aws

User Guide

AWS assistant



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

AWS assistant: 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 the AWS assistant? 1
Getting started with the AWS assistant 3
Step 1: Access the AWS assistant 3
Step 2: Verify required permissions 3
[Optional] Step 3: Review Pro tier access 4
Using the AWS assistant 5
Manage console conversations 5
Navigate the chat panel 6
Chat settings 6
Use visualizations 6
Use the prompt library 7
Chatting about your AWS resources 7
How resource analysis works 8
Required permissions 8
Ask resource questions 8
Troubleshoot resources 9
Chatting about your AWS costs 10
Cost capabilities 11
How cost analysis works 11
Prerequisites 11
Example cost questions 12
Using service-specific insights 13
Chatting about telemetry and operations 13
Chatting about message delivery 13
Using plugins 14
CloudZero 15
Datadog 22
Wiz 29
Diagnosing console errors 35
Required permissions 36
Diagnosing a console error 36
Contacting AWS Support 36
Prerequisites 37
Specify the right service 37

Create a support case ... 37
Security in the AWS assistant ... 39
Security best practices ... 39
Detective ... 39
Preventative ... 39
Identity and access management ... 39
Logging and incident response ... 40
Compliance validation ... 40
Resilience ... 41
Infrastructure security ... 41
Configuration and vulnerability analysis ... 41
Cross-service confused deputy prevention ... 42
Data protection ... 42
Encryption at rest ... 42
Encryption in transit ... 42
Key management ... 42
Data storage ... 43
Service improvement ... 44
Cross-Region processing ... 45
Permissions reference ... 46
Using `q:PassRequest` ... 49
Policy examples ... 50
User permissions ... 50
Service control policies ... 64
Supported AWS Regions ... 66
Regions that are enabled by default ... 66
Regions that are disabled by default ... 67
Administration Regions ... 68
Working with other services ... 69
Access, permission, and response errors ... 70
The AWS assistant icon appears, but you can't use it ... 70
Response reports insufficient permissions ... 70
The AWS assistant can't find a resource ... 71
Conversation or message is missing ... 71
Error related to AWS KMS grants ... 72
Response is incomplete or incorrect ... 72

Quotas 74
Conversation quotas 74
Usage quotas 74
Document history 76

What is the AWS assistant?

Amazon Q Developer in the AWS Management Console is now the AWS assistant

Amazon Q Developer in the AWS Management Console is now the AWS assistant. You can use the AWS assistant in the AWS Management Console and AWS Console Mobile Application.
Technical names didn't change. The q prefix still appears in AWS Identity and Access Management (IAM) actions and managed policies. It also appears in service principals, ARNs, AWS CloudTrail events, and resource names.

Use the AWS assistant, a generative AI-powered assistant, in the AWS Management Console and AWS Console Mobile Application. Ask questions in plain language to learn about AWS services. In the AWS Management Console, you can also ask about your resources and costs, fix problems, and contact AWS Support.

The AWS assistant can help you:

- Learn about AWS architecture, services, recommended practices, and documentation.
- Retrieve and analyze information about resources that your identity has permission to access.
- Analyze historical and forecasted costs, identify optimization opportunities, and understand AWS pricing.
- Troubleshoot resources, alarms, telemetry, and supported console errors.
- Review email-delivery information for supported services.
- Create an AWS Support case and transfer to a support agent when your support offering permits it.
- Generate short scripts and code snippets for AWS SDKs and the AWS Command Line Interface.

For complex resource questions, the AWS assistant creates a sequence of steps and explains the reasoning behind each one. If the first approach fails, it tries another method or asks you for more information.

For example, you can ask "What SQS queues invoke my AWS Lambda functions?" The AWS assistant makes read-only get, list, and describe calls to AWS Lambda and Amazon Simple Queue

Service (Amazon SQS). It explains each step and then summarizes which queues invoke which functions.

To get started, see Getting started with the AWS assistant.

To learn about features, see Using the AWS assistant. To control access and protect data, see Security in the AWS assistant. For help with problems, see Troubleshooting access, permission, and response errors in the AWS assistant and Contacting AWS Support with the AWS assistant.

The AWS assistant is available at the Free and Pro tiers. For tier requirements, see [Optional] Step 3: Review Pro tier access.

The AWS assistant is constantly updating its capabilities so your questions get the most contextually relevant and actionable answers.

> **Note**
>
> Features vary between the AWS Management Console and the AWS Console Mobile Application. Each topic in this guide says where a feature is available. The AWS assistant name doesn't apply to Amazon Q Developer in the IDE or on the command line, Amazon Q Developer in chat applications, Kiro, Amazon Q Business, Amazon Quick, AWS Transform, or other independently branded products.

Getting started with the AWS assistant

Use the AWS assistant for general questions in the AWS Management Console and AWS Console Mobile Application. Available capabilities vary between the AWS Management Console and the AWS Console Mobile Application.

Step 1: Access the AWS assistant

Choose where you want to use the AWS assistant.

AWS Management Console

1. Sign in to the AWS Management Console at https://console.aws.amazon.com/.
2. Choose the AWS assistant icon in the unified navigation bar. The AWS assistant chat panel opens as a side panel.
3. Enter a question in the text field, and review the response and its source links.

AWS Console Mobile Application

1. Open the AWS Console Mobile Application.
2. Choose the AWS assistant icon.
3. Enter a question, and review the response and its source links.

Step 2: Verify required permissions

Your IAM identity needs permission to use the AWS assistant. The existing `AmazonQDeveloperAccess` AWS managed policy provides quick access to the base conversational features.

To chat, your IAM identity needs permission for `q:StartConversation`, `q:SendMessage`, `q:GetConversation`, and `q:ListConversations`. If one of these actions isn't allowed, you get a permissions error. For all AWS assistant permissions, see Permissions reference for the AWS assistant.

You need additional permissions when the AWS assistant accesses service APIs or resources on your behalf. These permissions can cover cost data, support cases, or customer managed AWS Key Management Service (AWS KMS) keys.

The AWS assistant does not give you access to resources or actions that your IAM identity cannot access. For example, to list Amazon Simple Storage Service (Amazon S3) buckets, your identity must have `s3:ListAllMyBuckets`.

For example policies, see Identity-based policy examples for the AWS assistant.

[Optional] Step 3: Review Pro tier access

If you already have an Amazon Q Developer Pro subscription and want to use Pro tier benefits in the AWS Management Console, use AWS IAM Identity Center. If your AWS IAM Identity Center identity has an active Amazon Q Developer Pro subscription, the subscription is available automatically.

Customer managed AWS KMS keys are available only with a Pro subscription. For more information, see Key management.

If you access the AWS Management Console with IAM or IAM federation, you might be prompted to authenticate with AWS IAM Identity Center. This prompt can appear after you reach a Free tier quota or use a Pro-only capability. For tier quotas, see Amazon Q Developer pricing.

Note the following identity requirements:

- The AWS assistant doesn't support AWS Builder ID.
- Pro tier access in the AWS Management Console requires an organization instance of IAM Identity Center with identity-enhanced console sessions enabled. If your identity is in an account instance, you can use the Free tier only.
- If you're subscribed to Amazon Q Developer Pro in the management account of an organization in AWS Organizations, you can use your subscription while signed in to a member account only if an administrator enables profile sharing. Otherwise, you use the Free tier in member accounts. For more information, see Enabling profile sharing in Amazon Q Developer in the *Amazon Q Developer User Guide*.

Normal AWS service charges can apply when the AWS assistant calls other services on your behalf.

Using the AWS assistant

Use the AWS assistant to ask about AWS architecture, services, recommended practices, and documentation. In the AWS Management Console, you can also ask about your resources, costs, telemetry, and operations. The following topics say where each capability is available.

Topics

- Manage console conversations
- Navigate the chat panel
- Chat settings
- Use visualizations
- Use the prompt library
- Chatting about your AWS resources
- Chatting about your AWS costs
- Using service-specific insights
- Using plugins with the AWS assistant
- Diagnosing console errors with the AWS assistant
- Contacting AWS Support with the AWS assistant

Manage console conversations

The AWS assistant maintains the questions and responses in a conversation as context for later responses. You can save up to 80 conversations in the AWS Management Console.

To start a new conversation, choose the plus icon (+) in the chat panel header. To name a conversation, choose the conversation name field and enter a name. If you don't name it, the AWS assistant generates a title from the selected prompt or the first questions in the conversation.

To switch conversations, choose the clock icon in the chat panel header. The **Conversations** pop-up opens. Choose the conversation that you want to resume. To delete a conversation, choose the delete icon next to its name.

The AWS assistant deletes a conversation when you don't ask a new question in it for 90 days. It also deletes messages older than 90 days, even when the conversation remains active. In the AWS Management Console, the AWS assistant keeps your current conversation and context when

you navigate to another console page or browser tab. In the AWS Console Mobile Application, navigating to a new page starts a conversation without the previous context.

Navigate the chat panel

- To expand the panel to full screen, choose the maximize icon in the chat panel header. To exit full-screen mode, choose the resize icon.
- Use the divider to adjust the panel width.
- To close the panel, choose < in the chat panel header. To close the panel with visualizations, choose X. To reopen the panel, choose the AWS assistant icon in the navigation bar.
- Your work is saved when you switch between the assistant and service console views.

Chat settings

To view chat settings, choose the gear icon in the chat panel header. When you open the panel in the AWS Management Console, the AWS assistant uses the AWS Region that is selected on the console for resource questions, unless you specify another Region. To use a different Region, change the console Region or specify a Region in your prompt. The console Region doesn't determine where the AWS assistant processes your requests. For more information, see Cross-Region processing in the AWS assistant.

Use visualizations

With the AWS assistant, you can get responses that include visualizations, such as tables, charts, dashboards, time series graphs, and key-value pairs. When you ask a natural language question about resources or costs, the AWS assistant determines whether a visual response would be useful. It opens the visual response in a panel next to the conversation.

Resource tables can group resources by requested properties and include deep links to their pages in the applicable service console. Cost responses can include chart widgets for historical costs, forecasts, pricing, budgets, coverage, and utilization. You can paginate, filter, and sort results directly within the response.

> **Note**
>
> All data associated with AWS assistant visualizations is stored in US East (N. Virginia).

Example visualization prompts include:

- List my running EC2 instances.
- Create a chart of my costs by Region last month.
- Show me a bar chart of potential savings by optimization recommendation.
- Check traffic and internet accessibility to EC2 resources.

Use the prompt library

The AWS assistant recommends prompts when you start a conversation. To browse supported prompts, choose the book icon in the chat panel header. You can filter prompts by table or visualization response type.

Other example prompts include:

- List RDS databases without Amazon CloudWatch alarms.
- When should I put my resources in a VPC?
- Which container service should I use for a workload when I need to keep costs low?
- Create a short AWS Command Line Interface (AWS CLI) script to list my AWS Lambda functions.

Chatting about your AWS resources

With the AWS assistant, you can get answers about the resources in your AWS account. You can understand your infrastructure without navigating multiple service consoles, APIs, or scripts.

You can chat about your resources in the AWS Management Console and AWS Console Mobile Application.

The AWS assistant supports:

Resource listing and details

Request lists or configuration details for resources in your account.

Filtered queries

Filter by criteria such as AWS Region, resource category, resource ID, or tag.

Cross-service analysis

Ask about relationships, configurations, and dependencies across multiple services.

Troubleshooting

Identify potential causes and get tailored remediation steps and recommendations.

How resource analysis works

The AWS assistant uses AWS APIs to retrieve requested information. It can perform read-only get, list, and describe operations for multiple resources. For complex requests, it creates a sequence of read-only steps and explains them. It can request more information when needed.

The AWS assistant can display read-only tables and charts to summarize resources and costs. It cannot inspect data stored inside resources, such as the objects in an Amazon Simple Storage Service (Amazon S3) bucket. It also can't answer questions about your account security, identity, credentials, or cryptography.

Required permissions

In addition to the permissions in Step 2: Verify required permissions, resource questions require permission for the AWS APIs that retrieve information about the resources in your question. You also need to allow the AWS assistant to retrieve resource information on your behalf with the `q:PassRequest` permission. For more information, see Using `q:PassRequest`. For an example policy, see Allow users to chat about resources.

Normal AWS service charges apply when the AWS assistant performs read, list, or describe operations.

Ask resource questions

You can specify the Region in your prompt. If you don't specify one, the AWS assistant uses a Region established earlier in the conversation. Otherwise, it uses the current console Region. If you're on the console page of a global service, it uses the most recent console Region that you selected. It might ask a follow-up question when it needs a resource name, ID, Region, or time range.

Example prompts include:

- Describe the encryption settings for S3 bucket *`bucket-name`*.

- List the SQS queues that invoke my AWS Lambda functions.
- Find the MySQL RDS clusters that need updates.
- List my EC2 instances in *Region*.
- List the alarms configured for instance *instance-id*.
- List RDS databases without Amazon CloudWatch alarms.
- List S3 buckets with tag value *tag-value*.
- Show a chart of my costs by service last week.
- Show a bar chart of my ten most expensive resources.
- Create a chart showing budget compared with forecasted spend.

Troubleshoot resources

In the AWS Management Console, open the AWS assistant and describe the problem. It analyzes the available information and identifies potential root causes. It then provides solutions, step-by-step instructions, or recommendations.

You can troubleshoot resources only in the AWS Management Console. Enter your prompt in English.

The following table lists the supported services, issues, and example prompts.

AWS service	Supported issues	Example prompt
Amazon Simple Storage Service (Amazon S3)	Permissions	Why can't I put objects into my S3 bucket amzn-s3-demo-bucket?
AWS Glue	Job failures	Why did my AWS Glue job fail?
Amazon Athena	Query issues	Why didn't my Amazon Athena query return results?
Amazon Elastic Container Service (Amazon ECS)	Stopped tasks, AWS Fargate health checks, and disconnected agents	My ECS task stopped and I don't know why.

AWS service	Supported issues	Example prompt
Elastic Load Balancing	Load balancer health checks and HTTP 500, 502, 503, and 504 errors	Why is my load balancer returning 503 errors?
Amazon Elastic Kubernetes Service (Amazon EKS)	ALB controller and managed add-on issues	Why are the add-ons in my EKS cluster failing?
Amazon Elastic Container Registry (Amazon ECR)	Cross-account repository access	How do I allow another account to push and pull images?
Amazon CloudWatch	Metric and performance monitoring	Show me the CPU utilizati on for my EC2 instance i-1234567890abcdef0 over the last 24 hours.

For resource troubleshooting, you need the same permissions as for other resource questions.

Chatting about your AWS costs

The AWS assistant helps you manage AWS costs through natural conversation. It can analyze historical and forecasted costs from AWS Cost Explorer and find recommendations from Cost Optimization Hub and AWS Compute Optimizer. It can also explain AWS Savings Plans and reservation opportunities, check invoices and balances, and answer AWS pricing questions.

You can ask a specific question or request an open-ended analysis. For example:

- What were my EC2 costs last month?
- What were the biggest drivers of last week's cost decrease?

Responses can show the APIs and parameters used and link to matching console views.

You can chat about costs in the AWS Management Console.

Cost capabilities

Cost analysis

Analyze spending patterns, trends, forecasts, and major changes.

Optimization

Find idle or over-provisioned resources and review Cost Optimization Hub, AWS Compute Optimizer, AWS Savings Plans, and reservation recommendations.

Pricing

Estimate AWS service and architecture costs.

Payments

List recent invoices and check outstanding balances.

Visualizations

Generate custom charts of costs, usage, forecasts, budgets, pricing, and optimization data.

The AWS assistant maintains context so that you can ask follow-up questions and refine an analysis.

How cost analysis works

The AWS assistant retrieves data from AWS Cost Explorer, Cost Optimization Hub, AWS Compute Optimizer, AWS Budgets, AWS Free Tier, AWS Pricing, and AWS Savings Plans and reservation recommendation APIs. It performs calculations and analyzes patterns using your actual usage and spending data.

Prerequisites

- Your IAM identity must have permission to use the AWS assistant and access the billing and cost-management data required for your question. For an example policy, see Allow the AWS assistant to access cost data.
- Enable AWS Cost Explorer in the account. For instructions on enabling it, see Enabling AWS Cost Explorer.
- Enable optional services such as Cost Optimization Hub and AWS Budgets to use their related capabilities.

Example cost questions

Analysis and forecasting

- What were my costs last month?
- Show my EC2 spending trends for the past six months.
- Why did my costs increase last week?
- Analyze last month's spending and give me the most important insights.

Optimization and monitoring

- What are my top cost optimization opportunities?
- Which EC2 instances are over-provisioned?
- Do I have idle resources?
- Which AWS Savings Plans should I purchase?
- Have any teams exceeded their budgets?
- Do I have any cost anomalies?

Pricing

- How much does a c8g.2xlarge instance cost in `us-east-1`?
- What would it cost to store 1 PB in Amazon Simple Storage Service (Amazon S3) in Dublin?
- Estimate a basic three-tier web application with Amazon Elastic Compute Cloud (Amazon EC2), Amazon API Gateway, a small SQL database, and a frontend hosted through Amazon CloudFront.

Visualizations

- Graph my support charges by month for the last 12 months.
- Show an area chart of EC2 costs by instance type by day this month.
- Create a chart of S3 storage pricing by tier in `us-east-1`.
- Graph AWS Savings Plans coverage and utilization for the last three months.
- Graph EC2 cost per vCPU hour over the last three weeks.

For more information about cost analysis, see Overview of cost management capabilities in Amazon Q Developer in the *AWS Cost Management User Guide*.

Using service-specific insights

Chatting about telemetry and operations

The AWS assistant uses Amazon CloudWatch telemetry and operational data to answer questions about resource health and alarms. It can also help you troubleshoot. It might ask you for resource names and time ranges.

You can use this capability in the AWS Management Console.

Service health

Ask whether your AWS Lambda functions, Amazon Elastic Container Service (Amazon ECS) clusters, Amazon DynamoDB tables, or other supported resources are healthy.

Alarm troubleshooting

Find alarms in the `ALARM` state. Review the telemetry that triggered them.

Application Signals

Check service-level objectives and indicators to see how healthy your application is.

Example prompts include:

- Why is my alarm named *`alarm-name`* firing?
- Is my service healthy in environment *`environment-name`*?

For information about investigations, see CloudWatch investigations in the *Amazon CloudWatch User Guide*.

Chatting about message delivery

With the AWS assistant, you can set up message sending in Amazon SES, improve delivery and engagement, and fix sending problems. To help, the AWS assistant uses information about your sending identities, configuration sets, sending patterns, and mailbox provider responses.

You can use this capability in the AWS Management Console.

You need permission to use the AWS assistant. You also need access to the Amazon SES resources in your question.

Example questions include:

- Do I need to do anything to finish setting up Amazon SES?
- Which sending identities have the highest deliverability?
- How is my deliverability for messages sent to Yahoo?
- How can I improve message delivery?
- Were there recent events when my deliverability suddenly improved or got worse?

Using plugins with the AWS assistant

The AWS assistant integrates with third party monitoring tools and security platforms so you can access your AWS application insights without leaving the AWS builder environment. In the AWS Management Console, you can chat about metrics provided by these tools to understand and address application performance, errors, or vulnerabilities.

After you configure a plugin, add the plugin alias to the beginning of your question when you chat with the AWS assistant in the AWS Management Console. The AWS assistant calls the third party provider APIs to retrieve resources and generates a response with deep links to the external resources.

When the AWS assistant calls a third party API, the API will not appear in AWS CloudTrail logs. The CloudTrail log will only show when an AWS Secrets Manager secret is accessed by the AWS assistant to retrieve credentials to connect to the third party provider.

The AWS assistant doesn't share any information with third party providers when you configure or use plugins. For more information on how the AWS assistant uses your data, see Data protection in the AWS assistant.

> **Note**
>
> Member accounts within an AWS organization don't have access to plugins that are configured in the organization's management account profile. Each member account must create their own Q Developer profile before they can configure and use plugins in their account.

Warning

Third party provider user permissions are not detected by AWS assistant plugins. When an administrator configures a plugin in an AWS account, users with plugin permissions in that account have access to any resources in the third party provider account retrievable by the plugin.
You can configure IAM policies to restrict which plugins users have access to. For more information, see Allow users to chat with plugins from one provider.

To get started, see the topic for the plugin you want to use with the AWS assistant.

Topics

- Configuring the AWS assistant CloudZero plugin
- Configuring the AWS assistant Datadog plugin
- Configuring the AWS assistant Wiz plugin

Configuring the AWS assistant CloudZero plugin

CloudZero is a cloud cost optimization platform that evaluates costs to improve cloud efficiency. If you use CloudZero to monitor your AWS costs, you can use the CloudZero plugin in the AWS assistant to access cost insights without leaving the AWS Management Console.

You can use the CloudZero plugin to understand your AWS costs, get cost optimization insights, and track billing. After you receive a response, you can ask follow up questions, such as the status or cost impact of CloudZero insights.

To configure the plugin, you provide authentication credentials from your CloudZero account to enable a connection between the AWS assistant and CloudZero. After you configure the plugin, you can access CloudZero data by adding **`@cloudzero`** to the beginning of your question in the AWS assistant.

Warning

CloudZero user permissions are not detected by the CloudZero plugin in the AWS assistant. When an administrator configures the CloudZero plugin in an AWS account, users with plugin permissions in that account have access to any resources in the CloudZero account retrievable by the plugin.

You can configure IAM policies to restrict which plugins users have access to. For more information, see Configure user permissions.

Prerequisites

Add permissions

To configure plugins, the following administrator level permissions are required:

- Permissions to access the Amazon Q Developer console. For an example IAM policy that grants needed permissions, see Allow administrators to use the Amazon Q Developer console in the *Amazon Q Developer User Guide*.
- Permissions to configure plugins. For an example IAM policy that grants the needed permissions, see Allow administrators to configure plugins.

Acquire credentials

Before you begin, note the following information from your CloudZero account. These authentication credentials will be stored in an AWS Secrets Manager secret when you configure the plugin.

- **API key** – An access key that allows the AWS assistant to call the CloudZero API to access your organization's cost insights and billing information. You can find the API key in your CloudZero account settings. For more information, see the Authorization in the CloudZero documentation.

For more information on acquiring credentials from your CloudZero account, see the CloudZero documentation.

Secrets and service roles

AWS Secrets Manager secret

When you configure the plugin, the Amazon Q Developer console creates a new AWS Secrets Manager secret for you to store CloudZero authentication credentials. Alternatively, you can use an existing secret that you create yourself.

If you create a secret yourself, enter the API key as plaintext:

```
your-api-key
```

For more information about creating secrets, see Create a secret in the *AWS Secrets Manager User Guide*.

Service roles

To configure the CloudZero plugin for the AWS assistant, you need to create a service role that gives the AWS assistant permission to access your Secrets Manager secret. The AWS assistant assumes this role to access the secret where your CloudZero credentials are stored.

When you configure the plugin in the AWS Management Console, you have the option to create a new secret or use an existing one. If you create a new secret, the associated service role is created for you. If you use an existing secret and an existing service role, make sure your service role contains the following permissions, and has the following trust policy attached. The service role required depends on your secret encryption method.

If your secret is encrypted with an AWS managed KMS key, the following IAM service role is required:

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret-id"
            ]
        }
    ]
}
```

If your secret is encrypted with a customer managed AWS KMS key, the following IAM service role is required:

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret-
id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-id",
            "Condition": {
                "StringEquals": {
                "kms:ViaService": "secretsmanager.us-east-1.amazonaws.com"
                }
            }
        }
    ]
}
```

To allow the AWS assistant to assume the service role, the service role needs the following trust policy:

> **Note**
>
> The `codewhisperer` prefix is a legacy name from a service that merged with Amazon Q Developer. For more information, see Amazon Q Developer rename - Summary of changes in the *Amazon Q Developer User Guide*.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "q.amazonaws.com"
      },
```

```
      "Action": ["sts:AssumeRole", "sts:SetContext"],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333",
          "aws:SourceArn": "arn:aws:codewhisperer:us-
east-1:111122223333:profile/profile-id"
        }
      }
    }
  ]
}
```

For more information about service roles, see Create a role to delegate permissions to an AWS service in the *AWS Identity and Access Management User Guide*.

Configure the CloudZero plugin

You configure plugins in the Amazon Q Developer console. The AWS assistant uses credentials stored in AWS Secrets Manager to enable interactions with CloudZero.

To configure the CloudZero plugin, complete the following procedure:

1. Open the Amazon Q Developer console at https://console.aws.amazon.com/amazonq/developer/home
2. On the Amazon Q Developer console home page, choose **Settings**.
3. In the navigation bar, choose **Plugins**.
4. On the plugins page, choose the plus sign on the **CloudZero** panel. The plugin configuration page opens.
5. For **Configure AWS Secrets Manager**, choose either **Create a new secret** or **Use an existing secret**. The Secrets Manager secret is where your CloudZero authentication credentials will be stored.

   If you create a new secret, enter the following information:

   a. For **CloudZero API key**, enter the API key for your CloudZero organization.

   b. A service role will be created that the AWS assistant will use to access the secret where your CloudZero credentials are stored. Do not edit the service role that is created for you.

If you use an existing secret, choose a secret from the **AWS Secrets Manager secret** dropdown menu. The secret should include the CloudZero authentication credentials specified in the previous step.

For more information about the required credentials, see Acquire credentials.

6. For **Configure AWS IAM service role**, choose either **Create new service role** or **Use existing service role**.

> **Note**
>
> If you chose **Create a new secret** for step 6, you can't use an existing service role. A new role will be created for you.

If you create a new service role, a service role will be created that the AWS assistant will use to access the secret where your CloudZero credentials are stored. Do not edit the service role that is created for you.

If you use an existing service role, choose a role from the dropdown menu that appears. Make sure your service role has the permissions and trust policy defined in Service roles.

7. Choose **Save configuration**.
8. After the CloudZero plugin panel appears in the **Configured plugins** section on the Plugins page, users will have access to the plugin.

If you want to update the credentials for a plugin, you must delete your current plugin and configure a new one. Deleting a plugin removes all previous specifications. Any time you configure a new plugin, a new plugin ARN is generated.

Configure user permissions

To use plugins, the following permissions are required:

- Permissions to chat with the AWS assistant in the console. For an example IAM policy that grants permissions needed to chat, see Allow users to chat with the AWS assistant.
- The `q:UsePlugin` permission.

When you grant an IAM identity access to a configured CloudZero plugin, the identity gains access to any resources in the CloudZero account retrievable by the plugin. CloudZero user permissions are not detected by the plugin. If you want to control access to a plugin, you can do so by specifying the plugin ARN in an IAM policy.

Each time you create or delete and re-configure a plugin, it is assigned a new ARN. If you use a plugin ARN in a policy, it will need to be updated if you want to grant access to the newly configured plugin.

To locate the CloudZero plugin ARN, go to the **Plugins** page in the Amazon Q Developer console and choose the configured CloudZero plugin. On the plugin details page, copy the plugin ARN. You can add this ARN to a policy to allow or deny access to the CloudZero plugin.

If you create a policy to control access to CloudZero plugins, specify `CloudZero` for the plugin provider in the policy.

For examples of IAM policies that control plugin access, see Allow users to chat with plugins from one provider.

Chat with the CloudZero plugin

To use the CloudZero plugin, enter **`@cloudzero`** at the beginning of a question about CloudZero or your AWS application monitors and cases. Follow up questions or responses to questions from the AWS assistant must also include **`@cloudzero`**.

Following are some example use cases and associated questions you can ask to get the most of out of the CloudZero plugin:

- **Learn about using CloudZero with AWS** – Ask about how CloudZero features work. The AWS assistant might ask you for more information about what you're trying to do to provide the best answer.
  - **`@cloudzero how do I use CloudZero?`**
  - **`@cloudzero how do I get started with CloudZero?`**
- **List cost insights** – Get a list of cost insights or find out more about a specific insight.
  - **`@cloudzero list my top cost insights`**
  - **`@cloudzero tell me more about insight <insight ID>`**
- **Get billing information** – Ask the CloudZero plugin about your AWS billing information.

- **`@cloudzero what were my AWS costs for December 2024?`**

Configuring the AWS assistant Datadog plugin

Datadog is a monitoring and security platform that provides infrastructure, application, and network monitoring and analytics. If you use Datadog to monitor your AWS applications, you can use the Datadog plugin in the AWS assistant to access monitoring information without leaving the AWS Management Console.

You can use the Datadog plugin to learn about Datadog, understand how it works with AWS services, and ask about your Datadog cases and monitors. After you receive a response, you can ask follow up questions, including how to address an issue or for details about Datadog resources.

To configure the plugin, you provide authentication credentials from your Datadog account to enable a connection between the AWS assistant and Datadog. After you configure the plugin, you can access Datadog metrics by adding **`@datadog`** to the beginning of your question in the AWS assistant.

> ⚠ **Warning**
>
> Datadog user permissions are not detected by the Datadog plugin in the AWS assistant. When an administrator configures the Datadog plugin in an AWS account, users with plugin permissions in that account have access to any resources in the Datadog account retrievable by the plugin.
> You can configure IAM policies to restrict which plugins users have access to. For more information, see Configure user permissions.

Prerequisites

Add permissions

To configure plugins, the following administrator level permissions are required:

- Permissions to access the Amazon Q Developer console. For an example IAM policy that grants needed permissions, see Allow administrators to use the Amazon Q Developer console in the *Amazon Q Developer User Guide*.

- Permissions to configure plugins. For an example IAM policy that grants the needed permissions, see Allow administrators to configure plugins.

Acquire credentials

Before you begin, note the following information from your Datadog account. These authentication credentials will be stored in an AWS Secrets Manager secret when you configure the plugin.

- **Site parameter** – The Datadog site parameter you use. For example, `us3.datadoghq.com`. For more information, see Getting Started with Datadog Sites in the Datadog documentation.
- **API key and application key** – Access keys that allow the AWS assistant to call the Datadog API to access events and metrics. You can find these under **Organization Settings** in your Datadog account. For more information, see API and Application Keys in the Datadog documentation.

Secrets and service roles

AWS Secrets Manager secret

When you configure the plugin, the Amazon Q Developer console creates a new AWS Secrets Manager secret for you to store Datadog authentication credentials. Alternatively, you can use an existing secret that you create yourself.

If you create a secret yourself, make sure it includes the following credentials and uses the following JSON format:

```
{
   "ApiKey": "<your-api-key>",
   "AppKey": "<your-applicaiton-key>"
}
```

For more information about creating secrets, see Create a secret in the *AWS Secrets Manager User Guide*.

Service roles

To configure the Datadog plugin for the AWS assistant, you need to create a service role that gives the AWS assistant permission to access your Secrets Manager secret. The AWS assistant assumes this role to access the secret where your Datadog credentials are stored.

When you configure the plugin in the AWS Management Console, you have the option to create a new secret or use an existing one. If you create a new secret, the associated service role is created for you. If you use an existing secret and an existing service role, make sure your service role contains the following permissions, and has the following trust policy attached. The service role required depends on your secret encryption method.

If your secret is encrypted with an AWS managed KMS key, the following IAM service role is required:

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret-id"
            ]
        }
    ]
}
```

If your secret is encrypted with a customer managed AWS KMS key, the following IAM service role is required:

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret-
id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
```

```
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-id",
            "Condition": {
                "StringEquals": {
                "kms:ViaService": "secretsmanager.us-east-1.amazonaws.com"
                }
            }
        }
    ]
}
```

To allow the AWS assistant to assume the service role, the service role needs the following trust policy:

> **Note**
>
> The `codewhisperer` prefix is a legacy name from a service that merged with Amazon Q Developer. For more information, see Amazon Q Developer rename - Summary of changes in the *Amazon Q Developer User Guide*.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "q.amazonaws.com"
      },
      "Action": ["sts:AssumeRole", "sts:SetContext"],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333",
          "aws:SourceArn": "arn:aws:codewhisperer:us-
east-1:111122223333:profile/profile-id"
        }
      }
    }
  ]
}
```

For more information about service roles, see Create a role to delegate permissions to an AWS service in the *AWS Identity and Access Management User Guide*.

Configure the Datadog plugin

You configure plugins in the Amazon Q Developer console. The AWS assistant uses credentials stored in AWS Secrets Manager to enable interactions with Datadog.

To configure the Datadog plugin, complete the following procedure:

1. Open the Amazon Q Developer console at https://console.aws.amazon.com/amazonq/developer/home
2. On the Amazon Q Developer console home page, choose **Settings**.
3. In the navigation bar, choose **Plugins**.
4. On the plugins page, choose the plus sign on the **Datadog** panel. The plugin configuration page opens.
5. For **Site URL**, enter the URL of the Datadog site you use.
6. For **Configure AWS Secrets Manager**, choose either **Create a new secret** or **Use an existing secret**. The Secrets Manager secret is where your Datadog authentication credentials will be stored.

   If you create a new secret, enter the following information:

   a. For **Datadog API key**, enter the API key for your Datadog organization.

   b. For **Datadog application key**, enter the application key for your Datadog account.

   c. A service role will be created that the AWS assistant will use to access the secret where your Datadog credentials are stored. Do not edit the service role that is created for you.

   If you use an existing secret, choose a secret from the **AWS Secrets Manager secret** dropdown menu. The secret should include the Datadog authentication credentials specified in the previous step.

   For more information about the required credentials, see Acquire credentials.
7. For **Configure AWS IAM service role**, choose either **Create new service role** or **Use existing service role**.

> **Note**
>
> If you chose **Create a new secret** for step 6, you can't use an existing service role. A new role will be created for you.

If you create a new service role, a service role will be created that the AWS assistant will use to access the secret where your Datadog credentials are stored. Do not edit the service role that is created for you.

If you use an existing service role, choose a role from the dropdown menu that appears. Make sure your service role has the permissions and trust policy defined in Service roles.

8. Choose **Save configuration**.
9. After the Datadog plugin panel appears in the **Configured plugins** section on the Plugins page, users will have access to the plugin.

If you want to update the credentials for a plugin, you must delete your current plugin and configure a new one. Deleting a plugin removes all previous specifications. Any time you configure a new plugin, a new plugin ARN is generated.

Configure user permissions

To use plugins, the following permissions are required:

- Permissions to chat with the AWS assistant in the console. For an example IAM policy that grants permissions needed to chat, see Allow users to chat with the AWS assistant.
- The `q:UsePlugin` permission.

When you grant an IAM identity access to a configured Datadog plugin, the identity gains access to any resources in the Datadog account retrievable by the plugin. Datadog user permissions are not detected by the plugin. If you want to control access to a plugin, you can do so by specifying the plugin ARN in an IAM policy.

Each time you create or delete and re-configure a plugin, it is assigned a new ARN. If you use a plugin ARN in a policy, it will need to be updated if you want to grant access to the newly configured plugin.

To locate the Datadog plugin ARN, go to the **Plugins** page in the Amazon Q Developer console and choose the configured Datadog plugin. On the plugin details page, copy the plugin ARN. You can add this ARN to a policy to allow or deny access to the Datadog plugin.

If you create a policy to control access to Datadog plugins, specify `Datadog` for the plugin provider in the policy.

For examples of IAM policies that control plugin access, see Allow users to chat with plugins from one provider.

Chat with the Datadog plugin

To use the Datadog plugin, enter **`@datadog`** at the beginning of a question about Datadog or your AWS application monitors and cases. Follow up questions or responses to questions from the AWS assistant must also include **`@datadog`**.

Following are some example use cases and associated questions you can ask to get the most of out of the Datadog plugin:

- **Learn about using Datadog features in your AWS workload** – Ask about how Datadog features work with certain AWS services. The AWS assistant might ask you for more information about what you're trying to do to provide the best answer.
  - **`@datadog how do I use APM on EC2?`**
- **Retrieve and summarize cases and monitors** – Ask about a specific case or monitor, or specify properties to get information about monitors and cases like create date, status, or author. For more information about properties, see Properties in the Datadog documentation.
  - **`@datadog summarize the global outage case`**
  - **`@datadog summarize my top cases`**
- **Check monitors that are in an alarm state** – Ask the Datadog plugin to find your AWS application monitors that are in alarm. You can follow up with questions about the monitors it lists.
  - **`@datadog what monitors are in alarm?`**
  - **`@datadog what is the status for monitor <monitor ID>?`**

Configuring the AWS assistant Wiz plugin

Wiz is a cloud security platform that provides security posture management, risk assessment and prioritization, and vulnerability management. If you use Wiz to evaluate and monitor your AWS applications, you can use the plugin in the AWS assistant to access insights from Wiz without leaving the AWS Management Console.

You can use the plugin to identify and retrieve Wiz issues, assess your riskiest assets, and understand vulnerabilities or exposures. After you receive a response, you can ask follow up questions, including how to remediate an issue.

To configure the plugin, you provide authentication credentials from your Wiz account to enable a connection between the AWS assistant and Wiz. After you configure the plugin, you can access Wiz metrics by adding **`@wiz`** to the beginning of your question in the AWS assistant.

> **Warning**
>
> Wiz user permissions are not detected by the Wiz plugin in the AWS assistant. When an administrator configures the Wiz plugin in an AWS account, users with plugin permissions in that account have access to any resources in the Wiz account retrievable by the plugin. You can configure IAM policies to restrict which plugins users have access to. For more information, see Configure user permissions.

Prerequisites

Add permissions

To configure plugins, the following administrator level permissions are required:

- Permissions to access the Amazon Q Developer console. For an example IAM policy that grants needed permissions, see Allow administrators to use the Amazon Q Developer console in the *Amazon Q Developer User Guide*.
- Permissions to configure plugins. For an example IAM policy that grants the needed permissions, see Allow administrators to configure plugins.

Acquire credentials

Before you begin, note the following information from your Wiz account. These authentication credentials will be stored in an AWS Secrets Manager secret when you configure the plugin.

- **API endpoint URL** – The URL where you access Wiz. For example, `https://api.us1.app.wiz.io/graphql`. For more information, see API endpoint URL in the Wiz documentation.
- **Client ID and Client secret** – Credentials that allow the AWS assistant to call Wiz APIs to access your application. For more information, see Client ID and Client secret in the Wiz documentation.

Secrets and service roles

AWS Secrets Manager secret

When you configure the plugin, the Amazon Q Developer console creates a new AWS Secrets Manager secret for you to store Wiz authentication credentials. Alternatively, you can use an existing secret that you create yourself.

If you create a secret yourself, make sure it includes the following credentials and uses the following JSON format:

```
{
   "ClientId": "<your-client-id>",
   "ClientSecret": "<your-client-secret>"
}
```

For more information about creating secrets, see Create a secret in the *AWS Secrets Manager User Guide*.

Service roles

To configure the Wiz plugin for the AWS assistant, you need to create a service role that gives the AWS assistant permission to access your Secrets Manager secret. The AWS assistant assumes this role to access the secret where your Wiz credentials are stored.

When you configure the plugin in the AWS Management Console, you have the option to create a new secret or use an existing one. If you create a new secret, the associated service role is created for you. If you use an existing secret and an existing service role, make sure your service role

contains these permissions, and has the following trust policy attached. The service role required depends on your secret encryption method.

If your secret is encrypted with an AWS managed KMS key, the following IAM service role is required:

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret-id"
            ]
        }
    ]
}
```

If your secret is encrypted with a customer managed AWS KMS key, the following IAM service role is required:

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret-
id"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-id",
            "Condition": {
```

```
                "StringEquals": {
                "kms:ViaService": "secretsmanager.us-east-1.amazonaws.com"
                }
            }
        }
    ]
}
```

To allow the AWS assistant to assume the service role, the service role needs the following trust policy:

> **Note**
>
> The `codewhisperer` prefix is a legacy name from a service that merged with Amazon Q Developer. For more information, see Amazon Q Developer rename - Summary of changes in the *Amazon Q Developer User Guide*.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "q.amazonaws.com"
      },
      "Action": ["sts:AssumeRole", "sts:SetContext"],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333",
          "aws:SourceArn": "arn:aws:codewhisperer:us-
east-1:111122223333:profile/profile-id"
        }
      }
    }
  ]
}
```

For more information about service roles, see Create a role to delegate permissions to an AWS service in the *AWS Identity and Access Management User Guide*.

Configure the Wiz plugin

You configure plugins in the Amazon Q Developer console. The AWS assistant uses credentials stored in AWS Secrets Manager to enable interactions with Wiz.

To configure the Wiz plugin, complete the following procedure:

1. Open the Amazon Q Developer console at https://console.aws.amazon.com/amazonq/developer/home
2. On the Amazon Q Developer console home page, choose **Settings**.
3. In the navigation bar, choose **Plugins**.
4. On the plugins page, choose the plus sign on the **Wiz** panel. The plugin configuration page opens.
5. For **API endpoint URL**, enter the URL of API endpoint where you access Wiz.
6. For **Configure AWS Secrets Manager**, choose either **Create a new secret** or **Use an existing secret**. The Secrets Manager secret is where your Wiz authentication credentials will be stored.

   If you create a new secret, enter the following information:

   a. For **Client ID**, enter the Client ID for your Wiz account.

   b. For **Client Secret**, enter the Client Secret for your Wiz account.

   c. A service role will be created that the AWS assistant will use to access the secret where your Wiz credentials are stored. Do not edit the service role that is created for you.

   If you use an existing secret, choose a secret from the **AWS Secrets Manager secret** dropdown menu. The secret should include the Wiz authentication credentials specified in the previous step.

   For more information about the required credentials, see Acquire credentials.
7. For **Configure AWS IAM service role**, choose either **Create new service role** or **Use existing service role**.

   > **Note**
   >
   > If you chose **Create a new secret** for step 6, you can't use an existing service role. A new role will be created for you.

If you create a new service role, a service role will be created that the AWS assistant will use to access the secret where your Wiz credentials are stored. Do not edit the service role that is created for you.

If you use an existing service role, choose a role from the dropdown menu that appears. Make sure your service role has the permissions and trust policy defined in Service roles.

8. Choose **Save configuration**.
9. After the Wiz plugin panel appears in the **Configured plugins** section on the Plugins page, users will have access to the plugin.

If you want to update the credentials for a plugin, you must delete your current plugin and configure a new one. Deleting a plugin removes all previous specifications. Any time you configure a new plugin, a new plugin ARN is generated.

Configure user permissions

To use plugins, the following permissions are required:

- Permissions to chat with the AWS assistant in the console. For an example IAM policy that grants permissions needed to chat, see Allow users to chat with the AWS assistant.
- The `q:UsePlugin` permission.

When you grant an IAM identity access to a configured Wiz plugin, the identity gains access to any resources in the Wiz account retrievable by the plugin. Wiz user permissions are not detected by the plugin. If you want to control access to a plugin, you can do so by specifying the plugin ARN in an IAM policy.

Each time you create or delete and re-configure a plugin, it is assigned a new ARN. If you use a plugin ARN in a policy, it will need to be updated if you want to grant access to the newly configured plugin.

To locate the Wiz plugin ARN, go to the **Plugins** page in the Amazon Q Developer console and choose the configured Wiz plugin. On the plugin details page, copy the plugin ARN. You can add this ARN to a policy to allow or deny access to the Wiz plugin.

If you create a policy to control access to Wiz plugins, specify `Wiz` for the plugin provider in the policy.

For examples of IAM policies that control plugin access, see Allow users to chat with plugins from one provider.

Chat with the Wiz plugin

To use the Wiz plugin, enter **`@wiz`** at the beginning of a question about your Wiz issues. Follow up questions or responses to questions from the AWS assistant must also include **`@wiz`**.

Following are some example use cases and associated questions you can ask to get the most of out of the Wiz plugin:

- **View issues with critical severity** – Ask the Wiz plugin to list your issues with critical or high severity. The plugin can return up to 10 issues. You can also ask to list up to the top 10 most severe issues.
  - **`@wiz what are my critical severity issues?`**
  - **`@wiz can you specify the top 5?`**
- **List issues based on date or status** – Ask to list issues based on create date, due date, or resolved date. You can also specify issues based on properties like status, severity, and type.
  - **`@wiz which issues are due before <date>?`**
  - **`@wiz what are my issues that have been resolved since <date>?`**
- **Assess issues with security vulnerabilities** – Ask about the vulnerabilities or exposures that are posing security threats in your issues.
  - **`@wiz which issues are associated with vulnerabilities or external exposures?`**

Diagnosing console errors with the AWS assistant

With the AWS assistant, you can diagnose common console errors. Examples include missing permissions, incorrect settings, and exceeded service quotas. It supports errors in the following services:

- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon Elastic Container Service (Amazon ECS)
- Amazon Simple Storage Service (Amazon S3)
- AWS Lambda
- AWS Step Functions

It also diagnoses IAM permission errors on all console pages and some service-specific errors in other services.

This capability is available only in the AWS Management Console. The AWS assistant doesn't keep a history of earlier diagnosis sessions.

Required permissions

Your IAM identity needs permission to diagnose console errors. For an example policy, see Allow users to diagnose console errors.

Diagnosing a console error

To diagnose a console error

1. When a supported error occurs, choose **Diagnose with AWS assistant** in the error message.
2. Review the explanation and the steps to fix the error. The steps can take several seconds to appear.
3. To give feedback, choose the thumbs-up or thumbs-down icon. To add details, choose **Tell me more**.

If the diagnosis doesn't fix the problem, you can ask the AWS assistant to create an AWS Support case. To report a problem with the diagnosis itself, choose the thumbs-down icon. For information about support cases, see Contacting AWS Support with the AWS assistant.

Contacting AWS Support with the AWS assistant

You can create an AWS Support case from anywhere in the AWS Management Console, including AWS Support Center. The AWS assistant uses the context of your conversation to draft the case and includes the recent conversation in the description. After you submit the case, you can transfer to a support agent using an available contact method, including live chat in the same panel.

You can create and transfer an AWS Support case through the AWS assistant only in the AWS Management Console. You can't use this feature at the Pro tier if your Amazon Q Developer profile is in the Europe (Frankfurt) Region.

Cases created through the AWS assistant also appear in AWS Support Center. Use AWS Support Center to track later updates.

All AWS accounts include account and billing support. Technical support through the AWS assistant requires a support offering above the Basic tier.

> **Tip**
>
> Before creating a case, ask the AWS assistant to troubleshoot the problem or choose **Diagnose with AWS assistant** in the error message when it is available.

Prerequisites

- For technical support, the account has a support offering above the Basic tier.
- You have permission to use the AWS assistant.
- You have permission to create AWS Support cases. For more information, see Managing access to AWS Support Center.

Specify the right service

The AWS assistant selects a service based on your question. Correct the field before submission if it chooses the wrong service. For an issue with an assistant capability owned by another AWS service, select that service. For example, select Amazon Virtual Private Cloud (Amazon VPC) for an issue with network troubleshooting in Amazon VPC Reachability Analyzer. For an issue with the AWS assistant itself, create the case for the AWS assistant.

Create a support case

To create a support case

1. In the AWS Management Console, open the AWS assistant chat panel.
2. Enter a request that includes relevant details, such as `I want to speak to someone` or `Get support`.
3. Review the drafted case. To change a field or attach a file, choose **Edit**. If the AWS assistant can't provide a value for a field, enter it manually.
4. Choose **Submit**.
5. Choose an available contact method, such as **Chat**, **Email**, or **Call**.
6. After the interaction, provide feedback or choose **Skip** to return to the AWS assistant.

To stop without creating the case, choose **Cancel** before you submit it.

You can also start a case from a response that didn't help. Choose the thumbs-down icon, select a reason, and choose **Create a support case**. Then continue from the review step.

For details about each contact method, see the following list:

Chat

A support agent joins the conversation. To end the chat, choose **End this chat**. Keep the panel open during the chat. The chat ends if you refresh the page, navigate to a different console, or your console session expires. If you minimize the panel or navigate away from the page, you might miss notifications or be disconnected for inactivity.

Email

An agent sends a message to the address associated with your AWS account.

Call

Enter a phone number and submit it to join the call queue.

Security in the AWS assistant

Security best practices

Detective

- Review the source links and API details in responses before you act on the information.
- Review AWS CloudTrail events for the compatibility-sensitive q service prefix and for downstream service operations that the AWS assistant performs on your behalf.
- Review access-denied responses and the applicable identity, session, permissions boundary, and organization policies.

Preventative

- Give users only the permissions that they need to access the AWS assistant and the downstream services and resources in their requests.
- Use a customer managed AWS Key Management Service (AWS KMS) key when that option is available and required by your organization.
- Don't include confidential or sensitive information in resource names, tags, or other free-form fields that can appear in a request or response.

Identity and access management

Use IAM policies to control who can use the AWS assistant and which capabilities are available. The existing q service prefix, IAM actions, managed policies, service principals, ARNs, and AWS CloudTrail event names are compatibility-sensitive technical identifiers. The product name is now AWS assistant, but these identifiers are unchanged.

Common access patterns include policies that:

- Permit users to chat with the AWS assistant in the AWS Management Console.
- Allow access through an Amazon Q Developer Pro subscription.
- Allow the AWS assistant to access a customer managed AWS KMS key.
- Permit users to diagnose supported console errors.

- Allow resource retrieval through read, list, and describe operations.
- Allow access to cost and optimization data from AWS Cost Explorer, Cost Optimization Hub, AWS Compute Optimizer, AWS Budgets, and AWS Free Tier.
- Allow access to AWS Pricing, AWS Savings Plans, and reservation recommendation APIs.
- Allow or deny selected actions that the AWS assistant calls on your behalf, including restrictions based on `aws:CalledVia` or `aws:RequestedRegion`.

The AWS assistant can access only the resources and perform only the operations that your identity is authorized to access or perform.

For policies that implement these patterns, see Identity-based policy examples for the AWS assistant.

Logging and incident response

AWS CloudTrail records the AWS assistant API calls as events under the compatibility-sensitive q service prefix. Use the information that CloudTrail collects to find the request that was made, the IP address that it came from, who made it, and when. If you don't configure a trail, you can still view recent events in **Event history** in the CloudTrail console. For more information, see the AWS CloudTrail User Guide.

The AWS assistant also makes API calls with a `dryRun` parameter to verify that you have the necessary permissions for an action, without making the request. These calls are recorded in a CloudTrail log with `"dryRun" : true` in the `requestParameters` field.

If you suspect a security issue, contact AWS Support. For information about reporting a vulnerability, see Vulnerability reporting.

Compliance validation

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

The AWS global infrastructure is built around AWS Regions and Availability Zones. Regions provide multiple physically separated and isolated Availability Zones. These Availability Zones are connected with low-latency, high-throughput, and highly redundant networking. The AWS assistant also uses cross-Region inference to distribute traffic within a geography during periods of high demand. For more information, see Cross-Region inference.

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

Don't make operational recovery depend only on an AWS assistant response. Verify source links and use the applicable service console, API, or support process if the AWS assistant is unavailable.

Infrastructure security

As a managed service, the AWS assistant is protected by the AWS global network security procedures described in Security and compliance. Connections to the AWS assistant use TLS 1.2 or later.

Chatting with the AWS assistant in the AWS Management Console isn't supported through interface VPC endpoints (AWS PrivateLink).

For services that the AWS assistant calls on your behalf, use the network, endpoint, and access controls that each service documents.

Configuration and vulnerability analysis

Configuration and IT controls are a shared responsibility between AWS and you, our customer. AWS manages the underlying infrastructure for the AWS assistant. You are responsible for configuring IAM permissions, customer managed keys, and the services that the AWS assistant calls on your behalf. For more information, see the AWS shared responsibility model.

Review AWS security bulletins and the security guidance for each downstream service that you use with the AWS assistant.

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action.

The AWS assistant calls other services with the permissions of your IAM identity. It can't perform an action that you aren't allowed to perform. When the AWS assistant calls a service on your behalf, the request includes `q.amazonaws.com` in the `aws:CalledVia` condition key. You can use this key to restrict the actions that the AWS assistant performs. For examples, see Deny the AWS assistant permission to perform specific actions and Allow the AWS assistant permission to perform specific actions.

For general guidance, see The confused deputy problem in the *IAM User Guide*.

Data protection in the AWS assistant

Don't put confidential or sensitive information in tags, resource names, or other free-form fields. Questions and responses can be included in service and diagnostic data as described in this chapter.

Encryption at rest

Data at rest is stored using Amazon DynamoDB and Amazon Simple Storage Service (Amazon S3). It is encrypted by default with AWS owned AWS KMS keys.

You can use a customer managed key only for console chat and console error diagnosis.

Encryption in transit

Communication between customers, the AWS assistant, and its downstream dependencies uses TLS 1.2 or later.

Key management

If you're an administrator, you can configure a symmetric customer managed AWS KMS key for workforce users with a Pro subscription. The key applies to console chat and console error diagnosis. To use the key, enter it in the Amazon Q Developer console. You need AWS

KMS permissions to set up the key. The required permissions are in the example policy Allow administrators to use the Amazon Q Developer console in the *Amazon Q Developer User Guide*. For instructions on entering the key, see Managing the encryption method in Amazon Q Developer in the *Amazon Q Developer User Guide*.

When you configure the key, the AWS assistant uses an AWS KMS grant to permit authorized use. Make sure that your users have permission for the AWS assistant to use the key. For an example policy, see Allow the AWS assistant to access customer managed keys. If you see an error about AWS KMS grants, see Error related to AWS KMS grants.

If you change the AWS KMS key for console chat, start a new conversation. Conversations encrypted with the previous key aren't available unless the previous key is restored. If you change the key for diagnosis, start a new diagnosis session.

Data storage in the AWS assistant

The AWS assistant stores your questions, its responses, and additional context, such as console metadata, to generate responses to your questions and requests. For information about how data is encrypted, see Data protection in the AWS assistant. For information about how AWS may use some questions that you ask the AWS assistant and its responses to improve our services, see Service improvement for the AWS assistant.

AWS Regions where content is processed and stored

If you're an AWS IAM Identity Center workforce user, at the Amazon Q Developer Pro tier, your content is stored in the AWS Region where your Amazon Q Developer profile was created only for the following features:

- Chatting with the AWS assistant in the AWS Management Console
- Diagnosing console errors with the AWS assistant

When you use any other feature at the Amazon Q Developer Pro tier, your content may be stored and processed in a US Region. If you are using a Q Developer profile in a non-US Region, you can create a service control policy (SCP) to block access to features that store content and perform inference in the US. For an example SCP, see Example SCP: Deny access outside the Europe (Frankfurt) Region.

For other AWS assistant features, and when using the AWS assistant Free tier, your content is stored in a Region in the US. Data processed during diagnosing console error sessions is stored in

US West (Oregon). All other data is stored in US East (N. Virginia). Note the following features that store data differently.

> **Note**
>
> When you use visualizations, your visualizations-related content is stored in a US Region. When you create an investigation with Amazon CloudWatch investigations, your content may be stored and processed in other Regions. For more information, see Security in CloudWatch investigations in the *Amazon CloudWatch User Guide*.

With cross-Region inference, your requests to the AWS assistant may be processed in a different Region within the geography where your content is stored. For more information, see Cross-Region inference.

Service improvement for the AWS assistant

To help the AWS assistant provide the most relevant information, we may use certain content from the AWS assistant, such as questions that you ask the AWS assistant and its responses, for service improvement. This page explains what content we use and how to opt out.

Free tier content used for service improvement

We may use certain content from the AWS assistant Free tier for service improvement. The AWS assistant may use this content, for example, to provide better responses to common questions, fix AWS assistant operational issues, for debugging, or for model training.

Content that AWS may use for service improvement includes, for example, your questions to the AWS assistant and the responses that the AWS assistant generates.

We do not use content from Amazon Q Developer Pro for service improvement.

How to opt out

To opt out of the AWS assistant Free tier using content for service improvement, configure an AI services opt-out policy in AWS Organizations. The AWS assistant uses the Amazon Q Developer opt-out setting (the q service name key in the policy). If you have already configured an opt-out policy for Amazon Q Developer (the q service name key), that policy also applies to the AWS assistant. You can also opt out of all AI services. The policy applies to the AWS Management

Console and AWS Console Mobile Application. For more information, see AI services opt-out policies in the *AWS Organizations User Guide*.

Cross-Region processing in the AWS assistant

Request-processing Regions

The AWS assistant processes your requests in the following Regions:

- At the Free tier, the AWS assistant processes requests in US East (N. Virginia).
- At the Pro tier, the AWS assistant processes requests in US East (N. Virginia) by default. You can choose to process requests in Europe (Frankfurt) instead.

The Region selected in the AWS Management Console doesn't determine where the AWS assistant processes your requests.

Cross-Region inference

The AWS assistant is powered by Amazon Bedrock. It uses cross-Region inference to distribute traffic across AWS Regions. This improves large language model (LLM) performance and reliability. With cross-Region inference, you get:

- More throughput and resilience during periods of high demand
- Better performance
- Access to new capabilities that rely on the most capable LLMs hosted on Amazon Bedrock

Cross-Region inference keeps your requests within the Regions in the geography where your data resides. For example, a request from an Amazon Q Developer profile that you created in the United States stays within the United States Regions listed in Inference Regions by geography.

Some features and integrations might perform inference in Regions other than the Region where you created your profile.

Cross-Region inference doesn't change where your data is stored. However, your requests and results might move outside the Region where your data resides. The AWS assistant encrypts all data while it moves across the Amazon network. Cross-Region inference has no additional cost.

For information about where data is stored, see Data storage in the AWS assistant.

Inference Regions by geography

The following list shows the Regions where the AWS assistant might route your requests, based on the geography where the request originated.

Europe

Europe (Frankfurt) (`eu-central-1`), Europe (Stockholm) (`eu-north-1`), Europe (Ireland) (`eu-west-1`), and Europe (Paris) (`eu-west-3`)

United States

US East (N. Virginia) (`us-east-1`), US East (Ohio) (`us-east-2`), and US West (Oregon) (`us-west-2`)

For the Regions where you can use the AWS assistant, see Supported AWS Regions for the AWS assistant.

Cross-Region calls

Some requests require the AWS assistant to make API calls from one Region to another. For example, a question about resources in several Regions requires calls to each of those Regions. A response that needs information from a global AWS service endpoint can also result in calls outside the Region where your data is stored.

You can use IAM policies to deny actions that the AWS assistant performs on your behalf, including cross-Region calls. This also turns off capabilities that require calls on your behalf in the current Region. To restrict calls to specific Regions, use the `aws:RequestedRegion` global condition key. To restrict the actions that the AWS assistant calls, use the `aws:CalledVia` global condition key. For example policies, see Identity-based policy examples for the AWS assistant.

Permissions reference for the AWS assistant

You can use the following permissions as a reference when you write identity-based policies for the AWS assistant. The permissions use the q service prefix.

> **Important**
>
> To chat with the AWS assistant, an IAM identity needs permissions for the following actions:

- `q:StartConversation`
- `q:SendMessage`
- `q:GetConversation` (AWS Management Console only)
- `q:ListConversations` (AWS Management Console only)

If one of these actions isn't explicitly allowed by an attached policy, an IAM permissions error is returned when you try to chat with the AWS assistant.

AWS assistant permissions

Name	Description of permission granted
User permissions	
`q:StartConversation`	Start a conversation with the AWS assistant
`q:SendMessage`	Send a message to the AWS assistant
`q:GetConversation`	Get individual messages associated with a specific conversation with the AWS assistant (AWS Management Console only)
`q:ListConversations`	List individual conversations associated with a specific AWS assistant user (AWS Management Console only)
`q:UpdateConversation`	Update a conversation with the AWS assistant
`q:DeleteConversation`	Delete a conversation with the AWS assistant
`q:GetIdentityMetaData`	Allow the AWS assistant to fetch application identity-related metadata
`q:PassRequest`	Allow the AWS assistant to perform actions that an IAM identity has permission to perform
`q:StartTroubleshooting Analysis`	Start a troubleshooting analysis with the AWS assistant

Name	Description of permission granted
q:GetTroubleshootingResults	Get troubleshooting results with the AWS assistant
q:StartTroubleshootingResolutionExplanation	Start a troubleshooting resolution explanation with the AWS assistant
q:UpdateTroubleshootingCommandResult	Allow the AWS assistant to analyze resources to troubleshoot a console error
q:UsePlugin	Access plugins from the AWS assistant
Administrator permissions for plugins	
q:CreatePlugin	Create and configure a third-party plugin for the AWS assistant
q:GetPlugin	View information about a specific plugin
q:DeletePlugin	Delete a configured plugin
q:ListPlugins	View configured plugins
q:ListPluginProviders	View available plugins
q:UpdatePlugin	Update feature enablement controls for a third-party integration plugin
q:CreateAuthGrant	Register a third-party application user
q:UpdateAuthGrant	Update a third-party application user
q:CreateOAuthAppConnection	Register a third-party OAuth application
q:UpdateOAuthAppConnection	Update a third-party OAuth application

Name	Description of permission granted
q:SendEvent	Send events from a third-party application for processing

For example policies, see Identity-based policy examples for the AWS assistant. For all actions that use the q service prefix, see Actions, resources, and condition keys for Amazon Q in the *Service Authorization Reference*.

Using q:PassRequest

q:PassRequest is a permission that allows the AWS assistant to call AWS APIs on your behalf. When you add the q:PassRequest permission to an IAM identity, the AWS assistant gains permission to call any API that the IAM identity has permission to call. For example, if an IAM role has the s3:ListAllMyBuckets permission and the q:PassRequest permission, the AWS assistant can call the ListAllMyBuckets API when a user who assumes that role asks the AWS assistant to list their Amazon S3 buckets.

You can create IAM policies that restrict the scope of the q:PassRequest permission. For example, you can prevent the AWS assistant from performing a specific action, or permit it to perform only a subset of actions for a service. You can also specify which Regions the AWS assistant can make calls to when it performs actions on your behalf.

For example policies that control the use of q:PassRequest, see the following:

- Allow the AWS assistant to perform actions on your behalf
- Deny the AWS assistant permission to perform specific actions
- Allow the AWS assistant permission to perform specific actions
- Allow the AWS assistant to perform actions in specific Regions
- Deny the AWS assistant permission to perform actions on your behalf

Identity-based policy examples for the AWS assistant

> **The AWS assistant uses the q service prefix**
>
> Technical names didn't change when Amazon Q Developer in the AWS Management Console became the AWS assistant. The following example policies use the q service prefix in IAM actions, such as `q:SendMessage`.

The following example policies control access to the AWS assistant in the AWS Management Console and AWS Console Mobile Application.

For the `AmazonQDeveloperAccess` and `AmazonQFullAccess` AWS managed policies, see AWS managed policies for Amazon Q Developer in the *Amazon Q Developer User Guide*.

User permissions

Attach one or more of the following policies to the IAM identities that use the AWS assistant.

Allow users to access the AWS assistant with a Pro subscription

The following example policy grants permission to use the AWS assistant with a Amazon Q Developer Pro subscription. Without these permissions, you can access only the Free tier. To chat or use other capabilities, you also need permissions such as those in the following example policies.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowGetIdentity",
            "Effect": "Allow",
            "Action": [
                "q:GetIdentityMetaData"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowSetTrustedIdentity",
            "Effect": "Allow",
            "Action": [
                "sts:SetContext"
```

```
            ],
            "Resource": "arn:aws:sts::*:self"
        }
    ]
}
```

Allow the AWS assistant to access customer managed keys

The following example policy allows the AWS assistant to use a customer managed key. Users need this policy if an administrator has configured a customer managed key for encryption.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QKMSDecryptGenerateDataKeyPermissions",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo"
            ],
            "Resource": [
            "arn:aws:kms:us-east-1:111122223333:key/key_id"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                    "q.us-east-1.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Allow users to chat with the AWS assistant

The following example policy grants permission to chat with the AWS assistant in the AWS Management Console.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAmazonQConversationAccess",
      "Effect": "Allow",
      "Action": [
        "q:StartConversation",
        "q:SendMessage",
        "q:GetConversation",
        "q:ListConversations",
        "q:UpdateConversation",
        "q:DeleteConversation"
      ],
      "Resource": "*"
    }
  ]
}
```

Allow users to diagnose console errors

The following example policy grants permission to diagnose console errors with the AWS assistant.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAmazonQTroubleshooting",
      "Effect": "Allow",
      "Action": [
        "q:StartTroubleshootingAnalysis",
        "q:GetTroubleshootingResults",
        "q:StartTroubleshootingResolutionExplanation",
        "q:UpdateTroubleshootingCommandResult",
        "q:PassRequest"
      ],
      "Resource": "*"
    }
  ]
}
```

Allow users to chat about resources

The following example policy grants permission to chat about resources. It also allows the AWS assistant to retrieve resource information on your behalf. The AWS assistant can access only resources that your IAM identity can access.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAmazonQPassRequest",
      "Effect": "Allow",
      "Action": [
        "q:StartConversation",
        "q:SendMessage",
        "q:GetConversation",
        "q:ListConversations",
        "q:UpdateConversation",
        "q:DeleteConversation",
        "q:PassRequest"
      ],
      "Resource": "*"
    }
  ]
}
```

Allow the AWS assistant to perform actions on your behalf

The following example policy grants permission to chat with the AWS assistant. It also allows the AWS assistant to perform actions on your behalf. The AWS assistant can perform only actions that your IAM identity has permission to perform.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAmazonQPassRequest",
      "Effect": "Allow",
      "Action": [
        "q:StartConversation",
        "q:SendMessage",
        "q:GetConversation",
```

```
        "q:ListConversations",
        "q:UpdateConversation",
        "q:DeleteConversation",
        "q:PassRequest"
      ],
      "Resource": "*"
    }
  ]
}
```

Allow the AWS assistant to access cost data

The following example policy grants permission to chat about costs. It also allows the AWS assistant to access cost data and provide cost analysis and optimization recommendations. The policy includes permissions for AWS Cost Explorer, Cost Optimization Hub, AWS Compute Optimizer, AWS Budgets, AWS Free Tier, AWS Pricing, and AWS Savings Plans and reservation recommendations.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAmazonQChatAndPassRequest",
      "Effect": "Allow",
      "Action": [
        "q:StartConversation",
        "q:SendMessage",
        "q:GetConversation",
        "q:ListConversations",
        "q:UpdateConversation",
        "q:DeleteConversation",
        "q:PassRequest"
      ],
      "Resource": "*"
   },
    {
      "Sid": "AllowCostExplorerAccess",
      "Effect": "Allow",
      "Action": [
        "ce:GetCostAndUsage",
        "ce:GetCostAndUsageWithResources",
        "ce:GetCostForecast",
        "ce:GetUsageForecast",
```

```
        "ce:GetTags",
        "ce:GetCostCategories",
        "ce:GetDimensionValues",
        "ce:GetSavingsPlansUtilization",
        "ce:GetSavingsPlansCoverage",
        "ce:GetSavingsPlansUtilizationDetails",
        "ce:GetReservationUtilization",
        "ce:GetReservationCoverage",
        "ce:GetSavingsPlansPurchaseRecommendation",
        "ce:GetReservationPurchaseRecommendation",
        "ce:GetRightsizingRecommendation",
        "ce:GetAnomalies",
       "ce:GetCostAndUsageComparisons",
       "ce:GetCostComparisonDrivers"
      ],
      "Resource": "*"
    },
     {
      "Sid": "AllowCostOptimizationHubAccess",
      "Effect": "Allow",
      "Action": [
        "cost-optimization-hub:GetRecommendation",
        "cost-optimization-hub:ListRecommendations",
        "cost-optimization-hub:ListRecommendationSummaries"
      ],
      "Resource": "*"
    },
     {
      "Sid": "AllowComputeOptimizerAccess",
      "Effect": "Allow",
      "Action": [
        "compute-optimizer:GetAutoScalingGroupRecommendations",
        "compute-optimizer:GetEBSVolumeRecommendations",
        "compute-optimizer:GetEC2InstanceRecommendations",
        "compute-optimizer:GetECSServiceRecommendations",
        "compute-optimizer:GetRDSDatabaseRecommendations",
        "compute-optimizer:GetLambdaFunctionRecommendations",
        "compute-optimizer:GetIdleRecommendations",
        "compute-optimizer:GetLicenseRecommendations",
        "compute-optimizer:GetEffectiveRecommendationPreferences"
      ],
      "Resource": "*"
    },
     {
```

```
      "Sid": "AllowBudgetsAccess",
      "Effect": "Allow",
      "Action": [
        "budgets:ViewBudget"
      ],
      "Resource": "*"
    },
     {
      "Sid": "AllowFreeTierAccess",
      "Effect": "Allow",
      "Action": [
        "freetier:GetFreeTierUsage",
        "freetier:GetAccountPlanState",
        "freetier:ListAccountActivities",
       "freetier:GetAccountActivity"
      ],
      "Resource": "*"
    },
     {
      "Sid": "AllowPricingAccess",
      "Effect": "Allow",
      "Action": [
        "pricing:GetProducts",
        "pricing:GetAttributeValues",
        "pricing:DescribeServices"
      ],
      "Resource": "*"
    }
  ]
}
```

Allow administrators to configure plugins

The following example policy grants administrators permissions to view and configure third party plugins in the Amazon Q Developer console. For more information about plugins, see Using plugins with the AWS assistant.

> **Note**
>
> To access the Amazon Q Developer console, administrators also need the permissions in Allow administrators to use the Amazon Q Developer console in the *Amazon Q Developer User Guide*.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "q:CreatePlugin",
        "q:GetPlugin",
        "q:DeletePlugin",
        "q:ListPlugins",
        "q:ListPluginProviders",
        "q:UpdatePlugin",
        "q:CreateAuthGrant",
        "q:CreateOAuthAppConnection",
        "q:SendEvent",
        "q:UpdateAuthGrant",
        "q:UpdateOAuthAppConnection",
        "q:UpdatePlugin",
        "iam:CreateRole",
        "secretsmanager:CreateSecret"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "q.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

Allow administrators to configure plugins from one provider

The following example policy grants an administrator permission to configure plugins from one provider, specified by the plugin ARN with the name of the plugin provider and a wildcard character (*).

> **Note**
>
> To access the Amazon Q Developer console, administrators also need the permissions in Allow administrators to use the Amazon Q Developer console in the *Amazon Q Developer User Guide*.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateProviderPlugin",
            "Effect": "Allow",
            "Action": [
                "q:CreatePlugin",
                "q:GetPlugin",
                "q:DeletePlugin"
            ],
            "Resource": "arn:aws:qdeveloper:us-east-1:111122223333:plugin/plugin-provider/*"
        }
    ]
}
```

To use this policy, replace the following in the ARN in the `Resource` field:

- *`us-east-1`* – The AWS Region where the plugin will be created.
- *`111122223333`* – The AWS account ID of the account where your plugin is configured.
- *`plugin-provider`* – The name of the plugin provider that you want to allow configuration for, like `CloudZero`, `Datadog`, or `Wiz`. The plugin provider field is case sensitive.

Allow users to chat with plugins from one provider

The following example policy grants permission to chat with any plugin from a given provider that an administrator configures, specified by the plugin ARN with the name of the plugin provider and a wildcard character (*). If the plugin is deleted and re-configured, a user with these permissions keeps access to the newly configured plugin.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAmazonQConversationAccess",
            "Effect": "Allow",
            "Action": [
                "q:StartConversation",
                "q:SendMessage",
                "q:GetConversation",
                "q:ListConversations",
                "q:UpdateConversation",
                "q:DeleteConversation"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowAmazonQPluginAccess",
            "Effect": "Allow",
            "Action": [
                "q:UsePlugin"
            ],
            "Resource": "arn:aws:qdeveloper:us-east-1:111122223333:plugin/plugin-
provider/*"
        }
    ]
}
```

To use this policy, replace the following in the ARN in the `Resource` field:

- *`us-east-1`* – The AWS Region where the plugin was created.
- *`111122223333`* – The AWS account ID of the account where your plugin is configured.
- *`plugin-provider`* – The name of the plugin provider that you want to allow access to, like `CloudZero`, `Datadog`, or `Wiz`. The plugin provider field is case sensitive.

Deny the AWS assistant permission to perform specific actions

The following example policy grants permission to chat with the AWS assistant. It allows the AWS assistant to perform any action on your behalf that your IAM identity can perform, except Amazon EC2 actions. The policy uses the `aws:CalledVia` global condition key to deny Amazon EC2 actions only when the AWS assistant calls them.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "q:StartConversation",
        "q:SendMessage",
        "q:GetConversation",
        "q:ListConversations",
        "q:UpdateConversation",
        "q:DeleteConversation",
        "q:PassRequest"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": [
        "ec2:*"
      ],
      "Resource": "*",
      "Condition": {
            "ForAnyValue:StringEquals": {
               "aws:CalledVia": ["q.amazonaws.com"]
            }
       }
    }
  ]
}
```

Allow the AWS assistant permission to perform specific actions

The following example policy grants your IAM identity permission to perform any Amazon EC2 action. However, the AWS assistant can call only `ec2:describeInstances` on your behalf. The policy uses the [`aws:CalledVia` global condition key] to apply this restriction.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "q:StartConversation",
        "q:SendMessage",
        "q:GetConversation",
        "q:ListConversations",
        "q:UpdateConversation",
        "q:DeleteConversation",
        "q:PassRequest"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:*"
      ],
      "Resource": "*",
      "Condition": {
            "ForAnyValue:StringNotEquals": {
               "aws:CalledVia": ["q.amazonaws.com"]
            }
       }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:describeInstances"
      ],
      "Resource": "*",
       "Condition": {
            "ForAnyValue:StringEquals": {
               "aws:CalledVia": ["q.amazonaws.com"]
```

```
            }
        }
    }
  ]
}
```

Allow the AWS assistant to perform actions in specific Regions

The following example policy grants permission to chat with the AWS assistant. When the AWS assistant performs actions on your behalf, it can call only the `us-east-1` and `us-west-2` Regions. For more information, see aws:RequestedRegion in the *IAM User Guide*.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "q:StartConversation",
        "q:SendMessage",
        "q:GetConversation",
        "q:ListConversations",
        "q:UpdateConversation",
        "q:DeleteConversation",
        "q:PassRequest"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
            "aws:RequestedRegion": [
                "us-east-1",
                "us-west-2"
            ]
        }
      }
    }
  ]
}
```

Deny the AWS assistant permission to perform actions on your behalf

The following example policy prevents the AWS assistant from performing actions on your behalf.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAmazonQPassRequest",
      "Effect": "Deny",
      "Action": [
        "q:PassRequest"
      ],
      "Resource": "*"
    }
  ]
}
```

Deny access to the AWS assistant

The following example policy denies all permissions to use the AWS assistant.

> **Note**
>
> When you deny access, the AWS assistant icon and chat panel still appear in the AWS Management Console and AWS Console Mobile Application.

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

Service control policies

Service control policies (SCPs) are organization policies that manage permissions in your organization. You can use an SCP to control which AWS assistant capabilities are available. For more information, see Creating, updating, and deleting service control policies in the *AWS Organizations User Guide*.

Example SCP: Deny access outside the Europe (Frankfurt) Region

The following SCP denies access to AWS assistant chat and related Amazon Q Developer features outside the Europe (Frankfurt) Region (`eu-central-1`). Use it with an Amazon Q Developer profile in that Region to block features that store content and perform inference in the US.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAmazonQDeveloperOutsideEU",
      "Effect": "Deny",
      "Action": [
         "codewhisperer:GenerateRecommendations",
         "q:SendMessage",
         "q:GenerateCodeFromCommands",
         "sqlworkbench:GetQSqlRecommendations"
         ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals":
        {"aws:RequestedRegion": [ "eu-central-1"] }
      }
    }
  ]
}
```

Example SCP: Deny access to the AWS assistant

The following SCP denies access to the AWS assistant.

Note

When you deny access, the AWS assistant icon and chat panel still appear in the AWS Management Console and AWS Console Mobile Application.

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

Supported AWS Regions for the AWS assistant

This topic lists the AWS Regions where you can use the AWS assistant. For more information about Regions, see Specify which AWS Regions your account can use in the *AWS Account Management Reference Guide*.

Your requests might be processed in a different Region from the Region where you use the AWS assistant. For more information, see Cross-Region processing in the AWS assistant. For information about where data is stored, see Data storage in the AWS assistant.

> **Note**
>
> If a request needs information from an opt-in Region (Region that is disabled by default) that isn't listed on this page, the AWS assistant can call that Region. To control which Regions the AWS assistant can call, see Allow the AWS assistant to perform actions in specific Regions.

Regions that are enabled by default

You can use the AWS assistant in the AWS Management Console and AWS Console Mobile Application in the following Regions. You don't need to enable these Regions before you use them. Some features might not be available in every Region. Check the topic for each feature.

- US East (Ohio) (`us-east-2`)
- US East (N. Virginia) (`us-east-1`)
- US West (N. California) (`us-west-1`)
- US West (Oregon) (`us-west-2`)
- Asia Pacific (Mumbai) (`ap-south-1`)
- Asia Pacific (Osaka) (`ap-northeast-3`)
- Asia Pacific (Seoul) (`ap-northeast-2`)
- Asia Pacific (Singapore) (`ap-southeast-1`)
- Asia Pacific (Sydney) (`ap-southeast-2`)
- Asia Pacific (Tokyo) (`ap-northeast-1`)

- Canada (Central) (`ca-central-1`)
- Europe (Frankfurt) (`eu-central-1`)
- Europe (Ireland) (`eu-west-1`)
- Europe (London) (`eu-west-2`)
- Europe (Paris) (`eu-west-3`)
- Europe (Stockholm) (`eu-north-1`)
- South America (São Paulo) (`sa-east-1`)

Regions that are disabled by default

You can use the AWS assistant in the following opt-in Regions at the Free tier only. You must enable an opt-in Region for your AWS account before you can use it. For more information, see Opt-in Region in the *AWS Glossary*.

- Africa (Cape Town) (`af-south-1`)
- Asia Pacific (Hong Kong) (`ap-east-1`)
- Asia Pacific (Hyderabad) (`ap-south-2`)
- Asia Pacific (Jakarta) (`ap-southeast-3`)
- Asia Pacific (Malaysia) (`ap-southeast-5`)
- Asia Pacific (Melbourne) (`ap-southeast-4`)
- Asia Pacific (Thailand) (`ap-southeast-7`)
- Canada West (Calgary) (`ca-west-1`)
- Europe (Milan) (`eu-south-1`)
- Europe (Spain) (`eu-south-2`)
- Europe (Zurich) (`eu-central-2`)
- Mexico (Central) (`mx-central-1`)
- Middle East (Bahrain) (`me-south-1`)
- Middle East (UAE) (`me-central-1`)
- Israel (Tel Aviv) (`il-central-1`)

Administration Regions

Administrators manage Pro tier subscriptions in the Amazon Q Developer console with an Amazon Q Developer profile. The console and profile are supported in the following Regions:

- US East (N. Virginia) (`us-east-1`)
- Europe (Frankfurt) (`eu-central-1`)

This limitation doesn't restrict the Regions where you can use the AWS assistant.

If your profile is in the Europe (Frankfurt) Region, you can't use the following features at the Pro tier:

- Contacting AWS Support
- Troubleshoot resources
- AWS assistant in the AWS Console Mobile Application

Working with other AWS services

The AWS assistant works with the following AWS services. Each entry links to the topic in this guide that describes the integration.

AWS Identity and Access Management (IAM)

Control who can use the AWS assistant and which actions it performs on your behalf. For more information, see Identity and access management and Identity-based policy examples for the AWS assistant.

AWS Command Line Interface (AWS CLI)

Ask the AWS assistant to generate short AWS CLI scripts and code snippets for the AWS SDKs. For example prompts, see Use the prompt library.

AWS CloudTrail

Record the AWS assistant API calls as events. For more information, see Logging and incident response.

Amazon CloudWatch

Analyze telemetry, alarms, and application health. For more information, see Chatting about telemetry and operations.

AWS Cost Explorer and cost management services

Analyze costs, forecasts, and optimization opportunities. For more information, see Chatting about your AWS costs.

AWS Key Management Service (AWS KMS)

Encrypt console chat and console error diagnosis data with a customer managed key. For more information, see Key management.

AWS Organizations

Use service control policies and artificial intelligence services opt-out policies. For more information, see Service control policies and Service improvement for the AWS assistant.

AWS Support

Create a support case and contact a support agent. For more information, see Contacting AWS Support with the AWS assistant.

Troubleshooting access, permission, and response errors in the AWS assistant

Use the following information to diagnose and fix common problems with the AWS assistant. If a problem continues, see Contacting AWS Support with the AWS assistant.

The AWS assistant icon appears, but you can't use it

Possible causes:

- An IAM policy denies q actions for your identity.
- A service control policy (SCP) in your organization denies q actions.
- Your identity doesn't have the permissions that the AWS assistant requires.

To resolve these issues

1. Ask your administrator to check whether an IAM policy or SCP denies q actions. A policy that denies access to the AWS assistant doesn't hide the AWS assistant icon or chat panel.
2. Ask your administrator to give you the permissions that you need.

To prevent this problem, review deny statements for q actions when you change IAM policies or SCPs.

For example policies, see Identity-based policy examples for the AWS assistant.

Response reports insufficient permissions

Possible causes:

- You don't have permission to use the AWS assistant.
- You don't have permission for the service API or resource in your request.
- A permissions boundary, session policy, or organization policy denies the action.

To resolve these issues

1. Note the action that the response reports as denied.
2. Ask your administrator to review the denied action against your IAM, permissions boundary, session, and organization policies.
3. After your administrator updates your permissions, ask your question again.

To prevent this problem, give your users permissions for the services that they ask about, in addition to the permissions for the AWS assistant.

For more information, see Identity and access management.

The AWS assistant can't find a resource

Possible causes:

- Your prompt doesn't include the resource name, ARN, or Region.
- The resource doesn't exist, or your identity can't access it.
- The service doesn't support the required get, list, or describe operation.

To resolve these issues

1. Include the resource name or ARN and its Region in your prompt.
2. Verify that the resource exists and that your identity can access it.
3. For resource troubleshooting, enter your prompt in English.

To prevent this problem, specify the Region in prompts about Regional resources.

For more information, see Chatting about your AWS resources.

Conversation or message is missing

Possible causes:

- You didn't ask a new question in the conversation for 90 days.
- The message is older than 90 days.

- Your administrator changed the customer managed AWS Key Management Service (AWS KMS) key that encrypts chat data.

To resolve these issues

1. Check the age of the conversation or message. You can't recover conversations or messages after the 90-day retention period.
2. If your administrator changed the key, ask your administrator to restore the previous key to access conversations encrypted with it.

To keep a conversation, ask a new question in it within 90 days.

For more information, see Quotas for the AWS assistant and Key management.

Error related to AWS KMS grants

Possible causes:

- The AWS assistant doesn't have permission to create grants for the customer managed AWS KMS key.

To resolve these issues

1. Ask your administrator to open the Amazon Q Developer console.
2. In the permissions banner, your administrator chooses **Update permissions**.

To prevent this problem, make sure that your users have the permissions in Allow the AWS assistant to access customer managed keys.

For more information, see Key management.

Response is incomplete or incorrect

Possible causes:

- The prompt doesn't include enough detail, such as resource identifiers, Regions, or time ranges.
- The AWS assistant doesn't have access to the data that the question requires.

To resolve these issues

1. Add resource identifiers, Regions, time ranges, and the expected outcome to your prompt.
2. Review the source links and API details before you act on a response.
3. To report a problem, choose the thumbs-down icon.

To prevent this problem, start complex questions with the resource, Region, and time range that you want.

If the issue remains unresolved, see Contacting AWS Support with the AWS assistant.

Quotas for the AWS assistant

Your use of the AWS assistant is subject to the following quotas.

Conversation quotas

Name	Default	Adjustable	Description
Saved conversations	80	No	The maximum number of conversations that you can save in the AWS Management Console.
Inactive conversation retention	90 days	No	The AWS assistant deletes a conversation when you don't ask a new question in it for 90 days.
Message retention	90 days	No	The AWS assistant deletes messages older than 90 days, even in an active conversation.

For information about conversations, see Manage console conversations.

Usage quotas

Name	Default	Adjustable	Description
Free tier usage	1,000 queries per month	No	The maximum number of queries

Name	Default	Adjustable	Description
			that you can make each month at the Free tier. For pricing, see Amazon Q Developer pricing.

For information about tiers, see [Optional] Step 3: Review Pro tier access.

Document history for the AWS assistant User Guide

The following table describes the important changes to the documentation since the last release of the AWS assistant. For notification about updates to this documentation, you can subscribe to an RSS feed.

Change	Description	Date
General availability	This is the initial general availability release of the AWS assistant. For more informati on, see What is the AWS assistant?.	November 5, 2026