢。請參閱《AWS Lake Formation 開發人員指南》中的在 AWS Glue Data Catalog中管理 Amazon Redshift 命名空間的先決條件，以了解所需的許可。
- 您現在可以略過資料遮罩檢查，並將附加動態資料遮罩 (DDM) 的資料表新增至資料共用。
- 不允許使用略過查詢區塊的相互關聯參考，也稱為「略過層級相互關聯參考」。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的不支援的相互關聯子查詢模式。
- 新增支援使用資料共用和 3 部分標記法的相互關聯 SUPER 資料類型查詢。
- 新增 EXCLUDE SQL 關鍵字支援。
- 新增新的 GROUP BY ALL SQL 關鍵字支援。
- 改善使用角色設定的工作負載管理 (WLM) 佇列的效能，因而減少鎖定爭用。
- 新增在開啟歷史記錄模式的零 ETL 資料表上支援自動重新整理具體化視觀表。
- 新增支援零 ETL 複寫期間字串中內嵌的 null 字元。
- 改良 AI 驅動擴展和最佳化的模型。
- 將 `username` 欄新增至 `sys_query_history` 檢視。

Amazon Redshift 修補程式 187

此修補程式中的叢集版本：

- 1.0.107351 - Amazon Redshift 佈建叢集後續追蹤版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 2 月 13 日發行
- 1.0.106452 - Amazon Redshift 佈建叢集後續追蹤版本 - 已於 2025 年 2 月 5 日發行
- 1.0.106980 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 2 月 3 日發行
- 1.0.106452 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 1 月 24 日發行
- 1.0.106073 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 1 月 21 日發行

- 1.0.105722 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 1 月 10 日發行
- 1.0.105373 - 目前 Amazon Redshift 佈建叢集版本和 Amazon Redshift Serverless 工作群組版本 - 已於 2025 年 1 月 8 日發行
- 1.0.104930 - Amazon Redshift Serverless 工作群組版本 - 已於 2024 年 12 月 20 日發行

此修補程式中的新功能和改善

- 引進兩個新的空間 H3 函數 (H3_center 和 H3_boundary)。
- 在並行擴展叢集上啟用動態磁碟快取。
- 修正在擷取可能影響效能的排序資料表期間，記憶體用量效率不佳的情形。
- 修正 SELECT 子句內有子查詢的特定查詢 (例如 `SELECT ARRAY_FLATTEN((SELECT FROM ...))`) 在包含單一 SUPER 值的特定情況下不當觸發 `size >= min_partiql_size` XCHECK 錯誤的問題。
- 修正不可為 Null 的 SUPER 表達式 (`json_serialize()`、`json_size()`、`json_typeof()`、`is_object()` 和其他數種) 與 `CASE ... END` 或 `COALESCE()` 這類引數表達式結合時，有時會產生錯誤結果的問題。
- 修正在檢視上啟用資料列層級安全後，近期繫結檢視中的 SELECT 可能會引發 ERROR 的問題。
- 修正零 ETL 整合上的 CDC 導致領導節點的 CPU 耗用量高的問題。
- 新增功能至零 ETL 整合，以查詢所有狀態的資料表，包括更新期間。
- 新增功能至零 ETL 整合，以將映射的大型文字/字串截斷至 Amazon Redshift 上的 varchar 大小上限。
- 新增功能至零 ETL 整合，以將無效的 UTF-8 字元取代為您選擇的指定字元。
- 修正 Amazon RDS for MySQL 的零 ETL 整合的 'refresh interval' 問題。
- 新增針對 Aurora PostgreSQL 的零 ETL 整合支援 JSON/JSONB 資料類型。
- 修正查詢包含的表達式具有 `IS [NOT] {TRUE|FALSE|UNKNOWN}` 時，會遇到聲明錯誤的問題。
- 修正以下問題：在存取不存在的陣列元素或屬性、JSON `'null'` 值或空 JSON 字串時，`json_extract_path_text()` 和 `json_extract_array_element_text()` 函式會產生空字串 `''`，而不是 NULL (反之亦然)。
- 透過允許 INSERT/COPY 陳述式共用鎖定並繼續進行，直到其需要將資料寫入資料表的方式，改善寫入相同資料表之並行交易中上述陳述式的效能。

Amazon Redshift 修補程式 186

此修補程式中的叢集版本：

- 1.0.82096 - 後續追蹤版本 - 已於 2025 年 1 月 10 日發行
- 1.0.82000 - Amazon Redshift Serverless 版本 - 已於 2025 年 1 月 10 日發行
- 1.0.81981 - 目前追蹤版本 - 已於 2025 年 1 月 10 日發行
- 1.0.81475 - 後續追蹤版本 - 已於 2025 年 1 月 6 日發行
- 1.0.81473 - Amazon Redshift Serverless 版本 - 已於 2025 年 1 月 6 日發行
- 1.0.81462 - 目前追蹤版本 - 已於 2025 年 1 月 6 日發行
- 1.0.80643 - 後續追蹤版本 - 已於 2024 年 12 月 17 日發行
- 1.0.80583 - Amazon Redshift Serverless 版本 - 已於 2024 年 12 月 17 日發行
- 1.0.80560 - 目前追蹤版本 - 已於 2024 年 12 月 17 日發行
- 1.0.80498 - Amazon Redshift Serverless 版本 - 已於 2024 年 12 月 13 日發行
- 1.0.80491 - 目前追蹤版本 - 已於 2024 年 12 月 13 日發行
- 1.0.80036 - Amazon Redshift Serverless 版本 - 已於 2024 年 12 月 6 日發行
- 1.0.80009 - 目前追蹤版本 - 已於 2024 年 12 月 6 日發行
- 1.0.79372 - Amazon Redshift Serverless 版本 - 已於 2024 年 11 月 26 日發行
- 1.0.79237 - Amazon Redshift Serverless 版本 - 已於 2024 年 11 月 24 日發行
- 1.0.79229 - 目前追蹤版本 - 已於 2024 年 11 月 24 日發行
- 1.0.79003 - Amazon Redshift Serverless 版本 - 已於 2024 年 11 月 19 日發行
- 1.0.78987 - 目前追蹤版本 - 已於 2024 年 11 月 19 日發行
- 1.0.78890 - Amazon Redshift Serverless 版本 - 已於 2024 年 11 月 18 日發行
- 1.0.78881 - 目前追蹤版本 - 已於 2024 年 11 月 18 日發行
- 1.0.78646 - Amazon Redshift Serverless 版本 - 已於 2024 年 11 月 14 日發行
- 1.0.78641 - 目前追蹤版本 - 已於 2024 年 11 月 14 日發行
- 1.0.78178 - Amazon Redshift Serverless 版本 - 已於 2024 年 11 月 12 日發行
- 1.0.78160 - 目前追蹤版本 - 已於 2024 年 11 月 12 日發行
- 1.0.77809 - Amazon Redshift Serverless 版本 - 已於 2024 年 10 月 31 日發行
- 1.0.77777 - 目前追蹤版本 - 已於 2024 年 10 月 31 日發行

- 1.0.77292 - Amazon Redshift Serverless 版本 - 已於 2024 年 10 月 24 日發行
- 1.0.77272 - 目前追蹤版本 - 已於 2024 年 10 月 24 日發行
- 1.0.77040 - Amazon Redshift Serverless 版本 - 已於 2024 年 10 月 22 日發行
- 1.0.77028 - 目前追蹤版本 - 已於 2024 年 10 月 22 日發行

此修補程式中的新功能和改善

- 新增在 Amazon Aurora MySQL、Amazon Aurora PostgreSQL、Amazon RDS for MySQL 和 Amazon DynamoDB 的零 ETL 整合中，支援於資料表上自動和增量重新整理具體化視觀表。
- 改善 Amazon Redshift 使用相互關聯單列子查詢重寫查詢的方式，例如 `SELECT … WHERE key = (SELECT … correlated subquery)`。請注意，只有在子查詢產生單一列時，此類查詢才有效。由於重寫獲得改善，某些違反此條件的查詢現在可能會因為 ERROR：單列子查詢傳回超過一列而失敗，即使先前可允許。為了避免這種情況，這類子查詢可能需要更正，以保證其傳回單一列，例如藉由加上 MIN() 或 MAX() 彙總的方式。
- 在 Amazon Redshift 中新增 SQL 識別碼 'KAFKA'，以支援使用直接串流擷取從外部 Kafka 來源串流至 Amazon Redshift。這些來源包括 Confluent Managed Cloud 和 Apache Kafka 等外部 Kafka 來源。
- 新增使用資料共用的多倉儲寫入支援，如此您就能根據您的工作負載需求使用各種類型和大小的不同倉儲，來擴展您的寫入工作負載，並提升擷取、轉換和載入 (ETL) 工作負載的效能。
- 您現在可以使用 Amazon Redshift 在 Amazon S3 Tables 中管理的 Apache Iceberg 資料表上執行 SELECT 查詢。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的使用 Amazon Redshift 存取 Amazon S3 Tables。
- 新增在 Amazon Redshift 中執行跨資料庫寫入查詢的支援，如此您就能在 Amazon Redshift 叢集的多個資料庫中寫入。如需詳細資訊，請參閱跨資料庫查詢。

Amazon Redshift 修補程式 185

此修補程式中的叢集版本：

- 1.0.79878 - 後續追蹤版本 - 已於 2024 年 12 月 3 日發行
- 1.0.79868 - Amazon Redshift Serverless 版本 - 已於 2024 年 12 月 3 日發行
- 1.0.79845 - 目前追蹤版本 - 已於 2024 年 12 月 3 日發行
- 1.0.78946 - 後續追蹤版本 - 已於 2024 年 11 月 19 日發行
- 1.0.78354 - 後續追蹤版本 - 已於 2024 年 11 月 12 日發行

- 1.0.78130 - Amazon Redshift Serverless 版本 - 已於 2024 年 11 月 12 日發行
- 1.0.78125 - 目前追蹤版本 - 已於 2024 年 11 月 12 日發行
- 1.0.78016 - Amazon Redshift Serverless 版本 - 已於 2024 年 11 月 5 日發行
- 1.0.78014 - 目前追蹤版本 - 已於 2024 年 11 月 5 日發行
- 1.0.77707 - Amazon Redshift Serverless 版本 - 已於 2024 年 11 月 4 日發行
- 1.0.77687 - 目前追蹤版本 - 已於 2024 年 11 月 4 日發行
- 1.0.77467 - Amazon Redshift Serverless 版本 - 已於 2024 年 11 月 4 日發行
- 1.0.77433 - 目前追蹤版本 - 已於 2024 年 11 月 4 日發行
- 1.0.77467 - Amazon Redshift Serverless 版本 - 已於 2024 年 10 月 29 日發行
- 1.0.77433 - 目前追蹤版本 - 已於 2024 年 10 月 29 日發行
- 1.0.76991 - Amazon Redshift Serverless 版本 - 已於 2024 年 10 月 21 日發行
- 1.0.76913 - 目前追蹤版本 - 已於 2024 年 10 月 21 日發行
- 1.0.76645 - Amazon Redshift Serverless 版本 - 已於 2024 年 10 月 14 日發行
- 1.0.76642 - 目前追蹤版本 - 已於 2024 年 10 月 14 日發行
- 1.0.76242 - Amazon Redshift Serverless 版本 - 已於 2024 年 10 月 10 日發行
- 1.0.76230 - 目前追蹤版本 - 已於 2024 年 10 月 10 日發行

此修補程式中的新功能和改善

- 新增支援增量重新整理資料湖資料表上建立的具體化視觀表 (MV)。
- 導入更改具體化視觀表的分佈索引鍵和排序索引鍵的支援。
- 新增將 Amazon Redshift 機器學習 (ML) 與 Amazon Bedrock 整合的支援，以便利用來自簡單 SQL 命令的大型語言模型 (LLM) 以及 Amazon Redshift 中的資料。
- 修正允許處理 Amazon Redshift 空間資料中空白 shapefile 而未擲回錯誤的問題。
- 修正允許 varbit 和 varbinary 資料類型在零 ETL 整合中擷取 "" 形式的空字串值而非 NULL 的問題。
- 修正結果快取中的競爭條件，此情況會導致零 ETL 整合中跨資料庫查詢傳回過時的結果。
- 最佳化零 ETL 整合重新同步程序，以縮短重新同步時間。
- 改善復原和系統維護操作後的零 ETL 啟動程序時間。現在系統可以在最後一次查詢操作之後復原 CDC，而不需等待最新的可用 CDC 點才能復原。
- 透過新的系統資料表 `sys_integration_table_activity` 改善零 ETL 整合的可觀測性。此資料表會追蹤零 ETL 整合資料表插入、更新和刪除。

- 改善自動建立聯合角色的體驗。Amazon Redshift 管理員現在能夠更充分控制聯合身分使用者登入期間，自動建立的聯合角色。您現在可以針對每個身分提供者啟用、停用、套用篩選條件，以及設定自動建立設定。
- IAM Identity Center 使用者可以使用 S3 存取授權搭配 IAM Identity Center 憑證執行 COPY、UNLOAD 和 CREATE LIBRARY。
- 可從 Amazon S3 持續擷取檔案的自動複製 (COPY JOB) 現在已正式推出。

Amazon Redshift 修補程式 184

此修補程式中的叢集版本：

- 1.0.77706 - 後續追蹤版本 - 已於 2024 年 11 月 12 日發行
- 1.0.76832 - 後續追蹤版本 - 已於 2024 年 10 月 17 日發行
- 1.0.76169 - Amazon Redshift Serverless 版本 - 已於 2024 年 10 月 10 日發行
- 1.0.76142 - 目前追蹤版本 - 已於 2024 年 10 月 10 日發行
- 1.0.75677 - Amazon Redshift Serverless 版本 - 已於 2024 年 9 月 27 日發行
- 1.0.75672 - 目前追蹤版本 - 已於 2024 年 9 月 27 日發行
- 1.0.75504 - Amazon Redshift Serverless 版本 - 已於 2024 年 9 月 23 日發行
- 1.0.75449 - 目前追蹤版本 - 已於 2024 年 9 月 24 日發行
- 1.0.74765 - Amazon Redshift Serverless 版本 - 已於 2024 年 9 月 12 日發行
- 1.0.74754 - 目前追蹤版本 - 已於 2024 年 9 月 12 日發行

此修補程式中的新功能和改善

- Amazon Redshift 用於串流資料擷取的串流具體化視觀表 (MV) 已將 VARBYTE 欄大小從 1,024,000 個位元組增加。Amazon Redshift 現在可以從 Amazon Kinesis Data Streams 擷取最大 1,048,576 個位元組或 1MiB 的記錄。Redshift 可以從 Amazon Managed Streaming for Apache Kafka 擷取最大 16,777,216 個位元組或 16 MiB 的記錄。如果您使用 ATA SQL 命令 `ALTER TABLE <target_tbl> APPEND FROM <streaming_mv>`，請捨棄並重新建立您的 <target_tbl>，以取得對應的較大 VARBYTE 欄大小。
- Amazon Redshift 資料共用現在可以包含 Amazon S3 資料湖資料表和參考檢視， AWS Glue Data Catalog 包括由 Lake Formation 管理的資料表。
- 新增在 DynamoDB 的零 ETL 整合中，支援於資料表上自動和增量重新整理具體化視觀表。

- 新增在零 ETL 整合資料表上設定 "refresh interval" 的支援，以指定 Amazon Redshift 的複寫重新整理速率。您可以在為新整合建立資料庫時進行設定，或修改現有整合的資料庫。
- 在 Amazon Redshift 串流擷取中，針對 Amazon Managed Streaming for Apache Kafka 新增雙向 Transport Layer Security (mTLS) 身分驗證的支援。
- 新增查詢雜湊的支援，這是其以查詢的文字表示及其參數的值為基礎之 SQL 查詢的唯一識別碼。其可用於識別、分組和分析類似的查詢。查詢雜湊現在可在 SYS_QUERY_HISTORY 檢視中找到，且新增了兩個新欄：
  - `user_query_hash` - 使用者提交的雜湊，包括查詢常值。
  - `generic_query_hash` - 使用者提交的雜湊，無任何查詢常值。
- 修正下列少見情況的系統死結：零 ETL 執行 CDC 複寫和掃描，且使用結果快取執行資料表查詢。
- 解決工作負載管理 (WLM) 中的下列問題：當 Python UDF 查詢的 WLM 資源無法使用時，Python 使用者定義函式 (UDF) 查詢會先佔其他查詢。
- 解決工作負載管理 (WLM) 無法將查詢路由至對應新建立使用者角色之佇列的問題。
- 改善查詢大型生產者資料表的較小資料共用取用者的磁碟使用率。
- 改善佈建叢集的 INSERT 陳述式效能，以彈性方式調整為更大的規模。

Amazon Redshift 修補程式 183

此修補程式中的叢集版本：

- 1.0.75655 - 後續追蹤版本 - 已於 2024 年 9 月 30 日發行
- 1.0.75388 - Amazon Redshift Serverless 版本 - 已於 2024 年 9 月 25 日發行
- 1.0.75379 - 目前追蹤版本 - 已於 2024 年 9 月 25 日發行
- 1.0.74967 - Amazon Redshift Serverless 版本 - 已於 2024 年 9 月 17 日發行
- 1.0.74927 - 目前追蹤版本 - 已於 2024 年 9 月 17 日發行
- 1.0.74518 - Amazon Redshift Serverless 版本 - 已於 2024 年 9 月 11 日發行
- 1.0.74503 - 目前追蹤版本 - 已於 2024 年 9 月 11 日發行
- 1.0.74223 - Amazon Redshift Serverless 版本 - 已於 2024 年 9 月 5 日發行
- 1.0.74159 - 目前追蹤版本 - 已於 2024 年 9 月 5 日發行
- 1.0.74126 - Amazon Redshift Serverless 版本 - 已於 2024 年 8 月 30 日發行
- 1.0.74097 - 目前追蹤版本 - 已於 2024 年 8 月 30 日發行

- 1.0.73016 - Amazon Redshift Serverless 版本 - 已於 2024 年 8 月 8 日發行
- 1.0.72982 - 目前追蹤版本 - 已於 2024 年 8 月 8 日發行

此修補程式中的新功能和改善

- 新增透過 SVV_DATABASE_PRIVILEGES 和 SVV_SCHEMA_PRIVILEGES 探索限定範圍許可的支援。同時將欄 privilege_scope 導入 SVV_DATABASE_PRIVILEGES 和 SVV_SCHEMA_PRIVILEGES 中。
- 改善下列情況的查詢效能：在分組欄的不同值 (NDV) 數量較少時執行不同彙總操作。
- 改善佈建資料倉儲的 INSERT/COPY 陳述式效能，將其大小彈性調整為 2 倍以上。
- 支援動態資料遮罩政策內的工作階段內容變數。
- 新增將子查詢和檢視作為 MERGE 陳述式來源的支援。
- 支援預存程序，其中包含佈建並行擴展和無伺服器自動擴展運算的 MERGE 陳述式。
- 針對 COPY 命令和進行大小調整的倉儲，在工作負載管理中透過更精準的資源預測來改善查詢效能。
- 改善可用記憶體受限之叢集中記憶體不足錯誤的復原能力
- 新增支援以非 ASCII 字元作為 COPY 命令的欄位分隔符號。
- 新增支援以使用 COPY 命令擷取以 ISO-8859-1 字元集編碼的資料。
- 如果指定 URI，則不需要在 MSK 外部結構描述定義中指定 CLUSTER_ARN。
- 支援在零 ETL 整合資料表上進行掃描時，套用範圍掃描篩選條件。
- 支援將排序索引鍵新增至零 ETL 整合資料表。
- 在建立零 ETL 整合資料庫時，支援隨 CREATE DATABASE 陳述式指定 `serializable` 和 `collation` 這類資料庫選項。
- 修正了當零 ETL 整合中的資料篩選條件超過 2 KB 時，造成叢集重新啟動的問題。

Amazon Redshift 修補程式 182

此修補程式中的叢集版本：

- 1.0.73589 - 後續追蹤版本 - 已於 2024 年 8 月 22 日發行
- 1.0.73359 - Amazon Redshift Serverless 版本 - 已於 2024 年 8 月 15 日發行
- 1.0.73348 - 目前追蹤版本 - 已於 2024 年 8 月 15 日發行

- 1.0.72917 - Amazon Redshift Serverless 版本 - 已於 2024 年 8 月 12 日發行
- 1.0.72899 - 目前追蹤版本 - 已於 2024 年 8 月 12 日發行
- 1.0.72528 - Amazon Redshift Serverless 版本 - 已於 2024 年 8 月 7 日發行
- 1.0.72503 - 目前追蹤版本 - 已於 2024 年 8 月 8 日發行
- 1.0.72239 - Amazon Redshift Serverless 版本 - 已於 2024 年 8 月 1 日發行
- 1.0.71714 - Amazon Redshift Serverless 版本 - 已於 2024 年 7 月 24 日發行
- 1.0.71629 - 目前追蹤版本 - 已於 2024 年 7 月 24 日發行
- 1.0.70953 - Amazon Redshift Serverless 版本 - 已於 2024 年 7 月 11 日發行
- 1.0.70890 - 目前追蹤版本 - 已於 2024 年 7 月 11 日發行
- 1.0.70716 - Amazon Redshift Serverless 版本 - 已於 2024 年 7 月 8 日發行
- 1.0.70695 - 目前追蹤版本 - 已於 2024 年 7 月 8 日發行
- 1.0.69945 - Amazon Redshift Serverless 版本 - 已於 2024 年 6 月 27 日發行
- 1.0.69938 - 目前追蹤版本 - 已於 2024 年 6 月 27 日發行

此修補程式中的新功能和改善

- 更新 LISTAGG、MEDIAN、PERCENTILE_CONT 和 PERCENTILE_DISC，往後不再需要使用者定義的資料表。參考目錄資料表或不參考任何資料表的查詢也可以使用這些函式。
- 針對高並行工作負載，透過在單一工作階段中整合多個查詢中的暫存資料表，縮短資料共用讀取查詢的查詢規劃時間。
- 提供 Redshift Machine Learning (ML) 與 Amazon SageMaker AI Jumpstart 整合的一般可用性，讓您擁有自己的大型語言模型。
- 推出在 Redshift ML 中支援 SUPER 輸入和輸出資料類型。
- 當目標資料表受到動態資料遮罩政策保護並在 JOIN 子句中參考時，支援搭配 JOIN 子句使用 UPDATE 陳述式。
- 允許在 Redshift 上查詢零 ETL 資料庫，即使在整合後會從來源將其刪除。
- 修正可能導致零 ETL 整合失敗的複寫錯誤。這可讓整合有更強的恢復能力。
- 允許建立零 ETL 整合的使用者以外的使用者在 GRANT 許可之後查詢資料。
- 修正記憶體不足的問題，此問題可能造成叢集在啟用零 ETL 整合的 Amazon Redshift 佈建叢集中重新啟動。

- 可從來源 RDS 多可用區域資料庫叢集建立與 Redshift 的 RDS for MySQL 零 ETL 整合。多可用區域資料庫叢集是 Amazon RDS 的半同步、高可用性部署模式，具有兩個可讀取複本資料庫執行個體。
- 可讓使用者藉由在 Amazon Redshift 串流具體化視觀表與 Amazon MSK 主題建立關聯所需的外部結構描述定義中，指定 Amazon MSK 叢集的代理系統 URI，以便從 Amazon Redshift 串流取用者用戶端連線至 Amazon MSK 叢集。利用此功能就不需要透過網際網路閘道於 Amazon MSK 叢集上呼叫 GetBootStrapBroker API，來取得 Amazon MSK 啟動程序代理系統節點名稱。
- 解決恢復 Amazon Redshift Serverless 執行個體的問題，讓現有的資料庫使用者在 Amazon Redshift 查詢編輯器 v2 中連線至資料庫時，能夠使用 IAM Identity Center 身分驗證方法來恢復無伺服器執行個體。
- 透過移至資料表型分片，最佳化 CDC 複寫並減少 Redshift 運算的資源使用率。
- 在工作負載管理 (WLM) 中使用增強型資源預測，改善查詢效能。
- 修正並行擴展叢集上失敗的查詢，並顯示訊息：ran out of WLM queues for restart (用於重新啟動的 WLM 查詢已用盡)。
- 修正當客戶嘗試套用無效的 WLM 組態時，Amazon Redshift 恢復為手動 WLM 的工作負載管理 (WLM) 問題。
- 即使生產者因預定維護或非預定中斷而停機，仍允許資料共用取用者執行讀取查詢。
- 修正罕見的叢集重新啟動問題，此問題會在 ANY_VALUE 函數於彙總資料的查詢中使用時發生，例如 COUNT(DISTINCT) 彙總函數。

Amazon Redshift 修補程式 181

此修補程式中的叢集版本：

- 1.0.72031 - 目前追蹤版本 - 已於 2024 年 8 月 1 日發行
- 1.0.71912 - 後續追蹤版本 - 已於 2024 年 8 月 1 日發行
- 1.0.70665 - Amazon Redshift Serverless 版本 - 已於 2024 年 7 月 8 日發行
- 1.0.70634 - 目前追蹤版本 - 已於 2024 年 7 月 8 日發行
- 1.0.69954 - Amazon Redshift Serverless 版本 - 已於 2024 年 6 月 26 日發行
- 1.0.69952 - 目前追蹤版本 - 已於 2024 年 6 月 26 日發行
- 1.0.69497 - Amazon Redshift Serverless 版本 - 已於 2024 年 6 月 18 日發行
- 1.0.69451 - 目前追蹤版本 - 已於 2024 年 6 月 18 日發行
- 1.0.69076 - Amazon Redshift Serverless 版本 - 已於 2024 年 6 月 14 日發行

- 1.0.69065 - 目前追蹤版本 - 已於 2024 年 6 月 14 日發行
- 1.0.68555 - Amazon Redshift Serverless 版本 - 已於 2024 年 5 月 31 日發行
- 1.0.68540 - 目前追蹤版本 - 已於 2024 年 5 月 31 日發行
- 1.0.68328 - Amazon Redshift Serverless 版本 - 已於 2024 年 5 月 23 日發行
- 1.0.68205 - 目前追蹤版本 - 已於 2024 年 5 月 23 日發行
- 1.0.67796 - Amazon Redshift Serverless 版本 - 已於 2024 年 5 月 15 日發行
- 1.0.67788 - 目前追蹤版本 - 已於 2024 年 5 月 15 日發行
- 1.0.67308 - Amazon Redshift Serverless 版本 - 已於 2024 年 5 月 1 日發行
- 1.0.67305 - 目前追蹤版本 - 已於 2024 年 5 月 1 日發行

此修補程式中的新功能和改善

- 導入對 'lower_attribute_names()' 和 'upper_attribute_names()' 函式的支援，這些函式會修改 SUPER 物件值的屬性名稱大小寫。
- 修正使用身分欄時，CREATE TABLE LIKE 中的問題。先前新資料表會從來源資料表繼承識別碼。如果來源資料表之後遭到捨棄，這就會造成問題，因為新資料表中的識別碼會變成無效。
- 修正造成 SVV_ALL_TABLES 中某些外部資料表無法顯示的問題。
- 改善叢集啟動程序時間，並加速高並行工作負載的查詢初始化。
- 修正聯合查詢在將 split_part() 函式傳遞至 RDS 和 Aurora MySQL 的聯合來源時導致錯誤的問題
- 支援使用者透過 ALTER TABLE...ALTER DISTSTYLE KEY DISTKEY 命令對佈建並行擴展叢集和無伺服器自動擴展運算上的分佈索引鍵初始化的變更。
- 支援手動重新整理涉及佈建並行擴展和無伺服器自動擴展運算上彙總的具體化視觀表。
- 新增支援零 ETL 處理最大 16 MB 的記錄，以及支援最大 16 MB 的 SUPER 值。
- 藉由提供如結構描述和資料表名稱等額外詳細資訊的方式，強化零 ETL 中從 Aurora MySQL 進行初始同步期間的錯誤訊息。
- 推出使用 Amazon Redshift ML CREATE MODEL 標記的支援。進行這項改善後，您現在可以標記 Amazon Redshift ML 使用的 Amazon SageMaker AI 資源。標記有助於管理、識別、組織、搜尋和篩選資源。
- 透過使用 AWS Lambda最佳化資料處理，改善涉及 Lambda 使用者定義函式 (UDF) 的查詢效能。
- 在經過彈性調整大小的無伺服器叢集的排序資料表中，減少資料擷取期間的記憶體使用率。
- 新增支援檢視 SYS_QUERY_HISTORY 的 `query_text` 欄中以及檢視 SYS_QUERY_TEXT 的 `text` 欄中的新行 (\n)。

Amazon Redshift 修補程式 180

此修補程式中的叢集版本：

- 1.0.68520 - 後續追蹤版本 - 已於 2024 年 5 月 28 日發行
- 1.0.67699 - 後續追蹤版本 - 已於 2024 年 5 月 15 日發行
- 1.0.66960 - 後續追蹤版本 - 已於 2024 年 4 月 21 日發行
- 1.0.66954 - 目前追蹤版本 - 已於 2024 年 4 月 21 日發行
- 1.0.66276 - 目前追蹤版本 - 已於 2024 年 4 月 12 日發行
- 1.0.66290 - Amazon Redshift Serverless 版本 - 已於 2024 年 4 月 10 日發行
- 1.0.63590 - 目前追蹤版本 - 已於 2024 年 2 月 19 日發行
- 1.0.63567 - Amazon Redshift Serverless 版本 - 已於 2024 年 2 月 16 日發行
- 1.0.63282 - Amazon Redshift Serverless 版本 - 已於 2024 年 2 月 13 日發行
- 1.0.63269 - 目前追蹤版本 - 已於 2024 年 2 月 13 日發行
- 1.0.63215 - Amazon Redshift Serverless 版本 - 已於 2024 年 2 月 12 日發行
- 1.0.63205 - 目前追蹤版本 - 已於 2024 年 2 月 12 日發行
- 1.0.63030 - Amazon Redshift Serverless 版本 - 已於 2024 年 2 月 7 日發行
- 1.0.62913 - 目前追蹤版本 - 已於 2024 年 2 月 7 日發行
- 1.0.62922 - Amazon Redshift Serverless 版本 - 已於 2024 年 2 月 5 日發行
- 1.0.62878 - 目前追蹤版本 - 已於 2024 年 2 月 5 日發行
- 1.0.62698 - Amazon Redshift Serverless 版本 - 已於 2024 年 1 月 31 日發行
- 1.0.62614 - 目前追蹤版本 - 已於 2024 年 1 月 31 日發行
- 1.0.61687 — Amazon Redshift Serverless 版本 — 於 2024 年 1 月 5 日發行
- 1.0.61678 — 目前的追蹤版本 — 於 2024 年 1 月 5 日發行
- 1.0.61567 – Amazon Redshift Serverless 版本 – 於 2023 年 12 月 31 日發行
- 1.0.61559 — 目前的追蹤版本 — 於 2023 年 12 月 31 日發行
- 1.0.61430 — Amazon Redshift Serverless 版本 — 於 2023 年 12 月 29 日發行
- 1.0.61395 — 目前的追蹤版本 — 於 2023 年 12 月 29 日發行

此修補程式中的新功能和改善

- 將 CURRENT_USER 變更為不再將傳回的使用者名稱截斷為 64 個字元。

- 新增在標準檢視和近期繫結視觀表上套用資料遮罩政策的功能。
- 新增會將動態資料遮罩 (DDM) 套用至 SUPER 資料類型資料欄中的純量屬性的功能。
- 新增 OBJECT_TRANSFORM SQL 函數。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 OBJECT_TRANSFORM 函數。
- 新增將 AWS Lake Formation 精細存取控制套用至巢狀資料的功能，並使用 Amazon Redshift 資料湖分析進行查詢。
- 新增 INTERVAL 資料類型。
- 新增 CONTINUE_HANDLER，這是一種例外狀況處理常式，可控制預存程序的流程。若使用該處理常式，您就可以掌握和處理例外狀況，而不會結束現有的陳述式區塊。
- 除了個別物件外，還新增定義限定範圍 (結構描述或資料庫) 權限的功能。這可讓使用者和角色獲得限定範圍內所有目前和未來物件的權限。
- 新增從具有權限的資料共用建立資料庫的功能，讓取用者端管理員將共用資料庫物件的個別權限授予取用者端使用者和角色。
- 新增對來自遠端 BYOM 模型的 SUPER 傳回資料類型的支援。這會擴大可接受的 SageMaker AI 模型範圍，以納入傳回格式更複雜的模型。
- 將外部函數變更為現在隱含將含或不含小數部分的數字轉換為資料欄的數字資料類型。對於 int2、int4 和 int8 資料欄，除非數字超出範圍，否則具有小數位數的數字可以截斷至被接受。對於 float4 和 float8 資料欄，數字被接受，沒有小數位數。
- 新增三個與 H3 分層地理空間索引網格系統配合使用的空間函數：H3_FromLongLat、H3_FromPoint 和 H3_Polyfill。

Amazon Redshift 修補程式 179

此修補程式中的叢集版本：

- 1.0.62317 - Amazon Redshift Serverless 版本 - 已於 2024 年 1 月 29 日發行
- 1.0.62312 - 後續追蹤版本 - 已於 2024 年 1 月 29 日發行
- 1.0.61631 — Amazon Redshift Serverless 版本 — 於 2024 年 1 月 5 日發行
- 1.0.61626 — 目前的追蹤版本 — 於 2024 年 1 月 5 日發行
- 1.0.61191 — 目前的追蹤版本 — 於 2023 年 12 月 16 日發行
- 1.0.61150 — Amazon Redshift Serverless 版本 — 於 2023 年 12 月 16 日發行
- 1.0.60982 — Amazon Redshift Serverless 版本 — 於 2023 年 12 月 13 日發行
- 1.0.60854 — 目前的追蹤版本 — 於 2023 年 12 月 10 日發行

- 1.0.60354 — Amazon Redshift Serverless 版本 — 於 2023 年 11 月 22 日發行
- 1.0.60353 — 目前的追蹤版本 — 於 2023 年 11 月 21 日發行
- 1.0.60293 — Amazon Redshift Serverless 版本 — 於 2023 年 11 月 21 日發行
- 1.0.60292 — 目前的追蹤版本 — 於 2023 年 11 月 22 日發行
- 1.0.60161 — Amazon Redshift Serverless 版本 — 於 2023 年 11 月 18 日發行
- 1.0.60140 — 目前的追蹤版本 — 於 2023 年 11 月 18 日發行
- 1.0.60139 — Amazon Redshift Serverless 版本 — 於 2023 年 11 月 18 日發行
- 1.0.59947 — Amazon Redshift Serverless 版本 — 於 2023 年 11 月 16 日發行
- 1.0.59945 — 目前的追蹤版本 — 於 2023 年 11 月 16 日發行
- 1.0.59118 — Amazon Redshift Serverless 版本 — 於 2023 年 11 月 9 日發行
- 1.0.59117 — 目前的追蹤版本 — 於 2023 年 11 月 9 日發行

此修補程式中的新功能和改善

- 新增支援，讓具有適當權限的聯合使用者可以檢視資料列層級安全性和動態資料遮罩系統檢視，包括：
  - SVV_ATTACHED_MASKING_POLICY
  - SVV_MASKING_POLICY
  - SVV_RLS_ATTACHED_POLICY
  - SVV_RLS_POLICY
  - SVV_RLS_RELATION
- 新增功能，查詢只包含 FROM 子句中的純量函數，現在就會產生錯誤。新增功能，使得 FROM 子句中僅包含純量函數的查詢產生錯誤。
- 將具有永久目標資料表功能的 CREATE TABLE AS (CTAS) 陳述式新增至並行擴展叢集。並行擴展叢集目前支援更多查詢。
- 在 RA3 叢集上執行傳統調整大小後，新增下列系統資料表以追蹤資料表重新分佈狀態：
  - SYS_RESTORE_STATE 系統資料表會顯示資料表層級的重新分佈進度。
  - SYS_RESTORE_LOG 系統資料表會顯示資料重新分佈的歷史輸送量。
- 改善在 RA3 節點類型上執行傳統調整大小之後，EVEN 表上的配量偏斜最小化。這也適用於執行傳統調整大小的 178 個叢集的修補程式。
- 在並行擴展叢集上新增對具有 EXTENSION 的 UNLOAD 支援。

- 提升 HashJoins 和 NestLoop 聯結中包含 Λ UDF 的查詢效能。
- 提升 RA3 節點類型上彈性調整大小的效能。
- 提升資料共用查詢的效能。
- 提升在彈性調整大小的佈建叢集和無伺服器工作群組中手動起啟分析查詢的效能。
- 透過更好的工作負載管理資源預測功能，提升自動 WLM 查詢效能。
- 移除將叢集啟動到專用租用 VPC 的功能。此變更不會影響 VPC 中任何 EC2 執行個體的租用。您可以使用 `modify-vpc-tenancy` AWS CLI 命令將 VPC 租用修改為預設值。
- 已佈建的並行擴展叢集和無伺服器自動擴展運算現在支援具體化視觀表手動重新整理。
- 將對 INTERVAL 逐字的支援新增到 EXTRACT 函數。例如，`EXTRACT('hours' from Interval '50 hours')` 傳回 2 因為 50 小時被解譯為 2 天和 2 小時，並擷取 2 的小時分量。

Amazon Redshift 修補程式 178

此修補程式中的叢集版本：

- 1.0.63327 - 目前追蹤版本 - 已於 2024 年 2 月 9 日發行
- 1.0.63313 - 後續追蹤版本 - 已於 2024 年 2 月 9 日發行
- 1.0.60977 — 隨後的追蹤版本 — 於 2023 年 12 月 15 日發行
- 1.0.59596 — 目前的追蹤版本 — 於 2023 年 11 月 9 日發行
- 1.0.58593 — Amazon Redshift Serverless 版本 — 已於 2023 年 10 月 23 日發行
- 1.0.58558 — 目前的追蹤版本 — 已於 2023 年 10 月 23 日發行
- 1.0.57864 — 目前的追蹤版本 — 已於 2023 年 10 月 12 日發行
- 1.0.57850 — Amazon Redshift Serverless 版本 — 已於 2023 年 10 月 12 日發行
- 1.0.56952 — 目前的追蹤版本 — 已於 2023 年 9 月 25 日發行
- 1.0.56970 — Amazon Redshift Serverless 版本 — 已於 2023 年 9 月 25 日發行

此修補程式中的新功能和改善

- Amazon Redshift 現已改善了資料共用查詢效能，所憑借的方法是在生產者執行個體上發生並行資料變更時，以更快的速度重新整理取用者執行個體上的中繼資料。
- 新增了相關支援，可在具體化視觀表的基本資料表參考共用的資料時，以自動和增量的方式重新整理 Amazon Redshift 資料共用取用者執行個體上的具體化視觀表。

- 新增了相關支援，可在 SUPER 資料類型中存放最多 16 MB 大小的大型物件。從 JSON、PARQUET、TEXT 和 CSV 來源檔案擷取時，您可以將半結構化資料或文件載入為 SUPER 資料類型的值，最多 16 MB。
- 新增了相關支援，可彈性地調整大小，以便擴展為單節點 Amazon Redshift RA3 叢集或從單節點 Amazon Redshift RA3 叢集進行擴展。
- 單節點 Amazon Redshift RA3 叢集現在可以受益於加密功能的增強，減少整體加密時間，並改善加密程序進行期間的資料倉儲可用性。
- 改善了將儲存在 SUPER 資料類型的資料解除巢狀化和取消樞紐時，對於查詢的支援。
- 改善了以 SUPER 資料類型重新整理具體化視觀表的效能。
- 新增了對於使用 ANY_VALUE 函數來彙總 INTERVAL 常值的支援。
- 串流擷取現在支援使用下列新的 SQL 命令清除串流資料：`DELETE FROM streaming_materialized_views WHERE <where filter clause>`。
- 視等式條件的結果而定，DECODE 函數會將特定值取代為另一個特定值或預設值。DECODE 現在需要以下三個參數：
  - 表達式
  - search
  - result
- 對預存程序新增功能，以允許補捉資料溢位資料類型轉換錯誤，並在例外狀況處理區塊內進行處理。
- 現在，如果您將 enable_case_sensitive_identifier 變更為與工作階段預設設定不同，則會在查詢資料列層級安全或動態資料遮罩保護的關係時收到錯誤。此外，當您在佈建叢集或無伺服器命名空間中套用資料列層級安全或動態資料遮罩政策時，下列組態會遭到封鎖：

  ```
  ALTER USER <current_user> SET case-sensitive identifier.
  ```
- MERGE 命令現在支援只需要目標和來源資料表的簡化語法。如需詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 MERGE。
- 新增了相關支援，可將相同的動態資料遮罩政策附加到具有相同優先順序或不指定優先順序的多個使用者或角色。
- 您現在可以在透過 ALTER TABLE ADD COLUMN 新增新的資料欄時指定 COLLATION。
- 修正了會延遲對並行擴展叢集和 Amazon Redshift Serverless 強制執行 QMR 規則的問題。
- Amazon Redshift 聯合查詢已在 Amazon RDS for PostgreSQL 和 Amazon Aurora PostgreSQL 上擴大了對時區與時間戳記的下推支援。
- 您現在可以在聯合查詢中使用以數字開頭的 Amazon RDS for MySQL 和 Aurora MySQL 資料庫名稱。

- 新增 SYS_ANALYZE_HISTORY 檢視，其中包含 ANALYZE 操作的記錄詳細資訊。
- 新增 SYS_ANALYZE_COMPRESSION_HISTORY 檢視，其中包含 COPY 或 ANALYZE COMPRESSION 命令執行期間的壓縮分析操作記錄詳細資訊。
- 新增 SYS_SESSION_HISTORY 檢視，其中包含與作用中工作階段、歷史工作階段和重新啟動的工作階段相關的記錄詳細資訊。
- 新增 SYS_TRANSACTION_HISTORY 檢視，其中包含與交易層級分析相關的記錄詳細資訊，可提供遞交所花時間、遞交的 datasha 區塊數量和隔離層級。
- 新增 SVV_REDSHIFT_SCHEMA_QUOTA 檢視，其中包含與資料庫中每個結構描述之配額和目前的磁碟用量相關的記錄。
- 使用 SYS_PROCEDURE_CALL 檢視，其中包含與預存程序呼叫相關的記錄，包括開始時間、結束時間、預存程序呼叫的狀態，以及巢狀預存程序呼叫的呼叫階層。
- 新增 SYS_CROSS_REGION_DATASHARING_USAGE 檢視，其中包含與追蹤跨區域資料共用用量相關的記錄。
- 新增 SYS_PROCEDURE_MESSAGES 檢視，其中包含與關於所記錄預存程序訊息之追蹤資訊相關的記錄。
- 新增 SYS_UDF_LOG 檢視，其中包含與使用者定義函數的呼叫、錯誤、警告或追蹤 (如果適用) 的追蹤系統日誌訊息相關的記錄。
- 在 SYS_EXTERNAL_QUERY_DETAIL 中新增了新的資料欄 IS_RECURSIVE、IS_NESTED、S3LIST_TIME 和 GET_PARTITION_TIME。
- 新增 MaxRPU，這是適用於 Redshift Serverless 的新運算成本控制設定。若使用 MaxRPU，您可以選擇性地指定較高的運算閾值，以控制不同時間點的資料倉儲成本，方法是選取 Redshift Serverless 可以在每個工作群組調整的最大運算層級。
- 使用數值間隔字串更正 INTERVAL 常值的輸出。例如，指定為 `INTERVAL '1' YEAR` 的間隔現在會傳回 `1 YEAR` 而不是 `"00:00:00`。此外，INTERVAL 常值的輸出會截斷為指定的最小 INTERVAL 元件。例如，`INTERVAL '1 day 1 hour 1 minute 1.123 seconds' HOUR TO MINUTE` 會截斷至 `1 day 01:01:00`。

Amazon Redshift 修補程式 177

此修補程式中的叢集版本：

- 1.0.57922 — 隨後的追蹤版本 — 已於 2023 年 10 月 12 日發行
- 1.0.57799 — Amazon Redshift Serverless 版本 — 已於 2023 年 10 月 10 日發行
- 1.0.57798 — 目前的追蹤版本 — 已於 2023 年 10 月 10 日發行

- 1.0.57085 — 隨後的追蹤版本 — 已於 2023 年 9 月 26 日發行
- 1.0.56899 — Amazon Redshift Serverless 版本 — 已於 2023 年 9 月 21 日發行
- 1.0.56754 — 目前的追蹤版本 — 已於 2023 年 9 月 21 日發行
- 1.0.56242 — 目前的追蹤版本 — 已於 2023 年 9 月 11 日發行
- 1.0.55539 — Amazon Redshift Serverless 版本 — 已於 2023 年 8 月 28 日發行
- 1.0.55524 — 目前的追蹤版本 — 已於 2023 年 8 月 28 日發行
- 1.0.54899 — 目前的追蹤版本 — 已於 2023 年 8 月 15 日發行
- 1.0.54899 — 目前的追蹤版本 — 已於 2023 年 8 月 14 日發行
- 1.0.54899 — 目前的追蹤版本 — 已於 2023 年 8 月 15 日發行
- 1.0.54239 — 目前的追蹤版本 — 已於 2023 年 8 月 3 日發行
- 1.0.54321 — Amazon Redshift Serverless 版本 — 已於 2023 年 8 月 3 日發行

此修補程式中的新功能和改善

- 新增 SYS_MV_STATE 檢視，其中包含具體化視觀表每個狀態轉換的資料列。SYS_MV_STATE 可用於 Amazon Redshift Serverless 和 Amazon Redshift 佈建的執行個體的 MV 重新整理監控。
- 新增 SYS_USERLOG 檢視，其中會記錄 Create user、Drop user、Alter user (重新命名)、Alter user (更改屬性) 的資料庫使用者變更詳細資訊。
- 新增 SYS_COPY_REPLACEMENTS 檢視，其中會顯示一個日誌，日誌內記錄了搭配 ACCEPTINVCHARS 選項的 COPY 命令何時取代無效的 UTF-8 字元。
- 新增 SYS_SPATIAL_SIMPLIFY 檢視，其中包含使用 COPY 命令取得有關簡化空間幾何物件的資訊。
- 新增 SYS_VACUUM_HISTORY 檢視，可用來查看 VACUUM 操作的詳細資訊和結果。
- 新增 SYS_SCHEMA_QUOTA_VIOLATIONS 檢視，以記錄超過結構描述配額時的出現次數、時間戳記、XID 和其他有用的資訊。
- 新增 SYS_RESTORE_STATE 檢視，可用來在非同步傳統調整大小期間，監控叢集中每個資料表的重新分佈進度。
- 新增 SYS_EXTERNAL_QUERY_ERROR 檢視，其會傳回有關 Redshift Spectrum 掃描錯誤的資訊。
- 對 CREATE MODEL 命令新增了標籤參數，因此您現在可以使用自動駕駛訓練工作來追蹤訓練成本。
- 為 Amazon Redshift 叢集新增自訂網域名稱 (CNAME)。

- 新增 Apache Iceberg 的預覽支援，讓客戶能夠在 Amazon Redshift 內的 Apache Iceberg 資料表上執行分析查詢。
- 新增在工作負載管理 (WLM) 中搭配使用使用者角色與參數群組的支援。
- 新增自動掛載 的支援 AWS Glue Data Catalog，讓客戶更輕鬆地在資料湖中執行查詢。
- 新增一些功能，以便在使用不含 GROUP BY 子句的分組函數或在 WHERE 子句中使用分組操作時導致錯誤。
- 對預存程序新增功能，以允許補捉除以零錯誤，並在例外狀況處理區塊內進行處理。
- 修正當來源資料表是資料共用資料表時，查詢無法使用並行擴展以將資料寫入到資料表的錯誤。
- 將記錄在 enable_case_sensitive_identifier 的區分大小寫識別碼修正為現在可與 MERGE 陳述式搭配運作。
- 修正在函數 pg_get_late_binding_view_cols() 上的查詢偶爾會被忽略的錯誤。您現在可以隨時取消這類查詢。
- 改善在生產者上執行清空任務時，於取用者上執行的資料共用查詢的效能。
- 改善資料共用和並行擴展查詢的效能，尤其是生產者上的並行資料變更，或卸載至附加到取用者的並行擴展執行個體時。

Amazon Redshift 修補程式 176

此修補程式中的叢集版本：

- 1.0.56738 — 隨後的追蹤版本 — 已於 2023 年 9 月 21 日發行
- 1.0.55837 — 隨後的追蹤版本 — 已於 2023 年 9 月 11 日發行
- 1.0.54776 — 目前的追蹤版本 — 已於 2023 年 8 月 15 日發行
- 1.0.54052 — 目前的追蹤版本 — 已於 2023 年 7 月 26 日發行
- 1.0.53642 — Amazon Redshift Serverless 版本 — 已於 2023 年 7 月 20 日發行
- 1.0.53301 — 目前的追蹤版本 — 已於 2023 年 7 月 20 日發行
- 1.0.52943 — Amazon Redshift Serverless 版本 — 已於 2023 年 7 月 7 日發行
- 1.0.52931 — 目前的追蹤版本 — 已於 2023 年 7 月 7 日發行
- 1.0.52194 — Amazon Redshift Serverless 版本 — 已於 2023 年 6 月 21 日發行
- 1.0.51986 — 目前的追蹤版本 — 已於 2023 年 6 月 16 日發行
- 1.0.51594 — 目前的追蹤版本 — 已於 2023 年 6 月 9 日發行

此修補程式中的新功能和改善

- 改善為空白分組集寫入 GROUP BY () 時的錯誤處理。這之前會被忽略，現在則會傳回剖析器錯誤。
- 增強以增量方式重新整理有 SUPER 資料欄之具體化視觀表時的效能
- ALTER TABLE <target_tbl> APPEND FROM <streaming_mv> – (ATA) SQL 命令現在支援將所有記錄從做為來源的串流具體化視觀表 (MV) (除了以資料表做為來源外) 移動到目標資料表。串流 MV 上的 ATA 支援可讓使用者將串流 MV 中的所有記錄移至另一個資料表來管理資料成長，藉此快速清除串流 MV 中的所有記錄。
- TRUNCATE <streaming_mv> – SQL 命令現在支援截斷串流具體化視觀表 (MV) (除了資料表外) 中的所有記錄。TRUNCATE 會刪除串流 MV 中的所有記錄，同時讓串流 MV 的結構保持不變。在串流 MV 上執行 TRUNCATE 可讓客戶快速清除串流 MV 中的所有記錄，以管理資料成長。
- 在 SELECT 命令中新增了 QUALIFY 子句的功能。
- 透過與 Amazon Forecast 整合，Redshift 機器學習支援時間序列預測。
- AWS Glue Data Catalog 支援自動掛載，可簡化資料湖的查詢，而無需額外的步驟來建立外部結構描述參考。
- 現已支援更改 RLS 政策。如需詳細資訊，請參閱文件中的 ALTER RLS POLICY。
- Lambda UDF 現在可在 CREATE FUNCTION 陳述式中支援 STABLE 函數波動參數。當 CREATE FUNCTION 陳述式中使用了 STABLE 參數，且使用相同引數多次呼叫 Lambda UDF 時，預期的 Lambda UDF 函數叫用次數會減少。CREATE FUNCTION 參數中有 STABLE 函數波動類別的更詳細解釋。
- 多個 Lambda UDF 效能改善。具體而言，已改善在查詢受資料列層級安全 (RLS) 政策保護的資料表時的記錄批次處理支援。
- 減少 Amazon Redshift RA3 叢集的整體加密時間，並改善加密期間資料倉儲的可用性。如需詳細資訊，請參閱 Amazon Redshift 資料庫加密。
- Redshift 中已新增新的系統檢視 SYS_MV_REFRESH_HISTORY。針對具體化視觀表的重新整理活動，SYS_MV_REFRESH_HISTORY 檢視會包含一個資料列。您可以使用 SYS_MV_REFRESH_HISTORY 檢查具體化視觀表的重新整理歷史記錄。所有使用者都可看見 SYS_MV_REFRESH_HISTORY。超級使用者可以看見所有資料列；一般使用者只能看見自己的資料。

  系統檢視 SYS_QUERY_DETAIL 中已新增一個新的資料欄 SPILLED_BLOCK_LOCAL_DISK。新資料欄 SPILLED_BLOCK_LOCAL_DISK 可協助客戶判斷溢寫到本機磁碟的區塊。您可以使用 SYS_QUERY_DETAIL 來檢視步驟層級的查詢詳細資訊。所有使用者都可看見 SYS_QUERY_DETAIL。超級使用者可以看見所有資料列；一般使用者只能看見它們有權存取的中繼資料。

- 佈建的 Amazon Redshift Serverless 和 Amazon Redshift 中已新增新的系統檢視 SYS_QUERY_TEXT。SYS_QUERY_TEXT 檢視類似於佈建叢集的 SVL_STATEMENTTEXT。使用 SYS_QUERY_TEXT 檢視中的 sequence 資料欄可取得完整的 SQL 陳述式文字。

Amazon Redshift 修補程式 175

此修補程式中的叢集版本：

- 1.0.53064 — 目前的追蹤版本 — 已於 2023 年 7 月 7 日發行
- 1.0.51973 — 目前的追蹤版本 — 已於 2023 年 6 月 16 日發行
- 1.0.51781 — 目前的追蹤版本 — 已於 2023 年 6 月 10 日發行
- 1.0.51314 — Amazon Redshift Serverless 版本 — 已於 2023 年 6 月 3 日發行
- 1.0.51304 — 目前的追蹤版本 — 已於 2023 年 6 月 2 日發行
- 1.0.50708 — 目前的追蹤版本 — 已於 2023 年 5 月 19 日發行
- 1.0.50300 — 目前的追蹤版本 — 已於 2023 年 5 月 8 日發行
- 1.0.49710 — Amazon Redshift Serverless 版本 — 已於 2023 年 4 月 28 日發行
- 1.0.49676 — 目前的追蹤版本 — 已於 2023 年 4 月 28 日發行

此修補程式中的新功能和改善

- 次要錯誤修正。
- Amazon Redshift 串流擷取現在支援跨區域串流擷取，您的 source Amazon Kinesis Data Streams (KDS) 或 Amazon Managed Streaming for Apache Kafka (MSK) 主題可以位於 AWS 不同於 Amazon Redshift 資料倉儲所在的 AWS 區域。開始從 Amazon Kinesis Data Streams擷取串流的文件已進行修訂，會說明 REGION 關鍵字的使用方式。
- 埃及日光節約時間調整。
- 已改善 RA3 叢集加密的整體時間。

Amazon Redshift 修補程式 174

1.0.51296 — 已於 2023 年 6 月 2 日發行

發行至隨後的追蹤。無版本備註。

1.0.50468 — 已於 2023 年 5 月 12 日發行

維護版本。無版本備註。

1.0.49780、1.0.49868 和 1.0.49997 – 已於 2023 年 4 月 28 日發行

此版本的版本備註：

- 已改善 Lambda UDF 的批次處理支援。
- Lambda UDF 的增量批次處理。
- 有新的 MERGE SQL 命令，可將來源資料變更套用至 Amazon Redshift 資料表。
- 有新的動態資料遮罩功能，可簡化 Amazon Redshift 資料倉儲中敏感資料的保護程序。
- 與 Lake Formation 共用資料的新集中式存取控制，允許使用 Lake Formation APIs 和 AWS 主控台管理許可授予、檢視存取控制，以及稽核 Amazon Redshift 資料共用中資料表和檢視的許可。
- 埃及日光節約時間調整。

1.0.49087 – 已於 2023 年 4 月 12 日發行

維護版本。無版本備註。

1.0.48805 – 已於 2023 年 4 月 5 日發行

此版本的版本備註：

- Amazon Redshift 針對使用 BYTEDICT、有大量字串的查詢引入了額外的效能增強功能，其中 BYTEDICT 是 Amazon Redshift 中的新壓縮編碼，相較於替代壓縮編碼 (例如 LZO 或 ZSTD)，可將字串式資料的處理速度提升 5 倍到 63 倍。如需此功能的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的比較編碼。

1.0.48004 – 已於 2023 年 3 月 17 日發行

維護版本。無版本備註。

1.0.47470 – 已於 2023 年 3 月 11 日發行

此版本的版本備註：

- 改善 pg_catalog.svv_table_info 的查詢效能 此外，也新增了新的資料欄 create_time。在建立資料表時，這個資料欄會以 UTC 儲存日期/時間戳記。
- 新增對於在聯合查詢上指定工作階段層級逾時的支援。

Amazon Redshift 修補程式 173

1.0.49788 – 已於 2023 年 4 月 28 日發行

此版本的版本備註：

- 埃及日光節約時間調整。

1.0.49074 – 已於 2023 年 4 月 12 日發行

此版本的版本備註：

- 時區組態已更新為 IANA 程式庫版本 2022g。

1.0.48766 – 已於 2023 年 4 月 5 日發行

維護版本。無版本備註。

1.0.48714 – 已於 2023 年 4 月 5 日發行

維護版本。無版本備註。

1.0.48022 – 已於 2023 年 3 月 17 日發行

維護版本。無版本備註。

1.0.47357 – 已於 2023 年 3 月 7 日發行

維護版本。無版本備註。

1.0.46987 – 已於 2023 年 2 月 24 日發行

維護版本。無版本備註。

1.0.46806 – 已於 2023 年 2 月 18 日發行

維護版本。無版本備註。

1.0.46607 – 已於 2023 年 2 月 13 日發行

此版本的版本備註：

- 現在，如果具有手動設定之交錯排序索引鍵的資料表將其分散樣式設定為 DISTSTYLE KEY，我們會自動將該資料表轉換為複合排序索引鍵，以改善這些資料表的效能。這會在將快照還原到 Amazon Redshift Serverless 時完成。

1.0.45698 – 已於 2023 年 1 月 20 日發行

此版本的版本備註：

- 在 UNLOAD 命令中新增了副檔名參數，以便在檔案名稱後自動新增副檔名。
- 可在將受 RLS 保護的物件新增到資料共用時，或是這些物件已是資料共用的一部分時，支援保護這些物件。管理員現在可以關閉資料共用的 RLS，以允許取用者存取受保護的物件。
- 新增用於監控的新系統資料表：SVV_ML_MODEL_INFO、SVV_MV_DEPENDENCY 和 SYS_LOAD_DETAIL。此外，也在系統資料表 SYS_QUERY_DETAIL 中新增資料欄 data_skewness 和 time_skewness。

Amazon Redshift 修補程式 172

此修補程式中的叢集版本：

- 1.0.46534 – 已於 2023 年 2 月 18 日發行
- 1.0.46523 – 已於 2023 年 2 月 13 日發行
- 1.0.46206 – 已於 2023 年 2 月 1 日發行
- 1.0.45603 – 已於 2023 年 1 月 20 日發行
- 1.0.44924 – 已於 2022 年 12 月 19 日發行
- 1.0.44903 – 已於 2022 年 12 月 18 日發行
- 1.0.44540 – 已於 2022 年 12 月 13 日發行
- 1.0.44126 – 已於 2022 年 11 月 23 日發行
- 1.0.43980 – 已於 2022 年 11 月 17 日發行

此修補程式中的新功能和改善

- 依預設，由 CTAS 建立的資料表為 AUTO。
- 新增對於具體化視觀表的資料列層級安全 (RLS) 支援。
- 增加 S3 的逾時以改善跨區域資料共用。
- 新增了新的空間函數 ST_GeomFromGeohash。
- 改善從複合主索引鍵中自動選取分散索引鍵的功能，以改善立即可用的效能。
- 針對具有複合主索引鍵之資料表的分散索引鍵新增自動的主索引鍵，以改善立即可用的效能。
- 改善並行擴展功能，以允許更多查詢在資料發生變更時也能擴展。
- 改善資料共用查詢效能。
- 為分類模型新增 Machine Learning 機率指標。
- 新增用於監控的新系統資料表：SVV_USER_INFO、SVV_MV_INFO、SYS_CONNECTION_LOG、SYS_DATASHARE_USAGE_PRODUCER 和 SYS_DATASHARE_CHANGE_LOG。
- 新增對於在 Parquet 和 ORC 檔案類型的外部資料表中查詢 VARBYTE 資料欄的支援。

Amazon Redshift 修補程式 171

此修補程式中的叢集版本：

- 1.0.43931 – 已於 2022 年 11 月 16 日發行
- 1.0.43551 – 已於 2022 年 11 月 5 日發行
- 1.0.43331 – 已於 2022 年 9 月 29 日發行
- 1.0.43029 – 已於 2022 年 9 月 26 日發行

此修補程式中的新功能和改善

- CONNECT BY 支援：新增對 CONNECT BY SQL 建構模組的支援，可讓您根據該資料集內的父子關係遞迴查詢資料倉儲中的階層式資料。

Amazon Redshift 修補程式 170

此修補程式中的叢集版本：

- 1.0.43922 – 已於 2022 年 11 月 21 日發行
- 1.0.43573 – 已於 2022 年 11 月 7 日發行
- 1.0.41881 – 已於 2022 年 9 月 20 日發行
- 1.0.41465 – 已於 2022 年 9 月 7 日發行
- 1.0.40325 – 已於 2022 年 7 月 27 日發行

此修補程式中的新功能和改善

- ST_GeomfromGeoJSON：在 GeoJSON 表示中從 VARCHAR 建構 Amazon Redshift 空間幾何物件。

Amazon Redshift 修補程式 169

此修補程式中的叢集版本：

- 1.0.41050 – 已於 2022 年 9 月 7 日發行
- 1.0.40083 – 已於 2022 年 7 月 16 日發行
- 1.0.39734 – 已於 2022 年 7 月 7 日發行
- 1.0.39380 – 已於 2022 年 6 月 23 日發行
- 1.0.39251 – 已於 2022 年 6 月 15 日發行
- 1.0.39009 – 已於 2022 年 6 月 8 日發行

此修補程式中的新功能和改善

- 將角色新增為 Alter Default Privileges 命令的參數，以支援角色型存取控制。
- 新增 ACCEPTINVCHARS 參數，以支援在從 Parquet 和 ORC 檔案進行複製時取代無效的 UTF-8 字元。
- 新增 OBJECT(k,v) 函數以從鍵值組建構 SUPER 物件。

Amazon Redshift 修補程式 168

此修補程式中的叢集版本：

- 1.0.38698 – 已於 2022 年 5 月 25 日發行
- 1.0.38551 – 已於 2022 年 5 月 20 日發行
- 1.0.38463 – 已於 2022 年 5 月 18 日發行
- 1.0.38361 – 已於 2022 年 5 月 13 日發行
- 1.0.38199 – 已於 2022 年 5 月 9 日發行
- 1.0.38112 – 已於 2022 年 5 月 6 日發行
- 1.0.37684 – 已於 2022 年 4 月 20 日發行

此修補程式中的新功能和改善

- 在 Amazon Redshift ML 中新增對於線性學習程式模型類型的支援。
- 為 SQL 交易隔離層級新增 SNAPSHOT 選項。
- 新增 `farmhashFingerprint64` 做為 VARBYTE 和 VARCHAR 資料的新雜湊演算法。
- 在具體化視觀表的增量重新整理中支援 AVG 函數。
- 在 Redshift Spectrum 中對外部資料表支援相互關聯的子查詢。
- 為了改善立即可用的查詢效能，Amazon Redshift 會自動為特定資料表選擇單一資料欄主索引鍵做為分散索引鍵。

Amazon Redshift 中的行為變更

隨著 Amazon Redshift 不斷發展和改進，也會導入某些行為變更，以增強效能、安全和使用者體驗。此頁面可作為完整的資源，讓您隨時掌握這些重要更新、採取行動，並避免潛在的工作負載中斷。

即將發生的行為變更

以下說明即將發生的行為變更。

主題

- Amazon Redshift 從 2026 年 9 月 30 日開始拍攝暫停資料共用生產者叢集的自動快照
- SUPER 資料類型支援從修補程式 206 開始的較大個別字串
- AWS KMS 2026 年 8 月 17 日之後 Amazon Redshift Serverless APIs 的金鑰許可強制執行
- 從修補程式 208 開始，Amazon Redshift 會在多次嘗試登入失敗後強制執行使用者鎖定
- 自 2026 年 6 月 8 日起，Amazon Redshift Serverless 和 Amazon Redshift RG 執行個體上手動快照的增強型計費模型
- Lake Formation 資料表上的 Iceberg DELETE 需要從修補程式 202 開始的 DELETE 許可
- 從修補程式 202 開始，Amazon Redshift Serverless 會在快照還原時保留零 ETL 和 S3 事件整合
- 2026 年 12 月 31 日終止對 Amazon Redshift ODBC 1.x 驅動程式的支援
- 純量 Python UDF 將於 2026 年 6 月 30 日之後終止支援
- 2026 年 2 月 27 日之後的具體化視觀表 (MV) Auto-REFRESH 行為變更
- Amazon Redshift 在 2026 年 2 月 16 日之後，即不支援透過資料共用存取取用者資訊的函式
- 最低 Transport Layer Security (TLS) 版本變更自 2026 年 10 月 31 日開始生效
- Amazon Redshift 在 2025 年 10 月 30 日之後，即不支援建立新的純量 Python UDF

Amazon Redshift 從 2026 年 9 月 30 日開始拍攝暫停資料共用生產者叢集的自動快照

從 2026 年 9 月 30 日開始，Amazon Redshift 會拍攝暫停資料共用生產者叢集的自動快照。由於消費者可以寫入暫停的生產者，這些快照會擷取這些資料共用寫入並提供復原點。

如果您暫停資料共用生產者叢集，可能會受到此影響。

先前，Amazon Redshift 不會在暫停時拍攝這些叢集的自動快照，因此可用於還原的最新自動快照是在暫停叢集時拍攝的快照。自動化快照現在會在整個暫停期間繼續建立。

這些快照遵循叢集現有的自動化快照排程和保留期，而且不需要採取任何動作。對於非資料共用生產者的暫停叢集，Amazon Redshift 不會在叢集暫停時拍攝自動快照。

如需詳細資訊，請參閱自動快照。

SUPER 資料類型支援從修補程式 206 開始的較大個別字串

從修補程式 206 開始，Amazon Redshift 支援 SUPER 資料類型內最多 16，000，000 個位元組的個別字串（用來存放半結構化資料）。此限制也適用於記憶體內字串操作。若要使用此功能，請建立資料庫參數群組，並將 enable_large_strings_opt_in 參數設定為 Yes。然後，將參數群組連接至您的資料倉儲。對於佈建的叢集，您必須重新啟動叢集，才能套用參數變更。

如果您使用 SUPER 資料類型來存放大型 JSON 文件，此變更可能會影響您。如果您的工作負載依賴VARCHAR(MAX)檢視、使用者定義函數 (UDFs) 或處理超過 65，535 個位元組字串的預存程序，這也可能對您造成影響。

當您將 enable_large_strings_opt_in 設定為 時Yes，記憶體內操作會解析VARCHAR(MAX)為VARCHAR(16000000)。這些操作包括 CAST、檢視、UDFs和預存程序。然後，他們可以處理超過 65，535 個位元組的字串。此變更不會影響 VARCHAR 資料欄的持久性儲存。CREATE TABLE 與 會VARCHAR(MAX)繼續建立VARCHAR(65535)資料欄。當您將 參數設定為時No，Amazon Redshift 會保留資料倉儲的現有行為。若要檢查資料倉儲的目前狀態，請執行 SELECT is_large_string_enabled();。

當您將 enable_large_strings_opt_in設定為 時Yes：

- COPY 和 INSERT到 SUPER 資料欄支援大於先前 65，535 位元組限制的大小。JSON_PARSE、JSON_SERIALIZE和 等字串函數LISTAGG也支援這些較大的大小。
- 現有的檢視、UDFs和預存程序會維持在 65，535 個位元組的限制，直到您重新建立為止。
- 如果現有檢視使用 VARCHAR(MAX) 和其他物件（例如其他檢視）相依於它，則 CREATE OR REPLACE VIEW會失敗。錯誤是「無法變更檢視欄的資料類型」。若要更新檢視，請捨棄相依物件、捨棄並重新建立檢視，然後重新建立相依物件。如果沒有這些額外步驟，您可以取代使用 VARCHAR(MAX) 且沒有相依物件的檢視。
- 在資料共用生產者和消費者資料倉儲之間保持此設定一致。

當您將 enable_large_strings_opt_in設定為 時No：

- 您無法COPY將超過 65，535 個位元組的INSERT字串輸入 SUPER 資料欄。您仍然可以透過 SUPER 讀取現有的字串。不過，轉換為 VARCHAR(MAX) 會將值截斷為 65，535 個位元組。
- JSON_PARSE、 JSON_SERIALIZE和 等字串函數LISTAGG無法處理超過 65，535 個位元組的 VARCHAR 值。
- 您無法查詢處理超過 65，535 個位元組字串常值的檢視、預存程序和 UDFs。
- 如果您在 參數設定為 VARCHAR(MAX)時使用 建立檢視Yes，而其他物件 （ 例如其他檢視 ） 依賴它，則 會CREATE OR REPLACE VIEW失敗，並出現相同的「無法變更檢視資料欄的資料類型」錯誤。請遵循上述清單中所述的相同修復步驟。
- 跨執行個體快照還原可能無法擷取長字串。當快照在 SUPER 欄中包含超過 65，535 個位元組的字串時，就會發生此失敗。當您將該快照還原至將此參數設為 的叢集時，就會發生這種情況No。

如需修補程式版本的詳細資訊，請參閱 適用於 Amazon Redshift 的叢集版本。

AWS KMS 2026 年 8 月 17 日之後 Amazon Redshift Serverless APIs 的金鑰許可強制執行

Amazon Redshift Serverless 現在會執行新的授權檢查，要求呼叫特定 Amazon Redshift Serverless APIs IAM 主體擁有加密目標命名空間之客戶受管金鑰的明確 AWS KMS 許可。過去，Amazon Redshift 會使用 Amazon Redshift 服務角色代表您執行 AWS KMS 操作，而不驗證呼叫的 IAM 主體本身是否獲授權使用該金鑰。

啟用新的授權檢查後，來自缺乏必要 AWS KMS 許可之主體的 API 呼叫會失敗，並顯示 ValidationException：如果您已使用客戶受管金鑰加密 Amazon Redshift Serverless 命名空間，並使用 IAM 主體來呼叫下列受影響的 APIs，"Insufficient KMS permissions."則可能會受到此影響。

為了避免中斷，請檢閱您帳戶中的哪些 IAM 主體會針對使用客戶受管金鑰加密的命名空間呼叫下列 APIs，並確保這些主體具有金鑰所需的 AWS KMS 許可。您可以透過 AWS KMS 金鑰政策、IAM 身分型政策或授予來 AWS KMS 授予這些許可。

此變更不會影響使用預設 AWS擁有的金鑰加密的命名空間。如果您的 IAM 主體已經擁有下列必要 AWS KMS 許可，則不需要採取任何動作。

受影響的 APIs 和命名空間客戶受管金鑰所需的許可如下所示：

- CreateWorkgroup、RestoreFromSnapshot、 RestoreFromRecoveryPoint – 需要 kms:Encrypt、kms:CreateGrant、 kms:GenerateDataKey和 kms:Decrypt。

- `RestoreTableFromSnapshot`，`RestoreTableFromRecoveryPoint` – 需要 `kms:Decrypt`。

如需詳細資訊，請參閱《 AWS Key Management Service 開發人員指南》中的控制對 AWS KMS 金鑰的存取。如需 Amazon Redshift 資料庫加密的詳細資訊，請參閱 Amazon Redshift 資料庫加密。

從修補程式 208 開始，Amazon Redshift 會在多次嘗試登入失敗後強制執行使用者鎖定

從 Amazon Redshift 修補程式 208 開始，為了增強資料倉儲的安全狀態，Amazon Redshift 會在超過設定的連續失敗登入嘗試次數後自動鎖定資料庫使用者。此功能預設為啟用，閾值為 5 次失敗的嘗試（可設定為 2 到 50 次），並同時適用於佈建的叢集和無伺服器工作群組。

如果您使用資料庫（密碼型）連線進行連線，則可能會受到此影響。一旦鎖定，即使使用正確的密碼，登入嘗試也會遭到拒絕，直到管理員或超級使用者解鎖為止。透過 IAM 或 AWS IAM Identity Center 驗證的聯合身分不會受到影響，因為它們使用短期字符，而不是靜態密碼。

若要管理此行為：

- 使用 設定閾值 `ALTER SYSTEM SET max_failed_login_attempts TO value;`(2–50，預設 5)。
- 使用 檢查所有使用者的鎖定狀態`SHOW USER LOCKOUT;`。
- 使用 解鎖使用者`ALTER USER username LOGIN PASSWORD 'password';`，也會重設嘗試失敗的計數器。

如需詳細資訊，請參閱 SHOW USER LOCKOUT、ALTER USER 和 max_failed_login_attempts。

自 2026 年 6 月 8 日起，Amazon Redshift Serverless 和 Amazon Redshift RG 執行個體上手動快照的增強型計費模型

從 2026 年 6 月 8 日開始，Amazon Redshift 宣布在 Amazon Redshift Serverless 和 Amazon Redshift RG 執行個體上手動快照的增強型計費模式。透過此增強功能，手動快照現在會根據存放在您帳戶中所有作用中手動快照的唯一資料區塊來計量。出現在多個快照中的共用資料區塊只會計算一次，以您區域的現有手動快照費率計費。

如果您在 Amazon Redshift Serverless 工作群組或 Amazon Redshift RG 叢集上使用手動快照，則可能會受到此影響。這可降低維護多個快照的客戶手動快照成本。

無需採取任何動作。增強型帳單模型會自動套用到現有和新的手動快照。

如需快照定價的詳細資訊，請參閱 Amazon Redshift 定價。

Lake Formation 資料表上的 Iceberg DELETE 需要從修補程式 202 開始的 DELETE 許可

從 Amazon Redshift 修補程式 202 開始，Lake Formation (LF) 受管資料表上的 Iceberg DELETE 操作需要 DELETE Lake Formation 許可。UPDATE 和 MERGE 操作需要 INSERT 和 DELETE 許可。所有 Iceberg DML 操作都需要 ALTER 許可。

如果您針對 AWS Lake Formation 管理的 Iceberg 資料表執行 DELETE、UPDATE 或 MERGE 操作，則可能會受到此影響。

先前，只有 INSERT 許可（且沒有 DELETE 許可）的主體可以在 LF 受管 Iceberg 資料表上執行刪除操作。從修補程式 202 開始，DELETE 操作需要 DELETE Lake Formation 許可。完全依賴 INSERT 許可來執行刪除的委託人會收到許可錯誤，直到明確授予 DELETE 許可為止。S3 資料表不受此變更影響。

若要在此變更後繼續執行 DELETE 操作，請檢閱 Iceberg 資料表的 Lake Formation 許可授予，並確保執行刪除操作的委託人已授予 DELETE Lake Formation 許可。您可以使用 AWS Lake Formation 主控台或 `aws lakeformation list-permissions` AWS CLI 命令來驗證現有的授權。

如果您需要暫時還原至先前的行為，請聯絡 AWS Support，在叢集上停用此變更，而不需要部署程式碼。

如需修補程式版本的詳細資訊，請參閱 適用於 Amazon Redshift 的叢集版本。

從修補程式 202 開始，Amazon Redshift Serverless 會在快照還原時保留零 ETL 和 S3 事件整合

從 Amazon Redshift 修補程式 202 開始，當您將 Amazon Redshift Serverless 命名空間從快照或復原點還原至相同的無伺服器命名空間時，會自動維護與命名空間和快照/復原點相關聯的零 ETL 和 S3 事件整合。不需要其他動作。

如果您將快照或復原點還原至已設定零 ETL 或 S3 事件整合的 Amazon Redshift Serverless 命名空間，則可能會受到此影響。

先前，將快照或復原點還原至標記為關聯的零 ETL 和 S3 事件整合失敗的無伺服器命名空間，要求您在完成還原後手動重新建立它們。從修補程式 202 開始，這些整合預設為維護，並在還原完成後繼續操作。

此功能僅適用於還原至相同無伺服器命名空間的 Amazon Redshift Serverless。將快照還原至不同的命名空間不會維護整合。佈建叢集上的快照還原不會維持零 ETL 或 S3 事件整合。

若要在還原期間選擇退出維護整合，請取消勾選還原頁面上的維護整合方塊 AWS 管理主控台。如果您使用的是 AWS CLI，請在呼叫 restore-from-snapshot 或 restore-from-recovery-point API 操作時設定 `--no-maintain-integration` 參數。當您選擇退出時，整合會在還原後進入 `FAILED` 狀態。然後，您可以刪除失敗的整合並重新建立它們。

如需修補程式版本的詳細資訊，請參閱 適用於 Amazon Redshift 的叢集版本。

2026 年 12 月 31 日終止對 Amazon Redshift ODBC 1.x 驅動程式的支援

從 2026 年 12 月 31 日開始，Amazon Redshift 將停止支援 ODBC 1.x 驅動程式。根據客戶的意見回饋，我們已將原始end-of-support日期從 2026 年 9 月 30 日延長至 2026 年 12 月 31 日，以提供額外的遷移時間。這同時適用於 Amazon Redshift 佈建叢集和無伺服器工作群組。

如果您使用任何版本的 ODBC 1.x 驅動程式連線至 Amazon Redshift，則可能會受到此影響。若要驗證您是否使用 ODBC 1.x 驅動程式，請執行下列查詢：

```
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%');
```

若要繼續取得 Amazon Redshift ODBC 驅動程式連線的技術支援，請在 2026 年 12 月 30 日之前遷移至最新的 Amazon Redshift ODBC 2.x 驅動程式。

在生產環境中遷移至 ODBC 2.x 驅動程式之前，我們建議您進行徹底的概念驗證，以確認新驅動程式符合您的所有功能需求。

我們建議您使用最新版本的 Amazon Redshift ODBC 2.x 驅動程式，並將 ApplicationName 屬性設定為在連線至 Amazon Redshift 時識別您的應用程式。

純量 Python UDF 將於 2026 年 6 月 30 日之後終止支援

Amazon Redshift 將於 2026 年 6 月 30 日之後終止對 Python UDF 的支援。我們建議您使用 Lambda UDF 作為替代方案。

相較於 Python UDF，Lambda UDF 具備以下優點：

- Lambda UDF 可以從 UDF 邏輯內連線至外部服務和 API。
- Lambda UDF 使用 Lambda 運算資源。大量運算或記憶體密集型 Lambda UDF 不會影響 Amazon Redshift 的查詢效能或資源並行。
- Lambda UDF 支援執行 Python 程式碼。Lambda UDF 支援多個 Python 執行時期，取決於特定使用案例。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的使用 Python 建置。
- 您可以在單獨的服務界限中隔離自訂程式碼執行。這樣做可簡化維護、監控、預算編列和許可管理的工作。

如需有關建立和使用 Lambda UDF 的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的純量 Lambda UDF。如需將現有 Python UDF 轉換為 Lambda UDF 的相關資訊，請參閱部落格文章。

2026 年 2 月 27 日之後的具體化視觀表 (MV) Auto-REFRESH 行為變更

從 2026 年 2 月 27 日開始，Amazon Redshift 具體化視觀表的 Auto REFRESH 查詢會以使用者查詢的形式執行，而非背景自律程序。因此，Auto REFRESH 查詢現在會以與其他使用者查詢相同的優先順序執行。

此變更可改善已啟用 Auto REFRESH 的具體化視觀表的新鮮度，協助他們掌握其基礎資料表相較於先前行為的最新變更。

注意：MV Auto REFRESH 行為變更功能僅適用於修補程式版本 P198 及更新版本的 CURRENT 軌道上的 Amazon Redshift 佈建叢集。它目前在 Serverless 上已停用。

Amazon Redshift 在 2026 年 2 月 16 日之後，即不支援透過資料共用存取取用者資訊的函式

自 2026 年 2 月 16 日起，Amazon Redshift 將不再支援使用透過資料共用存取取用者使用者、角色或群組資訊的 `user_is_member_of` 和相關函數。

最低 Transport Layer Security (TLS) 版本變更自 2026 年 10 月 31 日開始生效

自 2026 年 10 月 31 日起，Amazon Redshift 將強制執行最低 Transport Layer Security (TLS) 版本 1.2。使用 TLS 1.0 或 1.1 版的傳入連線將遭到拒絕。這同時適用於 Amazon Redshift 佈建叢集和無伺服器工作群組。不是使用 TLS 的 Amazon Redshift 資料倉儲則不受此變更影響。

如果您使用 TLS 1.0 或 1.1 版連線到 Amazon Redshift，則可能會受此更新影響。

若要確認您目前使用的 TLS 版本，您可以：

對於 Amazon Redshift 佈建：查看 STL_CONNECTION_LOG 系統資料表 [1] 中的 sslversion 欄。

對於 Amazon Redshift Serverless 工作群組：查看 SYS_CONNECTION_LOG 系統資料表 [2] 中的 ssl_version 欄。

若要在此變更生效後持續存取 Amazon Redshift 資料倉儲，請遵循下列步驟：

1. 更新您的用戶端以支援 TLS 1.2 或更高版本
2. 安裝可支援 TLS 1.2+ 的最新驅動器版本

如果可能的話，建議您使用最新版本的 Amazon Redshift 驅動器 [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

Amazon Redshift 在 2025 年 10 月 30 日之後，即不支援建立新的純量 Python UDF

Amazon Redshift 自 2025 年 10 月 30 日起不再支援建立新的 Python UDF。現有 Python UDF 將繼續正常運作。我們強烈建議您在此日期之前，將現有的 Python UDF 移轉至 Lambda UDF。

相較於 Python UDF，Lambda UDF 具備以下優點：

- Lambda UDF 可以從 UDF 邏輯內連線至外部服務和 API。
- Lambda UDF 使用 Lambda 運算資源。大量運算或記憶體密集型 Lambda UDF 不會影響 Amazon Redshift 的查詢效能或資源並行。
- Lambda UDF 支援執行 Python 程式碼。Lambda UDF 支援多個 Python 執行時期，取決於特定使用案例。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的使用 Python 建置。
- 您可以在單獨的服務界限中隔離自訂程式碼執行。這樣做可簡化維護、監控、預算編列和許可管理的工作。

如需有關建立和使用 Lambda UDF 的資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的純量 Lambda UDF。如需將現有 Python UDF 轉換為 Lambda UDF 的相關資訊，請參閱部落格文章。

近期行為變更

主題

- Amazon Redshift 在 2025 年 8 月 26 日之後使用最新的 IANA 時區資料庫
- Amazon Redshift Serverless RPU 變更於 2025 年 8 月 15 日之後生效
- 資料庫稽核記錄變更於 2025 年 8 月 10 日之後生效
- 無伺服器工作群組的虛擬私有雲端點變更於 2025 年 6 月 27 日之後生效
- 查詢監控變更在 2025 年 5 月 2 日之後生效
- 安全變更在 2025 年 1 月 10 日之後生效

Amazon Redshift 在 2025 年 8 月 26 日之後使用最新的 IANA 時區資料庫

自 2025 年 8 月 26 日起，Amazon Redshift 採用最新的 IANA 時區資料庫修補程式來計算時區。此變更會更改特定時區和時段的日期和時間轉換運作方式。此更新會影響明確的時區轉換，例如使用 CONVERT_TIMEZONE 函式或 TIMEZONE 和 AT TIME ZONE 命令執行的轉換，以及在類型轉換操作期間進行的隱含轉換，特別是在 TIMESTAMP 和 TIMESTAMPTZ 格式之間。

以下是時區和時段組合的更新清單：

- 時區現在可正確遵守 2038 年之後的日光節約時間 (DST)。先前在 2038 年之後就沒有遵守 DST 的時區。
- `America/Toronto` 時區和與其連結的時區會在 1947 到 1950 年於當地時間凌晨 2 點進行 DST 切換，而不是在午夜。
- Amazon Redshift 現在可正確反映所有時區標準化之前時段的當地標準時間 (LMT)。此為時區專屬時段，大多數時區會在 19 世紀中期之前切換到標準化。
- `EET`、`CET`、`WET` 和 `MET` 現在視為正常時區，而不是縮寫。
- Amazon Redshift 中不再有下列時區名稱：
  - `Asia/Riyadh87`
  - `Asia/Riyadh88`
  - `Asia/Riyadh89`
  - `Mideast/Riyadh87`
  - `Mideast/Riyadh88`
  - `Mideast/Riyadh89`

- US/Pacific-New

如需 IANA 時區資料庫的詳細資訊，請參閱 IANA 時區資料庫網站上的時區資料庫。

Amazon Redshift Serverless RPU 變更於 2025 年 8 月 15 日之後生效

自 2025 年 8 月 15 日起，Amazon Redshift Serverless 基礎 Redshift 處理單元 (RPUs) AWS 的帳戶配額為前六個月的 3，200 個 RPUs 或最大彙總基礎 RPUs 1.5 倍。

資料庫稽核記錄變更於 2025 年 8 月 10 日之後生效

從 2025 年 8 月 10 日開始，Amazon Redshift 將變更資料庫稽核記錄，這點需要您採取行動。Amazon Redshift 會將您資料庫中連線和使用者活動的相關資訊記錄到 Amazon S3 儲存貯體和 CloudWatch。在 2025 年 8 月 10 日之後，Amazon Redshift 會停止對具有指定 Redshift IAM USER 之儲存貯體政策的 Amazon S3 儲存貯體進行資料庫稽核記錄。我們建議您更新政策，改為在 S3 儲存貯體政策內使用 Redshift SERVICE-PRINCIPAL 進行稽核記錄。如需有關稽核記錄的資訊，請參閱 Amazon Redshift 稽核記錄的儲存貯體許可。

為了避免任何記錄中斷，請在 2025 年 8 月 10 日之前檢閱並更新 S3 儲存貯體政策，以在相關聯區域中授予 Redshift service-principal 的存取權。如需資料庫稽核記錄的詳細資訊，請參閱 Amazon S3 中的日誌檔案。

如有問題或疑慮，請透過以下連結聯絡 AWS Support： AWS Support。

無伺服器工作群組的虛擬私有雲端點變更於 2025 年 6 月 27 日之後生效

從 2025 年 6 月 27 日開始，Amazon Redshift 將變更無伺服器工作群組的虛擬私有雲端點 (VPCE) 支援。在此日期之前，Amazon Redshift 會在建立工作群組期間將端點部署到單一可用區域 (AZ)，並隨著時間擴展 VPCE 支援至最多三個 AZ。在此日期之後，Amazon Redshift 會在建立工作群組期間指定的最多三個可用區域中部署 VPCE。

如需詳細資訊，請參閱使用 Amazon Redshift Serverless 時的考量。

如有問題或疑慮，請透過以下連結聯絡 AWS Support： AWS Support。

主題

- 查詢監控變更在 2025 年 5 月 2 日之後生效
- 安全變更在 2025 年 1 月 10 日之後生效

查詢監控變更在 2025 年 5 月 2 日之後生效

自 2025 年 5 月 2 日起，我們將不再針對現有和新建立的 Redshift Serverless 工作群組，於查詢限制索引標籤中提供查詢 CPU 時間 (max_query_cpu_time) 和查詢 CPU 用量 (max_query_cpu_percentage) 指標。在此日期之後，我們將根據所有 Redshift Serverless 工作群組中的這些指標自動移除所有查詢限制。

查詢限制的設計在於攔截失控的查詢。不過，查詢 CPU 時間 (max_query_cpu_time) 和查詢 CPU 用量 (max_query_cpu_percentage) 在查詢的生命週期內可能會有所不同，因此不一定是攔截失控查詢的有效方法。若要攔截失控的查詢，建議您利用查詢監控指標來提供一致且可行的資訊。一些範例包括：

- 查詢執行時間 (max_query_execution_time)：確保查詢在預期的時間範圍內完成。
- 傳回資料列計數 (max_scan_row_count)：監控正在處理的資料規模。
- 查詢佇列時間 (max_query_queue_time)：識別花時間佇列的查詢。

如需支援指標的完整清單，請參閱 Amazon Redshift Serverless 的查詢監控指標。

安全變更在 2025 年 1 月 10 日之後生效

安全是我們對於 Amazon Web Services (AWS) 的首要任務。為達成此任務，我們進一步強化 Amazon Redshift 環境的安全狀態，藉由導入增強型安全預設值來協助您遵守資料安全方面的最佳實務，而不需進行額外設定，同時降低設定錯誤的潛在風險。為了避免任何潛在的中斷，請檢閱佈建叢集和無伺服器工作群組建立組態、指令碼和工具，並於生效日期之前進行必要的變更，以符合新的預設設定。

預設停用公開存取

在 2025 年 1 月 10 日之後，所有新建立的佈建叢集，以及從快照還原的叢集，都將預設為停用公開存取。根據預設，此版本僅允許來自相同虛擬私有雲端 (VPC) 內用戶端應用程式的叢集連線。若要從另一個 VPC 中的應用程式存取您的資料倉儲，請設定跨 VPC 存取。此變更將反映在 CreateCluster 和 RestoreFromClusterSnapshot API 操作中，以及對應的 SDK 和 AWS CLI 命令中。如果您從 Amazon Redshift 主控台建立佈建叢集，則叢集的公開存取會預設為停用。

假如您仍需要公開存取，則需要覆寫預設值，並在執行 CreateCluster 或 RestoreFromClusterSnapshot API 操作時將 PubliclyAccessible 參數設為 true。使用可公開存取的叢集時，我們建議您使用安全群組或網路存取控制清單 (ACL) 來限制存取。如需詳細資訊，請參閱VPC security groups (VPC 安全群組)及設 Amazon Redshift 叢集或 Amazon Redshift Serverless 工作群組的安全群組通訊設定。

預設加密

在 2025 年 1 月 10 日之後，Amazon Redshift 將透過啟用加密作為所有新建立 Amazon Redshift 佈建叢集的預設設定，來進一步強化資料和叢集安全。這項變更不適用於從快照還原的叢集。

透過此變更，使用 AWS 管理主控台 AWS CLI或 API 建立佈建叢集而不指定 KMS 金鑰時，將無法再使用解密叢集的功能。叢集會自動使用 加密 AWS 擁有的金鑰。

如果您使用自動化指令碼建立未加密的叢集，或利用未加密的叢集進行資料共用，則可能會受到此更新的影響。為了確保順暢轉換，請更新建立未加密叢集的指令碼。此外，如果您定期建立新的未加密取用者叢集並將其用於資料共用，請檢閱您的組態，以確保生產者和取用者叢集都已加密，防止資料共用活動中斷。如需詳細資訊，請參閱Amazon Redshift 資料庫加密。

強制執行 SSL 連線

在 2025 年 1 月 10 日之後，Amazon Redshift 預設會對連線至新建立的佈建和還原叢集的用戶端強制執行 SSL 連線。此預設變更也會套用至無伺服器工作群組。

此變更生效後，所有新建立或還原的叢集都會導入名為 `default.redshift-2.0` 的新預設參數群組，且 `require_ssl` 參數會預設為 `true`。在未指定參數群組的情況下建立的任何新叢集，都會自動使用 `default.redshift-2.0` 參數群組。透過 Amazon Redshift 主控台建立叢集時，系統會自動選取新的 `default.redshift-2.0` 參數群組。此變更也會反映在 `CreateCluster`和 `RestoreFromClusterSnapshot` API 操作，以及對應的 SDK 和 AWS CLI 命令中。如果您使用現有或自訂參數群組，Amazon Redshift 將繼續遵守您的參數群組中指定的 `require_ssl` 值。您仍然可以視需要選擇變更自訂參數群組中的 `require_ssl` 值。

對於 Amazon Redshift Serverless 使用者，`config-parameters` 中的預設值 `require_ssl` 會變更為 `true`。任何建立新工作群組且將 `require_ssl` 設定為 `false` 的請求，都將遭拒。建立工作群組之後，您可以將 `require_ssl` 值變更為 `false`。如需詳細資訊，請參閱設定連線的安全選項。

請注意，您仍然可以修改叢集或工作群組設定來變更預設行為，以因應您的特定使用案例需要。

Amazon Redshift AWS SDKs的程式碼範例

下列程式碼範例示範如何搭配 AWS 軟體開發套件 (SDK) 使用 Amazon Redshift。

基本概念是程式碼範例，這些範例說明如何在服務內執行基本操作。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他 AWS 服務組合來完成特定任務的程式碼範例。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

程式碼範例

- 使用 AWS SDKs 的 Amazon Redshift 基本範例
  - Hello Amazon Redshift
  - 使用 AWS SDK 了解 Amazon Redshift 的基本概念
  - 使用 AWS SDKs的 Amazon Redshift 動作
    - CreateCluster 搭配 AWS SDK 或 CLI 使用
    - DeleteCluster 搭配 AWS SDK 或 CLI 使用
    - DescribeClusters 搭配 AWS SDK 或 CLI 使用
    - DescribeStatement 搭配 AWS SDK 使用
    - ExecuteStatement 搭配 AWS SDK 使用
    - GetStatementResult 搭配 AWS SDK 使用
    - ListDatabases 搭配 AWS SDK 使用
    - ModifyCluster 搭配 AWS SDK 或 CLI 使用
- 使用 AWS SDKs Amazon Redshift 案例
  - 建立 Amazon Redshift 項目追蹤器
  - 使用 CLI 開始使用無伺服器資料倉儲
  - 佈建資料倉儲叢集入門

使用 AWS SDKs 的 Amazon Redshift 基本範例

下列程式碼範例示範如何搭配 AWS SDK 使用 Amazon Redshift 的基本功能。

範例

- Hello Amazon Redshift
- 使用 AWS SDK 了解 Amazon Redshift 的基本概念
- 使用 AWS SDKs的 Amazon Redshift 動作
  - CreateCluster 搭配 AWS SDK 或 CLI 使用
  - DeleteCluster 搭配 AWS SDK 或 CLI 使用
  - DescribeClusters 搭配 AWS SDK 或 CLI 使用
  - DescribeStatement 搭配 AWS SDK 使用
  - ExecuteStatement 搭配 AWS SDK 使用
  - GetStatementResult 搭配 AWS SDK 使用
  - ListDatabases 搭配 AWS SDK 使用
  - ModifyCluster 搭配 AWS SDK 或 CLI 使用

Hello Amazon Redshift

下列程式碼範例示範如何開始使用 Amazon Redshift。

.NET

適用於 .NET 的 SDK (v4)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
/// <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}");
       }
   }
```

- 如需 API 詳細資訊，請參閱《適用於 .NET 的 AWS SDK API 參考》中的 DescribeClusters。

Go

SDK for Go V2

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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)
 }
}
```

- 如需 API 詳細資訊，請參閱《適用於 Go 的 AWS SDK API 參考》中的 DescribeClusters。

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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()));
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 DescribeClusters。

Python

適用於 Python 的 SDK (Boto3)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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"))
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 DescribeClusters。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

使用 AWS SDK 了解 Amazon Redshift 的基本概念

下列程式碼範例示範如何：

- 建立 Redshift 叢集。
- 列出叢集中的資料庫。
- 建立名為 Movies 的資料表。
- 填入 Movies 資料表。
- 依年份查詢 Movies 資料表。
- 修改 Redshift 叢集。
- 刪除 Amazon Redshift 叢集。

.NET

適用於 .NET 的 SDK (v4)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

建立 Redshift 包裝函式類別來管理操作。

```
/// <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}");
    }
}
```

執行示範 Redshift 基本概念的互動式案例。

```
/// <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; }
    }
}
```

- 如需 API 詳細資訊，請參閱《適用於 .NET 的 AWS SDK API 參考》中的下列主題。
  - CreateCluster
  - DescribeClusters
  - DescribeStatement
  - ExecuteStatement
  - GetStatementResult
  - ListDatabasesPaginator
  - ModifyCluster

Go

SDK for Go V2

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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)
```

```
    }
   }
  }
 }
```

- 如需 API 詳細資訊，請參閱《適用於 Go 的 AWS SDK API 參考》中的下列主題。
  - CreateCluster
  - DescribeClusters
  - DescribeStatement
  - ExecuteStatement
  - GetStatementResult
  - ListDatabasesPaginator
  - ModifyCluster

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

執行展示 Amazon Redshift 功能的互動式情境。

```
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 方法的包裝函式類別。

```
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());
                }
            });
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的下列主題。
  - CreateCluster
  - DescribeClusters
  - DescribeStatement
  - ExecuteStatement
  - GetStatementResult
  - ListDatabasesPaginator
  - ModifyCluster

Python

適用於 Python 的 SDK (Boto3)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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
```

顯示案例實作的主要函數。

```
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"
    )
```

案例中使用的包装函式。

```
    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
```

- 如需 API 詳細資訊，請參閱《適用於 Python (Boto3) 的AWS SDK API 參考》中的下列主題。
  - CreateCluster
  - DescribeClusters
  - DescribeStatement
  - ExecuteStatement
  - GetStatementResult
  - ListDatabasesPaginator
  - ModifyCluster

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

使用 AWS SDKs的 Amazon Redshift 動作

下列程式碼範例示範如何使用 AWS SDKs 執行個別 Amazon Redshift 動作。每個範例均包含 GitHub 的連結，您可以在連結中找到設定和執行程式碼的相關說明。

這些摘錄會呼叫 Amazon Redshift API，是必須在內容中執行之大型程式的程式碼摘錄。您可以在 使用 AWS SDKs Amazon Redshift 案例 中查看內容中的動作。

下列範例僅包含最常使用的動作。如需完整清單，請參閱《Amazon Redshift API 參考》。

範例

- CreateCluster 搭配 AWS SDK 或 CLI 使用
- DeleteCluster 搭配 AWS SDK 或 CLI 使用

- DescribeClusters 搭配 AWS SDK 或 CLI 使用
- DescribeStatement 搭配 AWS SDK 使用
- ExecuteStatement 搭配 AWS SDK 使用
- GetStatementResult 搭配 AWS SDK 使用
- ListDatabases 搭配 AWS SDK 使用
- ModifyCluster 搭配 AWS SDK 或 CLI 使用

`CreateCluster` 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 `CreateCluster`。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- 了解基本概念
- 佈建資料倉儲叢集入門

.NET

適用於 .NET 的 SDK (v4)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    /// <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;
        }
    }
```

- 如需 API 詳細資訊，請參閱《適用於 .NET 的 AWS SDK API 參考》中的 CreateCluster。

CLI

AWS CLI

建立具有最小參數的叢集；此範例會建立具有最小參數集的叢集。根據預設，輸出會使用 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
```

結果：

```
{
   "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"
   }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 CreateCluster。

Go

SDK for Go V2

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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
}
```

- 如需 API 詳細資訊，請參閱《適用於 Go 的 AWS SDK API 參考》中的 CreateCluster。

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

建立 叢集

```
    /**
     * 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);
                }
            });
    }
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 CreateCluster。

JavaScript

適用於 JavaScript (v3) 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

建立用戶端。

```
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 };
```

建立 叢集

```
// 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();
```

- 如需 API 詳細資訊，請參閱《適用於 JavaScript 的 AWS SDK API 參考》中的 CreateCluster。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

建立 叢集

```
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}")
    }
}
```

- 如需 API 詳細資訊，請參閱《適用於 Kotlin 的AWS SDK API 參考》中的 CreateCluster。

Python

適用於 Python 的 SDK (Boto3)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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
```

下列程式碼會執行個體化 RedshiftWrapper 物件。

```
    client = boto3.client("redshift")
    redhift_wrapper = RedshiftWrapper(client)
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 CreateCluster。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

建立 叢集

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateCluster。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`DeleteCluster` 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 `DeleteCluster`。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- 佈建資料倉儲叢集入門

.NET

適用於 .NET 的 SDK (v4)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    /// <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;
        }
    }
```

- 如需 API 詳細資訊，請參閱《適用於 .NET 的 AWS SDK API 參考》中的 DeleteCluster。

CLI

AWS CLI

刪除沒有最終叢集快照的叢集；此範例會刪除叢集，強制刪除資料，因此不會建立最終叢集快照。命令：

```
aws redshift delete-cluster --cluster-identifier mycluster --skip-final-cluster-snapshot
```

刪除叢集，允許最終叢集快照；此範例會刪除叢集，但會指定最終叢集快照。命令：

```
aws redshift delete-cluster --cluster-identifier mycluster --final-cluster-snapshot-identifier myfinalsnapshot
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DeleteCluster。

Go

SDK for Go V2

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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
}
```

- 如需 API 詳細資訊，請參閱《適用於 Go 的 AWS SDK API 參考》中的 DeleteCluster。

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

刪除叢集。

```
   /**
    * 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());
               }
           });
```

```
    }
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 DeleteCluster。

JavaScript

適用於 JavaScript (v3) 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

建立用戶端。

```
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 };
```

建立 叢集

```
// 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();
```

- 如需 API 詳細資訊，請參閱《適用於 JavaScript 的 AWS SDK API 參考》中的 DeleteCluster。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

刪除叢集。

```
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}")
    }
}
```

- 如需 API 詳細資訊，請參閱《適用於 Kotlin 的AWS SDK API 參考》中的 DeleteCluster。

Python

適用於 Python 的 SDK (Boto3)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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
```

下列程式碼會執行個體化 RedshiftWrapper 物件。

```
    client = boto3.client("redshift")
    redhift_wrapper = RedshiftWrapper(client)
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 DeleteCluster。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

刪除叢集。

```
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.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteCluster。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`DescribeClusters` 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 `DescribeClusters`。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- 了解基本概念
- 佈建資料倉儲叢集入門

.NET

適用於 .NET 的 SDK (v4)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   /// <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;
        }
    }
```

- 如需 API 詳細資訊，請參閱《適用於 .NET 的 AWS SDK API 參考》中的 DescribeClusters。

CLI

AWS CLI

取得所有叢集的描述；此範例會傳回帳戶所有叢集的描述。根據預設，輸出會使用 JSON 格式。命令：

```
aws redshift describe-clusters
```

結果：

```
{
   "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 選項，以文字格式取得相同的資訊。命令：

--output text 選項。命令：

選項。命令：

```
aws redshift describe-clusters --output text
```

結果：

```
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
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 DescribeClusters。

Go

SDK for Go V2

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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
}
```

- 如需 API 詳細資訊，請參閱《適用於 Go 的 AWS SDK API 參考》中的 DescribeClusters。

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

描述叢集。

```
    /**
     * 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);
            });
    }
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 DescribeClusters。

JavaScript

適用於 JavaScript (v3) 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

建立用戶端。

```
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 };
```

描述您的叢集。

```
// 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();
```

- 如需 API 詳細資訊，請參閱《適用於 JavaScript 的 AWS SDK API 參考》中的 DescribeClusters。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

描述叢集。

```
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}")
            }
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《適用於 Kotlin 的 AWS SDK API 參考》中的 DescribeClusters。

Python

適用於 Python 的 SDK (Boto3)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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
```

下列程式碼會執行個體化 RedshiftWrapper 物件。

```
    client = boto3.client("redshift")
    redhift_wrapper = RedshiftWrapper(client)
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 DescribeClusters。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

描述叢集。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeClusters。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`DescribeStatement` 搭配 AWS SDK 使用

下列程式碼範例示範如何使用 `DescribeStatement`。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- 了解基本概念

.NET

適用於 .NET 的 SDK (v4)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    /// <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;
        }
    }
```

- 如需 API 詳細資訊，請參閱《適用於 .NET 的 AWS SDK API 參考》中的 DescribeStatement。

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   /**
    * 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!");
                }
            });
    }
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 DescribeStatement。

Python

適用於 Python 的 SDK (Boto3)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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
```

下列程式碼會執行個體化 RedshiftDataWrapper 物件。

```
    client = boto3.client("redshift-data")
    redshift_data_wrapper = RedshiftDataWrapper(client)
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 DescribeStatement。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DescribeStatement。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`ExecuteStatement` 搭配 AWS SDK 使用

下列程式碼範例示範如何使用 `ExecuteStatement`。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- 了解基本概念

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

執行 SQL 陳述式以建立資料庫資料表。

```
    /**
     * 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");
               }
           });
   }
```

執行 SQL 陳述式，將資料插入資料庫資料表。

```
   /**
    * 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;
    });
  }
```

執行 SQL 陳述式，查詢資料庫資料表。

```
  /**
   * 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 "";
        });
    }
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 ExecuteStatement。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ExecuteStatement。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`GetStatementResult` 搭配 AWS SDK 使用

下列程式碼範例示範如何使用 `GetStatementResult`。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- 了解基本概念

.NET

適用於 .NET 的 SDK (v4)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    /// <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;
        }
    }
```

- 如需 API 詳細資訊，請參閱《適用於 .NET 的 AWS SDK API 參考》中的 GetStatementResult。

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

檢查陳述式結果。

```
    /**
```

```
    * 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
           });
   }
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 GetStatementResult。

Python

適用於 Python 的 SDK (Boto3)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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
```

下列程式碼會執行個體化 RedshiftDataWrapper 物件。

```
    client = boto3.client("redshift-data")
    redshift_data_wrapper = RedshiftDataWrapper(client)
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Python (Boto3) API Reference 中的 GetStatementResult。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

檢查陳述式結果。

```
    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.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetStatementResult。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`ListDatabases` 搭配 AWS SDK 使用

下列程式碼範例示範如何使用 `ListDatabases`。

.NET

適用於 .NET 的 SDK (v4)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
/// <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;
        }
    }
```

- 如需 API 詳細資訊，請參閱《適用於 .NET 的 AWS SDK API 參考》中的 ListDatabases。

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
   /**
    * 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);
       });
```

```
}
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 ListDatabases。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListDatabases。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`ModifyCluster` 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 `ModifyCluster`。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- 了解基本概念

.NET

適用於 .NET 的 SDK (v4)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    /// <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;
        }
    }
```

- 如需 API 詳細資訊，請參閱《適用於 .NET 的 AWS SDK API 參考》中的 ModifyCluster。

CLI

AWS CLI

將安全群組與叢集建立關聯；此範例說明如何將叢集安全群組與指定的叢集建立關聯。命令：

```
aws redshift modify-cluster --cluster-identifier mycluster --cluster-security-
groups mysecuritygroup
```

修改叢集的維護時段；此顯示如何將叢集的每週偏好維護時段變更為從星期日 11:15 PM 開始，到星期一上午 3:15 AM 結束的最短四個小時時段。命令：

```
aws redshift modify-cluster --cluster-identifier mycluster --preferred-
maintenance-window Sun:23:15-Mon:03:15
```

變更叢集的主密碼；此範例說明如何變更叢集的主密碼。命令：

```
aws redshift modify-cluster --cluster-identifier mycluster --master-user-password
 A1b2c3d4
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 ModifyCluster。

Go

SDK for Go V2

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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
}
```

- 如需 API 詳細資訊，請參閱《適用於 Go 的 AWS SDK API 參考》中的 ModifyCluster。

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

修改叢集。

```
    /**
     * 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");
                }
            });
    }
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Java 2.x API 參考》中的 ModifyCluster。

JavaScript

適用於 JavaScript (v3) 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

建立用戶端。

```
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 };
```

修改叢集。

```
// 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();
```

- 如需 API 詳細資訊，請參閱《適用於 JavaScript 的 AWS SDK API 參考》中的 ModifyCluster。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

修改叢集。

```
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",
        )
    }
}
```

- 如需 API 詳細資訊，請參閱《適用於 Kotlin 的AWS SDK API 參考》中的 ModifyCluster。

Python

適用於 Python 的 SDK (Boto3)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
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
```

下列程式碼會執行個體化 RedshiftWrapper 物件。

```
client = boto3.client("redshift")
redhift_wrapper = RedshiftWrapper(client)
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 ModifyCluster。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

修改叢集。

```
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.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ModifyCluster。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

使用 AWS SDKs Amazon Redshift 案例

下列程式碼範例示範如何在 Amazon Redshift AWS SDKs 中實作常見案例。這些案例示範如何呼叫 Amazon Redshift 中的多個函數，或與其他 AWS 服務結合，藉以完成特定任務。每個案例均包含完整原始碼的連結，您可在連結中找到如何設定和執行程式碼的相關指示。

案例的目標是獲得中等水平的經驗，協助您了解內容中的服務動作。

範例

- 建立 Amazon Redshift 項目追蹤器
- 使用 CLI 開始使用無伺服器資料倉儲
- 佈建資料倉儲叢集入門

建立 Amazon Redshift 項目追蹤器

下列程式碼範例說明如何使用 Amazon Redshift 資料庫建立可追蹤和報告工作項目的 Web 應用程式。

Java

適用於 Java 2.x 的 SDK

說明如何建立可追蹤和報告存放在 Amazon Redshift 資料庫中的工作項目的 Web 應用程式。

如需完整的原始碼以及如何設定 Spring REST API 以查詢 Amazon Redshift 資料並供 React 應用程式使用的說明，請參閱 GitHub 上的完整範例。

此範例中使用的服務

- Amazon Redshift
- Amazon SES

Kotlin

SDK for Kotlin

說明如何建立可追蹤和報告存放在 Amazon Redshift 資料庫中的工作項目的 Web 應用程式。

如需完整的原始碼以及如何設定 Spring REST API 以查詢 Amazon Redshift 資料並供 React 應用程式使用的說明，請參閱 GitHub 上的完整範例。

此範例中使用的服務

- Amazon Redshift
- Amazon SES

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

使用 CLI 開始使用無伺服器資料倉儲

以下程式碼範例顯示做法：

- 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 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在範例開發人員教學課程儲存庫中設定和執行。

```
#!/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)"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CreateNamespace
  - CreateRole
  - CreateSecret
  - CreateWorkgroup
  - DeleteNamespace
  - DeleteRole
  - DeleteRolePolicy
  - DeleteSecret
  - DeleteWorkgroup
  - GetNamespace
  - GetRole
  - GetSecretValue
  - GetWorkgroup
  - PutRolePolicy
  - UpdateNamespace

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

佈建資料倉儲叢集入門

以下程式碼範例顯示做法：

- 建立 Redshift 叢集
- 建立 S3 存取的 IAM 角色
- 建立資料表並載入資料
- 執行範例查詢
- 清除資源

Bash

AWS CLI 搭配 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在範例開發人員教學課程儲存庫中設定和執行。

```
#!/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)"
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的下列主題。
  - CreateCluster
  - CreateRole
  - DeleteCluster
  - DeleteRole
  - DeleteRolePolicy
  - DescribeClusters
  - GetRole
  - ModifyClusterIamRoles
  - PutRolePolicy
  - 等候

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用此服務。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

文件歷史紀錄

> **Note**
>
> 如需 Amazon Redshift 中新功能的說明，請參閱最新功能。

下表說明 2018 年 6 月後，《Amazon Redshift 管理指南》的重要文件變更。如需有關此文件更新的通知，您可以訂閱 RSS 訂閱源。

API 版本：2012-12-01

如需《Amazon Redshift 資料庫開發人員指南》的變更清單，請參閱 Amazon Redshift 資料庫開發人員指南文件歷史記錄。

變更	描述	日期
Amazon Redshift 修補程式 205 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 205。	2026 年 9 月 16 日
Amazon Redshift 修補程式 204 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 204。	2026 年 8 月 11 日
Amazon Redshift 修補程式 203 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能	2026 年 7 月 6 日

	使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 203。	
Amazon Redshift 修補程式 202 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 202。	2026 年 5 月 28 日
Amazon Redshift 修補程式 201 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 201。	2026 年 4 月 22 日
Amazon Redshift 修補程式 200 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 200。	2026 年 3 月 26 日
Amazon Redshift 修補程式 199 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 199。	2026 年 3 月 4 日

Amazon Redshift 修補程式 198 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 198。	2026 年 2 月 3 日
更新 Amazon Redshift 受管政策	新增了 `AmazonRedshiftFederatedAuthorization` 受管政策。	2025 年 11 月 21 日
Amazon Redshift 修補程式 197 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 197。	2025 年 11 月 20 日
Amazon Redshift 修補程式 196 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 196。	2025 年 11 月 10 日
Amazon Redshift 修補程式 195 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 195。	2025 年 10 月 24 日

Amazon Redshift 修補程式 194 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 194。	2025 年 10 月 1 日
Amazon Redshift 修補程式 193 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 193。	2025 年 8 月 26 日
Amazon Redshift 修補程式 192 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 192。	2025 年 7 月 30 日
Amazon Redshift 修補程式 191 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 191。	2025 年 6 月 24 日

Amazon Redshift 修補程式 190 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 190。	2025 年 5 月 7 日
更新 Amazon Redshift 受管政策	更新具有 `lakeformation:GetDataAccess` 許可的 `AmazonRedshiftServiceLinkedRolePolicy` 受管政策。	2025 年 3 月 13 日
Amazon Redshift 修補程式 189 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 189。	2025 年 3 月 7 日
Amazon Redshift 修補程式 188 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 188。	2025 年 2 月 5 日
Amazon Redshift 修補程式 187 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 187。	2024 年 12 月 20 日

更新 Amazon Redshift 受管政策	更新具有 `glue:GetCatalog` 和 `glue:GetCatalogs` 許可的 `AmazonRedshiftServiceLinkedRolePolicy` 受管政策。	2024 年 12 月 3 日
Amazon Redshift 修補程式 186 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 186。	2024 年 10 月 22 日
Amazon Redshift 修補程式 185 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 185。	2024 年 10 月 9 日
Amazon Redshift 修補程式 184 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 184。	2024 年 9 月 12 日

Amazon Redshift 修補程式 183 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 183。	2024 年 8 月 8 日
Amazon Redshift 修補程式 182 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 182。	2024 年 6 月 26 日
Amazon Redshift 修補程式 181 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 181。	2024 年 5 月 1 日
更新 Amazon Redshift 受管政策	具有許可 `servicequotas:GetServiceQuota` 能夠存取 AWS 配額或限制的 `AmazonRedshiftServiceLinkedRolePolicy` 受管政策更新。	2024 年 3 月 8 日

更新查詢編輯器 v2 受管政策	具有 redshift-serverless:ListNamespaces 和 redshift-serverless:ListWorkgroups 許可的 AmazonRedshiftQueryEditorV2NoSharing 、AmazonRedshiftQueryEditorV2ReadSharing 、AmazonRedshiftQueryEditorV2FullAccess 和 AmazonRedshiftQueryEditorV2ReadWriteSharing 受管政策更新。	2024 年 2 月 21 日
更新 Amazon Redshift 唯讀存取受管政策	具有許可 redshift:ListRecommendations 能夠列出 Amazon Redshift Advisor 建議的 AmazonRedshiftReadOnlyAccess 受管政策更新。	2024 年 2 月 7 日
Amazon Redshift 修補程式 180 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 180。	2023 年 12 月 29 日
Amazon Redshift 修補程式 179 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的詳細資訊，請參閱 Amazon Redshift 修補程式 179。	2023 年 11 月 9 日

更新 Amazon Redshift 受管政策	更新具有 `ec2:AssignIpv6Addresses` 和 `ec2:UnassignIpv6Addresses` 許可的 `AmazonRedshiftServiceLinkedRolePolicy` 受管政策。	2023 年 10 月 31 日
Amazon Redshift 修補程式 178 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的相關資訊，請參閱 Amazon Redshift 修補程式 178。	2023 年 9 月 25 日
更新查詢編輯器 v2 受管政策	更新具有 `sqlworkbench:GetAutocompletionMetadata` 和 `sqlworkbench:GetAutocompletionResource` 許可的 `AmazonRedshiftQueryEditorV2NoSharing` 、`AmazonRedshiftQueryEditorV2ReadSharing` 和 `AmazonRedshiftQueryEditorV2ReadWriteSharing` 受管政策。	2023 年 8 月 16 日
更新 Amazon Redshift 受管政策	`AmazonRedshiftServiceLinkedRolePolicy` 受管政策的更新，以授予 建立和管理管理員登入資料秘密 AWS Secrets Manager 的許可。	2023 年 8 月 14 日

Amazon Redshift 修補程式 177 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的相關資訊，請參閱 Amazon Redshift 修補程式 177。	2023 年 8 月 3 日
Amazon Redshift 修補程式 176 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的相關資訊，請參閱 Amazon Redshift 修補程式 176。	2023 年 6 月 8 日
Amazon Redshift 修補程式 175 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的相關資訊，請參閱 Amazon Redshift 修補程式 175。	2023 年 4 月 28 日
更新 Amazon Redshift 受管政策	更新 `AmazonRedshiftServiceLinkedRolePolicy` 受管政策，以移除 ec2 網路相關動作的許可。具體來說，這些許可與 Purpose:RedshiftMigrateToVpc 資源標籤相關聯。	2023 年 4 月 27 日
更新 Amazon Redshift 資料 API 受管政策	更新具有 `redshift:GetClusterCredentialsWithIAM` 許可的 `AmazonRedshiftDataFullAccess` 受管政策。	2023 年 4 月 7 日

更新查詢編輯器 v2 受管政策	更新具有 sqlworkbench:GetSchemaInference 許可的 AmazonRedshiftQueryEditorV2NoSharing 、AmazonRedshiftQueryEditorV2ReadSharing 和 AmazonRedshiftQueryEditorV2ReadWriteSharing 受管政策。	2023 年 3 月 21 日
Amazon Redshift 修補程式 174 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的相關資訊，請參閱 Amazon Redshift 修補程式 174。	2023 年 3 月 11 日
更新查詢編輯器 v2 受管政策	更新具有 sqlworkbench:AssociateNotebookWithTab 許可的 AmazonRedshiftQueryEditorV2NoSharing 、AmazonRedshiftQueryEditorV2ReadSharing 和 AmazonRedshiftQueryEditorV2ReadWriteSharing 受管政策。	2023 年 2 月 2 日

Amazon Redshift 修補程式 173 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的相關資訊，請參閱 Amazon Redshift 修補程式 173。	2023 年 1 月 20 日
Amazon Redshift 修補程式 172 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的相關資訊，請參閱 Amazon Redshift 修補程式 172。	2022 年 11 月 17 日
Amazon Redshift 修補程式 171 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的相關資訊，請參閱 Amazon Redshift 修補程式 171。	2022 年 11 月 9 日
Amazon Redshift 修補程式 170 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的相關資訊，請參閱 Amazon Redshift 修補程式 170。	2022 年 7 月 20 日

Amazon Redshift 修補程式 169 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的相關資訊，請參閱 Amazon Redshift 修補程式 169。	2022 年 6 月 8 日
Amazon Redshift 修補程式 168 已發行。	我們即將部署新的 Amazon Redshift 修補程式。在支援的所有 Amazon Redshift 中，新版本需要幾週的時間才能使用 AWS 區域。如需此版本的相關資訊，請參閱 Amazon Redshift 修補程式 168。	2022 年 4 月 19 日
針對 Amazon Redshift 驅動程式支援身分驗證設定檔	您現在可以使用身分驗證設定檔來連線至 Amazon Redshift。	2021 年 8 月 2 日
支援採用 技術的 Amazon Redshift 跨 VPC 端點 AWS PrivateLink	您現在可以將 Redshift 管理的 VPC 端點與 Amazon Redshift 搭配使用。	2021 年 4 月 1 日
支援 Amazon Redshift 查詢編輯器增強功能	您現在可以使用具有增強的 VPC 路由、更長的查詢執行時間和更多叢集節點類型的查詢編輯器。	2021 年 2 月 17 日
支援與合作夥伴進行主控台整合	您可以使用 Amazon Redshift 主控台來與合作夥伴進行整合。	2020 年 12 月 9 日
支援在可用區域之間移動叢集的功能	您現在可以在可用區域之間移動 RA3 叢集。	2020 年 12 月 9 日
支援 ra3.xlplus 節點類型	您現在可以建立 ra3.xlplus 節點類型。	2020 年 12 月 9 日

支援 JDBC 驅動程式 2.0	您現在可以設定 JDBC 驅動程式 2.0 版了。	2020 年 11 月 5 日
支援 Lambda UDF 和字符化	您現在可以撰寫 Lambda UDF 來啟用資料的外部記號化。	2020 年 10 月 26 日
支援排程 SQL 陳述式的執行	您現在可以在 Amazon Redshift 主控台上排程查詢。	2020 年 10 月 22 日
支援 Amazon Redshift 資料 API	您現在可以使用內建的資料 API 來存取 Amazon Redshift。文件更新包括《Amazon Redshift 資料 API 參考》。	2020 年 9 月 10 日
支援 Amazon Redshift 主控台查詢監控	更新指南，描述新的查詢監控圖表。	2020 年 5 月 7 日
支援用量限制	更新指南，說明用量限制。	2020 年 4 月 23 日
多重要素驗證	更新指南，描述多重要素驗證支援。	2020 年 4 月 20 日
彈性調整大小現在支援節點類型變更	更新彈性調整大小說明。	2020 年 4 月 6 日
支援具有受管儲存的 ra3.4xlarge 節點類型	更新指南，加入 ra3.4xlarge 節點類型。	2020 年 4 月 2 日
支援暫停和繼續	更新指南以描述暫停和繼續叢集操作。	2020 年 3 月 11 日
支援 Microsoft Azure AD 作為身分提供者	已更新指南，說明以 Microsoft Azure AD 作為身分提供者的步驟。	2020 年 2 月 10 日
支援 RA3 節點類型	更新指南，說明新的 RA3 節點類型。	2019 年 12 月 3 日

支援新主控台	更新指南以說明新的 Amazon Redshift 主控台。	2019 年 11 月 11 日
安全資訊更新	更新安全資訊文件。	2019 年 6 月 24 日
快照增強功能	Amazon Redshift 現可支援多項增強功能以管理和排程快照。	2019 年 4 月 4 日
並行擴展	您可以設定工作負載管理 (WLM) 以啟用並行擴展模式。如需詳細資訊，請參閱設定工作負載管理。	2019 年 3 月 21 日
已更新 JDBC 和 ODBC 驅動程式	Amazon Redshift 現在支援 JDBC 和 ODBC 驅動程式的新版本。如需詳細資訊，請參閱設定 JDBC 連線。	2019 年 2 月 4 日
延遲維護	如果您需要重新排程叢集的維護時段，您可以選擇將維護延遲最多 14 天。在延遲期間，如果我們需要更新硬體或進行必要的更新，我們會通知您並進行所需的變更。在這些更新期間，您的叢集無法使用。如需詳細資訊，請參閱延遲維護。	2018 年 11 月 20 日
事先通知	Amazon Redshift 會事先提供某些事件的通知。這些事件的事件類別為 `pending`。例如，如果叢集的其中一個節點要硬體更新，我們會傳送事先通知。如同其他 Amazon Redshift 事件一樣，您也可以訂閱待定事件。如需詳細資訊，請參閱訂閱 Amazon Redshift 事件通知。	2018 年 11 月 20 日

彈性調整大小	彈性調整大小是調整叢集大小最快的方法。彈性調整大小會在現有叢集上新增或移除節點，然後自動將資料重新配送到新節點。彈性調整大小操作不會建立新叢集，通常在幾分鐘之內就很快完成。如需詳細資訊，請參閱調整叢集大小。	2018 年 11 月 15 日
取消調整大小操作	您現在可以取消正在進行的調整大小操作。如需詳細資訊，請參閱調整大小操作概觀。	2018 年 11 月 2 日
修改叢集來變更加密	您可以使用 AWS受管金鑰或客戶受管金鑰，將未加密的叢集修改為使用 AWS Key Management Service (AWS KMS) 加密。當您修改叢集以啟用 KMS 加密時，Amazon Redshift 會自動將資料遷移到新的加密叢集。您也可以經由修改叢集，將未加密的叢集遷移到加密的叢集。	2018 年 10 月 16 日
Amazon Redshift Spectrum 支援增強型 VPC 路由	您現在可以針對叢集使用已啟用增強型 VPC 路由的 Redshift Spectrum。您可能需要執行額外的設定步驟。如需詳細資訊，請參閱使用 Amazon Redshift Spectrum 搭配增強型 VPC 路由。	2018 年 10 月 10 日
查詢編輯器	您現在可以從 Amazon Redshift 管理主控台執行 SQL 查詢。	2018 年 10 月 4 日

工作負載執行明細圖表	您現在可以在主控台查看工作負載執行明細圖表，以取得工作負載效能的詳細檢視。如需詳細資訊，請參閱分析工作負載效能。	2018 年 7 月 30 日
維護追蹤	您現在可以選擇維護追蹤，以查明您的叢集一律會更新為最新版的 Amazon Redshift 或以前的版本。如需詳細資訊，請參閱選擇叢集維護軌道。	2018 年 7 月 26 日

下表描述 2018 年 7 月前《Amazon Redshift 管理指南》的重要變更。

變更	說明	發行日期
新的 CloudWatch 指標	已增加新的 CloudWatch 指標來監控查詢效能。如需詳細資訊，請參閱Amazon Redshift 中的效能資料	2018 年 5 月 17 日
HSM 加密	Amazon Redshift 僅支援硬體安全模組 AWS CloudHSM (HSM) 金鑰管理。如需詳細資訊，請參閱Amazon Redshift 資料庫加密。	2018 年 3 月 6 日
IAM 角色鏈結	如果連接至叢集的 IAM 角色無法存取必要資源，您可鏈結其他角色，該角色甚至可以來自其他帳戶。您的叢集接著會暫時擔任鏈結的角色，以存取資料。您也可以用鏈結角色的方式來授予跨帳戶存取。鏈結中的每個角色都會擔任鏈結中的下一個角色，直到叢集擔任鏈結尾端的角色為止。您最多可以鏈結 10 個角色。如需詳細資訊，請參閱在 Amazon Redshift 中鏈結 IAM 角色。	2018 年 2 月 23 日
新的 DC2 節點類型	新一代的密集運算 (DC) 節點類型提供比 DC1 更好的效能，且價格不變。若要利用這項效能提升，可以將 DC1 叢集遷移到較新的 DC2 節點類型。如需詳細資訊，請參閱Amazon Redshift 中的叢集和節點。	2017 年 10 月 17 日

變更	說明	發行日期
ACM 憑證	Amazon Redshift 將會以 AWS Certificate Manager (ACM) 發行的憑證取代您叢集上的 SSL 憑證。ACM 是大多數現行系統信任的公有憑證授權機構 (CA)。您可能需要更新目前的信任根 CA 憑證，才能繼續使用 SSL 連線到叢集。如需詳細資訊，請參閱將 SSL 連線轉換為使用 ACM 憑證。	2017 年 9 月 18 日
服務連結角色	服務連結角色是直接連結至 Amazon Redshift 的一種特殊 IAM 角色類型。服務連結角色由 Amazon Redshift 預先定義，並包含服務代表 Amazon Redshift 叢集呼叫 AWS 服務所需的所有許可。如需詳細資訊，請參閱使用 Amazon Redshift 的服務連結角色。	2017 年 9 月 18 日
IAM 資料庫使用者身分驗證	您可以將系統設定為允許使用者建立使用者登入資料，並根據其 IAM 登入資料登入資料庫。您也可以設定系統，讓使用者透過符合 SAML 2.0 的身分提供者使用聯合單一登入進行登入。如需詳細資訊，請參閱使用 IAM 身分驗證產生資料庫使用者憑證。	2017 年 8 月 11 日
資料表層級還原支援增強型 VPC 路由	使用使用增強型 VPC 路由控制網路流量的叢集現在支援資料表層級的還原。如需詳細資訊，請參閱從快照還原資料表。	2017 年 7 月 19 日
查詢監控規則	使用 WLM 查詢監控規則，您可以為 WLM 查詢定義以指標為基礎的效能邊界，並指定在查詢超出那些邊界時 (日誌記錄、跳轉或中止)，要採取何種動作。請將查詢監控規則定義為工作負載管理 (WLM) 組態的一部分。如需詳細資訊，請參閱工作負載管理。	2017 年 4 月 21 日
增強型 VPC 路由	當您使用 Amazon Redshift 增強型 VPC 路由時，Amazon Redshift 會強制叢集與資料儲存庫之間所有的 COPY 與 UNLOAD 流量通過您的 Amazon VPC。如需詳細資訊，請參閱使用 Redshift 增強型 VPC 路由控制網路流量。	2016 年 9 月 15 日

變更	說明	發行日期
新的連線日誌欄位	連線日誌 稽核日誌有兩個新的欄位可追蹤 SSL 連線。如果您定期將稽核日誌載入 Amazon Redshift 資料表，則需要將以下新欄位新增到目標資料表：sslcompression 和 sslexpansion。	2016 年 5 月 5 日
可進行 COPY 和 UNLOAD 的 IAM 角色	您現在可以指定一或多個 AWS Identity and Access Management (IAM) 角色，讓叢集用來進行身分驗證以存取其他 AWS 服務。IAM 角色提供更安全的替代方法，可利用 COPY、UNLOAD 或 CREATE LIBRARY 命令提供身分驗證。如需詳細資訊，請參閱授權 Amazon Redshift 代表您存取 AWS 服務及使用 IAM 角色授權 COPY、UNLOAD、CREATE EXTERNAL FUNCTION 和 CREATE EXTERNAL SCHEMA 操作。	2016 年 3 月 29 日
從資料表還原	您可以從叢集快照將資料表還原至作用中叢集的新資料表。如需詳細資訊，請參閱從快照還原資料表。	2016 年 3 月 10 日
在政策中使用 IAM 條件	您可以在 IAM 政策中使用 Condition 元素，進一步限制對資源的存取。如需詳細資訊，請參閱使用 IAM 政策條件進行精細定義存取控制。	2015 年 12 月 10 日
修改是否可公開存取	您可以修改 VPC 中的現有叢集，變更是否可公開存取該叢集。如需詳細資訊，請參閱修改叢集。	2015 年 11 月 20 日
文件修正	發佈多項文件修正。	2015 年 8 月 28 日
文件更新	已更新有關網路設定的故障排除指南，以確保具有不同最大傳輸單位 (MTU) 大小的主機，可以決定連線的封包大小。如需詳細資訊，請參閱查詢似乎沒有回應且有時無法觸達叢集。	2015 年 8 月 25 日
文件更新	已修訂有關參數群組的整個小節，使內容更有組織也更清楚。如需詳細資訊，請參閱Amazon Redshift 參數群組。	2015 年 8 月 17 日

變更	說明	發行日期
WLM 動態屬性	WLM 組態參數現在支援動態套用部分屬性。其他屬性仍然是靜態變更，需要將關聯的叢集重新啟動才會套用組態變更。如需詳細資訊，請參閱WLM 動態和靜態屬性及Amazon Redshift 參數群組。	2015 年 8 月 3 日
將 KMS 加密叢集複製到另一個 AWS 區域	新增有關設定快照複製授予以啟用將 AWS KMS加密叢集複製到另一個 AWS 區域的內容。如需詳細資訊，請參閱將 AWS KMS加密的快照複製到另一個快照 AWS 區域。	2015 年 7 月 28 日
文件更新	更新資料庫加密章節，以更清楚說明 Amazon Redshift 如何使用 AWS KMS 或 HSMs 來管理金鑰，以及加密程序如何搭配每個選項運作。如需詳細資訊，請參閱Amazon Redshift 資料庫加密。	2015 年 7 月 28 日
新的節點類型	Amazon Redshift 現在提供了新的節點類型：DS2。已更新對現有節點類型的文件參考，以使用此版本中介紹的新名稱。也已修訂節點類型組合的小節，不但解釋更清楚，並闡明預設配額限制。如需詳細資訊，請參閱Amazon Redshift 中的叢集和節點。	2015 年 6 月 9 日
預留節點優惠	新增有關新的預留節點方案的內容。也已修訂可用方案的小節，更清楚解釋並加以比較，同時提供範例來示範隨需和預留節點定價如何影響計費。如需詳細資訊，請參閱預留節點。	2015 年 6 月 9 日
文件修正	發佈多項文件修正。	2015 年 4 月 30 日
新功能	此版本的 Amazon Redshift 引進最適合與 Amazon Redshift 搭配使用的新的 ODBC 和 JDBC 驅動程式。如需詳細資訊，請參閱使用 SQL 用戶端工具連線至 Amazon Redshift 資料倉儲。	2015 年 2 月 26 日

變更	說明	發行日期
新功能	此版本的 Amazon Redshift 引進叢集效能指標，可讓您檢視及分析查詢執行的詳細資訊。如需詳細資訊，請參閱檢視查詢和載入。	2015 年 2 月 26 日
文件更新	新增了新的範例政策，示範授予許可給 Amazon Redshift 所依賴的常見 AWS 服務動作和資源。如需詳細資訊，請參閱客戶管理政策範例。	2015 年 1 月 16 日
文件更新	已更新有關設定最大傳輸單位 (MTU) 以停用 TCP/IP 巨型訊框的指南。如需詳細資訊，請參閱使用 EC2 建立叢集及查詢似乎沒有回應且有時無法觸達叢集。	2015 年 1 月 16 日
文件更新	修訂`wlm_json_configuration` 參數的相關內容，並提供範例語法，以使用 Linux、macOS 和 Microsoft Windows 作業系統 AWS CLI 上的 來設定此參數。如需詳細資訊，請參閱工作負載管理。	2015 年 1 月 13 日
文件更新	新增遺失事件通知和說明的內容。如需詳細資訊，請參閱Amazon Redshift 佈建叢集事件通知。	2015 年 1 月 8 日
文件更新	更新有關 Amazon Redshift 動作和資源的 IAM 政策指導。已修訂該小節使內容更有組織更清楚。如需詳細資訊，請參閱Amazon Redshift 中的安全。	2014 年 11 月 21 日
新功能	此版本的 Amazon Redshift 推出使用來自 AWS Key Management Service (AWS KMS) 的加密金鑰來加密叢集的功能。 AWS KMS 結合安全、高可用性的硬體和軟體，以提供針對雲端擴展的金鑰管理系統。如需 Amazon Redshift AWS KMS 和 加密選項的詳細資訊，請參閱 Amazon Redshift 資料庫加密和 叢集操作。	2014 年 11 月 12 日
新功能	此版本 Amazon Redshift 引進可以標記資源 (如叢集、快照) 的功能。標記可讓您提供用戶定義的中繼資料，能根據成本分配來分類帳單報告，並幫助您更容易快速識別資源。如需詳細資訊，請參閱在 Amazon Redshift 中標記資源。	2014 年 11 月 4 日

變更	說明	發行日期
新功能	已將 dw1.8xlarge 和 dw2.8xlarge 節點大小的節點上限增加到128 個節點。如需詳細資訊，請參閱Amazon Redshift 中的叢集和節點。	2014 年 10 月 30 日
新功能	新增從 Amazon Redshift 主控台終止查詢和載入的功能。如需詳細資訊，請參閱檢視查詢和載入及檢視載入操作期間的叢集指標。	2014 年 10 月 28 日
文件修正	發佈多項文件修正。	2014 年 10 月 17 日
新內容	新增有關關閉叢集和刪除叢集的內容。如需詳細資訊，請參閱關閉及刪除叢集。	2014 年 8 月 14 日
文件更新	闡明叢集 Allow Version Upgrade (允許版本升級) 設定的行為。如需詳細資訊，請參閱Amazon Redshift 佈建叢集。	2014 年 8 月 14 日
文件更新	已修訂在 Amazon Redshift 主控台中處理叢集的程序、螢幕擷取畫面和主題組織。如需詳細資訊，請參閱叢集操作。	2014 年 7 月 11 日
新內容	新增有關調整 Amazon Redshift 叢集大小的新教學課程，包括如何在調整叢集大小的同時，使叢集處於唯讀模式的時間縮到最短。如需詳細資訊，請參閱調整叢集大小。	2014 年 6 月 27 日
新功能	新增重新命名叢集的功能。如需詳細資訊，請參閱重新命名叢集及修改叢集。	2014 年 6 月 2 日
新功能	新增選項，可在從快照還原叢集時，選取不同的參數群組和安全群組。如需詳細資訊，請參閱從快照還原叢集。	2014 年 5 月 12 日

變更	說明	發行日期
新功能	新增小節來描述如何設定預設 Amazon CloudWatch 警示，以監控 Amazon Redshift 叢集使用的磁碟空間百分比。此警示是叢集建立程序中的新選項。如需詳細資訊，請參閱預設磁碟空間警示。	2014 年 4 月 28 日
文件更新	闡明 Amazon Redshift 中的 Elliptic curve Diffie—Hellman Exchange (ECDHE) 支援的相關資訊。如需詳細資訊，請參閱SSL。	2014 年 4 月 22 日
新功能	新增有關 Amazon Redshift 對 Elliptic curve Diffie—Hellman (ECDH) 金鑰協議通訊協定的支援敘述。如需詳細資訊，請參閱SSL。	2014 年 4 月 18 日
文件更新	修訂並重新組織使用 SQL 用戶端工具連線至 Amazon Redshift 資料倉儲小節中的主題。新增有關 JDBC 和 ODBC 連線的更多資訊，並新增小節來描述連線問題的故障排除。	2014 年 4 月 15 日
文件更新	在指南中的 IAM 政策範例新增版本。	2014 年 4 月 3 日
文件更新	新增有關調整叢集大小時定價如何計算的資訊。如需詳細資訊，請參閱預留節點。	2014 年 4 月 2 日
新功能	新增介紹新參數 `max_cursor_result_set_size` 的小節，它可設定每個游標可以儲存的結果集大小上限 (以 MB 為單位)。此參數值也會影響叢集的並行作用中游標的數量。如需詳細資訊，請參閱Amazon Redshift 參數群組。	2014 年 3 月 28 日
新功能	新增有關 Cluster Version 欄位的說明，現在包括叢集引擎版本和資料庫修訂版號碼。如需詳細資訊，請參閱 Amazon Redshift 佈建叢集。	2014 年 3 月 21 日
新功能	已更新調整大小程序，在叢集的 Status (狀態) 標籤上會顯示新的調整大小進度資訊。如需詳細資訊，請參閱調整叢集大小。	2014 年 3 月 21 日

變更	說明	發行日期
文件更新	重新組織並更新什麼是 Amazon Redshift？，以及已修訂 Amazon Redshift 佈建叢集概觀。發佈多項文件修正。	2014 年 2 月 21 日
新功能	為 Amazon Redshift 叢集新增新的節點類型和大小，並根據意見回饋重寫相關的叢集概觀主題，使內容更有組織更清楚。如需詳細資訊，請參閱Amazon Redshift 佈建叢集。	2014 年 1 月 23 日
新功能	新增新的資訊，有關虛擬私有雲端中的可公開存取 Amazon Redshift 叢集使用彈性 IP (EIP) 地址。如需 Amazon Redshift 中 EIP 的相關資訊，請參閱VPC 中的 Redshift 資源和在 VPC 中建立 Redshift 佈建叢集或 Amazon Redshift Serverless 工作群組。	2013 年 20 月 12 日
新功能	新增 Amazon Redshift AWS CloudTrail 日誌的相關資訊。如需 Amazon Redshift 的 CloudTrail 支援的相關資訊，請參閱使用 CloudTrail 進行記錄。	2013 年 13 月 12 日
新功能	新增有關新的使用者活動日誌及 Amazon Redshift 中資料庫稽核日誌記錄功能的 `enable_user_activity_logging` 資料庫參數的資訊。如需資料庫稽核日誌的相關資訊，請參閱資料庫稽核記錄日誌。如需這資料庫參數的相關資訊，請參閱 Amazon Redshift 參數群組。	2013 年 6 月 12 日
新功能	更新描述設定 Amazon Redshift 以自動將自動和手動快照複製到次要 AWS 區域。如需設定跨區域快照複本的相關資訊，請參閱將快照複製到另一個 AWS 區域。	2013 年 11 月 14 日
新功能	新增小節來描述 Amazon Redshift 對於連線和使用者活動的稽核記錄，以及將這些日誌存放在 Amazon S3 中。如需資料庫稽核日誌的相關資訊，請參閱資料庫稽核記錄日誌。	2013 年 11 月 11 日

變更	說明	發行日期
新功能	新增描述 Amazon Redshift 加密新功能的小節，包括管理硬體安全模組 (HSM) 中的加密金鑰和輪換加密金鑰。如需加密、HSM、金鑰輪換的相關資訊，請參閱 Amazon Redshift 資料庫加密、使用硬體安全模組進行加密及加密金鑰輪換。	2013 年 11 月 11 日
新功能	更新來描述使用 Amazon SNS 發佈 Amazon Redshift 事件通知。如需 Amazon Redshift 事件通知的相關資訊，請參閱 Amazon Redshift 佈建叢集事件通知。	2013 年 11 月 11 日
新功能	已更新內容，描述 IAM 資源層級許可。如需 Amazon Redshift IAM 許可的相關資訊，請參閱 Amazon Redshift 中的安全。	2013 年 8 月 9 日
新功能	已更新內容，描述還原流程指標。如需詳細資訊，請參閱從快照還原叢集。	2013 年 8 月 9 日
新功能	已更新內容，描述叢集快照共用以及建立快照流程指標。如需詳細資訊，請參閱共用快照。	2013 年 7 月 17 日
文件修正	發佈多項文件修正。	2013 年 7 月 8 日
新的主控台畫面	更新了《Amazon Redshift 管理指南》以符合 Amazon Redshift 主控台的變更。	2013 年 22 月 4 日
新指南	這是《Amazon Redshift 管理指南》的第一版。	2013 年 2 月 14 日