aws

User Guide

AWS Health



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

AWS Health: User Guide

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

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is AWS Health? 1
Concepts for AWS Health 2
AWS Health event 2
Account-specific event 3
Public event 3
AWS Health Dashboard 3
AWS Health Dashboard – Service health 4
Event type code 4
Event type categories 4
Event status 6
Actionability 6
Personas 7
Affected entities 7
AWS Health events on Amazon EventBridge 7
AWS Health API 8
Organizational view 8
AWS User Notifications 8
Getting started 10
Sign up for an AWS account 10
View account events in AWS Health Dashboard 11
Open and recent issues 11
Scheduled changes 12
Other notifications 13
Event log 13
Event details 14
Events types 16
Calendar view 16
Affected resources view 17
Time zone settings 18
Your organization health 19
Alerts for AWS Health events 19
Configuring Amazon EventBridge 20
Manage notifications in AWS User Notifications 20
Configure your AWS managed notifications subscription for AWS Health events 21

AWS managed notifications FAQ ... 22
AWS Health Dashboard ... 24
Planned lifecycle events for AWS Health ... 27
What are planned lifecycle events? ... 27
What should I expect when I receive a planned lifecycle event notification? ... 28
Shared responsibility model for resilience ... 30
Accessing planned lifecycle events ... 31
Version catalog for AWS Health ... 32
What is the version catalog? ... 32
What you can do with the version catalog ... 32
How to access the version catalog ... 33
Details available in the version catalog ... 33
Top-level version entry ... 34
Lifecycle event object ... 34
Impact-risk tags ... 35
How to retrieve the version catalog information with the API ... 35
How to view the version catalog in the Health Dashboard ... 36
Integrating with other systems using the AWS Health API ... 38
Signing AWS Health API requests ... 39
Choosing endpoints for AWS Health API requests ... 39
Demos: Retrieving the last seven days of event data programmatically ... 41
Demo: Retrieving the last seven days of AWS Health event data using Java ... 41
Demo: Retrieving the last seven days of AWS Health event data using Python ... 44
Tutorial: Using the AWS Health API with Java examples ... 47
Step 1: Initialize credentials ... 47
Step 2: Initialize an AWS Health API client ... 48
Step 3: Use AWS Health API operations to get event information ... 48
Security ... 52
Data protection ... 53
Data encryption ... 53
Identity and access management ... 54
Audience ... 55
Authenticating with identities ... 55
Managing access using policies ... 56
How AWS Health works with IAM ... 58
Identity-based policy examples ... 63

Troubleshooting ... 76
Using service-linked roles ... 79
AWS managed policies for AWS Health ... 81
Logging and monitoring in AWS Health ... 87
Compliance validation ... 88
Resilience ... 88
Infrastructure security ... 89
Configuration and vulnerability analysis ... 89
Security best practices ... 89
Grant AWS Health users minimum possible permissions ... 89
View the Health Dashboard ... 89
Integrate AWS Health with Amazon Chime or Slack ... 90
Monitor for AWS Health events ... 90
Aggregating AWS Health events ... 91
Prerequisites ... 91
Enabling organizational view ... 92
Viewing organizational view ... 96
Disabling organizational view ... 101
Managing delegated administrator views for an organization ... 102
Registering a delegated administrator account ... 102
Removing a delegated administrator account ... 103
Monitoring for Health events with EventBridge ... 104
Creating EventBridge rules for AWS Region coverage ... 105
High availability setup (optional) ... 106
Simplified integration ... 106
Global events ... 106
Monitoring account-specific and public events for AWS Health ... 106
Backup rules for AWS Health events ... 107
Viewing paginated lists of AWS Health events on EventBridge ... 108
Aggregating AWS Health events using organizational view and delegated administrator access ... 108
Integrating AWS Health event monitoring and notifications with JIRA and ServiceNow ... 109
Configuring an EventBridge rule to send notifications about events ... 109
Using the API or AWS Command Line Interface ... 110
Configuring Amazon Q Developer in chat applications to send notifications about events ... 112
Prerequisites ... 112

Running operations on EC2 instances automatically in response to events 114
Prerequisites 115
Create a rule for EventBridge 119
Reference: AWS Health events Amazon EventBridge schema 122
AWS Health event schema 122
Public Health Event - Amazon EC2 operational issue 137
Account-specific AWS Health Event - Elastic Load Balancing API Issue 138
Account-specific AWS Health Event - backup event for Amazon EC2 Instance Store Drive Performance Degraded 139
Account-specific AWS Health Event - Amazon EC2 Instance Retirement 140
Account-specific AWS Health Event - Lambda Planned Lifecycle Event 141
Monitoring AWS Health 143
Logging AWS Health API calls with AWS CloudTrail 143
AWS Health information in CloudTrail 144
Example: AWS Health log file entries 145
FAQs on supported protocols 147
What are the AWS JSON and Smithy RPC v2 CBOR protocols, and how do they differ? 147
How do I get started with the Smithy RPC v2 CBOR protocol? 147
What are the risks of enabling the Smithy RPC v2 CBOR protocol? 148
What if my solution does not support Smithy RPC v2 CBOR? 148
What AWS Regions support the Smithy RPC v2 CBOR protocol? 148
What latency improvements can I expect when upgrading to SDK versions that support Smithy RPC v2 CBOR? 148
Will the AWS JSON protocol be deprecated? 148
Where can I find more information about Smithy RPC v2 CBOR? 149
Document history 150
Earlier updates 163

What is AWS Health?

AWS Health provides ongoing visibility into your resource performance and the availability of your AWS services and accounts. You can use AWS Health *events* to learn how service and resource changes might affect your applications running on AWS. AWS Health provides relevant and timely information to help you manage events in progress. AWS Health also helps you be aware of and to prepare for planned activities. The service delivers alerts and notifications triggered by changes in the health of AWS resources, so that you get near-instant event visibility and guidance to help accelerate troubleshooting.

All customers can use the AWS Health Dashboard, powered by the AWS Health API. The dashboard requires no setup, and it's ready to use for authenticated AWS users. For more service highlights, see the AWS Health Dashboard detail page.

AWS Health provides a console, called the AWS Health Dashboard, to all customers. You do not need to write code or perform any actions to set up the dashboard.

To learn the basics about AWS Health and terms that you will encounter while you use the service, To understand the basics of AWS Health see Concepts for AWS Health.

Notes

- The AWS Health Dashboard is available for all AWS customers at no additional cost.
- All AWS customers can receive AWS Health events through Amazon EventBridge at no additional cost.
- If you have an AWS Business Support+, AWS Enterprise Support, or AWS Unified Operations plan, you can use the AWS Health API to integrate with in-house and third-party systems. If you're in an AWS Region that doesn't offer one of these AWS Support plans, or if you haven't transitioned to one of these plans, you can use the AWS Health API with a Business, Enterprise On-Ramp, or Enterprise Support plan. For more information, see the AWS Health API Reference.
- For more information about available AWS Support plans, see AWS Support.

Concepts for AWS Health

Learn about AWS Health concepts and understand how you can use the service to maintain the health of your applications, services, and resources in your AWS account.

Topics

- AWS Health event
- AWS Health Dashboard
- Event type code
- Event type categories
- Event status
- Actionability
- Personas
- Affected entities
- AWS Health events on Amazon EventBridge
- AWS Health API
- Organizational view
- AWS User Notifications

AWS Health event

AWS Health events, also known as Health events, are notifications that AWS Health sends on behalf of other AWS services. You can use these events to learn about upcoming or scheduled changes that might affect your account. For example, AWS Health can send an event if AWS Identity and Access Management (IAM) plans to deprecate a managed policy or AWS Config plans to deprecate a managed rule. AWS Health also sends events when there are service availability issues in an AWS Region. You can review the event description to understand the issue, identify any affected resources, and take any recommended actions.

There are two types of Health events:

Contents

- Account-specific event

- Public event

Account-specific event

Account-specific events are local to either your AWS account or an account in your AWS organization. For example, if there's an issue with an Amazon Elastic Compute Cloud (Amazon EC2) instance type in a Region that you use, AWS Health provides information about the event and the name of the affected resources.

You can find account-specific events from your AWS Health Dashboard, the AWS Health API, or use Amazon EventBridge or AWS User Notifications to receive notifications.

Public event

Public events are reported service events that aren't specific to an account. For example, if there's a service issue for Amazon Simple Storage Service (Amazon S3) in the US East (Ohio) Region, AWS Health provides information about the event, even if you don't use that service or have S3 buckets in that Region. We recommend that you review public notifications before you take action on them.

You can find public events from your AWS Health Dashboard and the AWS Health Dashboard – Service health.

> **Important**
>
> Public events are not available through AWS managed notifications in AWS User Notifications. To receive public events, you must create an Amazon EventBridge rule or a User-configured notification (UCN) within AWS User Notifications. For more information, see Monitoring events in AWS Health with Amazon EventBridge and Managing notifications in AWS User Notifications.

If you have an account, see Getting started with your AWS Health Dashboard.

If you don't have an account, see AWS Health Dashboard.

AWS Health Dashboard

If you have an AWS account, your AWS Health Dashboard shows both *public* events and *account-specific* events.

We recommend that you use your AWS Health Dashboard to learn about events that provide general awareness, such as an upcoming maintenance issue for a service in a Region. You can also use the AWS Health Dashboard to learn about events that might affect you directly, such as a deprecated resource in your account.

You can sign in to the AWS Management Console to view your AWS Health Dashboard at https://health.aws.amazon.com/health/home.

For more information, see Getting started with your AWS Health Dashboard.

AWS Health Dashboard – Service health

If you don't have an account, you can use the AWS Health Dashboard – Service health at https://health.aws.amazon.com/health/status to view public events. Public events are reported service issues for AWS that provide information about service availability. This website only shows public events, which aren't specific to any account. You don't need to sign in or have an account to view this page.

For more information, see AWS Health Dashboard.

Event type code

The event type codes shown in a Health event include the affected service and the type of event. For example, if you receive a Health event that has the `AWS_EC2_SYSTEM_MAINTENANCE_EVENT` event type code, this means that the service is scheduling a maintenance event that might affect you. Use this information to plan ahead or take action for your account.

Event type categories

All Health events have an associated event type category. For some events, the event type category might appear in the event type code, such as the `AWS_RDS_MAINTENANCE_SCHEDULED` code. In this example, the category is *scheduled*. You can use this information to understand event categories at a high level.

It's a best practice that you monitor all event type categories. Note that each category appears for different types of events. You can also use the DescribeEventTypes API operation to find the event type category.

Account notification

These events provide information about the administration or security of your accounts and services. These events might be informative, or they might require urgent action from you. We recommend that you pay attention for these types of events and review all recommended actions.

The following are example event type codes for account notifications:

- `AWS_S3_OPEN_ACCESS_BUCKET_NOTIFICATION` – You have an Amazon S3 bucket that might allow public access.
- `AWS_BILLING_SUSPENSION_NOTICE` – Your account has outstanding charges and has been suspended, or you deactivated your account.
- `AWS_WORKSPACES_OPERATIONAL_NOTIFICATION` – There's a service issue for Amazon WorkSpaces.

Issue

These events are unexpected events that affect AWS services or resources. Common events in this category include communications about operational issues that are causing service degradation, or localized resource-level issues for your awareness.

The following are example event type codes for issues:

- `AWS_EC2_OPERATIONAL_ISSUE` – An operational issue for a service, such as delays in using a service.
- `AWS_EC2_API_ISSUE` – An operational issue for a service's API, such as increased latency for an API operation.
- `AWS_EBS_VOLUME_ATTACHMENT_ISSUE` – A localized resource-level issue that might affect your Amazon Elastic Block Store (Amazon EBS) resources.
- `AWS_ABUSE_PII_CONTENT_REMOVAL_REPORT` – This event means that your account might be suspended if you don't take action.

Scheduled change

These events provide information about upcoming changes to your services and resources. These events include planned lifecycle events such as end-of-support notifications and auto-upgrades for different versions. Some events might recommend that you take action to avoid service disruptions, while others will occur automatically without any action on your part. Your

resource might be temporarily unavailable during the scheduled change activity. All events in this category are account-specific events.

The following are example event type codes for scheduled changes:

- `AWS_EC2_INSTANCE_RETIREMENT_SCHEDULED` – An Amazon EC2 instance requires a reboot.
- `AWS_SAGEMAKER_SCHEDULED_MAINTENANCE` – SageMaker AI requires a maintenance event, such as fixing a service issue.
- `AWS_RDS_PLANNED_LIFECYCLE_EVENT` – Amazon RDS is scheduling a planned lifecycle event, such as an end-of-support event for one of its versions, which requires customer action.

> **Tip**
>
> If you use the AWS Health API or the AWS Command Line Interface (AWS CLI) to return event details, the `Event` object contains the `eventScopeCode` field with the `ACCOUNT_SPECIFIC` value. For more information, see the AWS Health API Reference.

Event status

The event status tells you if the Health event is open, closed, or upcoming. You can view Health events in the AWS Health Dashboard or the AWS Health API for up to 90 days.

Actionability

Actionability is a field that helps you prioritize Health events based on whether an action is required from you. Health events include an actionability status that indicates if you need to take action to mitigate risks to your AWS resources or if the event is informational in nature.

The actionability field can contain one of the following values:

- `ACTION_REQUIRED`: Events with this status require action from you to mitigate potential impact related to availability, billing, or security of your AWS resources.
- `ACTION_MAY_BE_REQUIRED`: Events with this status communicate changes that require action, based on your specific implementation, dependencies, and workflows. These events require your review to determine if action is needed.

- `INFORMATIONAL`: Events with this status provide ongoing visibility into operational information about the AWS services that you use. No immediate action is expected.

> **Note**
>
> Health events related to service issues don't include an actionability label, as the need for recovery actions depends on your specific application architecture.

Personas

The personas field provides a list of contacts that helps you route relevant information to appropriate teams within your organization. Each Health event can include one or more of the following personas:

- `OPERATIONS`: For events related to operational activities and service availability.
- `SECURITY`: For events related to security considerations.
- `BILLING`: For events with potential cost implications.

For example, when AWS sends an event about the end of standard support that converts into extended support, the event includes `BILLING` in addition to `OPERATIONS` within the persona list to help make sure the information reaches teams responsible for cost management.

Affected entities

Affected entities are AWS resources that might be affected by the event. For example, if you receive a scheduled event for Amazon EC2 maintenance for a specific instance type that you're using in your account, you can use the Health event to determine the ID of the affected instances. Use this information to address any potential service issue, such as creating or deprecating resources.

AWS Health events on Amazon EventBridge

You can setup Amazon EventBridge rules for your accounts to automate actions after the appropriate AWS Health event is received by an account. These can be general actions, such as sending all planned lifecycle event messages to a chat interface. Or, they can be specific actions, such as triggering a workflow in an IT service management tool.

For more information, see Monitoring events in AWS Health with Amazon EventBridge.

AWS Health API

You can use the AWS Health API to programmatically access the information that appears in the AWS Health Dashboard, such as the following:

- Get information about events that might affect your AWS services and resources
- Enable or disable the organizational view feature for your AWS organization
- Filter your events by specific services, event type categories, and event type codes

For more information, see the AWS Health API Reference.

> **Note**
>
> You must have a AWS Business Support+, AWS Enterprise Support, or AWS Unified Operations plan from AWS Support to use the AWS Health API. If you call the AWS Health API from an account that doesn't have a AWS Business Support+, AWS Enterprise Support, or AWS Unified Operations plan, you receive a `SubscriptionRequiredException` error.

Organizational view

You can use this feature to aggregate all health events for AWS accounts in your AWS Organizations into a single view in the AWS Health Dashboard. You can then sign in to the management account of your organization or use the AWS Health API to view all events that might affect the different accounts and resources. You can enable this feature from the AWS Health console or API. For more information, see Aggregating AWS Health events across accounts.

AWS User Notifications

AWS Health integrates with AWS User Notifications so that you can easily receive and control notifications about events affecting your AWS accounts and services. User Notifications offers managed notifications for AWS Health events by default. You can configure these subscriptions to control how often you receive messages through time-based aggregation, what kinds of AWS Health events you get notified about, and where notifications are delivered. To get started, open

User Notifications in the AWS Management Console. For more information, see Manage AWS Health notifications in AWS User Notifications

Getting started with your AWS Health Dashboard

You can use your AWS Health Dashboard to learn about AWS Health events. These events can affect your AWS services or AWS account. After you sign in to your account, the AWS Health Dashboard shows information in the following ways:

- **Your account events** – This page shows events that are specific to your account. You can view open, recent, and scheduled changes. You can also view notifications and an event log that shows all events from the past 90 days.
- **Your organization events** – This page shows events that are specific to your organization in AWS Organizations. You can view open, recent, and scheduled changes for your organization. You can also view notifications, as well as an event log that shows all organization events from the past 90 days.

> **Note**
>
> If you don't have an AWS account, you can use the AWS Health Dashboard to learn about general service availability.
> If you have an account, we recommend that you sign in to your AWS Health Dashboard to get deeper insights into events and upcoming changes that might affect your services and resources.

Topics

- Sign up for an AWS account
- Viewing your account events in the AWS Health Dashboard
- Configuring Amazon EventBridge
- Manage AWS Health notifications in AWS User Notifications

Sign up for an AWS account

To get started with AWS, you need an AWS account. There are two ways to sign up for AWS. If you want complete control over account configuration and where your content is stored, or if you have regulated workloads such as HIPAA or FedRAMP, use Sign up for AWS (advanced). If you are a new

user and want to use a login you already own to sign in to AWS and create an AWS environment with preconfigured defaults in select AWS Regions, use Sign up for AWS.

Viewing your account events in the AWS Health Dashboard

You can sign in to your account to get personalized events and recommendations.

To view account events in your AWS Health Dashboard

1. Open your AWS Health Dashboard at https://health.aws.amazon.com/health/home.
2. In the navigation pane, for **Your account health**, you can choose the following options:
   a. **Open and recent issues** – View recently opened and closed events.
   b. **Scheduled changes** – View upcoming events that might affect your services and resources.
   c. **Other notifications** – View all other notifications and ongoing events from the past seven days that might affect your account.
   d. **Event log** – View all events from the past 90 days.

Open and recent issues

Use the **Open and recent issues** tab to view all ongoing events from the past seven days that might affect your account.

When you choose an event from the dashboard, the **Details** pane appears with information about the event and a list of affected resources. For more information, see Event details.

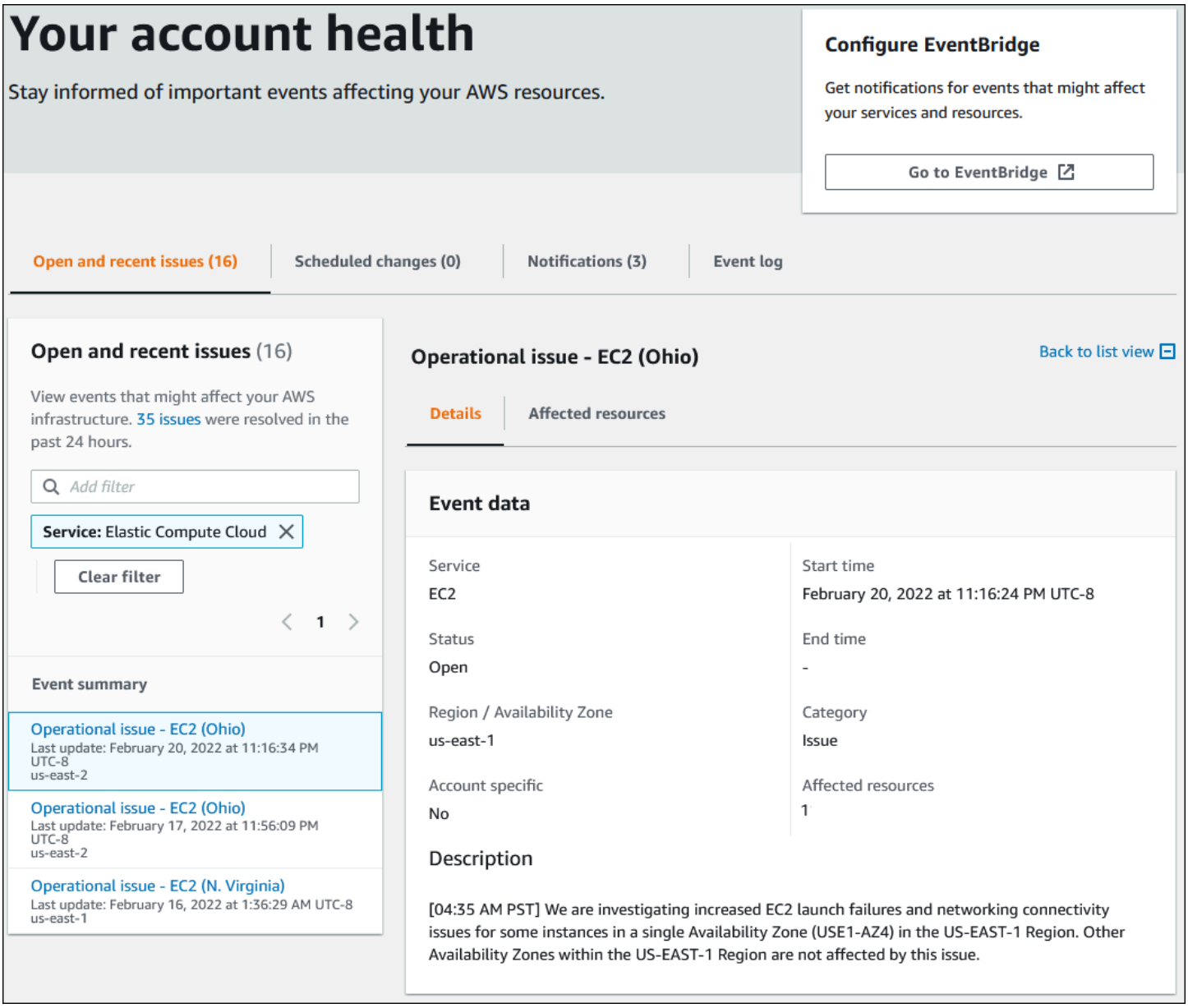
You can filter the events that appear in any tab by choosing options from the filter list. For example, you can narrow the results by Availability Zone, Region, event end time or last update time, AWS service, and so on.

To see all the events, rather than the recent ones that appear in the dashboard, choose the **Event log** tab.

> **Note**
>
> Currently, you can't delete notifications for events that appear in your AWS Health Dashboard. After an AWS service resolves an event, the notification is removed from your dashboard view.

Example: Operational issue event for Amazon Elastic Compute Cloud (Amazon EC2)

The following image shows an event for launch failures and connectivity issues for Amazon EC2 instances.



Scheduled changes

Use the **Scheduled changes** tab to view upcoming events that might affect your account. These events can include scheduled maintenance activities for services and planned lifecycle events that require action to resolve. To help you plan for these activities, a calendar view is provided so that you can map these scheduled changes into a monthly calendar. Filters are available. For more information about planned lifecycle events, see Planned lifecycle events for AWS Health.

Other notifications

Use the **Notifications** tab to view all other notifications and ongoing events from the past seven days that might affect your account. This can include events, such as certificate rotations, billing notifications, and security vulnerabilities.

Event log

Use the **Event log** tab to view all AWS Health events. The log table includes additional columns so that you can filter by **Status** and **Start time**.

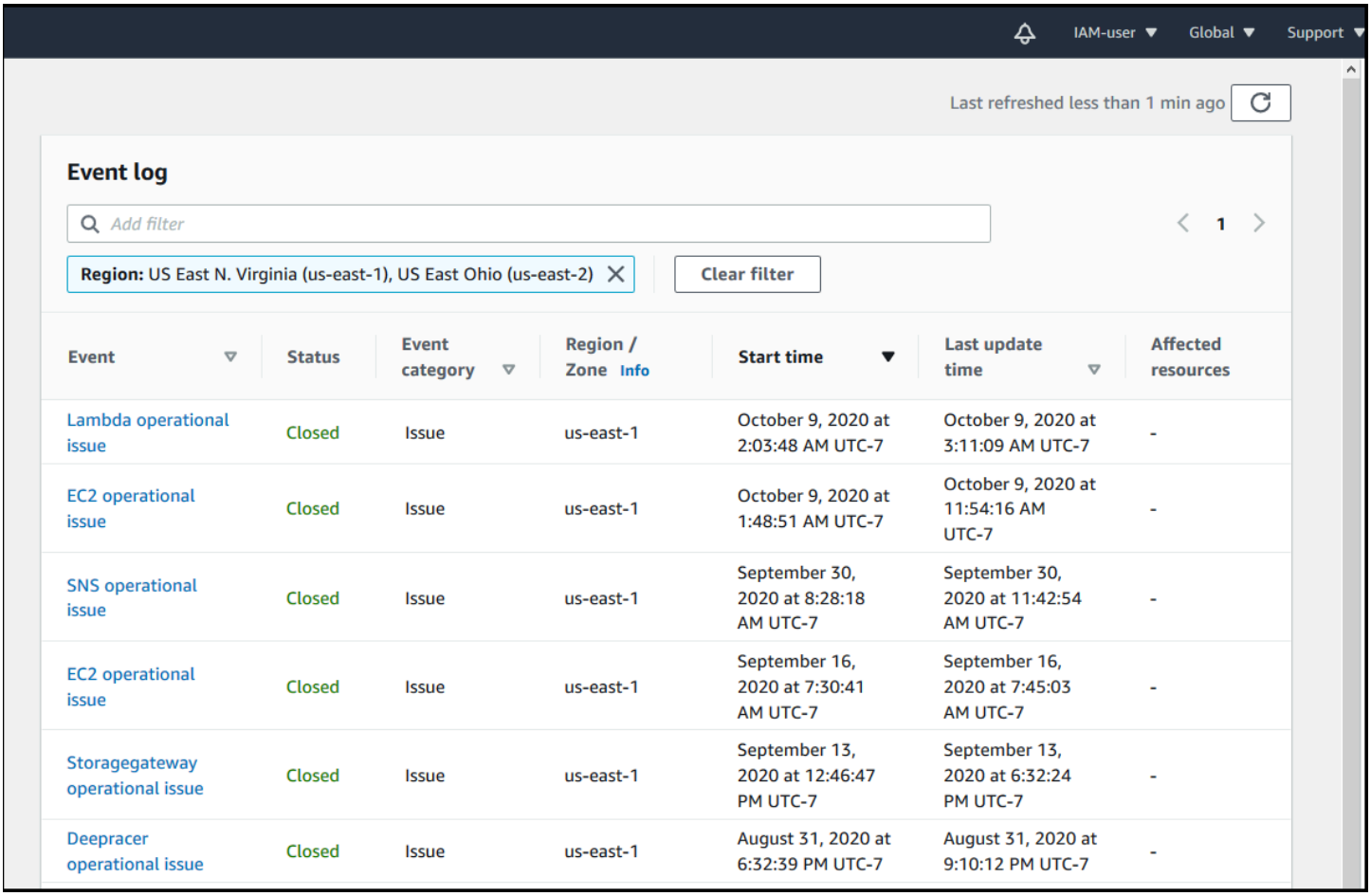
When you choose an event in the **Event log** table, the **Details** pane appears with information about the event and the list of affected resources. For more information, see Event details.

You can choose the following filter options to narrow your results:

- Availability Zone
- End time
- Event
- Event ARN
- Event category
- Last update time
- Region
- Resource ID / ARN
- Service
- Start time
- Status

Example: Event log

The following image shows recent events for the US East (N. Virginia) and US East (Ohio) Regions.



Event details

When you choose an event, two tabs appear about the event. The **Details** tab shows the following information:

- Service
- Status
- Region / Availability Zone
- Whether or not the event is account specific
- Start and end time
- Category
- Number of affected resources
- Description and a timeline of updates about the event

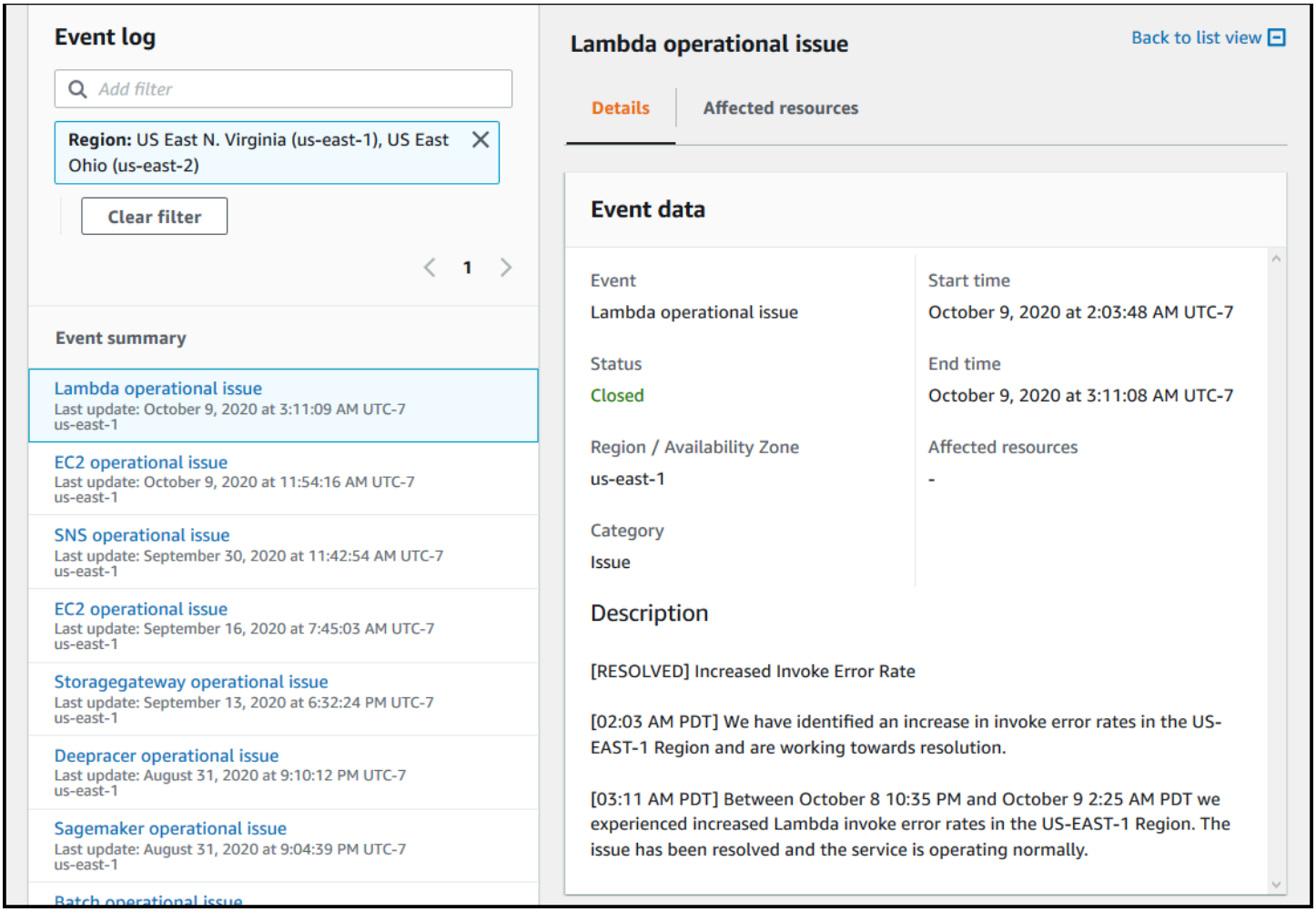
The **Affected resources** tab shows the following information about any AWS resources that are affected by the event:

- The resource ID (for example, an Amazon EBS volume ID such as `vol-a1b2c34f`) or Amazon Resource Name (ARN), if available or relevant.
- For planned lifecycle events, this affected resources list also contains the latest status of the resources (**Pending**, **Unknown**, or **Resolved**). This list usually refreshes once every 24 hours, but might take up to 72 hours to reflect the current status.

You can filter the items that appear in the resources. You can narrow your results by resource ID or ARN.

Example: AWS Health event for AWS Lambda

The following screenshot shows an example event for Lambda.



Events types

There are two types of AWS Health events:

- *Public events* are service events that aren't specific to an account. For example, if there is an issue with Amazon EC2 in an AWS Region, AWS Health provides information about the event, even if you don't use services or resources in that Region.
- *Account-specific* events are specific to your account or an account in your organization. For example, if there's an issue with an Amazon EC2 instance in an AWS Region that you use, AWS Health provides information about the event and the list of affected Amazon EC2 instances.

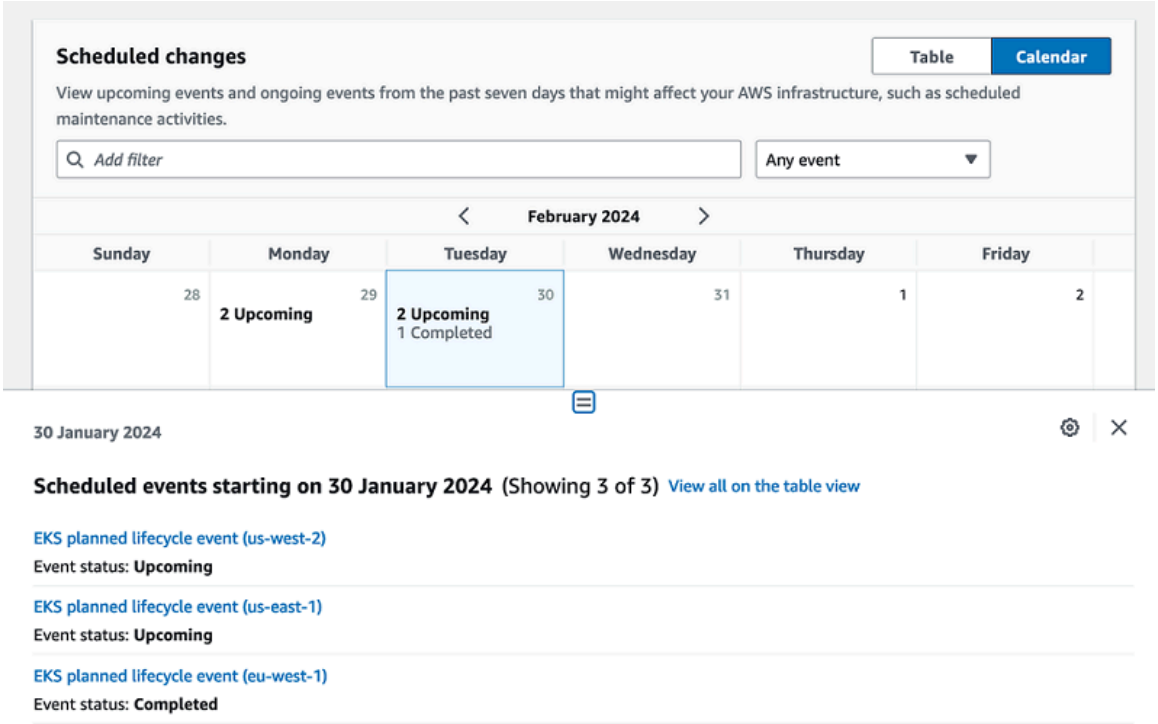
You can use the following options to identify if an event is public or account-specific:

- In the AWS Health Dashboard, choose the **Affected resources** tab for an event. Events with resources are specific to your account. Events without resources are public and are not specific to your account. For more information, see Getting started with your AWS Health Dashboard.
- Use the AWS Health API to return the `eventScopeCode` parameter. Events can have the `PUBLIC`, `ACCOUNT_SPECIFIC`, or `NONE` value. For more information, see the DescribeEventDetails operation in the *AWS Health API Reference*.

Calendar view

Calendar view is available in the **scheduled changes** tab to project AWS Health events into a monthly calendar. This view allows you to see scheduled changes up to 3 months into the past and a year into the future.

AWS Health events are displayed by date. Select a date to display a side panel that contains further details on the AWS Health event. **Upcoming** and **ongoing** events are displayed in black. **Completed** events are displayed in grey. If there are more than two events in a date, only the number of black and grey events are shown. Select a date to display a list of AWS Health events in the side panel. You can select an event in the side panel to display information about the event. The side panel has breadcrumbs to navigate to an earlier view.



Affected resources view

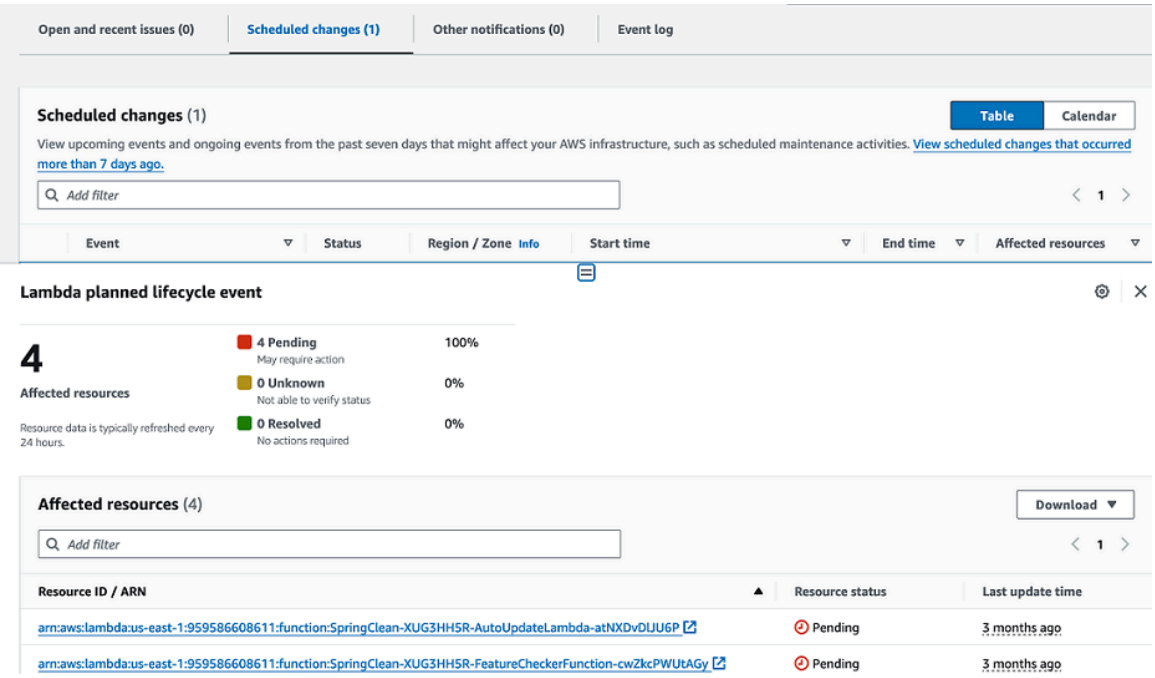
AWS Health events might specify the precise resources that are affected. You can view affected resources in the **Affected Resources** tab of the AWS Health event. To view the status, select the AWS Health event. The status displays in the affected resources tab in the side panel. For planned lifecycle events, AWS Health events provide daily updates of the affected resources' status.

Account-level AWS Health events display a summary of affected resources statuses at the top of the **Affected Resources** tab. A list of affected resources is displayed in a table along with the corresponding status. Planned lifecycle events are an example of event types that use the resource status field. To learn more about planned lifecycle events, see Planned lifecycle events for AWS Health.

When you access the organization view, AWS Health events display a summary of the status of all affected resources for all included accounts. After the summary is a list of affected accounts and the number of pending resources for that account. Select the account number or the number of pending resources to display the account view summary. The account view summary has breadcrumbs to navigate back to the organizational list of affected accounts. A summary of affected resource statuses is displayed at the top of the split panel.

You can download the list of affected resources in the affected resources tab in CSV or JSON format. In organizational view, the downloaded file includes all resources in the accounts listed. Navigate to the account level in organizational view to include only resources for that account in the downloaded file. Each affected resource in the downloaded file includes the AWS account ID, the eventARN, the entity name, the entityARN, status, and the last updated time of the resource. If filters are activated, the downloaded file only includes the filtered results.

You can download only one file at a time. The files are automatically downloaded into the default download folder of your browser and have a preset file name based on the AWS Region, the event title, the event start date, and the download date.



Time zone settings

You can view the events in the AWS Health Dashboard in your local time zone or in UTC. If you change the time zone in your AWS Health Dashboard, all timestamps in the dashboard and public events update to the time zone that you specify.

To update your time zone settings

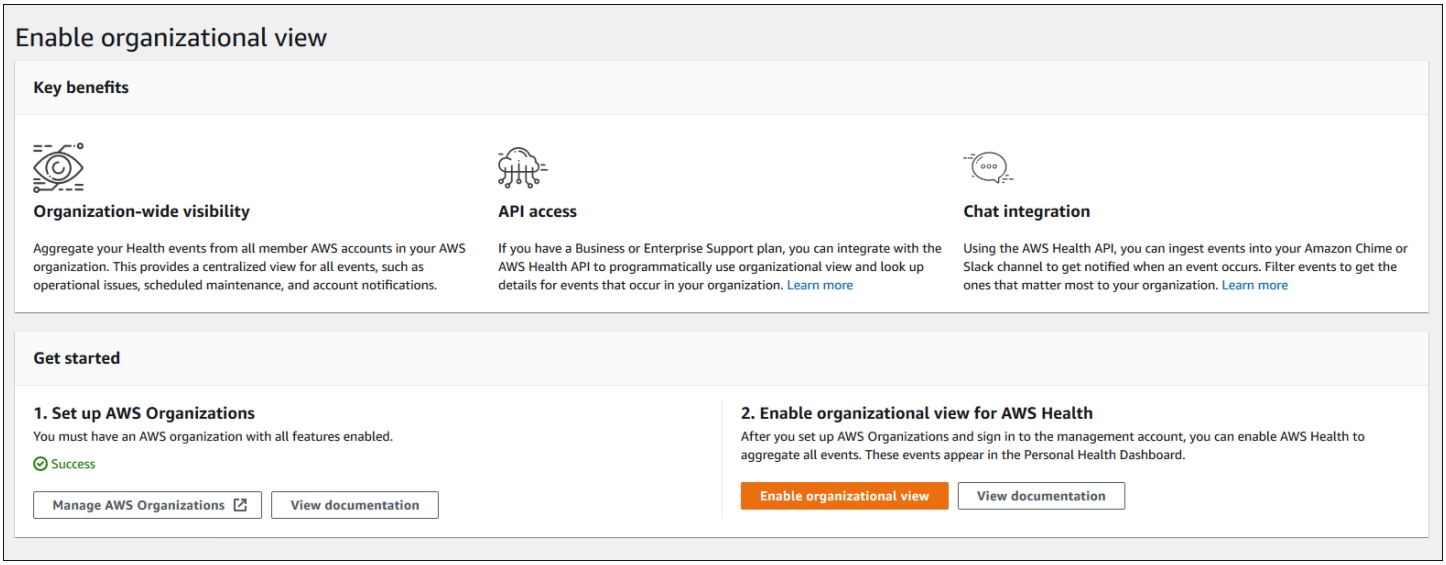
1. Open your AWS Health Dashboard at https://health.aws.amazon.com/health/home.
2. At the bottom of the page, choose **Cookie preferences**.
3. Select **Allowed** for Functional cookies. Then choose **Save preferences**.
4. In the navigation pane of your AWS Health Dashboard, choose **Time zone settings**.

5. Select a time zone for your AWS Health Dashboard sessions. Then choose **Save changes**.

Your organization health

AWS Health integrates with AWS Organizations so that you can view events for all accounts that are part of your organization. This provides you a centralized view for events that appear in your organization. You can use these events to monitor for changes in your resources, services, and applications.

For more information, see Aggregating AWS Health events across accounts.



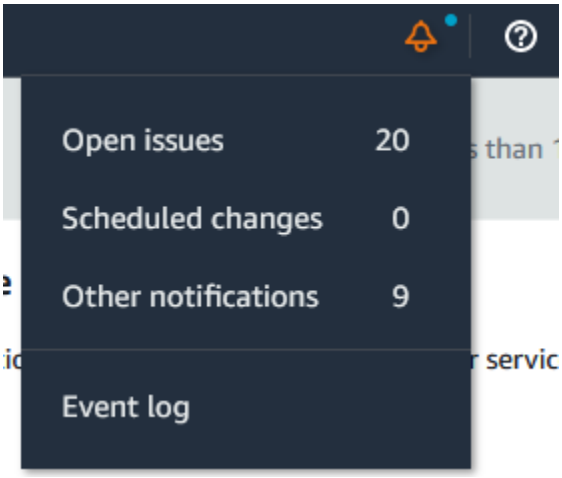
Alerts for AWS Health events

Your AWS Health Dashboard has a bell icon in the console navigation bar with an alert menu. This feature displays the number of recent AWS Health events that appear on the dashboard in each category. This bell icon appears on several AWS consoles, such as those for Amazon EC2, Amazon Relational Database Service (Amazon RDS), AWS Identity and Access Management (IAM), and AWS Trusted Advisor.

Choose the bell icon to see if recent events affect your account. You can then choose an event to navigate to your AWS Health Dashboard for more information.

Example: Open events

The following image shows open and notification events for an account.



Configuring Amazon EventBridge

Use EventBridge to detect and react to changes for AWS Health events. You can monitor specific AWS Health events that occur in your account, and then set up rules so that AWS Health notifies you, or you take action, when events change.

Use EventBridge with AWS Health

1. Open your AWS Health Dashboard at https://health.aws.amazon.com/health/home.
2. To navigate to the EventBridge console to create a rule, do one of the following:
   - From the navigation pane, under **Health Integrations**, choose **Amazon EventBridge**.
   - Under **Configure EventBridge**, choose **Go to EventBridge**.
3. Follow this procedure to create rules and monitor for events. See Monitoring events in AWS Health with Amazon EventBridge.

Manage AWS Health notifications in AWS User Notifications

Important

Public events are not available through AWS managed notifications in AWS User Notifications. To receive public events, you must create an Amazon EventBridge rule or a User-configured notification (UCN) within AWS User Notifications. For more information, see Monitoring events in AWS Health with Amazon EventBridge and Managing notifications in AWS User Notifications.

AWS managed notifications in AWS User Notifications lets you receive and manage notifications about events that affect your AWS accounts and services. When you use AWS managed notifications in AWS User Notifications, you can specify which AWS Health event categories to receive, set up organizational view for emails, and get consolidated notifications instead of multiple similar emails.

You can choose the following additional channels to receive your AWS Health events through AWS User Notifications:

- Email
- Chat
- Push notifications to the AWS Console Mobile Application

While these notifications aren't as detailed as direct AWS Health tools, they provide an effective way to notify stakeholders of issues and changes.

> **Note**
>
> For comprehensive visibility into AWS Health event details, including affected resource IDs, current status (open or closed), and resource status, it's a best practice to use one of the following AWS Health tools:
>
> - The AWS Health API
> - The aws.health source in Amazon EventBridge
> - The Health Dashboard
>
> These tools provide the most detailed and real-time information about ongoing events and changes that might affect your workloads.

Configure your AWS managed notifications subscription for AWS Health events

To configure your AWS managed notifications subscription, complete the following steps:

1. Open User Notifications in the AWS Management Console.
2. In the navigation pane, choose **AWS managed notifications subscriptions**.

3. You can manage your AWS Health event notifications by category. For more information, see Adding and removing account contacts for AWS managed notifications in AWS User Notifications.

> **Note**
>
> AWS Health migrated email delivery to AWS managed notifications in AWS User Notifications. Since December 15, 2025, you receive emails from AWS managed notifications. For more information, see *What changed in the migration to AWS managed notifications?* in the AWS managed notifications in AWS User Notifications FAQ.

AWS managed notifications in AWS User Notifications FAQ

What changed in the migration to AWS managed notifications?

By default, emails regarding managed notifications are sent to your existing account contacts (root, operations, billing, and security email addresses). The emails that you receive from AWS managed notifications come from `health@aws.com` instead of `no-reply-aws@amazon.com`, and the format of the email changes. If you previously set up email rules for AWS Health notifications, such as routing an email by sender ID or scraping the content of the email, then you must update this setup to match the new email format. If you require automation through push notifications, then we recommend that you evaluate AWS Health events sent through Amazon EventBridge as an alternative to managed notifications.

How does aggregation work for emails and how do I enable this feature?

AWS managed notification aggregates AWS Health events that impact multiple accounts within the same AWS Organizations organization into a single aggregated notification. You can view the aggregated organization in the management account's notifications center. Managed notifications emails the aggregated notification to the management account's contacts. To reduce duplicate emails, AWS managed notifications sends one notification when account contacts are shared between management and member accounts.

To enable aggregation, you must have AWS Organizations configured and grant trusted access between your management account and the AWS User Notifications service.

For more information, see AWS managed notifications aggregation in AWS User Notifications.

Do I have to enable AWS Organizations trusted access with AWS User Notifications to receive aggregated email from AWS managed notifications?

Yes, trusted access with AWS User Notifications from AWS Organizations is required.

What's the difference between enabling trusted access through AWS Organizations with AWS Health and with AWS User Notifications?

Organizational trust and the associated delegated administrator privileges are assigned by service and act as guardrails against overextended permissions. Trusted access for AWS Health enables organizational view for the Health Dashboard, the AWS Health APIs, AWS Health events sent through Amazon EventBridge, and notification configurations in User Notifications. Trusted access for AWS User Notifications enables aggregate notifications within AWS managed notifications. Because trusted access isn't shared, setting up delegated administrators needs to be added separately for each service.

Is there a way to keep plain text emails for my specific use case?

No. The current plain text AWS Health emails are disabled after the migration completes. If you use email rules to drive different workflows, we recommend that you evaluate AWS Health events sent through Amazon EventBridge as an alternative.

What do AWS managed notifications categories correspond to in the AWS Health schema?

Health operations, Security, and Billing notifications correspond to AWS Health account notifications and scheduled changes that have the operations, security, and billing persona respectively. AWS Health events with more than one persona tag are sent through the Security and Billing categories. Account-specific issues include issue category health events that are specific to an AWS account.

Public service events aren't available through AWS managed notifications.

AWS Health Dashboard

You can use the AWS Health Dashboard – Service health to view the health of all AWS services. This page shows reported service events for services across AWS Regions. You don't need to sign in or have an AWS account to access the AWS Health Dashboard – Service health page.

> **Tip**
>
> This website only shows *public* events, which are not specific to an AWS account. If you already have an account, we recommend that you sign in to view your AWS Health Dashboard and stay informed about events that can affect your account and services. For more information, see Getting started with your AWS Health Dashboard.

To view the AWS Health Dashboard – Service health

1. Navigate to the https://health.aws.amazon.com/health/status page.

   > **Note**
   >
   > If you are already signed in to your AWS account, page, you will be redirected to the **AWS Health Dashboard – Your account health** page.

2. Under **Service health**, choose **Open and recent issues** to view recently reported events. You can view the following information about the event:

   - The event name and affected Region. For example, **Operational issue – Amazon Elastic Compute Cloud (N. Virginia)**
   - The service name
   - The event's severity, such as **Impacted** or **Degraded**
   - A timeline of recent updates for the event
   - A list of AWS services that are also affected by this event

> **Note**
>
> You can view the events in your local time zone or in UTC. For more information, see Time zone settings.

3. Choose **Service history** to view the **Service history** table. This table shows all AWS service interruptions for the last 12 months.

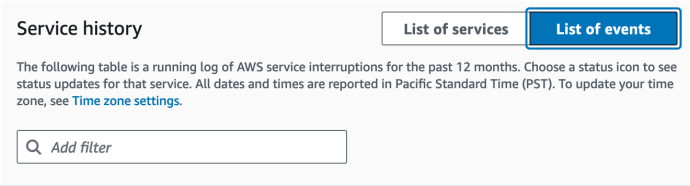
> **Tip**
>
> You can filter by **Service**, **AWS Region**, and date.

4. Next to an ongoing service event, choose the status icon

   ( )

   to view more information about the event.

5. (Optional) To view this as a list of historical events, choose the list of events button. Choose any event in the event column to view more information about that specific event in the pop-up side-panel.



> **Note**
>
> Selecting any public event after September 2023 will populate the URL in the browser with a link to that public AWS Health event. After you select this link, you navigate to the list of events view with that event pop-up.

6. (Optional) You can view the events in your local time zone or UTC. For more information, see Time zone settings.

7. (Optional) If you have an account, choose **Open your account health** to sign in. After you sign in, you can view events that are specific to your account. For more information, see Getting started with your AWS Health Dashboard.

Note

Although an RSS feed is available for health events, the format is subject to changes. So, scraping the RSS feed might not provide all the relevant data. To programmatically ingest health event data, we recommend integrating with Amazon EventBridge. For more information, see Monitoring events in AWS Health with Amazon EventBridge.

Planned lifecycle events for AWS Health

Learn about planned lifecycle events for AWS Health.

Topics

- What are planned lifecycle events?
- What should I expect when I receive a planned lifecycle event notification?
- Shared responsibility model for resilience
- Accessing planned lifecycle events

What are planned lifecycle events?

AWS Health communicates important changes that can affect the availability of your applications. In the AWS shared responsibility model, AWS takes action to keep the underlying hardware and infrastructure that supports your resources up to date and secure. However, some changes require customer action or coordination in order to avoid impact to your applications. AWS Health notifies you in advance of important changes such as:

- Open source software end of support - Some AWS services run open source versions of software. If the open source community ends support for software versions, then AWS informs you when you need to take action to upgrade and avoid impact to your applications.
  - Amazon RDS for MySQL engine version end of support
  - Amazon EKS Kubernetes version end of support
- Changes that affect AWS-owned resources that might require your action.
  - Amazon RDS Certificate Authority certificates expiration.

> **Note**
>
> All notifications that fit this criteria will be reported through AWS Health as Planned Lifecycle Events.

- **Dynamic resource burndown and improved metadata**: From the time you receive the notification through the lifespan of the AWS Health event, your affected resources are associated with the AWS Health event as affected entities with a specific entity status. Affected resources are specified in ARN format, where applicable. If your affected resource(s) require customer

action, then they are listed with a “PENDING” status. If your affected resource(s) had the requisite action performed or the resources were deleted, then the status is updated to “RESOLVED”.

Note

- Resource state updates are performed asynchronously and periodically and can have a delay of up to 72 hours in rare occasions.
- In the exceptions where dynamic updates are not provided, rather than resources having a “PENDING” or “RESOLVED” status, resources will not be assigned any status.
- Resource status updates are not supported in the AWS GovCloud (US) and China Regions.

What should I expect when I receive a planned lifecycle event notification?

The AWS Health experience for planned lifecycle events helps your teams learn about upcoming lifecycle changes and track action completion.

Type category: Scheduled change

Event type code: AWS_{SERVICE}_PLANNED_LIFECYCLE_EVENT

Event start time: Event start time is the soonest date at which your resources are affected by the change.

Event end time: Event end time is the date that the change finishes across all AWS resources. Note that end time is not always specified. It is important to treat the start time as the change date.

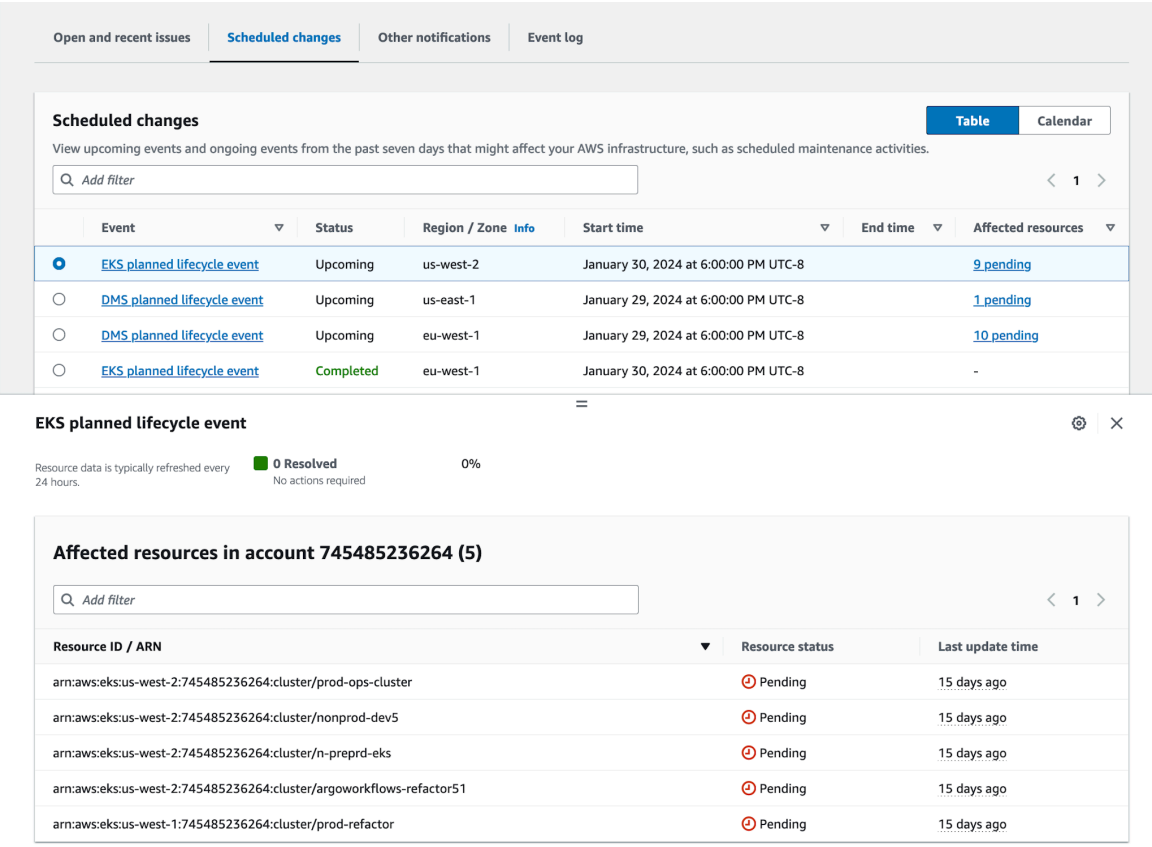
Note

Organizations can expect to receive a single event ARN for every planned lifecycle event grouped by Region where there are affected resources. But they might receive multiple ARNs if the organization has a large number of affected AWS accounts or resources.

Early visibility into planned lifecycle events: Planned lifecycle events are designed to have a minimum lead time of 180 days for major versions/changes and 90 days for minor versions/changes, where possible.

Dynamic resource burndown and improved metadata: From the time you receive the notification through the lifespan of the AWS Health event, your affected resources are associated with the AWS Health event as affected entities with a specific entity status. Affected resources are specified in ARN format, where applicable. If your affected resource(s) require customer action, then they are listed with a “PENDING” status. If your affected resource(s) had the requisite action performed or the resources were deleted, then the status is updated to “RESOLVED”.

> **Note**
>
> - AWS Health notifications provide status updates over time where possible, except for the AWS GovCloud (US) and China Regions.
> - Resource state updates are performed asynchronously and periodically and can have a delay of up to 72 hours in rare occasions.

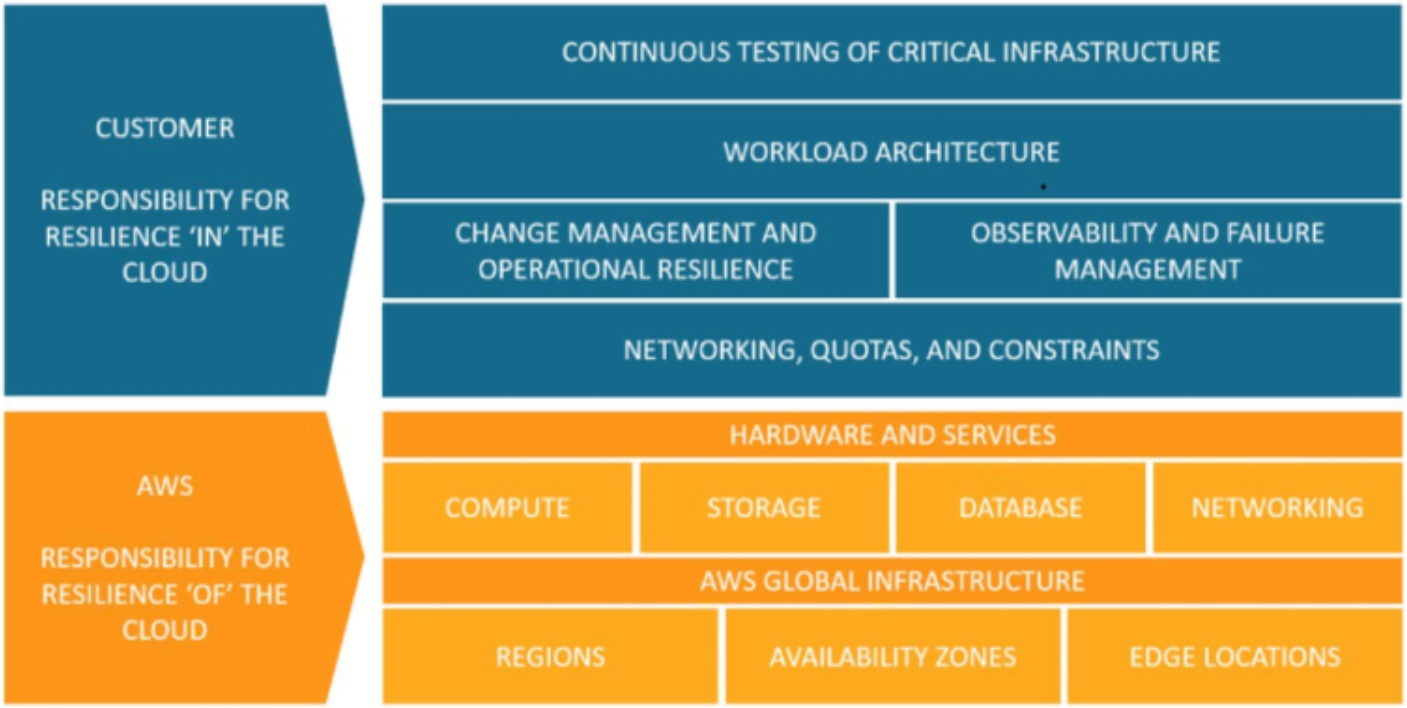


After the planned event date passes:

1. If applicable, the service might implement the described change to your resource any time after the start date of the event.
2. If you resolve all resources prior to the end of support date, then your AWS Health event changes to the status `Closed`.
3. If you have outstanding resources after the change date that aren't resolved, then the AWS Health event remains open for 4 years after the event start or end date (whichever is later). After this time, the AWS Health event is deleted.

Shared responsibility model for resilience

Security and compliance are shared responsibilities between AWS and the customer. Depending on the services deployed, this shared model can help relieve the customer's operational burden. This is because AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates. The customer assumes responsibility and management of the guest operating system (including updates and security patches) and other associated application software, in addition to the configuration of the security group firewall provided by AWS. For more information, see Shared responsibility model.



Accessing planned lifecycle events

Planned lifecycle events can be accessed and monitored using several channels:

- Use Amazon EventBridge
- Use the AWS Health dashboard
  - Calendar view
  - Affected resources view
- Use the AWS Health API

Version catalog for AWS Health

Learn about the version catalog for AWS Health, which provides a service-wide view of supported versions and their timelines.

Topics

- What is the version catalog?
- What you can do with the version catalog
- How to access the version catalog
- Details available in the version catalog
- How to retrieve the version catalog information with the API
- How to view the version catalog in the Health Dashboard

What is the version catalog?

If you run applications on AWS, you must stay current with software versions to maintain a strong security and operational posture. AWS Health already sends account-specific and resource-specific Planned lifecycle events for AWS Health well in advance of major version end-of-support milestones. The version catalog adds a service-wide view of supported versions and their timelines. You can use this view to build upgrade schedules and governance controls, and to move to newer supported versions before you receive an account-specific planned lifecycle event.

The version lifecycle API aggregates this information into a single, consistent schema. With one authenticated API call, you can query the support timeline of a specific release of an AWS service component, such as an AWS Lambda language runtime, an Amazon Relational Database Service (Amazon RDS) engine version, or an Amazon Elastic Kubernetes Service (Amazon EKS) Kubernetes version.

Use this data to build upgrade dashboards, alert on versions that approach end of support, and automate compliance checks for the versions that you run. You can also reduce technical debt by moving to newer supported versions before you receive an account-specific planned lifecycle event.

What you can do with the version catalog

The version lifecycle API gives you programmatic access to understand where every AWS service component sits in its support timeline. Common use cases include the following:

- **Vulnerability and compliance management** – Identify software running past its end of life that no longer receives security patches (for example, Python 3.9 reached end of life in October 2025), and feed end-of-life data into compliance dashboards for PCI DSS, SOC 2, and FedRAMP audits.
- **Infrastructure and fleet inventory auditing** – Cross-reference your configuration management database (CMDB) or resource inventory to flag unsupported database engines (Amazon RDS PostgreSQL, MySQL) or Lambda runtimes (Node.js, Java, .NET). This is especially useful for AWS-managed services, because the catalog tracks Amazon EKS, Amazon Aurora (Aurora) MySQL and PostgreSQL, Amazon RDS, Amazon OpenSearch Service, Amazon ElastiCache, Lambda runtimes, and more.
- **Proactive upgrade planning** – Build migration timelines and programs, and generate roadmaps that show when each dependency exits support relative to its active development phases.
- **CI/CD pipeline automation** – Integrate the API into your pipelines to fail builds or raise warnings when a project depends on a runtime or framework that is at or near end of life. You can also maintain a test matrix automatically (for example, test against all actively supported Python versions) without hardcoding version lists.
- **Developer tooling and dashboards** – Power internal operational dashboards that show the support status of every component used in a specific workload.
- **AWS service lifecycle tracking** – Track AWS-managed service versions, such as Amazon EKS and Lambda, and prepare for forced upgrades before they occur.

How to access the version catalog

You can view the version catalog in two ways:

- **Health Dashboard** – View supported versions and their lifecycle timelines directly in the console.
- **AWS Health API** – Customers on AWS Support can call the `DescribeServiceLifecycle` API to integrate lifecycle data into their operational workflows.

See the AWS Health API Reference for full details.

Details available in the version catalog

Each version in a response provides an overview and a timeline toward the various phases in the lifecycle.

Top-level version entry

Field	Description
`service`	The AWS service that the version belongs to, for example, `LAMBDA`, `RDS`, or `EKS`.
`title`	The human-readable name of the version, for example, `Python 3.14 Lambda Runtime`.
`version`	The service-specific version identifier that you pass in queries, for example, `python3.14` .
`lifecycleEvents`	A list of lifecycle events for this version, in chronological order. Each event marks a significant date, such as a release, an end of standard support, or an end of life, and includes impact-risk tags that indicate the consequence of that date passing. Read the events in order to understand the full timeline from release through end of support.

Lifecycle event object

Field	Description
`lifecycleEventType`	The milestone type, for example, `END_OF_SUPPORT` , `BLOCK_RESOURCE_CREATE` , or `BLOCK_RESOURCE_UPDATE` .
`date`	The date when the milestone takes effect.
`description`	A human-readable explanation of what happens on this date.
`impactRisks`	One or more impact-risk tags, such as `END_OF_SUPPORT` , `BILLING`, or `AVAILABILITY` , that indicate what stops working when the date passes.
`regions`	The Regions that the event applies to. A value of `["all"]` means all Regions.

> **Note**
>
> The values shown for `lifecycleEventType` and `impactRisks` are those visible in this example. See the AWS Health API Reference for the complete list of possible values.

Impact-risk tags

Each event lists one or more impact-risk tags so that you can assess severity. `AVAILABILITY` impacts are the most urgent, because the resource becomes unavailable after that date passes.

Tag	Meaning
`END_OF_SUPPORT`	The version no longer receives security patches or updates.
`BILLING`	Continuing to run this version might incur additional charges, for example, extended-support fees.
`AVAILABILITY`	The resource becomes unavailable after the date passes. This is the most urgent impact.

How to retrieve the version catalog information with the API

Query the lifecycle information by using the AWS Command Line Interface (AWS CLI).

```
aws health describe-service-lifecycle
```

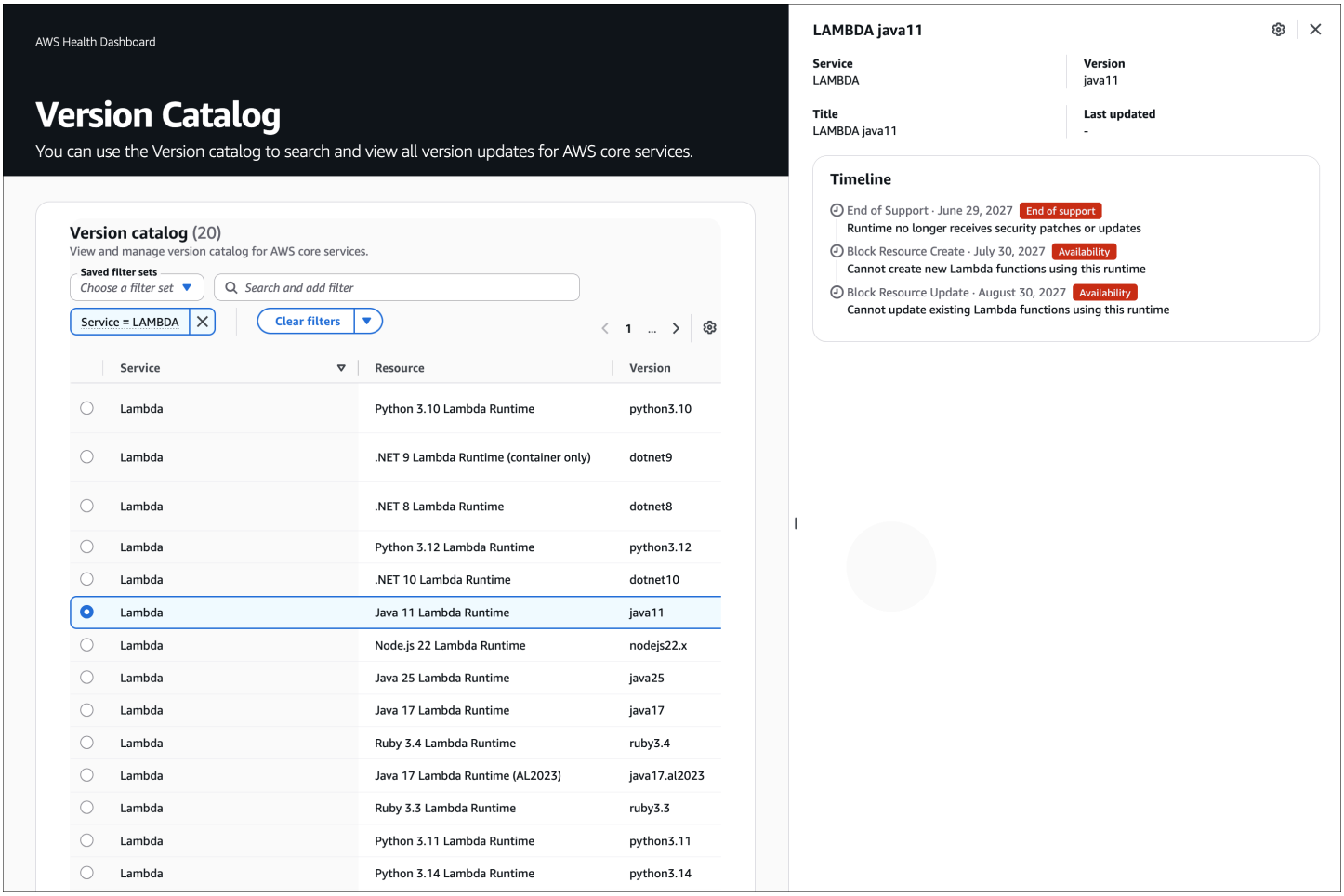
The following example shows a version entry from the `DescribeServiceLifecycle` API.

```
{
  "service": "LAMBDA",
  "title": "Python 3.14 Lambda Runtime",
  "version": "python3.14",
  "lifecycleEvents": [
    {
      "lifecycleEventType": "END_OF_SUPPORT",
      "date": 1877472000.0,
      "description": "Runtime no longer receives security patches or updates",
```

```
      "impactRisks": ["END_OF_SUPPORT"],
      "regions": ["all"]
    },
    {
      "lifecycleEventType": "BLOCK_RESOURCE_CREATE",
      "date": 1880150400.0,
      "description": "Cannot create new Lambda functions using this runtime",
      "impactRisks": ["AVAILABILITY"],
      "regions": ["all"]
    },
    {
      "lifecycleEventType": "BLOCK_RESOURCE_UPDATE",
      "date": 1882828800.0,
      "description": "Cannot update existing Lambda functions using this runtime",
      "impactRisks": ["AVAILABILITY"],
      "regions": ["all"]
    }
  ]
}
```

How to view the version catalog in the Health Dashboard

The following shows the version catalog feature in the Health Dashboard.


AWS Health Dashboard
Version Catalog
You can use the Version catalog to search and view all version updates for AWS core services.
Version catalog (20)
View and manage version catalog for AWS core services.
Saved filter sets
Choose a filter set
Search and add filter
Service = LAMBDA
Clear filters
1 ...
Service
Resource
Version
Lambda Python 3.10 Lambda Runtime python3.10
Lambda .NET 9 Lambda Runtime (container only) dotnet9
Lambda .NET 8 Lambda Runtime dotnet8
Lambda Python 3.12 Lambda Runtime python3.12
Lambda .NET 10 Lambda Runtime dotnet10
Lambda Java 11 Lambda Runtime java11
Lambda Node.js 22 Lambda Runtime nodejs22.x
Lambda Java 25 Lambda Runtime java25
Lambda Java 17 Lambda Runtime java17
Lambda Ruby 3.4 Lambda Runtime ruby3.4
Lambda Java 17 Lambda Runtime (AL2023) java17.al2023
Lambda Ruby 3.3 Lambda Runtime ruby3.3
Lambda Python 3.11 Lambda Runtime python3.11
Lambda Python 3.14 Lambda Runtime python3.14
LAMBDA java11
Service
LAMBDA
Version
java11
Title
LAMBDA java11
Last updated
-
Timeline
End of Support · June 29, 2027 End of support
Runtime no longer receives security patches or updates
Block Resource Create · July 30, 2027 Availability
Cannot create new Lambda functions using this runtime
Block Resource Update · August 30, 2027 Availability
Cannot update existing Lambda functions using this runtime

Integrating AWS Health with other systems using the AWS Health API

AWS Health is a RESTful web service that uses HTTPS as a transport and JSON as a message serialization format. Your application code can make requests directly to the AWS Health API. When you use the REST API directly, you must write the necessary code to sign and authenticate your requests. For more information about the AWS Health operations and parameters, see the AWS Health API Reference.

> **Note**
>
> You must have an AWS Business Support+, AWS Enterprise Support, or AWS Unified Operations plan from AWS Support to use the AWS Health API. If you're in an AWS Region that doesn't offer one of these AWS Support plans, or if you haven't transitioned to one of these plans, you can use the AWS Health API with a Business, Enterprise On-Ramp, or Enterprise Support plan. If you call the AWS Health API from an AWS account that isn't enrolled in one of these plans, then you receive a `SubscriptionRequiredException` error.

You can use the AWS SDKs to wrap the AWS Health REST API calls, which can simplify your application development. You specify your AWS credentials, and these libraries take care of authentication and request signing for you.

AWS Health also provides a AWS Health Dashboard in the AWS Management Console that you can use to view and search for events and affected entities. See Getting started with your AWS Health Dashboard.

Topics

- Signing AWS Health API requests
- Choosing endpoints for AWS Health API requests
- Demos: Retrieving the last seven days of AWS Health event data programmatically
- Tutorial: Using the AWS Health API with Java examples

Signing AWS Health API requests

When you use the AWS SDKs or the AWS Command Line Interface (AWS CLI) to make requests to AWS, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. For example, if you use the AWS SDK for Java for the previous high availability endpoint demo, you don't need to sign requests yourself.

Java code examples

For more examples on how to use the AWS Health API with the AWS SDK for Java, see this example code.

When you make requests, we strongly recommend that you don't use your AWS root account credentials for regular access to AWS Health. You can use the credentials for an IAM user. For more information, see Lock Away Your AWS Account Root User Access Keys in the *IAM User Guide*.

If you don’t use the AWS SDKs or the AWS CLI, then you must sign your requests yourself. We recommend that you use AWS Signature Version 4. For more information, see Signing AWS API Requests in the *AWS General Reference*.

Choosing endpoints for AWS Health API requests

The AWS Health API follows a multi-Region application architecture and has two regional endpoints in an active-passive configuration. To support active-passive DNS failover, AWS Health provides a single, global endpoint. You can perform a DNS lookup on the global endpoint to determine the active endpoint and corresponding signing AWS Region. This helps you know which endpoint to use in your code, so that you can get the latest information from AWS Health.

When you make a request to the global endpoint, you must specify your AWS access credentials to the regional endpoint that you target and configure the signing for your Region. Otherwise, your authentication might fail. For more information, see Signing AWS Health API requests.

For IPv6-only requests, we recommend performing a DNS lookup on the global endpoint to determine the active AWS Region and then calling the IPv6 supported dual-stack endpoint for that Region.

The following table represents the default configuration.

Description	Signing Region	Endpoint	Protocol
Active	us-east-1	health.us-east-1.amazonaws.com (IPv4-only) health.us-east-1.api.aws (IPv4 and IPv6 supported)	HTTPS
Passive	us-east-2	health.us-east-2.amazonaws.com (IPv4-only) health.us-east-2.api.aws (IPv4 and IPv6 supported)	HTTPS
Global	us-east-1 **Note** This is the signing Region of the current active endpoint.	global.health.amazonaws.com	HTTPS

To determine if an endpoint is the *active endpoint*, do a DNS lookup on the *global endpoint* CNAME, and then extract the AWS Region from the resolved name.

Example: DNS lookup on global endpoint

The following command completes a DNS lookup on the global.health.amazonaws.com endpoint. The command then returns the us-east-1 Region endpoint. This output tells you which endpoint you should use for AWS Health.

```
dig global.health.amazonaws.com | grep CNAME
```

```
global.health.amazonaws.com. 10 IN CNAME health.us-east-1.amazonaws.com
```

Tip

Both the active and passive endpoints return AWS Health data. However, the latest AWS Health data is only available from the active endpoint. Data from the passive endpoint will be eventually consistent with the active endpoint. We recommend that you restart any workflows when the active endpoint changes.

Demos: Retrieving the last seven days of AWS Health event data programmatically

In the following code examples, AWS Health uses a DNS lookup against the global endpoint to determine the active regional endpoint and signing Region. AWS Health uses this information to retrieve a report of the last seven days of event data. The code restarts the workflow if the active endpoint changes.

Topics

- Demo: Retrieving the last seven days of AWS Health event data using Java
- Demo: Retrieving the last seven days of AWS Health event data using Python

Demo: Retrieving the last seven days of AWS Health event data using Java

Prerequisite

You must install Gradle.

To use the Java example

1. Download the AWS Health high availability endpoint demo from GitHub.
2. Navigate to the demo project `high-availability-endpoint/java` directory.
3. In a command line window, enter the following command.

```
gradle build
```

4. Enter the following commands to specify your AWS credentials.

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
export AWS_SESSION_TOKEN="your-aws-token"
```

5. Enter the following command to run the demo.

```
gradle run
```

Example: AWS Health event output

The code example returns the recent AWS Health event in the last seven days in your AWS account. In the following example, the output includes an AWS Health event for the AWS Config service.

```
> Task :run
[main] INFO aws.health.high.availability.endpoint.demo.HighAvailabilityV2Workflow
 - EventDetails(Event=Event(Arn=arn:aws:health:global::event/CONFIG/
AWS_CONFIG_OPERATIONAL_NOTIFICATION/AWS_CONFIG_OPERATIONAL_NOTIFICATION_88a43e8a-
e419-4ca7-9baa-56bcde4dba3,
Service=CONFIG, EventTypeCode=AWS_CONFIG_OPERATIONAL_NOTIFICATION,
 EventTypeCategory=accountNotification, Region=global,
 StartTime=2020-09-11T02:55:49.899Z, LastUpdatedTime=2020-09-11T03:46:31.764Z,
StatusCode=open, EventScopeCode=ACCOUNT_SPECIFIC),
 EventDescription=EventDescription(LatestDescription=As part of our ongoing efforts
 to optimize costs associated with recording changes related to certain ephemeral
 workloads,
AWS Config is scheduled to release an update to relationships modeled within
 ConfigurationItems (CI) for 7 EC2 resource types on August 1, 2021.
Examples of ephemeral workloads include changes to Amazon Elastic Compute Cloud
 (Amazon EC2) Spot Instances, Amazon Elastic MapReduce jobs, and Amazon EC2
 Autoscaling.
This update will optimize CI models for EC2 Instance, SecurityGroup, Network
 Interface, Subnet, VPC, VPN Gateway, and Customer Gateway resource types to record
 direct relationships and deprecate indirect relationships.

A direct relationship is defined as a one-way relationship (A->B) between a
 resource (A) and another resource (B), and is typically derived from the Describe
 API response of resource (A).
An indirect relationship, on the other hand, is a relationship that AWS Config
 infers (B->A), in order to create a bidirectional relationship.
```

```
For example, EC2 instance -> Security Group is a direct relationship, since
 security groups are returned as part of the describe API response for an EC2
 instance.
But Security Group -> EC2 instance is an indirect relationship, since EC2 instances
 are not returned when describing an EC2 Security group.

Until now, AWS Config has recorded both direct and indirect relationships. With
 the launch of Advanced queries in March 2019, indirect relationships can easily be
 answered by running Structured Query Language (SQL) queries such as:

SELECT
 resourceId,
 resourceType
WHERE
 resourceType ='AWS::EC2::Instance'
AND
 relationships.resourceId = 'sg-234213'

By deprecating indirect relationships, we can optimize the information contained
 within a
Configuration Item while reducing AWS Config costs related to relationship
 changes.
This is especially useful in case of ephemeral workloads where there is a high
 volume of configuration changes for EC2 resource types.

Which resource relationships are being removed?

Resource Type: Related Resource Type
1 AWS::EC2::CustomerGateway: AWS::VPN::Connection
2 AWS::EC2::Instance: AWS::EC2::EIP, AWS::EC2::RouteTable
3 AWS::EC2::NetworkInterface: AWS::EC2::EIP, AWS::EC2::RouteTable
4 AWS::EC2::SecurityGroup: AWS::EC2::Instance, AWS::EC2::NetworkInterface
5 AWS::EC2::Subnet: AWS::EC2::Instance, AWS::EC2::NetworkACL,
 AWS::EC2::NetworkInterface, AWS::EC2::RouteTable
6 AWS::EC2::VPC: AWS::EC2::Instance, AWS::EC2::InternetGateway,
 AWS::EC2::NetworkACL, AWS::EC2::NetworkInterface, AWS::EC2::RouteTable,
 AWS::EC2::Subnet, AWS::EC2::VPNGateway, AWS::EC2::SecurityGroup
7 AWS::EC2::VPNGateway: AWS::EC2::RouteTable, AWS::EC2::VPNConnection

Alternate mechanism to retrieve this relationship information:
The SelectResourceConfig API accepts a SQL SELECT command, performs the
 corresponding search, and returns resource configurations matching the properties.
 You can use this API to retrieve the same relationship information.
```

```
For example, to retrieve the list of all EC2 Instances related to a particular VPC
 vpc-1234abc, you can use the following query:

SELECT
 resourceId,
 resourceType
WHERE
 resourceType ='AWS::EC2::Instance'
AND
 relationships.resourceId = 'vpc-1234abc'

If you have any questions regarding this deprecation plan, please contact AWS
 Support [1]. Additional sample queries to retrieve the relationship information
 for the resources listed above is provided in [2].

[1] https://aws.amazon.com/support
[2] https://docs.aws.amazon.com/config/latest/developerguide/
examplerelationshipqueries.html),
EventMetadata={})
```

Java resources

- For more information, see the Interface HealthClient in the *AWS SDK for Java API Reference* and the source code.
- For more information about the library used in this demo for DNS lookups, see the dnsjava in GitHub.

Demo: Retrieving the last seven days of AWS Health event data using Python

Prerequisite

You must install Python 3.

To use the Python example

1. Download the AWS Health high availability endpoint demo from GitHub.
2. Navigate to the demo project `high-availability-endpoint/python` directory.
3. In a command line window, enter the following commands.

```
pip3 install virtualenv
virtualenv -p python3 v-aws-health-env
```

> **Note**
>
> For Python 3.3 and later, you can use the built-in `venv` module to create the virtual environment, instead of installing `virtualenv`. For more information, see venv - Creation of virtual environments on the Python website.
>
> ```
> python3 -m venv v-aws-health-env
> ```

4. Enter the following command to activate the virtual environment.

```
source v-aws-health-env/bin/activate
```

5. Enter the following command to install the dependencies.

```
pip install -r requirements.txt
```

6. Enter the following commands to specify your AWS credentials.

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
export AWS_SESSION_TOKEN="your-aws-token"
```

7. Enter the following command to run the demo.

```
python3 main.py
```

Example: AWS Health event output

The code example returns the recent AWS Health event in the last seven days in your AWS account. The following output returns an AWS Health event for an AWS security notification.

```
INFO:botocore.credentials:Found credentials in environment variables.
INFO:root:Details: {'arn': 'arn:aws:health:global::event/SECURITY/
AWS_SECURITY_NOTIFICATION/AWS_SECURITY_NOTIFICATION_0e35e47e-2247-47c4-
a9a5-876544042721',
```

```
'service': 'SECURITY', 'eventTypeCode': 'AWS_SECURITY_NOTIFICATION',
 'eventTypeCategory': 'accountNotification', 'region': 'global', 'startTime':
 datetime.datetime(2020, 8, 19, 23, 30, 42, 476000,
tzinfo=tzlocal()), 'lastUpdatedTime': datetime.datetime(2020, 8, 20, 20, 44, 9,
 547000, tzinfo=tzlocal()), 'statusCode': 'open', 'eventScopeCode': 'PUBLIC'},
 description:
{'latestDescription': 'This is the second notice regarding TLS requirements on FIPS
 endpoints.\n\nWe
are in the process of updating all AWS Federal Information Processing Standard
 (FIPS) endpoints across all AWS regions
to Transport Layer Security (TLS) version 1.2 by March 31, 2021 . In order to avoid
 an interruption in service, we encourage you to act now, by ensuring that you
 connect to AWS FIPS endpoints at a TLS version of 1.2.
If your client applications fail to support TLS 1.2 it will result in connection
 failures when TLS versions below 1.2 are no longer supported.\n\nBetween now and
 March 31, 2021 AWS will remove TLS 1.0 and TLS 1.1 support from each FIPS endpoint
 where no connections below TLS 1.2 are detected over a 30-day period.
After March 31, 2021 we may deploy this change to all AWS FIPS endpoints, even if
 there continue
to be customer connections detected at TLS versions below 1.2. \n\nWe will provide
 additional updates and reminders on the AWS Security Blog, with a 'TLS' tag [1].
 If you need further guidance or assistance, please contact AWS Support [2] or your
 Technical Account Manager (TAM).
Additional information is below.\n\nHow can I identify clients that are connecting
 with TLS
1.0/1.1?\nFor customers using S3 [3], Cloudfront [4] or Application Load Balancer
 [5] you can use
your access logs to view the TLS connection information for these services, and
 identify client
connections that are not at TLS 1.2. If you are using the AWS Developer Tools on
 your clients,
you can find information on how to properly configure your client's TLS versions
 by visiting Tools to Build on AWS [7] or our associated AWS Security Blog has a
 link for each unique code language [7].\n\nWhat is Transport Layer Security (TLS)?
\nTransport Layer Security (TLS Protocols) are cryptographic protocols designed to
 provide secure communication across a computer network
[6].\n\nWhat are AWS FIPS endpoints? \nAll AWS services offer Transport Layer
 Security (TLS) 1.2 encrypted endpoints that can be used for all API calls. Some
 AWS services also offer FIPS 140-2 endpoints [9] for customers that require use
 of FIPS validated cryptographic libraries. \n\n[1] https://aws.amazon.com/blogs/
security/tag/tls/\n[2] https://aws.amazon.com/support\n[3]
https://docs.aws.amazon.com/AmazonS3/latest/dev/LogFormat.html\n[4] https://
docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/AccessLogs.html\n[5]
 https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-
```

```
access-logs.html\n[6] https://aws.amazon.com/tools\n[7] https://aws.amazon.com/
blogs/security/tls-1-2-to-become-the-minimum-for-all-aws-fips-endpoints\n[8]
https://en.wikipedia.org/wiki/Transport_Layer_Security\n[9] https://aws.amazon.com/
compliance/fips'}
```

8. When you're finished, enter the following command to deactivate the virtual machine.

```
deactivate
```

Python resources

- For more information about the `Health. Client`, see the AWS SDK for Python (Boto3) API Reference.
- For more information about the library used in this demo for DNS lookups, see the dnspython toolkit and the source code on GitHub.

Tutorial: Using the AWS Health API with Java examples

The following Java code examples demonstrate how to initialize an AWS Health client and retrieve information about events and entities.

Step 1: Initialize credentials

Valid credentials are required to communicate with the AWS Health API. You can use the key pair of any IAM user associated with the AWS account.

Create and initialize an AWSCredentials instance:

```
AWSCredentials credentials = null;
try {
       credentials = new ProfileCredentialsProvider("default").getCredentials();
} catch (Exception e) {
throw new AmazonClientException(
   "Cannot load the credentials from the credential profiles file. "
   + "Please make sure that your credentials file is at the correct "
   + "location (/home/username/.aws/credentials), and is in valid format.", e);
}
```

Step 2: Initialize an AWS Health API client

Use the initialized credentials object from the previous step to create an AWS Health client:

```
import com.amazonaws.services.health.AWSHealthClient;

AWSHealth awsHealthClient = new AWSHealthClient(credentials);
```

Step 3: Use AWS Health API operations to get event information

DescribeEvents

```
import com.amazonaws.services.health.model.DescribeEventsRequest;
import com.amazonaws.services.health.model.DescribeEventsResult;
import com.amazonaws.services.health.model.Event;
import com.amazonaws.services.health.model.EventFilter;

DescribeEventsRequest request = new DescribeEventsRequest();

EventFilter filter = new EventFilter();
// Filter on any field from the supported AWS Health EventFilter model.
// Here is an example for Region us-east-1 events from the EC2 service.
filter.setServices(singletonList("EC2"));
filter.setRegions(singletonList("us-east-1"));
request.setFilter(filter);

DescribeEventsResult response = awsHealthClient.describeEvents(request);
List<Event> resultEvents = response.getEvents();

Event currentEvent = null;
for (Event event : resultEvents) {
    // Display result event data; here is a subset.
    System.out.println(event.getArn());
    System.out.println(event.getService());
    System.out.println(event.getRegion());
    System.out.println(event.getAvailabilityZone());
    System.out.println(event.getStartTime());
    System.out.println(event.getEndTime());
 }
```

DescribeEventAggregates

```
import com.amazonaws.services.health.model.DescribeEventAggregatesRequest;
import com.amazonaws.services.health.model.DescribeEventAggregatesResult;
import com.amazonaws.services.health.model.EventAggregate;
import com.amazonaws.services.health.model.EventFilter;

DescribeEventAggregatesRequest request = new DescribeEventAggregatesRequest();
// set the aggregation field
request.setAggregateField("eventTypeCategory");

// filter more on result if needed
EventFilter filter = new EventFilter();
filter.setRegions(singleton("us-east-1"));
request.setFilter(filter);

DescribeEventAggregatesResult response =
 awsHealthClient.describeEventAggregates(request);

// print event count for each eventTypeCategory
for (EventAggregate aggregate: response.getEventAggregates()) {
    System.out.println("Event Category:" + aggregate.getAggregateValue());
    System.out.println("Event Count:" + aggregate.getCount());
 }
```

DescribeEventDetails

```
import com.amazonaws.services.health.model.DescribeEventDetailsRequest;
import com.amazonaws.services.health.model.DescribeEventDetailsResult;
import com.amazonaws.services.health.model.Event;
import com.amazonaws.services.health.model.EventDetails;


DescribeEventDetailsRequest describeEventDetailsRequest = new
 DescribeEventDetailsRequest();
// set event ARN and local value

describeEventDetailsRequest.setEventArns(singletonList("arn:aws:health:us-
east-1::event/service/eventTypeCode/eventId"));
describeEventDetailsRequest.setLocale("en-US");
filter.setEventArns
DescribeEventDetailsResult describeEventDetailsResult =
 awsHealthClient.describeEventDetails(request);
EventDetails eventDetail = describeEventDetailsResult.getSuccessfulSet().get(0);
```

```
// check event-related fields
Event event = eventDetail.getEvent();
System.out.println(event.getService());
System.out.println(event.getRegion());
System.out.println(event.getAvailabilityZone());
System.out.println(event.getStartTime());
System.out.println(event.getEndTime());

// print out event description
System.out.println(eventDetail.getEventDescription().getLatestDescription());
```

DescribeAffectedEntities

```
import com.amazonaws.services.health.model.AffectedEntity;
import com.amazonaws.services.health.model.DateTimeRange;
import com.amazonaws.services.health.model.DescribeAffectedEntitiesRequest;
import
 com.amdescribeEventDetailsRequestazonaws.services.health.model.DescribeAffectedEntitiesResult;

DescribeAffectedEntitiesRequest request = new DescribeAffectedEntitiesRequest();
EntityFilter filter = new EntityFilter();

filter.setEventArns(singletonList("arn:aws:health:us-
east-1::event/service/eventTypeCode/eventId"));

DescribeAffectedEntitiesResult response =
 awsHealthClient.describeAffectedEntities(request);

for (AffectedEntity affectedEntity: response.getEntities()) {
    System.out.println(affectedEntity.getEntityValue());
    System.out.println(affectedEntity.getAwsAccountId());
    System.out.println(affectedEntity.getEntityArn());
 }
```

DescribeEntityAggregates

```
import com.amazonaws.services.health.model.DescribeEntityAggregatesRequest;
import com.amazonaws.services.health.model.DescribeEntityAggregatesResult;
import com.amazonaws.services.health.model.EntityAggregate;

DescribeEntityAggregatesRequest request = new DescribeEntityAggregatesRequest();
```

```
request.setEventArns(singletonList("arn:aws:health:us-
east-1::event/service/eventTypeCode/eventId"));

DescribeEntityAggregatesResult response =
 awsHealthClient.describeEntityAggregates(request);

for (EntityAggregate entityAggregate : response.getEntityAggregates()) {
    System.out.println(entityAggregate.getEventArn());
    System.out.println(entityAggregate.getCount());
 }
```

Security in AWS Health

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to AWS Health, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You're also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS Health. The following topics show you how to configure AWS Health to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS Health resources.

Topics

- Data protection in AWS Health
- Identity and access management for AWS Health
- Logging and monitoring in AWS Health
- Compliance validation for AWS Health
- Resilience in AWS Health
- Infrastructure security in AWS Health
- Configuration and vulnerability analysis in AWS Health
- Security best practices for AWS Health

Data protection in AWS Health

The AWS shared responsibility model applies to data protection in . As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see Data Privacy FAQ. For information about data protection in Europe, see the General Data Protection Regulation (GDPR) Center.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Data encryption

See the following information about how AWS Health encrypts data.

Data encryption refers to protecting data while in-transit (as it travels from the service to your AWS account), and at rest (while it is stored in AWS services). You can protect data in transit using Transport Layer Security (TLS) or at rest using client-side encryption.

AWS Health doesn't record personal identifying information (PII) such as email addresses or customer names in events.

Encryption at rest

All data stored by AWS Health is encrypted at rest.

Encryption in transit

All data sent to and from AWS Health is encrypted in transit.

Key management

AWS Health doesn't support customer-managed encryption keys for data encrypted in the AWS Cloud.

Identity and access management for AWS Health

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use AWS Health resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Audience
- Authenticating with identities
- Managing access using policies
- How AWS Health works with IAM
- AWS Health identity-based policy examples
- Troubleshooting AWS Health identity and access
- Using service-linked roles for AWS Health
- AWS managed policies for AWS Health

Audience

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user** - request permissions from your administrator if you cannot access features (see Troubleshooting AWS Health identity and access)
- **Service administrator** - determine user access and submit permission requests (see How AWS Health works with IAM)
- **IAM administrator** - write policies to manage access (see AWS Health identity-based policy examples)

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see Tasks that require root user credentials in the *IAM User Guide*.

IAM users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by switching from a user to an IAM role (console) or by calling an AWS CLI or AWS API operation. For more information, see Methods to assume a role in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. You must specify a principal in a resource-based policy.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

AWS Health supports resource-based conditions. You can specify which AWS Health events that users can view. For example, you might create a policy that only allows an IAM user access to specific Amazon EC2 events in the AWS Health Dashboard.

For more information, see Resources.

Access control lists

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

AWS Health doesn't support ACLs.

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see Service control policies in the *AWS Organizations User Guide*.

- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How AWS Health works with IAM

Before you use IAM to manage access to AWS Health, you should understand what IAM features are available to use with AWS Health. To get a high-level view of how AWS Health and other AWS services work with IAM, see AWS Services That Work with IAM in the *IAM User Guide*.

Topics

- AWS Health identity-based policies
- AWS Health resource-based policies
- Authorization based on AWS Health tags
- AWS Health IAM roles

AWS Health identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. AWS Health supports specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see IAM JSON Policy Elements Reference in the *IAM User Guide*.

Actions

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Include actions in a policy to grant permissions to perform the associated operation.

Policy actions in AWS Health use the following prefix before the action: `health:`. For example, to grant someone permission to view detailed information about specified events with the DescribeEventDetails API operation, you include the `heath:DescribeEventDetails` action in the policy.

Policy statements must include an `Action` or `NotAction` element. AWS Health defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows.

```
"Action": [
      "health:action1",
      "health:action2"
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Describe`, include the following action.

```
"Action": "health:Describe*"
```

To see a list of AWS Health actions, see Actions Defined by AWS Health in the *IAM User Guide*.

Resources

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

An AWS Health event has the following Amazon Resource Name (ARN) format.

```
arn:${Partition}:health:*::event/service/event-type-code/event-ID
```

For example, to specify the EC2_INSTANCE_RETIREMENT_SCHEDULED_ABC123-DEF456 event in your statement, use the following ARN.

```
"Resource": "arn:aws:health:*::event/EC2/EC2_INSTANCE_RETIREMENT_SCHEDULED/
EC2_INSTANCE_RETIREMENT_SCHEDULED_ABC123-DEF456"
```

To specify all AWS Health events for Amazon EC2 that belong to a specific account, use the wildcard (*).

```
"Resource": "arn:aws:health:*::event/EC2/*/*"
```

For more information about the format of ARNs, see Amazon Resource Names (ARNs) and AWS Service Namespaces.

Some AWS Health actions can't be performed on a specific resource. In those cases, you must use the wildcard (*).

```
"Resource": "*"
```

AWS Health API operations can involve multiple resources. For example, the DescribeEvents operation returns information about events that meet a specified filter criteria. This means that an IAM user must have permissions to view this event.

To specify multiple resources in a single statement, separate the ARNs with commas.

```
"Resource": [
      "resource1",
      "resource2"
```

AWS Health supports only resource-level permissions for health events and only for the DescribeAffectedEntities and DescribeEventDetails API operations. For more information, see Resource- and action-based conditions.

To see a list of AWS Health resource types and their ARNs, see Resources Defined by AWS Health in the *IAM User Guide*. To learn with which actions you can specify the ARN of each resource, see Actions Defined by AWS Health.

Condition keys

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element specifies when statements execute based on defined criteria. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

AWS Health defines its own set of condition keys and also supports using some global condition keys. To see all AWS global condition keys, see AWS Global Condition Context Keys in the *IAM User Guide*.

The DescribeAffectedEntities and DescribeEventDetails API operations support the `health:eventTypeCode` and `health:service` condition keys.

To see a list of AWS Health condition keys, see Condition Keys for AWS Health in the *IAM User Guide*. To learn with which actions and resources you can use a condition key, see Actions Defined by AWS Health.

Examples

To view examples of AWS Health identity-based policies, see AWS Health identity-based policy examples.

AWS Health resource-based policies

Resource-based policies are JSON policy documents that specify what actions a specified principal can perform on the AWS Health resource and under what conditions. AWS Health supports resource-based permissions policies for health events. Resource-based policies let you grant usage permission to other accounts on a per-resource basis. You can also use a resource-based policy to allow an AWS service to access your AWS Health events.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, you must also grant the principal entity permission to access the resource. Grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based

policy is required. For more information, see How IAM Roles Differ from Resource-based Policies in the *IAM User Guide*.

AWS Health supports only resource-based policies for the DescribeAffectedEntities and DescribeEventDetails API operations. You can specify these actions in a policy to define which principal entities (accounts, users, roles, and federated users) can perform actions on the AWS Health event.

Examples

To view examples of AWS Health resource-based policies, see Resource- and action-based conditions.

Authorization based on AWS Health tags

AWS Health doesn't support tagging resources or controlling access based on tags.

AWS Health IAM roles

An IAM role is an entity within your AWS account that has specific permissions.

Using temporary credentials with AWS Health

You can use temporary credentials to sign in with federation, assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as AssumeRole or GetFederationToken.

AWS Health supports using temporary credentials.

Service-linked roles

Service-linked roles allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

AWS Health supports service-linked roles to integrate with AWS Organizations. The service-linked role is named `AWSServiceRoleForHealth_Organizations`. Attached to the role is the Health_OrganizationsServiceRolePolicy AWS managed policy. The AWS managed policy allows AWS Health to access health events from other AWS accounts in the organization.

You can use the EnableHealthServiceAccessForOrganization operation to create the service-linked role in the account. However, if you want to disable this feature, you must first call the

DisableHealthServiceAccessForOrganization operation. You can then delete the role through the IAM console, IAM API, or AWS Command Line Interface (AWS CLI). For more information, see Using service-linked roles in the *IAM User Guide*.

For more information, see Aggregating AWS Health events across accounts.

Service roles

This feature allows a service to assume a service role on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service roles appear in your IAM account and are owned by the account. This means that an IAM administrator can change the permissions for this role. However, doing so might break the functionality of the service.

AWS Health doesn't support service roles.

AWS Health identity-based policy examples

By default, IAM users and roles don't have permission to create or modify AWS Health resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see Creating Policies on the JSON Tab in the *IAM User Guide*.

Topics

- Policy best practices
- Using the AWS Health console
- Allow users to view their own permissions
- Accessing the AWS Health Dashboard and the AWS Health API
- Resource- and action-based conditions

Policy best practices

Identity-based policies determine whether someone can create, access, or delete AWS Health resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the AWS Health console

To access the AWS Health console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the AWS Health resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required

permissions, the console won't function as intended for entities (IAM users or roles) with that policy.

To ensure that those entities can still use the AWS Health console, you can attach the following AWS managed policy, AWSHealthFullAccess.

The `AWSHealthFullAccess` policy grants an entity full access to the following:

- Enable or disable the AWS Health organizational view feature for all accounts in an AWS organization
- The AWS Health Dashboard in the AWS Health console
- AWS Health API operations and notifications
- View information about accounts that are part of your AWS organization
- View the organizational units (OU) of the management account

Example: AWSHealthFullAccess

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "organizations:EnableAWSServiceAccess",
                "organizations:DisableAWSServiceAccess"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "organizations:ServicePrincipal": "health.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "health:*",
```

```
                "organizations:DescribeAccount",
                "organizations:ListAccounts",
                "organizations:ListDelegatedAdministrators",
                "organizations:ListParents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "health.amazonaws.com"
                }
            }
        }
    ]
}
```

Note

You can also use the `Health_OrganizationsServiceRolePolicy` AWS managed policy, so that AWS Health can view events for other accounts in your organization. For more information, see Using service-linked roles for AWS Health.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that you're trying to perform.

For more information, see Adding Permissions to a User in the *IAM User Guide*.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Accessing the AWS Health Dashboard and the AWS Health API

The AWS Health Dashboard is available for all AWS accounts. The AWS Health API is available only to accounts with a AWS Business Support+, AWS Enterprise Support, or AWS Unified Operations plan. For more information, see Support.

You can use IAM to create entities (users, groups, or roles), and then give those entities permissions to access the AWS Health Dashboard and the AWS Health API.

By default, IAM users don't have access to the AWS Health Dashboard or the AWS Health API. You give users access to your account's AWS Health information by attaching IAM policies to a single

user, a group of users, or a role. For more information, see Identities (Users, Groups, and Roles) and Overview of IAM Policies.

After you create IAM users, you can give those users individual passwords. Then, they can sign in to your account and view AWS Health information by using an account-specific sign-in page. For more information, see How Users Sign In to Your Account.

> **Note**
>
> An IAM user with permissions to view AWS Health Dashboard has read-only access to health information across all AWS services on the account, which can include, but is not limited to, AWS resource IDs such as Amazon EC2 instance IDs, EC2 instance IP addresses, and general security notifications.
> For example, if an IAM policy grants access only to AWS Health Dashboard and the AWS Health API, then the user or role that the policy applies to can access all information posted about AWS services and related resources, even if other IAM policies don't allow that access.

You can use two groups of APIs for AWS Health.

- Individual accounts – You can use the operations such as DescribeEvents and DescribeEventDetails to get information about AWS Health events for your account.
- Organizational account – You can use operations such as DescribeEventsForOrganization and DescribeEventDetailsForOrganization to get information about AWS Health events for accounts that are part of your organization.

For more information about the available API operations, see the AWS Health API Reference.

Individual actions

You can set the `Action` element of an IAM policy to `health:Describe*`. This allows access to the AWS Health Dashboard and AWS Health. AWS Health supports access control to events based on the `eventTypeCode` and service.

Describe access

This policy statement grants access to AWS Health Dashboard and any of the `Describe*` AWS Health API operations. For example, an IAM user with this policy can access the AWS Health

Dashboard in the AWS Management Console and call the AWS Health `DescribeEvents` API operation.

Example: Describe access

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "health:Describe*"
    ],
    "Resource": "*"
  }]
}
```

Deny access

This policy statement denies access to AWS Health Dashboard and the AWS Health API. An IAM user with this policy can't view the AWS Health Dashboard in the AWS Management Console and can't call any of the AWS Health API operations.

Example: Deny access

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
  {
    "Effect": "Deny",
    "Action": [
      "health:*"
    ],
    "Resource": "*"
  }]
}
```

Organizational view

If you want to enable organizational view for AWS Health, you must allow access to the AWS Health and AWS Organizations actions.

The `Action` element of an IAM policy must include the following permissions:

- `iam:CreateServiceLinkedRole`
- `organizations:EnableAWSServiceAccess`
- `organizations:DescribeAccount`
- `organizations:DisableAWSServiceAccess`
- `organizations:ListAccounts`
- `organizations:ListDelegatedAdministrators`
- `organizations:ListParents`

To understand the exact permissions needed for each APIs, see Actions Defined by AWS Health APIs and Notifications in the *IAM User Guide*.

> **Note**
>
> You must use credentials from the management account for an organization to access the AWS Health APIs for AWS Organizations. For more information, see Aggregating AWS Health events across accounts.

Allow access to AWS Health organizational view

This policy statement grants access to all AWS Health and AWS Organizations actions that you need for the organizational view feature.

Example: Allow AWS Health organizational view access

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "organizations:EnableAWSServiceAccess",
                "organizations:DisableAWSServiceAccess"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "organizations:ServicePrincipal": "health.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "health:*",
                "organizations:DescribeAccount",
                "organizations:ListAccounts",
                "organizations:ListDelegatedAdministrators",
                "organizations:ListParents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
health.amazonaws.com/AWSServiceRoleForHealth*"
        }
    ]
}
```

Deny access to AWS Health organizational view

This policy statement denies access to the AWS Organizations actions but allows access to the AWS Health actions for an individual account.

Example: Deny AWS Health organizational view access

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "health:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "organizations:EnableAWSServiceAccess",
                "organizations:DisableAWSServiceAccess"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "organizations:ServicePrincipal": "health.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "organizations:DescribeAccount",
                "organizations:ListAccounts",
                "organizations:ListDelegatedAdministrators",
                "organizations:ListParents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
health.amazonaws.com/AWSServiceRoleForHealth*"
        }
    ]
}
```

Note

If the user or group that you want to give permissions to already has an IAM policy, you can add the AWS Health-specific policy statement to that policy.

Resource- and action-based conditions

AWS Health supports IAM conditions for the DescribeAffectedEntities and DescribeEventDetails API operations. You can use resource- and action-based conditions to restrict events that the AWS Health API sends to a user, group, or role.

To do so, update the `Condition` block of the IAM policy or set the `Resource` element. You can use String Conditions to restrict access based on certain AWS Health event fields.

You can use the following fields when you specify an AWS Health event in your policy:

- `eventTypeCode`
- `service`

Notes

- The DescribeAffectedEntities and DescribeEventDetails API operations support resource-level permissions. For example, you can create a policy to allow or deny specific AWS Health events.
- The DescribeAffectedEntitiesForOrganization and DescribeEventDetailsForOrganization API operations don't support resource-level permissions.
- For more information, see Actions, resources, and condition keys for AWS Health APIs and Notifications in the *Service Authorization Reference*.

Example: Action-based condition

This policy statement grants access to AWS Health Dashboard and the AWS Health `Describe*` API operations, but denies access to any AWS Health events that relate to Amazon EC2.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "health:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "health:DescribeAffectedEntities",
                "health:DescribeEventDetails"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "health:service": "EC2"
                }
            }
        }
    ]
}
```

Example: Resource-based condition

The following policy has the same effect, but uses the `Resource` element instead.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "health:Describe*"
    ],
    "Resource": "*"
```

```
  },
  {
    "Effect": "Deny",
    "Action": [
      "health:DescribeEventDetails",
      "health:DescribeAffectedEntities"
    ],
    "Resource": "arn:aws:health:*::event/EC2/*/*"
  }]
}
```

Example: eventTypeCode condition

This policy statement grants access to AWS Health Dashboard and the AWS Health `Describe*` API operations, but denies access to any AWS Health events with the `eventTypeCode` that matches `AWS_EC2_*`.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "health:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "health:DescribeAffectedEntities",
                "health:DescribeEventDetails"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "health:eventTypeCode": "AWS_EC2_*"
                }
            }
        }
    ]
```

```
}
```

Important

If you call the DescribeAffectedEntities and DescribeEventDetails operations and don't have permission to access the AWS Health event, the `AccessDeniedException` error appears. For more information, see Troubleshooting AWS Health identity and access.

Troubleshooting AWS Health identity and access

Use the following information to diagnose and fix common issues that you might encounter when working with AWS Health and IAM.

Topics

- I'm not authorized to perform an action in AWS Health
- I'm not authorized to perform iam:PassRole
- I want to view my access keys
- I'm an administrator and want to allow others to access AWS Health
- I want to allow people outside of my AWS account to access my AWS Health resources

I'm not authorized to perform an action in AWS Health

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password.

The `AccessDeniedException` error appears when a user doesn't have permission to use AWS Health Dashboard or the AWS Health API operations.

In this case, the user's administrator must update the policy to allow the user access.

The AWS Health API requires a AWS Business Support+, AWS Enterprise Support, or AWS Unified Operations plan from AWS Support. If you call the AWS Health API from an account that doesn't have a AWS Business Support+, AWS Enterprise Support, or AWS Unified Operations plan, the following error code is returned: `SubscriptionRequiredException`.

I'm not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to AWS Health.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in AWS Health. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to view my access keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (for example, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

> **Important**
>
> Do not provide your access keys to a third party, even to help find your canonical user ID. By doing this, you might give someone permanent access to your AWS account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see Managing access keys in the *IAM User Guide*.

I'm an administrator and want to allow others to access AWS Health

To allow others to access AWS Health, you must grant permission to the people or applications that need access. If you are using AWS IAM Identity Center to manage people and applications, you assign permission sets to users or groups to define their level of access. Permission sets automatically create and assign IAM policies to IAM roles that are associated with the person or application. For more information, see Permission sets in the *AWS IAM Identity Center User Guide*.

If you are not using IAM Identity Center, you must create IAM entities (users or roles) for the people or applications that need access. You must then attach a policy to the entity that grants them the correct permissions in AWS Health. After the permissions are granted, provide the credentials to the user or application developer. They will use those credentials to access AWS. To learn more about creating IAM users, groups, policies, and permissions, see IAM Identities and Policies and permissions in IAM in the *IAM User Guide*.

I want to allow people outside of my AWS account to access my AWS Health resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether AWS Health supports these features, see How AWS Health works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.

- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Using service-linked roles for AWS Health

AWS Health uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to AWS Health. Service-linked roles are predefined by AWS Health and include all the permissions that the service requires to call other AWS services for you.

You can use a service-linked role to set up AWS Health to avoid manually adding the necessary permissions. AWS Health defines the permissions of its service-linked roles, and unless defined otherwise, only AWS Health can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy can't be attached to any other IAM entity.

Service-linked role permissions for AWS Health

AWS Health has three service-linked roles:

- AWSServiceRoleForHealth_Organizations – This role trusts the AWS Health (`health.amazonaws.com`) to assume the role to access AWS services for you. Attached to this role is the `Health_OrganizationsServiceRolePolicy` AWS managed policy.
- AWSServiceRoleForHealth_EventProcessor – This role trusts the AWS Health service principal (`event-processor.health.amazonaws.com`) to assume the role for you. Attached to this role is the `AWSHealth_EventProcessorServiceRolePolicy` AWS managed policy. The service principal uses the role to create an Amazon EventBridge managed rule for AWS Incident Detection and Response. This rule is the infrastructure required in your AWS account to deliver alarm state change information from your account to AWS Health.
- AWSServiceRoleForTagsHealth – This role trusts the AWS Health service principal (`tags.health.amazonaws.com`) to assume the role for you. Attached to this role is the TagsHealthServiceRolePolicy AWS managed policy. The service principal uses the role to read the tags on the resources in your account through the AWS Resource Groups Tagging API, so that AWS Health can show your AWS Health events by resource tag.

For more information about the AWS managed policies, see AWS managed policies for AWS Health.

Creating a service-linked role for AWS Health

You don't need to create the AWSServiceRoleForHealth_Organizations service-linked role. When you call the EnableHealthServiceAccessForOrganization operation, AWS Health creates this service-linked role in the account for you.

You must manually create the AWSServiceRoleForHealth_EventProcessor service-linked role in your account. For more information, see Creating a service-linked role in the *IAM User Guide*.

You don't need to create the AWSServiceRoleForTagsHealth service-linked role. When you turn on the tags feature for your account, AWS Health creates the role in your account for you. When you turn on the tags feature for your organization from your organization's management account, AWS Health creates the role in the management account and in every member account of your organization. AWS Health also creates the role in any account that joins your organization after you turn on the feature.

Editing a service-linked role for AWS Health

AWS Health doesn't allow you to edit the service-linked role. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a service-linked role in the *IAM User Guide*.

Deleting a service-linked role for AWS Health

To delete the AWSServiceRoleForHealth_Organizations role, you must first call the DisableHealthServiceAccessForOrganization operation. You can then delete the role through the IAM console, IAM API, or AWS Command Line Interface (AWS CLI).

To delete the AWSServiceRoleForHealth_EventProcessor role, contact AWS Support and ask that they offboard your workloads from AWS Incident Detection and Response. After this process completes, you can then delete either role through the IAM console, IAM API, or AWS CLI.

To delete the AWSServiceRoleForTagsHealth role, you must first turn off the tags feature for the account. If the account belongs to an organization that has the tags feature turned on, the organization's management account must also turn off the tags feature for the organization. If you try to delete the role while the tags feature is still enabled for the account or its organization, the deletion fails and IAM reports that the role is in use by AWS Health. After the feature is disabled, you can delete the role through the IAM console, IAM API, or AWS CLI.

Related information

For more information, see Using service-linked roles in the *IAM User Guide*.

AWS managed policies for AWS Health

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AWS Health has the following managed policies.

Contents

- AWS managed policy: AWSHealth_EventProcessorServiceRolePolicy
- AWS managed policy: Health_OrganizationsServiceRolePolicy
- AWS managed policy: TagsHealthServiceRolePolicy
- AWS managed policy: AWSHealthFullAccess
- AWS Health updates to AWS managed policies

AWS managed policy: AWSHealth_EventProcessorServiceRolePolicy

AWS Health uses the AWSHealth_EventProcessorServiceRolePolicy AWS managed policy. This managed policy is attached to the `AWSServiceRoleForHealth_EventProcessor` service-linked

role. The policy allows the service-linked role to complete actions for you. You can't attach this policy to your IAM entities. For more information, see Using service-linked roles for AWS Health.

The managed policy has the following permissions to allow AWS Health to access the Amazon EventBridge rule for AWS Incident Detection and Response.

Permissions details

This policy includes the following permissions.

- `events` – Describes and deletes EventBridge rules, and describes and updates the targets for those rules.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Condition": {
                "StringEquals": {"events:ManagedBy": "event-
processor.health.amazonaws.com"}
            },
            "Action": [
                "events:DeleteRule",
                "events:RemoveTargets",
                "events:PutTargets",
                "events:PutRule"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "events:ListTargetsByRule",
                "events:DescribeRule"
            ],
            "Resource": "*",
            "Effect": "Allow"
```

```
        }
    ]
}
```

For a list of changes to the policy, see AWS Health updates to AWS managed policies.

AWS managed policy: Health_OrganizationsServiceRolePolicy

AWS Health uses the Health_OrganizationsServiceRolePolicy AWS managed policy. This managed policy is attached to the `AWSServiceRoleForHealth_Organizations` service-linked role. The policy allows the service-linked role to complete actions for you. You can't attach this policy to your IAM entities. For more information, see Using service-linked roles for AWS Health.

This policy grants permissions that allow AWS Health to access required AWS Organizations details for the Health Organizational view.

Permissions details

This policy includes the following permissions.

- `organizations` – Describes the accounts in AWS Organizations and the AWS services that can be used with Organizations.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "organizations:ListAccounts",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListDelegatedAdministrators",
                "organizations:DescribeOrganization",
                "organizations:DescribeAccount"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

For a list of changes to the policy, see AWS Health updates to AWS managed policies.

AWS managed policy: TagsHealthServiceRolePolicy

AWS Health uses the TagsHealthServiceRolePolicy AWS managed policy. This managed policy is attached to the `AWSServiceRoleForTagsHealth` service-linked role. The policy allows the service-linked role to complete actions for you. You can't attach this policy to your IAM entities. For more information, see Using service-linked roles for AWS Health.

This policy grants read-only permissions that allow AWS Health to read the tags on the resources in your account. AWS Health uses these tags so that you can view and filter your AWS Health events by resource tag. With this policy, AWS Health can't create, change, or remove tags, and it can't access the resources themselves.

Permissions details

This policy includes the following permissions.

- `tag` – Retrieves the resources in your account and the tags attached to them through the AWS Resource Groups Tagging API.

To view the permissions for this policy, see TagsHealthServiceRolePolicy in the *AWS Managed Policy Reference*.

For a list of changes to the policy, see AWS Health updates to AWS managed policies.

AWS managed policy: AWSHealthFullAccess

AWS Health uses the AWSHealthFullAccess AWS managed policy. The policy grants entities (IAM users or roles) access to the AWS Health console. For more information, see Using the AWS Health console.

Permissions details

This policy includes the following permissions.

- `organizations` – Enable or disable the AWS Health organizational view feature for all accounts in an AWS organization, and view the organizational units (OU) of the management account
- `health` – Access to the AWS Health API operations and notifications
- `iam` – Creates an IAM role that is linked the AWS Health service

JSON

```
{
    "Version":"2012-10-17",
        "Statement": [
            {
                "Sid": "OrganizationWriteAccess",
                "Effect": "Allow",
                "Action": [
                    "organizations:EnableAWSServiceAccess",
                    "organizations:DisableAWSServiceAccess"
                ],
                "Resource": "*",
                "Condition": {
                    "StringEquals": {
                        "organizations:ServicePrincipal": "health.amazonaws.com"
                    }
                }
            },
            {
                "Sid": "HealthFullAccess",
                "Effect": "Allow",
                "Action": [
                    "health:*",
                    "organizations:DescribeAccount",
                    "organizations:ListAccounts",
                    "organizations:ListDelegatedAdministrators",
                    "organizations:ListParents"
                ],
                "Resource": "*"
            },
            {
                "Sid": "ServiceLinkAccess",
                "Effect": "Allow",
                "Action": "iam:CreateServiceLinkedRole",
                "Resource": "*",
```

```
                "Condition": {
                    "StringEquals": {
                        "iam:AWSServiceName": "health.amazonaws.com"
                    }
                }
            }
        ]
}
```

For a list of changes to the policy, see AWS Health updates to AWS managed policies.

AWS Health updates to AWS managed policies

View details about updates to AWS managed policies for AWS Health since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Document history for AWS Health page.

The following table describes important updates to the AWS Health managed policies since January 13, 2022.

AWS Health

Change	Description	Date
AWS managed policy: TagsHealthServiceRolePolicy - New policy	AWS Health added a new managed policy that is attached to the `AWSServiceRoleForTagsHealth` service-linked role. The policy grants read-only access to the AWS Resource Groups Tagging API so that AWS Health can read the tags on the resources in your account. For more information, see	October 30, 2026

Change	Description	Date
	Using service-linked roles for AWS Health.	
AWS managed policy: AWSHealthFullAccess - Update to an existing policy	AWS Health has expanded the AWSHealthFullAccess policy to AWS GovCloud (US) Regions and China Regions.	October 16, 2023
AWS managed policy: Health_OrganizationsServiceRolePolicy - Update to an existing policy	AWS Health added new AWS Organizations actions to allow service-linked role to describe the accounts and AWS services that can be used with AWS Organizations.	July 19, 2023
Change log published	Change log for the AWS Health managed policies.	January 13, 2023

Logging and monitoring in AWS Health

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Health and your other AWS solutions. AWS provides the following monitoring tools to watch AWS Health, report when something is wrong, and take actions when appropriate:

- *Amazon CloudWatch* monitors your AWS resources and the applications that you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track CPU usage or other metrics of your Amazon Elastic Compute Cloud (Amazon EC2) instances and automatically launch new instances when needed. For more information, see the Amazon CloudWatch User Guide.
- *Amazon EventBridge* delivers a near-real-time stream of system events that describe changes in AWS resources. EventBridge enables automated event-driven computing. You can write rules that watch for certain events and trigger automated actions in other AWS services when these events happen. For more information, see Monitoring events in AWS Health with Amazon EventBridge.

- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon Simple Storage Service (Amazon S3) bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

For more information, see Monitoring AWS Health.

Compliance validation for AWS Health

Our new AWS sign-up experience is not designed for regulated workloads. If you're using our new AWS sign-up experience, but you want to use AWS for regulated workloads, you can sign up for AWS (advanced) or activate advanced features for your AWS environment.

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more information about your compliance responsibility when using AWS services, see AWS Security Documentation.

Resilience in AWS Health

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

AWS Health events are stored and replicated across multiple Availability Zones. This approach ensures that you can access them from the Health Dashboard or the AWS Health API operations. You can view AWS Health events up to 90 days from when they occur.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in AWS Health

As a managed service, AWS Health is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access AWS Health through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Configuration and vulnerability analysis in AWS Health

Configuration and IT controls are a shared responsibility between AWS and you, our customer. For more information, see the AWS shared responsibility model.

Security best practices for AWS Health

See the following best practices for working with AWS Health.

Grant AWS Health users minimum possible permissions

Follow the principle of least privilege by using the minimum set of access policy permissions for your users and groups. For example, you might allow an AWS Identity and Access Management (IAM) user access to the Health Dashboard. However, you might not allow that same user to enable or disable access to AWS Organizations.

For more information, see AWS Health identity-based policy examples.

View the Health Dashboard

Check your Health Dashboard often to identify events that might affect your account or applications. For example, you might receive an event notification about your resources, such as an Amazon Elastic Compute Cloud (Amazon EC2) instance that needs to be updated.

For more information, see Getting started with your AWS Health Dashboard.

Integrate AWS Health with Amazon Chime or Slack

You can integrate AWS Health with your chat tools. This integration lets you and your team get notified about AWS Health events in real time. For more information, see the AWS Health Tools in GitHub.

Monitor for AWS Health events

You can integrate AWS Health with Amazon CloudWatch Events, so that you can create rules for specific events. When CloudWatch Events detects an event that matches your rule, you are notified and can then take action. CloudWatch Events events are Region-specific, so you must configure this service in the Region in which your application or infrastructure resides.

In some cases, the Region for the AWS Health event can't be determined. If that situation occurs, the event appears in the US East (N. Virginia) Region by default. You can set up CloudWatch Events in this Region to ensure that you monitor these events.

For more information, see Monitoring events in AWS Health with Amazon EventBridge.

Aggregating AWS Health events across accounts

By default, you can use AWS Health to view the AWS Health events of a single AWS account. If you use AWS Organizations, you can also view AWS Health events centrally across your organization. This feature provides access to the same information as single account operations. You can use filters to view events in specific AWS Regions, accounts, and services.

You can aggregate events to identify accounts in your organization that are affected by an operational event or get notified for security vulnerabilities. You can then use this information to proactively manage and automate resource maintenance events across your organization. Use this feature to stay informed of upcoming changes to AWS services that might require updates or code changes.

It's a best practice to use the Delegated Administrator feature to delegate access to the AWS Health Organizational view to a member account. This makes it easier for operational teams to access the AWS Health events in your organization. The Delegated Administrator feature allows you to keep your management account restricted, while providing teams with the visibility that they need to act on AWS Health events.

Important

- AWS Health events that were sent for accounts in your organization will appear in organizational view as long as the event is available, up to 90 days, even if one or more of those accounts leave your organization.
- Organizational events are available for 90 days before they're deleted. This quota can't be increased.

Prerequisites

Before you use organizational view, you must:

- Be part of an organization with all features enabled.
- Sign in to the management account as an AWS Identity and Access Management (IAM) user or assume an IAM role.

You can also sign in as the root user (not recommended) in your organization's management account. For more information, see Lock away your AWS account root user access keys in the *IAM User Guide*.

- If you sign in as an IAM user, use an IAM policy that grants access to the AWS Health and Organizations actions, such as the AWSHealthFullAccess policy. For more information, see AWS Health identity-based policy examples.

Topics

- Enabling organizational view
- Viewing organizational view
- Disabling organizational view
- Managing delegated administrator views for an organization

Enabling organizational view

You can use the AWS Health console to get a centralized view for health events in your AWS organization.

Organizational view is available in the AWS Health console for all AWS Support plans at no additional cost.

> **Note**
>
> If you want to allow users access to this feature in the management account, they must have permissions such as the AWSHealthFullAccess policy. For more information, see AWS Health identity-based policy examples.

Enabling organizational view (Console)

You can enable organizational view from the AWS Health console. You must sign in to the management account of your AWS organization.

To view the AWS Health Dashboard for your organization

1. Open your AWS Health Dashboard at https://health.aws.amazon.com/health/home.

2. In the navigation pane, under **Your organization health**, choose **Configurations**.
3. On the **Enable organizational view** page, choose **Enable organizational view**.
4. (Optional) If you want to make changes to your AWS organizations, such as creating organizational units (OUs), choose **Manage AWS Organizations**.

   For more information, see Getting started with AWS Organizations in the *AWS Organizations User Guide*.

> **Notes**
>
> - When you enable AWS Health organizational view, the initial account loading process runs in the background and might take several minutes to complete. You can close the AWS Health console and return later, as you don't need to wait for the process to finish. Historical health events (those created before you enabled the feature) might take up to 24 hours to appear in your organizational view.
> - If you have a AWS Business Support+, AWS Enterprise Support, or AWS Unified Operations plan, you can call the DescribeHealthServiceStatusForOrganization API operation to check the status of the process.
> - When you enable this feature, the `AWSServiceRoleForHealth_Organizations` service-linked role with the `Health_OrganizationsServiceRolePolicy` AWS managed policy is applied to the management account in the organization. For more information, see Using service-linked roles for AWS Health.

Enabling organizational view (CLI)

You can enable organizational view by using the EnableHealthServiceAccessForOrganization API operation.

You can use the AWS Command Line Interface (AWS CLI) or your own code to call this operation.

> **Note**
>
> - You must have a Business, Enterprise On-Ramp, or Enterprise Support plan to call the AWS Health API.

- You must use the US East (N. Virginia) Region endpoint.

Example

The following AWS CLI command enables this feature from your AWS account. You can use this command from the management account or from an account that can assume the role with the required permissions.

```
aws health enable-health-service-access-for-organization --region us-east-1
```

The following code examples call the EnableHealthServiceAccessForOrganization API operation.

Python

```
import boto3

client = boto3.client('health', region_name='us-east-1')

response = client.enable_health_service_access_for_organization()

print(response)
```

Java

You can use the AWS SDK for version Java 2.0 for the following example.

```
import software.amazon.awssdk.services.health.HealthClient;
import software.amazon.awssdk.services.health.HealthClientBuilder;

import software.amazon.awssdk.services.health.model.ConcurrentModificationException;
import
 software.amazon.awssdk.services.health.model.EnableHealthServiceAccessForOrganizationReques
import
 software.amazon.awssdk.services.health.model.EnableHealthServiceAccessForOrganizationRespon
import
 software.amazon.awssdk.services.health.model.DescribeHealthServiceStatusForOrganizationRequ
import
 software.amazon.awssdk.services.health.model.DescribeHealthServiceStatusForOrganizationResp

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
```

```
import software.amazon.awssdk.regions.Region;

public class EnableHealthServiceAccessDemo {
    public static void main(String[] args) {
        HealthClient client = HealthClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(
                DefaultCredentialsProvider.builder().build()
            )
            .build();

        try {
            DescribeHealthServiceStatusForOrganizationResponse statusResponse =
 client.describeHealthServiceStatusForOrganization(
                DescribeHealthServiceStatusForOrganizationRequest.builder().build()
            );

            String status =
 statusResponse.healthServiceAccessStatusForOrganization();
            if ("ENABLED".equals(status)) {
                System.out.println("EnableHealthServiceAccessForOrganization already
 enabled!");
                return;
            }

            client.enableHealthServiceAccessForOrganization(
                EnableHealthServiceAccessForOrganizationRequest.builder().build()
            );

            System.out.println("EnableHealthServiceAccessForOrganization is in
 progress");
        } catch (ConcurrentModificationException cme) {
            System.out.println("EnableHealthServiceAccessForOrganization is already
 in progress. Wait for the action to complete before trying again.");
        } catch (Exception e) {
            System.out.println("EnableHealthServiceAccessForOrganization FAILED: " +
 e);
        }
    }
}
```

For more information, see the AWS SDK for Java 2.0 Developer Guide.

When you enable this feature, the `AWSServiceRoleForHealth_Organizations` service-linked role with the `Health_OrganizationsServiceRolePolicy` AWS managed policy is applied to the management account in the organization.

> **Note**
>
> Enabling this feature is an asynchronous process and takes time to complete. You can call the DescribeHealthServiceStatusForOrganization operation to check the status of the process.

Viewing organizational view

You can use the AWS Health console to get a centralized view for health events in your AWS organization.

Organizational view is available in the AWS Health console for all AWS Support plans at no additional cost.

> **Note**
>
> If you want to allow users access to this feature in the management account, they must have permissions such as the AWSHealthFullAccess policy. For more information, see AWS Health identity-based policy examples.

Viewing organizational view events (Console)

After you enable organizational view, AWS Health displays health events for all accounts in your organization.

When an account joins your organization, AWS Health automatically adds the account to organizational view. When an account leaves your organization, new events from that account are no longer logged to organizational view. However, existing events remain and you can still query them up to the 90-day limit.

AWS retains the policy data for the account for 90 days from the effective date of the administrator account closure. At the end of the 90 day period, AWS permanently deletes all policy data for the account.

- To retain findings for more than 90 days, you can archive the policies. You can also use a custom action with an EventBridge rule to store the findings in an S3 bucket.
- As long as AWS retains the policy data, when you reopen the closed account, AWS reassigns the account as the service administrator and recovers the service policy data for the account.
- For more information, see Closing an account.

Important

For customers in the AWS GovCloud (US) Regions:

- Before closing your account, back up and then delete account resources. You will no longer have access to them after you close the account.

Note

When you enable this feature, the AWS Health console can display public events from the AWS Health Dashboard – Service health for the last 7 days. These public events aren't specific to accounts in your organization. Events from the AWS Health Dashboard – Service health provide public information about the regional availability of AWS services.

You can view organizational view events in the following pages:.

Open and Recent Issues

You can use the **Open and recent issues** tab to view events that might affect your AWS infrastructure, such as changes to AWS services and resources that affect your organization.

To view organizational view events

1. Open your AWS Health Dashboard at https://health.aws.amazon.com/health/home.
2. In the navigation pane, under **Your organization health**, choose **Open and recent issues** to view recently reported events.
3. Choose an event. On the **Details** tab, you can review the following information about the event:

- Event name
- Status
- Region / Availability Zone
- Affected accounts
- Start time
- End time
- Category
- Description

Scheduled Changes

Use the **Scheduled changes** tab to view upcoming events that might affect your organization. These events can include scheduled maintenance activities for services.

Other Notifications

Use the **Notifications** tab to view all other notifications and ongoing events from the past seven days that might affect your organization. This can include events, such as certificate rotations, billing notifications, and security vulnerabilities.

Event Log

You can also use the **Event log** tab to view AWS Health events for organizational view. The column layout and behavior are similar to the **Open and recent issues** tab, except that the **Event log** tab includes additional columns and filter options, such as the **Event category**, **Status**, and **Start time**.

To view organizational view events in the Event log tab

1. Open your AWS Health Dashboard at https://health.aws.amazon.com/health/home.
2. In the navigation pane, under **Your organization health**, choose **Event log**.
3. Under **Event log**, choose the event name. You can review the following information about the event:

   - Event name
   - Status

- Region / Availability Zone
- Affected accounts
- Start time
- End time
- Category
- Description

Viewing affected accounts and resources (Console)

Under **Your organization health**, you can view the accounts in your organization that are affected by the event and any related resources. For example, if there's an upcoming event for Amazon Elastic Compute Cloud (Amazon EC2) instance maintenance, accounts in your organization that have Amazon EC2 instances can appear in the **Details** tab. You can identify the specific resources and then contact the account owner.

To view affected accounts and resources

1. Open your AWS Health Dashboard at https://health.aws.amazon.com/health/home.
2. In the navigation pane, under **Your organization health**, choose one of the tabs.
3. Choose an event that has a value for **Affected accounts**.
4. Choose the **Affected accounts** tab.
5. Choose **Show account details** to view the following information for the accounts:
   - Account ID
   - Account name
   - Primary email
   - Organizational unit (OU)
6. Expand the account to view the affected resources.
7. If there are more than 10 resources, choose **View all resources** to view them.
8. To filter by account ID for this specific event, do the following:
   a. On the **Affected accounts** tab, choose **Add filter**, choose **Account ID**, and then enter the account ID. You can only enter one account ID at a time.
   b. Choose **Apply**. The account that you entered appears in the list.

Viewing organizational view events (CLI)

After you enable this feature, AWS Health starts to record events that affect accounts in the organization. When an account joins your organization, AWS Health automatically adds the account to organizational view.

> **Note**
>
> AWS Health doesn't record events that occurred in your organization before you enabled organizational view.

When an account leaves your organization, new events from that account are no longer logged to organizational view. However, existing events remain and you can still query them up to the 90-day limit.

AWS retains the policy data for the account for 90 days from the effective date of the administrator account closure. At the end of the 90 day period, AWS permanently deletes all policy data for the account.

- To retain findings for more than 90 days, you can archive the policies. You can also use a custom action with an EventBridge rule to store the findings in an S3 bucket.
- As long as AWS retains the policy data, when you reopen the closed account, AWS reassigns the account as the service administrator and recovers the service policy data for the account.
- For more information, see Closing an account.

> **Important**
>
> For customers in the AWS GovCloud (US) Regions:
>
> - Before closing your account, back up and then delete account resources. You will no longer have access to them after you close the account.

You can use the AWS Health API operations to return events from organizational view.

Example: Describe organizational view events

The following AWS CLI command returns health events for AWS accounts in your organization.

```
aws health describe-events-for-organization --region us-east-1
```

Disabling organizational view

If you don't want to aggregate events for your organization, you can turn off this feature from the management account or you can disable organizational view by using the DisableHealthServiceAccessForOrganization API operation.

Disabling organizational view events (Console)

AWS Health stops aggregating events for all other accounts in your organization. You can continue to view previous events from your organization until they're deleted.

To disable organizational view

1. Open your AWS Health Dashboard at https://health.aws.amazon.com/health/home.
2. In the navigation pane, under **Your organization health**, choose **Configurations**.
3. On the **Enable organizational view** page, choose **Disable organizational view**.

After you turn off this feature, AWS Health no longer aggregates events from your organization. However, the service-linked role remains in the management account until you delete it through the AWS Identity and Access Management (IAM) console, IAM API, or AWS Command Line Interface (AWS CLI). For more information, see Deleting a service-linked role in the *IAM User Guide*.

Disabling organizational view events (CLI)

Example

The following AWS CLI command disables this feature from your account.

```
aws health disable-health-service-access-for-organization --region us-east-1
```

> **Note**
>
> You can also disable the organizational feature by using the Organizations DisableAWSServiceAccess API operation. After you call this operation, AWS Health stops aggregating events for all other accounts in your organization. If you call the AWS

Health API operations for organizational view, AWS Health returns an error. AWS Health continues to aggregate health events for your AWS account.

After you disable this feature, AWS Health no longer aggregates events from your organization. However, the service-linked role remains in the management account until you delete it through the AWS Identity and Access Management (IAM) console, IAM API, or AWS CLI. For more information, see Deleting a service-linked Role in the *IAM User Guide*.

Managing delegated administrator views for an organization

With AWS Health, you can leverage the delegated administrator feature from AWS Organizations that allows an account other than the management account to view aggregated AWS Health events on the AWS Health Dashboard or programmatically through the AWS Health API. The delegated administrator feature provides the flexibility for different teams to view and manage health events across your organization. It's an AWS security best practice to delegate responsibilities outside of the management account where possible.

Contents

- Registering a delegated administrator for your organizational view
- Removing a delegated administrator from your organizational view

Registering a delegated administrator for your organizational view

After you enable organizational view for your organization, you can register up to five member accounts in your organization as a delegated administrator. To do this, call the RegisterDelegatedAdministrator API operation. After you register the member accounts, they are a delegated administrator account and can access the AWS Health organizational view from the AWS Health Dashboard. If the account has a Business, Enterprise On-Ramp, or Enterprise Support plan, then the delegated administrators can use the AWS Health API to access the AWS Health organizational view.

To establish a delegated administrator, from the management account in your organization, call the following AWS Command Line Interface (AWS CLI) command. You can use this command from the management account or from an account that can assume the role with the required AWS Identity and Access Management permissions. In the following example command, replace

ACCOUNT_ID with the member account ID that you want to register along with the AWS Health service principal "health.amazonaws.com".

```
aws organizations register-delegated-administrator --account-id ACCOUNT_ID --service-
principal  health.amazonaws.com
```

After a delegated administrator is registered, you have visibility into all AWS Health events affecting accounts across your organization. You can view historical events over the past 90 days or since the organizational view feature was first enabled, whichever is more recent. Note that enabling the delegated administrator feature is an asynchronous process and takes up to a minute to complete.

Removing a delegated administrator from your organizational view

To remove access for a delegated administrator, call the DeregisterDelegatedAdministrator API operation.

From your organization's management account, call the following AWS CLI command to remove a member account as delegated administrator. In the following example command, replace **ACCOUNT_ID** with the member account ID that you want to remove.

```
aws organizations deregister-delegated-administrator --account-id ACCOUNT_ID --service-
principal  health.amazonaws.com
```

Monitoring events in AWS Health with Amazon EventBridge

You can use Amazon EventBridge to detect and react to AWS Health events. Then, based on rules that you create, EventBridge invokes one or more target actions when an event matches the values that you specify in a rule. Depending on the type of event, you can capture event information, initiate additional events, send notifications, take corrective action, or perform other actions. For example, you can use AWS Health to receive email notifications if you have AWS resources in your AWS account that are scheduled for updates, such as Amazon Elastic Compute Cloud (Amazon EC2) instances.

Notes

- AWS Health delivers events on a *durable* basis and attempts to successfully deliver events to EventBridge at least once.
- Any EventBridge rules that you create can only receive notifications for your AWS account. To receive organizational events for other accounts within your AWS Organizations, see Aggregating AWS Health events using organizational view and delegated administrator access.
- Public health events might take up to one hour to start sending after you create an EventBridge rule.

You can choose between multiple target types for EventBridge as part of your AWS Health workflow, including:

- AWS Lambda functions
- Amazon Kinesis Data Streams
- Amazon Simple Queue Service (Amazon SQS) queues
- Built-in targets (such as CloudWatch alarm actions)
- Amazon Simple Notification Service (Amazon SNS) topics

For example, you can use a Lambda function to pass a notification to a Slack channel when an AWS Health event occurs. Or, you can use Lambda and EventBridge to send custom text or SMS notifications with Amazon SNS when an AWS Health event occurs.

For samples of automation and customized alerts that you can create in response to AWS Health events, see the AWS Health Tools in GitHub.

Topics

- Creating EventBridge rules for AWS Region coverage
- Monitoring account-specific and public events for AWS Health
- Viewing paginated lists of AWS Health events on EventBridge
- Aggregating AWS Health events using organizational view and delegated administrator access
- Integrating AWS Health event monitoring and notifications with JIRA and ServiceNow
- Configuring an EventBridge rule to send notifications about events in AWS Health
- Configuring Amazon Q Developer in chat applications to send notifications about events in AWS Health
- Running operations on EC2 instances automatically in response to events in AWS Health
- Reference: AWS Health events Amazon EventBridge schema

Creating EventBridge rules for AWS Region coverage

You can create an EventBridge rule for each Region that you want to receive AWS Health events for. For example, to receive events from the Europe (Frankfurt) Region, you can create a rule for this Region.

To enhance the reliability of AWS Health notifications, you can set up rules in the dedicated backup Regions. In the standard AWS partition, the US West (Oregon) Region acts as the backup region for all other Regions, while US East (N. Virginia) Region serves as the backup for the US West (Oregon) Region. When health events occur, they are automatically sent to both the primary Region and its designated backup Region. For example, if you're monitoring events in the Europe (Frankfurt) Region, then any health events are delivered to both the Europe (Frankfurt) Region and the US West (Oregon) Region. This system makes sure you continue to receive health notifications even if your primary Region experiences issues. To create a backup rule, follow the procedure for Configuring an EventBridge rule to send notifications about events in AWS Health.

If you prefer not to use backup functionality, then you must add a filter to your backup region rule. For example, implement a filter for `detail.backupEvent = False`. This prevents you from receiving backup events from other regions.

High availability setup (optional)

If you want to create an EventBridge integration with high availability, make sure you have implemented rules in both the relevant and backup Regions, and then implement de-duplication using `detail.communicationId`. This makes sure you receive all events while avoiding duplicates. For more information, see Reference: AWS Health events Amazon EventBridge schema.

Simplified integration

If you want to capture account-specific events from multiple AWS Regions, but prefer to configure only a single rule, then simplified integration is the appropriate option. To receive account-specific AWS Health events from all Regions in the standard AWS partition, you can set up a central rule in the US West (Oregon) Region. This single rule automatically aggregates account-specific events from all standard partition Regions. However, you won't have high availability configuration.

> **Note**
>
> Simplified integration doesn't allow you to receive public events that are shown in the AWS Health Dashboard. Simplified integration is only recommended for ingesting action-required AWS Health events. For more information about public and account-specific events, see Monitoring account-specific and public events for AWS Health.

Global events

Some AWS Health events are not Region-specific. Events that aren't specific to a Region are called global events. These include events sent for AWS Identity and Access Management (IAM). To receive global events, you must create a rule for the US East (N. Virginia) Region.

Monitoring account-specific and public events for AWS Health

When you create an EventBridge rule to monitor events from AWS Health, the rule delivers both account-specific events and public events:

- *Account-specific* events affect your account and resources, such as an event that tells you about a required update to an Amazon EC2 instance or other scheduled change events.
- *Public* events appear on the AWS Health Dashboard – Service health. Public events aren't specific to AWS accounts and provide public information about the Regional availability of a service.

> **Important**
>
> To receive both event types, your rule must use the `"source": [ "aws.health"]` value. Wildcards, such as `"source": [ "aws.health*"]` won't match the pattern to monitor for any events.

You can identify if an event is public or account-specific in EventBridge, by using the eventScopeCode parameter. Events can have the `PUBLIC` or `ACCOUNT_SPECIFIC`. You can also filter your rule on this parameter.

To view an example public event for Amazon Elastic Compute Cloud, see Public Health Event - Amazon EC2 operational issue.

Backup rules for AWS Health events

If you're monitoring public events from an AWS Region, we recommend that you create a back up rule. Public events for AWS Health are sent simultaneously to both the impacted Region and to the backup Region when a valid rule is set in the impacted Region.

> **Important**
>
> If you don't have a rule in the impacted Region, the backup Region will not automatically send out the public event as a backup. You must have a valid rule in the impacted Region for public events to be delivered to the backup Region.

AWS Health sends account-specific events to both the impacted Region and to the backup Region, regardless of any rules configured in the impacted Region.

We recommend that you deduplicate AWS Health events using `eventARN` and `communicationId` because these values remain consistent for AWS Health messages that are sent to the backup Region.

Viewing paginated lists of AWS Health events on EventBridge

AWS Health supports pagination of AWS Health events when the list of `resources` or `affectedEntities` causes the size of the message to exceed EventBridge's 256KB message size limit.

AWS Health includes all `resources` and `detail.affectedEntities` fields in the message. If this list of `resources` and `detail.affectedEntities` values exceeds 256KB, then AWS Health splits the health event into multiple pages and publish these pages as individual messages in EventBridge. Each page retains the same `eventARN` and `communicationId` values to help recombine the list of `resources` or `detail.affectedEntities` after all the pages are received.

These additional messages might cause unecessary messages, for example when the EventBridge rule is directed to a human readable interface such as email or chat. Customers with human readable notifications can add a filter for the `detail.page` field to process only the first page, which eliminates the unnecessary messages created from subsequent pages.

In the schema, each communicationId includes the hyphenated page number after the communicationId, even when there is only 1 page. The fields `detail.page` and `detail.totalPages` describe the current page number and the total number of pages for the AWS Health event. The information contained in each paginated message is the same except for the list of `detail.affectedEntities` or `resources`. These lists can be reconstructed after all the pages are received. The pages of affected resources and entities are order-agnostic.

Aggregating AWS Health events using organizational view and delegated administrator access

AWS Health supports organizational view and delegated administrator access for AWS Health events published on Amazon EventBridge. When organizational view is turned on in AWS Health, then the management account or a delegated administrator account receives a single feed of AWS Health events from all accounts within your organization in AWS Organizations.

This feature is designed to provide a centralized view to help manage AWS Health events across your organization. Setting up organizational view and an EventBridge rule in the management account doesn't deactivate EventBridge rules for other accounts in your organization.

For more information on enabling organizational view and delegated administrator access on AWS Health, see Aggregating AWS Health Events.

Integrating AWS Health event monitoring and notifications with JIRA and ServiceNow

End of support notice

On March 31, 2027, AWS will end support for AWS Service Management Connector. After March 31, 2027, you will no longer be able to access the Service Management Connector console or Service Management Connector resources. For more information, see AWS Service Management Connector end of support.

You can integrate AWS Health events with JIRA and ServiceNow to receive operational and account information, prepare for scheduled changes, and manage Health events using the Service Management Connector (SMC). The SMC Integration with AWS Health can use Health events sent through EventBridge to automatically create, map, and update JIRA tickets and ServiceNow incidents.

You can use organizational view and delegated administrator access to easily manage Health events across the organization within JIRA and ServiceNow, and incorporate AWS Health information directly into your team's workflow.

For more information on ServiceNow integration using the SMC, see Integrating AWS Health in ServiceNow.

For more information on JIRA Management Cloud integration using the SMC, see AWS Health in JIRA.

Configuring an EventBridge rule to send notifications about events in AWS Health

You can create an Amazon EventBridge rule to programmatically integrate AWS Health events with other services, applications, and workloads. EventBridge provides a drag and drop console interface and an API to set up rules that trigger when a matching AWS Health event is created for your account or organization. To learn how to set up a rule in EventBridge to capture AWS Health events, see Creating rules using the Enhanced Builder and Creating rules using the Advanced Builder in the *Amazon EventBridge User Guide*.

Depending on your integration, EventBridge allows you to add parameters to the EventBridge rule to filter only the AWS Health events that you want to integrate with your use case. For incident response use cases, you might want to focus on the `issue` event category and certain critical services. For change management use cases such as planned lifecycle events, you might want to focus on AWS Health events with `ACTION_REQUIRED` in the **Actionability** field. For integrating with security use cases, you might want to focus on all AWS Health Abuse events and AWS Health events with the `SECURITY` persona field.

You can use sample use cases to verify that your rule captures the events you need. Sample use cases are available in Reference: AWS Health events Amazon EventBridge schema. You can also find them in the EventBridge console under the **Use Sample events provided** option in the **Test event pattern - optional** panel

Using the API or AWS Command Line Interface

For a new or existing rule, use the PutRule API operation or the `aws events put-rule` command to update the event pattern. To view an example AWS CLI command, see put-rule in the *AWS CLI Command Reference*.

Example: Setting up rules for issues for only the Amazon EC2 service

The following event pattern creates a rule to monitor issue events for the Amazon EC2 service.

```
{
  "detail": {
    "eventTypeCategory": [
      "issue"
    ],
    "service": [
      "EC2"
    ]
  },
  "detail-type": [
    "AWS Health Event"
  ],
  "source": [
    "aws.health"
  ]
}
```

Example: Setting up rules for all action required AWS Health events, including planned lifecycle events

The following event pattern creates a rule to monitor all AWS Health events that require action, including planned lifecycle events.

```
{
  "detail": {
    "eventTypeCategory": [
      "accountNotification",
      "scheduledChange"
    ],
    "actionability": [
      "ACTION_REQUIRED"
    ]
  },
  "detail-type": [
    "AWS Health Event"
  ],
  "source": [
    "aws.health"
  ]
}
```

Example: Setting up rules for all AWS Health events for multiple services and event type categories

The following event pattern creates a rule to monitor events for the `issue`, `accountNotification`, and `scheduledChange` event type categories for three AWS services: Amazon EC2 Auto Scaling, Amazon VPC, and Amazon EC2.

```
{
  "detail": {
    "eventTypeCategory": [
      "issue",
      "accountNotification",
      "scheduledChange"
    ],
    "service": [
      "AUTOSCALING",
      "VPC",
      "EC2"
```

```
    ]
  },
  "detail-type": [
    "AWS Health Event"
  ],
  "source": [
    "aws.health"
  ]
}
```

Configuring Amazon Q Developer in chat applications to send notifications about events in AWS Health

You can receive AWS Health events directly in your chat clients, such as Slack and Amazon Chime. You can use this event to identify recent AWS service issues that might affect your AWS applications and infrastructure. Then, you can sign in to your AWS Health Dashboard to learn more about the update. For example, if you're monitoring for the `AWS_EC2_INSTANCE_STOP_SCHEDULED` event type in your AWS account, the AWS Health event can appear directly to your Slack channel.

Prerequisites

Before you get started, you must have the following:

- A chat client configured with Amazon Q Developer in chat applications. You can configure Amazon Chime and Slack. For more information, see Getting started with Amazon Q Developer in chat applications in the *Amazon Q Developer in chat applications Administrator Guide*.
- An Amazon SNS topic that you created and to which you're subscribed. If you already have an SNS topic, you can use an existing one. For more information, see Getting started with Amazon SNS in the *Amazon Simple Notification Service Developer Guide*.

To receive AWS Health events with Amazon Q Developer in chat applications

1. Follow the instructions to set up a rule in Amazon EventBridge to capture AWS Health events outlined in Creating rules using the Enhanced Builder and Creating rules using the Advanced Builder in the *Amazon EventBridge User Guide*. For more information, see Configuring an EventBridge rule to send notifications about events in AWS Health.

a. When you finish setting up the event pattern, add a comma to the last line of the pattern, and add the following line to remove unnecessary chat messages from paginated AWS Health events. See Viewing paginated lists of AWS Health events on EventBridge.

```
"detail.page": ["1"]
```

b. When you choose the target, choose an SNS topic. You use the same SNS topic in the Amazon Q Developer in chat applications console.

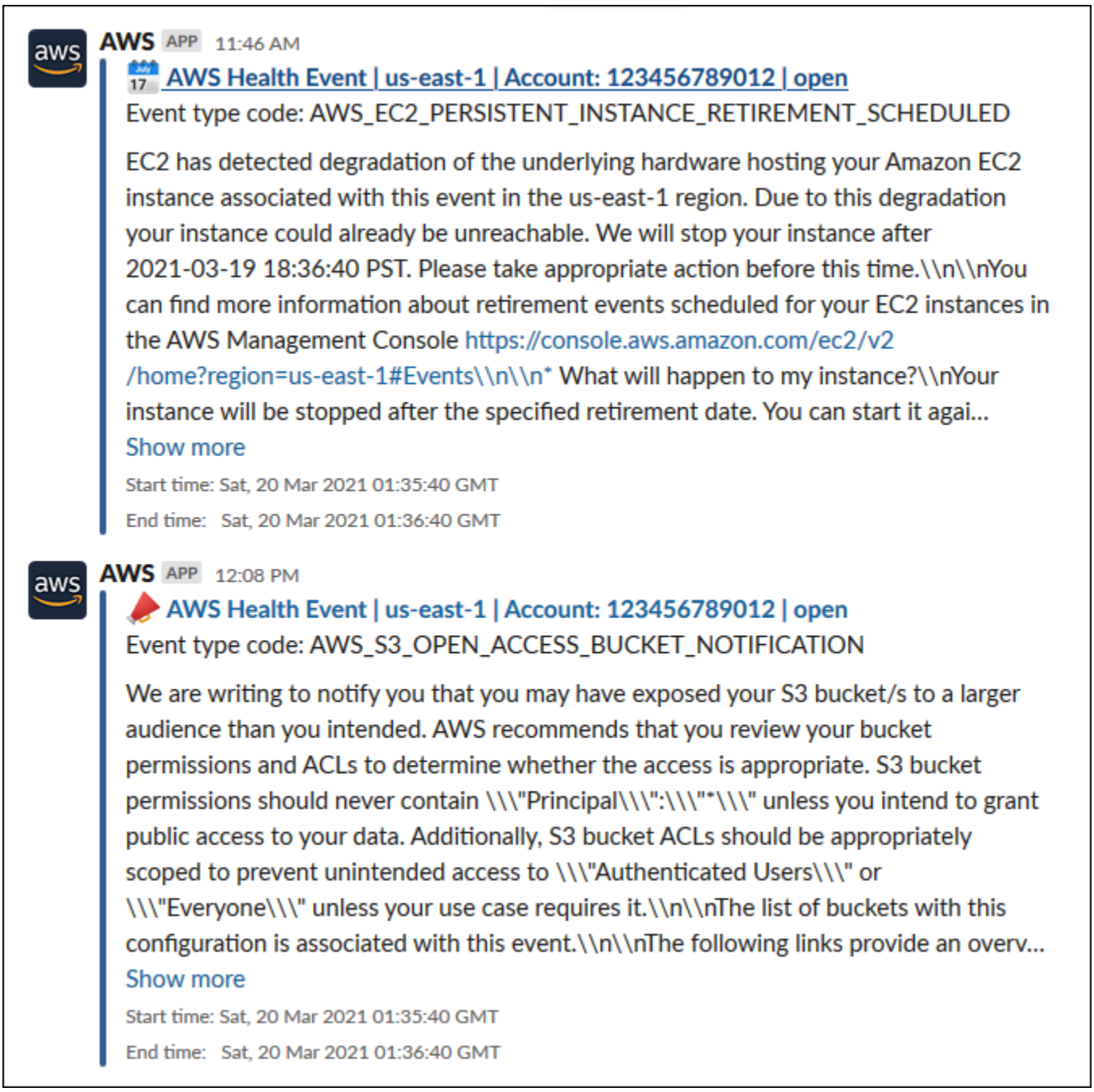
c. Complete the rest of the procedure to create the rule.

2. Navigate to the Amazon Q Developer in chat applications console.
3. Choose your chat client, such as your Slack channel name, and then choose **Edit**.
4. In the **Notifications - optional** section, for **Topics**, choose the same SNS topic that you specified in step 1.
5. Choose **Save**.

   When AWS Health sends an event to EventBridge that matches your rule, the AWS Health event will appear in your chat client.
6. Choose the event name to see more information in your AWS Health Dashboard.

Example: AWS Health events sent to Slack

The following is an example of two AWS Health events for Amazon EC2 and Amazon Simple Storage Service (Amazon S3) in the US East (N. Virginia) Region that appear in the Slack channel.



Running operations on EC2 instances automatically in response to events in AWS Health

You can automate actions that respond to scheduled events for your Amazon EC2 instances. When AWS Health sends an event to your AWS account, your EventBridge rule can then invoke targets, such as AWS Systems Manager Automation documents, to automate actions on your behalf.

For example, when an Amazon EC2 instance retirement event is scheduled for an Amazon Elastic Block Store (Amazon EBS)-backed EC2 instance, AWS Health will send the `AWS_EC2_PERSISTENT_INSTANCE_RETIREMENT_SCHEDULED` event type to your AWS Health Dashboard. When your rule detects this event type, you can automate the stop and start of the instance. This way, you don't have to perform these actions manually.

> **Note**
>
> To automate actions for your Amazon EC2 instances, the instances must be managed by Systems Manager.

For more information, see Automating Amazon EC2 with EventBridge in the *Amazon EC2 User Guide*.

Prerequisites

You must create an AWS Identity and Access Management (IAM) policy, create an IAM role, and update the role's trust policy before you can create a rule.

Create an IAM policy

Follow this procedure to create a customer managed policy for your role. This policy gives the role permission to perform actions on your behalf. This procedure uses the JSON policy editor in the IAM console.

To create an IAM policy

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Policies**.
3. Choose **Create policy**.
4. Choose the **JSON** tab.
5. Copy the following JSON and then replace the default JSON in the editor.

   JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:DescribeInstanceStatus"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": [
        "arn:aws:sns:*:*:Automation*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::123456789012:role/AutomationEVRole"
    }
  ]
}
```

a. In the `Resource` parameter, for the Amazon Resource Name (ARN), enter your AWS account ID.

b. You can also replace the role name or use the default. This example uses *`AutomationEVRole`*.

6. Choose **Next: Tags**.
7. (Optional) You can use tags as key–value pairs to add metadata to the policy.
8. Choose **Next: Review**.
9. On the **Review policy** page, enter a **Name**, such as *`AutomationEVRolePolicy`* and an optional **Description**.
10. Review the **Summary** page to see the permissions that the policy allows. If you're satisfied with your policy, choose **Create policy**.

This policy defines the actions that the role can take. For more information, see Creating IAM policies (console) in the *IAM User Guide*.

Create an IAM role

After you create the policy, you must create an IAM role, and then attach the policy to that role.

To create a role for an AWS service

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**, and then choose **Create role**.
3. For **Select type of trusted entity**, choose **AWS service**.
4. Choose **EC2** for the service that you want to allow to assume this role.
5. Choose **Next: Permissions**.
6. Enter the policy name that you created, such as *`AutomationEVRolePolicy`*, and then select the check box next to the policy.
7. Choose **Next: Tags**.
8. (Optional) You can use tags as key–value pairs to add metadata to the role.
9. Choose **Next: Review**.

10. For **Role name**, enter *AutomationEVRole*. This name must be the same name that appears in the ARN of the IAM policy that you created.
11. (Optional) For **Role description**, enter a description for the role.
12. Review the role and then choose **Create role**.

For more information, see Creating a role for an AWS service in the *IAM User Guide*.

Update the trust policy

Finally, you can update the trust policy for the role that you created. You must complete this procedure so that you can choose this role in the EventBridge console.

To update the trust policy for the role

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**.
3. In the list of roles in your AWS account, choose the name of the role that you created, such as *AutomationEVRole*.
4. Choose the **Trust relationships** tab, and then choose **Edit trust relationship**.
5. For **Policy Document**, copy the following JSON, remove the default policy, and paste the copied JSON in its place.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "ssm.amazonaws.com",
                    "events.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
```

```
    ]
}
```

6. Choose **Update Trust Policy**.

For more information, see Modifying a role trust policy (console) in the *IAM User Guide*.

Create a rule for EventBridge

Follow this procedure to create a rule in the EventBridge console so that you can automate the stop and start of EC2 instances that are scheduled for retirement.

To create a rule for EventBridge for Systems Manager automated actions

1. Open the Amazon EventBridge console at https://console.aws.amazon.com/events/.
2. In the navigation pane, under **Events**, choose **Rules**.
3. On the **Create rule** page, enter a **Name** and **Description** for your rule.
4. Under **Define pattern**, choose **Event pattern**, and then choose **Pre-defined pattern by service**.
5. For **Service provider**, choose **AWS**.
6. For **Service name**, choose **Health**.
7. For **Event type**, choose **Specific Health events**.
8. Choose **Specific service(s)** and then choose **EC2**.
9. Choose **Specific event type category(s)** and then choose **scheduledChange**.
10. Choose **Specific event types code(s)** and then choose the event type code.

    For example, for Amazon EC2 EBS-backed instances, choose **`AWS_EC2_PERSISTENT_INSTANCE_RETIREMENT_SCHEDULED`**. For Amazon EC2 instance store-backed instances, choose **`AWS_EC2_INSTANCE_RETIREMENT_SCHEDULED`**.
11. Choose **Any resource**.

    Your **Event pattern** will look similar to the following example.

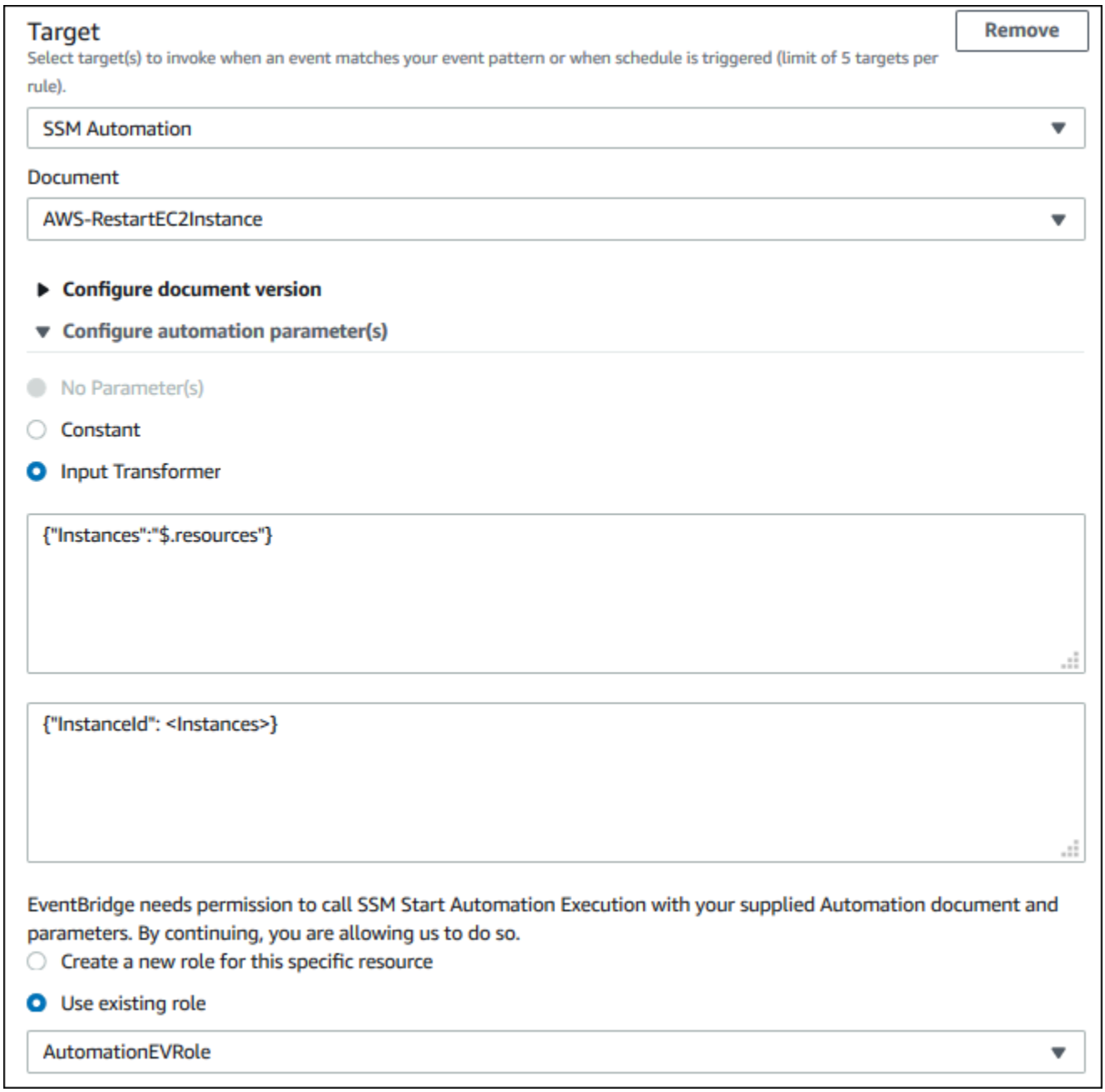
    Example

```
{
```

```
  "source": [
    "aws.health"
  ],
  "detail-type": [
    "AWS Health Event"
  ],
  "detail": {
    "service": [
      "EC2"
    ],
    "eventTypeCategory": [
      "scheduledChange"
    ],
    "eventTypeCode": [
      "AWS_EC2_PERSISTENT_INSTANCE_RETIREMENT_SCHEDULED"
    ]
  }
}
```

12. Add the Systems Manager Automation document target. Under **Select targets**, for **Target**, choose **SSM Automation**.
13. For **Document**, choose `AWS-RestartEC2Instance`.
14. Expand the **Configure automation parameters(s)** and then choose **Input Transformer**.
15. For the **Input Path** field, enter **`{"Instances":"$.resources"}`**.
16. For the second field, enter **`{"InstanceId": <Instances>}`**.
17. Choose **Use existing role**, and then choose the IAM role that you created, such as *`AutomationEVRole`*.

    Your target should look like the following example.



> **Note**
>
> If you don't have an existing IAM role with the required EC2 and Systems Manager permissions and trusted relationship, your role won't appear in the list. For more information, see Prerequisites.

18. Choose **Create**.

If an event occurs in your account that matches your rule, EventBridge will send the event to your specified target.

Reference: AWS Health events Amazon EventBridge schema

The following is the schema for AWS Health events. The contents of the details parameter follows in a second table. Sample payloads ares provided after the schema tables.

AWS Health event schema

AWS Health event schema

Parameter	Description	Required
version	EventBridge version, currently "0".	Yes
id	The unique identifier for the EventBridge event.	Yes
detail-type	The type of detail. For AWS Health events, supported values are `&AWS Health Event` and `AWS Health Abuse Event`	Yes
source	The event bus source.	Yes

Parameter	Description	Required
	For AWS Health events, the supported value is `aws.health`	

Parameter	Description	Required
account	The account ID to which the AWS Health event was sent . **Note** For organizational views this is a different account than the affected account if it's received in the management account or delegated administrator account.	Yes

Parameter	Description	Required
time	The time at which the notification was sent to EventBridge. Format: `yyyy-mm-ddThh:mm:ssZ` .	Yes

Parameter	Description	Required
region	The AWS Region that the notification was delivered to. **Note** This field doesn't indicate the impacted Region for this AWS Health event. That information is reported in `detail.entRegi`.	Yes

Parameter	Description	Required
resources	Describes the list of affected resources, if any, within an account. This field is empty if there are no resources referenced.	No
detail	The section containing details of the AWS Health event, as described in the table immediately following this one.	Yes

Schema content of the 'details' parameter

The following table documents the content of the **detail** parameter in the AWS Health event schema.

AWS Health event schema: detail parameter content

'detail' parameter content	Description	Required
eventArn	The unique identifier for the AWS Health event for the	Yes

'detail' parameter content	Description	Required
	specific Region, including the Region and event ID. **Note** An event ARN isn't unique to a specific AWS account or Region.	
service	The AWS service affected by the AWS Health event. For example, Amazon EC2, Amazon Simple Storage Service, Amazon Redshift, or Amazon Relational Database Service.	Yes

'detail' parameter content	Description	Required
eventTypeCode	The unique identifier for the event type. For example: `AWS_EC2_INSTANCE_NETWORK_MAINTENANCE_SCHEDULED` and `AWS_EC2_INSTANCE_REBOOT_MAINTENANCE_SCHEDULED`. Events that include `MAINTENANCE_SCHEDULED` are generally pushed out approximately two weeks before the start time. **Note** All new planned lifecycle events have the event type `AWS_{SERVICE}_PLANNED_LIFECYCLE_EVENT`.	Yes
eventTypeCategory	The category code of the event. The supported values include `issue`, `accountNotification`, `investigation`, and `scheduledChange`.	Yes

'detail' parameter content	Description	Required
eventScopeCode	Indicates whether the AWS Health event is account-specific or public. Supported values are `ACCOUNT_SPECIFIC` or `PUBLIC`.	Yes
communicationId	A unique identifier for this communication for the AWS Health event. Messages with the same communication ID might be backup messages or pages of a single AWS Health event. This identifier can be used with the account ID to help de-duplicate messages. With the AWS Health event pagination support, the communication ID includes the page number to keep the communication ID unique across pages, for example, 12345678910-1. For more information, see Viewing paginated lists of AWS Health events on EventBridge.	Yes
startTime	The start time of the AWS Health event, in the format `DoW, DD, MMM, YYYY, HH:MM:SS TZ`. The start time can be in the future for scheduled events.	Yes

'detail' parameter content	Description	Required
endTime	The end time of the AWS Health event, in the format:`DoW, DD MMM YYYY HH:MM:SS TZ`. The end time can't be provided for events scheduled for a future time.	No
lastUpdatedTime	The last update time for the AWS Health event, in the format `DoW, DD MMM YYYY HH:MM:SS TZ`.	Yes
statusCode	The status of the AWS Health event. Supported values include `open`, `closed`, and `upcoming`.	Yes
eventRegion	The impacted Region described by this AWS Health event.	Yes

'detail' parameter content	Description	Required
eventDescription	A section that describes the AWS Health event. This includes fields for language and text to describe the event. • **language** – The code for the language used in the AWS Health event. This is typically determined by the Region that the event is published to. For example, in the `us-east-1` Region, this is typically en_US. • **latestDescription** – Describes the AWS Health event as it is rendered from the AWS Health API and typically appears on the the AWS Health dashboard. **Note** For public events, this contains only the latest update and not the entire history of the event.	Yes

'detail' parameter content	Description	Required
eventMetadata	Additional event metadata that can be provided for the AWS Health event. • **<metadata key 1>** – Metadata key-value pair strings: "keystring1": "keyvalue1" The key-value pairs for event metadata are determined by the service that sent the AWS Health event.	No

'detail' parameter content	Description	Required
affectedEntities	An array that describes the resource value and status of affected resources within the AWS Health event. • **entityValue** – The resource/entity ID. • **lastUpdatedtime** – The time when this resource/entity status was last updated, in the format `DoW, DD MMM YYYY HH:MM:SS TZ`. • **status** – The status of the affected resource/entity. Supported values include `IMPAIRED`, `UNIMPAIRED` , `PENDING`, `RESOLVED`, and `UNKNOWN`. **Note** For planned lifecycle events, resource state updates are performed asynchron ously and periodically, and can have a delay of up to 72 hours in rare occasions. For more information, see What should I expect when I receive	No

'detail' parameter content	Description	Required
	a planned lifecycle event notification?.	
page	The page this message represents. For more information, see Viewing paginated lists of AWS Health events on EventBridge. **Note** Pagination occurs only on resources . If the 256KB size limit is exceeded for another reason, the communication to fail.	Yes
totalPages	The total number of pages for this health event. For more information, see Viewing paginated lists of AWS Health events on EventBridge. You can use this value to determine whether you received all of the pages of a multi-page communication for an account.	Yes

'detail' parameter content	Description	Required
backupEvent	This flag filters out backup events in the designated backup region within a partition if customers don't want to leverage redundancy. This value can be true or false.	Yes
affectedAccount	The account ID of the impacted account. This may be different from the value in the `account` field if this health event is sent to an account that is part of an AWS Organizations and is received in the management account or delegated administrator account.	Yes
actionability	Metadata to activate programmatic determination of which events require action without manual inspection. Possible (single) value can be `ACTION_REQUIRED`, `ACTION_MAY_BE_REQUIRED`, or `INFORMATIONAL`.	No

'detail' parameter content	Description	Required
personas	This list of metadata activates programmatic determination of which stakeholder to route the event to. Possible (multiple) values are `OPERATIONAL` , `SECURITY`, and `BILLING`.	No

Public Health Event - Amazon EC2 operational issue


```
{
    "version": "0",
    "id": "7bf73129-1428-4cd3-a780-95db273d1602",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
    "account": "123456789012",
    "time": "2023-01-27T09:01:22Z",
    "region": "af-south-1",
    "resources": [],
    "detail": {
        "eventArn": "arn:aws:health:af-south-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_7f35c8ae-af1f-54e6-a526-d0179ed6d68f",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
        "eventTypeCategory": "issue",
        "eventScopeCode": "PUBLIC",
        "communicationId": "01b0993207d81a09dcd552ebd1e633e36cf1f09a-1",
        "startTime": "Fri, 27 Jan 2023 06:02:51 GMT",
        "endTime": "Fri, 27 Jan 2023 09:01:22 GMT",
        "lastUpdatedTime": "Fri, 27 Jan 2023 09:01:22 GMT",
        "statusCode": "open",
        "eventRegion": "af-south-1",
        "eventDescription": [{
            "language": "en_US",
            "latestDescription": "Current severity level: Operating normally\n
\n[RESOLVED] \n\n [03:15 PM PST] We continue see recovery \n\nThe following AWS
 services were previously impacted but are now operating normally: APPSYNC, BACKUP,
 EVENTS."
```

```
        }],
        "affectedEntities": [],
        "page": "1",
        "totalPages": "1",
        "backupEvent": "false",
        "affectedAccount": "123456789012",
        "personas": ["OPERATIONS"]
    }
}
```

Account-specific AWS Health Event - Elastic Load Balancing API Issue

```
{
    "version": "0",
    "id": "121345678-1234-1234-1234-123456789012",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
    "account": "123456789012",
    "time": "2022-06-10T06:27:57Z",
    "region": "ap-southeast-2",
    "resources": [],
    "detail": {
        "eventArn": "arn:aws:health:ap-southeast-2::event/
AWS_ELASTICLOADBALANCING_API_ISSUE_90353408594353980",
        "service": "ELASTICLOADBALANCING",
        "eventTypeCode": "AWS_ELASTICLOADBALANCING_API_ISSUE",
        "eventTypeCategory": "issue",
        "eventScopeCode": "ACCOUNT_SPECIFIC",
        "communicationId": "01b0993207d81a09dcd552ebd1e633e36cf1f09a-1",
        "startTime": "Fri, 10 Jun 2022 05:01:10 GMT",
        "endTime": "Fri, 10 Jun 2022 05:30:57 GMT",
        "statusCode": "open",
        "eventRegion": "ap-southeast-2",
        "eventDescription": [{
            "language": "en_US",
            "latestDescription": "A description of the event will be provided here"
        }],
        "page": "1",
        "totalPages": "1",
        "backupEvent": "false",
        "affectedAccount": "123456789012",
        "personas": ["OPERATIONS"]
    }
```

```
}
```

Account-specific AWS Health Event - backup event for Amazon EC2 Instance Store Drive Performance Degraded

```
{
    "version": "0",
    "id": "121345678-1234-1234-1234-123456789012",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
    "account": "123456789012",
    "time": "2022-06-03T06:27:57Z",
    "region": "us-west-2",
    "resources": [
        "i-abcd1111"
    ],
    "detail": {
        "eventArn": "arn:aws:health:us-east-1::event/
AWS_EC2_INSTANCE_STORE_DRIVE_PERFORMANCE_DEGRADED_90353408594353980",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_INSTANCE_STORE_DRIVE_PERFORMANCE_DEGRADED",
        "eventTypeCategory": "issue",
        "eventScopeCode": "ACCOUNT_SPECIFIC",
        "communicationId": "01b0993207d81a09dcd552ebd1e633e36cf1f09a-1",
        "startTime": "Fri, 3 Jun 2022 05:01:10 GMT",
        "endTime": "Fri, 3 Jun 2022 05:30:57 GMT",
        "statusCode": "open",
        "eventRegion": "us-east-1",
        "eventDescription": [{
            "language": "en_US",
            "latestDescription": "A description of the event will be provided here"
        }],
        "affectedEntities": [{
            "entityValue": "i-abcd1111"
        }],
        "page": "1",
        "totalPages": "1",
        "backupEvent": "true",
        "affectedAccount": "123456789012",
        "personas": ["OPERATIONS"]
    }
}
```

Account-specific AWS Health Event - Amazon EC2 Instance Retirement


```
{
    "version": "0",
    "id": "7bf73129-1428-4cd3-a780-95db273d1602",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
    "account": "123456789012",
    "time": "2026-01-27T01:43:21Z",
    "region": "us-east-1",
    "detail": {
        "eventArn": "arn:aws:health:us-east-1::event/
AWS_EC2_INSTANCE_RETIREMENT_SCHEDULED_90353408594353983",
        "service": "EC2",
        "eventTypeCode": "AWS_EC2_INSTANCE_RETIREMENT_SCHEDULED",
        "eventTypeCategory": "scheduledChange",
        "eventScopeCode": "ACCOUNT_SPECIFIC",
        "communicationId": "1234abc01232a4012345678-1",
        "startTime": "Thu, 27 Aug 2026 13:19:03 GMT",
        "lastUpdatedTime": "Thu, 27 Jan 2026 13:44:13 GMT",
        "statusCode": "open",
        "eventRegion": "us-east-1",
        "eventDescription": [{
            "language": "en_US",
            "latestDescription": "A description of the event will be provided here"
        }],
        "eventMetadata": {
            "keystring1": "valuestring1",
            "keystring2": "valuestring2",
            "keystring3": "valuestring3",
            "keystring4": "valuestring4",
            "truncated": "true"
        },
        "affectedEntities": [{
            "entityValue": "arn:aws:ec2:us-east-1:123456789012:instance/
i-1234567890abcdef0",
            "lastUpdatedTime": "Thu, 26 Jan 2026 19:01:55 GMT",
            "status": "PENDING"
        }],
        "affectedAccount": "123456789012",
        "page": "1",
        "totalPages": "1",
        "backupEvent": "false",
```

```
        "personas": ["OPERATIONS"],
        "actionability": "ACTION_REQUIRED"
    }
}
```

Account-specific AWS Health Event - Lambda Planned Lifecycle Event

```
{
    "version": "0",
    "id": "7bf73129-1428-4cd3-a780-95db273d1602",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
    "account": "123456789012",
    "time": "2023-01-27T01:43:21Z",
    "region": "us-west-2",
    "resources": ["arn:lambda-1-101002929", "arn:lambda-1-101002930",
 "arn:lambda-1-101002931", "arn:lambda-1-101002932"],
    "detail": {
        "eventArn": "arn:aws:health:us-west-2::event/
AWS_LAMBDA_PLANNED_LIFECYCLE_EVENT_90353408594353980",
        "service": "LAMBDA",
        "eventTypeCode": "AWS_LAMBDA_PLANNED_LIFECYCLE_EVENT",
        "eventTypeCategory": "scheduledChange",
        "eventScopeCode": "ACCOUNT_SPECIFIC",
        "communicationId": "1234abc01232a4012345678-1",
        "startTime": "Thu, 27 Aug 2026 13:19:03 GMT",
        "lastUpdatedTime": "Thu, 27 Jan 2026 13:44:13 GMT",
        "statusCode": "open",
        "eventRegion": "us-west-2",
        "eventDescription": [{
            "language": "en_US",
            "latestDescription": "A description of the event will be provided here"
        }],
        "eventMetadata": {
            "keystring1": "valuestring1",
            "keystring2": "valuestring2",
            "keystring3": "valuestring3",
            "keystring4": "valuestring4",
            "truncated": "true"
        },
        "affectedEntities": [{
            "entityValue": "arn:lambda-1-101002929",
            "lastUpdatedTime": "Thu, 26 Jan 2026 19:01:55 GMT",
```

```
            "status": "PENDING"
        }, {
            "entityValue": "arn:lambda-1-101002930",
            "lastUpdatedTime": "Thu, 26 Jan 2026 19:05:12 GMT",
            "status": "PENDING"
        }, {
            "entityValue": "arn:lambda-1-101002931",
            "lastUpdatedTime": "Thu, 26 Jan 2026 19:07:13 GMT",
            "status": "PENDING"
        }, {
            "entityValue": "arn:lambda-1-101002932",
            "lastUpdatedTime": "Thu, 26 Jan 2026 19:10:59 GMT",
            "status": "RESOLVED"
        }],
        "affectedAccount": "123456789012",
        "page": "1",
        "totalPages": "10",
        "backupEvent": "false",
        "personas": ["OPERATIONS"],
        "actionability": "ACTION_REQUIRED"
    }
}
```

Monitoring AWS Health

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Health and your other AWS solutions. AWS provides the following monitoring tools to watch AWS Health, report when something is wrong, and take actions when appropriate:

- Amazon CloudWatch monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For more information, see the Amazon CloudWatch User Guide.

  You can use Amazon EventBridge so that you're notified about AWS Health events that might affect your services and resources. For example, if AWS Health publishes an event about your Amazon EC2 instances, you can use these notifications to take action and update or replace your resources as needed. For more information, see Monitoring events in AWS Health with Amazon EventBridge.
- AWS CloudTrail captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

Topics

- Logging AWS Health API calls with AWS CloudTrail

Logging AWS Health API calls with AWS CloudTrail

AWS Health is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Health. CloudTrail captures API calls for AWS Health as events. The calls captured include calls from the AWS Health console and code calls to the AWS Health API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS Health. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to AWS Health, the IP address that the request was made from, who made the request, when it was made, and additional details.

To learn more about CloudTrail, including how to configure and enable it, see the AWS CloudTrail User Guide.

AWS Health information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When supported event activity occurs in AWS Health, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for AWS Health, create a *trail*. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All AWS Health API operations are logged by CloudTrail and are documented in the AWS Health API Reference. For example, calls to the `DescribeEvents`, `DescribeEventDetails`, and `DescribeAffectedEntities` operations generate entries in the CloudTrail log files.

AWS Health supports logging the following actions as events in CloudTrail log files:

- Whether the request was made with root or IAM credentials
- Whether the request was made with temporary security credentials for a role or federated user
- Whether the request was made by another AWS service

For more information, see the CloudTrail userIdentity Element.

You can store your log files in your Amazon S3 bucket for as long as you want. You can also define Amazon S3 lifecycle rules to archive or delete log files automatically. By default, your log files are encrypted with Amazon S3 server-side encryption (SSE).

To be notified upon log file delivery, you can configure CloudTrail to publish Amazon SNS notifications when new log files are delivered. For more information, see Configuring Amazon SNS Notifications for CloudTrail.

You can also aggregate AWS Health log files from multiple AWS regions and multiple AWS accounts into a single Amazon S3 bucket.

For more information, see Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts.

Example: AWS Health log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the DescribeEntityAggregates operation.


```
{
   "Records": [
   {
   "eventVersion": "1.05",
   "userIdentity": {
      "type": "IAMUser",
      "principalId": "AIDACKCEVSQ6C2EXAMPLE",
      "arn": "arn:aws:iam::123456789012:user/JaneDoe",
      "accountId": "123456789012",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "userName": "JaneDoe",
      "sessionContext": {"attributes": {
         "mfaAuthenticated": "false",
         "creationDate": "2016-11-21T07:06:15Z"
      }},
      "invokedBy": "AWS Internal"
```

```
    },
   "eventTime": "2016-11-21T07:06:28Z",
   "eventSource": "health.amazonaws.com",
   "eventName": "DescribeEntityAggregates",
   "awsRegion": "us-east-1",
   "sourceIPAddress": "203.0.113.0",
   "userAgent": "AWS Internal",
   "requestParameters": {"eventArns": ["arn:aws:health:us-east-1::event/EBS/
EBS_LOST_VOLUME/EBS_LOST_VOLUME_123"]},
   "responseElements": null,
   "requestID": "05b299bc-afb9-11e6-8ef4-c34387f40bd4",
   "eventID": "e4deb9dc-dbc2-4bdb-8515-73e8abcbc29b",
   "eventType": "AwsApiCall",
   "recipientAccountId": "123456789012"
   }
   ],
   ...
}
```

FAQs on supported protocols

The AWS Health service is adding SDK support for the Smithy RPC v2 CBOR protocol in addition to the existing AWS JSON protocol. The Smithy RPC v2 CBOR protocol is more performant than AWS JSON for most workloads. AWS SDKs will prioritize the protocol that is the most performant for each language.

What are the AWS JSON and Smithy RPC v2 CBOR protocols, and how do they differ?

A wire protocol acts as a mediator between a client and the service. The AWS SDK handles the serialization (converting an object to the wire format) and deserialization (converting the wire format back to an object) between the client and server transparently for both protocols.

AWS JSON - JSON is one of the most widely used and accepted data formats for communication between systems. This service uses AWS JSON as a medium to communicate between an AWS SDK client and the server. An HTTP request for a service API operation accepts input in the form of JSON, the operation is executed, and the response is returned to the SDK client in the form of JSON.

Smithy RPC v2 CBOR - Smithy RPC v2 CBOR (Concise Binary Object Representation) is a protocol developed by Amazon for more efficient data transmission. It is a data format designed to keep message sizes compact and to allow the format to evolve without requiring version negotiation between systems. The key difference from AWS JSON is that AWS JSON uses human-readable text (like `{"name": "John"}`), while Smithy RPC v2 CBOR uses a binary format (sequences of 1s and 0s). This binary approach creates smaller data packages that travel faster over the network and reduces serialization time since computers process binary data directly.

How do I get started with the Smithy RPC v2 CBOR protocol?

To get started, use the latest AWS SDK/CLI version to use the Smithy RPC v2 CBOR protocol for this service. The SDK will automatically use the Smithy RPC v2 CBOR protocol when available. Upgrade your AWS SDK to the specified version or any subsequent version.

Starting 2026-10-06, the following AWS SDKs will use the Smithy RPC v2 CBOR protocol, while the other SDKs will continue using the AWS JSON protocol:

- AWS SDK for C++
- AWS SDK for Go V2
- AWS SDK for Java 2.x
- AWS SDK for .NET v4
- AWS Tools for PowerShell v5
- AWS SDK for Swift
- AWS SDK for Kotlin

What are the risks of enabling the Smithy RPC v2 CBOR protocol?

The added support of Smithy RPC v2 CBOR in the SDKs is expected to be fully transparent and effortless for customers. However, if you are using a custom implementation of the AWS SDK, or a combination of custom clients and the AWS SDK, that depends on AWS JSON-specific serialization behavior, it may be incompatible with the Smithy RPC v2 CBOR protocol.

What if my solution does not support Smithy RPC v2 CBOR?

To use the AWS JSON protocol, downgrade your SDK to a version released before 2026-10-06.

What AWS Regions support the Smithy RPC v2 CBOR protocol?

The Smithy RPC v2 CBOR protocol is supported in all AWS Regions where this service is available.

What latency improvements can I expect when upgrading to SDK versions that support Smithy RPC v2 CBOR?

Smithy RPC v2 CBOR provides up to 13% lower latency and up to 15% reduction in payload size compared to AWS JSON. The binary encoding eliminates the overhead of JSON text parsing and string escaping, which reduces both network transfer time and serialization cost.

Will the AWS JSON protocol be deprecated?

AWS JSON protocol will continue to be supported through older versions of SDKs. Also, some language-specific SDKs will continue using AWS JSON as it is more performant than Smithy RPC

v2 CBOR due to the available JSON implementation for the language being faster than its CBOR implementation.

Where can I find more information about Smithy RPC v2 CBOR?

You can find more information about the AWS JSON protocol at AWS JSON 1.0 protocol and AWS JSON 1.1 protocol in the Smithy documentation, and about Smithy RPC v2 CBOR at Smithy RPC v2 CBOR protocol in the Smithy documentation.

Document history for AWS Health

The following table describes the documentation for this release of AWS Health.

- **API version:** 2016-08-04

The following table describes important updates to the AWS Health documentation, beginning in August 28, 2020. You can subscribe to the RSS feed to receive notifications about the updates.

Change	Description	Date
Added the AWSServiceRoleForTagsHealth service-linked role and the TagsHealthServiceRolePolicy managed policy	Added the AWSServiceRoleForTagsHealth service-linked role and the TagsHealthServiceRolePolicy AWS managed policy. The policy grants AWS Health read-only access to the tags on the resources in your account. For more information, see Using service-linked roles for AWS Health and AWS Health updates to AWS managed policies.	October 30, 2026
Added FAQ page for supported protocols	Added a new FAQ page with answers to common questions about the AWS JSON and Smithy RPC v2 CBOR wire protocols. For more information, see FAQs on supported protocols.	October 6, 2026
Added the version catalog	Added a new page that describes the version catalog for AWS Health, which provides a service-wide view	October 2, 2026

	of supported versions and their timelines. For more information, see Version catalog for AWS Health.	
Clarified that public events aren't available through AWS User Notifications	Added callouts to clarify that public events aren't available through AWS managed notifications in AWS User Notifications. To receive public events, you must create an Amazon EventBridge rule or a User-configured notification (UCN). For more information, see Public event.	August 26, 2026
Added a note about resource state update timing for planned lifecycle events	Added a note to clarify that resource state updates for planned lifecycle events are performed asynchronously and can have a delay of up to 72 hours in rare occasions. For more information, see Reference: AWS Health events Amazon EventBridge schema.	August 26, 2026
Clarified Region coverage for simplified integration	Updated the simplified integration section to clarify that a single-Region rule aggregates only account-specific events, not public events. Public events are delivered only to the impacted Region and its backup Region. For more information, see Simplified integration.	July 2, 2026

Added end of support notice for AWS Service Management Connector	Added an end of support notice to the Service Management Connector integration section. AWS will end support for AWS Service Management Connector on March 31, 2027. For more information, see AWS Service Management Connector end of support.	April 28, 2026
Updated Configuring an EventBridge rule to send notifications about events in AWS Health	Simplified the procedure for creating EventBridge rules by linking to the Amazon EventBridge User Guide for general rule creation steps. The topic now focuses on AWS Health-specific filtering and use cases. For more information, see Configuring an EventBridge rule to send notifications about events in AWS Health.	March 13, 2026

Updated AWS Health events Amazon EventBridge schema examples	Updated the schema examples to include personas and actionability fields. Examples include Public Health Event for Amazon EC2 operational issue, Account-specific events for Elastic Load Balancing API Issue and Amazon EC2 Instance Store Drive Performance Degraded backup event, and Lambda Planned Lifecycle Event. For more information, see Reference: AWS Health events Amazon EventBridge schema.	March 13, 2026
Updated Manage AWS Health notifications in AWS User Notifications	Updated information to this section to reflect the migration of AWS Health events to AWS User Notifications. For more information, see Manage AWS Health notifications in AWS User Notifications.	December 22, 2025
Updated Monitoring account-specific and public events for AWS Health	Added information to this section detailing backup rule behavior for public events and account-specific events. For more information, see Backup rules for AWS Health events.	December 11, 2025

Added information about the Actionability and Personas fields for Health events	Added information for the Actionability and Personas fields in the Concepts for AWS Health section and the Schema content of the 'details' parameter in the Reference: AWS Health events Amazon EventBridge schema section. For more information, see Concepts for AWS Health and Reference: AWS Health events Amazon EventBridge schema.	November 20, 2025
Updated section: Creating EventBridge rules for AWS Region coverage	Updated information to create EventBridge rules. For more information, see Creating EventBridge rules for AWS Region coverage.	November 3, 2025
Updated section: *Manage AWS Health notifications in AWS User Notifications*	Updated information for the steps to Configure your AWS managed notifications subscription for AWS Health events . For more informati on, see Manage AWS Health notifications in AWS User Notifications.	September 16, 2025
Updated section: *Monitorin g events in AWS Health with Amazon EventBridge*	Updated information in note for AWS Health delivers events to EventBridge. For more information, see Monitoring events in AWS Health with Amazon EventBridge.	September 15, 2025

Updated section: *AWS Health Dashboard*	Removed optional steps to subscribe to RSS feed for health events. Added note that to receive notifications for health events, customers can use EventBridge. For more information, see AWS Health Dashboard .	August 15, 2025
Updated section: *Monitoring events in AWS Health with Amazon EventBridge*	Removed the topic, Installing a service-linked role to use AWS Incident Detection and Response in Monitoring events in AWS Health with Amazon EventBridge	August 8, 2025
Updated section: *Monitoring events in AWS Health with Amazon EventBridge*	Added information to the Notes section indicating that there might be a lag of up to one hour before you start to receive notifications for Public Health Events. For more information, see Monitoring events in AWS Health with Amazon EventBridge	July 22, 2025

Updated section: *Enabling organizational view*	Added information to the Notes section indicating that AWS Health automatically aggregates all historical health events across your organization when you enable organizational view. Historical events might take up to 24 hours to appear in your organizational view. For more information, see Enabling organizational view	June 27, 2025
Updated section: *Aggregating AWS Health events across accounts*	Removed note that AWS Health doesn't show events that occurred before you enabled organizational view. For more information, see Aggregating AWS Health events across accounts	June 27, 2025
WorkDocs deprecated	Removed references to deprecated WorkDocs in Planned lifecycle events for AWS Health.	June 19, 2025
Added note for AWS managed notifications migration timeline	Added a note regarding key dates for the email migration to AWS managed notifications in AWS User Notifications. For more information, see Manage AWS Health notifications in AWS User Notifications.	April 28, 2025

Updated planned lifecycle events	Updated planned lifecycle events to indicate that AWS Health events remain open for 4 years rather than 90 days for unresolved resources . For more information, see the *What should I expect when I receive a planned lifecycle event notification?* section in Planned lifecycle events for AWS Health.	April 18, 2025
Updated the description of the **Affected resources list** for planned lifecycle events	The **Affected resources list** for planned lifecycle events usually refreshes once every 24 hours, but might take up to 72 hours to reflect the current resource status. For more information, see the *Event details* section in Viewing your account events in the AWS Health Dashboard.	April 7, 2025
Added an FAQ for managing AWS Health notifications in AWS User Notifications	For more information, see Manage notifications in AWS User Notifications FAQ.	February 18, 2025
Added information regarding IPv6-only requests to endpoints.	For more information, see Choosing endpoints for AWS Health API requests.	January 28, 2025
Manage AWS Health notifications in AWS User Notifications	For more information, see Manage notifications in AWS User Notifications.	January 16, 2025

Corrected JSON in *Monitoring AWS Health events with Amazon EventBridge*	For more information, see Monitoring AWS Health events with Amazon EventBridge.	September 3, 2024
Updated information on downloading affected resources	For more information, see Affected resources view.	July 27, 2024
Removed Internetwork traffic privacy from the Security section AWS Health documentation	For more information, see Security in AWS Health.	March 27, 2024
Updated the AWS Health Dashboard – Service health and Planned lifecycle events for AWS Health documentation.	For more information, see AWS Health Dashboard – Service health and Planned lifecycle events for AWS Health.	February 15, 2024
Removed a duplicate bullet point in Creating an EventBridge rule for AWS Health	Removed a duplicate bullet point in Creating an EventBridge rule for AWS Health.	December 4, 2023
Added documentation for Planned Lifecycle Events	For more information, see Planned Lifecycle Events for AWS Health.	October 31, 2023
Updated documentation for `AWSHealthFullAccess`	You can now use the `AWSHealthFullAccess` managed policy in the AWS GovCloud (US) Regions. See AWS managed policies for AWS Health.	October 16, 2023

Added documentation for configuring AWS User Notifications in AWS Health.	You can now configure AWS User Notifications in AWS Health. For more information, see Configure AWS User Notifications for AWS Health.	August 30, 2023
Added documentation for the delegated administrator feature to the **Aggregating AWS Health events** section.	For more information, see Delegated administrator organizational view.	July 27, 2023
SLR policy update	Update to AWS managed policy: Health_OrganizationsServiceRolePolicy. For more information, see AWS managed policies for AWS Health.	July 19, 2023
AWS Health schema now supports event metadata	You can now receive event metadata from AWS Health events. For more information, see Monitoring AWS Health events with Amazon EventBridge.	June 20, 2023
Updated documentation for Amazon EventBridge	You can now use an Amazon EventBridge rule to monitor both account-specific and public events. For more information, see Monitoring AWS Health events with Amazon EventBridge.	May 2, 2023

Added documentation for AWS managed policies	Added documentation for the AWS managed policies for AWS Health and Using service-linked roles for AWS Health.	January 18, 2023
Added time zone setting documentation	Use the new time zone feature to view the AWS Health Dashboard in your local time zone or in UTC. For more information, see Getting started with your AWS Health Dashboard – Your account health and the AWS Health Dashboard – Service health.	September 21, 2022
Updated documentation	Added documentation for AWS Health Aware. For more information, see AWS Health Aware.	May 25, 2022
Updated documentation	The Service Health Dashboard and the AWS Personal Health Dashboard have been rebranded to the AWS Health Dashboard. For more information, see Getting started with your AWS Health Dashboard – Your account health and the AWS Health Dashboard – Service health.	February 28, 2022

Updated documentation for Amazon EventBridge	New topic for AWS Health to use Amazon EventBridge to monitor for Health events. For more information, see Monitoring AWS Health events with Amazon EventBridge.	February 3, 2022
Updated documentation	If you have an Enterprise On-Ramp Support plan, you can use the AWS Health API.	November 24, 2021
Added documentation	New topic for AWS Health concepts. For more information, see Concepts for AWS Health.	July 29, 2021
Updated documentation for CloudWatch Events	Added a section about how to create a rule for multiple services and event type categories. For more information, see Creating a rule for multiple services and categories.	May 7, 2021
Updated documentation for CloudWatch Events	Updated the section to automate AWS Systems Manager actions for Amazon CloudWatch Events rules. For more information, see Automating actions for Amazon EC2 instances.	April 28, 2021

Updated documentation for CloudWatch Events	Added a section to receive AWS Health events in your chat client. For more information, see Receiving AWS Health events with Amazon Q Developer in chat applications.	March 16, 2021
Updated documentation	The following topics are updated: • Updated the Aggregating AWS Health events topic • Reorganized and updated the Monitor for AWS Health events with Amazon CloudWatch Events topic • Updated the Resource- and action-based conditions section	January 29, 2021
Added the AWS Health Dashboard for organizational view in the AWS Health console	You can use the AWS Health console to enable the organizational view feature. You can then view health events for member accounts in your AWS organization.	December 14, 2020
High availability endpoint demo	You can use the example code to determine the active regional endpoint and signing AWS Region for AWS Health.	October 22, 2020

Updates to the *AWS Health User Guide*	Organization updates and added an RSS feed so that you can subscribe for the latest updates to the AWS Health documentation.	August 28, 2020

Earlier updates

Change	Description	Date
Updated the organizat ional view topic to include examples.	See Aggregating AWS Health events across accounts.	June 3, 2020
Security and AWS Health	Added information about security considerations when using AWS Health. See Security in AWS Health.	May 5, 2020
Added new section to explain how to use organizational view to events aggregated across all accounts in AWS Organizations.	See Aggregating AWS Health events across accounts.	December 18, 2019
Added new section "Resource- and Action-based Conditions" to explain Events restrictions vended by the AWS Health API.	See Identity and access management for AWS Health.	August 2, 2018
Added a note about the visibility of AWS Health information.	See Identity and access management for AWS Health.	August 16, 2017
Service release.	AWS Health released.	December 1, 2016