

User Guide

# Amazon Q Developer



## Amazon Q Developer: User Guide

Copyright © 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

<b>What is Amazon Q Developer?</b> .....	1
Get started .....	1
Amazon Q Developer pricing .....	3
<b>Features</b> .....	4
Analytics .....	4
Summarizing your data .....	4
Management and governance .....	4
Taking inventory .....	4
Use Amazon Q in the AWS Console Mobile Application .....	5
Diagnosing console errors .....	5
Compute .....	5
Choosing Amazon EC2 instances .....	5
Databases .....	6
Writing database queries with natural language .....	6
Networking and content delivery .....	6
Analyzing network reachability .....	6
Developer tools .....	7
Develop software .....	7
Get inline code suggestions .....	7
Chat about code .....	7
Scan your code for security vulnerabilities .....	8
Transforming code .....	8
Developing software in Amazon CodeCatalyst .....	8
Chatting about code in Amazon SageMaker Studio .....	8
Interacting with command line and AWS CloudShell .....	9
Application integration .....	9
Writing scripts to automate AWS services .....	9
Writing ETL scripts and integrating data .....	10
Cloud Financial Management .....	10
Understanding your costs .....	10
Customer support .....	10
Getting customer support directly from Amazon Q .....	10
Creating a support ticket .....	11
Amazon Q in AWS Chatbot .....	11

<b>Getting started .....</b>	<b>12</b>
Tiers of service .....	12
Pro tier .....	12
Setting up access .....	13
Managing subscriptions .....	18
Free tier .....	31
<b>On AWS .....</b>	<b>33</b>
Authenticating to your Amazon Q Developer Pro subscription .....	33
Chatting about AWS .....	34
Add permissions .....	34
Working with Amazon Q on AWS websites .....	34
Example questions .....	35
Chatting about your resources .....	36
Chatting about your costs .....	38
Using plugins .....	39
Datadog .....	40
Wiz .....	47
Console-to-Code .....	54
Console-to-Code .....	54
Where you can use Console-to-Code .....	55
Granting permissions .....	56
Using .....	56
Diagnosing console errors .....	58
Add permissions .....	58
Diagnose common errors in the console .....	59
Chatting with AWS Support .....	59
Prerequisites .....	59
Specify the right service .....	60
Create a support case .....	60
Leave feedback .....	63
Chatting with Amazon Q in AWS Chatbot .....	63
Add Amazon Q to a chat channel .....	63
Ask Amazon Q questions in your channel .....	37
<b>In your IDE .....</b>	<b>65</b>
Supported IDEs .....	65
Installing Amazon Q .....	67

Choose a supported version of your IDE .....	68
In Eclipse IDEs .....	68
In JetBrains IDEs .....	69
In Visual Studio Code .....	70
In Visual Studio .....	71
IAM principals in your AWS console .....	73
Chatting about code .....	73
Working with Amazon Q in your IDE .....	74
Example topics and questions .....	75
Reporting issues with responses .....	75
Explaining and updating code .....	75
Chatting inline .....	76
Adding workspace context .....	78
Generating inline suggestions .....	80
Pausing suggestions .....	81
Amazon Q code completion in action .....	85
Suggestions in AWS coding environments .....	91
Using shortcut keys .....	100
Using code references .....	106
Code examples .....	117
Transforming code .....	137
Transforming Java applications .....	138
Developing software .....	168
Develop features with <code>/dev</code> .....	169
Best practices .....	170
Example tasks .....	171
Quotas .....	172
Troubleshooting .....	172
Scanning your code .....	173
Types of scans .....	174
Quotas .....	172
Starting a scan .....	175
Addressing findings .....	180
Supported languages .....	182
Inline suggestions .....	182
Code Transformation .....	183

Security scans .....	183
<b>On your command line .....</b>	<b>185</b>
Supported environments .....	185
macOS .....	185
Linux .....	186
Installing .....	186
macOS .....	186
AppImage (Linux) .....	186
Ubuntu/Debian (Linux) .....	189
Verifying .....	190
macOS .....	190
AppImage (Linux) .....	191
Ubuntu/Debian (Linux) .....	192
Uninstalling .....	194
Chat .....	194
Context integration .....	194
Context modifiers .....	195
Usage .....	195
Command line completions .....	195
Popular settings .....	196
Over SSH .....	197
Inline .....	201
Usage .....	202
Limitations .....	203
Translating .....	203
Debugging .....	203
Expected output .....	203
Contributing .....	204
<b>Customizing .....</b>	<b>205</b>
Preparing .....	206
Authorizing .....	206
Preparing your data .....	206
Creating .....	207
Connecting to your data source .....	208
Customizations and your data .....	210
Troubleshooting .....	210

Deleting .....	211
Evaluating .....	211
Evaluating .....	212
Optimizing .....	212
Troubleshooting .....	213
Setting up log delivery .....	213
Understanding Amazon CloudWatch Logs .....	214
Understanding console error messages .....	216
Activating .....	217
Activating a version .....	217
Updating .....	218
Creating a new version .....	219
Adding users and groups .....	220
Using .....	220
<b>Dashboard .....</b>	<b>223</b>
Disabling the dashboard .....	223
Switching to the old dashboard .....	224
Old dashboard contents .....	224
Troubleshooting the dashboard .....	225
<b>Security .....</b>	<b>226</b>
Data protection .....	226
Data encryption .....	228
Service improvement .....	230
Opt out of data sharing in the IDE .....	231
Cross-Region calls .....	238
Cross region inference .....	238
Identity and access management .....	239
Audience .....	239
Authenticating with identities .....	240
Managing access using policies .....	243
How Amazon Q works with IAM .....	245
Manage access to Amazon Q .....	251
Amazon Q permissions reference .....	280
AWS managed policies for Amazon Q .....	282
Using service-linked roles .....	290
Compliance validation .....	298

Resilience .....	298
Infrastructure security .....	299
VPC endpoints (AWS PrivateLink) .....	299
Considerations for Amazon Q VPC endpoints .....	300
Prerequisites .....	300
Creating an interface VPC endpoint for Amazon Q .....	300
Using an on-premises computer to connect to a Amazon Q endpoint .....	301
Using an in-console coding environment to connect to a Amazon Q endpoint .....	301
Connecting to Amazon Q through AWS PrivateLink from a third-Party IDE on an Amazon EC2 instance .....	302
<b>Monitoring .....</b>	<b>303</b>
CloudTrail logs .....	303
Amazon Q Developer information in CloudTrail .....	303
Understanding Amazon Q Developer log file entries .....	304
Monitoring with CloudWatch .....	309
Across your organization .....	310
Expenses per user .....	311
Identifying actions by specific users .....	311
UserTriggerDecisionEvent .....	326
CodeScanEvent .....	326
CodeScanRemediationsEvent .....	326
ChatAddMessageEvent .....	327
ChatInteractWithMessageEvent .....	327
TerminalUserInteractionEvent .....	327
Accessing Amazon CloudWatch Logs .....	328
<b>Supported Regions .....</b>	<b>330</b>
Supported Regions (enabled by default) .....	330
Supported opt-in Regions .....	331
<b>Pro tier quotas .....</b>	<b>172</b>
Quotas for Amazon Q (in IDEs) .....	333
Quotas for Amazon Q (in the AWS Management Console) .....	333
Quotas for Amazon Q (in Amazon CodeCatalyst) .....	334
<b>Amazon Q Developer service rename .....</b>	<b>335</b>
<b>Document history .....</b>	<b>336</b>

# What is Amazon Q Developer?

## Note

Powered by Amazon Bedrock: AWS implements [automated abuse detection](#). Because Amazon Q Developer is built on Amazon Bedrock, users can take full advantage of the controls implemented in Amazon Bedrock to enforce safety, security, and the responsible use of artificial intelligence (AI).

Amazon Q Developer is a generative artificial intelligence (AI) powered conversational assistant that can help you understand, build, extend, and operate AWS applications. You can ask questions about AWS architecture, your AWS resources, best practices, documentation, support, and more. Amazon Q is constantly updating its capabilities so your questions get the most contextually relevant and actionable answers.

When used in an integrated development environment (IDE), Amazon Q provides software development assistance. Amazon Q can chat about code, provide inline code completions, generate new code, scan your code for security vulnerabilities, and make code upgrades and improvements, such as language updates, debugging, and optimizations.

Amazon Q is powered by [Amazon Bedrock](#), a fully managed service that makes foundation models (FMs) available through an API. The model that powers Amazon Q has been augmented with high quality AWS content to get you more complete, actionable, and referenced answers to accelerate your building on AWS.

## Note

This is the documentation for Amazon Q Developer. If you are looking for documentation for Amazon Q Business, see the [Amazon Q Business User Guide](#).

# Get started with Amazon Q Developer

To quickly get started using Amazon Q, you can access it in the following ways:

## AWS apps and websites

Add the [necessary permissions](#) to your IAM identity, and then choose the Amazon Q icon to start chatting in the AWS Management Console, AWS Documentation website, AWS website, or AWS Console Mobile Application. For more information, see [Using Amazon Q Developer on AWS apps and websites](#).

## IDEs

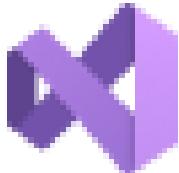
Download the Amazon Q extension and use your AWS Builder ID (no AWS account required) to sign in for free.



[Download Amazon Q in Visual Studio Code](#)



[Download Amazon Q in JetBrains IDEs](#)



[Download Amazon Q in the AWS Toolkit for Visual Studio](#)



[Download Amazon Q in Eclipse IDEs \(Preview\)](#)

From your IDE, choose the Amazon Q icon to start chatting or initiate a development workflow. For more information, see [Installing the Amazon Q Developer extension or plugin in your IDE](#).

## Command line

Download [Amazon Q for the macOS command line](#). For more information, see [Using Amazon Q Developer on the command line](#).

## AWS Chatbot for Microsoft Teams and Slack

Add the [AmazonQFullAccess](#) managed policy to your IAM identity and channel guardrails for Microsoft Teams or Slack channels configured with AWS Chatbot. For more information, see [Chatting with Amazon Q Developer in AWS Chatbot](#).

## Amazon Q Developer pricing

Amazon Q Developer is available through a Free tier and the Amazon Q Developer Pro subscription. For more information, see [Amazon Q Developer pricing](#).

# Amazon Q Developer features

Amazon Q Developer is available across AWS environments and services, and also as a coding assistant in third party IDEs.

Many of Amazon Q Developer's capabilities exist in a chat interface, where you can use natural language to ask questions about AWS, get help with code, explore resources, or troubleshoot. When you chat with Amazon Q, Amazon Q uses the context of your current conversation to inform its responses. You can ask follow-up questions or refer to its response when you ask a new question.

Other Amazon Q Developer features are available as a part of your workflows in AWS service consoles and supported IDEs. The following sections explain the different features of Amazon Q Developer that you might encounter across your AWS experience.

## Analytics

### Summarizing your data

With Amazon Q Amazon QuickSight, you can utilize the Generative BI authoring experience, create executive summaries of your data, ask and answer questions of data, and generate data stories.

For more information, see [Using Generative BI with Amazon Q Amazon QuickSight](#) in the *Amazon QuickSight User Guide*.

## Management and governance

### Taking inventory of your AWS resources

You can ask Amazon Q about your specific AWS account resources from anywhere in the AWS Management Console. You might not know where to locate relevant information about your resources, or you might be in one service console and want to access information about another service's resources without disrupting your workflow.

Amazon Q Developer answers your natural language questions about resources and provides deep links to those resources so you can quickly find them. You can ask Amazon Q to list a type of

resource in your account, for details about a specific resource, or to list resources based on a criteria such as region or state.

For example, you may want to know how many Amazon EC2 instances you currently have running. In that case, you can ask Amazon Q your question in natural language, and it will provide an answer based on your specific resources.

For more information, see [Chatting about your resources](#).

For information about specific limits for each type, and how they relate to pricing for specific subscription package, see [Amazon Q Developer pricing](#).

## Use Amazon Q in the AWS Console Mobile Application

Amazon Q is integrated with the AWS Console Mobile Application to answer questions about AWS. You configure access the same way that you get access to Amazon Q in the AWS Management Console. For more information, see [Getting started with Amazon Q Developer](#).

## Diagnosing console errors

In the AWS Management Console, Amazon Q Developer can diagnose common errors you receive while working with AWS services, such as insufficient permissions, incorrect configuration, and exceeding service limits.

For more information, see [Diagnosing common errors in the console with Amazon Q Developer](#).

## Compute

### Choosing Amazon Elastic Compute Cloud instances

With so many Amazon EC2 instance types available, finding the right instance types for your workload can be time-consuming and complex. The Amazon Q instance type selector considers your use case, workload type, CPU manufacturer preference, and how you prioritize price and performance, as well as additional parameters that you can specify. It then uses this data to provide suggestions and guidance for Amazon EC2 instance types that are best suited to your new workloads.

For more information, see [Get recommendations from Amazon EC2 instance type finder](#) in the [Amazon Elastic Compute Cloud User Guide](#).

The screenshot shows the Amazon EC2 Dashboard. On the left, a sidebar lists navigation items: EC2 Dashboard, EC2 Global View, Events, Console-to-Code (Preview), Instances (with sub-options: Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security (Security Groups). The main content area has three main sections: 'Resources' (listing running instances, auto scaling groups, dedicated hosts, elastic IPs, instances, key pairs, load balancers, placement groups, security groups, snapshots, and volumes), 'Launch instance' (with 'Launch instance' and 'Migrate a server' buttons and a note about launching in the US East (N. Virginia) Region), and 'Service health' (AWS Health Dashboard, status: 'This service is operating normally'). On the right, there's a panel for 'Account attributes' (Default VPC set to vpc-92304aeb, Settings, Data protection and security, Zones, EC2 Serial Console, Default credit specification, EC2 console preferences) and an 'Explore AWS' section with links for better price performance and AWS Graviton2.

## Databases

### Writing database queries with natural language

Amazon Q generative SQL uses generative AI to analyze user intent, query patterns, and schema metadata to identify common SQL query patterns directly within Amazon Redshift, accelerating the query authoring process for users and reducing the time required to derive actionable data insights.

For more information, see [Interacting with Amazon Q generative SQL](#) in the Amazon Redshift Management Guide.

## Networking and content delivery

### Analyzing network reachability

You can use Amazon Q, the generative AI assistant for AWS, to help you diagnose network connectivity issues for applications that run in your virtual private clouds (VPCs). Amazon

Q network reachability analysis can understand natural language queries, and works with Reachability Analyzer to provide relevant responses. With Amazon Q, you can ask network reachability questions in a conversational format.

For more information, see [Amazon Q network reachability analysis](#) in the *Amazon VPC Reachability Analyzer Guide*.

## Developer tools

Ask Amazon Q Developer questions about building at AWS and for assistance with software development. Amazon Q can explain coding concepts and code snippets, generate code and unit tests, and improve code, including debugging or refactoring.

### Develop software

After you explain, in natural language, the feature that you want to develop, Amazon Q can use the context of your current project to generate an implementation plan and the accompanying code. Amazon Q can help you build AWS projects or your own applications. For more information, see [Developing software with the Amazon Q Developer Agent for software development](#).

### Get inline code suggestions

Amazon Q provides you with code recommendations in real time. As you write code, Amazon Q automatically generates suggestions based on your existing code and comments. For more information, see [Generating inline suggestions with Amazon Q Developer](#).

### Chat about code

Within integrated development environments (IDEs), Amazon Q can answer questions related to the software development process, including conceptual questions about programming and how specific code works. You can also ask Amazon Q to update and improve code snippets from the chat panel. For more information, see [Chatting with Amazon Q Developer about code](#).

To write code and get development assistance in the most full-featured environment with Amazon Q Developer, see [Using Amazon Q Developer in the IDE](#).

To enable basic code completion functionality in other interfaces across AWS, see [Generating inline suggestions in AWS coding environments](#).

## Scan your code for security vulnerabilities

Within IDEs, Amazon Q scans your code for security vulnerabilities and code quality issues. You can get security recommendations as you code or scan entire projects to monitor the security posture of your applications throughout development. For more information, see [Scanning your code with Amazon Q](#).

## Transforming code

Within IDEs, Amazon Q can update the language version of your code files. Currently, Amazon Q Code Transformation supports updating Java 8 and Java 11 code to Java 17. For more information, see [Upgrading Java versions with Amazon Q Developer](#).

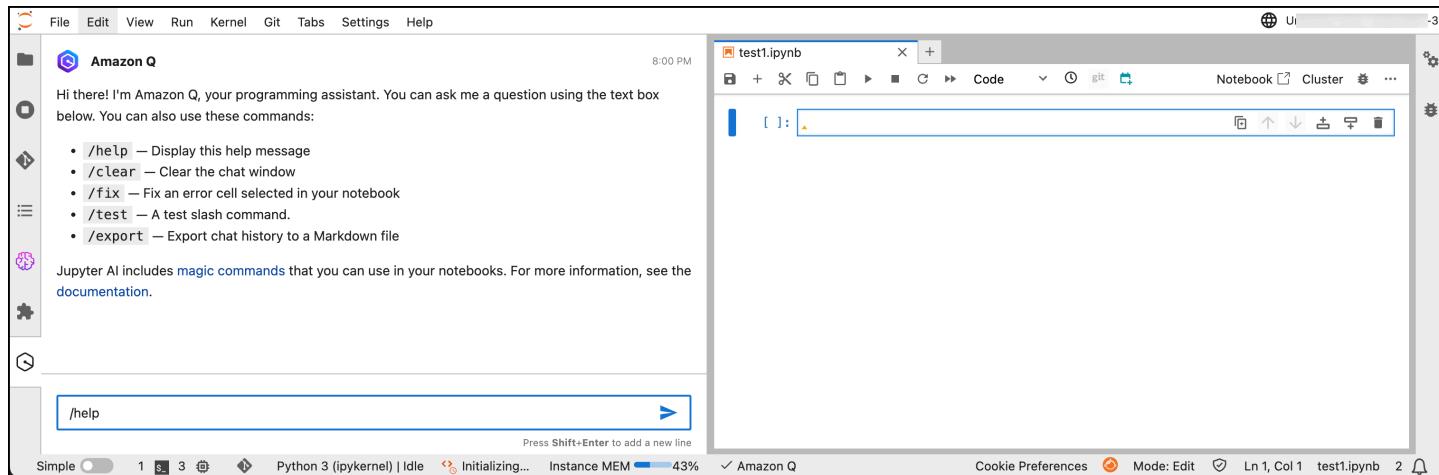
## Developing software in Amazon CodeCatalyst

Amazon Q Developer in CodeCatalyst includes generative AI features that can help users in projects in your space develop software faster. You can assign issues to Amazon Q or recommend tasks for Amazon Q. You can also ask Amazon Q to write a description or to summarize content.

For more information, see [Managing generative AI features in Amazon CodeCatalyst](#) in the *Amazon CodeCatalyst administrator guide*.

## Chatting about code in Amazon SageMaker Studio

Amazon SageMaker Studio is a web-based experience for running ML workflows. You can chat with Amazon Q Developer inside Studio to get guidance on SageMaker features, troubleshoot JupyterLab errors, and get sample code.



For more information, see [Use Amazon Q to Expedite Your Machine Learning Workflows](#) in the *SageMaker Developer Guide*.

## Interacting with command line and AWS CloudShell

After installing Amazon Q for command line, you can use it to complete CLI commands as it populates contextually relevant subcommands, options and arguments. It provides AI-generated completions as you type in the command line. Additionally, you can use Amazon Q to write natural language instructions that are instantly translated to an executable shell code snippet. For more information, see [Using Amazon Q Developer on the command line](#).

You can also use Amazon Q CLI in AWS CloudShell to interact in natural language conversations, ask questions, and receive responses from Amazon Q in your terminal. You can get the related shell command that reduces the need to search for or remember syntax. With Amazon Q, you can receive command suggestions as you type in the terminal. For more information, see [Using Amazon Q AWS CLI in AWS CloudShell](#).

## Application integration

### Writing scripts to automate AWS services

You may know exactly what to do with your AWS resources, and you may find yourself taking the same actions repeatedly. In that case, you can ask Amazon Q to write code that will automate the repetitive tasks.

For example, you may be working on a project that uses Amazon VPCs, Amazon EC2 instances, and Amazon RDS databases. In the course of your testing, you find that every time you create a Amazon VPC, spin up a server, and deploy a database, the configuration is the same. You always choose the same instance and database type, with the same options selected, using the same security groups, in subnets with the same NACL configuration. You don't want to have to go through the same manual process every time you want to re-create your test conditions.

You can use Amazon Q's Console-to-Code feature to automate a workflow instead of performing it manually every time. First, you activate Console-to-Code in the Amazon EC2 console. Then, Amazon Q records your actions as you go through the process of configuring and launching your instance. Finally, Amazon Q provides you with code, in a language of your choice, that automates the process you just performed.

For more information, see [Automating AWS services with Amazon Q Developer Console-to-Code](#).

## Writing ETL scripts and integrating data

AWS Glue is a serverless data integration service that makes it easy for analytics users to discover, prepare, move, and integrate data from multiple sources.

Amazon Q data integration in AWS Glue includes the following capabilities:

- **Chat** – Amazon Q data integration in AWS Glue can answer natural language questions in English about AWS Glue and data integration domains like AWS Glue source and destination connectors, AWS Glue ETL jobs, Data Catalog, crawlers and AWS Lake Formation, and other feature documentation, and best practices. Amazon Q data integration in AWS Glue responds with step-by-step instructions, and includes references to its information sources.
- **Data integration code generation** – Amazon Q data integration in AWS Glue can answer questions about AWS Glue ETL scripts, and generate new code given a natural language question in English.
- **Troubleshoot** – Amazon Q data integration in AWS Glue is purpose built to help you understand errors in AWS Glue jobs and provides step-by-step instructions, to root cause and resolve your issues.

For more information, see [Amazon Q data integration in AWS Glue](#) in the *AWS Glue User Guide*.

## Cloud Financial Management

### Understanding your costs

You can ask Amazon Q about your AWS bill and account costs in the AWS Management Console. Amazon Q can retrieve your cost data, explain costs, and analyze cost trends.

For more information, see [Chatting about your costs](#).

### Customer support

### Getting customer support directly from Amazon Q

Amazon Q can answer your questions about account activation, cost spikes, bill adjustment, fraud and health events.

## Creating a support ticket

Amazon Q can help you create a support case and then connect you to a human support agent at AWS.

For more information, see [Using Amazon Q Developer to chat with AWS Support](#).

## Amazon Q in AWS Chatbot

You can activate Amazon Q in your Slack and Microsoft Teams channels that are configured with AWS Chatbot to ask questions about building at AWS. To add Amazon Q to your channels, see [Chatting with Amazon Q Developer in AWS Chatbot](#). For more information, see [Get started with Slack](#) and [Get started with Microsoft Teams](#) in the *AWS Chatbot Administrator Guide*.

# Getting started with Amazon Q Developer

The following topics describe how to set up Amazon Q Developer Pro for workforce users in your organization, and how individual users can access the Amazon Q Developer Free tier.

## Topics

- [Understanding tiers of service for Amazon Q Developer](#)
- [Amazon Q Developer Pro tier](#)
- [Amazon Q Developer Free tier](#)

## Understanding tiers of service for Amazon Q Developer

When you use Amazon Q Developer, you either use Amazon Q Developer Pro, which is a paid subscription service, or you use Amazon Q Developer at the Free tier. Review the following information to understand what's offered at each tier.

- **Amazon Q Developer Pro tier** – The Pro tier is a paid version of the Amazon Q Developer service. This gives you access to advanced features, such as customization, as well as higher usage limits. To use Amazon Q Developer Pro, you must be a user in IAM Identity Center, and your administrator must subscribe you to Amazon Q Developer Pro. For more information, see [Amazon Q Developer Pro tier](#).
- **Amazon Q Developer Free tier** – Amazon Q Developer offers a perpetual Free tier with monthly limits, including for users authenticating with AWS Builder ID. The features available to you depends on your interface and on how you authenticate. For more information on authentication and access at the Free tier, see [Amazon Q Developer Free tier](#).

For more information about pricing tiers and feature availability, see the [Amazon Q Developer pricing page](#).

## Amazon Q Developer Pro tier

With a paid subscription to the Amazon Q Developer Pro tier, your workforce users can access all features of Amazon Q Developer, including ML-assisted software development in your IDE and chatting with Amazon Q in the AWS Management Console without hard monthly limits.

This section describes how to set up and manage Amazon Q Developer Pro tier subscriptions.

For more information on what you can access with the Pro tier, see [the Amazon Q Developer pricing page](#).

## Topics

- [Setting up access to the Amazon Q Developer Pro tier](#)
- [Managing Amazon Q Developer Pro subscriptions](#)

## Setting up access to the Amazon Q Developer Pro tier

This section provides instructions for setting up access to an Amazon Q Developer Pro subscription. It presents considerations for choosing which type of IAM Identity Center instance (organization or account) you should enable, including the advantages and disadvantages of each option.

After you choose the instance type that you will use to enable users in your organization, choose the corresponding topic for the prerequisites and setup instructions.

## Topics

- [Supported IAM Identity Center Regions for Amazon Q Developer Pro](#)
- [Considerations for choosing your instance type for IAM Identity Center](#)
- [Subscribing users to the Amazon Q Developer Pro tier with an organization instance](#)
- [Subscribing users to the Amazon Q Developer Pro tier with an account instance](#)

## Supported IAM Identity Center Regions for Amazon Q Developer Pro

You can set up an IAM Identity Center instance for your Amazon Q Developer Pro subscription in the Regions listed on the [Supported Regions page](#), except for opt-in Regions. Subscriptions are not supported for IAM Identity Center instances in opt-in Regions. If your IAM Identity Center instance is in an opt-in Region, you will only have access to the Free tier in the AWS console, and you won't have access to Amazon Q in the IDE. Regardless of where you set up an IAM Identity Center instance or use Amazon Q Developer, data is sent to and stored in a US Region. For more information, see [Data protection in Amazon Q Developer](#).

## Considerations for choosing your instance type for IAM Identity Center

There are two ways to enable the Amazon Q Developer Pro tier for your workforce users, depending on your need, security requirements, and feature access level:

- **(Recommended) Organization instance:** An organization instance of IAM Identity Center is the primary form of deploying IAM Identity Center. AWS recommends that you use an organization instance in most cases. If you want access to all Amazon Q Developer features, and for your administrator to have enterprise access controls across multiple AWS accounts, then you should use an organization instance. To set up access with an organization instance, see [Subscribing users to the Amazon Q Developer Pro tier with an organization instance](#).
- **Account instance:** If you cannot adopt an organization instance in IAM Identity Center, you can use an account instance of IAM Identity Center to manage user and group access to Amazon Q Developer features. With an account instance of IAM Identity Center, you can create an isolated deployment of Amazon Q in a single AWS account. To set up access with an account instance, see [Subscribing users to the Amazon Q Developer Pro tier with an account instance](#).

## Use cases for account instances with Amazon Q Developer Pro

Although we recommend that you use an organization instance of IAM Identity Center with Amazon Q Developer Pro, there are a few situations in which it might make sense to use account instances. These situations include:

- You are trying out Amazon Q Developer Pro, and you haven't yet decided that you want to deploy it to multiple AWS accounts across an organization in AWS Organizations.
- You are the administrator of a single AWS account within an organization. Instead of waiting for the administrator of your enterprise to implement Amazon Q Developer Pro, you want to use Amazon Q in just for the AWS account that you control.
- Your enterprise is large, and does not have a single identity provider, or a single identity store, containing the entire user base that you want to give access to Amazon Q Developer.

Disadvantages to using an account instance include:

- The dashboard will only provide information about users and groups associated with the one account.
- The administrative settings (such as whether to include suggestions with code references) will only be available for the one account.

For more information on these instance types, see [Manage organization and account instances of IAM Identity Center](#) in the *AWS IAM Identity Center User Guide*.

## Subscribing users to the Amazon Q Developer Pro tier with an organization instance

Use the procedures on this page to subscribe workforce users in your organization to Amazon Q Developer Pro. To subscribe users, you must have administrator permissions to set up AWS IAM Identity Center and Amazon Q Developer subscriptions.

If you can't adopt an organization instance in IAM Identity Center, see [Subscribing users to the Amazon Q Developer Pro tier with an account instance](#). Note that you can't convert or merge an account instance into an organization instance.

### Prerequisites

Before you can subscribe users, you must complete the following requirements:

- If you don't have an AWS account, [Set up your AWS account](#) in the *AWS Setup User Guide*.
- If you haven't enabled IAM Identity Center, [Set up an organization instance of AWS IAM Identity Center](#).
- Add users to your IAM Identity Center instance, from an external identity provider if applicable.
- From the IAM Identity Center console, copy [the AWS access portal URL](#). Your end users will need this to authenticate to Amazon Q Developer in the IDE.
- Attach administrator permissions to configure subscriptions and Amazon Q Developer settings in the AWS Management Console.
  - For permissions needed to subscribe users in the Amazon Q subscriptions console, see [Allow administrators to use the Amazon Q subscription console](#).
  - For permissions needed to manage Amazon Q Developer Pro settings in the Amazon Q Developer console, see [Allow administrators to use the Amazon Q Developer console](#)

### Subscribe users to Amazon Q Developer Pro

To subscribe users, you'll use the Amazon Q console, *not* the Amazon Q Developer console. The Amazon Q Developer console is for configuring features and settings, and viewing dashboard metrics.

#### To subscribe users

1. Sign in to the Amazon Q console.

In the main pane, you see the following message:



### Amazon Q connected to IAM Identity Center

#### Note

If you don't see this message, it might be because you're in the wrong console or AWS Region. Make sure you're in the Amazon Q console, in the same Region as your IAM Identity Center instance. (It's OK if the Amazon Q and IAM Identity Center consoles are in a different Region from the Amazon Q Developer console.)

2. (Optional) Enable identity-aware sessions. This is necessary to activate certain features of Amazon Q Developer Pro, including higher monthly limits on queries.

For more information, see [Enabling identity-aware console sessions](#) in the *AWS IAM Identity Center User Guide*.

3. Under **Amazon Q Developer Pro**, choose **Subscribe**.
4. In the **Create Amazon Q Developer Pro application** dialog box, choose **Create application**.
5. In the **Assign users and groups** dialog box, choose **Get started**.
6. Search for and select the users and groups who should have subscriptions to Amazon Q Developer Pro. Then choose **Done**.
7. To start using Amazon Q Developer, end users can sign in with their IAM Identity Center credentials and the access portal URL that you obtained from the AWS IAM Identity Center console.

To help your end users get started with Amazon Q Developer Pro, see [Using Amazon Q Developer on AWS apps and websites](#), [Using Amazon Q Developer in the IDE](#), and [Using Amazon Q Developer on the command line](#).

To manage subscriptions and settings, including what features are enabled in your organization, see [Managing Amazon Q Developer Pro subscriptions](#).

## Subscribing users to the Amazon Q Developer Pro tier with an account instance

Use the procedures on this page to subscribe to Amazon Q Developer Pro with an account instance. To subscribe users, you must have administrator permissions to set up AWS IAM Identity Center and Amazon Q Developer subscriptions.

Note that [Using Amazon Q Developer on AWS apps and websites](#) requires an [organization instance](#), and you can't convert or merge an account instance into an organization instance. To set up Amazon Q Developer with an organization instance, see [Subscribing users to the Amazon Q Developer Pro tier with an organization instance](#).

### Prerequisites

Before you can subscribe users, you must complete the following requirements:

- If you don't have an AWS account, [Set up your AWS account](#) in the *AWS Setup User Guide*.
- If you haven't enabled IAM Identity Center, [Set up an account instance of AWS IAM Identity Center](#).
- Add users to your IAM Identity Center instance, from an external identity provider if applicable.
- From the IAM Identity Center console, copy [the AWS access portal URL](#). Your end users will need this to authenticate to Amazon Q Developer in the IDE.
- Attach administrator permissions to configure subscriptions and Amazon Q Developer settings in the AWS Management Console.
  - For permissions needed to subscribe users in the Amazon Q subscriptions console, see [Allow administrators to use the Amazon Q subscription console](#).
  - For permissions needed to manage Amazon Q Developer Pro settings in the Amazon Q Developer console, see [Allow administrators to use the Amazon Q Developer console](#).

### Subscribe users to Amazon Q Developer Pro

To subscribe users, you'll use the Amazon Q console, *not* the Amazon Q Developer console. The Amazon Q Developer console is for configuring features and settings, and viewing dashboard metrics.

#### To subscribe users

1. Sign in to the Amazon Q console.

In the main pane, you see the following message:



**Amazon Q connected to an account instance of IAM Identity Center**

**Note**

If you don't see this message, it might be because you're in the wrong console or AWS Region. Make sure you're in the Amazon Q console, in the same Region as your IAM Identity Center instance. (It's OK if the Amazon Q and IAM Identity Center consoles are in a different Region from the Amazon Q Developer console.)

2. Under **Amazon Q Developer Pro**, choose **Subscribe**.
3. In the **Create Amazon Q Developer Pro application** dialog box, choose **Create application**.
4. In the **Assign users and groups** dialog box, choose **Get started**.
5. Search for and select the users and groups who should have subscriptions to Amazon Q Developer Pro. Then choose **Done**.
6. To start using Amazon Q Developer, end users can sign in with their IAM Identity Center credentials and the access portal URL that you obtained from the AWS IAM Identity Center console.

To help your end users get started with Amazon Q Developer Pro, see [Using Amazon Q Developer in the IDE](#), and [Using Amazon Q Developer on the command line](#).

To manage subscriptions and settings, including what features are enabled in your organization, see [Managing Amazon Q Developer Pro subscriptions](#).

## Managing Amazon Q Developer Pro subscriptions

This section helps you understand and manage Amazon Q Developer subscriptions, including configuring Amazon Q Developer settings, unsubscribing users, and troubleshooting common subscription issues to ensure your users have access to Amazon Q Developer features.

### Topics

- [Understanding subscriptions in Amazon Q Developer](#)
- [Managing Amazon Q Developer Pro settings](#)
- [Troubleshooting Amazon Q Developer Pro subscriptions](#)
- [Unsubscribing users from Amazon Q Developer Pro](#)

## Understanding subscriptions in Amazon Q Developer

The Amazon Q subscription console gives administrators visibility into how end users are utilizing Amazon Q Developer Pro, Amazon Q Business Pro, and Amazon Q Business Lite subscriptions. It also provides a list of *associations*, which encompass the applications and accounts that an end user has access to through their subscriptions. Administrators can associate end users with a subscriptions, applications, or accounts.

In the Amazon Q subscription console, administrators can view a list of subscribed users, their subscription status (active, pending, under free trial, or canceled), and their corresponding associations. The associations that administrators can view depends on the permissions assigned to those administrators.

A subscription is pending until the first time an end user accesses a feature of Amazon Q Developer, including sending a message in Amazon Q chat or using inline suggestions in the IDE. You won't be charged for a subscription until it is active.

If [trusted access](#) is enabled for the organization, organization management account administrators can see all subscription associations across applications and across various accounts. However, member account administrators can only view the subscriptions and applications within the accounts that they administer. This means that if one account administrator subscribes an end user to an Amazon Q subscription and provides access to an application in that account, another member account administrator in a different account won't be able to see it.

Both organization management administrators and member account administrators can view end users' applications and AWS account associations that they have access to, filter and search end users, and download reports.

## Managing Amazon Q Developer Pro settings

This page describes how to configure settings for your Amazon Q Developer Pro subscription in the Amazon Q Developer Pro console, including managing access to features and enabling trusted access.

To access the Amazon Q Developer Pro console, you must have the permissions defined in [Allow administrators to use the Amazon Q Developer console](#).

## Managing account details

To manage what features of Amazon Q Developer are available to your users and access the Start URL that your users need to access Amazon Q Developer in the IDE, use the following procedure.

1. Open the Amazon Q Developer console.
2. Choose **Settings**.

To update what features are available to your users, choose **Edit** in the Amazon Q Developer account details panel.

The screenshot shows the 'Amazon Q Developer account details' settings page. At the top right, there is a 'Delete profile' button and an 'Edit' button, which is highlighted with a red oval. The page is divided into several sections:

- Enabled features**:
  - Amazon Q in the IDE**: Includes Amazon Q Developer Agent for code transformation, IDE inline completion, IDE chat, Amazon Q Developer Agent for software development, and Code security and code quality scans.
  - Amazon Q in the command line**: Includes Command line chat, Command line classic completions, and Command line AI completions.
  - Amazon Q in the AWS Console**: Includes Amazon Q assistant, Amazon Q troubleshooting, and Amazon Q actions.
  - Amazon CodeCatalyst**: Includes Amazon Q Developer Agent for software development, PR comment summary, and Write PR description.
- Q ChatBots**: Includes AWS Console Mobile Application and AWS chatbot in Slack and Teams.
- Deployment settings**: Includes options for **Include suggestions with code references** (set to True), **Encryption key** (Default AWS managed key), **Region** (us-east-1), and **ARN** (represented by a truncated string of black and white bars).
- Start URL**: A text field containing a URL, with a note below it stating: "Give this URL to your users. This is required for them to access Amazon Q Developer in their IDEs. [Learn more](#)".

3. To copy the Start URL, choose the clipboard icon under **Start URL**.

Provide the Start URL to your developers when they need to authenticate with their Amazon Q Developer Pro subscription in the IDE. For more information, see [Using Amazon Q Developer in the IDE](#).

## Manage encryption method

By default, Amazon Q Developer uses an AWS managed key for encryption. For some features, you can set up a customer managed key to encrypt data. For a list of features that support encryption with customer managed keys, see [Data encryption](#).

To set the key used for encryption, complete the following procedure.

1. Open the Amazon Q Developer console.
2. Choose **Settings**.
3. Choose **Edit** in the Amazon Q Developer account details panel.

The screenshot shows the 'Settings' page for an Amazon Q Developer account. At the top right is a 'Delete profile' button. Below it is an 'Edit' button, which is circled in red. The page is divided into sections: 'Amazon Q Developer account details', 'Enabled features', 'Deployment settings', and 'Start URL'. Under 'Enabled features', there are four main categories: 'Amazon Q in the IDE', 'Amazon Q in the AWS Console', 'Q ChatBots', and 'Amazon Q in the command line'. Each category lists specific features or actions. Under 'Deployment settings', there are four columns: 'Include suggestions with code references' (set to True), 'Encryption key' (set to Default AWS managed key), 'Region' (set to us-east-1), and 'ARN' (a long string of characters). Under 'Start URL', there is a placeholder URL and a 'Learn more' link.

4. On the **Edit details** page, expand the **Encryption key - optional** section.
5. To use a customer managed key for encryption, select **Customize encryption settings (advanced)**.
6. In the search bar that appears, search for the name of the key you want to use for encryption or enter the key ARN.

If you haven't created a key yet, choose **Create an AWS KMS key**, and then return to this page to add your key.

7. To disable encryption with your customer managed key and revert to an AWS managed key for encryption, deselect **Customize encryption settings (advanced)**.

## Enabling trusted access

Amazon Q Developer uses trusted access to share the settings made in the AWS Organizations management account with member accounts in the same organization.

For example, the Amazon Q Developer Pro administrator, working in the Organizations management account, may enable suggestions with code references. If trusted access is enabled, then suggestions with code references will also be enabled for all member accounts in that organization.

When you subscribe to Amazon Q Developer Pro while using an AWS Organizations management account, you are given the option to share settings with member accounts. You may also share or un-share those settings after subscribing.

To enable or disable trusted access after subscribing, use the following procedure.

1. On the Amazon Q Developer **Settings** page, under **Member account settings**, choose **Edit**.
2. In the pop-up window, select **On** or **Off**.
3. Choose **Save**.

To learn more about trusted access, see [Enabling trusted access for AWS Account Management](#) in the *AWS Organizations User Guide*.

## Enable prompt logging

Administrators can enable the logging of all inline or chat prompts used with Amazon Q Developer in the IDE by developers in your organization. These logs can help with auditing, debugging, analytics, and ensuring compliance.

When developers chat with Amazon Q or use inline code completion in the IDE, Amazon Q will log both the developers' prompts and Amazon Q's responses. When developers chat with [the Amazon Q Agent for software development](#) using the `/dev` command, only the prompts will be logged. (Prompts and responses are also known as request and response parameters, respectively.)

Amazon Q stores the prompt logs in an Amazon S3 bucket that you create, at the following path:  
`bucketName/prefix/AWSLogs/accountId/QDeveloperLogs`

## Prerequisite

Create an Amazon S3 bucket to hold the prompt logs. The bucket must:

- Be in the US East (N. Virginia) Region regardless of where your IAM Identity Center is located.
- Have a bucket policy like the one that follows. Replace *bucketName*, *region*, *accountId*, and *prefix* with your own information.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "QDeveloperLogsWrite",  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "q.amazonaws.com"  
            },  
            "Action": [  
                "s3:PutObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::bucketName/prefix/*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "aws:SourceAccount": "accountId"  
                },  
                "ArnLike": {  
                    "aws:SourceArn": "arn:aws:codewhisperer:region:accountId:*"  
                }  
            }  
        }  
    ]  
}
```

If you're configuring SSE-KMS on the bucket, add the below policy on the KMS key:

```
{  
    "Effect": "Allow",  
    "Principal": {  
        "Service": "q.amazonaws.com"  
    },  
    "Action": "kms:GenerateDataKey",  
    "Resource": "*",  
    "Condition": {  
        "StringEquals": {
```

```
        "aws:SourceAccount": "accountId"  
    },  
    "ArnLike": {  
        "aws:SourceArn": "arn:aws:codewhisperer:region:accountId:*"  
    }  
}  
}
```

To learn about protecting the data in your Amazon S3 bucket, see [Protecting data with encryption](#) in the *Amazon Simple Storage Service User Guide*.

## To enable prompt logging

1. Sign in to the Amazon Q Developer console:

If you set up Amazon Q Developer [with an organization instance](#) of IAM Identity Center, then sign in as a management account administrator. (Member account administrators cannot enable prompt logging.)

If you set up Amazon Q Developer [with an account instance](#) of IAM Identity Center, then sign in as an administrator of the account associated with that instance.

2. Choose **Settings**.
3. Under **Preferences**, choose **Edit**.
4. In the Edit preferences window, toggle **Q Developer prompt logging**.
5. Under Amazon S3 location, enter the Amazon S3 URI that you will use to receive the logs.  
Example: s3://amzn-s3-demo-bucket/qdev-prompt-logs/

## Enabling user activity reports

You can configure Amazon Q to collect user activity telemetry of Amazon Q Developer subscribers in your organization and present that information in a report. You can then review the report to gain insights into Amazon Q usage across your user base.

Amazon Q generates the report every day at midnight, and saves it in a CSV file at the following path:

s3://*bucketName*/*prefix*/*month*/*day*/*timestamp*-report.csv

The CSV file is laid out as follows:

- Each row shows a user who interacted with Amazon Q that day.
- Each column shows a metric, as described in the following table. Metrics are calculated based on the user telemetry collected over the course of the day.

If the CSV file surpasses 1 MB, Amazon Q splits the data into several CSV files for the day.

Metric name	Description
Chat_AICodeLines	Lines of code generated by Amazon Q and accepted by the user. This metric includes code that was generated by <a href="#">Amazon Q chat</a> (not <a href="#">inline chat</a> ) and inserted into the IDE.
Chat_MessagesInteracted	Number of chat messages where the user has interacted positively with Amazon Q. Examples of positive interactions: clicking a link, inserting suggestions, and upvoting messages. This metric includes code that was generated by <a href="#">Amazon Q chat</a> (not <a href="#">inline chat</a> ) and inserted into the IDE.
Chat_MessagesSent	Number of messages sent to and from Amazon Q. This metric includes code that was generated by <a href="#">Amazon Q chat</a> (not <a href="#">inline chat</a> ) and inserted into the IDE.
Inline_AICodeLines	Lines of code generated by Amazon Q and accepted by the user. This metric includes code that was accepted as <a href="#">inline suggestions</a> .
Inline_AcceptanceCount	Number of <a href="#">inline suggestions</a> accepted by the user.
Inline_SuggestionsCount	Number of <a href="#">inline suggestions</a> displayed to the user.

### Note

When you enable user activity reports, Amazon Q collects telemetry regardless of how a developer has set the **Enable Amazon Q to send usage data to AWS** setting in their IDE. That setting controls whether telemetry can be used by the *AWS corporation*, not your organization. For more information about this setting, see [Opting out of sharing your client-side telemetry](#).

Use the following instructions to enable user activity reports.

### Prerequisite

Create an Amazon S3 bucket to hold the user activity report CSV file. The bucket must:

- Be in the US East (N. Virginia) Region regardless of where your IAM Identity Center is located.
- (Optional but recommended) Be different from the bucket you might be using for [prompt logging](#).
- Include a prefix, also known as a subfolder, where Amazon Q will save the CSV file. The CSV file cannot be saved in the root of the bucket.
- Have a bucket policy like the one that follows. Replace *bucketName*, *region*, *accountId*, and *prefix* with your own information.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "QDeveloperLogsWrite",  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "q.amazonaws.com"  
            },  
            "Action": [  
                "s3:PutObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::bucketName/prefix/*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "AWS:SourceRegion": "region"  
                }  
            }  
        }  
    ]  
}
```

```
        "aws:SourceAccount": "accountId"  
    },  
    "ArnLike": {  
        "aws:SourceArn": "arn:aws:codewhisperer:region:accountId:*"  
    }  
}  
]  
}  
}
```

If you're configuring SSE-KMS on the bucket, add the below policy on the KMS key:

```
{  
    "Effect": "Allow",  
    "Principal": {  
        "Service": "q.amazonaws.com"  
    },  
    "Action": "kms:GenerateDataKey",  
    "Resource": "*",  
    "Condition": {  
        "StringEquals": {  
            "aws:SourceAccount": "accountId"  
        },  
        "ArnLike": {  
            "aws:SourceArn": "arn:aws:codewhisperer:region:accountId:*"  
        }  
    }  
}
```

To learn about protecting the data in your Amazon S3 bucket, see [Protecting data with encryption](#) in the *Amazon Simple Storage Service User Guide*.

## To enable user activity telemetry reports

### 1. Sign in to the Amazon Q Developer console:

- If you set up Amazon Q Developer [with an organization instance](#) of IAM Identity Center, then sign in using a management account or member account.
- If you set up Amazon Q Developer [with an account instance](#) of IAM Identity Center, then sign in using the account associated with that instance.

2. Choose **Settings**.
3. Under **Q Developer user activity reports**, choose **Edit**.
4. Toggle **Collect granular metrics per user**.
5. Under **S3 location**, enter the Amazon S3 URI that you will use to hold the CSV reports.  
Example: `s3://amzn-s3-demo-bucket/user-activity-reports/`

## Troubleshooting Amazon Q Developer Pro subscriptions

If you are having trouble accessing Amazon Q Developer with a Amazon Q Developer Pro subscription, review the following scenarios to understand possible issues and how to resolve them.

- **You were recently added to a group, and your subscription is not yet active.**

If your administrator manages Amazon Q subscription access using identity provider groups, there might be a delay when they add users to the group. You might need to wait up to 24 hours for access to be activated.

- **You don't have an active subscription.**

Try refreshing the page to use the Amazon Q Developer Free tier.

- **Your access to the Amazon Q Developer Pro managed application was revoked.**

You still have an active subscription, but your access to the Amazon Q Developer Pro managed application was revoked or the managed application was deleted. Contact your administrator to restore your access.

- **You don't have sufficient IAM permissions.**

You or your AWS administrator must update your IAM permissions to allow the use of Amazon Q Developer. For more information, see [Identity-based policy examples for Amazon Q Developer](#). After you obtain the necessary permissions, reload the console page to access Amazon Q.

## Unsubscribing users from Amazon Q Developer Pro

Review the following topics to understand how to unsubscribe users from various Amazon Q subscriptions.

## Unsubscribing from Amazon Q Developer Pro

If you want to remove all associations for an end user who is not part of a group, the organization administrator should first review all associations mapped to that user.

To review associations for a user, the organization administrator must:

1. Sign in to the Amazon Q subscription console
2. Select the **User** tab and review each association for that end user

After you have reviewed your chosen user's associations, you can remove those associations by unsubscribing the user in the appropriate part of the Amazon Q subscription console. Organization management account administrators can only unsubscribe users if they created that subscription. They can also view other accounts the user is subscribed through by choosing **Share settings profile with member accounts** on the **Settings** page. This allows them to coordinate with respective member account administrators for unsubscription. Alternatively, if they have the appropriate permissions, they can sign in as a member account administrator and unsubscribe the user directly.

To unsubscribe a group, the organization or member account administrator who created the association must:

1. Sign in to the Amazon Q subscription management console.
2. Select the group that you want to unsubscribe
3. Choose **Unsubscribe**.

Account administrators can also unsubscribe users or groups from accounts they manage, and have created associations for. Note that member account administrators can only view and unsubscribe users subscribed within the account they have permissions for, and only if they created that specific subscription. This means that if both a member account administrator and an organization administrator have subscribed a user, both account administrators must unsubscribe the user for them to be fully unsubscribed.

To unsubscribe a user you manage, use the following procedure:

1. Sign in to the Amazon Q subscription console.
2. Select the **User** tab, and select the end user you want to unsubscribe.

3. Review the associations for that end user.
4. Choose **Unsubscribe**.

For more information, see [Managing AWS accounts in your organization](#).

## Unsubscribing from Amazon Q Business

To unsubscribe from Amazon Q Business, see [Deleting user subscriptions](#) in the *Amazon Q Business User Guide*.

## Unsubscribing from Amazon QuickSight Q

To unsubscribe from Amazon QuickSight Q, see [Unsubscribing from Amazon QuickSight Q](#) in the *Amazon QuickSight User Guide*.

 **Note**

If a user is using both Amazon QuickSight and Amazon Q Business as part of Amazon Q Business Pro, then you must remove the user from both services in order to avoid further billing for that user.

## Amazon Q Developer Free tier

Amazon Q Developer is available at the Free tier on AWS apps and websites, in the IDE, and on the command line. When you use the Free tier, access to Amazon Q Developer features is limited.

If you want to get started quickly with some Amazon Q Developer features, you can create an [AWS Builder ID](#) to use Amazon Q Developer in the IDE or on the command line. Builder ID is the authentication that we recommend for exploring the Free tier of Amazon Q Developer. To get started with a Builder ID, see [Install the Amazon Q Developer extension or plugin in your IDE](#).

To use Amazon Q Developer at the Free tier on AWS websites, sign in to your AWS account to try chat and other features in the AWS Management Console. Not all Amazon Q console features are available at the Free tier.

The following table describes what features of Amazon Q Developer at the Free tier are available to you, depending on your sign in or authentication method:

Amazon Q Developer feature	Free tier access type		
	AWS Builder ID	IAM Identity Center	IAM federatio n/principals
Amazon Q on AWS websites			
	No	Yes	Yes
Amazon Q Developer in the IDE			
	Yes	No	No
Amazon Q Developer on the command line			
	Yes	No	No

For details about the features available at the Free tier, see [the Amazon Q Developer pricing page](#).

# Using Amazon Q Developer on AWS apps and websites

Use Amazon Q Developer in the AWS Management Console, AWS Console Mobile Application, AWS Marketing website, AWS Documentation website, and chat channels integrated with AWS Chatbot to ask questions about AWS. You can ask Amazon Q about AWS architecture, best practices, support, and documentation. Amazon Q can also help with code that you're writing with the AWS SDKs and AWS Command Line Interface (AWS CLI).

In the AWS Management Console, you can ask Amazon Q about your AWS resources and costs, contact AWS Support directly, and diagnose common console errors.

To quickly provide access to features of Amazon Q Developer on AWS, attach the [AmazonQDeveloperAccess](#) AWS managed policy to the IAM identity using Amazon Q. For permissions needed for specific features, see the topic for the feature you want to use.

## Topics

- [Authenticating to your Amazon Q Developer Pro subscription](#)
- [Chatting with Amazon Q Developer about AWS](#)
- [Using Amazon Q Developer plugins](#)
- [Automating AWS services with Amazon Q Developer Console-to-Code](#)
- [Diagnosing common errors in the console with Amazon Q Developer](#)
- [Using Amazon Q Developer to chat with AWS Support](#)
- [Chatting with Amazon Q Developer in AWS Chatbot](#)

## Authenticating to your Amazon Q Developer Pro subscription

To access Amazon Q at the Free tier, sign in to the AWS Management Console. Any Free tier features are available as long as you have the required permissions.

To access Amazon Q at the Pro tier, you must be a user registered with IAM Identity Center, your IAM Identity Center identity must be subscribed to Amazon Q Developer Pro, and you must sign in to the console with IAM Identity Center.

If you sign in to the AWS console with IAM or federation with IAM, then you will only have access to the Free tier by default. If your organization administrator has enabled identity-aware sessions,

then you will be prompted to authenticate with IAM Identity Center when you reach a Free tier limit or attempt to use a feature only available at the Pro tier.

If you sign in to the AWS console with IAM Identity Center, including authenticating through an external identity provider that is connected to IAM Identity Center, you will automatically have access to the Pro tier if your IAM Identity Center identity is subscribed to Amazon Q Developer Pro.

For more information on the Amazon Q Developer Pro tier, see [Understanding tiers of service for Amazon Q Developer](#).

## Chatting with Amazon Q Developer about AWS

Chat with Amazon Q in the AWS Management Console, AWS Console Mobile Application, AWS website, AWS Documentation website, and chat channels integrated with AWS Chatbot to learn about AWS services. You can ask Amazon Q about best practices, recommendations, step-by-step instructions for AWS tasks, and architecting your AWS resources and workflows.

You can also ask Amazon Q about your AWS resources and account costs. Amazon Q additionally generates short scripts or code snippets to help you get started using the AWS SDKs and AWS CLI.

### Topics

- [Add permissions](#)
- [Working with Amazon Q on AWS websites](#)
- [Example questions](#)
- [Chatting about your resources](#)
- [Chatting about your costs](#)

## Add permissions

For an IAM policy that grants permissions needed for chatting with Amazon Q, see [Allow users to chat with Amazon Q](#).

## Working with Amazon Q on AWS websites

To chat with Amazon Q Developer in the AWS Management Console, choose the Amazon Q icon in the right sidebar. To chat on the AWS website or any AWS service's documentation page, choose the Amazon Q icon in the bottom right corner.

To ask Amazon Q a question, enter your question into the text bar in the Amazon Q panel. Amazon Q generates a response to your question with a sources section that links to its references.

After you receive a response, you can optionally leave feedback by using the thumbs-up and thumbs-down icons. You can also copy the response to your clipboard by choosing the copy icon.

## Conversation history

Amazon Q maintains your conversation within a given session as context to inform future responses. You can ask follow-up questions or refer to previous questions and responses throughout the duration of your session.

If you're using Amazon Q in the console, your current conversation and associated context are maintained when you navigate to another place in the console or to another browser or tab.

If you're using Amazon Q on the AWS website, Documentation website, or Console Mobile Application, a new conversation starts without any context when you navigate to a new page, browser, or tab.

If you want to restart your conversation and clear the context provided by previous questions and responses, choose **New conversation**. Your previous conversation will no longer be used to inform responses from Amazon Q.

## Chat settings

To update your chat settings in Amazon Q, choose the gear icon in the top right corner of the chat panel.

You can specify the following settings:

- **Region** — Amazon Q defaults to the AWS Region set in the AWS Management Console when you open the chat panel. To update the Region used by Amazon Q, change your console Region.
- **Cross-region calls** — To provide some features, such as listing your AWS resources in the chat, Amazon Q might need to make cross-region calls to retrieve information about your resources in other Regions. Enable cross-region calls to grant Amazon Q permission to make calls to other Regions.

## Example questions

You can ask Amazon Q questions about AWS and AWS services, such as finding the right service or understanding best practices.

You can also ask about software development with the AWS SDKs and AWS CLI. Amazon Q in the console can generate short scripts or code snippets to help you get started using the AWS SDKs and AWS CLI.

The following are example questions that demonstrate how Amazon Q can help you build on AWS:

- What's the maximum runtime for a Lambda function?
- When should I put my resources in a VPC?
- What's the best container service to use to run my workload if I need to keep my costs low?
- How do I list my Amazon S3 buckets?
- How do I create and host a website on AWS?

## Chatting about your resources

In the AWS Management Console, you can ask Amazon Q about your AWS account resources. In the chat panel, ask Amazon Q to list a type of resource in your account, for details about a specific resource, or to list resources based on a criteria such as region or state.

Amazon Q can perform get, list, and describe actions to retrieve your AWS resources. For example, you can ask "List my S3 buckets" or "Show my running EC2 instances in us-east-1". Amazon Q can't answer questions about the data stored in your resources, such as listing objects in an Amazon S3 bucket, or questions related to your account security, identity, credentials, or cryptography.

Amazon Q lists up to 10 resources in the chat panel, and the response includes details about each resource, a resource ARN that you can copy, and, if applicable, a link to the service console. When you ask about one resource, the response also includes the resource information in JSON format.

### Topics

- [Prerequisites](#)
- [Ask Amazon Q about resources](#)

## Prerequisites

For Amazon Q to show your resources, the following prerequisites must be met.

### Add permissions

To chat about your resources, your IAM identity must have the following permissions:

- Permissions to chat with Amazon Q and to allow Amazon Q to access your resources. For an IAM policy that grants the required permissions, see [Allow Amazon Q to perform actions on your behalf in chat](#).
- Permissions to access the resources you ask about. For example, if you ask Amazon Q to list your Amazon S3 buckets, you must have the s3>ListAllMyBuckets permission.

Amazon Q will never access resources that your IAM identity doesn't have access to.

### **Important**

Normal fees apply when you ask Amazon Q to perform read, list, or describe actions. For more information, see the pricing page for the AWS service you are asking Amazon Q about.

## Cross-Region consent

Amazon Q chat makes calls from US East (N. Virginia), so it might have to make cross-Region calls to access your resources in another Region, including to opt-in Regions. Amazon Q requires cross-Region consent to retrieve resources. To consent to cross-Region calls, complete the following steps:

1. Open the AWS Management Console and choose the Amazon Q icon.
2. If you haven't consented already, a notification about cross-Region calls appears above the text bar. Choose **Continue** to consent to Amazon Q making cross-Region calls to access your resources. If you choose **Don't allow**, you won't be able to ask Amazon Q about your AWS resources.
3. If you want to modify your cross-Region settings, choose the gear icon in the top right corner of the chat panel.

## Ask Amazon Q about resources

You can ask Amazon Q to list your resources or get details about a specific resource or group of resources.

When you ask Amazon Q about your resources, you can specify the Region that Amazon Q calls to locate your resources. If no Region is specified, Amazon Q uses your current console Region, or

the most recent console Region if you are using a global console Region. If no Region is found, it defaults to calling US East (N. Virginia).

Amazon Q might need additional information to retrieve your resources, such as the name of a resource or a resource ARN. When Amazon Q asks a follow up, reply with the requested details.

Following are example questions you can ask Amazon Q about your resources:

- Describe the encryption settings for S3 bucket `<name>`
- List my EC2 instances in us-west-2
- Get the configuration for my lambda function `<name>`
- What alarms are configured for instance `<instance ID>`?

## Chatting about your costs

You can ask Amazon Q about your AWS bill and account costs in the AWS Management Console. Amazon Q can retrieve your cost data, explain costs, and analyze cost trends, so you can understand your costs without referring to documentation or interrupting your workflow.

When you ask Amazon Q about your costs, its response includes information about the requested cost figure, including the metric, time period and granularity, and any groups or filters applied to retrieve the cost data. It also provides a link to an AWS Cost Explorer view with the same specifications, so you can see visualizations or download the data for your own analysis.

For more information about cost analysis in Amazon Q, see [Analyzing your Cost Explorer data with Amazon Q](#) in the *AWS Cost Management User Guide*.

## Prerequisites

### Add permissions

To chat about your costs, your IAM identity must have the following permissions:

- Permissions to chat with Amazon Q and to allow Amazon Q to access your billing data. For an IAM policy that grants the required permissions, see [Allow Amazon Q to perform actions on your behalf in chat](#).
- `ce:GetCostAndUsage`, `ce:GetCostForecast`, `ce:GetTags`, `ce:GetCostCategories`, `ce:GetDimensionValues` to get Cost Explorer billing data.

## Enable AWS Cost Explorer

To chat about your costs with Amazon Q, you must enable AWS Cost Explorer in your AWS account. To enable Cost Explorer, open the Cost Explorer console. For more information, see [Enabling Cost Explorer in the AWS Cost Management User Guide](#).

## Example questions

Following are example questions about costs that you can ask Amazon Q:

- How much did we spend on SageMaker in January?
- What are the top contributing services to my AWS bill in the 'eu-central-1' region?
- What were my Amazon EC2 costs by instance type last week?
- What was my cost breakdown by service for the past three months?
- Which linked accounts increased their spend the most from January to February?
- Which linked accounts spent the most on DynamoDB last month?
- What were my cost trends by region over the last three months?

## Using Amazon Q Developer plugins

Amazon Q Developer 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 Amazon Q in the AWS console. Amazon Q calls the third party provider APIs to retrieve resources and generates a response with deep links to the external resources.

When Amazon Q 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 Amazon Q to retrieve credentials to connect to the third party provider.

Amazon Q doesn't share any information with third party providers when you configure or use plugins. For more information on how Amazon Q uses your data, see [Data protection](#).

Regardless of where you use plugins, your requests may be processed in the US East (N. Virginia) Region, the US West (Oregon), or the US East (Ohio) Region, even if the AWS Management Console is set to a different AWS Region.

### Warning

Third party provider user permissions are not detected by Amazon Q Developer 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 Amazon Q Developer.

### Topics

- [Configuring the Amazon Q Developer Datadog plugin](#)
- [Configuring the Amazon Q Developer Wiz plugin](#)

## Configuring the Amazon Q Developer 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 Amazon Q Developer chat 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 details about Datadog resources.

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

## Warning

Datadog user permissions are not detected by the Datadog plugin in Amazon Q. 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](#).
- 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 URL** – The URL of the Datadog site you use. For example, <https://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 Amazon Q 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, Amazon Q 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-application-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 in Amazon Q Developer, you need to create a service role that gives Amazon Q permission to access your Secrets Manager secret. Amazon Q assumes this role to access the secret where your Datadog credentials are stored.

When you configure the plugin in the AWS 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": [  
        "kms:Decrypt",  
        "kms:DescribeKey",  
        "kms:ListAliases",  
        "kms:ListKeys",  
        "kms:PutAlias",  
        "kms:PutKeyPolicy",  
        "kms:PutKeyRotationPolicy",  
        "kms:PutRetireEncryptionKey",  
        "kms:RetireEncryptionKey",  
        "kms:UpdateKeyDescription",  
        "kms:UpdateKeyPolicy",  
        "kms:UpdateKeyRotationPolicy"  
      ]  
    }  
  ]  
}
```

```
        "secretsmanager:GetSecretValue"
    ],
    "Resource": [
        "arn:aws:secretsmanager:{}{{your-region}}:{}{{your-account-id}}: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:{}{{region}}:{}{{accountId}}:secret:
{{secretId}}"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:{}{{region}}:{}{{accountId}}:key/{{keyId}}",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "secretsmanager.{{region}}.amazonaws.com"
                }
            }
        }
    ]
}
```

To allow Amazon Q 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](#).

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

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. Amazon Q 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 Amazon Q 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 Amazon Q 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**.

- 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 credentials and 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 Amazon Q in the console. For an example IAM policy that grants permissions needed to chat, see [Allow users to chat with Amazon Q](#).
- 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.

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 Amazon Q must also include **@datadog**.

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

- **Learn about using Datadog features in your AWS workload** – Ask about how Datadog features work with certain AWS services. Amazon Q 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 Amazon Q 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 Amazon Q Developer 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 Amazon Q chat 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 Amazon Q and Wiz. After you configure the plugin, you can access Wiz metrics by adding **@wiz** to the beginning of your question in Amazon Q chat.

### Warning

Wiz user permissions are not detected by the Wiz plugin in Amazon Q. 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](#).
- 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 Amazon Q 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, Amazon Q 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 in Amazon Q Developer, you need to create a service role that gives Amazon Q permission to access your Secrets Manager secret. Amazon Q assumes this role to access the secret where your Wiz credentials are stored.

When you configure the plugin in the AWS 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:{}{your-region}:{}{your-account-id}{}: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:{}{{region}}:{}{{accountId}}:secret:{}{{secretId}}"
    },
    {
        "Effect": "Allow",
        "Action": [
            "kms:Decrypt"
        ],
        "Resource": "arn:aws:kms:{}{{region}}:{}{{accountId}}:key/{{keyId}}",
        "Condition": {
            "StringEquals": {
                "kms:ViaService": "secretsmanager.{{region}}.amazonaws.com"
            }
        }
    }
]
```

To allow Amazon Q 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](#).

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

```
        "StringEquals": {
            "aws:SourceAccount": "{{accountId}}",
            "aws:SourceArn": "arn:aws:codewhisperer:{{region}}:{{accountId}}:profile/{{profileId}}"
        }
    }
}
]
```

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. Amazon Q 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 Amazon Q 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 Amazon Q 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 credentials and 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 Amazon Q in the console. For an example IAM policy that grants permissions needed to chat, see [Allow users to chat with Amazon Q](#).
- 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.

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 Amazon Q Wiz plugin, enter **@Wiz** at the beginning of a question about your Wiz issues. Follow up questions or responses to questions from Amazon Q must also include **@Wiz**.

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

- **View issues with critical severity** – Ask the Amazon Q 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?**

# Automating AWS services with Amazon Q Developer Console-to-Code

## What is Console-to-Code?

Console-to-Code is a feature of Amazon Q Developer that can help you write code to automate your use of other AWS services. Console-to-Code records your console actions, then uses generative AI to suggest code in your preferred language and format.

### Tiers of service

Since Console-to-Code is a part of Amazon Q Developer, your use of it is subject to Amazon Q Developer's tiers of service.

- At the Free tier, there is no fixed monthly limit to the number of times you can record your console actions and generate CLI commands based on those actions. However, there is a limit to how many times per month you can generate code to use with the AWS CDK or AWS CloudFormation based on your recorded actions.

To access the Free tier, sign into the AWS Management Console. After you reach the monthly code generations limit, you must authenticate to the Pro tier in order to generate more code.

- At the Pro tier, there is no fixed monthly limit to the number of times you can generate code for the AWS CDK or CloudFormation.

To access the Pro tier, you must be a user registered with IAM Identity Center, and your IAM Identity Center identity must be subscribed to Amazon Q Developer Pro. For more information, see [Authenticating to your Amazon Q Developer Pro subscription](#) or contact your AWS administrator.

For more information on pricing tiers, visit the [Amazon Q Developer pricing page](#).

#### Note

When you record an action, you will still be charged for the action itself, if applicable. For example, if you record yourself provisioning an Amazon EC2 instance, then you will still be charged for the instance. There is no additional cost for recording the action.

## Supported code formats

Console-to-Code can currently generate infrastructure-as-code (IaC) in the following languages and formats:

- CDK Java
- CDK Python
- CDK TypeScript
- CloudFormation JSON
- CloudFormation YAML

## Where can you use Console-to-Code?

### Using Console-to-Code across multiple services

Console-to-Code works across multiple services, saving its own state for as long as your browser tab is open.

For example, you may record your actions during a complete setup of a web server:

- In the VPC console, you provision two subnets (one public and one private), security groups, NACLs, a custom routing table, and an internet gateway.
- In the Amazon EC2 console, you provision an Amazon EC2 instance and place it in the public subnet.
- In the Amazon RDS console, you provision an Amazon RDS DB instance and place it in the private subnet.

Even if you perform your actions in different parts of the console and they use different AWS services, Console-to-Code can include them in a single recording.

### AWS services that support Console-to-Code

Currently, Console-to-Code is available to record your actions when using the AWS management console with the following services:

- Amazon EC2

- Amazon VPC
- Amazon RDS

## Granting permissions to use Console-to-Code

To use Console-to-Code, the following permissions are required:

- `q:GenerateCodeFromCommands` to use Console-to-Code. For an example IAM policy that grants the needed permission, see [Allow users to generate code from CLI commands with Amazon Q](#).
- Permissions to take the actions that you're going to record.

## Using Console-to-Code

Using Console-to-Code consists of three steps.

### Step 1: Start recording

To start recording with Console-to-Code, use the following procedure.

1. Go to the console of one of the integrated services (Amazon VPC, Amazon RDS, or Amazon EC2).
2. On the right edge of the browser window, choose the Console-to-Code icon.
3. In the Console-to-Code side panel, choose **Start recording**.

### Step 2: Take actions

In the consoles of any of the integrated services, proceed to take any actions that you want to record.

The Console-to-Code side panel retains its own state. You can move between the consoles of the integrated services, creating one recording that involves actions for multiple services.

The Console-to-Code side panel will retain your actions until your Console-to-Code session ends. The session will end when you close the browser tab, or when your AWS Management Console session ends, whichever comes first.

When you have finished taking actions that you want to convert to code, choose **Stop** from the top of the Console-to-Code panel.

## Step 3: Gather CLI commands and generating code

You can follow either Step 3a or Step 3b.

### Step 3a: Gather CLI commands

To use Console-to-Code to generate CLI commands based on your actions, use the following procedure.

1. In the Console-to-Code panel, review your recorded actions.

You can filter the recorded actions using the dropdown, search box, or filter widget at the top of the Console-to-Code panel.

2. At the top of the Console-to-Code panel, toggle the **Show CLI** setting. This will display the CLI command corresponding to each selected action.
3. Select the commands that you want to use. Only the commands with checked boxes will be used in the following steps.
4. Consolidate the CLI commands. Near the bottom of the Console-to-Code panel, choose the accordion icon (^). The CLI commands that you selected will appear by themselves in the panel.
5. Copy or download your chosen commands.

To learn more about the AWS CLI, see [What is the AWS Command Line Interface?](#) in the *AWS Command Line Interface User Guide*.

### Step 3b: Generate code

1. In the Console-to-Code panel, review your recorded actions. You can filter the recorded actions using the dropdown, search box, or filter widget at the top of the Console-to-Code panel.
2. Select the actions that you want to convert into code. Only the actions with checked boxes will be used in the following steps.
3. Indicate the type of code that you want to generate. From the reverse dropdown menu at the lower right of the Console-to-Code panel, select the language and (if applicable) format of the code to be generated.

#### 4. Choose **Generate chosen language**.

The generated code will appear, along with the equivalent CLI commands.

## Diagnosing common errors in the console with Amazon Q Developer

In the AWS Management Console, Amazon Q Developer can diagnose common errors you receive while working with AWS services, such as insufficient permissions, incorrect configuration, and exceeding service limits. This feature is available for errors that arise while using the following services in the AWS Management Console:

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

You can diagnose most common console errors with Amazon Q, except for simple validation errors. Amazon Q doesn't maintain a history of previous error diagnosing sessions.

If you're unable to diagnose your error with Amazon Q, you can use Amazon Q to create a support case with AWS Support. For more information, see [Using Amazon Q Developer to chat with AWS Support](#). If you have an issue specific to the Amazon Q error diagnosing feature, you can use the thumbs-down icon to report an issue.

 **Note**

Diagnosing console errors with Amazon Q is currently available in the US West (Oregon) and US East (N. Virginia) AWS Regions.

## Add permissions

For an IAM policy that grants permissions needed for diagnosing console errors, see [Allow users to diagnose console errors with Amazon Q](#).

## Diagnose common errors in the console

To use Amazon Q to diagnose an error in the AWS Management Console, use the following procedure.

1. If you receive an error that Amazon Q can help you with, a **Diagnose with Amazon Q** button appears in the error message. If you want to use Amazon Q to diagnose the error, choose **Diagnose with Amazon Q** to proceed.
2. A window appears where Amazon Q provides information about the error and an overview of how to resolve it. To see detailed steps for how to resolve the error, choose **Help me resolve**.
3. It can take several seconds for Amazon Q to generate instructions. After they appear, follow the instructions to resolve the error.
4. To provide feedback, you can use the thumbs-up and thumbs-down icons. To provide detailed feedback, choose the **Tell me more** button that appears after you select an icon.

## Using Amazon Q Developer to chat with AWS Support

You can use Amazon Q Developer to create a support case and contact AWS Support from anywhere in the AWS Management Console, including the AWS Support Center Console. Amazon Q uses the context of your conversation to draft a support case on your behalf automatically. It also adds your recent conversation to the support case description. After creating the case, Amazon Q can transfer you to a support agent in the method of your choice, including live chat in the same interface.

When you create a support case in Amazon Q, the case is also updated in the Support Center Console. To track updates on cases created with Amazon Q, use the Support Center Console.

The type of AWS Support available to you depends on the support plan for your AWS account. All AWS users have access to account and billing support as part of the Basic Support plan. For technical support questions, only users with support plans other than the Basic Support plan can contact AWS Support with Amazon Q. For more information about AWS Support, see [Getting started with AWS Support](#) in the *AWS Support User Guide*.

## Prerequisites

To create cases in Amazon Q, you must meet the following requirements:

- You have a support plan higher than the Basic Support plan. Only users with support plans other than the Basic Support plan can contact AWS Support with Amazon Q.
- You have permissions to chat with Amazon Q. For more information, see [Allow users to chat with Amazon Q](#).
- You have permissions to create AWS Support cases. For more information, see [Manage access to AWS Support Center](#).

## Specify the right service

When you create a support case with Amazon Q, it populates the service field based on your question. If Amazon Q chooses the wrong service, update the case with the correct service. If your question has to do with multiple services, specify the service that's most applicable.

To contact AWS Support about an Amazon Q feature that is part of another AWS service, create a support case for the other AWS service, not for Amazon Q. For example, if you're using Amazon Q network troubleshooting in Amazon VPC Reachability Analyzer, choose Amazon VPC for the service in the support case.

To contact AWS Support about features in either Amazon Q Developer or Amazon Q Business, create a support case for Amazon Q.

## Create a support case

To create an AWS Support case with Amazon Q, use the following steps.

1. You can create an AWS Support case through Amazon Q in one of two ways:
  - a. Ask for help directly by entering a question such as "I want to speak to someone" or "Get support".

To provide more context for Amazon Q to create the support case, you can add more information when requesting support directly. Following is an example of providing more information in a request:

"I am unable to connect to my bastion instance. I have tried restarting it and generating new key pairs but still nothing works. This started this morning after a planned deployment. I can confirm that no other network related changes were made. Can I talk to someone?"

- b. If an Amazon Q response didn't help you, choose the thumbs-down icon on the response and then choose a reason that you're providing the feedback. To contact AWS Support, choose **Create a support case**.

The following image shows the **Create a support case** button in the Amazon Q chat panel that appears after you leave feedback.

### [3] Troubleshoot instances with failed status checks - Amazon Elastic Compute Cloud



Thank you for your feedback. If you need further assistance related to this issue, you may contact support.

[Create a support case](#)

2. A support case appears in the chat panel. If you had a conversation with Amazon Q before requesting support, it will use the context of your conversation to autopopulate the fields in the case. To update any field in the support case, choose **Edit**. You can also attach files that help explain your issue.

If you didn't chat with Amazon Q before requesting support or Amazon Q otherwise can't complete the fields in the support case, you can input your support case information into the case manually.

The following image is an example of a filled-out support case in the Amazon Q chat panel.

Sure, I've drafted the following support case for you. Review details and make required changes before continuing. I will also add our recent conversation to the case description once submitted.

Support Level	<b>Enterprise Support</b>	<a href="#">Change </a>
Description	<b>The customer is unable to connect to their instance after a recent deployment. They have tried restarting the instance and generating new key pairs but are still unable to connect. No other changes were made to the network configuration. The issue is problematic and the customer would like to chat with support.</b>	
Case type	<b>Technical</b>	
Service	<b>Elastic Compute Cloud (EC2 - Linux)</b>	
Category	<b>Instance Issue</b>	
Severity	<b>General guidance</b>	
Additional Contacts	<b>None</b>	
Attachments	<a data-bbox="497 1024 677 1066" href="#"> Attach Files</a> You can attach up to 3 files. Each file can be up to 5 MB.	
<a data-bbox="236 1151 318 1182" href="#">Cancel</a> <a data-bbox="856 1151 922 1182" href="#">Edit</a> <a data-bbox="1003 1151 1101 1182" href="#">Submit</a>		

3. After confirming that the support case describes your needs, choose **Submit** to create the support case. If you no longer want to create the case, choose **Cancel**.
4. To contact AWS Support, choose the method that you want to use. Depending on your case details, you can chat, email, or request a phone call from a live support agent:
  - a. **Chat** – If you choose to chat with an agent, a live support agent will enter the conversation. To end the chat with the support agent, choose **End this chat** at any time during the chat.

If you refresh your page, navigate to a different console, or get signed out of the console because of session expiration, the conversation will end.

- If you minimize the chat panel or leave the page, you might miss notifications and be disconnected because of inactivity. We recommend that you keep the chat panel open throughout the duration of your support chat.
- b. **Email** – If you choose to send an email message to an agent, a support agent will contact you at the email address that's associated with your AWS account.
  - c. **Call** – If you choose to call an agent, enter your phone number when prompted, and choose **Submit**. You will be added to the call queue.
5. You can leave feedback or choose **Skip** to return to the Amazon Q chat panel.

## Leave feedback

After the support chat has ended, you can optionally leave feedback.

Rate your experience, enter any additional feedback, and then choose **Submit feedback**.

## Chatting with Amazon Q Developer in AWS Chatbot

You can chat with Amazon Q Developer in Microsoft Teams and Slack messaging platforms that are configured with AWS Chatbot. Amazon Q in AWS Chatbot can answer questions about best practices for building solutions, troubleshooting issues, and identifying next steps.

To add Amazon Q to a Microsoft Teams or Slack channel that is already configured with AWS Chatbot, complete the following steps. To set up AWS Chatbot in your channels for the first time and allow users to use Amazon Q, see [Get started with Microsoft Teams](#) and [Get started with Slack](#) in the *AWS Chatbot Administrator Guide*.

## Add Amazon Q to a chat channel

You can add Amazon Q to a Microsoft Teams or Slack channel that is already configured with AWS Chatbot. First, update your AWS Identity and Access Management (IAM) role settings to include the [AmazonQDeveloperAccess](#) managed policy, and then add the policy as a channel guardrail. If you need administrator access, add the [AmazonQFullAccess](#) policy instead.

1. Add the [AmazonQDeveloperAccess](#) managed policy to your IAM role:
  - a. Sign in to the AWS Management Console and open the [IAM console](#).
  - b. In the navigation pane of the IAM console, choose **Roles**.

- c. Choose the name of the role that you want to modify.
  - d. In **Permissions policies**, choose **Add permissions** and **Attach policies**.
  - e. Enter AmazonQDeveloperAccess in the search.
  - f. Select **AmazonQDeveloperAccess**.
  - g. Choose **Add permissions**.
2. Add the AmazonQDeveloperAccess managed policy to your channel guardrails:
    - a. Open the [AWS Chatbot console](#).
    - b. Choose a configured client.
    - c. Select a configured channel.
    - d. Choose **Set guardrails**.
    - e. Enter AmazonQDeveloperAccess in the search.
    - f. Select **AmazonQDeveloperAccess**.
    - g. Choose **Save**.

## Ask Amazon Q questions in your channel

To check that your configuration was successful, ask Amazon Q a question. Enter @aws followed by your question. For example, you can enter @aws what is fargate?

Following are some examples of questions that you can ask Amazon Q from your configured channel:

- @aws how do I troubleshoot lambda concurrency issues?
- @aws what are the best practices for securing S3 buckets?
- @aws what is the maximum zipped file size for a lambda?

# Using Amazon Q Developer in the IDE

Use Amazon Q Developer in integrated development environments (IDEs) to learn about AWS and get assistance with your software development needs. In IDEs, Amazon Q includes capabilities to provide guidance and support across various aspects of software development, such as answering questions about building on AWS, generating and updating code, security scanning, and optimizing and refactoring code.

To install Amazon Q in your IDE, see [Installing the Amazon Q Developer extension or plugin in your IDE](#).

## Topics

- [Supported IDEs and available features](#)
- [Installing the Amazon Q Developer extension or plugin in your IDE](#)
- [Chatting with Amazon Q Developer about code](#)
- [Generating inline suggestions with Amazon Q Developer](#)
- [Transforming code in the IDE with Amazon Q Developer](#)
- [Developing software with the Amazon Q Developer Agent for software development](#)
- [Scanning your code with Amazon Q](#)
- [Supported languages for Amazon Q Developer in the IDE](#)

## Supported IDEs and available features

The features you have access to depend on the IDE where you use Amazon Q. The following table describes the IDEs supported by Amazon Q and the availability and limitations of features in each IDE.

If no language support is specified, the IDE supports languages listed in the [Supported languages](#) topic.

IDE	Supported features
Eclipse IDEs (Preview)	<a href="#">Chat</a> <a href="#">Inline suggestions</a>

IDE	Supported features
	<a href="#">Customizations</a>
JetBrains IDEs	<a href="#">Chat</a> <a href="#">Inline chat</a> <a href="#">Workspace context in chat</a> <a href="#">Inline suggestions</a> <a href="#">Agent for code transformation</a> <a href="#">Agent for software development</a> <a href="#">Security scanning</a> <a href="#">Customizations</a>
Visual Studio Code	<a href="#">Chat</a> <a href="#">Inline chat</a> <a href="#">Workspace context in chat</a> <a href="#">Inline suggestions</a> <a href="#">Agent for code transformation</a> <a href="#">Agent for software development</a> <a href="#">Security scanning</a> <a href="#">Customizations</a>
Visual Studio	<a href="#">Chat</a> <a href="#">Inline suggestions – C, C++, and C#</a> <a href="#">Security scanning – C#</a>
AWS coding environments	<a href="#">Inline suggestions</a>

# Installing the Amazon Q Developer extension or plugin in your IDE

To set up Amazon Q Developer in your integrated development environment (IDE), complete the following steps. After installing the Amazon Q extension or plugin, authenticate through IAM Identity Center or AWS Builder ID. You can use Amazon Q for free, without an AWS account, by authenticating with Builder ID.

To get started, download the Amazon Q extension or plugin for your IDE:

- [Download Amazon Q for Eclipse \(Preview\)](#)
- [Download Amazon Q for Visual Studio Code](#)
- [Download Amazon Q for JetBrains IDEs](#)
- [Download Amazon Q in the AWS Toolkit for Visual Studio](#)

## Note

In general, the default duration for a session that is authenticated through IAM Identity Center is 8 hours. However, in the case of Amazon Q, the default session lasts 90 days (if you set up IAM Identity Center on April 18, 2024 or later). For more information refer to [How to extend the session duration for Amazon Q in the IDE](#) in the *IAM Identity Center User Guide*.

To sign in and authenticate, complete the steps in this section.

## Steps

- [Prerequisite: Choose a supported version of your IDE](#)
- [Authenticating in Eclipse IDEs \(Preview\)](#)
- [Authenticating in JetBrains IDEs](#)
- [Authenticating in Visual Studio Code](#)
- [Authenticating in Visual Studio](#)
- [Using an IAM principal in your AWS console](#)

## Prerequisite: Choose a supported version of your IDE

- The minimum version of Eclipse supported by Amazon Q is 2024-06 (4.32).
- The minimum version of JetBrains IDEs (including IntelliJ and PyCharm) supported by Amazon Q is 232.1. JetBrains versions 232, 233 and 241 are also supported.
- The minimum version of Visual Studio Code supported by Amazon Q is 1.85.0.
- Only Visual Studio for Windows is supported by Amazon Q. The minimum version of Visual Studio supported is Visual Studio 2022 version 17.7. All Visual Studio 2022 editions are supported.

## Authenticating in Eclipse IDEs (Preview)

You can authenticate for free with AWS Builder ID or with IAM Identity Center with a Amazon Q Developer Pro subscription. Choose your authentication method to see steps to start using Amazon Q in Eclipse.

### Builder ID

This procedure does not require you to have Builder ID. If you have not yet signed up for Builder ID, you will have the opportunity to do so during the sign-in process.

1. Install the [Amazon Q plugin](#) in Eclipse.
2. Choose the Amazon Q icon in the top right corner of the IDE.
3. An Amazon Q tab opens at the bottom of the IDE. Under **Choose a sign-in option**, choose **Use for free**, and then choose **Continue**. You are redirected to your browser.
4. Follow the instructions in your browser to authenticate with Builder ID. When you've completed authentication, return to the Eclipse IDE.
5. To begin using Amazon Q, choose the Amazon Q icon to open the chat Amazon Q panel.

### Amazon Q Developer Pro license

Before you begin this procedure, your administrator should have:

- Created an identity for you in IAM Identity Center
- Subscribed that identity to Amazon Q Developer Pro

After your identity has been subscribed to Amazon Q Developer Pro, complete the following steps to authenticate:

1. Install the [Amazon Q plugin](#) in Eclipse.
2. Choose the Amazon Q icon in the top right corner of the IDE.
3. An Amazon Q tab opens at the bottom of the IDE. Under **Choose a sign-in option**, choose **Use with Pro license**, and then choose **Continue**.
4. Enter the **Start URL** that your administrator got from [the Amazon Q subscription console](#).
5. Choose the AWS Region in which your administrator set up your [IAM Identity Center instance](#).
6. Choose **Continue**. You are redirected to your browser.
7. Follow the instructions in your browser to authenticate with IAM Identity Center. When you've completed authentication, return to the Eclipse IDE.
8. To begin using Amazon Q, choose the Amazon Q icon to open the chat Amazon Q panel.

## Authenticating in JetBrains IDEs

You can authenticate for free with AWS Builder ID or with IAM Identity Center with a Amazon Q Developer Pro subscription. Choose your authentication method to see steps to start using Amazon Q in your JetBrains IDE.

### Builder ID

This procedure does not require you to have Builder ID. If you have not yet signed up for Builder ID, you will have the opportunity to do so during the sign-in process.

1. Install the [Amazon Q plugin](#) in your JetBrains IDE.
2. Choose the Amazon Q icon in your IDE.

The icon will be on the side of the interface by default.

3. Follow the instructions in your browser to authenticate with Builder ID.
4. To begin using Amazon Q, choose the Amazon Q icon to chat with Amazon Q, or choose **Amazon Q** from the navigation bar at the bottom of your IDE.

## Amazon Q Developer Pro license

Before you begin this procedure, your administrator should have:

- Created an identity for you in IAM Identity Center
- Subscribed that identity to Amazon Q Developer Pro

After your identity has been subscribed to Amazon Q Developer Pro, complete the following steps to authenticate:

1. Install the [Amazon Q plugin](#) in your JetBrains IDE.
2. Choose the Amazon Q icon in your IDE.

The icon will be on the side of the interface by default.

3. Choose **Use with Pro license**.
4. Fill in the **Start URL** that your administrator got from [the Amazon Q subscription console](#).
5. Fill in the AWS Region in which your administrator set up your IAM Identity Center [instance](#).
6. Choose **Continue**. The focus will switch to your web browser.
7. Follow the instructions in your browser to authenticate with IAM Identity Center.
8. To begin using Amazon Q, choose the Amazon Q icon to chat with Amazon Q, or choose **Amazon Q** from the navigation bar at the bottom of your IDE.

## Authenticating in Visual Studio Code

You can authenticate for free with AWS Builder ID or with IAM Identity Center with a Amazon Q Developer Pro subscription. Choose your authentication method to see steps to start using Amazon Q in VS Code.

### Builder ID

This procedure does not require you to have Builder ID. If you have not yet signed up for Builder ID, you will have the opportunity to do so during the sign-in process.

1. Install the [Amazon Q extension](#) in VS Code.
2. Choose the Amazon Q icon in your IDE.

The icon will be on the side of the interface by default.

3. Follow the instructions in your browser to authenticate with Builder ID.
4. To begin using Amazon Q, choose the Amazon Q icon to chat with Amazon Q, or choose **Amazon Q** from the navigation bar at the bottom of your IDE.

## Amazon Q Developer Pro license

Before you begin this procedure, your administrator should have:

- Created an identity for you in IAM Identity Center
- Subscribed that identity to Amazon Q Developer Pro

After your identity has been subscribed to Amazon Q Developer Pro, complete the following steps to authenticate:

1. Install the [Amazon Q extension](#) in VS Code.
2. Choose the Amazon Q icon in your IDE.

The icon will be on the side of the interface by default.

3. Choose **Use with Pro license**.
4. Fill in the **Start URL** that your administrator got from [the Amazon Q subscription console](#).
5. Fill in the AWS Region in which your administrator set up your IAM Identity Center [instance](#).
6. Choose **Continue**. The focus will switch to your web browser.
7. Follow the instructions in your browser to authenticate with IAM Identity Center.
8. To begin using Amazon Q, choose the Amazon Q icon to chat with Amazon Q, or choose **Amazon Q** from the navigation bar at the bottom of your IDE.

## Authenticating in Visual Studio

To connect to your AWS accounts from the Toolkit for Visual Studio, open the **Getting Started with the AWS Toolkit** User Interface (connection UI) by completing the following procedure.

1. From the Visual Studio main menu, expand **Extensions** then expand the **AWS Toolkit**.

2. From the **AWS Toolkit** menu options choose **Getting Started**.
3. The **Getting Started with the AWS Toolkit** connection UI opens in Visual Studio.

You can authenticate for free with AWS Builder ID or with IAM Identity Center with a Amazon Q Developer Pro subscription. Choose your authentication method to see steps to start using Amazon Q in Visual Studio.

## Builder ID

1. From the **Getting Started with the AWS Toolkit** connection UI, select the **Amazon Q Developer** radial to expand the Amazon Q Developer authentication options.
2. From the **I'm using Amazon Q Developer on my own** section, choose the **Sign up or Sign in** button to open the **Log in with AWS Builder ID** dialog.
3. Choose the **Proceed to Browser** button to open the **AWS Authorize request** site in your default web browser.
4. Follow the prompts in your default web browser. You're notified when the authorization process is complete, and it's safe to close your browser and return to Visual Studio.

## Amazon Q Developer Pro license

1. From the **Getting Started with the AWS Toolkit** connection UI, select the **Amazon Q Developer** radial to expand the Amazon Q Developer authentication options.

 **Note**

If no stored credentials exist, proceed to **Step 3** to add or update your IAM Identity Center credentials.

2. From the **My organization has enabled Amazon Q Developer** section, expand the **Choose from an existing Profile or add new** drop-down menu to choose from your list of stored credentials.
3. From the **Profile Type** drop-down menu, choose **AWS IAM Identity Center**.
4. In the **Profile Name** text field, enter the **Profile Name** of the IAM Identity Center profile you want to authenticate with.
5. In the **Start URL** text field, enter the **Start URL** that's attached to your IAM Identity Center credentials.

6. From the **Profile Region (defaults to us-east-1)** drop-down menu, choose the **Profile Region** that's defined by the IAM Identity Center user profile you're authenticating with.
7. From the **SSO Region (defaults to us-east-1)** drop-down menu, choose the **SSO Region** that's defined by your IAM Identity Center credentials, then choose the **Connect** button to open the **Log in with AWS IAM Identity Center** dialog.
8. From the **Log in with AWS IAM Identity Center** dialog, choose the **Proceed to Browser** button to open the **AWS Authorize request** site in your default web browser.
9. Follow the prompts in your default web browser. You're notified when the authorization process is complete, and it's safe to close your browser and return to Visual Studio.

For more information about authenticating in the Toolkit for Visual Studio, see [Getting Started](#) in the *AWS Toolkit for Visual Studio User Guide*.

## Using an IAM principal in your AWS console

Depending on how you use AWS, you may be accustomed to using your IAM credentials to sign in to the console for all AWS services. However, you cannot use Amazon Q Developer in the IDE as an IAM principal, or with an IAM role. You must authenticate with credentials from either IAM Identity Center or Builder ID.

## Chatting with Amazon Q Developer about code

Chat with Amazon Q Developer in your integrated development environment (IDE) to ask questions about building at AWS and for assistance with software development. Amazon Q can explain coding concepts and code snippets, generate code and unit tests, and improve code, including debugging or refactoring.

### Topics

- [Working with Amazon Q in your IDE](#)
- [Example topics and questions](#)
- [Reporting issues with responses from Amazon Q](#)
- [Explaining and updating code with Amazon Q Developer](#)
- [Chatting inline with Amazon Q Developer](#)
- [Adding workspace context to Amazon Q Developer chat in the IDE](#)

## Working with Amazon Q in your IDE

You can ask Amazon Q questions, update your code, and initiate actions with quick commands all from the Amazon Q chat panel in your IDE.

To start chatting with Amazon Q, choose the Amazon Q icon from the navigation bar in your IDE and enter your question in the text bar. To start chatting with Amazon Q in Visual Studio, choose **View** from the main menu and then choose **Amazon Q chat**.

When you ask Amazon Q a question, it uses the current file that is open in your IDE as context, including the programming language and the file path.

If Amazon Q includes code in its response, you can copy the code or insert it directly into your file by choosing **Insert at cursor**. Amazon Q might include inline references to its sources in its response. To view a list of sources, expand the **Sources** section at the bottom of a response.

Amazon Q maintains the context of your conversation within a given session for future responses. You can ask follow up questions or refer to previous questions and responses throughout the duration of your session. To start a new conversation with Amazon Q, open a new tab in the panel. You can open up to 10 tabs at a time. Amazon Q doesn't retain context across different conversations.

### Chat commands

You can enter the following commands in the chat panel to access Amazon Q features, depending on your IDE. For information about what Amazon Q features are available in your IDE, see [Supported IDEs](#).

- **/transform** - Use this command to update the code language version of an entire project. For more information, see [Upgrading Java versions](#).
- **/dev** - Use this command to get an implementation plan to develop a feature with Amazon Q. For more information, see [Developing software](#).
- **/clear** - Use this command to clear a current conversation. This removes all previous conversation from the chat panel and clears the context that Amazon Q has about your previous conversation.
- **/help** - Use this command to see an overview of what Amazon Q can and can't do, example questions, and quick commands.

## Example topics and questions

Within IDEs, Amazon Q can answer questions about AWS services and software development, in addition to generating code. Amazon Q is particularly useful for answering questions related to the following subject areas:

- Building on AWS, including AWS service selection, limits, and best practices
- General software development concepts, including programming language syntax and application development
- Writing code, including explaining code, debugging code, and writing unit tests

Following are some example questions that you can ask to get the most out of Amazon Q in your IDE:

- How do I debug issues with my Lambda functions locally before deploying to AWS?
- How do I choose between AWS Lambda and Amazon EC2 for a scalable web application backend?
- What is the syntax of declaring a variable in TypeScript?
- How do I write an app in React?
- Provide me a description of what this [*selected code or application*] does and how it works.
- Generate test cases for [*selected code or function*].

## Reporting issues with responses from Amazon Q

You can optionally leave feedback for every response Amazon Q generates by using the thumbs-up and thumbs-down icons. To report an issue with a response, choose the thumbs-down icon, and enter information in the feedback window that appears.

## Explaining and updating code with Amazon Q Developer

Amazon Q Developer can explain and update specific lines of code in your integrated development environment (IDE). To update your code, ask Amazon Q to make changes to a given line or block of code, and it will generate new code that reflects the changes that you asked it to make. Then, you can insert the updated code directly into the file where the code originated.

You can choose from the following options:

- **Explain** – Get your code explained in natural language.
- **Refactor** – Improve code readability or efficiency, among other improvements.
- **Fix** – Debug code.
- **Optimize** – Enhance code performance.
- **Send to prompt** – Send the highlighted code to the Amazon Q chat panel, and ask questions that you have about the code.

## Send code to Amazon Q

To get your code explained or updated by Amazon Q, complete the following steps.

1. Highlight a section of a code file in your IDE.
2. Right-click your highlighted code to open a context window. Choose **Amazon Q**, and then choose **Explain**, **Refactor**, **Fix**, **Optimize**, or **Send to prompt**.

If you choose **Send to prompt**, Amazon Q copies the highlighted code to the chat panel, where you can enter questions that you have about the code.

3. To replace the highlighted code with the newly generated code, you can copy the code or insert it directly into your file by choosing **Insert code**. Amazon Q replaces the original code with the updated code.

## Chatting inline with Amazon Q Developer

The *inline chat* feature lets you transform existing code—or generate new code—right from your IDE's main coding window. To use the inline chat feature, you highlight code that you want suggestions for, and provide instructions in the small input screen. Amazon Q proceeds to generate code for you, which it presents in a diff within the main coding window. You can then choose to accept or reject the changes.

The advantage of inline chat is that it eliminates the context switching that occurs when moving between a chat window and the main coding window.

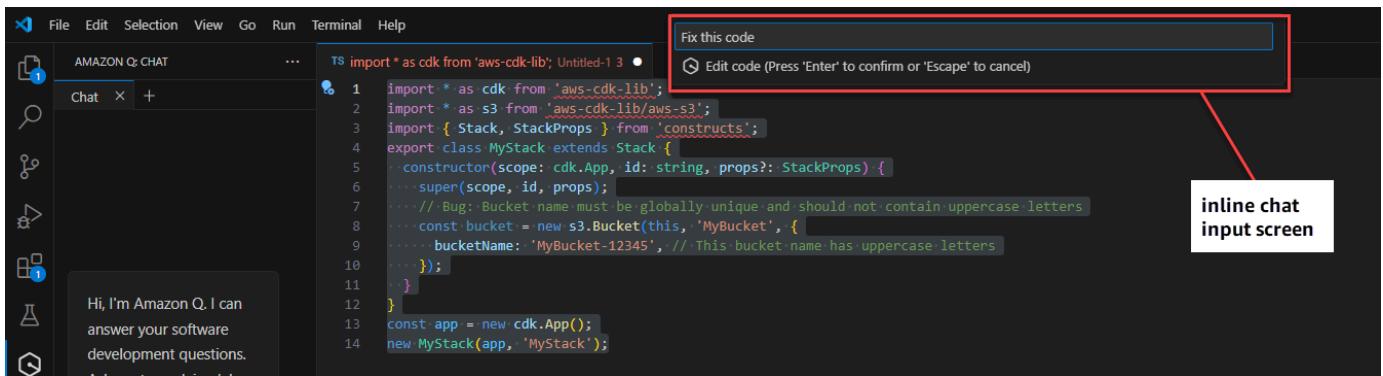
You would typically use the inline chat feature when you're reviewing code, writing unit tests, or performing other tasks that require code-based answers. For situations where you want text-based answers (for example, an answer to "Explain this code") then using the [chat window](#) is a better option.

Amazon Q considers the code in the current file when generating a code recommendation through the inline chat. It won't look at code in other files or projects.

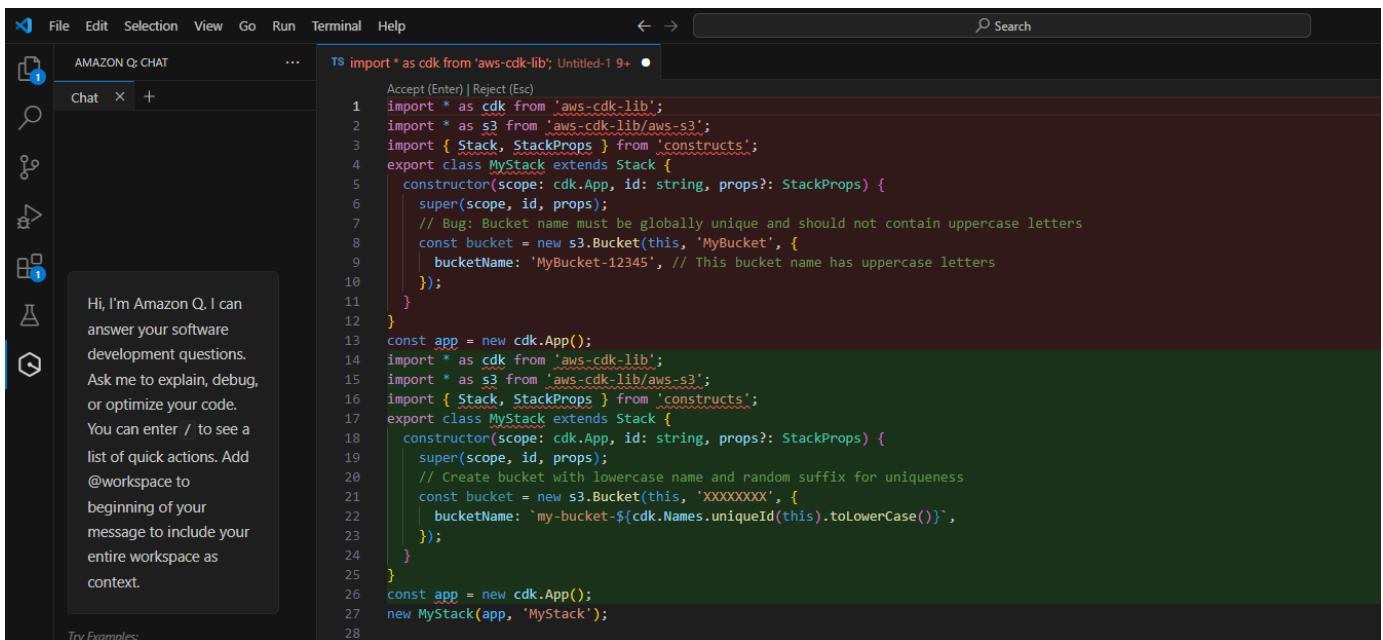
## Amazon Q inline chat in action

An inline chat session unfolds as follows.

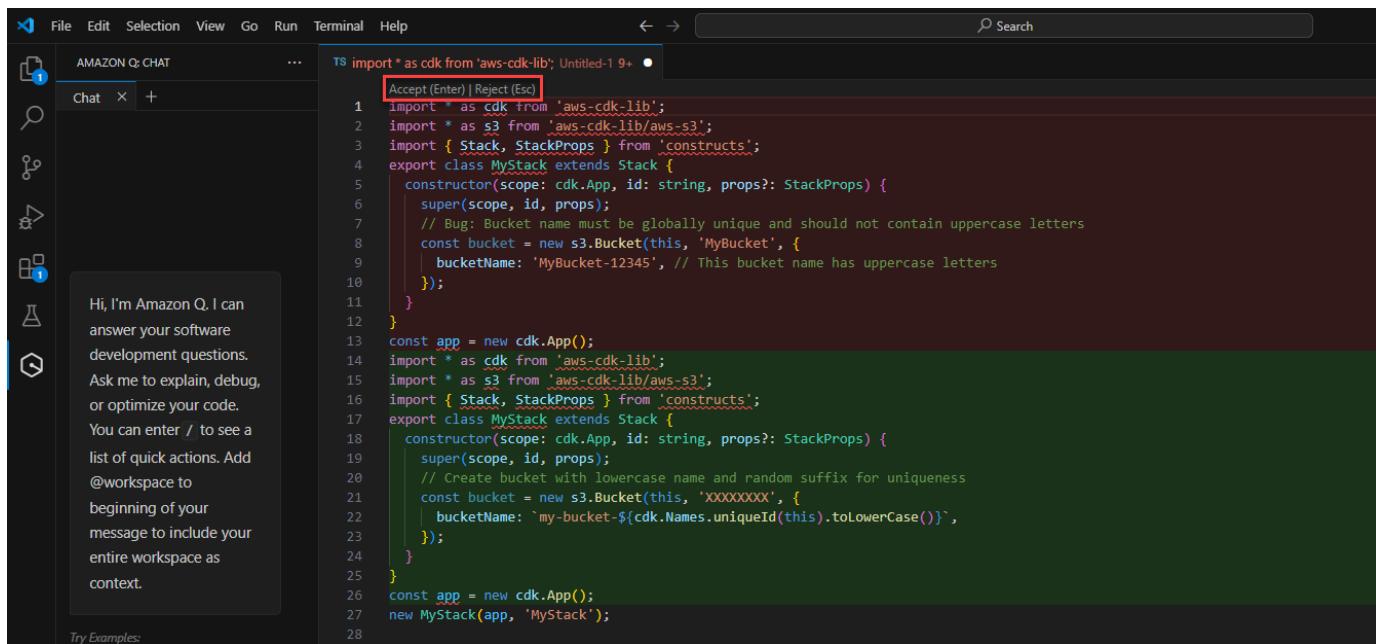
1. You highlight the code that you want suggestions for and press #+I (Mac) or Ctrl+I (Windows). Alternatively, you can right-click the selection and choose **Amazon Q, Inline chat**. This launches a small input screen at the top of the main coding window where you can enter a question like, **Fix this code**.



2. Amazon Q generates code and presents it in a diff.



3. You accept or reject the change by choosing **Accept** or **Reject**, or by pressing the keyboard equivalents (Enter or Esc).



The screenshot shows the Amazon Q Developer interface within a code editor. On the left, there's a sidebar with various icons. In the center, a dark-themed code editor window displays a TypeScript file named 'Untitled-1'. The file contains code related to AWS CDK and S3 buckets. A floating 'CHAT' window is overlaid on the code editor, containing a message from Amazon Q: "Hi, I'm Amazon Q. I can answer your software development questions. Ask me to explain, debug, or optimize your code. You can enter / to see a list of quick actions. Add @workspace to beginning of your message to include your entire workspace as context." At the bottom of the CHAT window, there's a link "Try Examples". The top of the screen has a navigation bar with File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar.

```
TS import * as cdk from 'aws-cdk-lib'; Untitled-1 9+ ●
Accept (Enter) | Reject (Esc)
1 import * as cdk from 'aws-cdk-lib';
2 import * as s3 from 'aws-cdk-lib/aws-s3';
3 import { Stack, StackProps } from 'constructs';
4 export class MyStack extends Stack {
5   constructor(scope: cdk.App, id: string, props?: StackProps) {
6     super(scope, id, props);
7     // Bug: Bucket name must be globally unique and should not contain uppercase letters
8     const bucket = new s3.Bucket(this, 'MyBucket', {
9       bucketName: 'MyBucket-12345', // This bucket name has uppercase letters
10    });
11  }
12  const app = new cdk.App();
13  import * as cdk from 'aws-cdk-lib';
14  import * as s3 from 'aws-cdk-lib/aws-s3';
15  import { Stack, StackProps } from 'constructs';
16  export class MyStack extends Stack {
17    constructor(scope: cdk.App, id: string, props?: StackProps) {
18      super(scope, id, props);
19      // Create bucket with lowercase name and random suffix for uniqueness
20      const bucket = new s3.Bucket(this, 'xxxxxxxx', {
21        bucketName: `my-bucket-${cdk.Names.uniqueId(this).toLowerCase()}`,
22      });
23    }
24  }
25  const app = new cdk.App();
26  new MyStack(app, 'MyStack');
```

## Example topics and questions

The inline chat always returns code as the answer, so you can enter prompts like:

- Document this code
- Refactor this code
- Write unit tests for this function

## Diff format

The inline chat displays the diff in multiple blocks, with the existing code on the top, and the suggested code on the bottom. A side-by-side diff is not supported.

## Adding workspace context to Amazon Q Developer chat in the IDE

When you chat with Amazon Q in the integrated development environment (IDE), you can add **@workspace** to your question to automatically include the most relevant chunks of your workspace code as context. Amazon Q Developer determines relevance based on an index that is updated periodically.

With workspace context, Amazon Q has enhanced capabilities, including locating files, understanding how code is used across files, and generating code that leverages multiple files, including files that aren't opened.

## Topics

- [Setup](#)
- [Ask questions with workspace context](#)

## Setup

Before you continue, make sure you have the latest version of your IDE installed. You can then complete the following setup steps.

### Enable indexing

To use your workspace as context, Amazon Q creates a local index of your workspace repository, including code files, configuration files, and project structure. During indexing, non-essential files like binaries or those specified in `.gitignore` files are filtered out.

It can take 5 to 20 minutes to index a new workspace. During this time, you can expect elevated CPU usage in your IDE. After initial indexing, the index is incrementally updated when you make changes to your workspace.

The first time you add workspace context, you must enable indexing in your IDE. Complete the following steps to enable indexing:

1. Add **@workspace** to your question in the Amazon Q chat panel.
2. Amazon Q prompts you to enable indexing. Choose **Settings** to be redirected to Amazon Q settings in your IDE. You can also choose **Open Settings** from the Amazon Q task bar.
3. Select the box under **Amazon Q: Local Workspace Index**.

### Configure indexing (optional)

No configuration is necessary for the indexing process, however you can choose to specify the number of threads dedicated to indexing. If you increase the number of threads used, indexing will complete faster, and it will use more of your CPU. To update the indexing configuration, specify the number of threads in Amazon Q settings under **Amazon Q: Local Workspace Index Threads**.

## Ask questions with workspace context

To add your workspace as context to your conversation with Amazon Q, open the workspace you want to ask questions about, and then add **@workspace** to your question in the chat panel. You must add **@workspace** to any question that you want to add workspace context to.

If you want to start chatting about a different workspace, open the workspace, and then open a new chat tab. Include `@workspace` in your question to add the new workspace as context.

You can ask Amazon Q about any file in your workspace, including unopened files. Amazon Q can explain files, locate code, and generate code across files, in addition to existing conversational coding capabilities.

Following are example questions you can ask Amazon Q that leverage workspace context in the chat:

- `@workspace where is the code that handles authorization?`
- `@workspace what are the key classes with application logic in this project?`
- `@workspace explain main.py`
- `@workspace add auth to this project`
- `@workspace what third-party libraries or packages are used in this project, and for what purpose?`
- `@workspace add unit tests for function <function name>`

## Generating inline suggestions with Amazon Q Developer

Amazon Q can provide you with code recommendations in real time. As you write code, Amazon Q automatically generates suggestions based on your existing code and comments. Your personalized recommendations can vary in size and scope, ranging from a single line comment to fully formed functions.

When you start typing out single lines of code or comments, Amazon Q makes suggestions based on your current and previous inputs. Filenames are also taken into consideration.

Inline suggestions are automatically enabled when you download the Amazon Q extension. To get started, start writing code, and Amazon Q will begin generating code suggestions.

You can also customize the suggestions Amazon Q generates to your software development team's internal libraries, proprietary algorithmic techniques, and enterprise code style. For more information on customizing suggestions, see [Customizing suggestions](#).

### Topics

- [Pausing suggestions with Amazon Q](#)

- [Amazon Q code completion in action](#)
- [Generating inline suggestions in AWS coding environments](#)
- [Using shortcut keys](#)
- [Using code references](#)
- [Code examples](#)

## Pausing suggestions with Amazon Q

Choose your IDE to see steps for pausing and resuming inline code suggestions in Amazon Q.

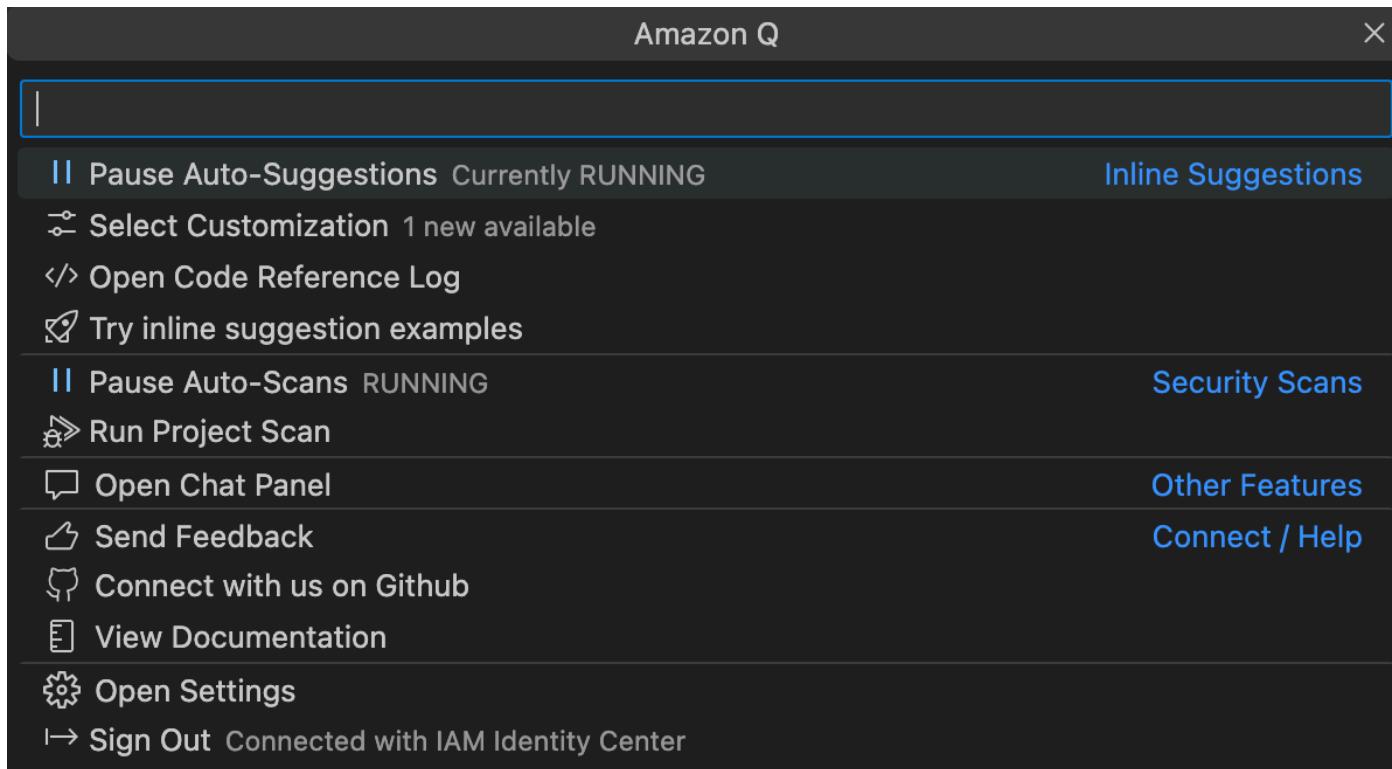
### Visual Studio Code

1. In VS Code, choose **Amazon Q** from the component tray at the bottom of the IDE window.

The Amazon Q task bar opens at the top of the IDE window.

2. Choose **Pause Auto-Suggestions** or **Resume Auto-Suggestions**.

The following image shows the Amazon Q task bar in VS Code.



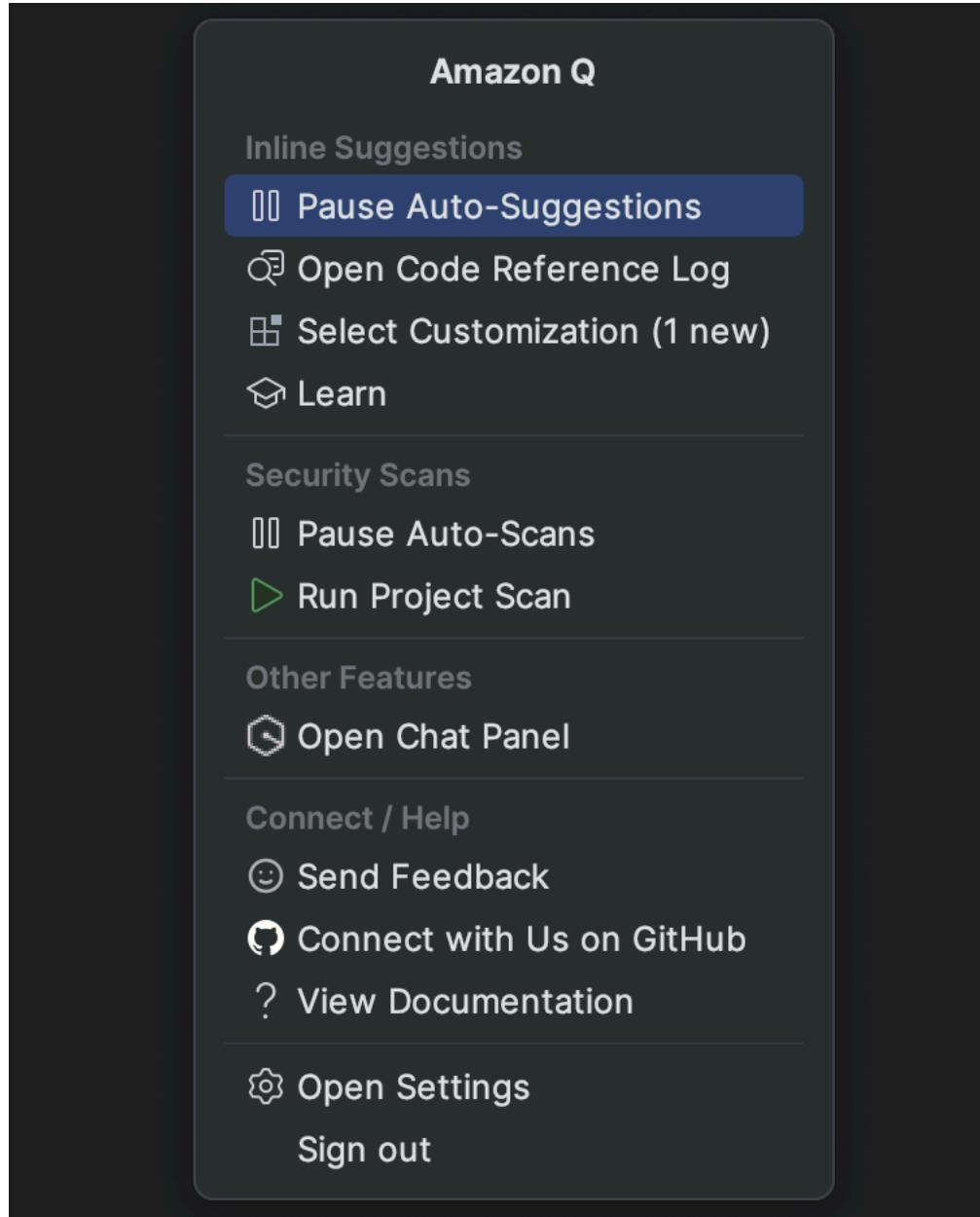
## JetBrains

1. In your JetBrains IDE, choose **Amazon Q** from the status bar at the bottom of the IDE window.

The Amazon Q task bar opens above the status bar.

2. Choose **Pause Auto-Suggestions** or **Resume Auto-Suggestions**.

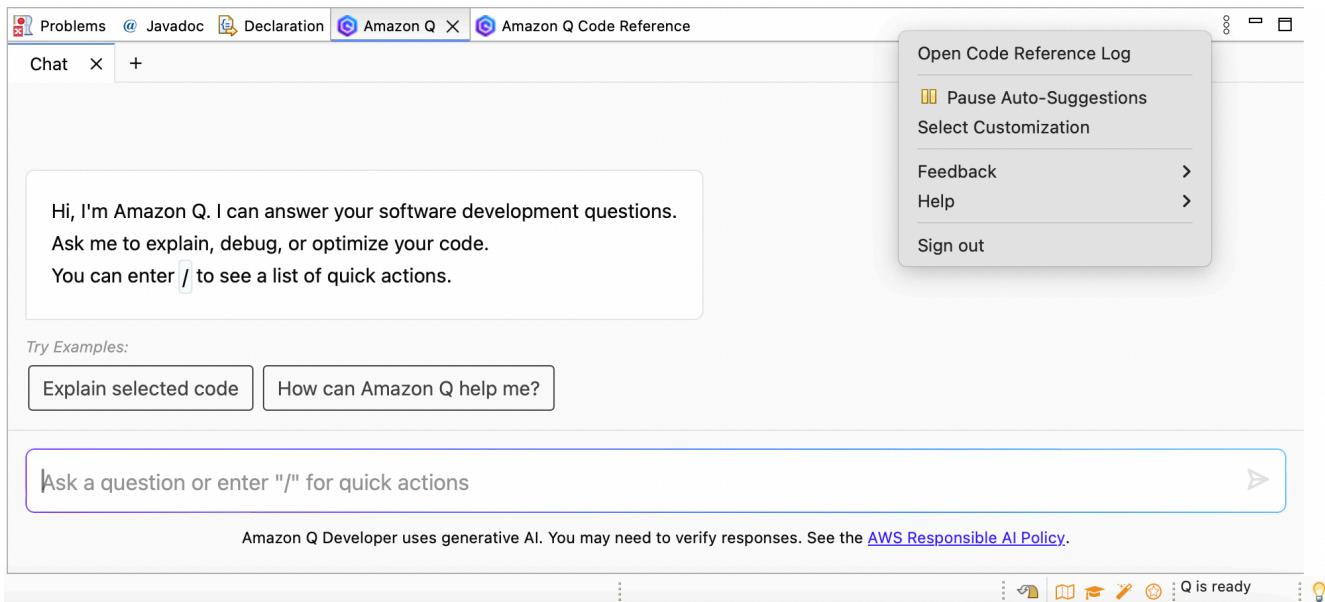
The following image shows the Amazon Q task bar in a JetBrains IDE.



## Eclipse

1. In your Eclipse IDE, choose the **Amazon Q** icon in the top right corner of the IDE.
2. With the Amazon Q chat tab open, choose the ellipsis icon in the top right corner of the tab. The Amazon Q task bar opens.

The following image shows the Amazon Q task bar in an Eclipse IDE.

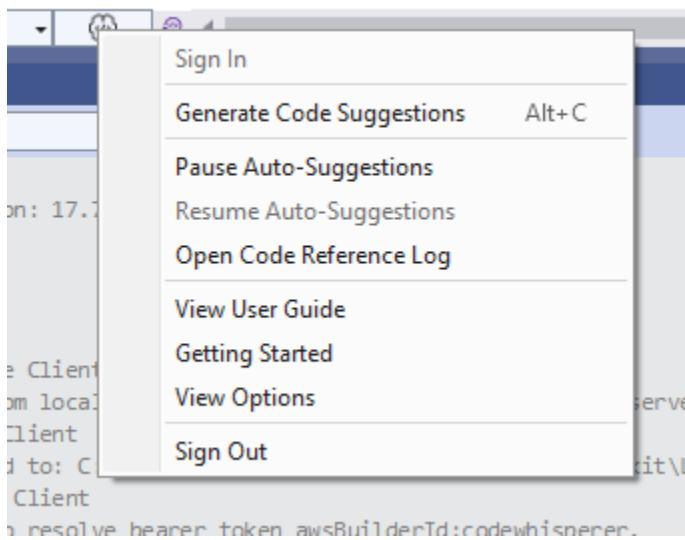


3. Choose **Pause Auto-Suggestions** or **Resume Auto-Suggestions**.

## Visual Studio

1. From the edge of the window, choose the Amazon Q icon.
2. Select **Pause Auto-Suggestions** or **Resume Auto-Suggestions**

The following image shows the Amazon Q task bar in a Visual Studio.



## AWS Cloud9

Amazon Q does not support toggling suggestions on and off in AWS Cloud9.

To stop receiving Amazon Q suggestions in AWS Cloud9, remove the IAM policy that gives Amazon Q access to AWS Cloud9 from the role or user that you are using to access AWS Cloud9.

## AWS Lambda

To deactivate or re-activate Amazon Q code suggestions in Lambda:

1. In the Lambda console, open the screen for a particular Lambda function.
2. In the **Code source** section, from the toolbar, choose **Tools**.
3. From the dropdown menu, choose **Amazon Q Code Suggestions**.

## Amazon SageMaker Studio

1. In the SageMaker Studio console, choose Amazon Q from the bottom of the window.

The Amazon Q panel will open.

2. Choose **Pause Auto-Suggestions** or **Resume Auto-Suggestions**.

## JupyterLab

1. In the JupyterLab console, choose Amazon Q from the bottom of the window.

The Amazon Q panel will open.

2. Choose **Pause Auto-Suggestions** or **Resume Auto-Suggestions**.

## AWS Glue Studio Notebook

1. In the AWS Glue Studio Notebook console, choose Amazon Q from the bottom of the window.

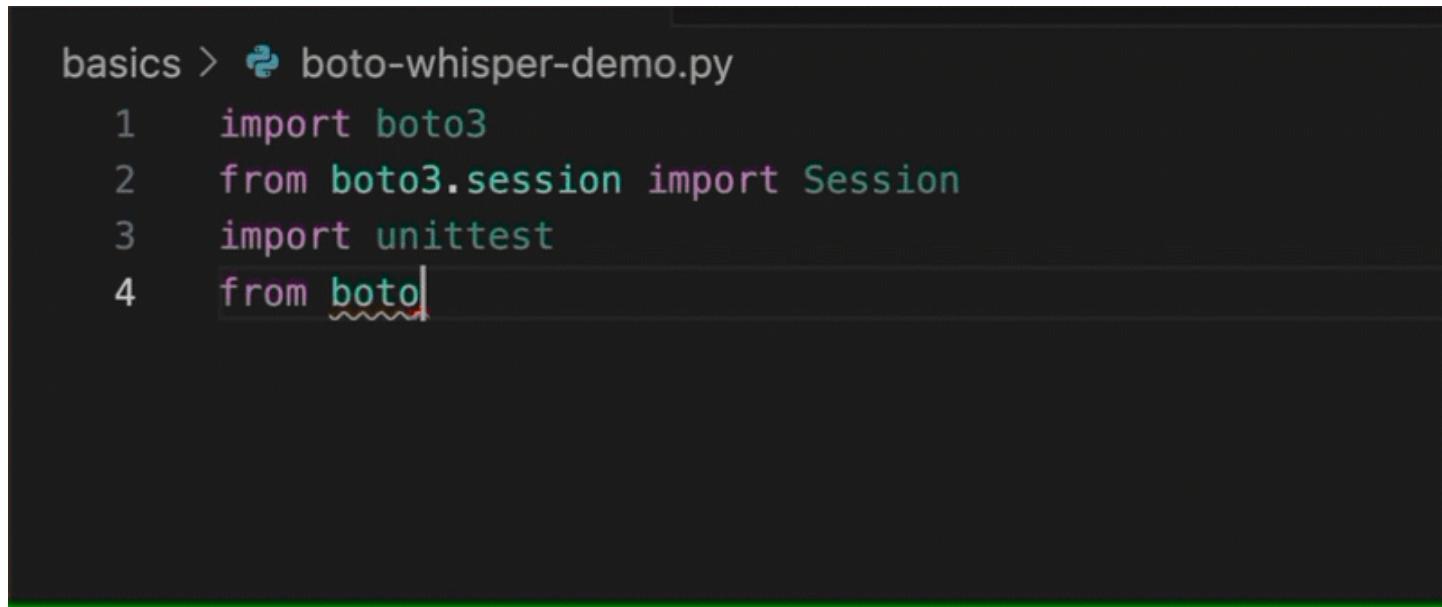
The Amazon Q panel will open.

2. Choose **Pause Auto-Suggestions** or **Resume Auto-Suggestions**.

## Amazon Q code completion in action

This section demonstrates how Amazon Q can help you write a complete application. This application creates an Amazon S3 bucket and a Amazon DynamoDB table, plus a unit test that validates both tasks.

Here, Amazon Q helps the developer choose which libraries to import. Using the arrow keys, the developer toggles through multiple suggestions.



```
basics > 🐍 boto-whisper-demo.py
1 import boto3
2 from boto3.session import Session
3 import unittest
4 from boto
```

Here, the developer enters a comment, describing the code they intend to write on the next line.

Amazon Q correctly anticipates the method to be called. The developer can accept the suggestion with the tab key.

```
basics > 🐍 boto-whisper-demo.py  
1 import boto3  
2 from boto3.session import Session  
3 import unittest  
4 from botocore.exceptions import ClientError  
5 import logging  
6 import time  
7  
8 # set up logging  
9 logging.basicConfig(level=logging.INFO)
```

Here, the developer prepares to define constants.

Amazon Q correctly anticipates that the first constant will be REGION and that its value will be us-east-1, which is the default.

```
basics > 🐍 boto-whisper-demo.py > ...  
8 # set up logging  
9 logging.basicConfig(level=logging.INFO)  
10  
11 #Create a new session  
12 session = Session()  
13  
14 # define constants  
15 DEFAULTREGION = 'us-east-1'
```

Here, the developer prepares to write code that will open sessions between the user and both Amazon S3 and DynamoDB.

Amazon Q, familiar with AWS APIs and SDKs, suggests the correct format.

```
8 # set up logging
9 logging.basicConfig(level=logging.INFO)
10
11 #Create a new session
12 session = Session()
13
14 # define constants
15 DEFAULT_REGION = 'us-east-1'
16 TEST_BUCKET_NAME = 'my-test-bucket' + str(int(time.time()))
17 TEST_TABLE_NAME = 'my-test-table' + str(int(time.time()))
18
19 # AWS Clients with session
20 s3 = session.client('s3', region_name=DEFAULT_REGION)
    dynamodb = session.client('dynamodb', region_name=DEFAULT_REGION)
```

The developer has merely written the name of the function that will create the bucket. But based on that (and the context), Amazon Q offers a full function, complete with try/except clauses.

Notice the use of TEST\_BUCKET\_NAME, which is a constant declared earlier in the same file.

```
18
19 # AWS Clients with session
20 s3_client = session.client('s3', region_name=us-east-1)
21 dynamodb_client = session.client('dynamodb', region_name=us-east-1)
22
23 def create_s3_bucket():
    """
    Creates a new S3 bucket
    """
    try:
        s3_client.create_bucket(Bucket=TEST_BUCKET_NAME)
    except ClientError as e:
        logging.error(e)
        return False
    return True
```

The developer has only just begun to type in the name of the function that will create a DynamoDB table. But Amazon Q can tell where this is going.

Notice that the suggestion accounts for the DynamoDB session created earlier, and even mentions it in a comment.

```
40  def create_dynamodb_table(table_name, region=None):
    # global dynamodb # Use the global dynamodb client created with the session
    print(f"Using region: {region}")
    print(f"DynamoDB endpoint URL: {dynamodb.meta.endpoint_url}") # Print the end
    try:
        print(f"Creating table in region: {region}") # Add this line to debug
        if region is None or region.lower() == 'us-east-1':
            response = dynamodb.create_table(
                TableName=table_name,
                KeySchema=[
                    {
                        'AttributeName': 'id',
                        'KeyType': 'HASH' # Partition key
                    }
                ],
            )
    
```

The developer has done little more than write the name of the unit test class, when Amazon Q offers to complete it.

Notice the built-in references to the two functions created earlier in the same file.

The developer has only just begun to type in the name of the function that will create a DynamoDB table. But Amazon Q can tell where this is going.

Notice that the suggestion accounts for the DynamoDB session created earlier, and even mentions it in a comment.

```
69 # Unit test class
70 class TestBotoWhisper(unittest.TestCase):
71     def setUp(self):
72         self.s3 = session.client('s3', region_name=DEFAULT_REGION)
73         self.dynamodb = session.client('dynamodb', region_name=DEFAULT_REGION)
74         self.s3_resource = session.resource('s3', region_name=DEFAULT_REGION)
75         self.dynamodb_resource = session.resource('dynamodb', region_name=DEFAULT_REGION)
76
77     def tearDown(self):
78         self.s3.delete_bucket(Bucket=TEST_BUCKET_NAME)
79         self.dynamodb.delete_table(TableName=TEST_TABLE_NAME)
80
81     def test_create_s3_bucket(self):
82         self.assertTrue(create_s3_bucket(TEST_BUCKET_NAME, DEFAULT_REGION))
83
84     def test_create_dynamodb_table(self):
85         self.assertTrue(create_dynamodb_table(TEST_TABLE_NAME, DEFAULT_REGION))
```

Based only on a comment and the context, Amazon Q supplies the entire main function.

```
basics > 🐍 boto-whisper-demo.py > ...
80     def test_create_dynamodb_table(self):
81         create_dynamodb_table('my-test-table')
82         client = boto3.client('dynamodb', region_name='us-east-1')
83         response = client.list_tables()
84         self.assertIn('my-test-table', response['TableNames'])
85
86     # Main function to create bucket and table
87     def main():
88         create_s3_bucket(TEST_BUCKET_NAME, region='us-east-1')
89         create_dynamodb_table(TEST_TABLE_NAME, region='us-east-1')
```

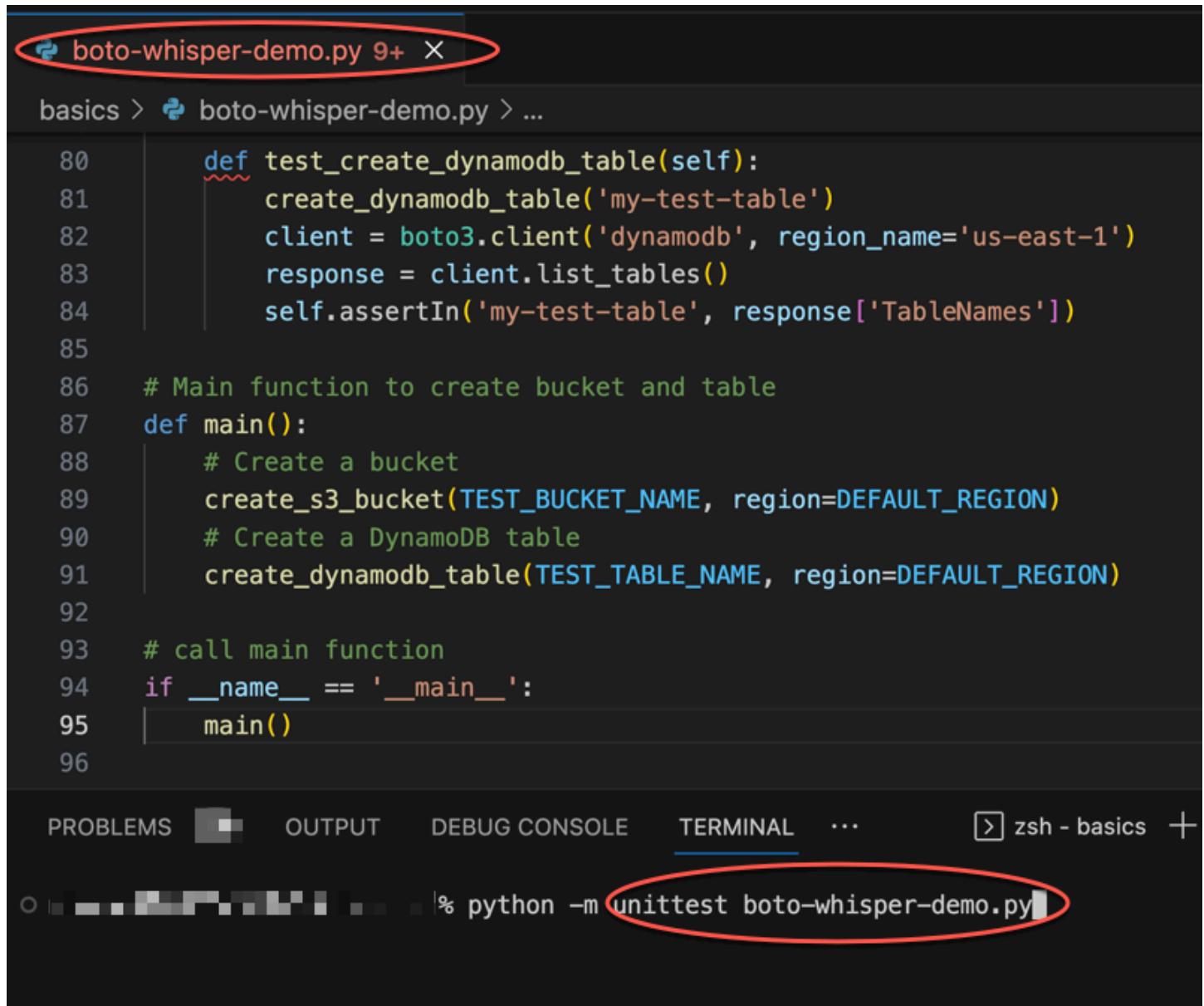
All that's left is the main guard, and Amazon Q knows it.

Based only on a comment and the context, Amazon Q supplies the entire main function.

```
# Main function to create bucket and table
def main():
    # Create a bucket
    create_s3_bucket(TEST_BUCKET_NAME, region=DEFAULT_REGION)
    # Create a DynamoDB table
    create_dynamodb_table(TEST_TABLE_NAME, region=DEFAULT_REGION)

# call| main function
if __name__ == '__main__':
    main()
```

Finally, the developer runs the unit test from the terminal of the same IDE where the coding took place.



```
boto-whisper-demo.py 9+ X
basics > boto-whisper-demo.py > ...
80     def test_create_dynamodb_table(self):
81         create_dynamodb_table('my-test-table')
82         client = boto3.client('dynamodb', region_name='us-east-1')
83         response = client.list_tables()
84         self.assertIn('my-test-table', response['TableNames'])
85
86     # Main function to create bucket and table
87     def main():
88         # Create a bucket
89         create_s3_bucket(TEST_BUCKET_NAME, region=DEFAULT_REGION)
90         # Create a DynamoDB table
91         create_dynamodb_table(TEST_TABLE_NAME, region=DEFAULT_REGION)
92
93     # call main function
94     if __name__ == '__main__':
95         main()
96
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    ...    zsh - basics +

```
% python -m unittest boto-whisper-demo.py
```

## Generating inline suggestions in AWS coding environments

In addition to third-party IDEs, Amazon Q Developer can generate inline suggestions within AWS services that provide their own coding environments.

The following sections describe how to set up Amazon Q inline code suggestions within integrated AWS services.

**Note**

If you are using Amazon Q as part of an enterprise, then you are using Amazon Q Developer Pro. In that case, administrators at your organization must complete additional steps before you can start coding. For more information, see [Getting started with Amazon Q Developer](#).

**Topics**

- [Using Amazon Q Developer with Amazon SageMaker Studio](#)
- [Using Amazon Q Developer with JupyterLab](#)
- [Using Amazon Q Developer with Amazon EMR Studio](#)
- [Using Amazon Q Developer with AWS Glue Studio](#)
- [Using Amazon Q Developer with AWS Lambda](#)
- [Using Amazon Q Developer with AWS Cloud9](#)
- [Using Amazon Q Developer with other services](#)

## Using Amazon Q Developer with Amazon SageMaker Studio

You can chat with Amazon Q inside Amazon SageMaker Studio. You can also make code recommendations automatically as you write your code.

To set up and activate Amazon Q for Amazon SageMaker Studio, see [Set up Amazon Q Developer for your users](#) in the *Amazon SageMaker User Guide*.

To use Amazon Q Developer with Amazon SageMaker Studio, you must Add the Amazon Q-related permissions to your SageMaker execution role.

Create an IAM policy containing the following statement. Then attach that policy to the execution role (IAM) or permission set (IAM Identity Center) associated with your user profile.

**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](#).

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AmazonQDeveloperPermissions",  
      "Effect": "Allow",  
      "Action": ["codewhisperer:GenerateRecommendations"],  
      "Resource": "*"  
    }  
  ]  
}
```

For more information about policies, see [Creating IAM policies](#) and [Adding and removing IAM identity permissions](#) in the *IAM User Guide*.

## Using Amazon Q Developer with JupyterLab

This page describes how to set up and activate Amazon Q Developer for JupyterLab. Once activated, Amazon Q can make code recommendations automatically as you write your code.

### Note

Python is the only programming language that Amazon Q supports in JupyterLab.

## Installing JupyterLab

Install [JupyterLab](#) on your computer or if you already have JupyterLab installed, check its version by running the following command.

```
pip show jupyterlab
```

Note the version in the response, and follow the use the corresponding directions in one of the following sections.

### Installation using pip for Jupyter Lab version >= 4.0

You can install and enable the Amazon Q extension for JupyterLab 4 with the following commands.

```
# JupyterLab 4
```

```
pip install amazon-q-developer-jupyterlab-ext
```

## Installation using pip for Jupyter Lab version >= 3.6 and < 4.0

You can install and enable the Amazon Q extension for JupyterLab 3 with the following commands.

```
# JupyterLab 3
pip install amazon-q-developer-jupyterlab-ext~=1.0
jupyter server extension enable amazon-q-developer-jupyterlab-ext
```

## Authenticating with AWS Builder ID

In the following procedure, you will set up Builder ID, which you will use to authenticate when you enable Amazon Q.

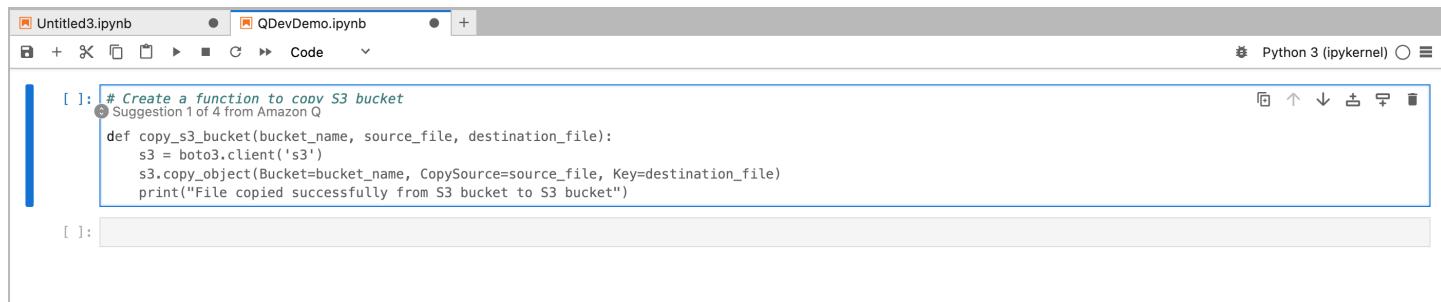
1. Refresh the browser tab on which you are using JupyterLab.
2. From the Amazon Q panel at the bottom of the window, choose **Start Amazon Q**.
3. From the pop-up window, choose **Copy Code and Proceed**.
4. On the **Create AWS Builder ID** page, if you don't have a Builder ID, enter a personal email address and choose **Next**.

If you already have a Builder ID, skip to the step about the **Authorize request** page.

5. On the next **Create your AWS Builder ID** page, enter a name and choose **Next**.
6. After you receive your email verification code, enter it in the blank field and choose **Verify**.
7. On the next screen, choose and confirm a password, then choose **Create AWS Builder ID**
8. On the next page choose **Allow** to allow Amazon Q to access your data.

Now you should be logged into Amazon Q in JupyterLab with Builder ID.

To begin coding, see [Using shortcut keys](#).



The screenshot shows a JupyterLab interface with two tabs: 'Untitled3.ipynb' and 'QDevDemo.ipynb'. The 'Code' tab is active. A code cell is visible, containing the following Python code:

```
[ ]: # Create a function to copy S3 bucket
Suggestion 1 of 4 from Amazon Q
def copy_s3_bucket(bucket_name, source_file, destination_file):
    s3 = boto3.client('s3')
    s3.copy_object(Bucket=bucket_name, CopySource=source_file, Key=destination_file)
    print("File copied successfully from S3 bucket to S3 bucket")
```

## Using Amazon Q Developer with Amazon EMR Studio

This page describes how to set up and activate Amazon Q Developer for Amazon EMR Studio. Once activated, Amazon Q can make code recommendations automatically as you write your ETL code.

### Note

Amazon Q supports Python, which can be used to code ETL scripts for Spark jobs in Amazon EMR Studio.

Use the following procedure to set up Amazon EMR Studio to work with Amazon Q.

1. Set up [Amazon EMR Studio Notebook](#).
2. Attach the following policy to the IAM user role for Amazon EMR Studio Notebook.

### 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](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AmazonQDeveloperPermissions",  
            "Effect": "Allow",  
            "Action": [  
                "codewhisperer:GenerateRecommendations"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

3. Open the [Amazon EMR console](#).
4. Under Amazon EMR Studio, choose **Workspaces (Notebooks)**.

5. Select your desired Workspace and choose **Quick launch**.

## Using Amazon Q Developer with AWS Glue Studio

This page describes how to set up and activate Amazon Q Developer for [AWS Glue Studio Notebook](#). Once activated, Amazon Q can make code recommendations automatically as you write your ETL code.

 **Note**

Amazon Q supports both Python and Scala, the two languages used for coding ETL scripts for Spark jobs in AWS Glue Studio.

In the following procedure, you will set up AWS Glue to work with Amazon Q.

1. [Set up AWS Glue Studio Notebook](#).
2. Attach the following policy to your IAM role for Glue Studio notebook

 **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](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AmazonQDeveloperPermissions",  
            "Effect": "Allow",  
            "Action": [  
                "codewhisperer:GenerateRecommendations"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

3. Open the [Glue console](#)
4. Under **ETL jobs**, choose **Notebooks**.
5. Verify that **Jupyter Notebook** is selected. Choose **Create**.
6. Enter a **Job name**.
7. For IAM role, select the role that you configured to interact with Amazon Q
8. Choose **Start notebook**.

## Using Amazon Q Developer with AWS Lambda

This document describes how to set up and activate Amazon Q Developer for the Lambda console. Once activated, Amazon Q can make code recommendations on demand in the Lambda code editor as you develop your function.

 **Note**

In the Lambda console, Amazon Q only supports functions using the Python and Node.js runtimes.

## AWS Identity and Access Management permissions for Lambda

For Amazon Q to provide recommendations in the Lambda console, you must enable the correct IAM permissions for either your IAM user or role. You must add the `codewhisperer:GenerateRecommendations` permission, as outlined in the sample IAM policy below:

 **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](#).

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

```
        "Sid": "AmazonQDeveloperPermissions",
        "Effect": "Allow",
        "Action": ["codewhisperer:GenerateRecommendations"],
        "Resource": "*"
    }
]
}
```

It is best practice to use IAM policies to grant restrictive permissions to IAM principals. For details about working with IAM for AWS Lambda, see [Identity and access management in AWS Lambda](#) in the *AWS Lambda Developer Guide*.

## Activating Amazon Q Developer with Lambda

To activate Amazon Q in the Lambda console code editor, complete these steps.

 **Note**

Amazon Q for Lambda is only supported in US East (N. Virginia).

1. Open the [Functions page](#) of the Lambda console, and choose the function that you want to edit.
2. In the code editor under **Code source**, choose **Tools** in the top menu bar.
3. Choose **Amazon Q code suggestions**. This immediately activates the Amazon Q service, and a check mark appears next to this option. To deactivate, choose this option again.

For shortcut keys, see [Using shortcut keys](#).

## Using Amazon Q Developer with AWS Cloud9

### AWS Identity and Access Management permissions for AWS Cloud9

For Amazon Q to provide recommendations in AWS Cloud9 console, you must enable the correct IAM permissions for either your IAM user or role. You must add the `codewhisperer:GenerateRecommendations` permission, as outlined in the sample IAM policy below:

### 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](#).

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AmazonQDeveloperPermissions",  
      "Effect": "Allow",  
      "Action": ["codewhisperer:GenerateRecommendations"],  
      "Resource": "*"  
    }  
  ]  
}
```

It is best practice to use IAM policies to grant restrictive permissions to IAM principals. For details about working with IAM for AWS Cloud9, see [Identity and access management in AWS Cloud9](#) in the *AWS Cloud9 user guide*.

## Activating Amazon Q with AWS Cloud9

To activate Amazon Q in the AWS Cloud9 console code editor, complete these steps.

1. From inside your existing AWS Cloud9 environment, choose the AWS logo on the left edge of the window. A panel will expand rightward.
2. In the lower part of the panel, under **Developer tools**, open the **Amazon Q** dropdown.
3. Choose **Resume Auto-Suggestions**.

For examples of how Amazon Q integrates with AWS Cloud9 and displays code suggestions in the AWS Cloud9 IDE, see [Code examples](#).

## Using Amazon Q Developer with other services

### AWS Identity and Access Management permissions for other services

For Amazon Q to provide recommendations in the context of another service, you must enable the correct IAM permissions for either your IAM user or role. You must add the

`codewhisperer:GenerateRecommendations` permission, as outlined in the sample IAM policy below:

 **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](#).

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AmazonQDeveloperPermissions",  
      "Effect": "Allow",  
      "Action": ["codewhisperer:GenerateRecommendations"],  
      "Resource": "*"  
    }  
  ]  
}
```

It is best practice to use IAM policies to grant restrictive permissions to IAM principals. For details about working with IAM, see [Security best practices](#) in the *IAM user guide*.

## Using shortcut keys

While getting inline suggestions from Amazon Q, you can use keyboard shortcuts for common actions you take, such as initiating Amazon Q or accepting a recommendation.

Choose the integrated development environment (IDE) where you are developing code to see keyboard shortcuts for your IDE.

### Visual Studio Code

Action	Keyboard shortcut
Manually initiate Amazon Q	MacOS: Option + C Windows: Alt + C

Action	Keyboard shortcut
Accept a recommendation	Tab
Next recommendation	Right arrow
Previous recommendation	Left arrow
Reject a recommendation	ESC, backspace, or keep typing and the recommendation will disappear as soon as there is a character mismatch.
Accept next word	Option + right arrow

To change keybindings in VS Code, see [Key Bindings for Visual Studio Code](#) on the VS Code website.

 **Note**

The inline suggestions toolbar in VS Code is disabled by default. For more information, see [Redesigned inline suggestions toolbar](#) on the VS Code website.

JetBrains

Action	Keyboard shortcut
Manually initiate Amazon Q	MacOS: Option + C Windows: Alt + C
Accept a recommendation	Tab
Next recommendation	Right arrow
Previous recommendation	Left arrow

Action	Keyboard shortcut
Reject a recommendation	ESC, backspace, or keep typing and the recommendation will disappear as soon as there is a character mismatch.

To change keybindings in IntelliJ, see [IntelliJ IDEA keyboard shortcuts](#) on the JetBrains website.

## Eclipse

Action	Keyboard shortcut
Manually initiate Amazon Q	MacOS: Option + C Windows: Alt + C
Accept a recommendation	Tab
Next recommendation	MacOS: Option + ] Windows: Alt + ]
Previous recommendation	MacOS: Option + [ Windows: Alt + [
Reject a recommendation	ESC, backspace, or keep typing and the recommendation will disappear as soon as there is a character mismatch.

To change keybindings in Eclipse, see [Changing the key bindings](#) in the Eclipse documentation.

## Toolkit for Visual Studio

Action	Keyboard shortcut
Manually initiate Amazon Q	Windows: Alt + C

Action	Keyboard shortcut
AwSToolkit.CodeWhisperer.GetSuggestion in the keybindings	
Accept a recommendation	Tab
Next recommendation	Windows: Alt + .
Edit.NextSuggestion in the keybindings	
Previous recommendation	Windows: Alt + ,
Edit.PreviousSuggestion in the keybindings	
Reject a recommendation	ESC, backspace, or keep typing and the recommendation will disappear as soon as there is a character mismatch.

See also Microsoft's [Visual Studio default keyboard shortcuts](#).

To change keybindings in Visual Studio, use Tools -> Options -> Keyboard.

## Amazon SageMaker

Action	Keyboard shortcut
Manually initiate Amazon Q	MacOS: Option + C Windows: Alt + C
Accept a recommendation	Tab
Next recommendation	Down arrow
Previous recommendation	Up arrow
Reject a recommendation	ESC

## JupyterLab

Action	Keyboard shortcut
Manually initiate Amazon Q	MacOS: Option + C Windows: Alt + C
Accept a recommendation	Tab
Next recommendation	Down arrow
Previous recommendation	Up arrow
Reject a recommendation	ESC

## AWS Glue Studio Notebook

Action	Keyboard shortcut
Manually initiate Amazon Q	MacOS: Option + C Windows: Alt + C
Accept a recommendation	Tab
Next recommendation	Down arrow
Previous recommendation	Up arrow
Reject a recommendation	ESC

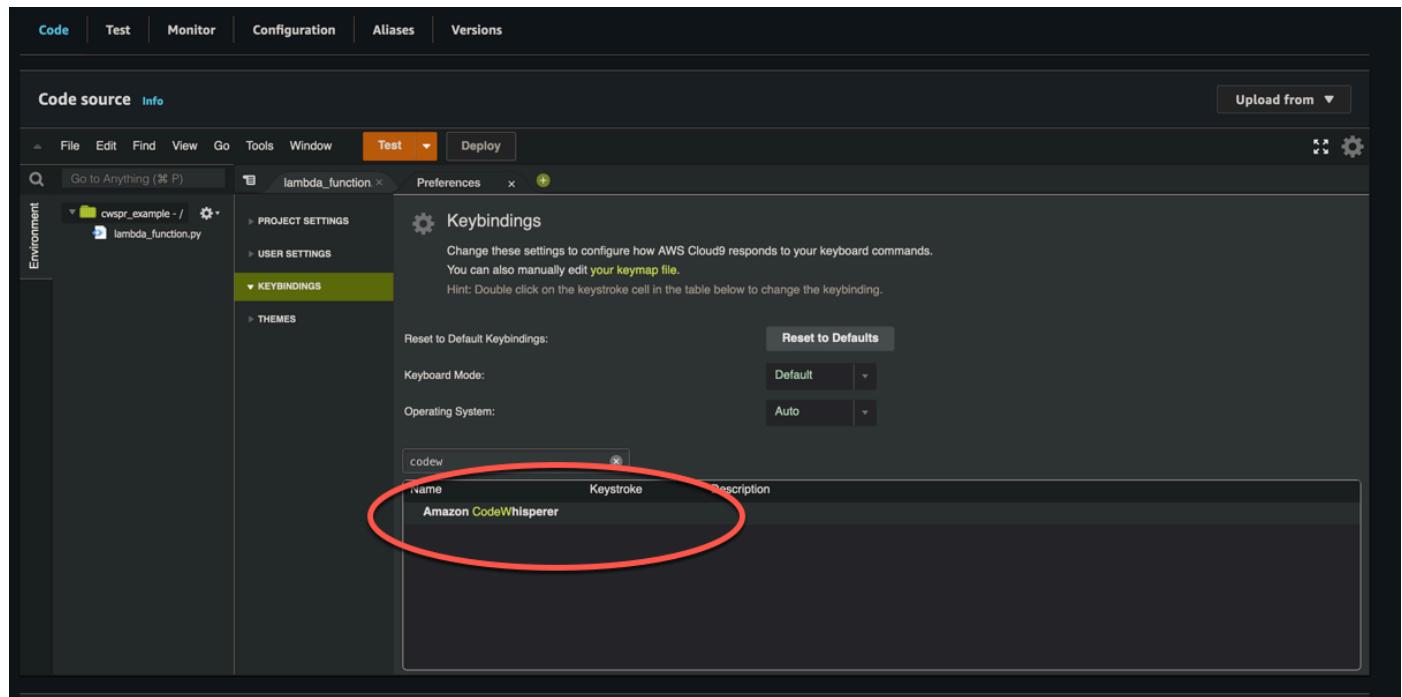
## AWS Lambda

Action	Keyboard shortcut
Manually fetch a code suggestion	MacOS: Option + C Windows: Alt + C

Action	Keyboard shortcut
Accept a suggestion	Tab
Reject a suggestion	ESC, Backspace, scroll in any direction, or keep typing and the recommendation automatically disappears.

To change the key bindings, use the following procedure.

1. While viewing a particular function, choose the gear icon to open the **Preferences** tab.
2. On the **Preferences** tab, select **Keybindings**.
3. In the keybindings search box, enter Amazon Q.

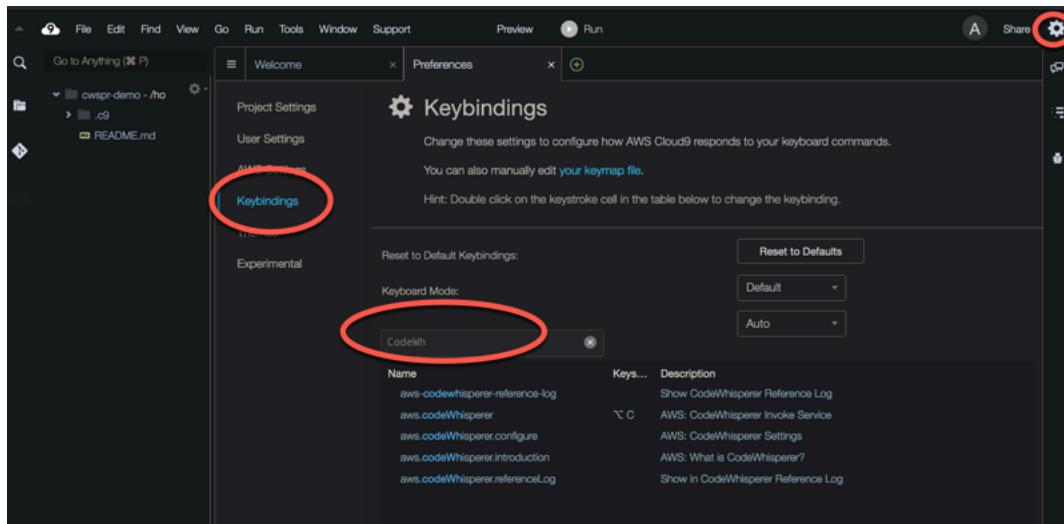


## AWS Cloud9

Action	Keyboard shortcut
Manually fetch a code suggestion	MacOS: Option + C Windows: Alt + C

Action	Keyboard shortcut
Accept a suggestion	Tab
Reject a suggestion	ESC, Backspace, scroll in any direction, or keep typing and the recommendation automatically disappears.

1. While viewing a particular environment, choose the gear icon to open the **Preferences** tab.
2. On the **Preferences** tab, select **Keybindings**.
3. In the keybindings search box, enter Amazon Q.
4. In the Keystroke column, double-click the space corresponding to the function you're interested in.
5. Enter the keys that you want to bind the function to.



## Using code references

Amazon Q learns, in part, from open-source projects. Sometimes, a suggestion it's giving you may be similar to a specific piece of training data. Code references include information about the source Amazon Q used to generate a recommendation.

### Topics

- [View and update code references](#)
- [Turn code references off and on](#)
- [Opt out of code with references](#)

## View and update code references

With the reference log, you can view references to code recommendations that are similar to training data. You can also update and edit code recommendations suggested by Amazon Q.

Choose your IDE to see steps for how to view and update code references.

### Visual Studio Code

To display the Amazon Q reference log in VS Code, use the following procedure.

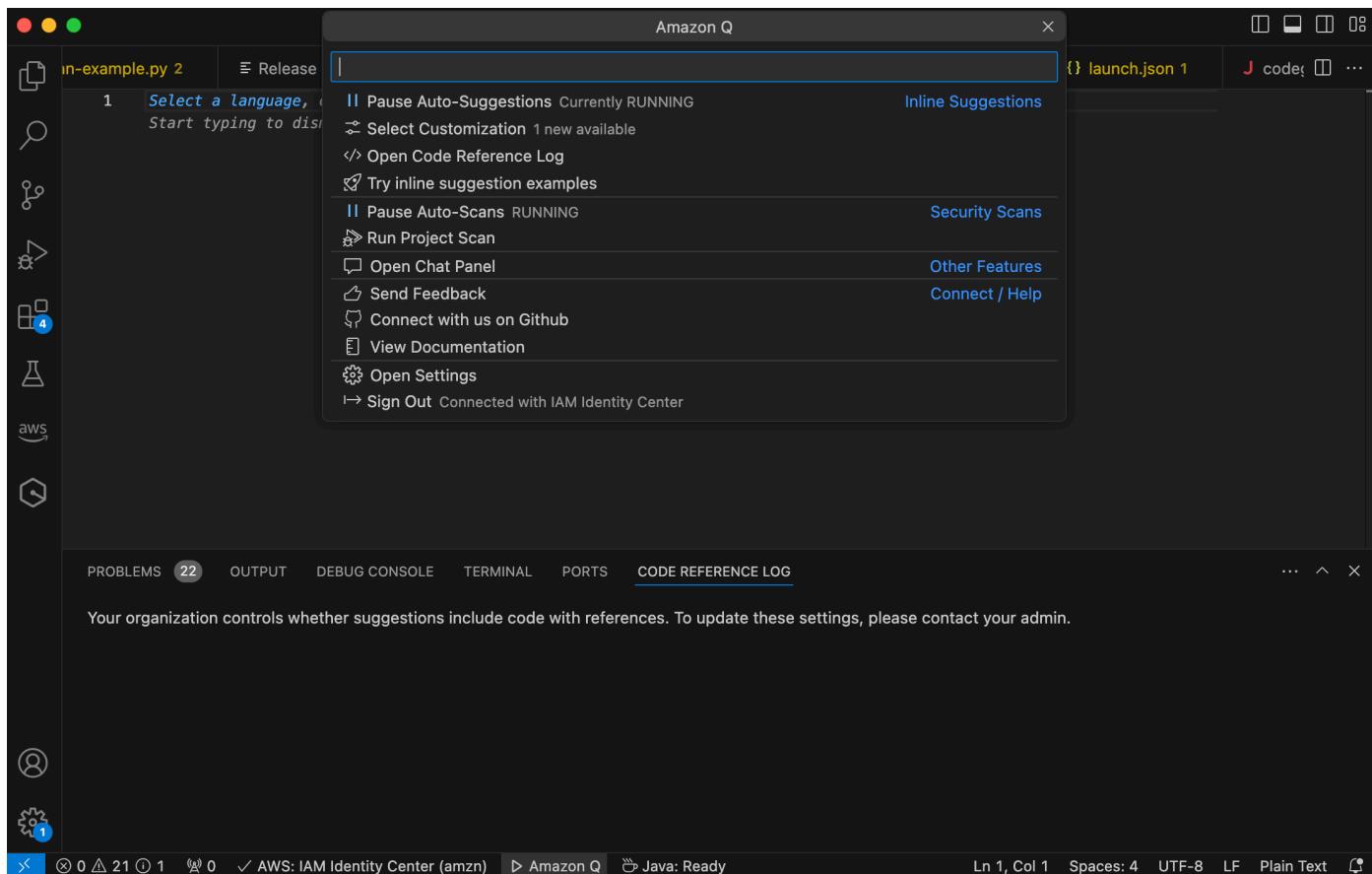
1. Make sure you are using the latest version of both VS Code and the Amazon Q extension.
2. In VS Code, choose **Amazon Q** from the component tray at the bottom of the IDE window.

The Amazon Q task bar opens at the top of the IDE window.

3. Choose **Open Code Reference Log**.

The code reference log tab opens. Any references to code recommendations are listed.

The following image shows the open Amazon Q task bar and code reference log tab.



## JetBrains

To display the Amazon Q reference log in JetBrains IDEs, use the following procedure.

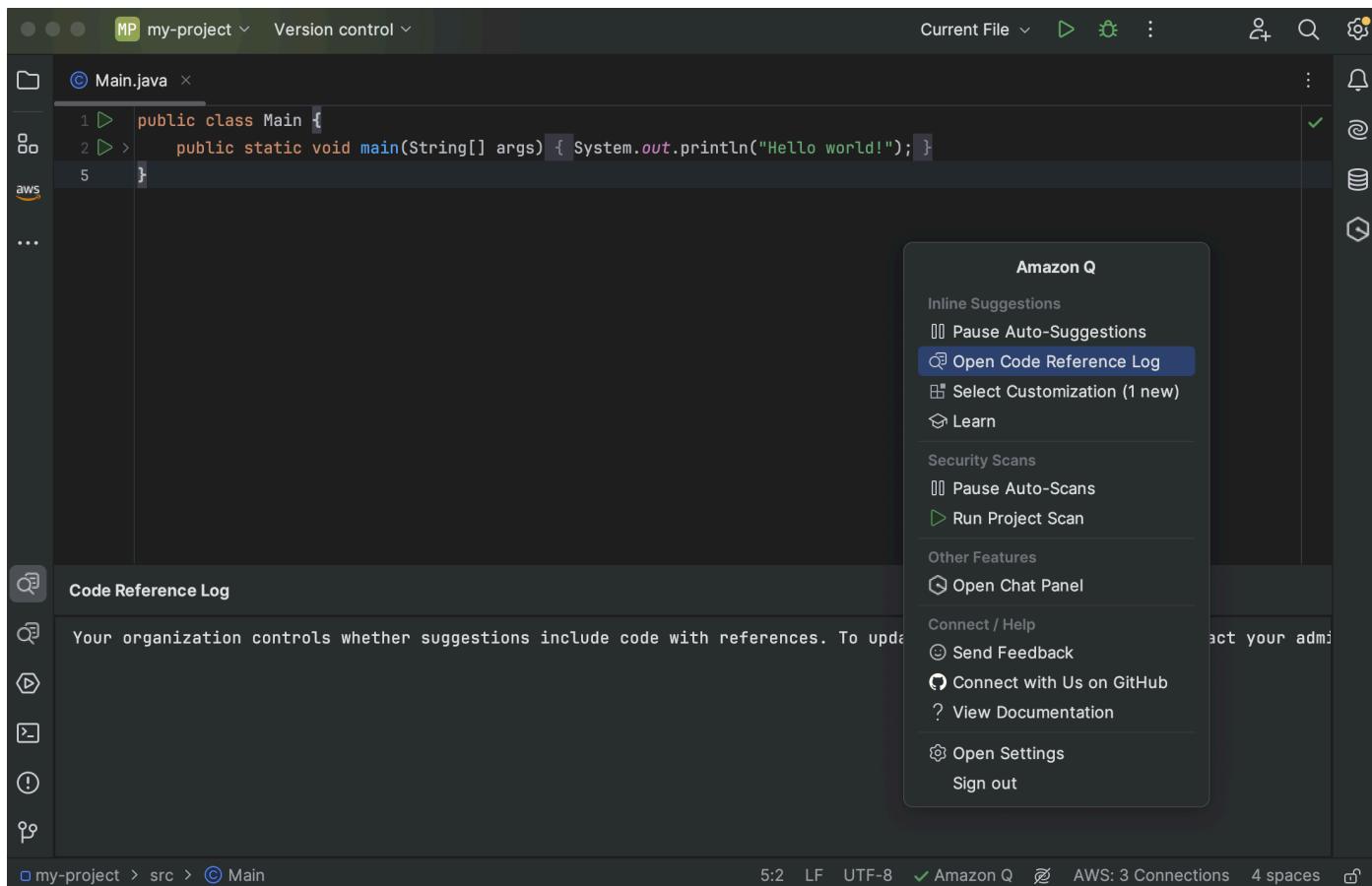
1. Make sure you are using the latest version of both your JetBrains IDE and the Amazon Q plugin.
2. In JetBrains, choose **Amazon Q** from the status bar at the bottom of the IDE window.

The Amazon Q task bar opens above the status bar.

3. Choose **Open Code Reference Log**.

The code reference log tab opens. Any references to code recommendations are listed.

The following image shows the open Amazon Q task bar and code reference log tab.

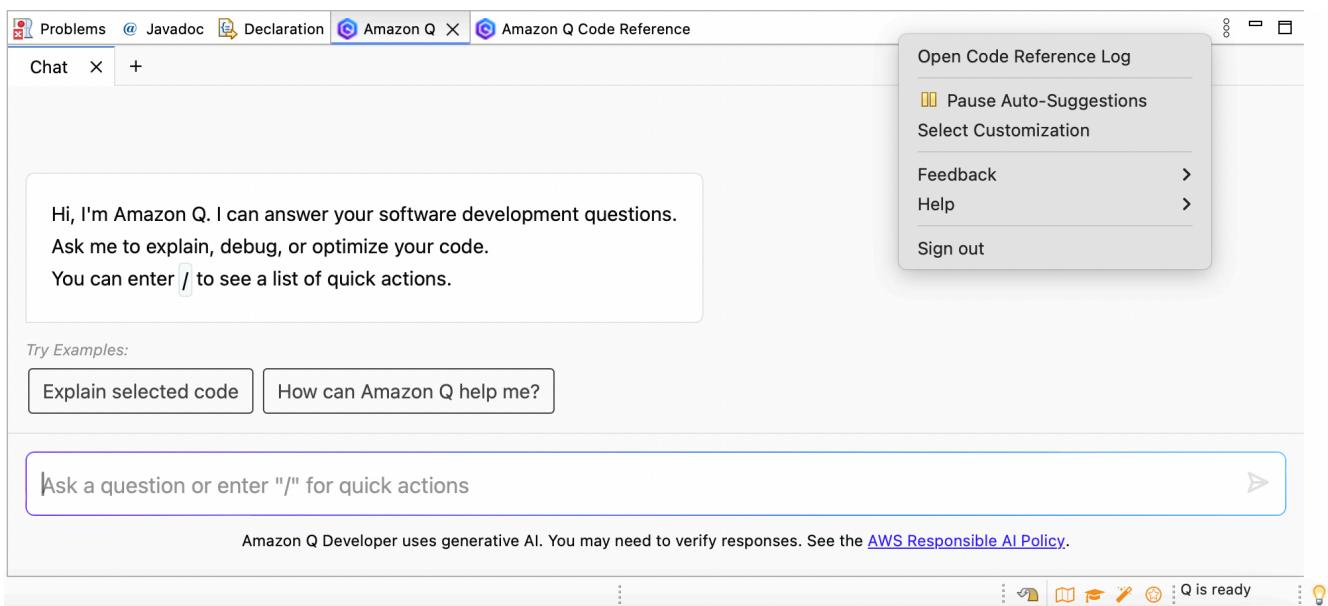


## Eclipse

To display the Amazon Q reference log in Eclipse IDEs, use the following procedure.

1. Make sure you are using the latest version of both the Eclipse IDE and the Amazon Q plugin.
2. In your Eclipse IDE, choose the **Amazon Q** icon in the top right corner of the IDE.
3. With the Amazon Q chat tab open, choose the ellipsis icon in the top right corner of the tab. The Amazon Q task bar opens.

The following image shows the Amazon Q task bar in an Eclipse IDE.



#### 4. Choose Open Code Reference Log.

The code reference log tab opens. Any references to code recommendations are listed.

#### Toolkit for Visual Studio

When Amazon Q suggests code that contains a reference in the Toolkit for Visual Studio, the reference type appears in the suggestion description.

```
# Create function to create a DynamoDB Table
def Suggestion(License: MIT) 1 / 1 | Tab to accept | 
    table = dynamodb.create_table(
        TableName='Products',
        KeySchema=[
            {
                'AttributeName': 'id'.

```

All accepted suggestions that contain references are captured in the reference log.

To access the reference log, choose the AWS icon, then select **Open Code Reference Log**.

A list of accepted suggestions that contain references will appear. This list includes:

- The location where the suggestion was accepted. Double clicking on this will take you to that location in your code.
- The associated license
- The referenced source code

- The fragment of code attributed to the reference

The screenshot shows the AWS Cloud 9 Output console. At the top, there is a dropdown menu labeled "Show output from: CodeWhisperer Reference Log". Below the dropdown are several icons for managing the log. The main area displays a snippet of Python code from a file named "Code\cloud.py". The code includes a line that accepted a recommendation with an MIT license. A portion of the code is highlighted in blue, indicating it is a reference from CodeWhisperer.

## AWS Cloud 9

When you use Amazon Q with AWS Cloud 9, code references are on by default.

To turn them off, or to turn them back on later, use the following procedure.

1. On the AWS Cloud 9 console, in the upper left corner, choose the AWS Cloud 9 logo.
2. From the dropdown menu, choose **Preferences**.

On the right side of the console, the **Preferences** tab will open.

3. On the **Preferences** tab, under **Project Settings**, under **Extensions**, select **AWS Toolkit**.
4. Select or deselect **Amazon Q: Include Suggestions With Code References**.

## Lambda

Amazon Q in Lambda does not support code references. When you use Amazon Q with Lambda, any code suggestions with references are omitted.

## SageMaker Studio

To display the Amazon Q reference log in SageMaker Studio, use the following procedure.

1. At the bottom of the SageMaker Studio window, open the Amazon Q panel.
2. Choose **Open Code Reference Log**.

## JupyterLab

To display the Amazon Q reference log in JupyterLab, use the following procedure.

1. At the bottom of the JupyterLab window, open the Amazon Q panel.
2. Choose **Open Code Reference Log**.

## AWS Glue Studio Notebook

To display the Amazon Q reference log in AWS Glue Studio Notebook, use the following procedure.

1. At the bottom of the AWS Glue Studio Notebook window, open the Amazon Q panel.
2. Choose **Open Code Reference Log**.

## Turn code references off and on

In most IDEs, code references are on by default. Choose your IDE to see steps for how to turn code references off or on.

### Visual Studio Code

When you use Amazon Q with VS Code, code references are on by default.

To turn them off, or to turn them back on later, use the following procedure.

1. Make sure you are using the latest version of both VS Code and the Amazon Q extension.
2. In VS Code, choose **Amazon Q** from the component tray at the bottom of the IDE window.  
The Amazon Q task bar opens at the top of the IDE window.
3. Choose **Open Settings**. The settings tab opens with the options related to Amazon Q displayed.

4. Select or deselect the box next to **Show Inline Code Suggestions with Code References**.

## JetBrains

When you use Amazon Q with your JetBrains IDE, code references are on by default.

To turn them off, or to turn them back on later, use the following procedure.

1. Make sure you are using the latest version of both your JetBrains IDE and the Amazon Q plugin.
2. In JetBrains, choose **Amazon Q** from the status bar at the bottom of the IDE window.  
The Amazon Q task bar opens above the status bar.
3. Choose **Open Settings**. The settings window opens with the options related to Amazon Q displayed.
4. Select or deselect the box next to **Include suggestions with code references**.

## Eclipse

When you use Amazon Q with Eclipse, code references are on by default.

To turn them off, or to turn them back on later, use the following procedure.

1. Make sure you are using the latest version of both the Eclipse IDE and the Amazon Q plugin.
2. Open **Settings** in your Eclipse IDE.
3. Choose **Amazon Q** from the left navigation bar.
4. Select or deselect the box next to **Show Inline Code Suggestions with Code References**.
5. Choose **Apply** to save your changes.

## AWS Cloud 9

When you use Amazon Q with AWS Cloud 9, code references are on by default.

To turn them off, or to turn them back on later, use the following procedure.

1. On the AWS Cloud 9 console, in the upper left corner, choose the AWS Cloud 9 logo.

2. From the dropdown menu, choose **Preferences**.

On the right side of the console, the **Preferences** tab will open.

3. On the **Preferences** tab, under **Project Settings**, under **Extensions**, select **AWS Toolkit**.
4. Select or deselect **Amazon Q: Include Suggestions With Code References**.

## Lambda

Amazon Q in Lambda does not support code references. When you use Amazon Q with Lambda, any code suggestions with references are omitted.

## SageMaker Studio

When you use Amazon Q with SageMaker Studio, code references are on by default.

To turn them off, or to turn them back on later, use the following procedure.

1. From the top of the SageMaker Studio window choose **Settings**.
2. From the **Settings** dropdown, choose **Advanced Settings Editor**.
3. In the Amazon Q dropdown, select or deselect the box next to **Enable suggestions with code references**.

## JupyterLab

When you use Amazon Q with JupyterLab, code references are on by default.

To turn them off, or to turn them back on later, use the following procedure.

1. From the top of the JupyterLab window choose **Settings**.
2. From the **Settings** dropdown, choose **Advanced Settings Editor**.
3. In the Amazon Q dropdown, select or deselect the box next to **Enable suggestions with code references**.

## AWS Glue Studio Notebook

1. From the bottom of the AWS Glue Studio Notebook window choose **Amazon Q**.
2. From the pop-up menu, toggle the switch next to **Code with references**.

**Note**

Pausing code references will be valid only for the duration of the current AWS Glue Studio Notebook.

## Opt out of code with references

In some IDEs, you can opt out of receiving suggestions with references at the administrator level.

Choose your IDE to see steps for opting out as an administrator.

### Visual Studio Code

If you are an enterprise administrator, you can opt out of suggestions with code references for your entire organization. If you do this, individual developers in your organization will not be able to opt back in through the IDE. Those developers will be able to select and deselect the box discussed in the previous section, but it will have no effect if you have opted out at the enterprise level.

To opt out of suggestions with references at the enterprise level, use the following procedure.

1. In the Amazon Q Developer console, choose **Settings**.
2. In the **Amazon Q Developer account details** pane, choose **Edit**.
3. On the Edit details page, in the **Advanced settings** pane, deselect **Include suggestions with code references**.
4. Choose **Save changes**.

### JetBrains

If you are an enterprise administrator, you can opt out of suggestions with code references for your entire organization. If you do this, individual developers in your organization will not be able to opt back in through the IDE. Those developers will be able to select and deselect the box discussed in the previous section, but it will have no effect if you have opted out at the enterprise level.

To opt out of suggestions with references at the enterprise level, use the following procedure.

1. In the Amazon Q Developer console, choose **Settings**.

2. In the **Amazon Q Developer account details** pane, choose **Edit**.
3. On the Edit details page, in the **Advanced settings** pane, deselect **Include suggestions with code references**.
4. Choose **Save changes**.

## Eclipse

If you are an enterprise administrator, you can opt out of suggestions with code references for your entire organization. If you do this, individual developers in your organization will not be able to opt back in through the IDE. Those developers will be able to select and deselect the box discussed in the previous section, but it will have no effect if you have opted out at the enterprise level.

To opt out of suggestions with references at the enterprise level, use the following procedure.

1. In the Amazon Q Developer console, choose **Settings**.
2. In the **Amazon Q Developer account details** pane, choose **Edit**.
3. On the Edit details page, in the **Advanced settings** pane, deselect **Include suggestions with code references**.
4. Choose **Save changes**.

## Toolkit for Visual Studio

To opt out of suggestions with references at the enterprise level, use the following procedure.

1. You can get to the code references setting in one of two ways:
  - a. Choose the Amazon Q icon at the edge of the window, and then choose **Options...**
  - b. Go to **Tools -> AWS Toolkit -> Amazon Q**
2. Change the toggle to **True or False**, depending on whether you want to include suggestions with references.

## AWS Cloud 9

Amazon Q in AWS Cloud 9 does not support opting out of code suggestions with references at the enterprise level.

To opt out at the individual developer level, see [Toggling code references](#).

## Lambda

Amazon Q in Lambda does not support code references. When you use Amazon Q with Lambda, any code suggestions with references are omitted.

## SageMaker Studio

Amazon Q does not support opting out of code suggestions with references at the enterprise level in SageMaker Studio.

## JupyterLab

Amazon Q does not support opting out of code suggestions with references at the enterprise level in JupyterLab.

## AWS Glue Studio Notebook

Amazon Q does not support opting out of code suggestions with references in AWS Glue Studio Notebook.

## Code examples

Amazon Q can suggest code in different scenarios. To understand how it can help you as you write code in your programming language of choice, view the following code examples.

### Topics

- [Using Amazon Q Developer for single-line code completion](#)
- [Using Amazon Q Developer for full function generation](#)
- [Using Amazon Q Developer for block completion](#)
- [Using Amazon Q Developer for Docstring, JSDoc, and Javadoc completion](#)
- [Using Amazon Q Developer for line-by-line recommendations](#)

## Using Amazon Q Developer for single-line code completion

When you start typing out single lines of code, Amazon Q makes suggestions based on your current and previous inputs.

## C++

```
17 int main(int argc, char **argv) {
18     Aws::SDKOptions options;
19     Aws::InitAPI(options); // Should only be called once.
20     {
21         Aws::Client::ClientConfiguration clientConfig;
22
23         clientConfig.region = "us-east-1";
24
25         Aws::SQS::SQSClient sqsClient(clientConfig);
26
27         Aws::Vector<Aws::String> allQueueUrls;
28         Aws::String nextToken; // Next token is used to handle a paginated response.
29         do {
30             Aws::SQS::Model::ListQueuesRequest request;
31
32             I
33
34         } while (!nextToken.empty());
35     }
36 }
```

## JavaScript

In this example, Amazon Q completes a line of code that the developer begins.

```
1  /*
2  * Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
3  * SPDX-License-Identifier: Apache-2.0
4  */
5
6 // Upload an object to Amazon S3 bucket.
7 I
```

## TypeScript

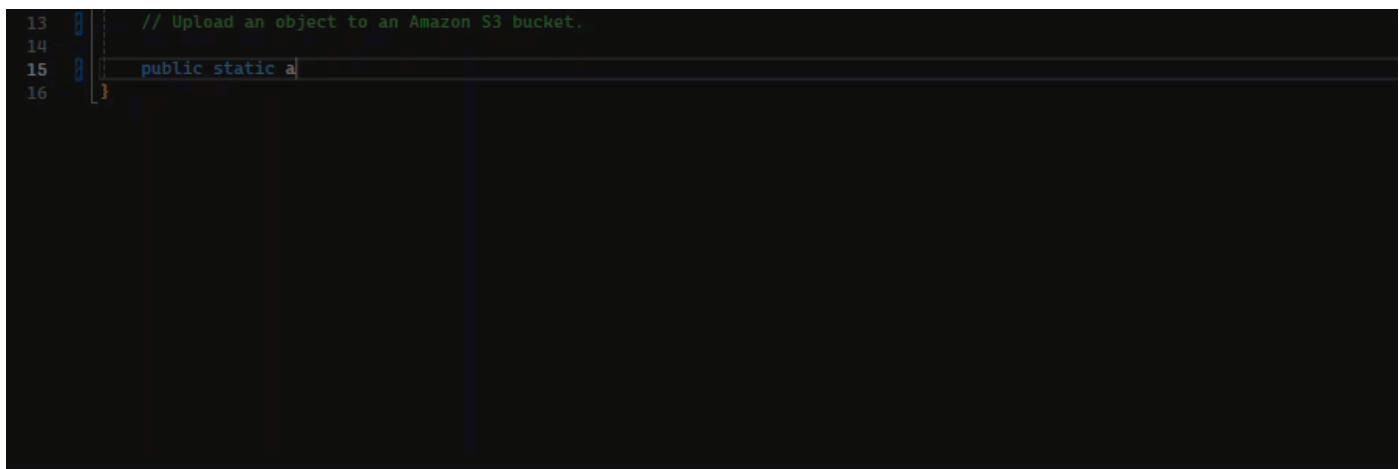
In this example, the user enters a full comment, and then Amazon Q supplies the code that goes with it.



```
TS index.ts  X
TS index.ts > ...
1   import {S3Client} from "@aws-sdk/client-s3";
2
3   const client = new S3Client({});
4
5   |
```

## C#

In this example, Amazon Q provides a single-line recommendation based on a comment.



```
13 // Upload an object to an Amazon S3 bucket.
14
15 public static a
16 }
```

## Shell

In the image below, Amazon Q offers recommendations on how to complete a single line of code.

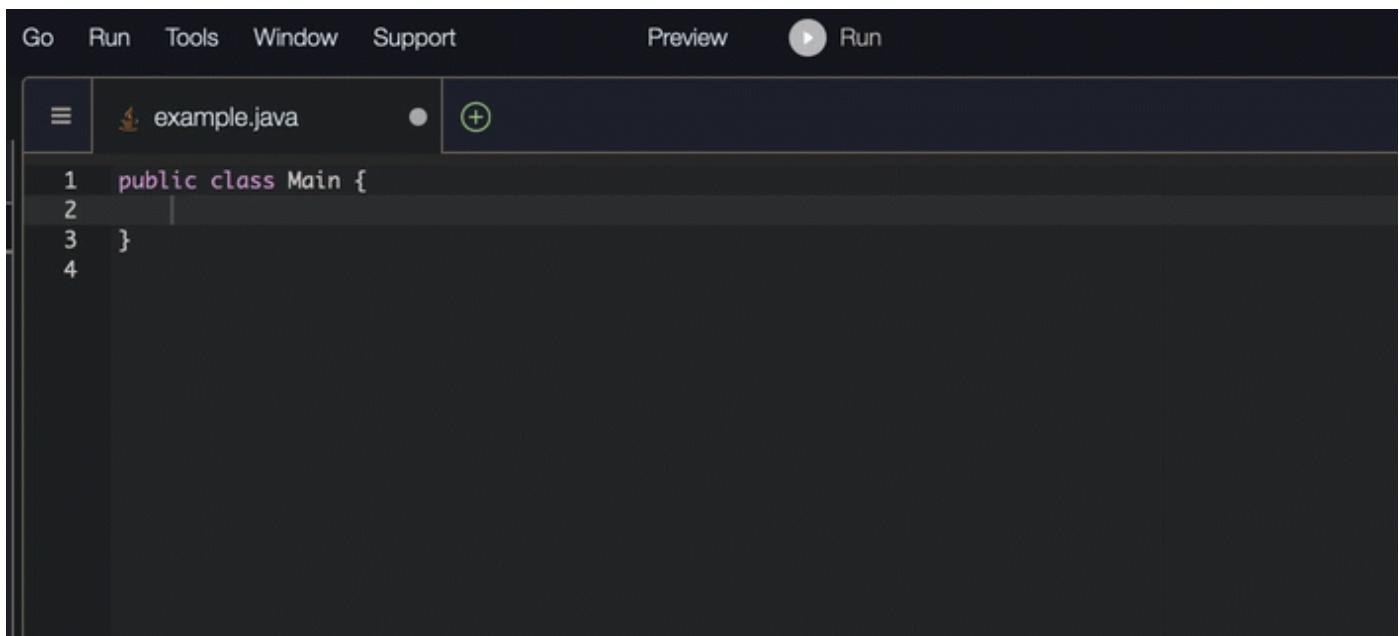
```
local access_key_response
access_key_response=$(iam_create_user_access_key -u "$user_name")
# shellcheck disable=SC2181
if [[ ${?} != 0 ]]; then
    errecho "The access key failed to create. This demo will exit."
    clean_up "$user_name"
    return 1
fi
```

## Java

When you start typing out single lines of code, Amazon Q makes suggestions based on your current and previous inputs.

In the example below, in Java, a user enters the string `public` into an existing class.

Based on the input, Amazon Q generates a suggestion for the signature of the main method.



The screenshot shows a Java code editor interface. The menu bar includes 'Go', 'Run', 'Tools', 'Window', 'Support', 'Preview', and 'Run'. A file named 'example.java' is open. The code currently contains:

```
1 public class Main {  
2  
3 }  
4
```

A cursor is positioned at the end of the first line. A green pop-up suggestion box appears, containing the text 'public static void main(String[] args) {'. This indicates that Amazon Q has suggested a standard main method signature based on the context of the code being typed.

## Python

In this example, Amazon Q recommends a single line of code, based on the developer's comment.

```
sagemaker_session = sage.Session()
bucket = sagemaker_session.default_bucket()
runtime = boto3.client("runtime.sagemaker")
s3 = boto3.resource("s3")

# Create a prefix called sampledata.
prefix = "sampledata"

# Create a filename called rawdata.csv
filename = "rawdata.csv"
```

## Using Amazon Q Developer for full function generation

Amazon Q can generate an entire function based on a comment that you've written. As you finish your comment Amazon Q will suggest a function signature. If you accept the suggestion, Amazon Q automatically advances your cursor to the next part of the function and makes a suggestion. Even if you enter an additional comment or line of code in between suggestions, Amazon Q will refactor based on your input.

C

```
32
33     bool AwsDoc::SQS::createQueue(const Aws::String &queueName,
34                                     const Aws::Client::ClientConfiguration &clientConfigurat
```

## C++

A screenshot of a code editor showing a partial C++ function definition. The code is as follows:

```
32     bool AwsDoc::SQS::createQueue(const Aws::String &queueName,
33                                     const Aws::Client::ClientConfiguration &clientConfigurat
```

## JavaScript

In the following example, the user generates, and then edits, a full function based on a set of comments.

A screenshot of a code editor showing a generated AWS Lambda handler function named index.js. The code is as follows:

```
1 /**
2  * AWS Lambda handler
3  *
4  * Exports a single function that takes the "operand" property from the event
5  * input, squares it, and returns it.
6 */
7 |
8 |
9 |
10
```

In the following image, a user has written a function signature for reading a file from Amazon S3. Amazon Q then suggests a full implementation of the `read_from_s3` method.

A screenshot of a code editor showing a function signature and an AI-generated implementation. The code is as follows:

```
def read_from_s3(bucket, key):
    import boto3
    s3 = boto3.client('s3')
    obj = s3.get_object(Bucket=bucket, Key=key)
    return obj['Body'].read().decode('utf-8')
```

**Note**

Sometimes, as in the previous example, Amazon Q includes import statements as part of its suggestions. As a best practice, manually move these import statements to the top of your file.

As another example, in the following image, a user has written a function signature. Amazon Q then suggests a full implementation of the quicksort method.

```
def quicksort(a):
    if len(a) <= 1:
        return a
    else:
        pivot = a[0]
        less = [i for i in a[1:] if i <= pivot]
        greater = [i for i in a[1:] if i > pivot]
        return quicksort(less) + [pivot] + quicksort(greater)
```

Amazon Q considers past code snippets when making suggestions. In the following image, the user in the previous example has accepted the suggested implementation for quicksort above. The user then writes another function signature for a generic sort method. Amazon Q then suggests an implementation based on what has already been written.

```
def quicksort(a):
    if len(a) <= 1:
        return a
    else:
        pivot = a[0]
        less = [i for i in a[1:] if i <= pivot]
        greater = [i for i in a[1:] if i > pivot]
        return quicksort(less) + [pivot] + quicksort(greater)

def sort(a):
    return quicksort(a)
```

In the following image, a user has written a comment. Based on this comment, Amazon Q then suggests a function signature.

```
# Binary search function
```

```
def binary_search(arr, l, r, x):
```

In the following image, the user in the previous example has accepted the suggested function signature. Amazon Q can then suggest a complete implementation of the `binary_search` function.

```
# Binary search function
```

```
def binary_search(arr, l, r, x):
```

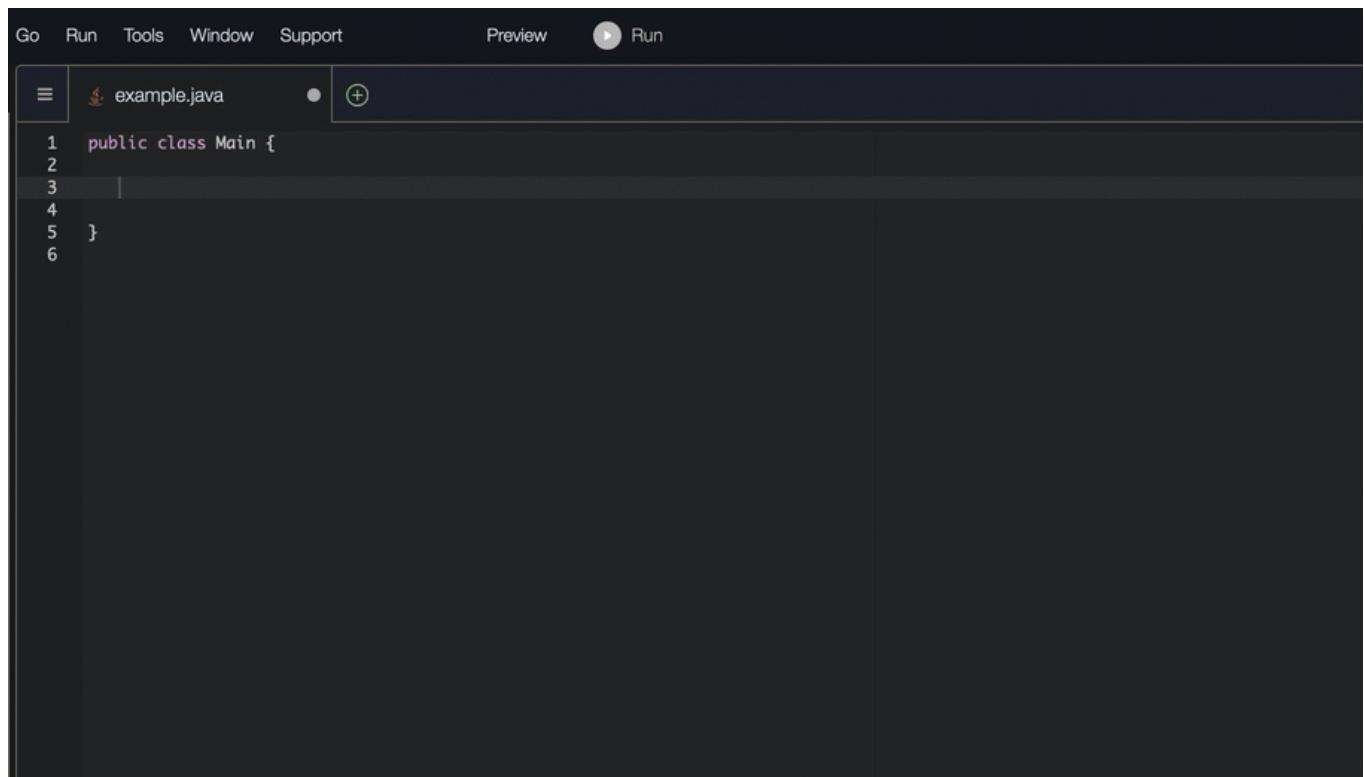
```
    while l <= r:
        mid = l + (r - l) // 2
        if arr[mid] == x:
            return mid
        elif arr[mid] < x:
            l = mid + 1
        else:
            r = mid - 1
```

Java

The following list contains examples of how Amazon Q makes suggestions and advances you through the entire process of creating a function.

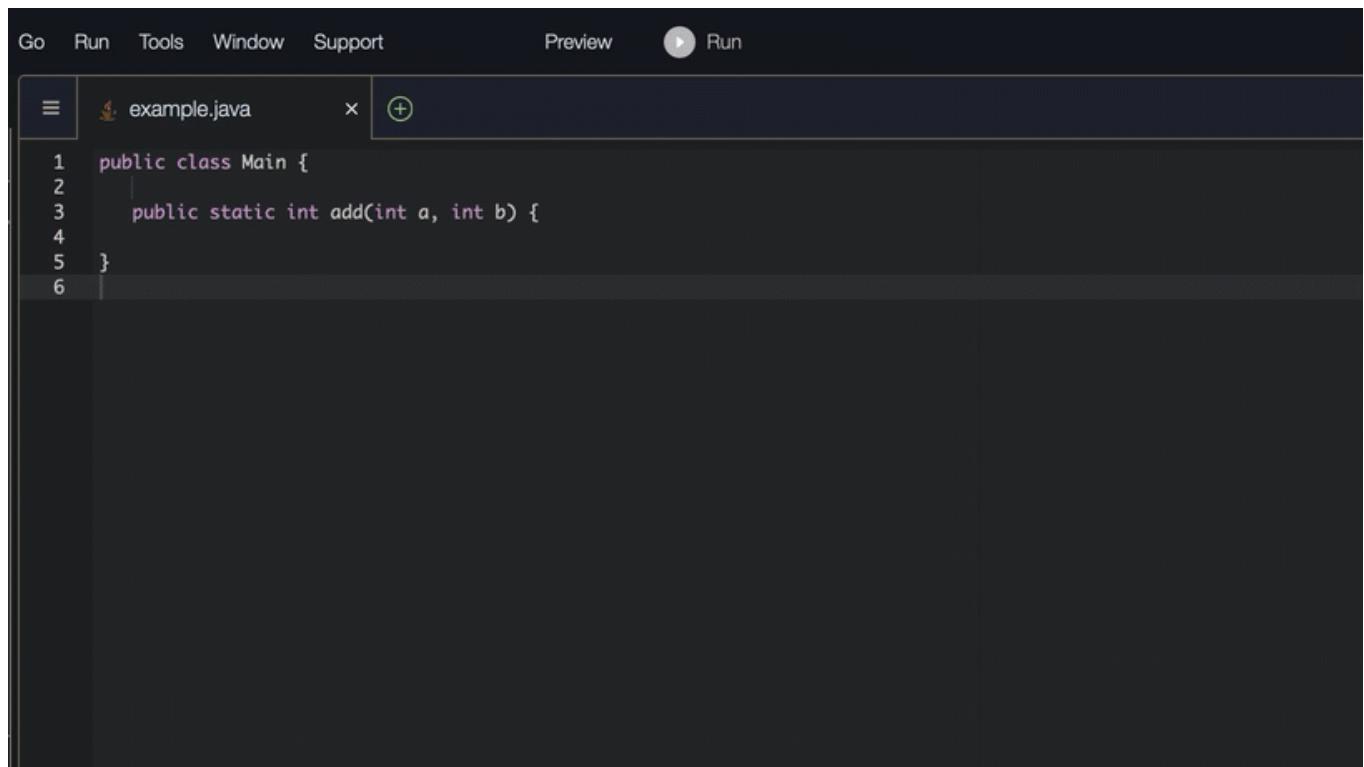
1. In the following example, a user inputs a comment. Amazon Q suggests a function signature.

After the user accepts that suggestion, Amazon Q suggests a function body.



```
1 public class Main {  
2     // This is a test.  
3 }  
4  
5 }
```

2. In the image below, a user inputs a comment in the body of the function prior to accepting a suggestion from Amazon Q. On the following line, Amazon Q generates a suggestion based on the comment.



```
1 public class Main {  
2     // Add two numbers.  
3     public static int add(int a, int b) {  
4         return a + b;  
5     }  
6 }
```

## C#

In the following example, Amazon Q recommends a full function.

```
15  // Create a function that outputs DynamoDB table names
16
17  public static async Task ListTables(AmazonDynamoDBClient |
18 }
```

## TypeScript

In the following example, Amazon Q generates a function based on the user's docstrings.

```
/** 
 * Upload a large file to an S3 bucket in multiple parts.
 * @param {string} fileName -- The name of the file to upload.
 * @param {string} bucketName -- The name of the bucket to upload to.
 */
```

## Python

Amazon Q can generate an entire function based on a comment that you've written. As you finish your comment, Amazon Q will suggest a function signature. If you accept the suggestion, Amazon Q automatically advances your cursor to the next part of the function and makes a suggestion. Even if you enter an additional comment or line of code in between suggestions, Amazon Q will refactor based on your input.

In the following example, Amazon Q generates both a full function and the corresponding unit test.

```
1 import boto3
2 ddb_client = boto3.client('dynamodb')
3
```

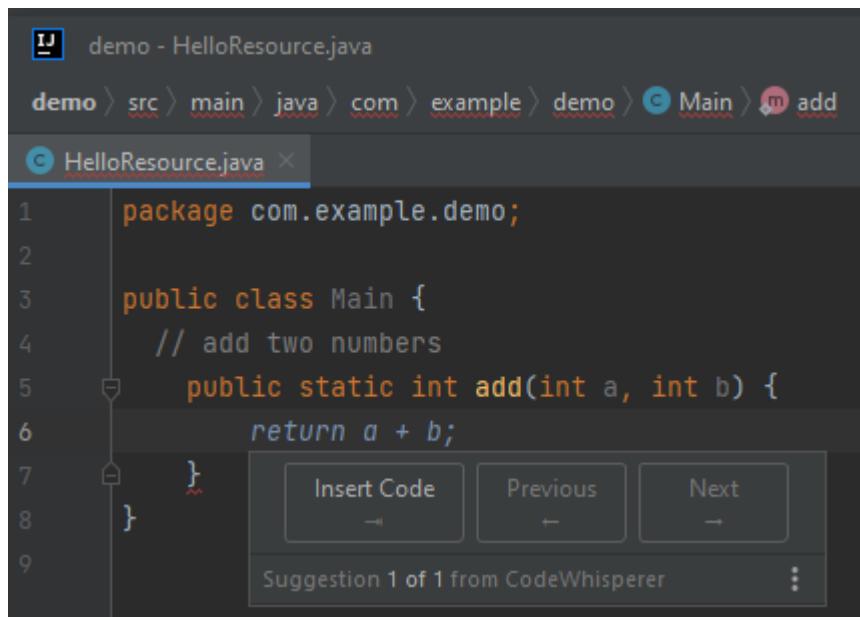
The following list contains examples of how Amazon Q makes suggestions and advances you through the entire process of creating a function.

1. In the image below, a user has input a comment. The function signature, located below the comment, is a suggestion from Amazon Q.

A screenshot of a Java code editor showing a file named `HelloResource.java`. The code defines a class `Main` with a static method `add`. A tooltip or suggestion box is overlaid on the code editor, containing three buttons: `Insert Code`, `Previous`, and `Next`. Below the buttons, the text `Suggestion 2 of 2 from CodeWhisperer` is visible. The code in the editor is:

```
1 package com.example.demo;
2
3 class Main {
4     //add two numbers
5     public static int add(int a, int b) {
6     }
7 }
```

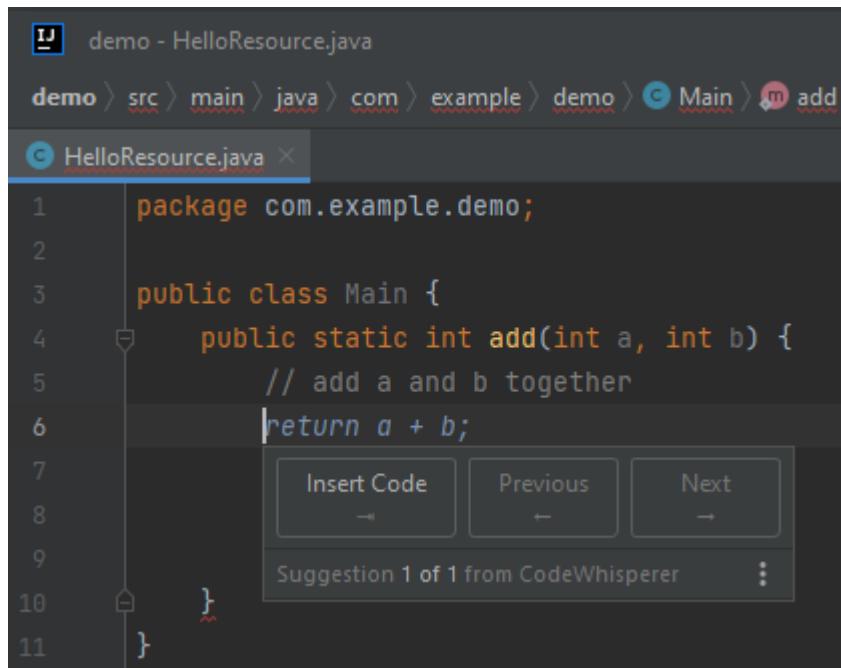
2. In the image below, the user has accepted the Amazon Q suggestion for a function signature. Accepting the suggestion automatically advanced the cursor and Amazon Q has made a new suggestion for the function body.



A screenshot of a Java code editor showing a file named `HelloResource.java`. The code defines a class `Main` with a static method `add` that returns the sum of two integers. A tooltip from Amazon Q is displayed over the opening brace of the `add` method, containing three buttons: `Insert Code`, `Previous`, and `Next`. Below the buttons, it says "Suggestion 1 of 1 from CodeWhisperer".

```
1 package com.example.demo;
2
3 public class Main {
4     // add two numbers
5     public static int add(int a, int b) {
6         return a + b;
7     }
8 }
9
```

3. In the image below, a user input a comment in the body of the function prior to accepting a suggestion from Amazon Q. On the following line, Amazon Q has generated a new suggestion based on the content of the comment.



A screenshot of a Java code editor showing the same `HelloResource.java` file. The user has added a comment `// add a and b together` before the `return` statement. A tooltip from Amazon Q is displayed over the `return` statement, containing the same three buttons: `Insert Code`, `Previous`, and `Next`. Below the buttons, it says "Suggestion 1 of 1 from CodeWhisperer".

```
1 package com.example.demo;
2
3 public class Main {
4     public static int add(int a, int b) {
5         // add a and b together
6         return a + b;
7     }
8 }
9
```

In this example, Amazon Q recommends a full function after the user types part of the signature.

```
examplebucketname = "example-bucket-1"
```

## Using Amazon Q Developer for block completion

Block completion is used to complete your if/for/while/try code blocks.

C

```
9
10 // function to pop the stack
11 int pop(Stack *stack) {
12     if (stack->top == -1) {
13         printf("Stack is empty\n");
14         return -1;
15     }
16     return stack->array[stack->top--];
17 }
18
19 // function to push the stack
20 void push(Stack *stack, int data) {
21 }
```

## C++

A screenshot of a code editor showing a C++ file. The code is as follows:

```
33     bool AwsDoc::RDS::describeDBInstance(const Aws::String &dbInstanceIdentifier,
34                                         Aws::RDS::Model::DBInstance &instanceResult,
35                                         const Aws::RDS::RDSClient &client) {
36
37     }
38
39 }
```

The cursor is at the end of the closing brace on line 38. A dropdown menu is open above the cursor, showing suggestions for the completion of the if statement. The suggestions include:

- if (dbInstanceIdentifier == "my-instance") {
- if (dbInstanceIdentifier != "my-instance") {
- if (dbInstanceIdentifier == "my-instance") {
- if (dbInstanceIdentifier != "my-instance") {

## Java

In the example below, a user enters the signature of an `if` statement. The body of the statement is a suggestion from Amazon Q.

A screenshot of a code editor showing a Java file named `example.java`. The code is as follows:

```
1 public class Main {
2
3     |
4
5     }
6
```

The cursor is at the end of the opening brace on line 5. A dropdown menu is open above the cursor, showing suggestions for the completion of the if statement. The suggestions include:

- if (true) {
- if (false) {
- if (true) {
- if (false) {

## C#

In the image below, Amazon Q recommends a way to complete the function.



## TypeScript

In the image below, Amazon Q recommends a way to complete the function.

A screenshot of a code editor showing a portion of a TypeScript file named index.ts. The code is as follows:

```
ts index.ts 2 ×
ts index.ts > [?] uploadFile
1   import { S3Client } from "@aws-sdk/client-s3";
2
3   const client = new S3Client({});
4
5   /**
6    * Upload local file to bucket
7   */
8   export const uploadFile = async (
```

A completion suggestion, represented by a yellow bracket, is shown at the end of the line where the function returns.

## Python

In this example, Amazon Q recommends a block of code, based on the context.

A screenshot of a code editor showing a portion of a Python file. The code is as follows:

```
examplebucketname = "example-bucket-1"

def print_bucket_contents(bucket_name):
    ...
    Print the contents of a bucket.
    ...
    print(f"Printing bucket contents for bucket {bucket_name}")
    for obj in s3.Bucket(bucket_name).objects.all():
        print(obj)
```

A completion suggestion, represented by a yellow bracket, is shown at the end of the line where the function returns.

## Using Amazon Q Developer for Docstring, JSDoc, and Javadoc completion

Amazon Q can help you generate or complete documentation inside your code.

### C++

```
7  /// <summary>
8  /// This example shows how to attach a policy to an IAM role.
9  /// </summary>
10 /// <param name="roleName">
11 bool AwsDoc::IAM::putRolePolicy(
12     const Aws::String &roleName,
13     const Aws::String &policyName,
14     const Aws::String &policyDocument,
15     const Aws::Client::ClientConfiguration &clientConfig) {
16     Aws::IAM::IAMClient iamClient(clientConfig);
17     Aws::IAM::Model::PutRolePolicyRequest request;
18
19     request.SetRoleName(roleName);
20     request.SetPolicyName(policyName);
21     request.SetPolicyDocument(policyDocument);
22
23     Aws::IAM::Model::PutRolePolicyOutcome outcome = iamClient.PutRolePolicy(request);
24     if (!outcome.IsSuccess()) {
25         std::cerr << "Error putting policy on role. " <<
26             outcome.GetError().GetMessage() << std::endl;
27 }
```

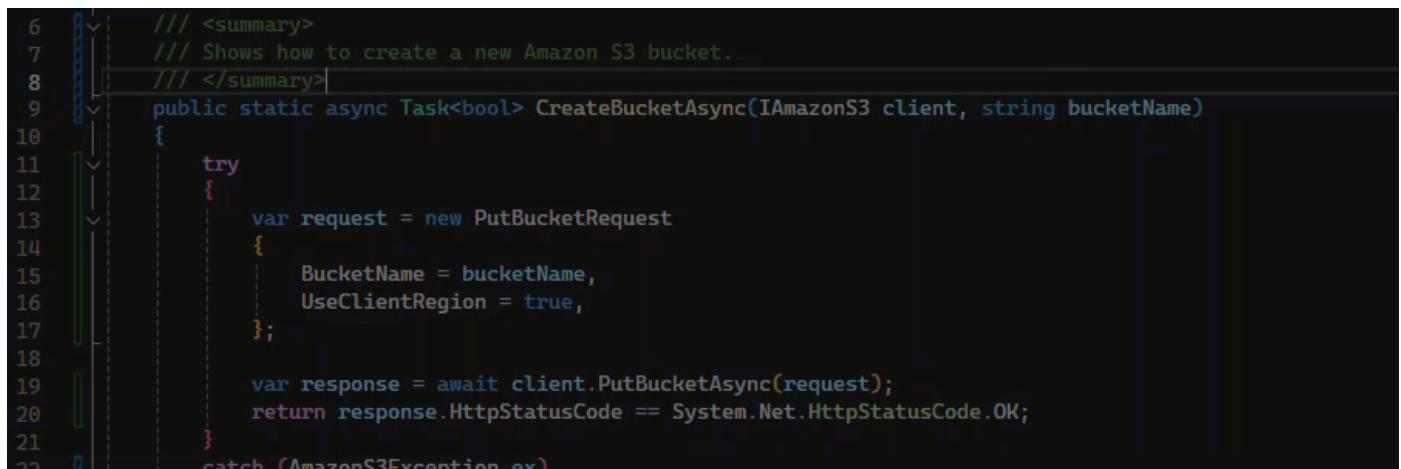
### Javascript

In this example, Amazon Q fills in JSDoc parameters based on existing constants.

```
1 import {PutObjectCommand, S3Client} from "@aws-sdk/client-s3";
2
3 const client = new S3Client({});
4
5 /**
6 * @param {Object} params
7 */
8 export const putObject = async (bucketName, key, body) => {
9     const params = {
10         Bucket: bucketName,
11         Key: key,
12         Body: body,
13     };
14     return client.send(new PutObjectCommand(params));
15 }
```

### C#

In this example, Amazon Q fills in JSDoc parameters based on existing constants.

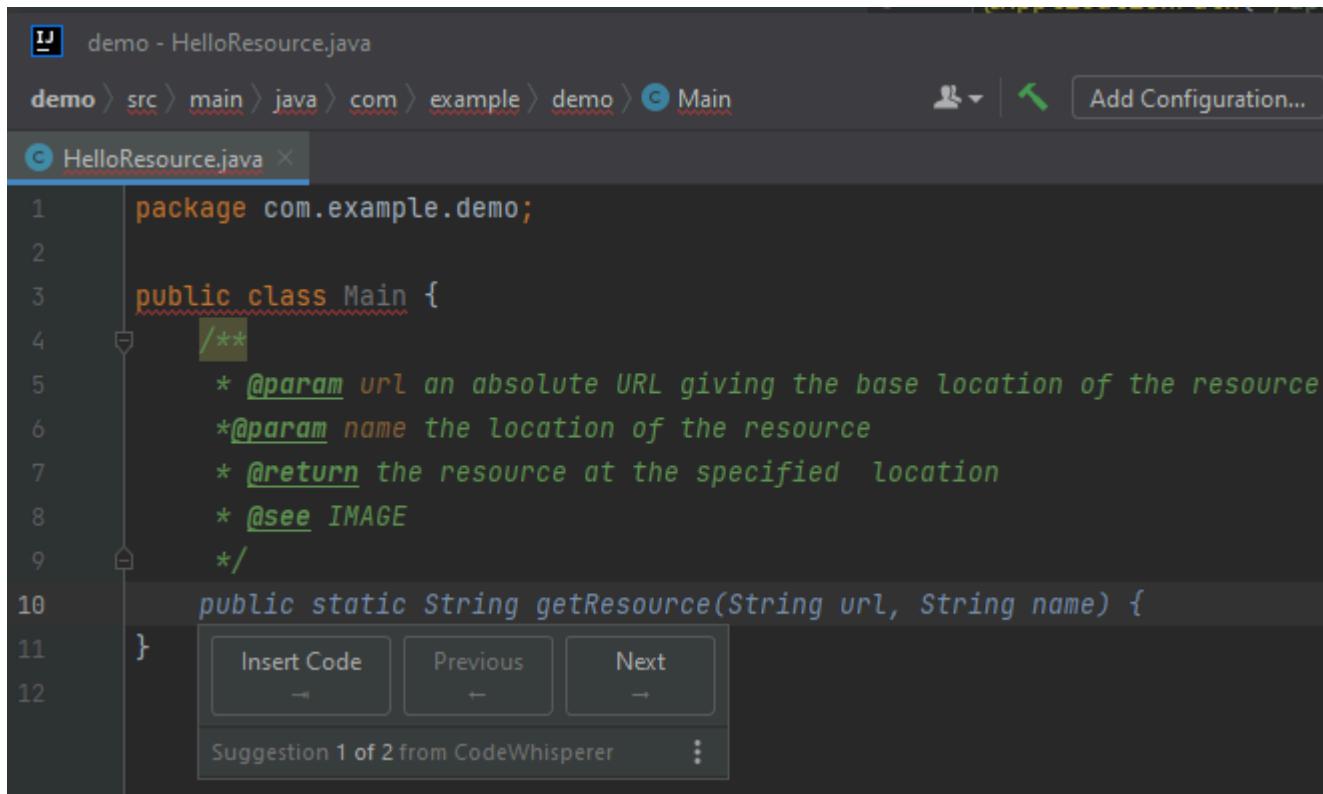


```
6     /// <summary>
7     /// Shows how to create a new Amazon S3 bucket.
8     /// </summary>
9     public static async Task<bool> CreateBucketAsync(IAmazonS3 client, string bucketName)
10    {
11        try
12        {
13            var request = new PutBucketRequest
14            {
15                BucketName = bucketName,
16                UseClientRegion = true,
17            };
18
19            var response = await client.PutBucketAsync(request);
20            return response.HttpStatusCode == System.Net HttpStatusCode.OK;
21        }
22        catch (AmazonS3Exception ex)
```

## Java

The following example is adapted from [an example on the Oracle website](#).

In the image below, the user has entered a docstring. Amazon Q has suggested a function to complete the docstring.



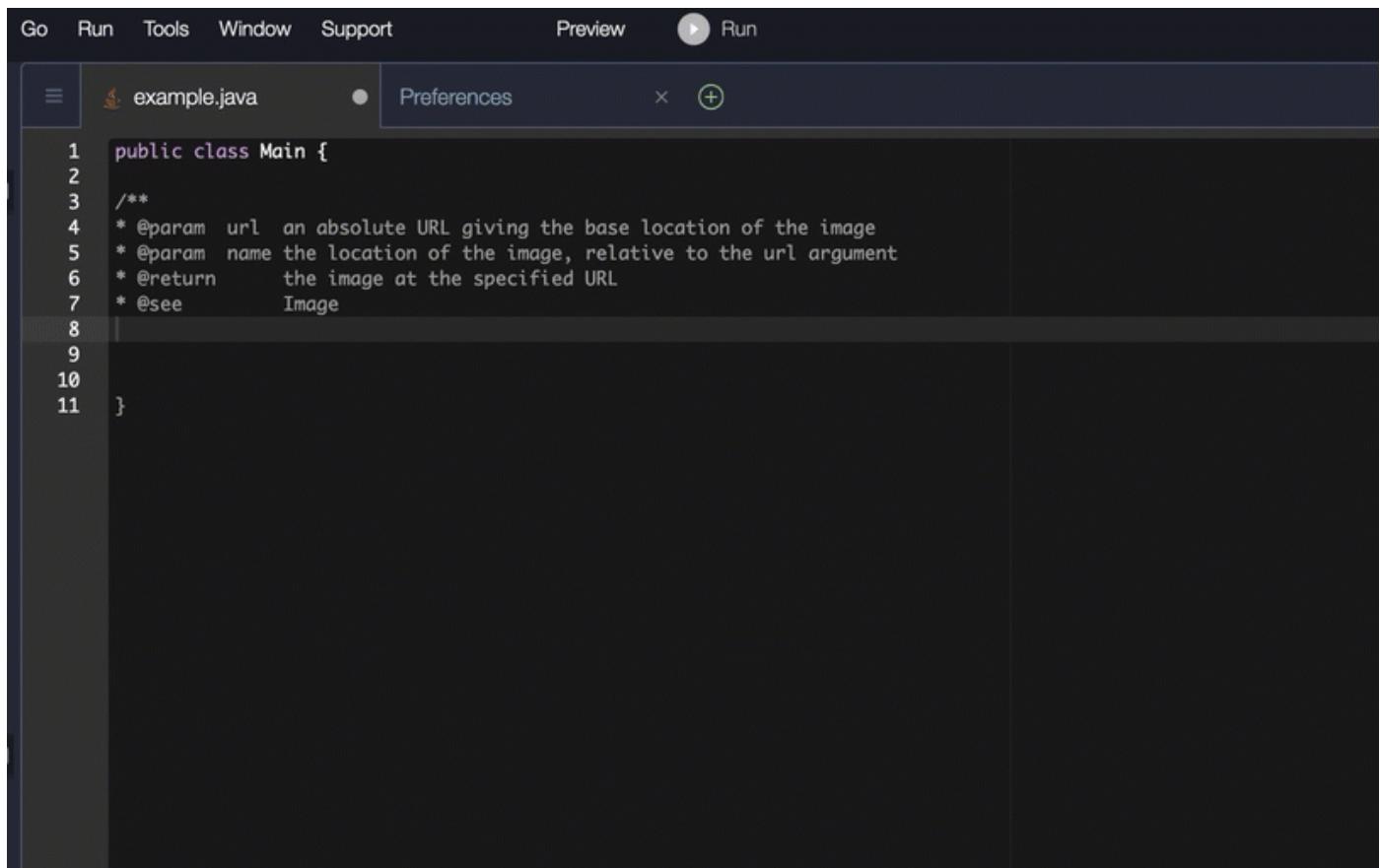
```
demo - HelloResource.java
demo > src > main > java > com > example > demo > Main
HelloResource.java x
package com.example.demo;

public class Main {
    /**
     * @param url an absolute URL giving the base location of the resource
     * @param name the location of the resource
     * @return the resource at the specified location
     * @see IMAGE
     */
    public static String getResource(String url, String name) {
}
```

Insert Code ← Previous → Next  
Suggestion 1 of 2 from CodeWhisperer :

The following example is adapted from [an example on the Oracle website](#).

In the example below, in Java, the user enters a docstring. Amazon Q suggests a function to process the docstring.

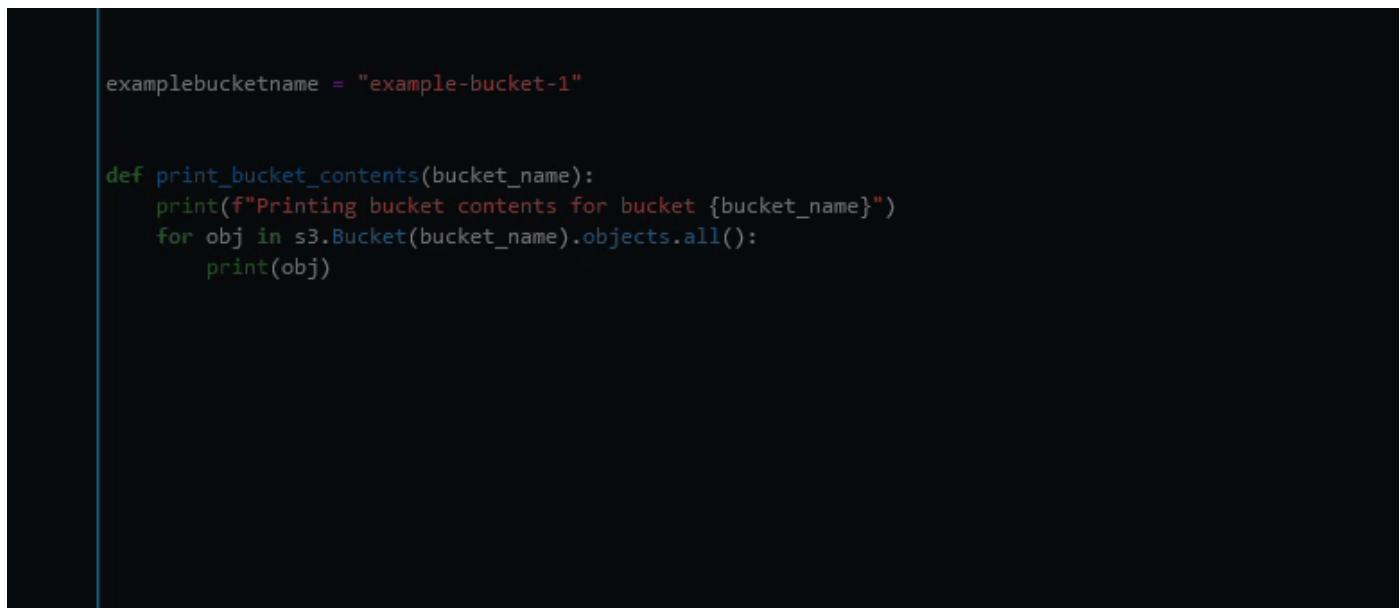


The screenshot shows the Amazon Q developer interface with a dark-themed code editor. At the top, there's a navigation bar with links for Go, Run, Tools, Window, Support, Preview, and Run. Below the navigation bar, the title bar displays "example.java" and "Preferences". The main area contains the following Java code:

```
1 public class Main {  
2  
3     /**  
4      * @param url an absolute URL giving the base location of the image  
5      * @param name the location of the image, relative to the url argument  
6      * @return the image at the specified URL  
7      * @see Image  
8  
9  
10 }  
11 }
```

## Python

In this example, Amazon Q recommends a Docstring, based on the surrounding context.



The screenshot shows the Amazon Q developer interface with a dark-themed code editor. The code editor displays the following Python script:

```
examplebucketname = "example-bucket-1"  
  
def print_bucket_contents(bucket_name):  
    print(f"Printing bucket contents for bucket {bucket_name}")  
    for obj in s3.Bucket(bucket_name).objects.all():  
        print(obj)
```

## Using Amazon Q Developer for line-by-line recommendations

Depending on your use case, Amazon Q may not be able to generate an entire function block in one recommendation. However, Amazon Q can still provide line-by-line recommendations.

### Go and GoLand

In this example, Amazon Q provides line-by-line recommendations.

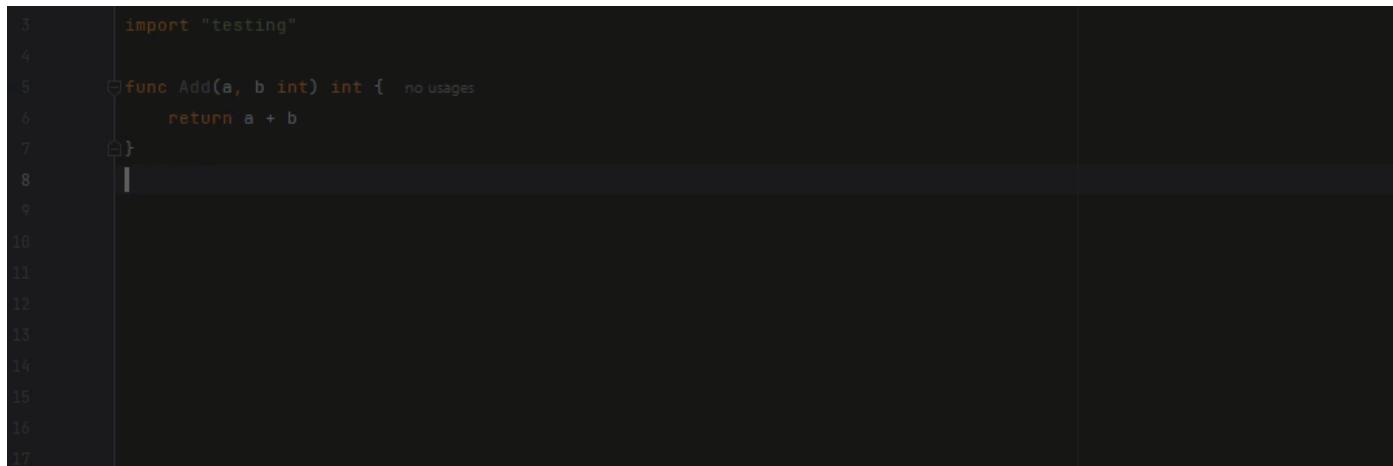


A screenshot of the GoLand IDE interface. It shows a code editor with a dark theme containing a Go function definition. The code is as follows:

```
10 func ListBuckets() { no usages
11     var err error
12     cfg, err := config.LoadDefaultConfig(context.TODO())
13     if err != nil {
14         panic("configuration error, " + err.Error())
15     }
16     s3Client := s3.NewFromConfig(cfg)
17 }
```

The line `s3Client := s3.NewFromConfig(cfg)` has a yellow lightbulb icon next to it, indicating a suggestion or error. A vertical line from the top of the code editor extends down to the bottom of the screenshot, representing the height of the IDE window.

Here is another example of line-by-line recommendations, this time with a unit test.



A screenshot of the CLion IDE interface. It shows a code editor with a dark theme containing a C++ function definition and a unit test import statement. The code is as follows:

```
3 import "testing"
4
5 func Add(a, b int) int { no usages
6     return a + b
7 }
```

The line `return a + b` has a vertical bar and a small lightbulb icon next to it, indicating a suggestion or error. A vertical line from the top of the code editor extends down to the bottom of the screenshot, representing the height of the IDE window.

### C++ and CLion

In this example, Amazon Q provides line-by-line recommendations.

```
34  
35     bool CreateBucket(const Aws::String &bucketName,  
36                           const Aws::Client::ClientConfiguration &clientConfig) {  
37         |  
38     }  
39  
40  
41  
42  
43  
44  
45  
46
```

## Python

In the following image, the customer has written an initial comment indicating that they want to publish a message to an Amazon CloudWatch Logs group. Given this context, Amazon Q is only able to suggest the client initialization code in its first recommendation, as shown in the following image.

```
# Publish a message to a CloudWatch Logs Group
```

```
client = boto3.client('logs')
```

However, if the user continues to request line-by-line recommendations, Amazon Q also continues to suggest lines of code based on what's already been written.

```
# Publish a message to a CloudWatch Logs Group  
client = boto3.client('logs')  
response = client.put_log_events(  
    |
```

```
        logGroupName='VPCFlowLogs',
```

### Note

In the example above, VPCFlowLogs may not be the correct constant value. As Amazon Q makes suggestions, remember to rename any constants as required.

Amazon Q can eventually complete the entire code block as shown in the following image.

```
# Publish a message to a CloudWatch Logs Group
client = boto3.client('logs')
response = client.put_log_events(
    logGroupName='VPCFlowLogs',
    logStreamName='VPCFlowLogs',
    logEvents=[
        {
            'timestamp': int(round(time.time() * 1000)),
            'message': json.dumps(event)
        }
    ]
)
```

No recommendations

In this example, Amazon Q provides recommendations, one line at a time.

```
role = get_execution_role()

sagemaker_session = sage.Session()
bucket = sagemaker_session.default_bucket()
runtime = boto3.client("runtime.sagemaker")
s3 = boto3.resource("s3")
```

## Transforming code in the IDE with Amazon Q Developer

Amazon Q Developer can transform your code in integrated development environments (IDEs) by performing automated language upgrades and conversions. You provide the code to be transformed, and Amazon Q generates changes that you can review and apply to your files.

To get started, install Amazon Q in an IDE that supports transformations. Then, see the topic for the type of transformation you'd like to perform with Amazon Q.

For more information on IDEs that support transformation and how to install Amazon Q, see [Using Amazon Q Developer in the IDE](#).

### Topics

- [Transforming Java applications with Amazon Q Developer](#)

# Transforming Java applications with Amazon Q Developer

Amazon Q supports the following types of transformations for Java applications:

- Java language and dependency version upgrades
- Embedded SQL conversion for Oracle to PostgreSQL database migration

To get started, see the topic for the type of transformation you'd like to perform.

## Topics

- [Upgrading Java versions with Amazon Q Developer](#)
- [Converting embedded SQL in Java applications with Amazon Q Developer](#)
- [Transforming code on the command line with Amazon Q Developer](#)
- [Troubleshooting issues with Java transformations](#)

## Upgrading Java versions with Amazon Q Developer

With the Amazon Q Developer Agent for code transformation, you can upgrade your Java applications to newer language versions in the integrated development environment (IDE). Changes Amazon Q can make to upgrade your code include updating deprecated code components and APIs as well as upgrading libraries, frameworks, and other dependencies in your code.

To transform your code, Amazon Q first builds your code in the source language version and verifies that it has the information necessary to perform the transformation. After Amazon Q successfully transforms your code, you verify and accept the changes in your IDE. For more information about how Amazon Q transforms your code, see [How Amazon Q Developer transforms code for Java language upgrades](#).

## Topics

- [Supported Java upgrades and IDEs](#)
- [Step 1: Prerequisites](#)
- [Step 2: Configure your project](#)
- [Step 3: Transform your code](#)
- [How Amazon Q Developer transforms code for Java language upgrades](#)

## Supported Java upgrades and IDEs

Amazon Q currently supports the following Java upgrades:

- Upgrade Java 8 and Java 11 code to Java 17 code
- Upgrade Java 17 code with up to date libraries and other dependencies

Amazon Q supports Java upgrades in the following IDEs:

- Modules in JetBrains IDEs
- Projects and workspaces in Visual Studio Code

### Step 1: Prerequisites

Before you continue, make sure you've completed the steps in [Set up Amazon Q in your IDE](#).

Make sure that the following prerequisites are met before you begin a Code Transformation job:

- Your project is written in a supported Java version and is built on Maven.
- Your project successfully builds with Maven in your IDE. Amazon Q Developer Agent for code transformation supports Maven 3.8 or later.
- Your project source JDK is available locally and is the version of your source code. For example, if you are transforming Java 8 code, your local JDK installation should be JDK 8.
- Your project builds in 55 minutes or less.
- Your project is configured correctly, and the correct JDK version is specified. For more information, see [Step 2: Configure your project](#).
- Your project doesn't require access to resources on your private network, including a virtual private cloud (VPC) or on-premise network. For example, if your project contains unit tests that connect to a database in your network, the transformation will fail.
- Your project doesn't use plugins that package languages other than Java in your Java project. For example, if your project uses the [frontend-maven-plugin](#) for executing front-end JavaScript code in addition to your Java source code, the transformation will fail.
- Your local network allows uploads to Amazon S3 buckets that Amazon Q uses to transform your code. For more information, see [Allow access to Amazon S3 buckets in data perimeters](#).
- Your minimum compiler version for the maven-compiler-plugin is v13.13.0.

## Step 2: Configure your project

To configure your project, use the following information for the IDE you're using.

### Configure a project in JetBrains

To configure your project in JetBrains, you might need to specify the following project and module settings.

If your modules use the same JDK and language level as your project, you don't need to update module settings.

- Project SDK – The JDK used to compile your project.
- Project language level – The Java version used in your project.
- Module SDK – The JDK used to compile your module.
- Module language level – The Java version used in your module.
- Maven Runner JRE – The JDK you build your module with.

### Update project and module settings

To update your SDK and language level settings for your project or module, complete the following steps:

1. From your JetBrains IDE, choose **File** and then **Project Structure**.
2. The Project Structure window opens. Under **Project Settings**, choose **Project**.
  - a. To update your project JDK, choose from the dropdown list next to **SDK**.
  - b. To update your project language, choose from the dropdown next to **Language level**.
3. Under **Project Settings**, choose **Modules**.
  - a. To update your module JDK, choose from the dropdown list next to **SDK**.
  - b. To update your module language, choose from the dropdown next to **Language level**.

For more information, see [Project structure settings](#) and [Module structure settings](#) in the JetBrains documentation.

### Update Maven settings

To update your Maven Runner JRE, complete the following steps:

1. From your JetBrains IDE, choose the gear icon, and then choose **Settings** in the menu that appears.
2. In the **Settings** window, choose **Build, Execution, Deployment**, then **Build Tools**, then **Maven**, and then **Runner**.
3. In the JRE field, choose the JDK used to build the module you're transforming.

## Configure a project in VS Code

To configure your project in VS Code, your project must contain the following:

- A pom.xml file in the project root folder
- A .java file in the project directory

If your project contains a Maven wrapper executable (mvnw for macOS or mvnw.cmd for Windows), make sure it's at the root of your project. Amazon Q will use the wrapper, and no other Maven configuration is necessary.

If you aren't using a Maven wrapper, install Maven. For more information, see [Installing Apache Maven](#) in the Apache Maven documentation.

After installing Maven, add it to your PATH variable. For more information, see [How do I add Maven to my PATH?](#) Your Java runtime variable should also be pointing to a JDK and not to a JRE. To confirm your configuration is correct, run mvn -v. The output should show your Maven version and the runtime variable pointing to the path to your JDK.

## Step 3: Transform your code

Before you transform your own code, you might want to test that your IDE is setup correctly by transforming a sample project. Following is a sample GitHub project that is eligible for code transformation: <https://github.com/aws-samples/aws-appconfig-java-sample>.

To test your IDE setup, download and unzip the sample project, and complete the following steps for your IDE. If you are able to view the proposed changes and transformation summary, you are ready to transform your own code project. If the transformation fails, your IDE is not configured correctly. To address configuration issues, review [Step 2: Configure your project](#) and [Troubleshooting](#).

### Note

If you navigate away from your IDE before the transformation starts, the transformation will fail and you will have to restart.

To upgrade the language version of your code project or module, complete the following steps for your IDE.

#### JetBrains

1. Open the module that you want to upgrade in JetBrains. Make sure you've successfully built your project in the IDE.
2. Choose the Amazon Q logo, and then enter **/transform** in the Amazon Q chat panel that opens.
3. A **Transform your application** pop-up appears. Choose the project that you want to upgrade from the dropdown list, and then choose **Transform**.
4. Amazon Q begins the transformation. You can view progress on the **Transformation details** tab.
5. After the transformation is complete, you can verify the upgraded code before updating your project. To view the new code, go to the **Transformation details** tab and then choose **View diff**. In the **Apply patch** window that appears, choose a file to open a diff view with your source code and upgraded code.
6. To accept the changes that Amazon Q made, choose **View diff** to open the **Apply patch** window. Select all the updated files, and choose **OK** to update your project in place.
7. To get details about how your code was upgraded and suggested next steps, on the **Transformation details** tab, choose **View transformation summary**.

#### Visual Studio Code

1. Open the project or workspace that you want to upgrade in VS Code. Make sure that you've successfully built your project in the IDE.
2. Choose the Amazon Q logo, and then enter **/transform** in the Amazon Q chat panel that opens.
3. Choose the project that you want to upgrade from the search bar at the top of the IDE.

4. If Amazon Q can't find the version of your source code, it prompts you to choose your code version. Choose the version that your source code is written in, and then choose **Transform** in the pop-up to proceed.
5. If prompted, enter the JAVA\_HOME path to your JDK. For more information, see [Configure your VS Code project](#).
6. Amazon Q begins the transformation. You can view progress on the **Transformation Hub** tab.
7. After the transformation is complete, the **Proposed Changes** tab opens. To verify the upgraded code before updating your project, choose **Download proposed changes**. Choose a file to open a diff view with your source code and upgraded code.
8. To accept the changes Amazon Q made, go to the **Proposed Changes** tab and choose **Accept**.
9. To get details about how your code was upgraded and suggested next steps, on the **Transformation Hub**, choose the **Views and More Actions** ellipsis button, and then choose **Show Transformation Summary**.

## How Amazon Q Developer transforms code for Java language upgrades

To transform your code, the Amazon Q Developer Agent for code transformation generates a transformation plan that it uses to upgrade the code language version of your project. After transforming your code, it provides a transformation summary and a file diff for you to review changes before accepting them. The following sections provide more details on how Amazon Q performs the transformation.

### Building your code and creating a transformation plan

To begin transforming your code, Amazon Q builds your project locally and generates a build artifact that contains your source code, project dependencies, and build logs.

After generating the build artifact, Amazon Q builds your code in a secure build environment and creates a transformation plan, which is customized to the project or module you're upgrading. The transformation plan outlines the specific changes Amazon Q will attempt to make, including new dependency versions, major code changes, and suggested replacements for deprecated code. These changes are based on the preliminary build of your code, and might change during the transformation.

Before the transformation begins, you have the option to split up the changes Amazon Q proposes into multiple diffs. This allows you to update and test your code with fewer changes at a time, and will require you to accept changes incrementally after the transformation is complete.

## Transforming your code

To transform your code, Amazon Q attempts to upgrade your code based on the proposed changes in the transformation plan. As it makes changes, it re-builds and runs existing unit tests in your source code to iteratively fix any encountered errors.

Amazon Q attempts to make the following changes when upgrading your code:

- Update deprecated code components according to Java 17 recommendations
- Upgrade popular libraries and frameworks to a version compatible with Java 17. This includes updating the following libraries and frameworks to their latest available major versions:
  - Apache Commons IO
  - Apache HttpClient
  - bc-fips
  - Cucumber-JVM
  - Hibernate
  - jackson-annotations
  - JakartaEE
  - Javax
  - javax.servlet
  - jaxb-api
  - jaxb-impl
  - jaxen
  - jcl-over-slf4j
  - json-simple
  - jsr305
  - junit
  - junit-jupiter-api
  - Log4j
  - **Micronaut**

- Mockito
- mockito-core
- Okio
- PowerMockito
- Quarkus
- slf4j
- slf4j-api
- Spring Boot
- Spring Framework
- Spring Security
- Swagger
- testng

## Reviewing the transformation summary and accepting changes

After the transformation is complete, Amazon Q provides a transformation summary with details about the changes it made, including the status of the final build which indicates whether your entire project was upgraded. You can also view a build log summary to understand any issues that prevented Amazon Q from building your code in the upgraded version.

The transformation summary additionally includes the differences between the changes proposed in the transformation plan and the changes Amazon Q ultimately made to upgrade your code, and any additional changes that weren't in the original plan.

After you review the transformation summary, you can view the changes Amazon Q is proposing in a file diff view. Any code changes Amazon Q suggests will not affect your current project files until you accept the changes. The transformed code is available up to 24 hours after the transformation completes.

If you chose to have Amazon Q split up the changes into multiple diffs, it will provide one or more diffs with changes, depending on the upgrades required for your application. If applicable to your application, the changes are categorized in the following groups:

- Minimum compatible library versions to Java 17: Upgrade dependencies, including Springboot, JUnit, and PowerMockito, to the minimum compatible versions in Java 17.

- Popular enterprise specifications application frameworks: Upgrade popular enterprise and application frameworks like Jakarta EE, Hibernate, and Micronaut 3.
- HTTP client utilities web frameworks: Upgrade HTTP client libraries, Apache Commons utilities, and Struts frameworks.
- Minimum compatible library versions to Java 17: Upgrade dependencies, including Springboot, JUnit, and PowerMockito, to the minimum compatible versions in Java 17.
- Testing tools frameworks: Upgrade testing tools like ArchUnit, Mockito, and TestContainers and build tools like Jenkins and Maven Wrapper.
- Miscellaneous processing documentation: Multiple types of upgrades, including ORMs, XML processing, and API documentation like Swagger to SpringDoc/OpenAPI.
- Deprecated APIs, outdated dependencies, and formatting: Replace deprecated APIs, make additional dependency version upgrades, and format code changes

Amazon Q will provide one diff file at a time, and you must accept changes in a diff to review the following diff. If you reject changes in a diff, you won't be able to review any other diffs. After you accept changes from one diff, you can review the updates to your code and test the changes before continuing to accept subsequent changes.

Note that there may be minor version downgrades across the diffs. For example, you may see a dependency version upgraded to a newer version in the first diff, and in a subsequent diff see an older version of the dependency. This is because Amazon Q might need to adjust versions to accommodate changes made in each diff. After you accept changes from all the diffs, your code will contain the most recent reliable versions.

The title of the diff will indicate whether the proposed changes will result in a full or partial transformation. If the diff indicates the transformation was only partially successful, then it does not include all necessary changes for a complete transformation and a successful code build. To get the most upgraded version of your code, accept all the code diffs, including partially successful ones. Then, you can manually update your code to address the errors Amazon Q wasn't able to fix.

### **Completing partially successful transformations**

Depending on the complexity and specifics of your codebase, there might be instances where the transformation is partially successful. This means that Amazon Q was able to transform only certain files or areas of code in your project. In this case, you have to manually update the remaining code for your project to be buildable in the updated language version.

To help transform the rest of your code, you can use Amazon Q chat in the IDE. You can ask Amazon Q to review the partially updated files and provide new code to address issues, such as compilation errors. You can also use features like [/dev](#) and [@workspace](#) to include more of your project as context and get suggestions for multiple files at a time.

## Converting embedded SQL in Java applications with Amazon Q Developer

The Amazon Q Developer agent for code transformation in the IDE can help you convert embedded SQL to complete Oracle to PostgreSQL database migration with AWS Database Migration Service (AWS DMS).

AWS DMS is a cloud service that makes it possible to migrate relational databases, data warehouses, NoSQL databases, and other types of data stores. DMS Schema Conversion in AWS DMS helps you convert database schemas and code objects that you can apply to your target database. For more information, see [What is AWS Database Migration Service?](#) in the *AWS Database Migration Service User Guide*.

When you use AWS DMS and DMS Schema Conversion to migrate a database, you might need to convert the embedded SQL in your application to be compatible with your target database. Rather than converting it manually, you can use Amazon Q in the IDE to automate the conversion. Amazon Q uses metadata from a DMS Schema Conversion to convert embedded SQL in your application to a version that is compatible with your target database.

Currently, Amazon Q can convert SQL in Java applications for Oracle databases migrating to PostgreSQL. You will only see the option to transform SQL code in the IDE if your application contains Oracle SQL statements. See the prerequisites for more information.

### Step 1: Prerequisites

Before you continue, make sure you've completed the steps in [Set up Amazon Q in your IDE](#).

Before you begin a code transformation job for SQL conversion, make sure the following prerequisites are met:

- You are migrating a Java application with embedded SQL from an Oracle database to a PostgreSQL database. Your application must contain Oracle SQL statements for it to be eligible for transformation.
- You have completed the process for converting your database schema using AWS DMS Schema Conversion. For more information, see [Migrating Oracle databases to Amazon RDS for PostgreSQL with DMS Schema Conversion](#) in the *Database Migration Guide*.

- After schema conversion is complete, you have downloaded the migration project file from the AWS DMS console.

## Step 2: Configure your application

To convert your embedded SQL code, your Java project must contain at least one .java file.

If you are using a JetBrains IDE, you must set the SDK field in Project Structure settings to the applicable JDK. For information on configuring Project Structure settings, see [Project structure settings](#) in the JetBrains documentation.

## Step 3: Convert embedded SQL

To convert the embedded SQL code in your Java application to a format that is compatible with your PostgreSQL target database, complete the following steps:

1. In your IDE where Amazon Q is installed, open the Java codebase that contains the embedded SQL you need to convert.
2. Choose the Amazon Q icon to open the chat panel.
3. Enter **/transform** in the chat panel.
4. If your Java application is eligible for SQL conversion, Amazon Q will prompt you to choose which type of transformation you'd like to perform. Enter **SQL conversion**.
5. Amazon Q prompts you to upload the schema metadata file you retrieved from Amazon S3. In the chat, Amazon Q provides instructions for retrieving the file.
6. Amazon Q prompts you to provide the project that contains the embedded SQL as well as the database schema file. Choose the appropriate files from the dropdown menus in the chat panel.
7. Confirm the details Amazon Q retrieved from the database schema are accurate.
8. Amazon Q begins converting your SQL code. This might take a few minutes.
9. After Amazon Q converts the SQL code, it provides a diff with any updates it has made to your files. Review the changes in the diffs, and then accept the changes to update your code.

Amazon Q also provides a transformation summary with details about the changes it made.

10. After updating your code, return to the AWS DMS console to verify the new SQL is compatible with the migrated database.

## Transforming code on the command line with Amazon Q Developer

Transforming code with Amazon Q on the command line is in preview, and is subject to change.

You can transform your Java applications from the command line with the Amazon Q Developer command line transformation tool. You provide the path to your source code, and Amazon Q upgrades your Java language version, including deprecated code, libraries, and other dependencies. For more information on the types of upgrades Amazon Q makes to transform Java applications, see [How Amazon Q transforms your code](#).

In addition to the transformation Amazon Q runs, you have the option to create custom transformations to make changes unique to your organization's code. For more information, see [Customizing transformations on the command line with Amazon Q Developer](#).

To transform your code, Amazon Q creates a new branch in your repository where it commits the code changes. Throughout the transformation, Amazon Q runs builds on your local environment to verify changes. For more information, see [Building code in your local environment](#). When the transformation is complete, you can merge the branch into your original branch to incorporate the changes into your codebase.

To get started, install the command line tool and authenticate, and then see the commands to configure and start a transformation.

### Topics

- [Building code in your local environment](#)
- [Commands](#)
- [Running a transformation on the command line with Amazon Q Developer](#)
- [Customizing transformations on the command line with Amazon Q Developer](#)

### Building code in your local environment

During a transformation, Amazon Q performs verification builds in your local environment. Amazon Q transforms your code on the server side in multiple steps. After each step, Amazon Q sends the code to your local environment to build and test the changes it made. The code is then sent back to the server side to continue the transformation.

The build in your local environment helps verify the transformed code by allowing Amazon Q to run tests that require access to private resources. To minimize security risks associated with building AI-generated code in your local environment, Amazon Q reviews and updates the code it generates to address security concerns.

## Commands

For step-by-step instructions for running these commands, see [Running a transformation on the command line with Amazon Q Developer](#).

To configure a transformation and authenticate to Amazon Q Developer Pro, run:

```
qct configure
```

To start a transformation, run:

```
qct transform --source_folder <path-to-folder>
```

To start a transformation with a [customization](#), run:

```
qct transform --source_folder <path-to-folder>
--custom_transformation_file <path-to-orchestrator-file>
```

To get help with transformations, run:

```
qct -h
```

## Running a transformation on the command line with Amazon Q Developer

Complete these steps to transform your code on the command line with the Amazon Q Developer command line tool.

### Prerequisites

Before you begin a transformation, see the [prerequisites for upgrading Java versions with Amazon Q](#).

In addition, the following prerequisites must be met:

- You are [subscribed to Amazon Q Developer Pro](#) and have a Start URL to access your subscription. You or your administrator can find the Start URL in the Amazon Q Developer console. For more information see, [Managing account details](#).
- You have Python installed on your command line environment. This is how you will install the command line tool. The minimum supported Python version is 3.12.
- You are running the transformation on macOS or Linux.
- The size of your application is 2 GB or smaller.
- If you are performing custom transformations, you have installed the ast-grep tool. To install and set up ast-grep, see <https://ast-grep.github.io/guide/quick-start.html#installation>

## Step 1: Install the tool

1. [Download the Amazon Q command line tool for transformations](#).
2. We recommend that you set up a virtual environment in Python to install the tool. To create a virtual environment, open a terminal window and run:

```
python -m venv qct-cli
```

3. To activate the virtual environment, run:

```
source qct-cli/bin/activate
```

4. To install the tool on your command line, run:

```
pip install amzn_qct_cli-0.1.0-py3-none-any.whl
```

5. To verify that the tool was installed, run:

```
which qct
```

## Step 2: Configure and authenticate

Before you can begin a transformation, you must provide configuration details and authenticate to your Amazon Q Developer Pro subscription with IAM Identity Center.

1. To start the configuration process, run the following command:

```
qct configure
```

- First, your Maven version is verified. If you have at least the minimum supported version, you will see the following output:

```
Running command: mvn --version at: path/to/current/directory  
Your Maven version is supported for transformations.
```

If you don't have a supported version of Maven, you must update it to continue. For more information, see the [Prerequisites](#).

- You are then prompted to enter a JDK path for Java 8, 11, and 17. You only need to specify the path to the JDK of the Java version you are upgrading.
- Next, you are prompted to provide the Start URL to authenticate to Amazon Q Developer Pro through OpenID Connect identity provider. For more information, see the [Prerequisites](#).
- Your configuration preferences are saved to a configuration.ini file.

### Step 3: Run a transformation

To transform your code, you provide the source code file, and optionally a customization file. For information on customizations and creating custom transformation files, see [Customizing transformations](#).

- Run the following command to start a transformation. Replace <path-to-folder> with the path to the folder with the code you're transforming.

```
qct transform --source_folder <path-to-folder>
```

- Amazon Q begins the transformation. It will output status updates throughout the transformation. When it's complete, Amazon Q provides the path where the transformation results, logs, and configuration files are outputted.

### Customizing transformations on the command line with Amazon Q Developer

#### Note

This feature is only available for transformations on the command line.

You can customize transformations by providing custom logic in the form of ast-grep rules that Amazon Q uses to make changes to your code. ast-grep is an abstract syntax tree tool that can be used to rewrite code. Amazon Q leverages ast-grep to run customized transformations. For more information, see [What is ast-grep?](#) in the ast-grep documentation.

Amazon Q performs the custom transformation locally. The custom transformation happens in addition to the Java upgrades in an Amazon Q transformation.

To configure a custom transformation, you provide two file types that specify the custom logic:

- An orchestrator file, where you define what custom transformations to run before the Amazon Q transformation, and which ones to run after
- One or more custom transformation files, where you define an ast-grep rule

After creating an orchestrator file and your custom transformation files, you can start a transformation job with the customization option and the path to your orchestrator file. Following is the command you run to start a transformation with a custom transformation:

```
qct transform --source_folder <path-to-folder>
  --custom_transformation_file <path-to-orchestrator-file>
```

## Orchestrator files

An orchestrator file is a YAML file where you provide the paths to the custom transformation files that Amazon Q will run, and specify when to run the rules (before or after the Amazon Q transformation).

The following fields are required in the transformation file:

- name
- description
- At least one of the following:
  - To run a custom transformation before the Amazon Q transformation, add the path to a custom transformation file under pre\_qct\_actions :
  - To run a custom transformation after the Amazon Q transformation, add the path to a custom transformation file under post\_qct\_actions :

Following is an example of the syntax in an orchestrator file:

```
name: custom_change_1
description: My collection of custom transformations to run before and after a
transformation.

pre_qct_actions:
  ast-grep:
    rules:
      - /path/to/custom-transformation3.yaml
      - /path/to/custom-transformation2.yaml

post_qct_actions:
  ast-grep:
    rules:
      - /path/to/custom-transformation3.yaml
```

## Custom transformation files

Custom transformation files are YAML files where you define the code changes you want Amazon Q to make in the form of an ast-grep rule. Amazon Q only supports ast-grep compatible rules for custom transformations.

Amazon Q can run custom transformations before or after it runs a transformation. See the following guidance on when custom transformation types should be run:

- Custom transformations you run before the Amazon Q transformation should focus on code preprocessing tasks. Your code must be compilable after the custom transformations are run in order to continue with the Amazon Q transformation.
- Custom transformations run after the Amazon Q transformation can involve tasks like upgrading internal libraries or other tasks related to private resources. If these tasks break the code build, Amazon Q can debug and fix issues that arise from the custom transformation.

Following is an example of a custom transformation file with an ast-grep rule:

```
id: no-unused-vars
language: java
rule:
  kind: local_variable_declaration
  all:
    - has:
        has:
```

```
        kind: identifier
        pattern: $IDENT
    - not:
        precedes:
            stopBy: end
        has:
            stopBy: end
            any:
                - { kind: identifier, pattern: $IDENT }
                - { has: {kind: identifier, pattern: $IDENT, stopBy: end} }
fix: ''
```

You can learn more about how this example works at <https://ast-grep.github.io/catalog/java/>.

## Troubleshooting issues with Java transformations

The following information can help you troubleshoot common issues when transforming Java applications with Amazon Q Developer.

### Topics

- [Why can't Amazon Q upload my project?](#)
- [Why are my Maven commands failing?](#)
- [How do I add Maven to my PATH?](#)
- [Why can't Amazon Q build my code?](#)
- [Why did my transformation fail after 55 minutes?](#)
- [Why can't I download my transformed code?](#)
- [How do I access code transformation logs?](#)
- [How do I find my transformation job ID?](#)

### Why can't Amazon Q upload my project?

If your project fails to upload, it's likely due to one of the following issues. See the topic that corresponds to the error you see from Amazon Q.

### Topics

- [Reduce project size](#)
- [Configure proxy settings in your IDE](#)

- [Allow access to Amazon S3](#)

## Reduce project size

To transform your code, Amazon Q generates a project artifact, which includes your source code, project dependencies, and build logs. The maximum project artifact size for a transformation job is 2 GB. If you get an error related to project artifact size, you must decrease the size of your project or try transforming a smaller project. You can view the size of your project artifact file in the code transformation logs. For more information, see [How do I access code transformation logs?](#)

## Configure proxy settings in your IDE

To transform your code, Amazon Q uploads your project artifact to a service-owned Amazon S3 bucket. Part of the upload process involves using SSL or TLS certificates to establish communication between Amazon S3 and your IDE. If you are using a proxy server, the SSL or TLS certificates used by your proxy server must be trusted, otherwise Amazon Q is not able to upload your project.

If you receive an error related to your proxy or certificates, you likely need to configure your IDE or operating system to trust your certificates or update other proxy settings.

### Note

You might also encounter issues unrelated to certificates if you are behind your organization's proxy server or firewall. If you complete the following procedures to configure your certificates and still have issues, contact your network administrator to ensure you are allowed to communicate with Amazon S3 from your IDE. For more information, see [Allow access to Amazon S3](#).

## Configure certificates in JetBrains

To configure your JetBrains IDE Java Runtime Environment (JRE) to trust the SSL or TLS certificates used by your proxy server, you must import the SSL or TLS certificates to the cacerts file in the JRE. The cacerts file is a file that contains trusted root certificates for secure connections such as HTTPS and SSL, and it's part of the JRE's security settings. To import a certificate, complete the following procedure.

**Note**

We recommend making a backup of the cacerts file before modifying it, as any mistakes can cause issues with secure connections.

1. Determine the path to the cacerts file in your JRE. The path of the cacerts file in the internal JRE shipped with your JetBrains IDE depends on the operating system and the version of the JetBrains IDE you're using.

Following are examples of paths to the cacerts file in common operating systems. Choose your operating system to see examples.

**Note**

<JetBrains Installation Folder> refers to the directory where JetBrains products are installed. This directory is typically chosen during the installation process. The jbr folder represents the JRE bundled with JetBrains IDEs, which is a specific version of the JRE tailored for use with JetBrains IDEs.

## Windows

The cacerts file path for a JetBrains IDE installed on Windows is:

```
<JetBrains Installation Folder>\jbr\bin\cacerts
```

For example, if you installed a JetBrains IDE on Windows in the default location, the path might be:

```
C:\Program Files\JetBrains\jbr\bin\cacerts
```

## macOS

The cacerts file path for a JetBrains IDE installed on macOS is:

```
/Applications/JetBrains Toolbox/<version>/JetBrains Toolbox.app/Contents/jbr/Contents/Home/lib/security/cacerts
```

For example, if you installed a JetBrains IDE on macOS in the default location, the path might be:

```
/Applications/JetBrains Toolbox/2022.3.4/JetBrains Toolbox.app/Contents/jbr/Contents/Home/lib/security/cacerts
```

## Linux

The cacerts file path for a JetBrains IDE installed on Linux is:

```
/opt/jetbrains/jbr/lib/security/cacerts
```

2. Determine the certificate you need to import to the cacerts file. The certificate file typically has a .cer, .crt, or .der file extension. If you aren't sure which certificates you need to add, contact your network administrator.
3. Import the certificate to the cacerts keystore. You can do this with the Java keytool command.
  - a. Open a command prompt and enter the following command:

```
keytool -import -alias <alias> -file <certificate_file> -keystore <path_to_cacerts>
```
  - b. For <alias>, you can add a name for the certificate you are importing to refer to it later. This option is optional.
  - c. For <certificate\_file>, specify the path to the certificate you are importing. This should be a path to the .cer, .crt, or .der file containing the certificate.
  - d. For <path\_to\_cacerts>, specify the path to the cacerts keystore file you saved in step 1. This is the file where you are importing the certificate.

For example, if you want to import a certificate named my\_certificate.cer into the cacerts keystore of the bundled JRE in IntelliJ IDEA on Windows, and you want to give the alias myalias to the certificate, the command might be:

```
keytool -import -alias myalias -file my_certificate.cer -keystore "C:\Program Files \JetBrains\IntelliJ IDEA 2022.3.2\jbr\bin\cacerts"
```

4. During the import process, you'll be prompted to enter the keystore password. The default password for the cacerts keystore is changeit.
5. After running the command, you'll be asked to trust the certificate. To confirm the certificate is trusted and complete the import, enter yes.
6. You might also need to add the certificates to the IDE itself, in addition to the JRE. For more information, see [Server Certificates](#) in the JetBrains documentation.

## Configure certificates in Visual Studio Code

To configure Visual Studio Code to trust the SSL or TLS certificates used by your proxy server, make sure you have configured the following proxy settings for your operating system.

### Configure certificates in Visual Studio Code on macOS

Configure the following proxy settings for Visual Studio Code on macOS.

#### Add certificates to your macOS keychain

If you haven't already, you must add the certificates used by your proxy server to your macOS keychain. For information on adding certificates to your keychain, see [Add certificates to a keychain using Keychain Access on Mac](#) in the Keychain Access User Guide.

#### Install the Mac CA VSCode extension

The [Mac CA VSCode extension](#) allows Amazon Q to access the certificates you added to Keychain Access on your Mac.

To install the extension:

1. Search for mac-ca-vscode in the VS Code extensions pane, and choose **Install**.
2. Restart VS Code.

#### Update proxy settings in VS Code on macOS

Update the following settings to make sure VS Code is configured properly for your proxy.

1. Open settings in VS Code.
2. Enter proxy in the search bar.

3. In the **Http: Proxy** field, add your proxy URL.
4. Deselect **Http: Proxy Strict SSL**.
5. In the **Http: Proxy Support** dropdown list, choose **on**.
6. In the settings search bar, enter `http.experimental.systemCertificatesV2`. Select **Http > Experimental: System Certificates V2**.

## Configure certificates in Visual Studio Code on Windows

Configure the following proxy settings for Visual Studio Code on Windows.

### Add certificate as a trusted root certificate on Windows

If you haven't already, you must add the certificates used by your proxy server to your Trusted Root Certification Authorities store on Windows. To add a certificate, complete the following procedure:

1. Open the search tool or a Run command window.
2. Enter the following to open the Certificate Manager tool:

```
certmgr.msc
```

3. Choose the **Trusted Root Certification Authorities** store.
4. Right-click **Certificates**, choose **All Tasks**, and then choose **Import....**
5. Follow the instructions given to import your proxy certificate.
6. After you've imported your certificate, confirm the certificate was added.

In the **Trusted Root Certification Authorities** store, double click **Certificates**. Right-click the certificate you added and choose **Properties**. Under **Certificate purposes**, the option **Enable all purposes for this certificate** should be selected.

### Install the Win-CA VSCode extension

The [Win-CA VSCode extension](#) allows Amazon Q to access the certificates you added to Trusted Root Certificates in Windows.

To install the extension:

1. Search for `win-ca` in the VS Code settings pane.

2. In the **Inject** dropdown list, choose **append**.

## Update proxy settings in VS Code on Windows

Update the following settings to make sure VS Code is configured properly for your proxy.

1. Open settings in VS Code.
2. Enter proxy in the search bar.
3. In the **Http: Proxy** field, add your proxy URL.
4. Deselect **Http: Proxy Strict SSL**.
5. In the **Http: Proxy Support** dropdown list, choose **on**.
6. In the settings search bar, enter `http.experimental.systemCertificatesV2`. Select **Http > Experimental: System Certificates V2**.
7. Restart VS Code.

## Allow access to Amazon S3

During a transformation, the Amazon Q Developer Agent for code transformation uploads your code to a service-owned Amazon S3 bucket. If your network or organization hasn't configured access to Amazon S3, Amazon Q isn't able to upload your project.

To ensure Amazon Q can upload your project, make sure your proxy configuration and other network components, such as Data Lost Prevention (DLP) policies, are configured to allow access to Amazon S3. You might also need to allowlist the Amazon S3 bucket where Amazon Q uploads your project. For more information, see [Data perimeters for Amazon Q resources](#).

If you transform a large project, DLP policies or other network components might cause delays and prevent a successful upload if they aren't configured to allowlist the Amazon S3 bucket. If you choose not to allowlist the bucket, you might need to transform a smaller project so that Amazon Q can upload it.

## Why are my Maven commands failing?

Following are Maven configuration issues that you might see in the JetBrains and Visual Studio Code IDEs. If you address the issues and still see Maven errors, there might be an issue with your project. Use the information in the error logs to address any issues with your project, and then try transforming your project again.

## Update Maven configuration in JetBrains

If a transformation fails in JetBrains due to Maven command issues, the error logs appear on the **Run** tab. Use the information in the logs to address the issue. Following are some issues that you might need to address:

- Make sure that your Maven home path is set to **Bundled**. Go to **Settings**, and then expand the **Build, Execution, Deployment** section. Expand the **Build Tools** section and then expand **Maven**. In the **Maven home path** dropdown list, choose **Bundled**.
- Make sure that the Java runtime environment (JRE) is using your project JDK. Go to **Settings**, and then expand the **Build, Execution, Deployment** section. Expand **Maven** and choose **Runner**. In the **JRE** dropdown list, choose **Use Project JDK**.
- Make sure that Maven is enabled. Go to **Settings** and choose **Plugins**. Search for Maven and choose the Maven plugin. If you see an **Enable** button, choose it to enable Maven.

## Update Maven configuration in Visual Studio Code

If a transformation fails in VS Code because of Maven command issues, a text file that contains the error logs opens in a new tab. Use the information in the logs to address the issue.

Make sure that you have configured either one of the following options:

- Your project contains a Maven wrapper in the project root folder
- A version of Maven supported by Amazon Q is available on your PATH

For more information, see [How do I add Maven to my PATH?](#)

### How do I add Maven to my PATH?

To transform your code in VS Code without using a Maven wrapper, you must install Maven and add it to your PATH variable.

To check if you have Maven installed correctly already, run `mvn -v` in a new OS terminal outside of Visual Studio Code. You should see an output with your Maven version.

If you get an output in your Visual Studio Code terminal but not in your OS terminal, or if the command isn't found, you need to add Maven to your PATH.

To add Maven to your PATH, follow the instructions for your machine.

## macOS

To add Maven to your macOS PATH, complete the following steps.

1. Locate your Maven installation directory, or the folder where you installed Maven, and save the path to that folder.
2. Open the configuration file for your shell in an editor of your choice. For recent macOS versions, the default shell is zsh and the default configuration file is located at `~/.zshrc`.

Add the following lines to the bottom of the configuration file. Set the value of `M2_HOME` to the path you saved in step 1:

```
export M2_HOME="your Maven installation directory"
export PATH="$M2_HOME/bin:$PATH"
```

These commands make the `mvn` command available in all terminals.

3. Close all OS terminal windows and quit all Visual Studio Code instances.
4. To verify that Maven was added to your PATH, open a new OS terminal and run the following command:

```
mvn -v
```

You should see an output with your Maven version.

5. After seeing your Maven output, restart Visual Studio Code. You might also need to restart your machine. Open a new Visual Studio Code terminal and run the following command:

```
mvn -v
```

The output should be identical to the output in step 4. If the Visual Studio Code output is different, try the following to make sure your setup is correct:

- Check your PATH variable in Visual Studio Code. An IDE extension might be altering the PATH such that it differs from your local PATH variable. Uninstall the extension to remove it from your PATH.
- Check your default shell in Visual Studio Code. If it's set to something other than zsh, repeat these steps for your shell.

## Windows

To add Maven to your Windows PATH, complete the following steps:

1. Locate your Maven installation directory, or the folder where you installed Maven, and save the path to that folder.
2. Open the Environment Variables window:
  - a. Choose the Windows button to open the search bar.
  - b. Enter Edit environment variables for your account and choose it.
3. In the **Environment Variables** window, look for the Path variable. If you have a Path variable already, choose **Edit...** to update it. If you don't see a Path variable, choose **New...** to add one.
4. In the **Edit environment variable** window that appears, double click the existing path to edit it, or choose **New** to add a new path entry.

Replace the existing Maven path entry with the path you saved in step 1, or add the path as a new entry. At the end of the path, add \bin as a suffix, as in the following example:

```
C:\Users\yourusername\Downloads\apache-maven-3.9.6-bin\apache-maven-3.9.6\bin
```

5. Choose **OK** to save the path entry, and then choose **OK** again in the **Environment Variables** window.
6. Open a new Command Prompt and run the following command:

```
mvn -v
```

You should see an output with your Maven version.

## Why can't Amazon Q build my code?

If the transformation fails when Amazon Q is building your code, your project may not be configured properly for the environment where Amazon Q builds your code. You might need to update your build configuration or code implementation.

Review the build log output Amazon Q provides to determine if there are changes you can make to your project. Following are some common issues that might prevent Amazon Q from building your code.

## Remove absolute paths in pom.xml

If you have an absolute path in your pom.xml file, Amazon Q won't be able to find the relevant files, and as a result might not be able to build your code.

Following is an example of an absolute path that you could have in your pom.xml file:

```
<toolspath>
    <path>/Library/Java/JavaVirtualMachines/jdk-11.0.11.jdk/Contents/Home/lib/
    tools.jar</path>
</toolspath>
```

Instead of using an absolute path, you can create a relative path using a pointer. Following is an example of how you can replace the previous absolute path with a relative path:

```
<toolspath>
    <path>${java.home}/../lib/tools.jar</path>
</toolspath>
```

## Remove local or external databases in unit tests

Amazon Q runs any unit tests in your project when it builds your code. If a unit test calls a local or external database, Amazon Q won't have access to the database, causing the build to fail. To prevent the build from failing, you must either remove the database call from the unit test or remove the unit test before submitting the transformation.

## Why did my transformation fail after 55 minutes?

If your code transformation job fails after 55 minutes, your code build time likely exceeds the build time limit. There is currently a time limit of 55 minutes for building your code.

If your local build time takes 55 minutes or longer, reduce your project's build time to transform your code. If your local build is faster than the build with Code Transformation, check your project for tasks that might be failing or take a longer time in a different environment. Consider disabling long-running test cases. Also consider using timeouts for attempts to access resources that might not be available from the secure IDE environment or the internet.

## Why can't I download my transformed code?

If you aren't able to download your code after your transformation is complete, it's likely due to one of the following issues. See the topic that corresponds to the error you see from Amazon Q.

## Topics

- [Reduce project size](#)
- [Download code diff within 24 hours](#)
- [Configure proxy settings in your IDE](#)
- [Remove wildcard characters in JetBrains proxy settings](#)

### Reduce project size

After the transformation is complete, Amazon Q generates an output artifact that contains a diff with your upgraded code and a transformation summary with information about the changes it made. The output artifact must be 1 GB or less in order for the IDE to download it.

If the output artifact exceeds the limit, you will not be able to download your upgraded code or transformation summary. Try transforming a smaller project to prevent a large output artifact. If the issue persists, contact AWS Support. For information about contacting AWS Support with Amazon Q, see [Using Amazon Q Developer to chat with AWS Support](#).

### Download code diff within 24 hours

The code diff file with your upgraded code is only available for 24 hours after the transformation is complete. If it's been over 24 hours since the transformation completed, restart the transformation to download the diff file.

### Configure proxy settings in your IDE

The Amazon Q Developer Agent for code transformation downloads your upgraded code from a service-owned Amazon S3 bucket. Part of the download process involves using SSL or TLS certificates to establish communication between Amazon S3 and your IDE. If you are using a proxy server, the SSL or TLS certificates used by your proxy server must be trusted, otherwise Amazon Q is not able to upload your project.

To download your code, you might need to configure your IDE to trust certificates or update other proxy settings. For more information on updating your proxy settings, see [Configure proxy settings in your IDE](#).

### Remove wildcard characters in JetBrains proxy settings

If you have configured proxy settings in your JetBrains IDE, you might see the following error when downloading your upgraded code:

```
software.amazon.awssdk.core.exception.SdkClientException:  
Unable to execute HTTP request: Dangling meta character '*' near index 0
```

This is likely caused by the presence of a wildcard character (\*) in the **No proxy for** field of your IDE's proxy settings. The Java SDK used by the Amazon Q Developer Agent for code transformation doesn't support wildcard entries in this field.

To download your code, remove any wildcards from the **No proxy for** field, and then restart your IDE. If you need to specify hosts that should bypass the proxy, use a regular expression instead of a wildcard. To update proxy settings in your JetBrains IDE, see [HTTP Proxy](#) in the JetBrains documentation.

## How do I access code transformation logs?

### Access logs in JetBrains

For information about how to access JetBrains log files, see [Locating IDE log files](#) in the JetBrains documentation.

To find logs emitted by the Amazon Q Developer Agent for code transformation in JetBrains, search the IDE logs for the following string:

```
software.aws.toolkits.jetbrains.services.codemodernizer
```

Code transformation logs start with the preceding string. Logs generated by Maven are displayed on the **Run** tab and have the preceding string before and after the log entry.

### Access logs in Visual Studio Code

To find logs emitted by the Amazon Q Developer Agent for code transformation in VS Code, complete the following steps:

1. Choose **View** in the top navigation bar, and then choose **Command Palette**.
2. Search Amazon Q: **View Logs** in the command palette that appears.
3. The logs open in the IDE. To search the log files for CodeTransformation, use **CMD + F** or **Control + F**.

Code transformation logs in VS Code are prefixed with **CodeTransformation:**. Following is an example of a log generated in VS Code for a Maven copy dependencies error:

2024-02-12 11:29:16 [ERROR]: CodeTransformation: Error in running Maven copy-dependencies command mvn = /bin/sh: mvn: command not found

## How do I find my transformation job ID?

### Find your job ID in JetBrains

To find a transformation job ID in JetBrains, go to the **Transformation details** tab in the **Transformation Hub** and choose the **Show Job Status** (clock) icon.

### Find your job ID in Visual Studio Code

To find a transformation job ID in VS Code, go to the **Transformation Hub** and choose the **Show Job Status** (clock) icon.

## Developing software with the Amazon Q Developer Agent for software development

The Amazon Q Developer Agent for software development can help you develop code features or make code changes to projects in your integrated development environment (IDE). You explain the task you want to accomplish, and Amazon Q uses the context of your current project or workspace to generate code to implement the changes. Amazon Q can help you build AWS projects or your own applications.

You can start an entirely new project, or work on an open project in your IDE. When you develop in an existing project, Amazon Q uses all relevant files in your workspace root as context to generate code. Amazon Q filters out files or folders defined in a `.gitignore` file, and only uses supported file types to generate code. For a list of supported file types, see the following GitHub links for your IDE:

- Supported file types for JetBrains IDEs – <https://github.com/aws/aws-toolkit-jetbrains/blob/main/plugins/core/jetbrains-community/src/software/aws/toolkits/jetbrains/services/telemetry/TelemetryUtils.kt>
- Supported file types for Visual Studio Code – <https://github.com/aws/aws-toolkit-vscode/blob/master/packages/core/src/shared/filetypes.ts>

To get started, open up a new or existing project and enter **/dev** in the Amazon Q chat panel. A new chat tab opens where you interact with the Agent for software development to generate new code for your feature.

## Topics

- [Develop features with /dev](#)
- [Best practices](#)
- [Example tasks](#)
- [Quotas](#)
- [Troubleshooting issues with the Amazon Q Developer Agent for software development](#)

## Develop features with /dev

To work on a task with the Agent for software development in your IDE, complete the following steps.

1. In your IDE, open a new or existing project or workspace where you want to develop features.
2. Choose the Amazon Q icon to open the Amazon Q chat panel.
3. Enter **/dev** in the Amazon Q chat panel followed by a description of the task you want to accomplish or the issue you want to resolve. You can provide a brief overview of a task, or add more details. Amazon Q uses your description and the code in your project to generate code.

Following is an example of a code change you can ask Amazon Q to implement:

```
/dev Create a new REST API endpoint /api/authenticate to handle user authentication. This endpoint should accept POST requests with user credentials and return a JWT token upon successful authentication. Additionally, update the user management system to integrate with the new authentication endpoint and enforce authentication for relevant API endpoints.
```

You can also enter **/dev** only to see example tasks.

4. A new tab opens. If you haven't already, enter a description of your task or issue.
5. Amazon Q begins to generate code for the task you've described. During the code generation, Amazon Q provides a summary of the steps it's taking and the files it's using or updating to suggest new code. This step can take a few minutes.

If you want Amazon Q to stop generating code, choose **Stop**. The incomplete code generation will count towards the code generation quota for this task. After you stop a generation, you have the option to enter another task description to keep working on your task.

6. After Amazon Q has generated code for your task, it provides a list of files with suggested changes. Choose a file to view a file diff with the changes.
7. To update your files with all of the suggested code changes, choose **Accept all changes**. You can also accept and reject changes to individual files. To accept an individual file change, hover your cursor over a file name and choose the green check mark. To reject a change, choose the red X. You can also choose **Accept remaining changes** to apply any changes you haven't accepted yet.

Once you accept a code change, you can't undo it. If you reject a code change, you can revert it by hovering over the file name and choosing the arrow. You then have the option to accept or reject the code change again.

8. If you aren't satisfied with the proposed changes, you can provide feedback on what can be improved. Choose **Provide feedback & regenerate**, and then describe the changes that you'd like Amazon Q to make.

Amazon Q generates new code based on your feedback. When the generated code meets your requirements, choose **Accept all changes** or accept individual files to update the code in your project or workspace.

9. After you've updated your code, Amazon Q prompts you to start working on a new task.

To start working on another task in your project or workspace, choose **Yes, I have another task**. Your previous conversation and generated code aren't used as context for the new task.

To end the session with the Amazon Q Developer Agent for software development, choose **No, thanks**. To keep chatting or initiate another workflow with Amazon Q, open a new chat tab.

## Best practices

To make the most out of the Amazon Q Developer Agent for software development, follow these best practices:

- Provide a detailed description of the new feature or code changes you want to make, including the specifics of what the code should achieve. This allows Amazon Q to propose comprehensive and implementable code changes. For examples, see [Example tasks](#).
- Your feature shouldn't require updates to more than 5 files at a time. Asking Amazon Q to make larger changes might impact the quality and manageability of the implementation of your feature. If your file diff includes changes to many files, try reducing the scope of your feature description.

## Example tasks

Following are example feature development tasks you can ask the Amazon Q Developer Agent for software development to perform, with the corresponding description you might provide.

- **Update CSS Styles for Responsive Layout:** Enhance the responsiveness of the application's layout by updating CSS styles. Focus on adjusting the layout for different screen sizes, ensuring optimal display across various devices.
- **Fix User Profile Image Upload:** Resolve the issue preventing users from uploading profile images. Investigate the file upload process, ensure proper file type validation, and address any server-side errors preventing successful uploads.
- **Refactor Code for Code Readability:** Improve the readability of a specific code module by refactoring it. Break down complex functions into smaller, more manageable units, and apply meaningful variable and function names for better clarity.
- **Implement Input Validation for Contact Form:** Add client-side and server-side input validation to the contact form to prevent submission of empty or invalid data. Display appropriate error messages to users for any validation failures.
- **Resolve Broken Links in Navigation Menu:** Investigate and fix broken links in the navigation menu of the application. Update the URLs or routes to ensure all navigation links lead to the correct pages.
- **Optimize Image Loading for Faster Page Loads:** Optimize image loading on key pages of the application to reduce page load times. Implement lazy loading or asynchronous loading techniques to prioritize the display of visible content.
- **Add Error Logging for Critical API Endpoints:** Enhance error handling by implementing logging for critical API endpoints. Log relevant error details such as request parameters and stack traces to facilitate debugging and troubleshooting.

- **Update Documentation for API Endpoints:** Review and update the documentation for existing API endpoints to reflect recent changes or additions. Ensure accuracy and completeness of information for developers consuming the API.
- **Refactor Database Queries for Efficiency:** Analyze and optimize database queries to improve efficiency and reduce resource consumption. Identify and eliminate redundant queries, optimize indexing, and consider caching strategies where applicable.

## Quotas

The Amazon Q Developer Agent for software development maintains the following quotas:

- **Code generations per task** – The number of times the Agent for software development can generate code for a given development task, including the initial code generation.
- **Code project size** – The maximum size of the code file or folder that the Agent for software development can use as context to generate new code.

### Amazon Q Developer Agent for software development Quotas

Resource	Quota
Code generations per task	3
Code project size	200 MB uncompressed
	50 MB compressed

## Troubleshooting issues with the Amazon Q Developer Agent for software development

The following information can help you troubleshoot common issues with the Amazon Q Developer Agent for software development.

### How do I find my conversation ID?

You might need your conversation ID to get support for issues with the Amazon Q Developer Agent for software development. For some error messages, the conversation ID appears in the message from Amazon Q. If you don't see it, use the following steps to find the conversation ID in your IDE.

## Visual Studio Code

1. Open the Command Palette, and run the following command:

Amazon Q: View Logs.

2. An output console opens in the VS Code terminal. Search for the following string:

Amazon Q Developer Agent for software development Conversation ID:

The conversation ID is listed after the colon. Confirm that the timestamp corresponds to the conversation that you need the ID for.

## JetBrains

1. At the top of the IDE, choose **Help**, and then **Show Log in Finder**.

Depending on your JetBrains IDE and operating system, the logs menu text might look different. For more information, see [Locating IDE log files](#) in the IntelliJ IDEA documentation.

2. Your file finder opens. Choose **idea.log** to open the log file.
3. Search for the following string:

Amazon Q Developer Agent for software development Conversation ID:

The conversation ID is listed after the colon. Confirm that the timestamp corresponds to the conversation that you need the ID for.

## Scanning your code with Amazon Q

Amazon Q can scan your codebase for security vulnerabilities and code quality issues to improve the posture of your applications throughout the development cycle. You can initiate a scan of an entire codebase, analyzing all files in your local project or workspace, or enable auto scans that assess your code as you write it.

When Amazon Q discovers a potential security vulnerability or quality issue in your code, it generates a finding with a description of the issue and a recommended fix. Some findings include an automatic fix, which updates your code files in-place.

Scans are powered by [security detectors](#) that are informed by years of AWS and Amazon.com security best practices. As security policies are updated and detectors are added, scans automatically incorporate new detectors to ensure your code is compliant with the most up-to-date policies.

## Topics

- [Types of scans](#)
- [Quotas](#)
- [Starting a scan](#)
- [Addressing findings](#)

## Types of scans

Amazon Q performs code security and code quality analysis in every scan. The following sections explain the types of security and quality issues Amazon Q detects.

### Code security scanning

Amazon Q detects security policy violations and vulnerabilities in your code with static application security testing (SAST), secrets detection, and infrastructure as code (IaC) scanning. For a complete list of the detectors Amazon Q uses to scan your code, see the [Detector Library](#).

- **SAST scanning — Detect security vulnerabilities in your source code.** Amazon Q identifies various security issues, such as resource leaks, SQL injection, and cross-site scripting.
- **Secrets scanning — Prevent the exposure of sensitive or confidential information in your codebase.** Amazon Q scans your code and text files for secrets such as hardcoded passwords, database connection strings, and usernames. Secrets findings include information about the unprotected secret and how to protect it.
- **IaC scanning — Evaluate the security posture of your infrastructure files.** Amazon Q can scan your infrastructure as code (IaC) code files to detect misconfiguration, compliance, and security issues.

### Code quality scanning

Amazon Q detects quality and maintainability issues in your code to ensure your codebase is meeting quality and efficiency best practices. Amazon Q generates findings related to various

quality issues, including but not limited to performance, machine learning rules, and AWS best practices.

## Quotas

Amazon Q security scans maintain the following quotas:

- **Input artifact size** – The size of all the files within an IDE project workspace, including third-party libraries, build JAR files, and temporary files.
- **Source code size** – The size of the source code that Amazon Q scans after filtering all third-party libraries and unsupported files.

The following table describes the quotas maintained for auto scans and full project scans.

Resource	Auto scans	Project scans
Maximum input artifact size	200 KB	500 MB
Maximum source code size	200 KB	50 MB

## Starting a scan

Amazon Q can scan your entire codebase, or auto-scan your code as you write it.

### Topics

- [Scan as you code](#)
- [Scan your project](#)

## Scan as you code

### Note

The following steps assume that you have already [installed the Amazon Q plugin in your IDE](#), and that you have [subscribed to Amazon Q Developer Pro](#).

Amazon Q auto-scans are only available with a Amazon Q Developer Pro subscription in JetBrains and Visual Studio Code.

Amazon Q automatically scans the file you are actively coding in, generating findings as soon as they are detected in your code. Auto-scans are enabled by default when you use Amazon Q.

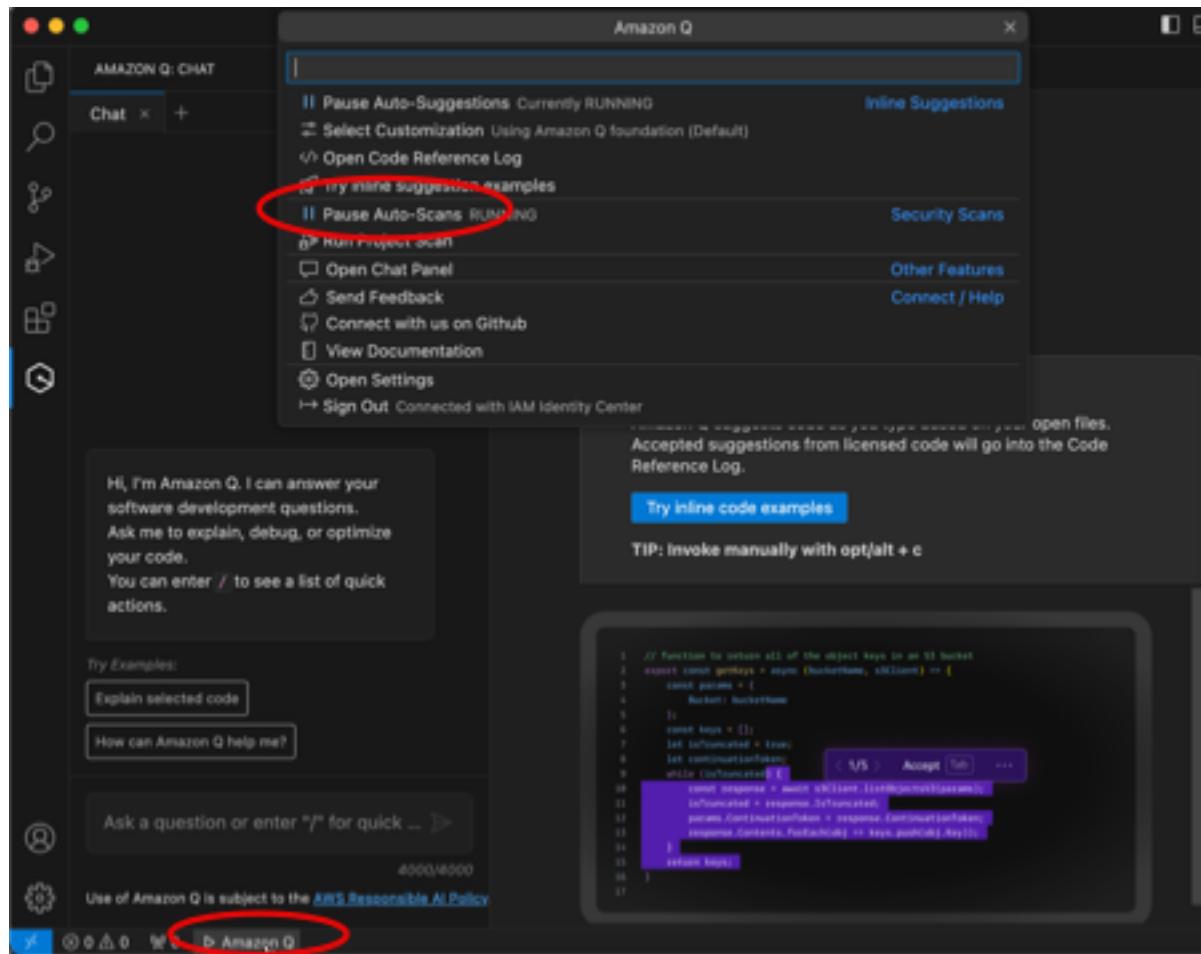
## Pausing auto-scans

To pause auto scans, complete the following steps.

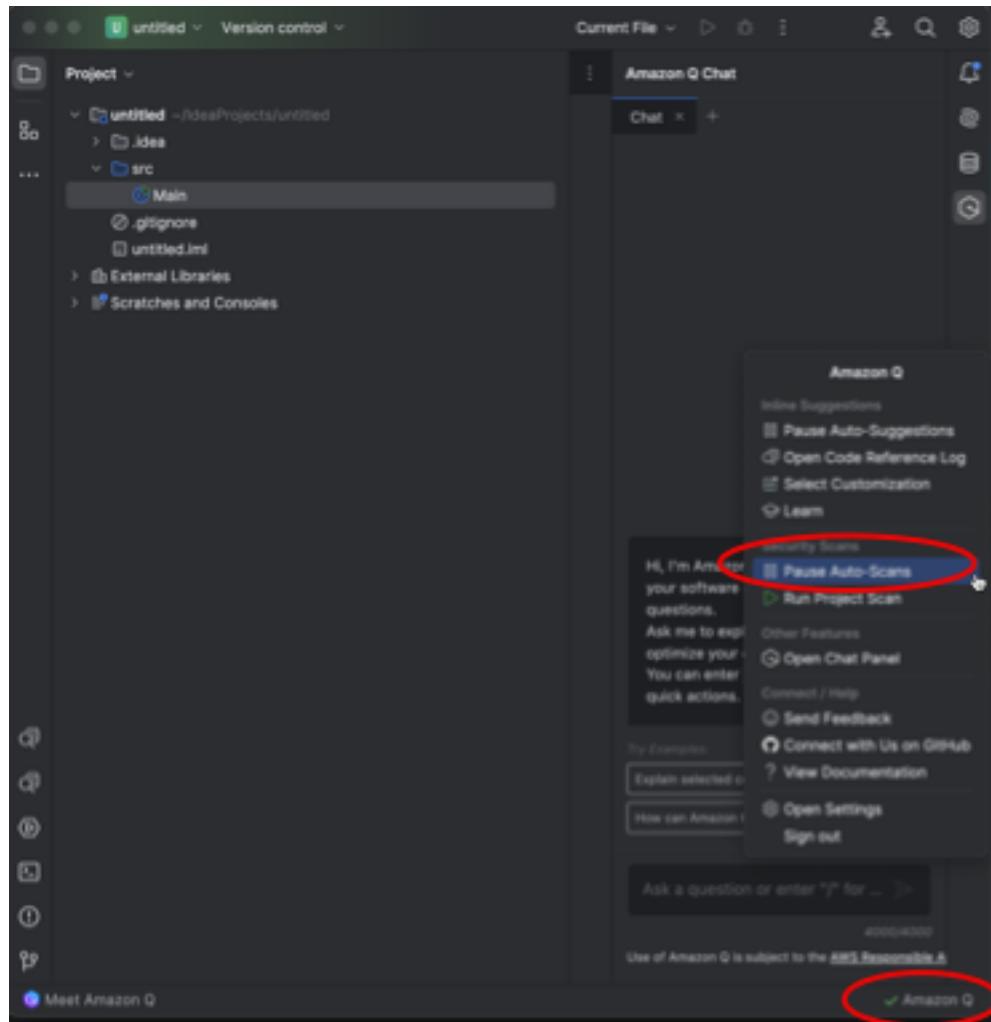
1. Choose the text, **Amazon Q**, from the rectangular area at the bottom of the IDE window. In VS Code, this area is called the *component tray*. In JetBrains IDEs, this area is called the *status bar*.

After you choose the Amazon Q text the Amazon Q task bar opens at the top of the IDE window.

In the following image, in Visual Studio Code, the user chooses the text **Amazon Q** at the bottom of the interface, prompting a drop-down window from which the user may choose **Pause Auto-Scans**.



In the following image, in IntelliJ (a JetBrains IDE), the user chooses the text **Amazon Q** at the bottom of the interface, prompting a drop-down window from which the user may choose **Pause Auto-Scans**.



## 2. Choose **Pause Auto-Scans**.

### Scan your project

You can scan a project to evaluate the security posture of your entire codebase at once.

To manually run a scan, complete the following steps for your IDE.

JetBrains

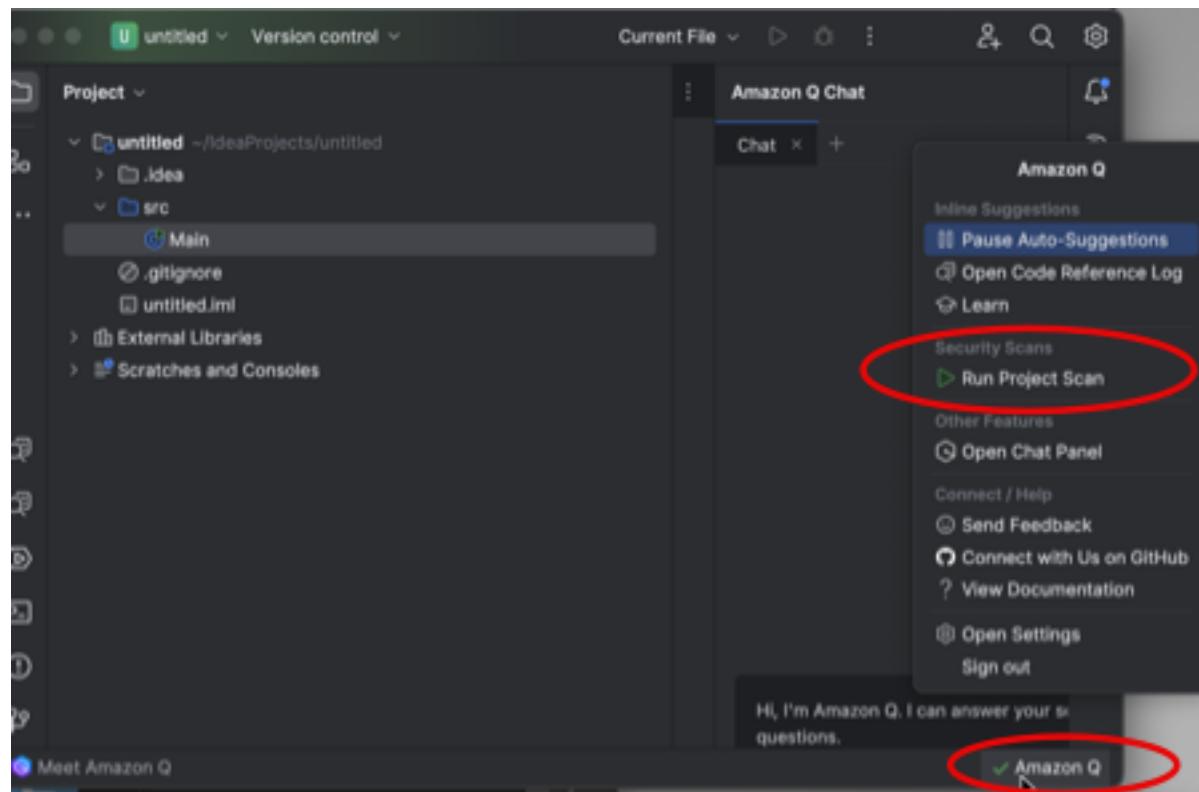
To scan a project in a JetBrains IDE:

1. Open up a file from the project you want to scan in your JetBrains IDE.
2. Choose **Amazon Q** from the status bar at the bottom of the IDE window.

The Amazon Q task bar opens on the right of the IDE window.

3. Choose **Run Project Scan** from the task bar. Amazon Q begins scanning your project.

In the following image, in IntelliJ (a JetBrains IDE), the user chooses **Amazon Q** from the status bar, prompting a task bar from which the user may choose **Run Project Scan**.



4. When the scan is complete, the findings Amazon Q generated appear in the **Problems** tab. You can choose individual findings to be redirected to the specific area of the file where the vulnerability or low-quality code was detected.
5. Hover your cursor over the underlined code to view details about the finding. For information about addressing findings, see [Addressing findings](#).

## Visual Studio Code

To scan a project in a VS Code:

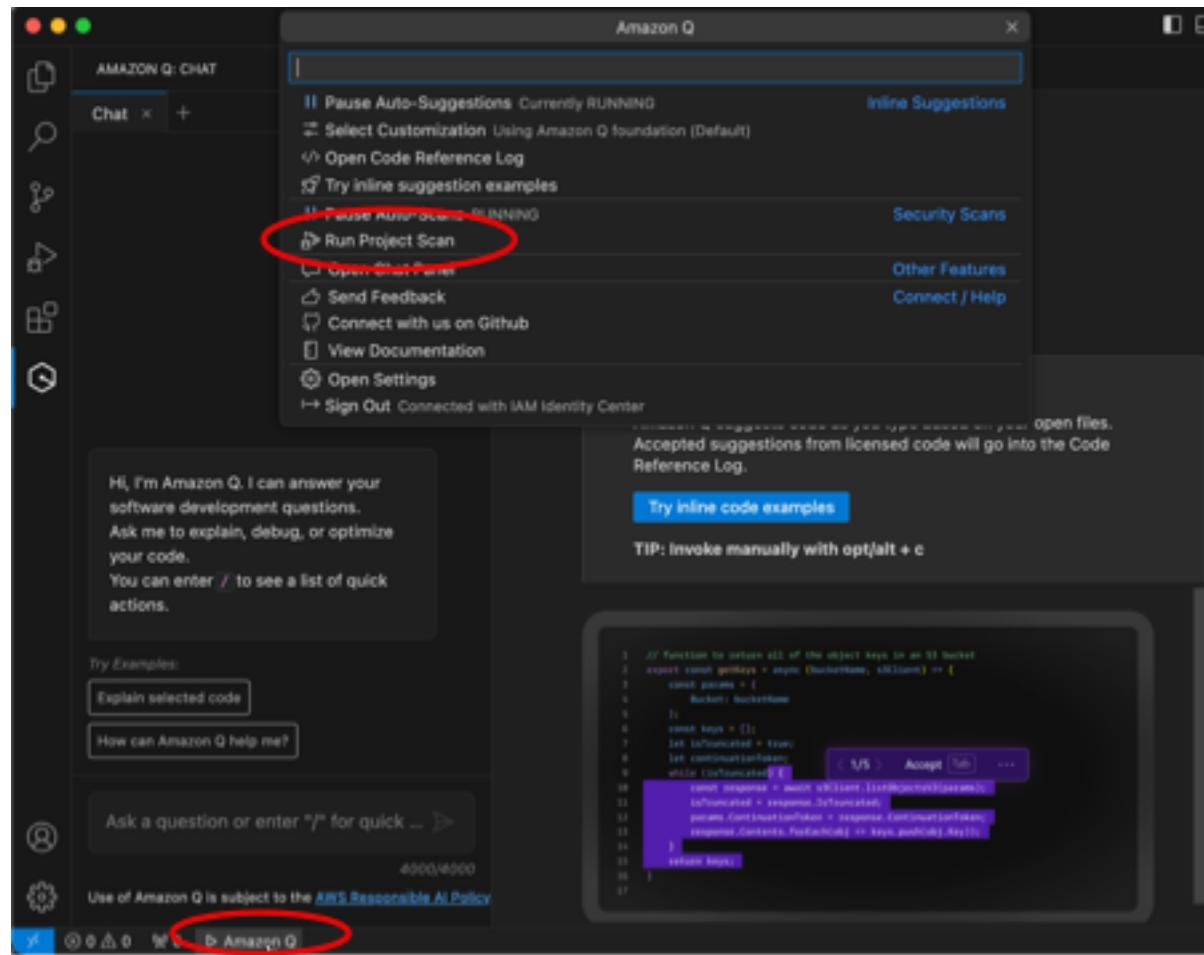
1. Open up a file from the project you want to scan in VS Code.

2. Choose **Amazon Q** from the component tray at the bottom of the IDE window.

The Amazon Q task bar opens at the top of the IDE window.

3. Choose **Run Project Scan**. Amazon Q begins scanning your project.

In the following image, in VS Code, the user chooses **Amazon Q** from the component tray, prompting a task bar from which the user may choose **Run Project Scan**.



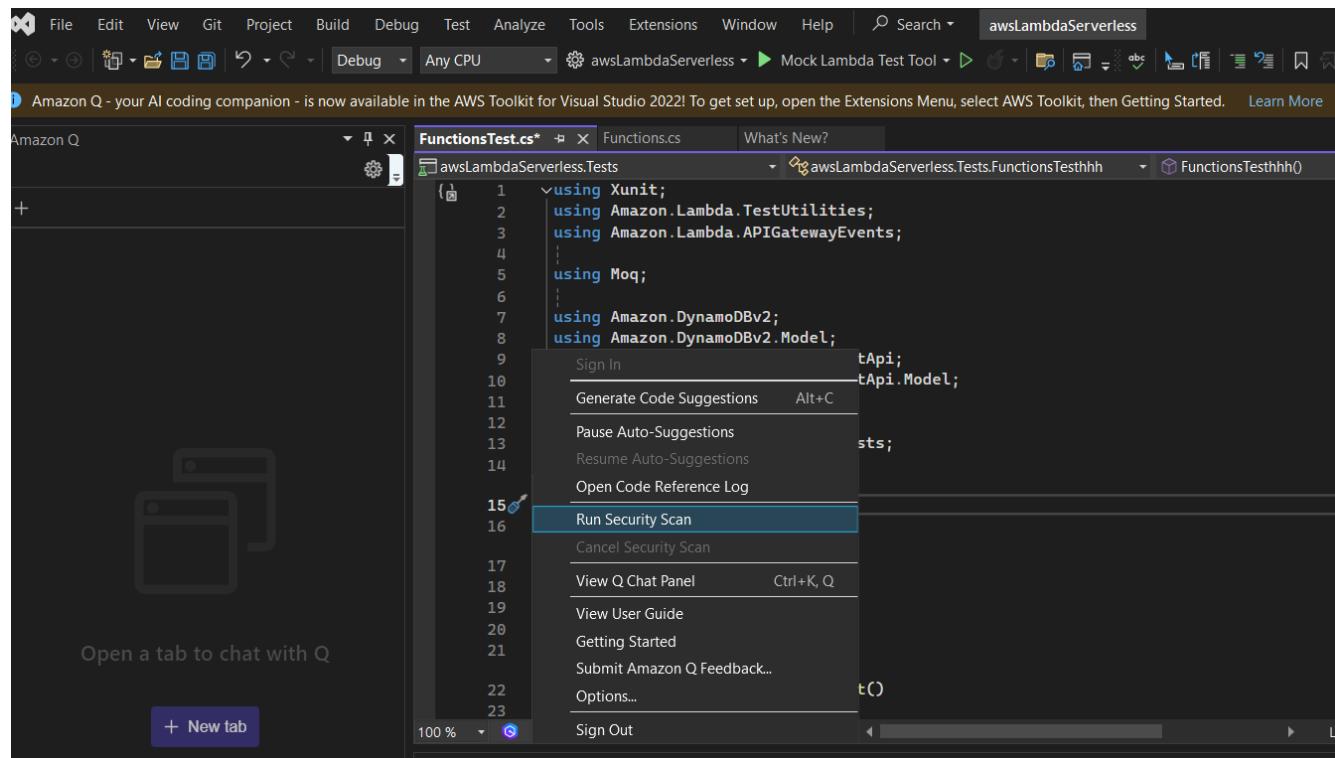
4. When the scan is complete, the findings Amazon Q generated appear in the **Problems** tab. You can choose individual findings to be redirected to the specific area of the file where the vulnerability or low-quality code was detected.
5. Hover your cursor over the underlined code to view details about the finding. For information about addressing findings, see [Addressing findings](#).

## Visual Studio

To scan a project in a Visual Studio:

1. Open up a file from the project you want to scan in Visual Studio.
2. Choose the Amazon Q icon at the bottom of your file to open the Amazon Q task bar.
3. From the task bar, choose **Run Security Scan**. Amazon Q begins scanning your project.

In the following image, in Visual Studio, the user chooses the **Amazon Q** icon, prompting a task bar from which the user may choose **Run Security Scan**.



4. The status of your scan is updated in the Visual Studio output pane. You're notified when the scan is complete.

For information about viewing and addressing findings, see [Addressing findings](#).

## Addressing findings

When Amazon Q generates a finding, it underlines the vulnerable or low-quality code. Findings include details about the security or quality issue and how to address it in your code.

### Topics

- [Address findings in JetBrains and Visual Studio Code](#)
- [Address findings in Visual Studio](#)

## Address findings in JetBrains and Visual Studio Code

You can view a list of detected findings in the **Problems** tab. Choose a finding to be redirected to the specific area of the file where the vulnerability or low-quality code was detected. To see a pop-up with information about the finding, hover your cursor over the underlined code.

From the pop-up, you have a few options to address a finding:

- **Automatic fix** — This option appears for some vulnerabilities. Choose **Fix** to update your code in-place and address the finding.
- **Explain with Q** — Amazon Q explains the finding and recommends how to fix it. In the pop-up, choose **Explain** to send the finding and associated code to the Amazon Q chat panel.
- **Learn more about the finding** — The finding describes how you can approach remediating the vulnerability. Use this information to replace the code with improved code.

## Address findings in Visual Studio

To view findings in Visual Studio, open the Visual Studio **Error List** by expanding the **View** heading in the Visual Studio main menu and choosing **Error List**.

You can use the information in the finding to update your code. After updating your code, scan your code again to see if the findings were addressed.

By default, the Visual Studio **Error List** displays all of the warnings and errors for your code base. To filter your Amazon Q findings from the Visual Studio **Error List**, create a filter by completing the following procedure.

### Note

Scan findings are only visible after you've run a security scan in which Amazon Q detected issues.

Findings appear as warnings in Visual Studio. In order to view findings from your **Error List**, the **Warnings** option in the **Error List** heading must be selected.

## Filter findings in the Error List

1. From the Visual Studio main menu, choose **view** and then **Error List** to open the **Error List** pane.

2. From the **Error List** pane, right-click the header row to open the context menu.
3. From the context menu, expand **Show Columns**, and then select **Tool** in the expanded menu.
4. The **Tool** column is added to your **Error List**.
5. From the **Tool** column header, select the **Filter** icon and choose **Amazon Q** to filter for Amazon Q security scan findings.

## Supported languages for Amazon Q Developer in the IDE

Amazon Q Developer provides support for a wide range of programming languages across its various features and capabilities. This page outlines the languages and versions supported for inline code suggestions, code transformation, security scans, and automatic code fixes.

### Language support for inline suggestions

Amazon Q supports inline code suggestions for multiple programming languages. The accuracy and quality of the code generation for a programming language depends on the size and quality of the training data.

In terms of the quality of the training data, the programming languages with the most support are:

- C
- C++
- C#
- Dart
- Go
- Java
- JavaScript
- Kotlin
- Lua
- PHP
- PowerShell
- Python
- R
- Rust

- SQL
- Swift
- SystemVerilog
- TypeScript

The Infrastructure as Code (IaC) languages with the most support are:

- JSON
- YAML
- HCL (Terraform)
- CDK (TypeScript, Python)

Amazon Q also supports code generation for:

- Ruby
- Shell
- Scala

## Language support for Code Transformation

The Amazon Q Developer Agent for code transformation supports upgrading code from Java 8 or Java 11 to Java 17.

## Language support for security scans

Amazon Q security scans supports the following language versions:

- **Java** - Java 17 and earlier
- **JavaScript** - ECMAScript 2021 and earlier
- **Python** - Python 3.11 and earlier, within the Python 3 series
- **C#** - All versions (.NET 6.0 and later recommended)
- **TypeScript** - All versions
- **Ruby** - Ruby 2.7 and 3.2
- **Go** - Go 1.18

- **C** - C11 and earlier
- **C++** - C++17 and earlier
- **PHP** - PHP 8.2 and earlier
- **Kotlin** - Kotlin 2.0.0 and earlier
- **Scala** - Scala 3.2.2 and earlier
- **JSX** - React 17 and earlier
- **Infrastructure as Code (IaC) languages**
  - **AWS CloudFormation** - 2010-09-09
  - **Terraform** - 1.6.2 and earlier
  - **AWS CDK** - TypeScript and Python

Automatic code fixes are available for scans in the following languages and versions:

- **Java** - Java 17 and earlier
- **JavaScript** - ECMAScript 2021 and earlier
- **Python** - Python 3.11 and earlier, within the Python 3 series
- **C#** - All versions (.NET 6.0 and later recommended)
- **TypeScript** - All versions
- **Infrastructure as Code (IaC) languages**
  - **AWS CloudFormation** - 2010-09-09
  - **Terraform** - 1.6.2 and earlier
  - **AWS CDK** - TypeScript and Python

# Using Amazon Q Developer on the command line

This chapter explains how you can use Amazon Q Developer to enable IDE-style completions for hundreds of popular CLIs like git, npm, docker, and aws. Start typing, and Amazon Q populates contextually relevant subcommands, options and arguments.

## Topics

- [Supported command line environments](#)
- [Installing Amazon Q for command line](#)
- [Verifying your download](#)
- [Uninstalling Amazon Q for command line](#)
- [Chatting with Amazon Q in the command line](#)
- [Generating command line completions](#)
- [Amazon Q inline on the command line](#)
- [Translating from natural language to bash](#)
- [Debugging Amazon Q Developer for the command line](#)
- [Contributing completion specs to Amazon Q Developer](#)

## Supported command line environments

You can install the Amazon Q command line for macOS and specific Linux environments, including AppImage and Ubuntu. You can then optionally verify the installation for both macOS or Linux.

The following environments are supported for both macOS and Linux:

- Shells: bash, zsh, fish
- CLIs: Over 500 of the most popular CLIs such as git, aws, docker, npm, and yarn

## macOS

Amazon Q command line integrates with the following environments for macOS:

- Terminal emulators: iTerm2, macOS terminal, Hyper, Alacritty, Kitty, WezTerm
- IDEs: VS Code terminal, Jetbrains terminals (except Fleet)

## Linux

Amazon Q command line integrates with the following environments for Linux:

- Platform requirements: Amazon Q command line for Linux supports Ubuntu 22 and 24. It may otherwise work with GNOME v42+ or environments where the display server is Xorg and the input method framework is IBus.
- Terminal emulators: GnomeConsole, GnomeTerminal, Kitty, Hyper, WezTerm, Alacritty, Tilix, Terminator
- Architecture: amd64

## Installing Amazon Q for command line

You can install the Amazon Q command line for macOS by initiating a file download for the Amazon Q application. For more information, see [Supported command line environments](#).

## macOS

To install Amazon Q for command line for macOS, complete the following steps:

1. [Download Amazon Q for command line for macOS](#).
2. (Optional) Verify the downloaded file for Amazon Q command line for macOS. For more information, see [Verifying your download](#).
3. Authenticate with [Builder ID](#), or with [IAM Identity Center](#) using the start URL given to you by your account administrator.
4. Follow the instructions to install the shell integrations, and to grant macOS accessibility permissions.

The screenshot shows the Amazon Q developer interface. On the left is a sidebar with various links: Getting started, Help & support (highlighted with a red button), Features (CLI Completions), Chat, Translate, Inline, Settings, Keybindings, Integrations, and Preferences. The main area has a section titled "Automated checks" with two items: "Shell integrations" (green checkmark) and "Enable accessibility" (red X). A red oval highlights the "Enable accessibility" item. Below this is a section titled "Still having issues?" with two steps: "Run q doctor to automatically debug" and "Run q issue to create an auto-populated issue". It also says "You can also check out the following: User manual →".

## AppImage (Linux)

Amazon Q command line is available as an [AppImage](#), a single file with all dependencies included. It executes on a wide variety of Linux environments. You can set the file executable and you can begin using it.

To install Amazon Q for command line for Linux with AppImage, complete the following steps:

1. Download Amazon Q for command line for AppImage. Install the latest version using the following command:

```
curl --proto '=https' --tlsv1.2 -sSf https://desktop-release.q.us-east-1.amazonaws.com/latest/amazon-q.appimage -o amazon-q.appimage
```

For a specific version of the Amazon Q command line, you can replace ... /latest/ ... in the URL with the version number.

**Note**

Amazon Q command line follows semantic versioning.

2. (Optional) Verify the downloaded Amazon Q command line for AppImage. For more information, see [Verifying your download](#).
3. Make the downloaded AppImage executable using the following command:

```
chmod +x amazon-q.appimage
```

4. (Optional) Move the downloaded AppImage to a location on your PATH.
5. Execute the AppImage and follow the onboarding steps.

The AppImage bundles an XDG Desktop Entry file that can be installed and managed locally for you if you prefer. If your system supports the XDG Autostart specification, this will enable autostart capabilities, ensuring that the application is run on startup.

**Note**

Your system must have FUSE version 2 installed to run AppImages. For Ubuntu 22 and above, you can install FUSE 2 by using the following command:

```
sudo apt install libfuse2
```

**Note**

If the desktop application doesn't open, try running `amazon-q.appimage` directly in your terminal to view any output error logs. You can open a support issue in the [Amazon Q for command line discussions GitHub repository](#).

6. Authenticate with [Builder ID](#), or with [IAM Identity Center](#) using the start URL provided by your account administrator.
7. In a new terminal, use the following command:

```
q doctor
```

Follow any remediation instructions until you see the output: "✓ Everything looks good!"

## Ubuntu/Debian (Linux)

The Amazon Q command line is available as a prebuilt deb that you can install manually. The deb is compatible with a variety of Debian style systems. If you encounter issues with Ubuntu, consider AppImage, which should work on most Linux environments.

To install Amazon Q for command line for Linux with Ubuntu/Debian, complete the following steps:

1. Download the Amazon Q for command line deb package using the following command:

```
curl --proto '=https' --tlsv1.2 -sSF https://desktop-release.q.us-east-1.amazonaws.com/latest/amazon-q.deb -o amazon-q.deb
```

For a specific version of the Amazon Q command line, you can replace ... /latest/ ... in the URL with the version number.

 **Note**

Amazon Q command line follows semantic versioning.

2. (Optional) Verify the downloaded Amazon Q command line for Ubuntu/Debian. For more information, see [Verifying your download](#).
3. Install the Amazon Q command line using the following apt command:

```
sudo apt install -y ./amazon-q.deb
```

The apt command installs the binaries q, qterm, and q-desktop under /usr/local/bin.

4. Open the desktop application using the following command:

```
q
```

**Note**

If the desktop application doesn't open, run q-desktop directly in your terminal to view any output error logs. You can open a support issue by running q issue.

The deb packages also includes a desktop entry that is installed under /usr/share/applications/amazon-q.desktop. If your environment supports the XDG Desktop Entry specifications, you can open the desktop application by searching for Amazon Q.

5. Follow the onboarding steps.
6. Authenticate with [Builder ID](#), or with [IAM Identity Center](#) using the start URL provided by your account administrator.
7. In a new terminal, use the following command:

```
q doctor
```

Follow any remediation instructions until see the output: "✓ Everything looks good!"

## Verifying your download

Before installing Amazon Q for command line and using its features, you can verify the download for macOS and Linux.

### macOS

After you download Amazon Q for command line for macOS, you can verify its code signature using the following command:

```
codesign -v /Applications/Amazon\ Q.app
```

If there's no output, then the application's code signature is valid, and it has not been tampered with since it was signed.

For more verbose information about the app signature, use the following the command:

```
codesign -dv --verbose=4 /Applications/Amazon\ Q.app
```

To learn more about the macOS codesign utility, see the [Code Signing Guide](#) on the Apple developer website.

## AppImage (Linux)

After you download Amazon Q for command line for AppImage, you can verify the download by using the GnuPG tool. The AppImage is cryptographically signed using a PGP signature that can be verified by using the GnuPG tool. If there's damage or alteration of the files, verification will fail and you shouldn't proceed with installation.

To verify the downloaded deb, complete the following steps:

1. Import the Amazon Q command line PGP Public Key and verify the integrity of your downloaded zip file.
  - a. Download and install the gpg command using your package manager. For more information about GnuPG, see the [GnuPG documentation](#).
  - b. To create the public key file, create a text file and paste in the following text.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----  
  
mDMEZig60RYJKwYBBAHaRw8BAQdAy/+G05U5/E0A72W1cD4WkYn5SInri8pc4Z6D  
BKNNG0m0JEfTYXpvbiBRIENMSSBUZWftIDxxLWNsaUBhbWF6b24uY29tPoiZBBMW  
CgBBFIEEmvYEF+gnQskUPgPsUNx6jcJMVmcFAmYo0tECGwMFCQPCZwAFCwkIBwIC  
IgIGFQoJCAAsCBYCAwEChgcCF4AACgkQUNx6jcJMVmef5QD/QWWEGG/c0nbDnp68  
SJXuFkwiNwlH2rPw9ZRIQMnfAS0A/0V6ZsGB4k0ylBfc7CNfzRFGtovdBBgHqA6P  
zQ/PNscGuDgEZig60RIKKwYBBAGXVQEFAQEHQC4qle0NMBCq3+wJwbZSr0vbuRba  
D1xr4wUPh4Avn4AnAwEIB4h+BBgWCgAmFIEEmvYEF+gnQskUPgPsUNx6jcJMVmcF  
AmYo0tECGwwFCQPCZwAACgkQUNx6jcJMVmchMgEA613RveCM0YHAGQaSFMkguoAo  
vK6Fg0kDawgP0NPIP2oA/jIA04gsAntuQgM0sPunEdDeji2t+AhV02+DQIsXZpoB  
=f8yY  
-----END PGP PUBLIC KEY BLOCK-----
```

- c. Import the Amazon Q command line public key with the following command, substituting public-key-file-name with the file name of the public key you created.

```
gpg --import public-key-file-name  
gpg: directory '/home/username/.gnupg' created  
gpg: keybox '/home/username/.gnupg/pubring.kbx' created  
gpg: /home/username/.gnupg/trustdb.gpg: trustdb created
```

```
gpg: key 50DC7A8DC24C5667: public key "Amazon Q command line Team <q-command  
line@amazon.com>" imported  
gpg: Total number processed: 1  
gpg:                      imported: 1
```

2. Download the Amazon Q command line signature file for the AppImage. It has the same path and name as the .appimage file it corresponds to but has the extension .sig. The following example shows how to save it to the current directory as a file named amazon-q.appimage.sig. For the latest version of the Amazon Q command line, use the following command:

```
curl --proto '=https' --tlsv1.2 -sSF "https://desktop-release.q.us-  
east-1.amazonaws.com/latest/amazon-q.appimage.sig" -o "amazon-q.appimage.sig"
```

For a specific version of the Amazon Q command line, you can replace ... /latest/ ... in the URL with the version number.

3. Verify the signature, passing both the downloaded .sig and .appimage file names as parameters. Use the following GnuPG command:

```
gpg --verify amazon-q.appimage.sig amazon-q.appimage
```

The output should look similar to the following:

```
gpg: Signature made Wed 24 Apr 2024 12:08:49 AM UTC  
gpg:                               using EDDSA key 9AF60417E82742C9143E03EC50DC7A8DC24C566  
gpg: Good signature from "Amazon Q command line Team <q-command  
line@amazon.com>" [unknown]  
gpg: WARNING: This key is not certified with a trusted signature!  
gpg:                               There is no indication that the signature belongs to the owner.  
Primary key fingerprint: 9AF6 0417 E827 42C9 143E 03EC 50DC 7A8D C24C 5667
```

## Ubuntu/Debian (Linux)

The deb file includes a PGP signature that can be verified by using the GnuPG tool. If there's damage or alteration of the files, verification will fail and you shouldn't proceed with installation.

To verify the downloaded deb, complete the following steps:

1. Import the Amazon Q command line PGP Public Key and verify the integrity of your downloaded zip file.
  - a. Download and install the gpg command using your package manager. For more information about GnuPG, see the <https://gnupg.org/documentation/index.html> GnuPG documentation.
  - b. To create the public key file, create a text file and paste in the following text.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

```
mDMEZig60RYJKwYBBAHaRw8BAQdAy/+G05U5/E0A72W1cD4WkYn5SInri8pc4Z6D  
BKNNGOm0JEFtYXpvbiBRIENSSBUZWftIDxxLWNsaUBhbWF6b24uY29tPoiZBBMW  
CgBBFiEEmvYEF+gnQskUPgPsUNx6jcJMVmcFAmYo0tECGwMFCQPCZwAFCwkIBwIC  
IgIGFQoJCAsCBBYCAwECHgcCF4AACgkQUNx6jcJMVmf5QD/QWWEGG/c0nbDnp68  
SJXuFkwiNw1H2rPw9ZRIQMnfAS0A/0V6ZsGB4k0y1Bfc7CNfzRFGtovdBBgHqA6P  
zQ/PNscGuDgEZig60RIKKwYBBAGXVQEFAQEHQC4q1e0NMBCq3+wJwbZSr0vbuRba  
D1xr4wUPn4Avn4AnAwEIB4h+BBgWCgAmFEEmvYEF+gnQskUPgPsUNx6jcJMVmcF  
AmYo0tECGwwFCQPCZwAACgkQUNx6jcJMVmchMgEA613RveCM0YHAGQaSFMkguoAo  
vK6Fg0kDawgP0NPIP2oA/jIA04gsAntuQgM0sPunEdDeji2t+AhV02+DQIsXZpoB  
=f8yY
```

```
-----END PGP PUBLIC KEY BLOCK-----
```

- c. Import the Amazon Q command line public key with the following command, substituting public-key-file-name with the file name of the public key you created.

```
gpg --import public-key-file-name  
gpg: directory '/home/username/.gnupg' created  
gpg: keybox '/home/username/.gnupg/pubring.kbx' created  
gpg: /home/username/.gnupg/trustdb.gpg: trustdb created  
gpg: key 50DC7A8DC24C5667: public key "Amazon Q command line Team <q-command  
line@amazon.com>" imported  
gpg: Total number processed: 1  
gpg:                      imported: 1
```

2. Verify the downloaded file by using the GnuPG command:

```
gpg --verify amazon-q.deb
```

The output should look similar to the following:

```
gpg: Signature made Wed 24 Apr 2024 12:08:49 AM UTC
```

```
gpg:           using EDDSA key 9AF60417E82742C9143E03EC50DC7A8DC24C566
gpg: Good signature from "Amazon Q command line Team <q-command
line@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:           There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9AF6 0417 E827 42C9 143E 03EC 50DC 7A8D C24C 5667
```

## Uninstalling Amazon Q for command line

You can uninstall Amazon Q for command line that was previously installed in your macOS or Linux environment. Uninstalling Amazon Q for command line may remove resources related to Amazon Q for command line.

To uninstall Amazon Q for command line, complete the following steps:

1. Open a terminal window.
2. Uninstall Amazon Q for command line using the following command, depending on the environment:
  - **macOS or AppImage**

```
q uninstall
```

- **Ubuntu/Debian**

```
sudo apt purge amazon-q
```

## Chatting with Amazon Q in the command line

The Amazon Q command line is a command-line interface that allows you to interact with the Amazon Q. With the Amazon Q command line, you can engage in natural language conversations, ask questions, and receive responses from Amazon Q, all within your terminal environment.

## Context integration

One of the powerful features of the Amazon Q command line is its ability to integrate contextual information from your local development environment. By using context modifiers, you can provide the Amazon Q with relevant context, such as your git repository status, local shell

environment variables, and shell command history. This context integration enhances the AI model's understanding of your specific use case, enabling it to provide more relevant and context-aware responses.

## Context modifiers

The Amazon Q command line supports the following context modifiers:

- **@git:** This modifier allows you to pass information about your git repository status, including the current branch, staged and unstaged changes, and commit history.
- **@env:** By using this modifier, you can provide the Amazon Q with your local shell environment variables, which can be useful for understanding your development setup and configuration.
- **@history:** The @history modifier allows you to share your recent shell command history with Amazon Q, giving it insights into the actions you've taken and the context in which you're working.

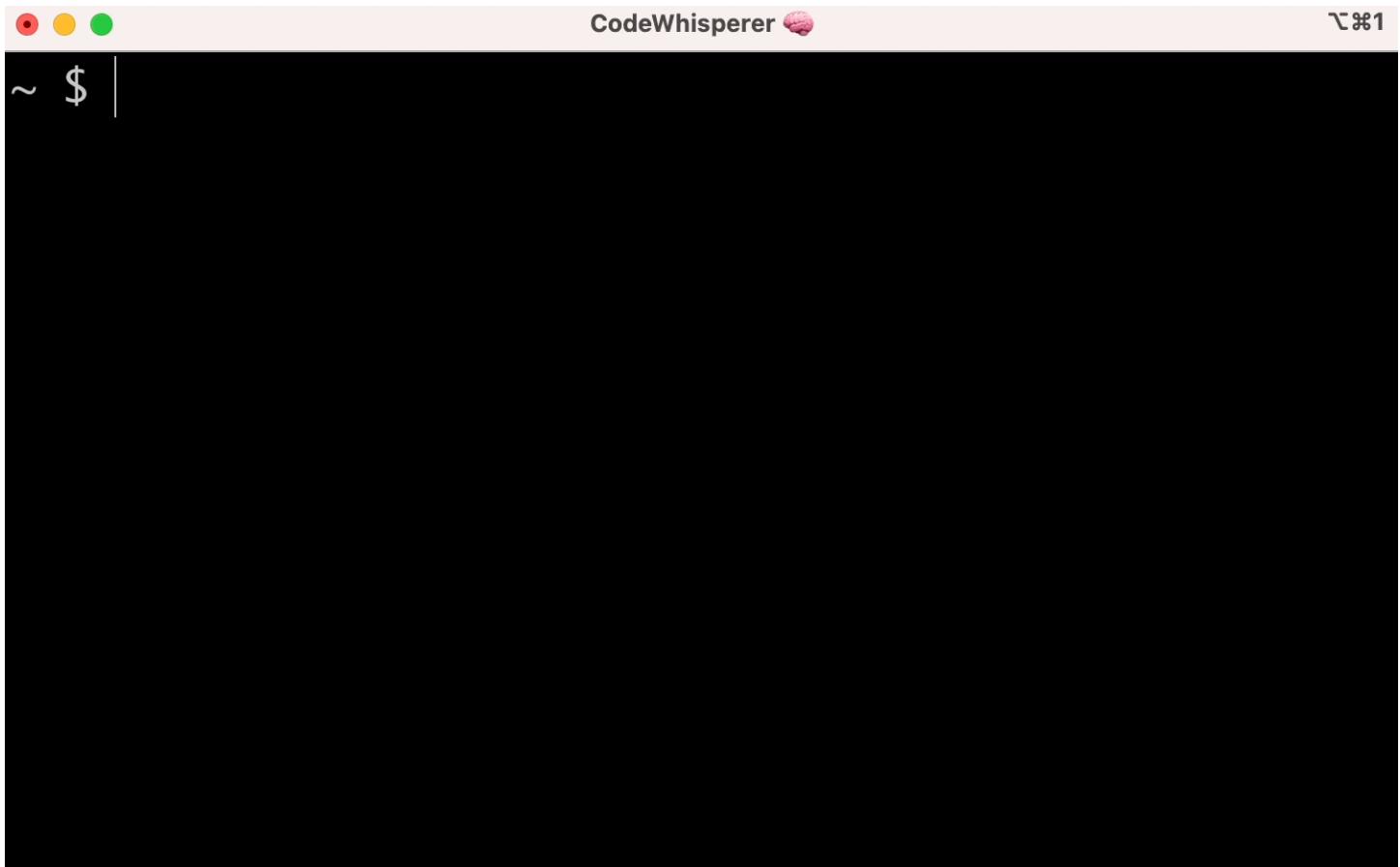
## Usage

To start using the Amazon Q command line, follow these steps:

1. [Install the Amazon Q command line.](#)
2. Open your terminal or command prompt.
3. Run the q chat command to initiate a conversation with Q.
4. Optionally, include any context modifiers (for example, @git, @env, @history) in your input to provide additional context to the model.
5. Type your query or input, and press **Enter**.
6. The Amazon Q will process your input, considering any provided context, and respond with its output.
7. Continue the conversation by providing additional input or asking follow-up questions.

## Generating command line completions

You can use Amazon Q Developer to add IDE-style completions to hundreds of popular CLIs like git, npm, docker, and aws. Start typing, and Amazon Q will pop up contextually relevant subcommands, options and arguments.



## Popular settings

The default settings provided by Amazon Q for command line may not "feel right," and may disrupt your existing workflow. You can customize your settings at any time by running `q` to open the settings dashboard. Here are a few popular settings:

- Keybindings – Changing the tab keybinding to "Insert common prefix or navigate" may make command line Completions feel more like traditional shell completions while "Insert common prefix or insert" will feel more like an IDE.
- Theme – You know what this is. Choose your favorite.
- Instant execute after space – A lot developers habitually type a space character just before they execute it. Enable this setting to avoid Amazon Q blocking you.
- First token completion – Enable this setting to get completions for CLIs themselves, not just the subcommands, options, and arguments.

# Using command line Autocomplete on a remote machine with SSH

When you install Amazon Q for the command line locally, it adds autocomplete for over 500 command line tools to your existing terminal in your local environment. By enabling SSH integration, you can make command line autocomplete accessible on remote machines as well.

## Local macOS Integration

To enable SSH integration enable the local SSH integrations by running the following command:

```
q integrations install ssh
```

## Remote Linux Integration

### Install and update requirements

- You must be able to extract or "unzip" the downloaded package. If your operating system doesn't have the built-in unzip command, use an equivalent.
- The Amazon Q command line uses glibc 2.34 or newer. It is included by default in most major distributions of Linux released since 2021.
- We support the Amazon Q command line on 64-bit versions of recent distributions of Fedora, Ubuntu, and Amazon Linux 2023.
- Because AWS doesn't maintain third-party repositories, we can't guarantee that they contain the latest version of the Q command line.

### Install the Amazon Q command line

1. Download the installation file in one of the following ways:

#### Linux x86-64

```
curl --proto '=https' --tlsv1.2 -sSf "https://desktop-release.codewhisperer.us-east-1.amazonaws.com/latest/q-x86_64-linux.zip" -o "q.zip"
```

#### Linux ARM (aarch64)

```
curl --proto '=https' --tlsv1.2 -sSf "https://desktop-release.codewhisperer.us-east-1.amazonaws.com/latest/q-aarch64-linux.zip" -o "q.zip"
```

## 2. (Optional) Verifying the integrity of your downloaded zip file

If you chose to manually download the Amazon Q command line installer package .zip in the above steps, you can use the following steps to verify the signatures by using the GnuPG tool.

The Amazon Q command line installer package .zip files are cryptographically signed using PGP signatures. If there is any damage or alteration of the files, this verification fails and you should not proceed with installation.

- a. Download and install the gpg command using your package manager. For more information about GnuPG, see the GnuPG website.
- b. To create the public key file, create a text file and paste in the following text.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

```
mDMEZig60RYJKwYBBAHaRw8BAQdAy/+G05U5/E0A72W1cD4WkYn5SInri8pc4Z6D  
BKNNG0m0JEftYXpvbiBRIENSSBUZWftIDxxLWNsaUBhbWF6b24uY29tPoiZBBMW  
CgBBFfEEmvYEF+gnQskUPgPsUNx6jcJMVmcFAmYo0tECGwMFCQPCZwAFCwkIBwIC  
IgIGFQoJCAsCBYCAwECHgcCF4AACgkQUNx6jcJMVmf5QD/QWWEGG/c0nbDnp68  
SJXuFkwiNwlH2rPw9ZRIQMnfAS0A/0V6ZsGB4k0y1Bfc7CNfzRFGtovdBBgHqA6P  
zQ/PNscGuDgEZig60RIKKwYBBAGXVQEFAQEHQC4qle0NMBCq3+wJwbZSr0vbuRba  
D1xr4wUPn4Avn4AnAwEIB4h+BBgWCgAmFiEEmvYEF+gnQskUPgPsUNx6jcJMVmcF  
AmYo0tECGwwFCQPCZwAACgkQUNx6jcJMVmcFgEA613RveCM0YHAGQaSFMkguoAo  
vK6Fg0kDawgP0NPIP2oA/jIA04gsAntuQgM0sPunEdDeji2t+AhV02+DQIsXZpoB  
=f8yY
```

```
-----END PGP PUBLIC KEY BLOCK-----
```

- c. Import the Amazon Q command line public key with the following command, substituting public-key-file-name with the file name of the public key you created.

```
gpg --import public-key-file-name  
gpg: directory '/home/username/.gnupg' created  
gpg: keybox '/home/username/.gnupg/pubring.kbx' created  
gpg: /home/username/.gnupg/trustdb.gpg: trustdb created  
gpg: key 50DC7A8DC24C5667: public key "Amazon Q command line Team <q-command  
line@amazon.com>" imported  
gpg: Total number processed: 1  
gpg:                      imported: 1
```

- d. Download the Amazon Q command line signature file for the package you downloaded. It has the same path and name as the .zip file it corresponds to, but has the extension .sig. In the following examples, we save it to the current directory as a file named q.zip.sig.

#### Linux x86-64

For the latest version of the Amazon Q command line, use the following command block:

```
curl --proto '=https' --tlsv1.2 -sSF "https://desktop-release.codewhisperer.us-east-1.amazonaws.com/latest/q-x86_64-linux.zip.sig" -o "q.zip.sig"
```

For a specific version of the Amazon Q command line, replace the latest with the version number. For this example the path for version 1.1.0 would be /1.1.0/q-linux-x86\_64.zip.sig resulting in the following command:

```
q curl --proto '=https' --tlsv1.2 -sF "https://desktop-release.codewhisperer.us-east-1.amazonaws.com/1.1.0/q-x86_64-linux.zip.sig" -o "q.zip.sig"
```

#### Linux ARM (aarch64)

For the latest version of the Amazon Q command line, use the following command block:

```
curl --proto '=https' --tlsv1.2 -sSF "https://desktop-release.codewhisperer.us-east-1.amazonaws.com/latest/q-aarch64-linux.zip.sig" -o "q.zip.sig"
```

For a specific version of the Amazon Q command line, replace the latest with the version number. For this example the path for version 1.1.0 would be /1.1.0/q-linux-aarch64.zip.sig resulting in the following command:

```
curl --proto '=https' --tlsv1.2 -sF "https://desktop-release.codewhisperer.us-east-1.amazonaws.com/1.1.0/q-aarch64-linux.zip.sig" -o "q.zip.sig"
```

For a specific version of the Amazon Q command line, replace the latest with the version number. For this example the path for version 1.1.0 would be /1.1.0/q-linux-aarch64.zip.sig resulting in the following command:

```
curl --proto '=https' --tlsv1.2 -sSf "https://desktop-release.codewhisperer.us-east-1.amazonaws.com/1.1.0/q-aarch64-linux.zip.sig" -o "q.zip.sig"
```

- e. Verify the signature, passing both the downloaded .sig and .zip file names as parameters to the gpg command.

```
gpg --verify q.zip.sig q.zip
```

The output should look similar to the following.

```
gpg: Signature made Wed 24 Apr 2024 12:08:49 AM UTC
gpg:                               using EDDSA key 9AF60417E82742C9143E03EC50DC7A8DC24C566
gpg: Good signature from "Amazon Q command line Team <q-command
line@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                               There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9AF6 0417 E827 42C9 143E 03EC 50DC 7A8D C24C 5667
```

### Note

The warning in the output is expected and doesn't indicate a problem. It occurs because there isn't a chain of trust between your personal PGP key (if you have one) and the Amazon Q for command line PGP key. For more information, see [Web of trust](#).

3. Unzip the installer. If your Linux distribution doesn't have a built-in unzip command, use an equivalent to unzip it. The following example command unzips the package and creates a directory named q under the current directory.

```
unzip q.zip
```

4. Run the install program. The installation command uses a file named install in the newly unzipped q directory. By default, the files are all installed to `~/.local/bin`.

```
./q/install.sh
```

5. Install ssh config integrations. To do this you must edit your `sshd_config` to add the `AcceptEnv` and `AllowStreamLocalForwarding` setting. To edit the `sshd_config` run:

```
sudo -e /etc/ssh/sshd_config
```

Once you are editing the sshd\_config add the following to the end of the config file:

```
AcceptEnv Q_SET_PARENT  
AllowStreamLocalForwarding yes
```

Note, you must also restart the sshd process after installing, if you are using systemd you may run:

```
sudo systemctl restart sshd
```

6. To finish setting up the integrations you must disconnect from the SSH instance and reconnect. Once you do you can login to Q by running:

```
q login
```

To check for any other installation issues run:

```
q doctor
```

## Known limitations

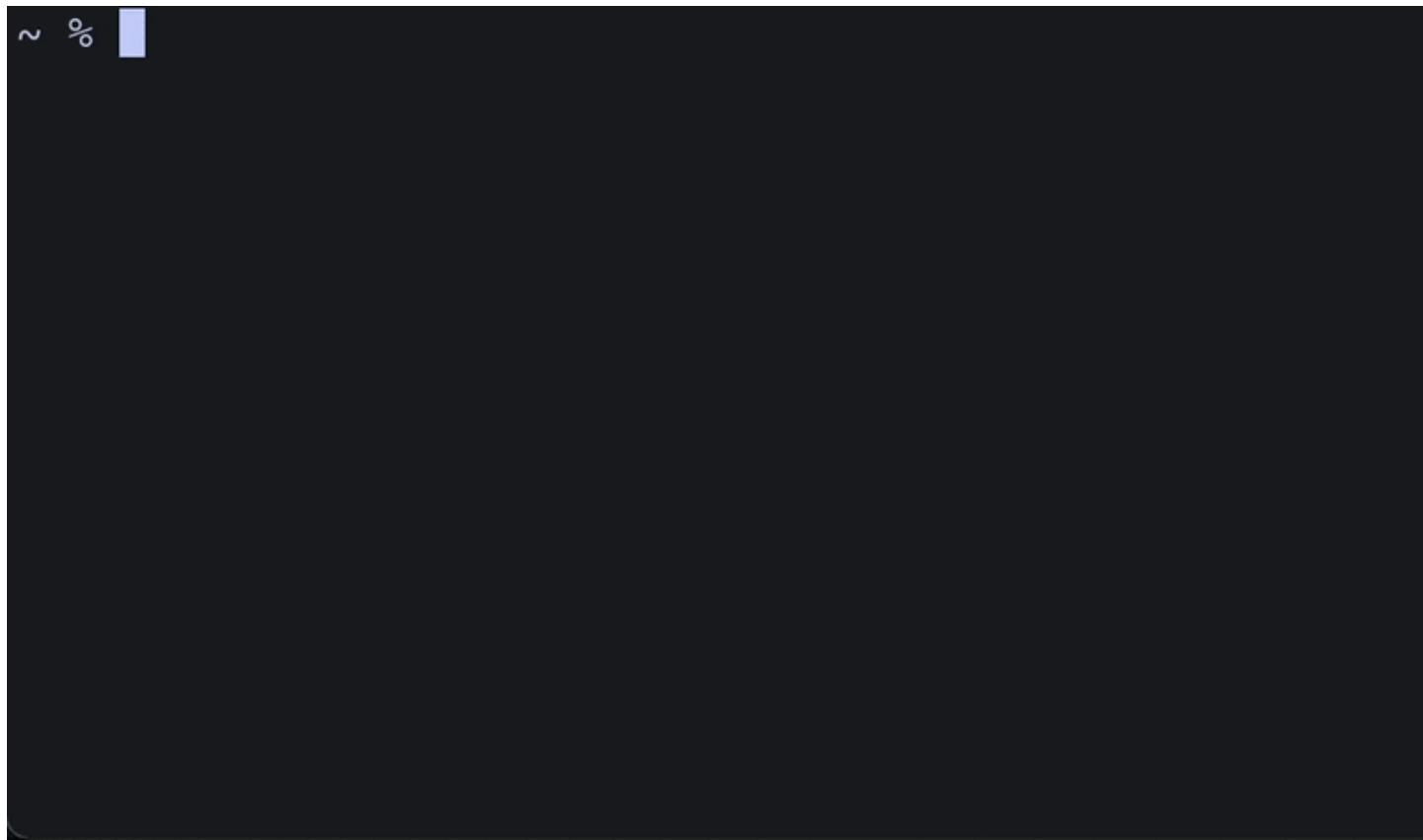
A known limitation is that if the Amazon Q desktop client is quit while connected to a remote machine via SSH, an error message will print repeatedly by SSH. For example:

```
connect to /var/folders/tg/u1vx4xfmvqav0oxfa4zfknaxiwmbbsbr/T/cwrun/remote.sock port -2  
failed: Connection refused
```

To get rid of the error message, either exit the SSH session and reconnect or restart the Amazon Q desktop client.

## Amazon Q inline on the command line

The Amazon Q command line provides AI-generated completions as you type in the command line.



## Using Amazon Q inline

To start using the Amazon Q command line:

1. [Install the Amazon Q command line.](#)
2. Open a new terminal or command prompt.
3. When you start typing Amazon Q will make suggestions based on your current input and previous commands. To accept a suggestion, press the right arrow key. The suggestion will be inserted into your shell.

To disable the inline feature for new terminal sessions, run the following command:

```
q inline disable
```

This command will only affect new terminal sessions, and the inline feature will remain enabled in any existing terminal windows.

## Amazon Q inline Limitations

Currently, inline is only supported in zsh on macOS, and on Linux via the SSH integration.

The inline feature is known to conflict with some popular shell extensions, including zsh-autosuggestions. If you have zsh-autosuggestions installed, you must disable it to use Amazon Q inline in the command line.

## Translating from natural language to bash

The `q translate` command lets you write a natural language instruction such as “copy all files in my current directory to Amazon S3”. Amazon Q will then translate it to an instantly executable shell code snippet. The `q translate` command is useful in those common situations where the correct bash syntax is easy to forget. Examples include reversing a git commit, finding strings inside files with `grep`, or compressing files with `tar`.

To get started, run either of the following

- `q translate prompt`
- `# prompt`

To opt out of using `#` to invoke Amazon Q, go to **Settings -> Translate** and toggle off **Hashtag substitution**.

## Debugging Amazon Q Developer for the command line

If you're having a problem with Amazon Q Developer for command line, run `q doctor`.

`q doctor` identifies and fixes common issues. Most of the time, you won't need to do anything else.

## Expected output

```
$ q doctor  
# Everything looks good!
```

Amazon Q still not working? Run `q issue` to let us know!

If your output doesn't look like the output above, follow the prompts to resolve your issue. If it's still not working, run `q issue` to report the bug.

## Contributing completion specs to Amazon Q Developer

A completion spec is a declarative schema that specifies the subcommands, options and arguments for a command line tool. Amazon Q Developer for command line uses these schemas to generate suggestions.

You can contribute a completion spec to the open source repository to improve the quality of completions. To add your spec, see the [Amazon Q Developer for command line GitHub repository..](#)

# Customizing suggestions

## Note

Customizations are supported for the following features of Amazon Q Developer:

- [Generating inline suggestions](#)
- [Chatting about code](#)

With customizations, Amazon Q Developer can assist with software development in ways that conform to your team's internal libraries, proprietary algorithmic techniques, and enterprise code style.

An Amazon Q customization is a set of elements that enables Amazon Q to provide you with suggestions based on your company's codebase. You connect a data source that contains your code base, and Amazon Q uses your content to provide assistance that caters to the style of your organization's developers.

Your customizations are fully isolated from each other within your account. They are also isolated from the data of other customers. Only users [specified by a Amazon Q Developer administrator](#) have access to any specific customization. Before a Amazon Q administrator can specify which users can access which customizations, you must authorize that administrator permission to do so. For more information, see [Prerequisites for Amazon Q customizations](#).

## Topics

- [Prerequisites for Amazon Q customizations](#)
- [Creating your customization](#)
- [Deleting your customization](#)
- [Evaluating and optimizing your customization](#)
- [Logging and troubleshooting](#)
- [Activating your Amazon Q customizations](#)
- [Updating your Amazon Q customizations](#)
- [Adding users and groups to your Amazon Q customizations](#)
- [Using Amazon Q customizations](#)

# Prerequisites for Amazon Q customizations

Amazon Q customizations build upon the foundation of Amazon Q Developer Pro, and uses its features.

To use Amazon Q customizations you must first follow the Amazon Q Developer Pro setup process under [Getting started with Amazon Q Developer](#). This includes adding any users to your Amazon Q Developer Pro profile that you also wish to grant access to Amazon Q Customizations.

## Authorizing your administrator

When you use Amazon Q Customizations, your Amazon Q administrator must be authorized to access your codebase, which you can store on Amazon S3 or through AWS CodeConnections. However, during the standard setup process for Amazon Q Developer Pro, your AWS Organizations administrator does not provide the Amazon Q administrator with access to those services.

This means that to create customizations, administrators need additional permissions. For an example policy that grants the needed permissions, see [Allow administrators to create customizations](#).

### Note

If you are using GitHub as your data source, you can restrict usage to certain repositories. See [Create a connection to GitHub](#) in the *Developer Tools Console User Guide*.

### Note

The encryption key that you set up for Amazon Q Developer Pro is also used for customizations.

## Preparing your data

It's important to create your customization using the best possible source material. When preparing your data source, add code containing patterns that are encouraged on your team. Avoid code containing anti-patterns, bugs, security vulnerabilities, performance issues, and so forth.

To prepare your data source, follow these guidelines:

- Your data source must contain at least 2 MB, and at most 20 GB, of source code files from supported languages. Any file in your data source that's larger than 10 MB will be ignored.
- There is no limit on the number of files in your data source, but you must include at least 10 files for each language that you want your customization to support.
- File names and individual directory names must not exceed 255 characters. (Cumulatively, they can exceed 255 characters.) Exceeding these limits causes the customization creation to fail.
- In the Amazon S3 data source, all source code must be placed within a directory and not at the root level. Any files at the root level will be ignored.

 **Note**

Amazon Q uses the following file types to train customizations:

- Java (.java)
- JavaScript (.js, .jsx)
- Markdown (.md, .mdx)
- Python (.py)
- reStructuredText (.rst)
- Text (.txt)
- TypeScript (.ts, .tsx)

You can store information about the creation of your customizations in Amazon CloudWatch Logs. For more information, see [Accessing customization-related messages in Amazon CloudWatch Logs](#).

## Creating your customization

This section explains how to create a customization with Amazon Q.

To create your customization, follow this procedure:

1. Complete your setup of Amazon Q Developer Pro. This includes enabling IAM Identity Center and authorizing an administrator to Amazon Q Developer, and activating the Amazon Q Developer console.

2. Open the Amazon Q Developer console.
3. From the navigation pane on the left, choose **Customizations**.
4. The customizations page will appear.
5. Choose **Create customization**.
6. Enter a customization name and (optional) description.

**Note**

Use both names and descriptions that will be informative to your developers.

Developers from your organization who are authorized to use Amazon Q Developer Pro will be able to see them in their IDE through the AWS plugin.

## Connecting to your data source

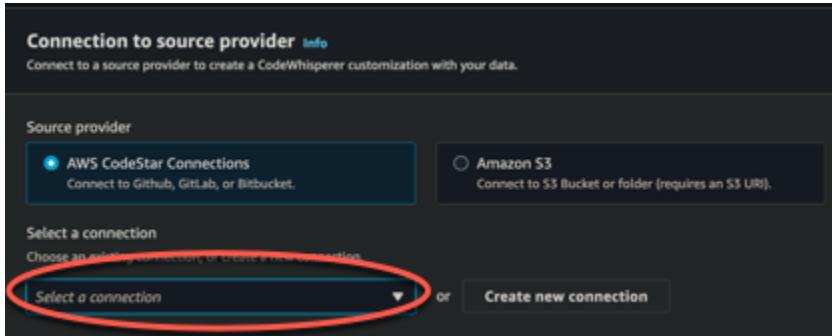
Before you create a customization, you must connect to the data source that contains your codebase. How you do this depends on where your data source is.

If your data source is in Github, GitLab, or Bitbucket, then you can connect to it with AWS CodeConnections. Otherwise, place your data in a folder within an Amazon S3 bucket.

To learn more about CodeConnections, see [What are connections?](#) in the *Developer Tools console User Guide*.

To connect to your data source through CodeConnections, follow this procedure:

1. Under **Connection to source provider**, select **AWS CodeStar CodeConnections**.
2. If you want to use an existing connection, choose **Select a connection**.



Then, under **Choose repository selection**, do one of the following:

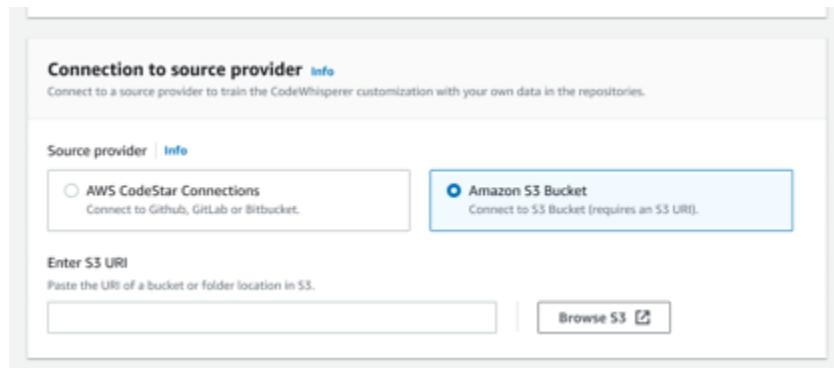
- To use all the repositories in the connection to train the customization, choose **Use all repositories in this connection**.
- To select specific repositories to train the customization, choose **Select specific repositories** and then choose **Choose repositories**. In the the pop-up window, find the repositories you want to use, and then choose **Add**.

You can select a maximum of 100 repositories. If you want to use more than 100 repositories, place them in an Amazon S3 bucket and then follow the instructions for connecting your data source through Amazon S3.

3. If you want to create a new connection, choose **Create a new connection** and follow the remaining steps of this procedure.
4. In the pop-up window that opens, navigate to your data source and follow the instructions in the console.
5. After you create your data source, return to the **Create customization** page.
6. Under **Select a connection**, select your connection from the dropdown.

To connect to your data source through Amazon S3, follow this procedure:

1. Under **Connection to source provider**, select Amazon S3.



2. Choose **Browse Amazon S3**.
3. Navigate to your codebase and make a note of the URI. The codebase must be in a folder within the Amazon S3 bucket, not the bucket's root.

For more information, see [Creating, configuring, and working with Amazon S3 buckets](#) and [Access control best practices](#) in the [Amazon S3 User Guide](#).

4. Paste the URL into the field labeled **Enter Amazon S3 URI**.

Before you create your customization, you have the option of adding tags to it.

To learn more about tags, see [the Tagging your AWS resources User Guide](#).

After following the procedures above, choose **Create customization**.

## Customizations and your data

Amazon Q customizations use your content to present suggestions to you in the style of your organization's developers.

However, AWS will not store or use your content in any context that does not directly serve your enterprise.

AWS will not use your content to provide code suggestions to other customers.

Amazon Q will not reference [security scans](#) for other customers (or for you).

For more information, see [Amazon Q Developer service improvement](#).

## Troubleshooting the creation of your customization

- You may receive the error: Total size of the provided repositories exceeds the maximum allowed size of *number* for a customization.  
In that case, remove a repository from your data source and try again.
- You may receive the error: Insufficient data to create a customization. Add more files from supported languages and retry.

In order for code written in a particular language to be used to create a customization, there must be at least 10 files containing code in that language in your data source. Your data source must contain at least 2 MB, while it is recommended to have close to 20MB, and at most 20 GB, of source code files from supported languages.

Some files, even if they are in the relevant language, will not count toward the 10 MB. For example, duplicate files and files in an unsupported format will not be counted.

If you receive this error, add more files containing the programming language that is the focus of your customization, and try again.

- You may receive the error: Encountered an issue when retrieving some of the selected repositories from CodeConnections. Check the customization's log deliveries for details.

If you receive this error, try creating or updating the customization again with valid repositories that your connection has access to.

## Deleting your customization

This section explains how to delete a customization with Amazon Q.

### Warning

Deleting a customization will delete all versions associated with the resource.

To delete your customization, follow this procedure:

1. Open the Amazon Q Developer console.
2. From the navigation pane on the left, choose **Customizations**.
3. The customizations page will appear.
4. If the customization that you want to delete is still active, choose **Deactivate**.
5. Choose **Delete**.

### Note

You can also delete a customization from the page that gives the details of that customization.

To do that, just choose **Delete** from the upper right corner of the customization detail page.

## Evaluating and optimizing your customization

This page explains how to evaluate and optimize your customization.

## Evaluating your customization

This section explains how to evaluate your customization.

1. In the Amazon Q Developer console, from the navigation panel, choose **Customizations**.
2. Choose the name of the customization to examine.
3. The right side of the window will display an evaluation score. This score indicates Amazon Q's evaluation of how effective your customization may be.

With your evaluation score in mind, you must now consider whether or not to activate your customization. In making this decision, take the following factors into consideration.

- **Very good 8-10:** Amazon Q recommends that you activate this customization.
- **Fair 5-7:** Amazon Q recommends that you activate this customization.

If you do not see significant improvement, consider the optimization suggestions below. If those are not effective, consider switching to a different code source.

- **Poor 1-4:** The customization may not be effective. To improve the quality of customization, consider the optimization suggestions below. If those are not effective, consider switching to a different code source.

## Optimizing your customization

This section contains suggestions for optimizing your suggestion in order to achieve a higher evaluation score.

- Consider expanding your data source to include more code repositories.
- If you primarily included data from limited programming languages, consider expanding to more languages.
- Remove auto-generated files and repositories, or those generated from templates. Training a customization to generate or complete such files is typically not valuable, and tends to just add noise.
- It is possible that your codebase does not frequently use internal libraries. If you know this to be true, then the core Amazon Q model may already have been performing as well as possible.

## Optimizing for the languages you use

In order for code in a particular language to be used in a customization, you must include at least 20 data files containing that language, and all of your source files together must come to at least 2 MB. If your developers write code in a language that is not supported by your customization, Amazon Q's recommendations in that language will come from the Amazon Q base model (not your customization). In other words, they will be the same recommendations that you would receive if you did not have a customization. This, in turn, could affect the metrics on your dashboard. For example, the "Lines of code generated by Amazon Q" may be less than what it would have been if the language commonly used by your developers had been included in your customization.

## Logging and troubleshooting

### Setting up log delivery

Amazon Q can provide you with log files that will help you understand and troubleshoot issues with your customization.

You can have your log files sent to a [Amazon CloudWatch Logs](#) group, an [Amazon S3](#) bucket, an [Amazon Data Firehose](#), or any combination.

To set up log delivery, select the Log deliveries tab on the console page for your customization. Follow the instructions in the interface to configure your log deliveries. Then choose **Create log deliveries**.

The prefix of logs delivered to an Amazon S3 bucket will be: AWSLogs/*account\_id*/codeWhispererCustomizationLogs/*region*/*customization\_id*/year/*month*/*day*/*hour*/

The files will be zipped, with the naming format:

*account\_id*\_codeWhispererCustomizationLogs\_*customization\_id\_date\_file\_id*.log.gz

#### Warning

In order to get the most use out of customization logs, it's best to set up log delivery within five minutes of creating the customization.

To learn more about the permissions required to delivery logs to multiple resources, see [Logging that requires additional permissions \[V2\]](#) in the *Amazon CloudWatch Logs User Guide*.

## Understanding customization-related log messages

The following table lists log messages that may help you understand issues with your customization.

Log message	Log level
Starting to ingest <i>number</i> repos from source <i>source</i>	Info
Downloading data from repo: <i>repo name</i>	Info
Received <i>amount</i> MB of supported data. <i>amount</i> MB required. Add more data and retry.	Error
The provided CodeStar Connection ARN: <i>Arn</i> is invalid.	Error
Access denied when attempting to reach the provided CodeStar Connection: <i>Arn</i>	Error
Failed to download with AWS CodeStar Connection: <i>Arn</i> probably deleted by customer	Error
ProviderThrottlingException from CodeStar Connection: <i>Arn</i> while cloning repository: <i>repository</i>	Error
Processing data from S3: <i>S3 URI</i>	Info
Invalid S3 path specified: <i>S3 Directory</i>	Error

Log message	Log level
Unable to access the provided S3 bucket: <i>bucket name</i>	Error
The provided S3 bucket: <i>bucket name</i> does not exist.	Error
The provided S3 key <i>S3 URI</i> does not exist.	Error
Failed to ingest <i>number of failed repos / total number of repos</i> repositories	Error
Unable to process repository: <i>repo name</i> , with a size of <i>repo size</i> GB, exceeds the limit of <i>max size</i> GB.	Warn
Unable to process file: <i>file name</i> , with a size of <i>file size</i> , which exceeds the limit of <i>max file size</i> MB	Error
Unable to process collection: <i>collection name</i> , with total size of <i>total repo size</i> MB, which exceeds the limit of <i>max total repo size</i> MB	Error
The following languages will be used for customization: <i>list of languages</i> . Languages may be excluded from customization if they are not sufficiently represented in your files.	Info

## Understanding customization-related error messages in the console

The following table will help you understand customization-related messages in the Amazon Q console.

Error message	Suggested action
You have activated the maximum number of customizations.	Deactivate an active customization and try again.
You have exceeded the maximum number of group permissions limit of <i>limit</i> .	Remove a group and retry.
You have exceeded the maximum number of user permissions limit of <i>limit</i> .	Remove a user and retry.
Maximum active jobs reached.	Wait until an in-progress job in the same account has finished. Retry the operation.
Encountered an unexpected error when processing the request.	Retry the operation. If it continues to fail, contact customer support.
Encountered an issue when retrieving some of the selected repositories from CodeConnections. Check the customization's log deliveries for details.	Try creating or updating the customization again with valid repositories that your connection has access to.
Access denied when attempting to reach the provided AWS CodeConnections connection.	Validate permissions on your connection and on your third-party provider. Then retry the operation.
One or more repositories not found while accessing the provided AWS CodeConnections connection.	Validate permissions and list of repos from the third-party provider. Then retry the operation.
The provided AWS CodeConnections connection ARN is invalid.	Update the customization with a corrected Connection ARN.

Error message	Suggested action
The Host associated with the provided AWS CodeConnections connection is unavailable.	Try again in 5 minutes.
Invalid Amazon S3 path specified.	Update the customization with a valid Amazon S3 URI.
Unable to access the provided Amazon S3 bucket.	Validate permissions for the admin's role. Retry after fixing any permission issues.
The provided Amazon S3 bucket does not exist.	Update the customization with a valid Amazon S3 URI.
The provided Amazon S3 key does not exist.	Update the customization with a valid Amazon S3 URI.
Insufficient data to create a customization. Add more files from supported languages and retry.	Add more data to the same data source, and update the customization with the same reference.
Total size of the provided repositories exceeds the maximum allowed size of <b>size</b> for a customization.	Remove some data from the provided data source. Update the customization with the same reference.
You have created the maximum number of customizations. Delete an existing customization and try again.	Delete the current customization and retry.
Customizations exist within the account. You must delete all customizations prior to deleting the profile.	Delete all customizations associated with the account and retry.

## Activating your Amazon Q customizations

### Activating a version

This section describes how to activate and deactivate a version of your customization.

You can activate a new version of a customization, even while developers from your organization are using the previous version. After you activate the new version, the developers will seamlessly begin using it, with no adjustments needed on the development side.

You can also roll your customization back to a previously active state. However, Amazon Q does not actually re-activate a previously activated version. Instead, it creates a new version by copying a previous version and then activating the copy.

For example, suppose that you have three versions: 1, 2, and 3. The active version is 3. You decide to go back to version 1. But "re-activating" version 1 is actually just copying version 1 and creating version 4. That's the version you use: version 4, the new copy of the old version.

To activate a version of your customization, follow this procedure:

1. Open the Amazon Q Developer console.
2. From the navigation pane on the left, choose **Customizations**.  
The customizations page will appear.
3. Choose the customization you want to activate a version for.  
The customization details page will appear.
4. Choose the version you want to activate from the **Versions** table.
5. Choose **Activate**.

To deactivate a customization, choose **Deactivate** from the dropdown.

## Updating your Amazon Q customizations

A customization is trained based on a snapshot of your data source at the time the customization was created. You might want to update your Amazon Q customization if:

- You updated the files in your data source, and you want to re-train your customization with the new files.
- You want to switch the data source from AWS CodeConnections to Amazon S3, or the reverse.
- You want to change the repositories referenced in a CodeConnections data source.

A customization can have multiple versions.

Amazon Q administrators have access to a maximum of three versions for each customization:

- the latest version
- the currently active version
- the most recently active version that is not currently active

## Creating a new version

To create a new version of your customization, follow this procedure:

1. Open the Amazon Q Developer console.
2. From the navigation pane on the left, choose **Customizations**.  
The customizations page will appear.
3. Choose the customization for which you want to create a new version.  
The customization details page will appear.
4. Do one of the following:
  - Select **Create new version** from the **Actions** dropdown.
  - Choose the **Sources** tab, and then choose **Update**.

The **Update customization** page appears.

5. Select **Create new version** from the **Actions** dropdown.
6. (Optional) Change the data source.

**Create new version** Info

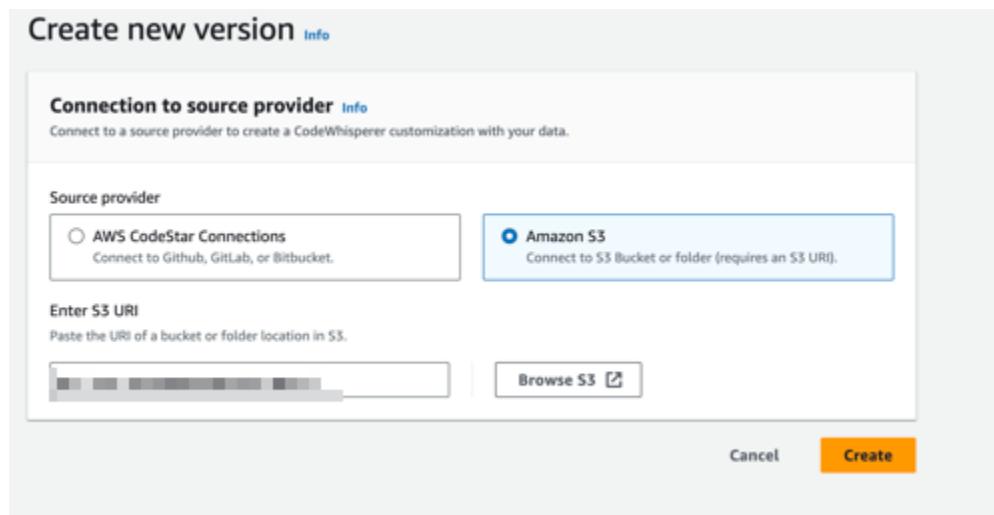
**Connection to source provider** Info  
Connect to a source provider to create a CodeWhisperer customization with your data.

Source provider

AWS CodeStar Connections  
Connect to Github, GitLab, or Bitbucket.

Amazon S3  
Connect to S3 Bucket or folder (requires an S3 URI).

Enter S3 URI  
Paste the URI of a bucket or folder location in S3.



7. (Optional) If you selected the CodeConnections data source, change the repositories associated with the connection.
8. Choose **Create**.

If you receive error messages, see [Troubleshooting the creation of your customization](#).

## Adding users and groups to your Amazon Q customizations

This section contains information about how to add users and groups to customizations.

 **Note**

You must activate a customization before you can add users to it.

 **Note**

You can only add a user or group to a customization if you have already added the user or group to your Amazon Q Developer Pro profile. For more information, see [Amazon Q Developer Pro tier](#).

1. In the Amazon Q Developer console, from the navigation panel, choose **Customizations**.
2. Choose the name of the customization to which you want to add users or groups.
3. In the bottom half of the window, if necessary, select the **Users and groups** tab. and then the **Users** or **Groups** sub-tab.
4. Select the users or groups that require access to your customization.
5. Choose **Add users** or **Add groups**.

## Using Amazon Q customizations

This section contains information about how to use customizations as a developer.

Amazon Q only supports customizations in VS Code and JetBrains IDEs.

## Visual Studio Code

To use customizations with Visual Studio Code:

1. Authenticate to Amazon Q Developer Pro with IAM Identity Center using the steps in [Installing the Amazon Q Developer extension or plugin in your IDE](#).
2. In the **Developer Tools** pane, under Amazon Q, choose **Select Customization**.
3. At the top of the window, from the dropdown menu, select the appropriate customization.

## JetBrains

To use customizations in JetBrains IDEs:

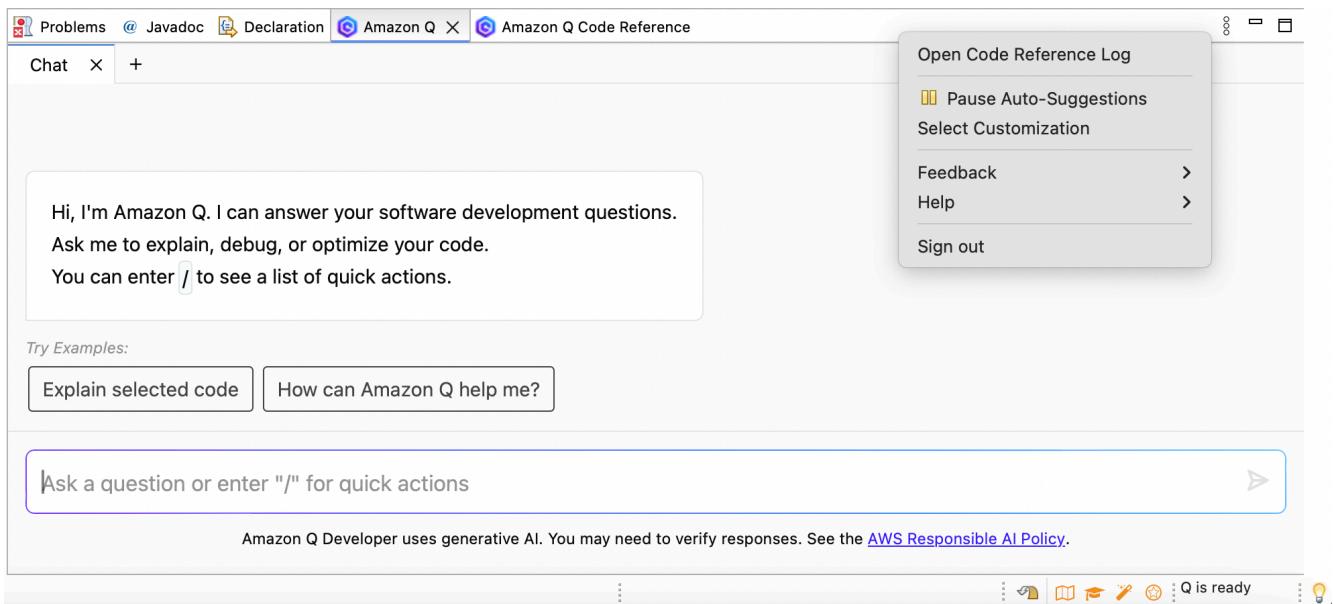
1. Authenticate to Amazon Q Developer Pro with IAM Identity Center using the steps in [Installing the Amazon Q Developer extension or plugin in your IDE](#).
2. In the **Developer Tools** pane, under Amazon Q, choose **Select Customization**.
3. In the pop-up window, select the appropriate customization.
4. Choose **Connect**.

## Eclipse

To use customizations in Eclipse IDEs:

1. Authenticate to Amazon Q Developer Pro with IAM Identity Center using the steps in [Installing the Amazon Q Developer extension or plugin in your IDE](#).
2. In your Eclipse IDE, choose the **Amazon Q** icon in the top right corner of the IDE.
3. With the Amazon Q chat tab open, choose the ellipsis icon in the top right corner of the tab. The Amazon Q task bar opens.

The following image shows the Amazon Q task bar in an Eclipse IDE.



4. Choose **Select Customization**.
5. In the pop-up window, select the appropriate customization.
6. Choose **Select**.

# Amazon Q Developer dashboard

Available only for Amazon Q Developer administrators, the Amazon Q Developer dashboard summarizes useful data about how your Pro tier subscribers use the service. Among the useful metrics is the acceptance rate, which indicates how often you take inline suggestions from Amazon Q.

Amazon Q generates and displays new metrics on an hourly basis.

## To view the dashboard

1. Open the Amazon Q Developer console.
2. From the navigation pane, choose **Dashboard**.
3. (Optional) Filter the information by date range or programming language.

### Notes:

- If the **Dashboard** link is not available in the navigation pane, see [Troubleshooting the dashboard](#).
- If you started using Amazon Q Developer before November 22, 2024, you can switch to the old dashboard to see historical metrics. For more information, see [Switching to the old dashboard](#).
- If you'd like to send user metrics to a daily report with a per-user breakdown of their Amazon Q Developer usage, see [Enabling user activity reports](#).

# Disabling the Amazon Q Developer dashboard

You might want to disable the Amazon Q Developer dashboard if you have concerns about data privacy, page load times, or other potential issues. When you disable the dashboard, the dashboard page (and any links to it) will no longer be available in the Amazon Q Developer console.

For more information about the dashboard, see [Dashboard](#).

## To disable the dashboard

1. Open the Amazon Q Developer console:
  - If you set up Amazon Q Developer with an organization instance of AWS IAM Identity Center, then sign in using a management account or member account.

- If you set up Amazon Q Developer with an account instance of IAM Identity Center, then sign in using the account associated with that instance.
2. Choose **Settings**, and in the **Amazon Q Developer user activity** section, choose **Edit**.
  3. Disable **Amazon Q Developer usage dashboard**.

## Switching to the old dashboard

If you started using Amazon Q Developer before November 22, 2024, you can switch to the old dashboard. You might want to switch to the old dashboard to see historical metrics, such as the number of security scans performed. These metrics are not currently available in the new dashboard.

If you started using Amazon Q after November 22, 2024, you will only be able to see the new dashboard.

### To switch to the old dashboard

1. Open the Amazon Q Developer console.
2. In the navigation pane, choose **Dashboard**.
3. In the banner, choose **Go to the old dashboard**.

For more information about the dashboard, see [Dashboard](#).

## Old dashboard contents

The Amazon Q Developer old dashboard includes sections that describe metrics related to user activity, code impacts, and security scans. You can filter the data by date range. The minimum range is two weeks and the maximum is one year. You can also filter by programming language.

### User activity

The **User activity** section indicates how many of your Amazon Q Developer subscriptions are being used on a daily basis.

### Code impact

*The Lines of code generated by Amazon Q* simply indicates how many lines of code were suggested by Amazon Q and accepted by your developers.

*Accepted recommendations with references* indicates the number of suggestions from Amazon Q that are based on open-source projects, the references to which Amazon Q makes available to you.

If you use Amazon Q very little over a two-week period, the **Code impact** section will be affected as follows:

- If no recommendations are invoked for two weeks, then no data will appear in the **Code impact** section.
- If recommendations are invoked, but none are accepted or rejected, then no data will appear in the **Code impact** section.
- If recommendations are invoked, and none are accepted, but some are rejected, then the **Acceptance rate (0%)** will be displayed, but no data will appear for **Lines of code generated by Amazon Q** or **Accepted recommendations with references**.

## Security scans

Amazon Q security scans identify security vulnerabilities in your developers' code.

The data shown indicates how many scans your developers have successfully run in their IDEs.

## Troubleshooting the Amazon Q Developer dashboard

If the Amazon Q Developer dashboard page is not available, do the following:

- **Verify your permissions.** To view the dashboard, you need the following permissions:
  - For the new dashboard: `q>ListDashboardMetrics`
  - For the old dashboard: `cloudwatch:GetMetricData` and `cloudwatch>ListMetrics`For more information about permissions, see [Allow administrators to use the Amazon Q Developer console](#).
- **Verify your settings.** In the Amazon Q Developer console, choose **Settings** and make sure that the **Amazon Q Developer usage dashboard** toggle is enabled.

For more information about the dashboard, see [Dashboard](#).

# Security in Amazon Q Developer

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

Security is a shared responsibility between AWS and you. The [shared responsibility model](#) describes this as security of the cloud and security in the cloud:

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

This documentation helps you understand how to apply the shared responsibility model when using Amazon Q Developer. It shows you how to configure Amazon Q to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon Q resources.

## Topics

- [Data protection in Amazon Q Developer](#)
- [Identity and access management for Amazon Q Developer](#)
- [Compliance validation for Amazon Q Developer](#)
- [Resilience in Amazon Q Developer](#)
- [Infrastructure security in Amazon Q Developer](#)
- [Amazon Q Developer and interface VPC endpoints \(AWS PrivateLink\)](#)

## Data protection in Amazon Q Developer

The AWS [shared responsibility model](#) applies to data protection in Amazon Q Developer. As described in this model, AWS is responsible for protecting the global infrastructure that runs all

of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the [Data Privacy FAQ](#). For information about data protection in Europe, see the [AWS Shared Responsibility Model and GDPR](#) blog post on the [AWS Security Blog](#).

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

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-2](#).

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Amazon Q or other AWS services using the AWS Management Console, API, AWS Command Line Interface (AWS CLI), or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs.

Amazon Q stores your questions, its responses, and additional context, such as console metadata and code in your IDE, to generate responses to your questions. Your code is also stored for features like code transformation and software development in the IDE. This data is stored for up to 90 days to provide the service, and then is permanently deleted. For information about how data is encrypted, see [Data encryption in Amazon Q Developer](#). For information about how AWS may use some questions that you ask Amazon Q and its responses to improve our services, see [Amazon Q Developer service improvement](#).

Regardless of where you use Amazon Q Developer, data is sent to and stored in an AWS Region in the US. With [cross-region inferencing](#), your requests to Amazon Q Developer may be processed

in the US East (N. Virginia) Region, the US West (Oregon) Region, or the US East (Ohio) Region, even if the AWS Management Console is set to a different AWS Region. Data processed during troubleshooting console error sessions is stored in the US West (Oregon) Region. All other data is stored in the US East (N. Virginia) Region.

## Topics

- [Data encryption in Amazon Q Developer](#)
- [Amazon Q Developer service improvement](#)
- [Opt out of data sharing in the IDE](#)
- [Cross-Region calls in Amazon Q Developer](#)
- [Cross region inference in Amazon Q Developer](#)

## Data encryption in Amazon Q Developer

This topic provides information specific to Amazon Q Developer about encryption in transit and encryption at rest.

### Encryption in transit

All communication between customers and Amazon Q and between Amazon Q and its downstream dependencies is protected using TLS 1.2 or higher connections.

### Encryption at rest

Amazon Q stores data at rest using Amazon DynamoDB and Amazon Simple Storage Service (Amazon S3). The data at rest is encrypted using AWS encryption solutions by default. Amazon Q encrypts your data using AWS owned encryption keys from AWS Key Management Service (AWS KMS). You don't have to take any action to protect the AWS managed keys that encrypt your data. For more information, see [AWS owned keys](#) in the *AWS Key Management Service Developer Guide*.

For subscribers to Amazon Q Developer Pro, administrators can set up encryption with customer managed KMS keys for data at rest for the following features:

- Chat in the AWS console
- Diagnosing AWS console errors
- Customizations

- Agent for software development
- Agent for code transformation
- Security scans

You can only encrypt data with a customer managed key for the listed features of Amazon Q in the AWS console and the IDE. Your conversations with Amazon Q on the AWS website, AWS Documentation pages, and in chat channels integrated with AWS Chatbot are only encrypted with AWS-owned keys.

Customer managed keys are KMS keys in your AWS account that you create, own, and manage to directly control access to your data by controlling access to the KMS key. Only symmetric keys are supported. For information on creating your own KMS key, see [Creating keys](#) in the *AWS Key Management Service Developer Guide*.

When you use a customer managed key, Amazon Q Developer makes use of KMS grants, allowing authorized users, roles, or applications to use a KMS key. When an Amazon Q Developer administrator chooses to use a customer managed key for encryption during configuration, a grant is created for them. This grant is what allows the end user to use the encryption key for data encryption at rest. For more information on grants, see [Grants in AWS KMS](#).

If you change the KMS key used to encrypt chats with Amazon Q in the AWS console, you must start a new conversation to begin using the new key to encrypt your data. Your conversation history that was encrypted with the previous key won't be retained in future chats, and only future chats will be encrypted with the updated key. If you want to maintain your conversation history from a previous encryption method, you can revert to the key you were using during that conversation. If you change the KMS key used to encrypt diagnosing console error sessions, you must start a new diagnose session to begin using the new key to encrypt your data.

## Using customer managed KMS keys

After creating a customer managed KMS key, an Amazon Q Developer administrator must provide the key in the Amazon Q Developer console to use it to encrypt data. For information on adding the key in the Amazon Q Developer console, see [Manage encryption method](#).

To set up a customer managed key to encrypt data in Amazon Q Developer, administrators need permissions to use AWS KMS. The required KMS permissions are included in the example IAM policy, [Allow administrators to use the Amazon Q Developer console](#).

To use features that are encrypted with a customer managed key, users need permissions to allow Amazon Q to access the customer managed key. For a policy that grants the needed permissions, see [Allow Amazon Q access to customer managed keys](#).

If you see an error related to KMS grants while using Amazon Q Developer, you likely need to update your permissions to allow Amazon Q to create grants. To automatically configure the needed permissions, go to the Amazon Q Developer console and choose **Update permissions** in the banner at the top of the page.

## Amazon Q Developer service improvement

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

### Amazon Q Developer Free tier content used for service improvement

We may use certain content from Amazon Q Developer Free tier for service improvement. Amazon Q may use this content, for example, to provide better responses to common questions, fix Amazon Q operational issues, for de-bugging, or for model training.

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

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

### How to opt out

The way you opt out of Amazon Q Developer Free Tier using content for service improvement depends on the environment where you use Amazon Q.

For the AWS Management Console, AWS Console Mobile Application, AWS websites, and AWS Chatbot, configure an AI services opt-out policy in AWS Organizations. For more information, see [AI services opt-out policies](#) in the *AWS Organizations User Guide*.

In the IDE, for Amazon Q Developer Free Tier, adjust your settings in the IDE. For more information, see [Opt out of data sharing in the IDE](#).

## Opt out of data sharing in the IDE

This page explains how to opt out of sharing your data in the IDE where you use Amazon Q, including third-party IDEs and AWS coding environments. For information on how Amazon Q uses this data, see [Amazon Q Developer service improvement](#).

### Opting out of sharing your client-side telemetry

Your client-side telemetry quantifies your usage of the service. For example, AWS may track whether you accept or reject a recommendation. Your client-side telemetry does not contain actual code.

To learn more about the telemetry data collected by Amazon Q in the IDE, see the [commonDefinitions.json](#) document in the `aws-toolkit-common` Github repository.

For detailed information about the telemetry data collected by each IDE where you use Amazon Q, reference the resource documents in the following GitHub repositories:

- [Amazon Q extension for VS Code](#)
- [Amazon Q plugin for JetBrains](#)
- [Amazon Q plugin for Eclipse](#)
- [AWS Visual Studio Toolkit with Amazon Q](#)

 **Note**

Don't add personally identifiable information (PII) or other confidential or sensitive information in free text fields in the IDE.

Choose your IDE for instructions on opting out of sharing your client-side telemetry.

#### Visual Studio Code

To opt out of sharing your telemetry data in VS Code, use this procedure:

1. Open **Settings** in VS Code.
2. If you are using VS Code workspaces, switch to the **Workspace** sub-tab. In VS Code, workspace settings override user settings.

3. In the Settings search bar, enter Amazon Q: Telemetry.
4. Deselect the box.

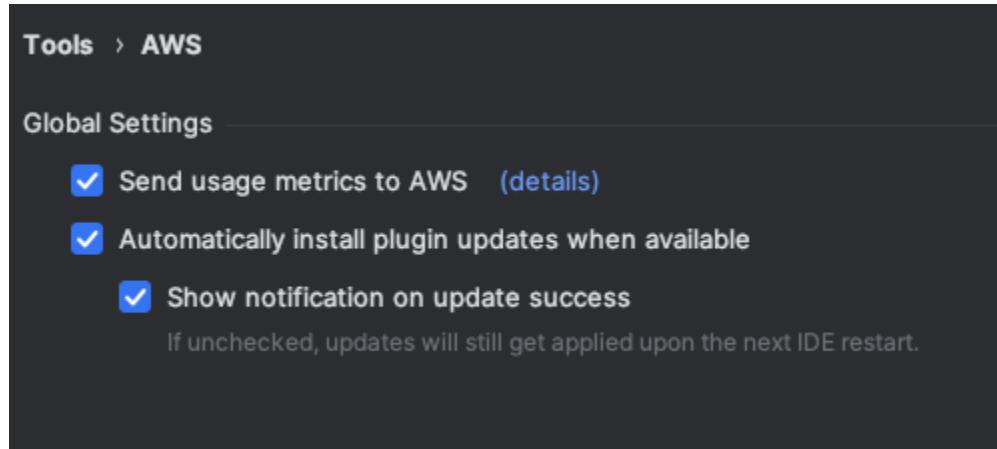
**Note**

This is a decision for each developer to make inside their own IDE. If you are using Amazon Q as part of an enterprise, your administrator will not be able to change this setting for you.

## JetBrains

To opt out of sharing your telemetry data in JetBrains, use this procedure:

1. In your JetBrains IDE, open **Preferences** (on a Mac, this will be under **Settings**).
2. In the left navigation bar, choose **Tools**, and then choose **AWS**.
3. Deselect **Send usage metrics to AWS**.



**Note**

This is a decision for each developer to make inside their own IDE. If you are using Amazon Q as part of an enterprise, your administrator will not be able to change this setting for you.

## Eclipse

To opt out of sharing your telemetry data in Eclipse IDEs, use this procedure:

1. Open **Settings** in your Eclipse IDE.
2. Choose **Amazon Q** from the left navigation bar.
3. Deselect the box next to **Send usage metrics to AWS**.
4. Choose **Apply** to save your changes.

 **Note**

This is a decision for each developer to make inside their own IDE. If you are using Amazon Q as part of an enterprise, your administrator will not be able to change this setting for you.

## Visual Studio

To opt out of sharing your telemetry data in the AWS Toolkit for Visual Studio, use this procedure:

1. Under **Tools**, choose **Options**.
2. In the **Options** pane, choose **AWS Toolkit**, and then choose **General**.
3. Deselect **Allow AWS Toolkit to collect usage information**.

 **Note**

This is a decision for each developer to make inside their own IDE. If you are using Amazon Q as part of an enterprise, your administrator will not be able to change this setting for you.

## AWS Cloud9

1. From inside your AWS Cloud9 IDE, choose the AWS Cloud9 logo at the top of the window, then choose **Preferences**.

2. On the **Preferences** tab choose **AWS Toolkit**.
3. Next to **AWS: client-side telemetry**, toggle the switch to the off position.

 **Note**

This setting affects whether or not you share your AWS Cloud9 client-side telemetry in general, not just for Amazon Q.

## Lambda

When you use Amazon Q with Lambda, Amazon Q does not share your client-side telemetry with AWS.

## SageMaker Studio

1. From the top of the SageMaker Studio window choose **Settings**.
2. From the **Settings** dropdown, choose **Advanced Settings Editor**.
3. In the Amazon Q dropdown, select or deselect the box next to **Share usage data with Amazon Q**.

## JupyterLab

1. From the top of the JupyterLab window choose **Settings**.
2. From the **Settings** dropdown, choose **Advanced Settings Editor**.
3. In the Amazon Q dropdown, select or deselect the box next to **Share usage data with Amazon Q**.

## AWS Glue Studio Notebook

1. From the bottom of the AWS Glue Studio Notebook window choose **Amazon Q**.
2. From the pop-up menu, toggle the switch next to **Share telemetry with AWS**.

**Note**

Pausing the sharing of client-side telemetry will be valid only for the duration of the current AWS Glue Studio Notebook.

## Command line

In the command line tool, under **Preferences**, toggle **Telemetry**.

## Opting out of sharing your content

For information on content AWS uses, see [Amazon Q Developer service improvement](#).

## Visual Studio Code

At the Amazon Q Developer Pro Tier, Amazon Q does not collect your content.

At the Amazon Q Developer Free Tier, to opt out of sharing your content in VS Code, use the following procedure.

1. Open **Settings** in VS Code.
2. If you are using VS Code workspaces, switch to the **Workspace** sub-tab. In VS Code, workspace settings override user settings.
3. In the Settings search bar, enter Amazon Q: Share Content.
4. Deselect the box.

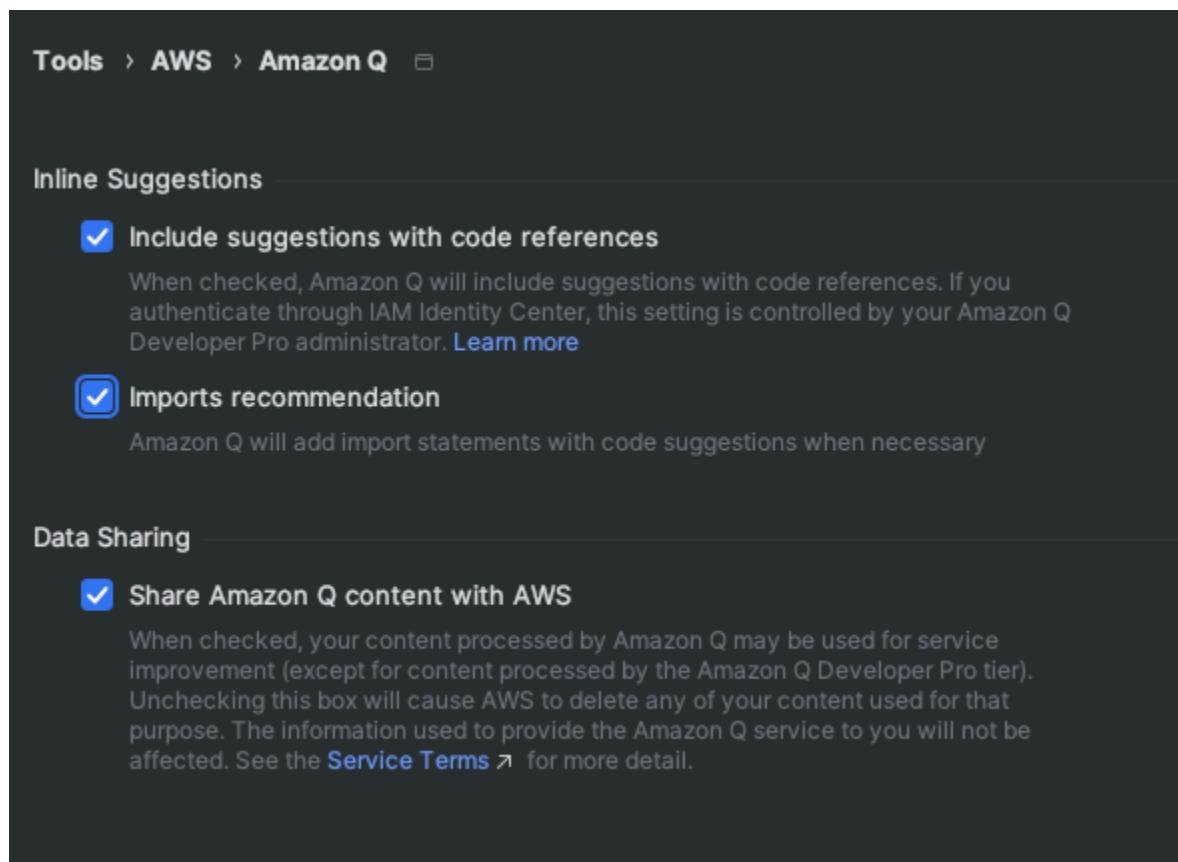
## JetBrains

At the Amazon Q Developer Pro Tier, Amazon Q does not collect your content.

At the Amazon Q Developer Free Tier, to opt out of sharing Amazon Q data in JetBrains, use the following procedure.

1. Make sure you are using the latest version of JetBrains.
2. In your JetBrains IDE, open **Preferences** (on a Mac, this will be under **Settings**).
3. In the left navigation bar, choose **Tools --> AWS --> Amazon Q**.

4. Under **Data sharing**, deselect **Share Amazon Q content with AWS**.



## Eclipse

At the Amazon Q Developer Pro tier, Amazon Q does not collect your content.

At the Amazon Q Developer Free tier, to opt out of sharing Amazon Q data in Eclipse IDEs, use the following procedure.

1. Make sure you are using the latest version of your Eclipse IDE.
2. In your Eclipse IDE, open **Settings**.
3. In the left navigation bar, choose **Amazon Q**.
4. Deselect the box next to **Share Amazon Q content with AWS**.
5. Choose **Apply** to save your changes.

## Visual Studio

At the Amazon Q Developer Pro Tier, Amazon Q does not collect your content.

At the Amazon Q Developer Free Tier, to opt out of sharing your content in Visual Studio, use the following procedure.

Bring up the Amazon Q options menu one of two ways:

- Choose the AWS Toolkit icon from the edge of the window, then choose **Options...**
- Go to **Tools -> Options -> AWS Toolkit -> Amazon Q**

Toggle **Share Amazon Q Content with AWS** to **True** or **False**.

## AWS Cloud9

When you use Amazon Q with AWS Cloud9, Amazon Q does not share your content with AWS.

### Note

The AWS Cloud9 settings do contain a toggle switch for sharing Amazon Q content with AWS, but that switch is non-functional.

## Lambda

When you use Amazon Q with Lambda, Amazon Q does not share your content with AWS.

## SageMaker Studio

When you use Amazon Q with SageMaker Studio, Amazon Q does not share your content with AWS.

## JupyterLab

1. From the top of the JupyterLab window choose **Settings**.
2. From the **Settings** dropdown, choose **Advanced Settings Editor**.
3. In the Amazon Q dropdown, select or deselect the box next to **Share content with Amazon Q**.

## AWS Glue Studio Notebook

When you use Amazon Q with AWS Glue Studio Notebook, Amazon Q does not share your content with AWS.

## Command line

In the command line tool, under **Preferences**, toggle **Share Amazon Q content with AWS**.

## Cross-Region calls in Amazon Q Developer

While using Amazon Q Developer, you might be asked to consent to cross-Region calls. Cross-Region calls are API calls made by Amazon Q from one AWS Region to another AWS Region.

Amazon Q Developer is hosted in US East (N. Virginia), but can make calls to Regions listed on the Supported Regions page, including opt-in Regions. You must consent to cross-Region calls to use Amazon Q Developer features that access your AWS resources, since Amazon Q Developer might have to make calls to other Regions to access your resources.

You must consent to cross-Region calls for each feature that requires consent. If you don't provide consent, Amazon Q Developer will not make cross-Region calls.

When you use a feature of Amazon Q Developer that requires consent to cross-Region calls, you will be prompted to give consent. The way you update cross-Region consent settings depends on the feature you're using. See the topic for the feature you're using for information about updating your cross-Region preferences.

To update cross-Region settings in Amazon Q chat, see [Chat settings](#).

## Cross region inference in Amazon Q Developer

Amazon Q Developer is powered by Amazon Bedrock, and uses cross-region inference to distribute traffic across different AWS Regions to enhance large language model (LLM) inference performance and reliability. With cross-region inference, you get:

- Increased throughput and resilience during high demand periods
- Improved performance
- Access to newly launched Amazon Q Developer capabilities and features that rely on the most powerful LLMs hosted on Amazon Bedrock

Today, regardless of where you use Amazon Q Developer, your data is processed in a US Region. With cross-region inferencing, your requests to Amazon Q Developer may be processed in any of our US regions (currently US East (N. Virginia) Region, the US West (Oregon) Region, or the US

East (Ohio) Region), even if you are using Amazon Q Developer in a different AWS Region. For information on where data is stored during processing, see [Data protection](#). For information on where you can use Amazon Q Developer, see [Supported Regions for Amazon Q Developer](#). There's no additional cost for using cross-region inference.

## Identity and access management for Amazon Q Developer

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

### Topics

- [Audience](#)
- [Authenticating with identities](#)
- [Managing access using policies](#)
- [How Amazon Q Developer works with IAM](#)
- [Manage access to Amazon Q Developer with policies](#)
- [Amazon Q Developer permissions reference](#)
- [AWS managed policies for Amazon Q Developer](#)
- [Using service-linked roles for Amazon Q Developer and User Subscriptions](#)

## Audience

How you use IAM differs, depending on the work you do in Amazon Q.

**Service user** – If you use the Amazon Q service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Amazon Q features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator.

**Service administrator** – If you're in charge of Amazon Q resources at your company, you probably have full access to Amazon Q. It's your job to determine which Amazon Q features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand

the basic concepts of IAM. To learn more about how your company can use IAM with Amazon Q, see [How Amazon Q works with IAM](#).

**IAM administrator** – If you’re an IAM administrator, you might want to learn details about how you can write policies to manage access to Amazon Q. If you’re an IAM administrator, consider learning the details about how you can write policies to manage IAM user access to services. For information that’s specific to Amazon Q, see [AWS Regions managed policies for Amazon Q](#).

## Authenticating with identities

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

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company’s single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see [How to sign in to your AWS account](#) in the *AWS Sign-In User Guide*.

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see [Multi-factor authentication](#) in the *AWS IAM Identity Center User Guide* and [Using multi-factor authentication \(MFA\) in AWS](#) in the *IAM User Guide*.

### AWS account root user

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don’t use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see [Tasks that require root user credentials](#) in the *IAM User Guide*.

## Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A federated identity is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see [What is IAM Identity Center?](#) in the *AWS IAM Identity Center User Guide*.

## IAM users and groups

An [IAM user](#) is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see [Rotate access keys regularly for use cases that require long-term credentials](#) in the *IAM User Guide*.

An [IAM group](#) is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. For more information, see [When to create an IAM user \(instead of a role\)](#) in the *IAM User Guide*.

## IAM roles

An [IAM role](#) is an identity within your AWS account that has specific permissions. An IAM role is similar to an IAM user but is not associated with a specific person. You can temporarily assume an

IAM role in the AWS Management Console by [switching roles](#). You can assume a role by calling an AWS Command Line Interface (AWS CLI) or AWS API operation or by using a custom URL. For more information about methods for using roles, see [Using IAM roles](#) in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see [Creating a role for a third-party Identity Provider](#) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see [Permission sets](#) in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). For more information about the difference between roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see [Creating a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an Amazon EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the Amazon EC2 instance. To assign an IAM role to an Amazon EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the Amazon EC2 instance to get temporary credentials. For more information, see [Using an IAM role to grant permissions to applications running on Amazon EC2 instances](#) in the *IAM User Guide*.

For more information about whether to use IAM roles, see [When to create an IAM role \(instead of a user\)](#) in the *IAM User Guide*.

## Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see [Overview of JSON policies](#) in the *IAM User Guide*.

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

Every IAM entity (user or role) starts with no permissions. By default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

## Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, role, or group. These policies control what actions users and roles can perform,

on which resources, and under what conditions. For more information about how to create an identity-based policy, see [Creating IAM policies](#) in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. For more information about how to choose between a managed policy or an inline policy, see [Choosing between managed policies and inline policies](#) in the *IAM User Guide*.

## Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource such as an Amazon S3 bucket. Service administrators can use these policies to define what actions a specified principal (account member, user, or role) can perform on that resource and under what conditions. Resource-based policies are inline policies. There are no managed resource-based policies.

## Access control lists (ACLs)

Access control lists (ACLs) are a type of policy that controls which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format. Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. For more information about ACLs, see [Access Control List \(ACL\) overview](#) in the *Amazon S3 User Guide*.

## Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the Principal field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see [Permissions boundaries for IAM entities](#) in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a

service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply SCPs to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see [How SCPs work](#) in the *AWS Organizations User Guide*.

- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see [Session policies](#) in the *IAM User Guide*.

## Multiple policy types

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

## How Amazon Q Developer works with IAM

Before you use IAM to manage access to Amazon Q Developer, learn what IAM features are available to use with Amazon Q Developer.

### IAM features you can use with Amazon Q Developer

IAM feature	Amazon Q support
<a href="#">Identity-based policies</a>	Yes
<a href="#">Resource-based policies</a>	No
<a href="#">Policy actions</a>	Yes
<a href="#">Policy resources</a>	No
<a href="#">Policy condition keys</a>	No
<a href="#">ACLs</a>	No

IAM feature	Amazon Q support
<a href="#">ABAC (tags in policies)</a>	No
<a href="#">Temporary credentials</a>	Yes
<a href="#">Principal permissions</a>	Yes
<a href="#">Service roles</a>	No
<a href="#">Service-linked roles</a>	Yes

To get a high-level view of how Amazon Q and other AWS services work with most IAM features, see [AWS services that work with IAM](#) in the *IAM User Guide*.

## Identity-based policies for Amazon Q

**Supports identity-based policies:** Yes

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

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see [IAM JSON policy elements reference](#) in the *IAM User Guide*.

## Identity-based policy examples for Amazon Q

To view examples of Amazon Q Developer identity-based policies, see [Identity-based policy examples for Amazon Q Developer](#).

## Resource-based policies within Amazon Q

**Supports resource-based policies:** No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must [specify a principal](#) in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see [Cross account resource access in IAM](#) in the *IAM User Guide*.

## Policy actions for Amazon Q

**Supports policy actions:** Yes

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

The Action element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Amazon Q actions, see [Manage access to Amazon Q Developer with policies](#).

Policy actions in Amazon Q use the following prefix before the action:

q

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

```
"Action": [  
    "q:action1",  
    "q:action2"  
]
```

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

```
"Action": "q:Get*"
```

To view examples of Amazon Q Developer identity-based policies, see [Identity-based policy examples for Amazon Q Developer](#).

## Policy resources for Amazon Q

**Supports policy resources:** No

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

The Resource JSON policy element specifies the object or objects to which the action applies. Statements must include either a Resource or a NotResource element. As a best practice, specify a resource using its [Amazon Resource Name \(ARN\)](#). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (\*) to indicate that the statement applies to all resources.

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

To view examples of Amazon Q Developer identity-based policies, see [Identity-based policy examples for Amazon Q Developer](#).

## Policy condition keys for Amazon Q

**Supports service-specific policy condition keys:** No

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

The Condition element (or Condition *block*) lets you specify conditions in which a statement is in effect. The Condition element is optional. You can create conditional expressions that use [condition operators](#), such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple Condition elements in a statement, or multiple keys in a single Condition element, AWS evaluates them using a logical AND operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical OR operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see [IAM policy elements: variables and tags](#) in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see [AWS global condition context keys](#) in the *IAM User Guide*.

To view examples of Amazon Q Developer identity-based policies, see [Identity-based policy examples for Amazon Q Developer](#).

## ACLs in Amazon Q

**Supports ACLs:** No

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

## ABAC with Amazon Q

**Supports ABAC (tags in policies):** No

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the [condition element](#) of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see [Define permissions with ABAC authorization](#) in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see [Use attribute-based access control \(ABAC\)](#) in the *IAM User Guide*.

## Using temporary credentials with Amazon Q

**Supports temporary credentials:** Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see [AWS services that work with IAM](#) in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see [Switch from a user to an IAM role \(console\)](#) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see [Temporary security credentials in IAM](#).

## Cross-service principal permissions for Amazon Q

**Supports forward access sessions (FAS):** Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal.

When you use some services, you might perform an action that then initiates another action in a

different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see [Forward access sessions](#).

## Service roles for Amazon Q

**Supports service roles:** No

A service role is an [IAM role](#) that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see [Create a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.

 **Warning**

Changing the permissions for a service role might break Amazon Q functionality. Edit service roles only when Amazon Q provides guidance to do so.

## Service-linked roles for Amazon Q

**Supports service-linked roles:** Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing Amazon Q service-linked roles, see [Using service-linked roles for Amazon Q Developer and User Subscriptions](#).

## Manage access to Amazon Q Developer with policies

 **Note**

The information on this page pertains to accessing Amazon Q Developer. For information about managing access to Amazon Q Business, see [Identity-based policy examples for Amazon Q Business](#) in the *Amazon Q Business User Guide*.

The policies and examples in this topic are specific to Amazon Q in the AWS Management Console, AWS Console Mobile Application, AWS website, AWS Documentation, and AWS Chatbot. Other services integrated with Amazon Q might require different policies or settings. End users of Amazon Q in third-party IDEs are not required to use IAM policies. For more information, see the documentation for the service that contains an Amazon Q feature or integration.

By default, users and roles don't have permission to use Amazon Q. IAM administrators can manage access to Amazon Q Developer and its features by granting permissions to IAM identities.

The quickest way for an administrator to grant access to users is through an AWS managed policy. The `AmazonQFullAccess` policy can be attached to IAM identities to grant full access to Amazon Q Developer and its features. For more information about this policy, see [AWS managed policies for Amazon Q Developer](#).

To manage specific actions that IAM identities can perform with Amazon Q Developer, administrators can create custom policies that define what permissions a user, group, or role has. You can also use service control policies (SCPs) to control what Amazon Q features are available in your organization.

For a list of all Amazon Q permissions you can control with policies, see the [see the Amazon Q Developer permissions reference](#).

## Topics

- [Policy best practices](#)
- [Assign permissions](#)
- [Manage access with service control policies \(SCPs\)](#)
- [Data perimeters for Amazon Q resources](#)
- [Identity-based policy examples for Amazon Q Developer](#)

## Policy best practices

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

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

For more information about best practices in IAM, see [Security best practices in IAM](#) in the *IAM User Guide*.

## Assign permissions

To provide access, add permissions to your users, groups, or roles:

- Users and groups in AWS IAM Identity Center:

Create a permission set. Follow the instructions in [Create a permission set](#) in the *AWS IAM Identity Center User Guide*.

- Users managed in IAM through an identity provider:

Create a role for identity federation. Follow the instructions in [Create a role for a third-party identity provider \(federation\)](#) in the *IAM User Guide*.

- IAM users:

- Create a role that your user can assume. Follow the instructions in [Create a role for an IAM user](#) in the *IAM User Guide*.
- (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in [Adding permissions to a user \(console\)](#) in the *IAM User Guide*.

## Manage access with service control policies (SCPs)

Service control policies (SCPs) are a type of organization policy that you can use to manage permissions in your organization. You can control what Amazon Q Developer features are available in your organization by creating an SCP that specifies permissions for some or all Amazon Q actions.

For more information about using SCPs to control access in your organization, see [Creating, updating, and deleting service control policies](#) and [Attaching and detaching service control policies](#) in the *AWS Organizations User Guide*.

The following is an example of an SCP that denies access to Amazon Q. This policy restricts access to Amazon Q chat, console error troubleshooting, and network troubleshooting.

### Note

Denying access to Amazon Q will not disable the Amazon Q icon or chat panel in the AWS console, AWS website, AWS documentation pages, or AWS Console Mobile Application.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "DenyAmazonQFullAccess",  
      "Effect": "Deny",  
      "Action": "lambda:InvokeFunction",  
      "Resource": "arn:aws:lambda:us-east-1:123456789012:function:AmazonQ"  
    }  
  ]  
}
```

```
        "Effect": "Deny",
        "Action": [
            "q:*"
        ],
        "Resource": "*"
    }
]
}
```

## Data parameters for Amazon Q resources

For some features, Amazon Q uploads artifacts to AWS service-owned Amazon S3 buckets. If you are using data parameters to control access to Amazon S3 in your environment, you might need to explicitly allow access to these buckets to use the corresponding Amazon Q features.

The following table lists the ARN and URL of each of the Amazon S3 buckets that Amazon Q requires access to, and the features that use each bucket. You can use the bucket ARN or bucket URL to allowlist these buckets, depending on how you control access to Amazon S3.

Amazon S3 bucket ARN	Amazon S3 bucket URL	Description
<code>arn:aws:s3:::amazonq-code-scan-us-east-1-29121b44f7b</code>	<code>https://amazonq-code-scan-us-east-1-29121b44f7b.s3.amazonaws.com/</code>	An Amazon S3 bucket used to upload artifacts for <a href="#">Amazon Q code scans</a>
<code>arn:aws:s3:::amazonq-code-transformation-us-east-1-c6160f047e0</code>	<code>https://amazonq-code-transformation-us-east-1-c6160f047e0.s3.amazonaws.com/</code>	An Amazon S3 bucket used to upload artifacts for the <a href="#">Amazon Q Developer Agent for code transformation</a>
<code>arn:aws:s3:::amazonq-feature-development-us-east-1-a5b980054c6</code>	<code>https://amazonq-feature-development-us-east-1-a5b980054c6.s3.amazonaws.com/</code>	An Amazon S3 bucket used to upload

Amazon S3 bucket ARN	Amazon S3 bucket URL	Description
		artifacts for the <a href="#">Amazon Q Developer Agent for software development</a>

## Identity-based policy examples for Amazon Q Developer

The following example IAM policies control permissions for various Amazon Q Developer actions. Use them to allow or deny Amazon Q Developer access for your users, roles, or groups.

### Note

The following example policies grant permissions for features of Amazon Q Developer, but users might need additional permissions to access Amazon Q with an Amazon Q Developer Pro subscription. For more information, see [Allow users to access Amazon Q with an Amazon Q Developer Pro subscription](#).

You can use these policies as written, or you can add permissions for the individual Amazon Q features you want to use. For more information about configuring IAM permissions with Amazon Q, see [Manage access to Amazon Q Developer with policies](#).

For a list of all Amazon Q permissions you can control with policies, see the [Amazon Q Developer permissions reference](#).

### Topics

- [Administrator permissions](#)
- [User permissions](#)

### Administrator permissions

The following policies allow Amazon Q Developer administrators to perform administrative tasks in the Amazon Q subscription management console and Amazon Q Developer Pro console.

For policies that enable the use of Amazon Q Developer features, see [User permissions](#).

## Allow administrators to use the Amazon Q subscription console

The following example policy grants permissions for a user to create and manage subscriptions in the Amazon Q subscription management console. The Amazon Q subscription management console is where you configure Amazon Q's integration with AWS IAM Identity Center and AWS Organizations, choose which Amazon Q package to subscribe to, and attach users and groups to subscriptions.

To configure Amazon Q Developer Pro after subscribing users, someone in your enterprise will also need access to the Amazon Q Developer Pro console. For more information, see [Allow administrators to use the Amazon Q Developer console](#).

## Note

The codewhisisperer 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](#).

```
    "sso:PutApplicationAuthenticationMethod",
    "sso:PutApplicationAssignmentConfiguration",
    "sso:PutApplicationGrant",
    "sso:PutApplicationAccessScope",
    "sso:DescribeApplication",
    "sso:DeleteApplication",
    "sso:GetSSOStatus",
    "sso>CreateApplicationAssignment",
    "sso:DeleteApplicationAssignment"
],
"Resource": [
    "*"
]
},
{
    "Effect": "Allow",
    "Action": [
        "sso-directory:DescribeUsers",
        "sso-directory:DescribeGroups",
        "sso-directory:SearchGroups",
        "sso-directory:SearchUsers",
        "sso-directory:DescribeGroup",
        "sso-directory:DescribeUser",
        "sso-directory:DescribeDirectory"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "signin>ListTrustedIdentityPropagationApplicationsForConsole",
        "signin>CreateTrustedIdentityPropagationApplicationForConsole"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "codewhisperer>ListProfiles",
        "codewhisperer>CreateProfile",
        "codewhisperer>DeleteProfile"
    ]
}
```

```
    "codewhisperer>DeleteProfile"
],
"Resource": [
  "*"
]
},
{
  "Effect": "Allow",
  "Action": [
    "user-subscriptions>ListClaims",
    "user-subscriptions>ListUserSubscriptions",
    "user-subscriptions>CreateClaim",
    "user-subscriptions>DeleteClaim",
    "user-subscriptions>UpdateClaim"
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "q>CreateAssignment",
    "q>DeleteAssignment"
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "iam>CreateServiceLinkedRole"
  ],
  "Resource": [
    "arn:aws:iam::*:role/aws-service-role/user-subscriptions.amazonaws.com/
AWSServiceRoleForUserSubscriptions"
  ]
}
]
```

## Allow administrators to use the Amazon Q Developer console

The following example policy grants permissions for a user to access the Amazon Q Developer console. On the Amazon Q Developer console, administrators can configure various aspects of Amazon Q Developer and its features, including code references, customizations, and chat plugins. This policy also includes permissions to create and configure customer managed KMS keys.

To configure Amazon Q Developer Pro subscriptions, someone in your enterprise will also need access to the Amazon Q subscription management console. For more information, see [Allow administrators to use the Amazon Q subscription console](#).

### Note

If you're using customizations, then your Amazon Q Developer Pro administrator will require additional permissions.

- For permissions needed for customizations, see [Prerequisites for customizations](#).
- For permissions needed for plugins, see [Allow administrators to configure plugins](#).

You will need one of two policies to use the Amazon Q Developer console. The policy you need depends on if you're setting up Amazon Q Developer for the first time or if you're configuring a legacy Amazon CodeWhisperer profile.

### 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](#).

For new administrators of Amazon Q Developer, use the following policy:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "sso-directory:GetUserPoolInfo"  
      ],  
      "Resource": "  
        arn:aws:sso-directory:  
        <region>:  
        <account>:  
        <group>  
      "
```

```
"Resource": [
    "*"
],
},
{
    "Effect": "Allow",
    "Action": [
        "iam>ListRoles"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "sso>DescribeRegisteredRegions",
        "sso>GetSSOStatus"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "organizations>DescribeAccount",
        "organizations>DescribeOrganization"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kms>ListAliases",
        "kms>CreateGrant",
        "kms>Encrypt",
        "kms>Decrypt",
        "kms>GenerateDataKey*",
        "kms>RetireGrant",
        "kms>DescribeKey"
    ],
    "Resource": [
        "*"
    ]
},
```

```
"Resource": [
    "*"
],
},
{
    "Effect": "Allow",
    "Action": [
        "codeguru-security:UpdateAccountConfiguration"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole"
    ],
    "Resource": [
        "arn:aws:iam::*:role/aws-service-role/q.amazonaws.com/
AWSServiceRoleForAmazonQDeveloper"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "codewhisperer:UpdateProfile",
        "codewhisperer>ListProfiles",
        "codewhisperer:TagResource",
        "codewhisperer:UnTagResource",
        "codewhisperer>ListTagsForResource",
        "codewhisperer>CreateProfile"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "q>ListDashboardMetrics",
        "cloudwatch:GetMetricData",
        "cloudwatch>ListMetrics"
    ],
    "Resource": [
        "*"
    ]
}
```

```
    "Resource": [
        "*"
    ]
}
]
}
```

For legacy Amazon CodeWhisperer profiles, the following policy will enable an IAM principal to administer a CodeWhisperer application.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sso-directory:SearchUsers",
                "sso-directory:SearchGroups",
                "sso-directory:GetUserPoolInfo",
                "sso-directory:DescribeDirectory",
                "sso-directory>ListMembersInGroup"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam>ListRoles"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "pricing:GetProducts"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

```
},
{
  "Effect": "Allow",
  "Action": [
    "sso:AssociateProfile",
    "sso:DisassociateProfile",
    "sso:GetProfile",
    "sso>ListProfiles",
    "sso>ListApplicationInstances",
    "sso:GetApplicationInstance",
    "sso>CreateManagedApplicationInstance",
    "sso:GetManagedApplicationInstance",
    "sso>ListProfileAssociations",
    "sso:GetSharedSsoConfiguration",
    "sso>ListDirectoryAssociations",
    "sso:DescribeRegisteredRegions",
    "sso:GetSsoConfiguration",
    "sso:GetSSOStatus"
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "identitystore>ListUsers",
    "identitystore>ListGroups"
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "organizations>DescribeAccount",
    "organizations>DescribeOrganization"
  ],
  "Resource": [
    "*"
  ]
},
{
```

```
"Effect": "Allow",
"Action": [
    "kms>ListAliases",
    "kms>CreateGrant",
    "kms>Encrypt",
    "kms>Decrypt",
    "kms>GenerateDataKey*",
    "kms>RetireGrant",
    "kms>DescribeKey"
],
"Resource": [
    "*"
],
},
{
"Effect": "Allow",
"Action": [
    "codeguru-security:UpdateAccountConfiguration"
],
"Resource": [
    "*"
],
},
{
"Effect": "Allow",
"Action": [
    "iam>CreateServiceLinkedRole"
],
"Resource": [
    "arn:aws:iam::*:role/aws-service-role/q.amazonaws.com/
AWSServiceRoleForAmazonQDeveloper"
]
},
{
"Effect": "Allow",
"Action": [
    "codewhisperer:UpdateProfile",
    "codewhisperer>ListProfiles",
    "codewhisperer:TagResource",
    "codewhisperer:UnTagResource",
    "codewhisperer>ListTagsForResource",
    "codewhisperer>CreateProfile"
],
"Resource": [
```

```
    "/*"
]
},
{
  "Effect": "Allow",
  "Action": [
    "q>ListDashboardMetrics",
    "cloudwatch:GetMetricData",
    "cloudwatch>ListMetrics"
  ],
  "Resource": [
    "/*"
  ]
}
]
```

## Allow administrators to create customizations

The following policy grants administrators permission to create and manage customizations in Amazon Q Developer.

To configure customizations in the Amazon Q Developer Pro console, your Amazon Q Developer administrator will require access to the Amazon Q Developer Pro console. For more information, see [Allow administrators to use the Amazon Q Developer console](#).

### 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](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sso-directory:DescribeUsers"
      ],
      "Resource": [
        "/*"
      ]
    }
]
```

```
},
{
    "Effect": "Allow",
    "Action": [
        "codewhisperer>CreateCustomization",
        "codewhisperer>DeleteCustomization",
        "codewhisperer>ListCustomizations",
        "codewhisperer>UpdateCustomization",
        "codewhisperer>GetCustomization",
        "codewhisperer>ListCustomizationPermissions",
        "codewhisperer>AssociateCustomizationPermission",
        "codewhisperer>DisassociateCustomizationPermission"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "codeconnections>ListConnections",
        "codeconnections>ListOwners",
        "codeconnections>ListRepositories",
        "codeconnections>GetConnection"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": "codeconnections:UseConnection",
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "codeconnections:ProviderAction": [
                "GitPull",
                "ListRepositories",
                "ListOwners"
            ]
        }
    }
},
{
```

```
        "Effect": "Allow",
        "Action": [
            "s3:GetObject*",
            "s3:GetBucket*",
            "s3>ListBucket*"
        ],
        "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.

### Note

In order to access the Amazon Q Developer console, users also needs the permissions defined in [Allow administrators to use the Amazon Q Developer console](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "q>CreatePlugin",
                "q>GetPlugin",
                "q>DeletePlugin",
                "q>ListPlugins",
                "q>ListPluginProviders",
                "iam>CreateRole",
                "secretsmanager>CreateSecret"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
```

```
"Action": [
    "iam:PassRole"
],
"Resource": "*",
"Condition": {
    "StringEquals": {
        "iam:PassedToService": [
            "q.amazonaws.com"
        ]
    }
}
]
}
```

## User permissions

The following policies allow users to access features of Amazon Q Developer on AWS apps and websites.

For policies that enable administrative access to Amazon Q Developer, see [Administrator permissions](#).

### Allow users to access Amazon Q with an Amazon Q Developer Pro subscription

The following example policy grants permission to use Amazon Q with an Amazon Q Developer Pro subscription. Without these permissions, users can only access the Free tier of Amazon Q. To chat with Amazon Q or use other Amazon Q features, users need additional permissions, such as those granted by the example policies in this section.

```
{
    "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 Amazon Q access to customer managed keys

The following example policy grants users permissions to access features encrypted with a customer managed key by allowing Amazon Q access to the key. This policy is required to use Amazon Q if an administrator has set up 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:{region}:{account_id}:key/[[key_id]]"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "q.{region}.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

## Allow users to chat with Amazon Q

The following example policy grants permissions to chat with Amazon Q in the console.

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

## Allow users to use Amazon Q CLI with AWS CloudShell

The following example policy grants permissions to use Amazon Q CLI with AWS CloudShell.

### 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](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "codewhisperer:GenerateRecommendations",  
                "codewhisperer>ListCustomizations"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

```
},
{
  "Effect": "Allow",
  "Action": [
    "q:SendMessage"
  ],
  "Resource": "*"
}
]
```

## Allow users to diagnose console errors with Amazon Q

The following example policy grants permissions to diagnose console errors with Amazon Q.

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

## Allow users to generate code from CLI commands with Amazon Q

The following example policy grants permissions to generate code from recorded CLI commands with Amazon Q, which enables the use of the Console-to-Code feature.

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

```
{  
    "Sid": "AllowAmazonQConsoleToCode",  
    "Effect": "Allow",  
    "Action": "q:GenerateCodeFromCommands",  
    "Resource": "*"  
}  
]  
}
```

## Allow Amazon Q to perform actions on your behalf in chat

The following example policy grants permission to chat with Amazon Q, and allows Amazon Q to perform actions on your behalf. Amazon Q only has permission to perform 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:PassRequest"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

## Deny Amazon Q permission to perform specific actions on your behalf

The following example policy grants permission to chat with Amazon Q, and allows Amazon Q to perform any action on your behalf that your IAM identity has permission to perform, except for Amazon EC2 actions. This policy uses the [aws:CalledVia global condition key](#) to specify that Amazon EC2 actions are only denied when Amazon Q calls them.

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

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "q:StartConversation",
      "q:SendMessage",
      "q:GetConversation",
      "q>ListConversations",
      "q:PassRequest"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Deny",
    "Action": [
      "ec2:)"
    ],
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": ["q.amazonaws.com"]
      }
    }
  }
]
```

## Allow Amazon Q permission to perform specific actions on your behalf

The following example policy grants permission to chat with Amazon Q, and allows Amazon Q to perform any action on your behalf that your IAM identity has permission to perform, with the exception of Amazon EC2 actions. This policy grants your IAM identity permission to perform any Amazon EC2 action, but only allows Amazon Q to perform the ec2:describeInstances action. This policy uses the [aws:CalledVia global condition key](#) to specify that Amazon Q is only allowed to call ec2:describeInstances, and not any other Amazon EC2 actions.

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

```
        "q:SendMessage",
        "q:GetConversation",
        "q>ListConversations",
        "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 Amazon Q permission to perform actions on your behalf in specific regions

The following example policy grants permission to chat with Amazon Q, and allows Amazon Q to make calls to only the us-east-1 and us-west-2 Regions when performing actions on your behalf. Amazon Q can't make calls to any other Region. For more information on how to specify what Regions you can make calls to, see [aws:RequestedRegion](#) in the *AWS Identity and Access Management User Guide*.

{

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

## Deny Amazon Q permission to perform actions on your behalf

The following example policy prevents Amazon Q from performing actions on your behalf.

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

## Allow users to chat with plugins from one provider

The following example policy grants permission to chat with any Datadog plugin that an administrator configures, specified by the plugin ARN with a wildcard character (\*). If the plugin is deleted and re-configured, a user with these permissions will retain access to the newly configured plugin. To use this policy, replace the AWS account ID in the ARN with the ID of the account where your plugin is configured.

You can grant these permissions for other plugins by replacing Datadog with the name of another plugin, such as Wiz.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowAmazonQConversationAccess",  
            "Effect": "Allow",  
            "Action": [  
                "q:StartConversation",  
                "q:SendMessage",  
                "q:GetConversation",  
                "q>ListConversations"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "AllowPluginAccess",  
            "Action": [  
                "q:UsePlugin"  
            ],  
            "Resource": "arn:aws:q::123456789012:plugin/Datadog/*"  
        }  
    ]  
}
```

## Allow users to chat with a specific plugin

The following example policy grants permission to chat with a specific Wiz plugin, specified by the plugin ARN. If the plugin is deleted and re-configured, a user will not have access to the new plugin unless the plugin ARN is updated in this policy. To use this policy, replace the example ARN with the ARN of the plugin you want to allow access to.

You can grant these permissions for other plugins by replacing Wiz with the name of another plugin, such as Datadog.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowAmazonQConversationAccess",  
            "Effect": "Allow",  
            "Action": [  
                "q:StartConversation",  
                "q:SendMessage",  
                "q:GetConversation",  
                "q>ListConversations"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "AllowPluginAccess",  
            "Action": [  
                "q:UsePlugin"  
            ],  
            "Resource": "arn:aws:q::123456789012:plugin/Wiz/ABCDEFGHIJKL"  
        }  
    ]  
}
```

## Deny access to Amazon Q

The following example policy denies all permissions to use Amazon Q.

### Note

Denying access to Amazon Q will not disable the Amazon Q icon or chat panel in the AWS console, AWS website, AWS documentation pages, or AWS Console Mobile Application.

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

```
{  
    "Sid": "DenyAmazonQFullAccess",  
    "Effect": "Deny",  
    "Action": [  
        "q:*"  
    ],  
    "Resource": "*"  
}  
]  
}
```

## Allow users to view their permissions

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

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

```
        "iam>ListUsers"
    ],
    "Resource": "*"
}
]
```

## Amazon Q Developer permissions reference

Amazon Q Developer uses two types of APIs to provide the service:

- User and administrator permissions, which can be used in policies to control usage of Amazon Q
- Other APIs used to provide the service, which can't be used in policies to control usage of Amazon Q

This section provides information about the APIs used by Amazon Q Developer, and what they do.

### Topics

- [Amazon Q Developer permissions](#)
- [Amazon Q User Subscriptions permissions](#)
- [Other Amazon Q Developer APIs](#)

## Amazon Q Developer permissions

You can use the following permissions as a reference when you are setting up [Authenticating with identities in Amazon Q](#) and writing permissions policies that you can attach to an IAM identity (identity-based policies).

The following table shows the Amazon Q Developer permissions that you can allow or deny access to in policies.

### Important

To chat with Amazon Q, an IAM identity needs permissions for the following actions:

- StartConversation
- SendMessage

- [GetConversation \(console only\)](#)
- [ListConversations \(console only\)](#)

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 Amazon Q.

### 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](#).

## Using q:PassRequest

`q:PassRequest` is an Amazon Q permission that allows Amazon Q to call AWS APIs on your behalf. When you add the `q:PassRequest` permission to an IAM identity, Amazon Q 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, Amazon Q is able to call the `ListAllMyBuckets` API when a user assuming that IAM role asks Amazon Q 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 Amazon Q from performing a specific action, or only permit Amazon Q to perform a subset of actions for a service. You can also specify what regions Amazon Q can make calls to when performing actions on your behalf.

For examples of IAM policies that control the use of `q:PassRequest`, see the following identity-based policy examples:

- [Allow Amazon Q to perform actions on your behalf in chat](#)
- [Deny Amazon Q permission to perform specific actions on your behalf](#)
- [Allow Amazon Q permission to perform specific actions on your behalf](#)
- [Allow Amazon Q permission to perform actions on your behalf in specific regions](#)
- [Deny Amazon Q permission to perform actions on your behalf](#)

## Amazon Q User Subscriptions permissions

Amazon Q Developer administrators must have the following permissions to create and manage subscriptions for users and groups in their organization.

The following terminology is useful in understanding what subscriptions permissions do:

### User

An individual user, represented within AWS IAM Identity Center by a unique user ID.

### Group

A collection of users, represented within AWS IAM Identity Center by a unique group ID.

### Subscription

A subscription is tied to a single Identity Center user, and entitles them to use Amazon Q features. A subscription does not authorize a user to use Amazon Q features. For example, if Adam is subscribed to Amazon Q Developer Pro, they are entitled to use Amazon Q Developer features, but they don't have access to those features until their administrator grants them the needed permissions.

## Other Amazon Q Developer APIs

The following table shows the APIs that are used by features of Amazon Q in the IDE. These APIs aren't used to control access to features of Amazon Q, but they will appear in AWS CloudTrail logs in management accounts when users access the associated feature.

### 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](#).

## AWS managed policies for Amazon Q Developer

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

The quickest way for an administrator to grant access to users is through an AWS managed policy. The following AWS managed policies for Amazon Q Developer can be attached to IAM identities:

- `AmazonQFullAccess` provides full access to enable interactions with Amazon Q Developer, including administrator access.
- `AmazonQDeveloperAccess` provides full access to enable interactions with Amazon Q Developer, without administrator access.

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

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

For more information, see [AWS managed policies](#) in the *IAM User Guide*.

## AmazonQFullAccess

The `AmazonQFullAccess` managed policy provides administrator access to allow users in your organization to access Amazon Q Developer. It also provides full access to enable interactions with Amazon Q Developer, including logging in with IAM Identity Center to access Amazon Q through an Amazon Q Developer Pro subscription.

### Note

To enable full access to complete administrative tasks in the Amazon Q subscription management console and Amazon Q Developer Pro console, additional permissions are needed. For more information, see [Administrator permissions](#).

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

```
"Sid": "AllowAmazonQFullAccess",
"Effect": "Allow",
>Action": [
    "q:StartConversation",
    "q:SendMessage",
    "q:GetConversation",
    "q>ListConversations",
    "q:PassRequest",
    "q:StartTroubleshootingAnalysis",
    "q:GetTroubleshootingResults",
    "q:StartTroubleshootingResolutionExplanation",
    "q:UpdateTroubleshootingCommandResult",
    "q:GetIdentityMetadata",
    "q>CreateAssignment",
    "q>DeleteAssignment",
    "q:GenerateCodeFromCommands",
    "q>CreatePlugin",
    "q:GetPlugin",
    "q:DeletePlugin",
    "q>ListPlugins",
    "q>ListPluginProviders",
    "q:UsePlugin",
    "q:TagResource",
    "q:UntagResource",
    "q>ListTagsForResource"
],
"Resource": "*"
},
{
    "Sid": "AllowCloudControlReadAccess",
    "Effect": "Allow",
    "Action": [
        "cloudformation:GetResource",
        "cloudformation>ListResources"
    ],
    "Resource": "*"
},
{
    "Sid": "AllowSetTrustedIdentity",
    "Effect": "Allow",
    "Action": [
        "sts:SetContext"
    ],
    "Resource": "arn:aws:sts::*:self"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "q.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

## AmazonQDeveloperAccess

The AmazonQDeveloperAccess managed policy provides full access to enable interactions with Amazon Q Developer, without administrator access. It includes access to log in with IAM Identity Center to access Amazon Q through an Amazon Q Developer Pro subscription.

To use some features of Amazon Q, you might need additional permissions. See the topic for the feature you want to use for information on permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAmazonQDeveloperAccess",
            "Effect": "Allow",
            "Action": [
                "q:StartConversation",
                "q:SendMessage",
                "q:GetConversation",
                "q>ListConversations",
                "q:PassRequest",
                "q:StartTroubleshootingAnalysis",
                "q:StartTroubleshootingResolutionExplanation",
                "q:GetTroubleshootingResults",
                "q:UpdateTroubleshootingCommandResult",
                "q:ListTroubleshootingCommands"
            ]
        }
    ]
}
```

```
        "q:GetIdentityMetaData",
        "q:GenerateCodeFromCommands",
        "q:UsePlugin"

    ],
    "Resource": "*"
},
{
    "Sid": "AllowCloudControlReadAccess",
    "Effect": "Allow",
    "Action": [
        "cloudformation:GetResource",
        "cloudformation>ListResources"
    ],
    "Resource": "*"
},
{
    "Sid": "AllowSetTrustedIdentity",
    "Effect": "Allow",
    "Action": [
        "sts:SetContext"
    ],
    "Resource": "arn:aws:sts::*:self"
}
]
}
```

## AWSServiceRoleForAmazonQDeveloperPolicy

This AWS managed policy grants permissions commonly needed to use Amazon Q Developer. The policy is added to the AWSServiceRoleForAmazonQDeveloper service linked role that is created when you onboard to Amazon Q.

You can't attach AWSServiceRoleForAmazonQDeveloperPolicy to your IAM entities. This policy is attached to [a service-linked role](#) that allows Amazon Q to perform actions on your behalf. For more information, see [Using service-linked roles for Amazon Q Developer and User Subscriptions](#).

This policy grants **administrator** permissions that allows metrics to be published for Billing / Usage.

### Permissions details

This policy includes the following permissions.

- cloudwatch – Allows principals to publish usage metrics to CloudWatch for Billing / Usage. This is required so that you can track your usage of Amazon Q in CloudWatch.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "cloudwatch:PutMetricData"  
            ],  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "cloudwatch:namespace": [  
                        "AWS/Q"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

To view this policy in the context of other AWS managed policies, see [AmazonQDeveloperPolicy](#).

## AWSServiceRoleForUserSubscriptionPolicy

This AWS managed policy grants permissions commonly needed to use Amazon Q Developer. The policy is added to the AWSServiceRoleForUserSubscriptions service-linked role that is created when you create Amazon Q subscriptions.

You can't attach AWSServiceRoleForUserSubscriptionPolicy to your IAM entities. This policy is attached to [a service-linked role](#) that allows Amazon Q to perform actions on your behalf. For more information, see [Using service-linked roles for Amazon Q Developer and User Subscriptions](#).

This policy provides access for Amazon Q Subscriptions to your Identity Center resources to automatically update your subscriptions.

## Permissions details

This policy includes the following permissions.

- **identitystore** – Allows principals to track Identity Center directory changes so that subscriptions can be automatically updated.
- **organizations** – Allows principals to track AWS Organizations changes so that subscriptions can be automatically updated.
- **sso** – Allows principals to track Identity Center instance changes so that subscriptions can be automatically updated.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "identitystore:DescribeGroup",  
                "identitystore:DescribeUser",  
                "identitystore:IsMemberInGroups",  
                "identitystore>ListGroupMemberships",  
                "organizations:DescribeOrganization",  
                "sso:DescribeApplication",  
                "sso:DescribeInstance",  
                "sso>ListInstances"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

To view this policy in the context of other AWS managed policies, see [AWS Service Role for User Subscription Policy](#).

## Policy updates

View details about updates to AWS managed policies for Amazon Q Developer since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the [Document history for Amazon Q Developer User Guide](#) page.

Change	Description	Date
<a href="#"><u>AmazonQDeveloperAccess</u></a> - Updated policy	Additional permissions have been added to enable the use of Amazon Q Developer plugins.	November 13, 2024
<a href="#"><u>AmazonQFullAccess</u></a> - Updated policy	Additional permissions have been added to configure and use Amazon Q Developer plugins and to create and manage tags for Amazon Q Developer resources.	November 13, 2024
<a href="#"><u>AmazonQDeveloperAccess</u></a> - Updated policy	Additional permissions have been added to enable code generation from CLI commands with Amazon Q.	October 28, 2024
<a href="#"><u>AmazonQFullAccess</u></a> - Updated policy	Additional permissions have been added to enable code generation from CLI commands with Amazon Q.	October 28, 2024
<a href="#"><u>AmazonQFullAccess</u></a> - Updated policy	Additional permissions have been added to enable Amazon Q to access downstream resources.	July 9, 2024
<a href="#"><u>AmazonQDeveloperAccess</u></a> - New policy	Provides full access to enable interactions with Amazon Q Developer, without administrator access.	July 9, 2024
<a href="#"><u>AmazonQFullAccess</u></a> - Updated policy	Additional permissions have been added to enable subscriptions checks for Amazon Q Developer.	April 30, 2024
<a href="#"><u>AWSServiceRoleForUserSubscr</u></a>	Allows Amazon Q Subscriptions to automatically update subscriptions from changes in AWS IAM Identity Center, AWS	April 30, 2024

Change	Description	Date
<a href="#">optionPolicy</a> - New policy	IAM Identity Center directory and AWS Organizations on your behalf.	
<a href="#">AWSServiceRoleForAmazonQDeveloperPolicy</a> - New policy	Allows Amazon Q to call Amazon CloudWatch and Amazon CodeGuru on your behalf.	April 30, 2024
<a href="#">AmazonQFullAccess</a> - New policy	Provides full access to enable interactions with Amazon Q Developer.	November 28, 2023
Amazon Q Developer started tracking changes	Amazon Q Developer started tracking changes to AWS managed policies.	November 28, 2023

## Using service-linked roles for Amazon Q Developer and User Subscriptions

Amazon Q Developer uses AWS Identity and Access Management (IAM) [service-linked roles](#). A service-linked role is a unique type of IAM role that is linked directly to Amazon Q Developer. Service-linked roles are predefined by Amazon Q Developer and include all the permissions that the service requires to call other AWS services on your behalf.

### Topics

- [Using service-linked roles for Amazon Q Developer](#)
- [Using service-linked-roles for User Subscriptions](#)

## Using service-linked roles for Amazon Q Developer

Amazon Q Developer uses AWS Identity and Access Management (IAM) [service-linked roles](#). A service-linked role is a unique type of IAM role that is linked directly to Amazon Q Developer.

Service-linked roles are predefined by Amazon Q Developer and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up Amazon Q Developer easier because you don't have to manually add the necessary permissions. Amazon Q Developer defines the permissions of its service-linked roles, and unless defined otherwise, only Amazon Q Developer can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your Amazon Q Developer resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see [AWS services that work with IAM](#) and look for the services that have **Yes** in the **Service-linked roles** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Learn about [AWS managed policies for Amazon Q Developer](#).

## Service-linked role permissions for Amazon Q Developer

Amazon Q Developer uses the service-linked role named **AWSServiceRoleForAmazonQDeveloper** – This role grants permissions to Amazon Q to access data in your account to calculate billing, provides access to create and access security reports in Amazon CodeGuru, and emit data to CloudWatch.

The AWSServiceRoleForAmazonQDeveloper service-linked role trusts the following services to assume the role:

- q.amazonaws.com

The role permissions policy named AWSServiceRoleForAmazonQDeveloperPolicy allows Amazon Q Developer to complete the following actions on the specified resources:

- Action: cloudwatch:PutMetricData on AWS/Q CloudWatch namespace

You must configure permissions to allow your users, groups, or roles to create, edit, or delete a service-linked role. For more information, see [Service-linked role permissions](#) in the *IAM User Guide*.

## Creating a service-linked role for Amazon Q Developer

You don't need to manually create a service-linked role. When you create a profile for Amazon Q in the AWS Management Console, Amazon Q Developer creates the service-linked role for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you update the settings, Amazon Q creates the service-linked role for you again.

You can also use the IAM console or AWS CLI to create a service-linked role with the `q.amazonaws.com` service name. For more information, see [Creating a service-linked role](#) in the *IAM User Guide*. If you delete this service-linked role, you can use this same process to create the role again.

## Editing a service-linked role for Amazon Q Developer

Amazon Q Developer does not allow you to edit the `AWSServiceRoleForAmazonQDeveloper` service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see [Editing a service-linked role](#) in the *IAM User Guide*.

## Deleting a service-linked role for Amazon Q Developer

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

### Note

If the Amazon Q Developer service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

## To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the `AWSServiceRoleForAmazonQDeveloper` service-linked role. For more information, see [Deleting a service-linked role](#) in the *IAM User Guide*.

## Supported Regions for Amazon Q Developer service-linked roles

Amazon Q Developer does not support using service-linked roles in every Region where the service is available. You can use the AWSServiceRoleForAmazonQDeveloper role in the following Regions. For more information, see [AWS Regions and endpoints](#).

Region name	Region identity	Support in Amazon Q Developer
US East (N. Virginia)	us-east-1	Yes
US East (Ohio)	us-east-2	No
US West (N. California)	us-west-1	No
US West (Oregon)	us-west-2	No
Africa (Cape Town)	af-south-1	No
Asia Pacific (Hong Kong)	ap-east-1	No
Asia Pacific (Jakarta)	ap-southeast-3	No
Asia Pacific (Mumbai)	ap-south-1	No
Asia Pacific (Osaka)	ap-northeast-3	No
Asia Pacific (Seoul)	ap-northeast-2	No
Asia Pacific (Singapore)	ap-southeast-1	No
Asia Pacific (Sydney)	ap-southeast-2	No
Asia Pacific (Tokyo)	ap-northeast-1	No
Canada (Central)	ca-central-1	No
Europe (Frankfurt)	eu-central-1	No
Europe (Ireland)	eu-west-1	No
Europe (London)	eu-west-2	No

Region name	Region identity	Support in Amazon Q Developer
Europe (Milan)	eu-south-1	No
Europe (Paris)	eu-west-3	No
Europe (Stockholm)	eu-north-1	No
Middle East (Bahrain)	me-south-1	No
Middle East (UAE)	me-central-1	No
South America (São Paulo)	sa-east-1	No
AWS GovCloud (US-East)	us-gov-east-1	No
AWS GovCloud (US-West)	us-gov-west-1	No

## Using service-linked-roles for User Subscriptions

User Subscriptions uses AWS Identity and Access Management (IAM) [service-linked roles](#). A service-linked role is a unique type of IAM role that is linked directly to User Subscriptions. Service-linked roles are predefined by User Subscriptions and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up User Subscriptions easier because you don't have to manually add the necessary permissions. User Subscriptions defines the permissions of its service-linked roles, and unless defined otherwise, only User Subscriptions can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your User Subscriptions because you can't inadvertently remove permissions required by the resources.

For information about other services that support service-linked roles, see [AWS services that work with IAM](#) and look for the services that have **Yes** in the **Service-linked roles** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

## Service-linked role permissions for User Subscriptions

User Subscriptions uses the service-linked role named **AWSServiceRoleForUserSubscriptions**

- Provides access for User Subscriptions to your IAM Identity Center resources to automatically update your subscriptions.

The AWSServiceRoleForUserSubscriptions service-linked role trusts the following services to assume the role:

- `user-subscriptions.amazonaws.com`

The role permissions policy named AWSServiceRoleForUserSubscriptionPolicy allows User Subscriptions to complete the following actions on the specified resources:

- Action: `identitystore:DescribeGroup` on \*

Action: `identitystore:DescribeUser` on \*

Action: `identitystore:IsMemberInGroups` on \*

Action: `identitystore>ListGroupMemberships` on \*

Action: `organizations:DescribeOrganization` on \*

Action: `sso:DescribeApplication` on \*

Action: `sso:DescribeInstance` on \*

Action: `sso>ListInstances` on \*

You must configure permissions to allow your users, groups, or roles to create, edit, or delete a service-linked role. For more information, see [Service-linked role permissions](#) in the *IAM User Guide*.

## Creating a service-linked role for User Subscriptions

You don't need to manually create a service-linked role. When you create a User Subscription in the AWS Management Console, User Subscriptions creates the service-linked role for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you update the settings, User Subscriptions creates the service-linked role for you again.

You can also use the IAM console or AWS CLI to create a service-linked role with the `q.amazonaws.com` service name. For more information, see [Creating a service-linked role](#) in the *IAM User Guide*. If you delete this service-linked role, you can use this same process to create the role again.

## Editing a service-linked role for User Subscriptions

User Subscriptions does not allow you to edit the `AWSServiceRoleForUserSubscriptions` service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see [Editing a service-linked role](#) in the *IAM User Guide*.

## Deleting a service-linked role for User Subscriptions

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

### Note

If the User Subscriptions service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

## To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the `AWSServiceRoleForUserSubscriptions` service-linked role. For more information, see [Deleting a service-linked role](#) in the *IAM User Guide*.

## Supported Regions for User Subscriptions service-linked roles

Amazon Q Developer Subscriptions supports using service-linked roles in all of the Regions where the service is available. For more information, see [AWS Regions and endpoints](#).

Amazon Q Developer Subscriptions does not support using service-linked roles in every Region where the service is available. You can use the `AWSServiceRoleForUserSubscriptions` role in the following Regions.

Region name	Region identity	Support in User Subscriptions
US East (N. Virginia)	us-east-1	Yes
US West (Oregon)	us-west-2	Yes
US East (N. Virginia)	us-east-1	Yes
US East (Ohio)	us-east-2	Yes
US East (Ohio)	us-east-2	Yes
US West (N. California)	us-west-1	Yes
Asia Pacific (Mumbai)	ap-south-1	Yes
Asia Pacific (Osaka)	ap-northeast-3	Yes
Asia Pacific (Seoul)	ap-northeast-2	Yes
Asia Pacific (Singapore)	ap-southeast-1	Yes
Asia Pacific (Sydney)	ap-southeast-2	Yes
Asia Pacific (Tokyo)	ap-northeast-1	Yes
Canada (Central)	ca-central-1	Yes
Europe (Frankfurt)	eu-central-1	Yes
Europe (Ireland)	eu-west-1	Yes
Europe (London)	eu-west-2	Yes
Europe (Paris)	eu-west-3	Yes
Europe (Stockholm)	eu-north-1	Yes
South America (São Paulo)	sa-east-1	Yes

# Compliance validation for Amazon Q Developer

For a list of AWS services in scope of specific compliance programs, see [AWS services in Scope by Compliance Program](#). 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](#) in the *AWS Artifact User Guide*.

Your compliance responsibility when using Amazon Q Developer is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- [Security and Compliance Quick Start Guides](#) – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- [Architecting for HIPAA Security and Compliance Whitepaper](#) – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- [AWS Compliance Resources](#) – This collection of workbooks and guides might apply to your industry and location.
- [Evaluating Resources with Rules](#) in the *AWS Config Developer Guide* – AWS Config assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- [AWS Security Hub](#) – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

## Resilience in Amazon Q Developer

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

For more information about AWS Regions and Availability Zones, see [AWS Global Infrastructure](#).

# Infrastructure security in Amazon Q Developer

As a managed service, Amazon Q is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see [AWS Cloud Security](#). To design your AWS environment using the best practices for infrastructure security, see [Infrastructure Protection](#) in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access Amazon Q Developer through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

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

## Amazon Q Developer and interface VPC endpoints (AWS PrivateLink)

### Note

Amazon Q Developer supports VPC endpoints for features available [in your IDE](#). Chatting with Amazon Q [on AWS apps and websites](#) is not supported.

You can establish a private connection between your VPC and Amazon Q Developer by creating an *interface VPC endpoint*. Interface endpoints are powered by [AWS PrivateLink](#), a technology that enables you to privately access Amazon Q APIs without an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to communicate with Amazon Q APIs. Traffic between your VPC and Amazon Q does not leave the Amazon network.

Each interface endpoint is represented by one or more [Elastic Network Interfaces](#) in your subnets.

For more information, see [Interface VPC endpoints \(AWS PrivateLink\)](#) in the *Amazon VPC User Guide*.

## Considerations for Amazon Q VPC endpoints

Before you set up an interface VPC endpoint for Amazon Q, ensure that you review [Interface endpoint properties and limitations](#) in the *Amazon VPC User Guide*.

Amazon Q supports making calls to all of its API actions from your VPC, in the context of services that are configured to work with Amazon Q.

## Prerequisites

Before you begin any of the procedures below, ensure that you have the following:

- An AWS account with appropriate permissions to create and configure resources.
- A VPC already created in your AWS account.
- Familiarity with AWS services, especially Amazon VPC and Amazon Q.

## Creating an interface VPC endpoint for Amazon Q

You can create a VPC endpoint for the Amazon Q service using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see [Creating an interface endpoint](#) in the *Amazon VPC User Guide*.

Create a VPC endpoint for Amazon Q using the following service name:

- com.amazonaws.*region*.q

### Warning

If you are connecting to Amazon Q Developer through AWS PrivateLink from a third-Party IDE, then you must also create the following endpoint:

- com.amazonaws.*region*.codewhisperer

If you enable private DNS for the endpoint, you can make API requests to Amazon Q using its default DNS name for the Region, for example, q.us-east-1.amazonaws.com.

For more information, see [Accessing a service through an interface endpoint](#) in the *Amazon VPC User Guide*.

 **Note**

Currently, you can only create an interface endpoint for Amazon Q Developer in the US East (N. Virginia) Region.

## Using an on-premises computer to connect to a Amazon Q endpoint

This section describes the process of using an on-premises computer to connect to Amazon Q through a AWS PrivateLink endpoint in your AWS VPC.

1. [Create a VPN connection between your on-premises device and your VPC.](#)
2. [Create an interface VPC endpoint for Amazon Q.](#)
3. [Set up an inbound Amazon Route 53 endpoint.](#) This will enable you to use the DNS name of your Amazon Q endpoint from your on-premises device.

## Using an in-console coding environment to connect to a Amazon Q endpoint

This section describes the process of using an in-console coding environment to connect to a Amazon Q endpoint.

In this context, an in-console IDE is an IDE that you access inside the AWS console, and authenticate to with IAM. Examples include AWS Cloud9, SageMaker Studio, and AWS Glue Studio.

1. [Create an interface VPC endpoint for Amazon Q.](#)
2. Set up Amazon Q with the in-console coding environment
  - [AWS Cloud9](#)
  - [SageMaker Studio](#)
  - [AWS Glue Studio](#)
3. Configure the coding environment to use the Amazon Q endpoint.
  - [AWS Cloud9](#)

- [SageMaker Studio](#)
- [AWS Glue Studio](#)

## Connecting to Amazon Q through AWS PrivateLink from a third-Party IDE on an Amazon EC2 instance

This section will walk you through the process of installing a third-party Integrated Development Environment (IDE) like Visual Studio Code or JetBrains on an Amazon EC2 instance, and configuring it to connect to Amazon Q using AWS PrivateLink.

1. [Create an interface VPC endpoint for Amazon Q.](#)
2. Launch an Amazon EC2 instance in your desired subnet within your VPC. You can choose an Amazon Machine Image (AMI) that is compatible with your third-party IDE. For example, you can select an Amazon Linux 2 AMI.
3. Connect to the Amazon EC2 instance.
4. Install and Configure the IDE (Visual Studio Code or JetBrains).
5. [Install the Amazon Q extension or plugin.](#)
6. Configure the IDE to connect via AWS PrivateLink.
  - [Network connections in Visual Studio Code](#)
  - [JetBrains remote development](#)

# Monitoring Amazon Q Developer

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

- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon Simple Storage Service (Amazon S3) bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the [AWS CloudTrail User Guide](#).
- *Amazon CloudWatch* monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track the number of times that Amazon Q has been invoked on your account, or the number of daily active users. For more information, see the [Amazon CloudWatch User Guide](#).

## Logging Amazon Q Developer API calls using AWS CloudTrail

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

For more information about CloudTrail, see the [AWS CloudTrail User Guide](#).

## Amazon Q Developer information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon Q Developer, that activity is recorded in a CloudTrail event along with other AWS service

events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#) in the *AWS CloudTrail User Guide*.

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

- [Overview for creating a trail](#)
- [CloudTrail supported services and integrations](#)
- [Configuring Amazon SNS notifications for CloudTrail](#)
- [Receiving CloudTrail log files from multiple Regions](#)
- [Receiving CloudTrail log files from multiple accounts](#)

All Amazon Q Developer actions are logged by CloudTrail and generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials
- Whether the request was made with temporary security credentials for a role or federated user
- Whether the request was made by another AWS service

For more information, see [CloudTrail userIdentity element](#) in the *AWS CloudTrail User Guide*.

## Understanding Amazon Q Developer log file entries

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

Amazon Q Developer also makes API calls with a `dryRun` parameter to verify that you have the necessary permissions for the action, without actually making the request. Calls to Amazon Q Developer APIs with the `dryRun` parameter are captured as events and recorded in a CloudTrail log with `"dryRun" : true` in the `requestParameters` field.

The following example shows a CloudTrail log entry that demonstrates the `SendMessage` action.

```
{  
    "eventVersion": "1.08",  
    "userIdentity": {  
        "type": "AssumedRole",  
        "principalId": "AROAXD12ABCDEF3G4HI5J:aws-user",  
        "arn": "arn:aws:sts::123456789012:assumed-role/PowerUser/aws-user",  
        "accountId": "123456789012",  
        "accessKeyId": "ASIAAB12CDEFG34HIJK",  
        "sessionContext": {  
            "sessionIssuer": {  
                "type": "Role",  
                "principalId": "AROAXD12ABCDEF3G4HI5J",  
                "arn": "arn:aws:iam::123456789012:role/PowerUser",  
                "accountId": "123456789012",  
                "userName": "PowerUser"  
            },  
            "webIdFederationData": {},  
            "attributes": {  
                "creationDate": "2023-11-28T10:00:00Z",  
                "mfaAuthenticated": "false"  
            }  
        }  
    },  
    "eventTime": "2023-11-28T10:00:00Z",  
    "eventSource": "q.amazonaws.com",  
    "eventName": "SendMessage",  
    "awsRegion": "us-east-1",  
    "sourceIPAddress": "123.456.789.012",  
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0",  
    "requestParameters": {  
        "Origin": "https://conversational-experience-worker.widget.console.aws.amazon.com",  
        "conversationId": "a298ec0d-0a49-4d2e-92bd-7d6e629b4619",  
        "source": "CONSOLE",  
        "conversationToken": "****",  
    }  
}
```

```
        "utterance": "***"
    },
    "responseElements": {
        "result": {
            "content": {
                "text": {
                    "body": "***",
                    "references": []
                }
            },
            "format": "PLAINTEXT",
            "intents": {},
            "type": "TEXT"
        },
        "Access-Control-Expose-Headers": "x-amzn-RequestId, x-amzn-ErrorType, x-amzn-ErrorMessage, Date",
        "metadata": {
            "conversationExpirationTime": "2024-02-25T19:31:38Z",
            "conversationId": "a298ec0d-0a49-4d2e-92bd-7d6e629b4619",
            "conversationToken": "***",
            "utteranceId": "3b87b46f-04a9-41ef-b8fe-8abf52d2c053"
        },
        "resultCode": "LLM"
    },
    "requestID": "19b3c30e-906e-4b7f-b5c3-509f67248655",
    "eventID": "a552c487-7d97-403a-8ec4-d49539c7a03d",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

The following example shows a CloudTrail log entry that demonstrates the PassRequest action.

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDA60N6E4XEGIEXAMPLE",
        "arn": "arn:aws:iam::555555555555:user/Mary",
        "accountId": "555555555555",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    }
}
```

```
"sessionContext": {  
    "sessionIssuer": {  
        "type": "Role",  
        "principalId": "AIDA60N6E4XEGIEXAMPLE",  
        "arn": "arn:aws:iam::555555555555:user/Mary",  
        "accountId": "555555555555",  
        "userName": "Mary"  
  
    },  
    "attributes": {  
        "creationDate": "2024-04-10T20:03:01Z",  
        "mfaAuthenticated": "false"  
    },  
    "invokedBy": "q.amazonaws.com"  
},  
"eventTime": "2024-04-10T20:04:42Z",  
"eventSource": "q.amazonaws.com",  
"eventName": "PassRequest",  
"awsRegion": "us-east-1",  
"sourceIPAddress": "q.amazonaws.com",  
"userAgent": "q.amazonaws.com",  
"requestParameters": null,  
"responseElements": null,  
"requestID": "2d528c76-329e-410b-9516-EXAMPLE565dc",  
"eventID": "ba0801a1-87ec-4d26-be87-EXAMPLE75bbb",  
"readOnly": false,  
"eventType": "AwsApiCall",  
"managementEvent": true,  
"recipientAccountId": "555555555555",  
"eventCategory": "Management"  
}
```

The following example shows a CloudTrail log entry that demonstrates Amazon Q calling the `s3>ListBuckets` action on your behalf.

```
{  
    "eventVersion": "1.09",  
    "userIdentity": {  
        "type": "AssumedRole",  
        "principalId": "AIDA60N6E4XEGIEXAMPLE",  
        "arn": "arn:aws:iam::555555555555:user/Paulo",  
        "accountId": "555555555555",  
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",  
        "sessionContext": {  
            "sessionIssuer": {  
                "type": "Role",  
                "principalId": "AIDA60N6E4XEGIEXAMPLE",  
                "arn": "arn:aws:iam::555555555555:user/Paulo",  
                "accountId": "555555555555",  
                "userName": "Paulo"  
            },  
            "attributes": {  
                "creationDate": "2024-04-10T20:03:01Z",  
                "mfaAuthenticated": "false"  
            },  
            "invokedBy": "q.amazonaws.com"  
        },  
        "eventTime": "2024-04-10T20:04:42Z",  
        "eventSource": "q.amazonaws.com",  
        "eventName": "PassRequest",  
        "awsRegion": "us-east-1",  
        "sourceIPAddress": "q.amazonaws.com",  
        "userAgent": "q.amazonaws.com",  
        "requestParameters": null,  
        "responseElements": null,  
        "requestID": "2d528c76-329e-410b-9516-EXAMPLE565dc",  
        "eventID": "ba0801a1-87ec-4d26-be87-EXAMPLE75bbb",  
        "readOnly": false,  
        "eventType": "AwsApiCall",  
        "managementEvent": true,  
        "recipientAccountId": "555555555555",  
        "eventCategory": "Management"  
    }  
}
```

```
"sessionContext": {
    "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDA60N6E4XEGIEXAMPLE",
        "arn": "arn:aws:iam::555555555555:user/Paulo",
        "accountId": "555555555555",
        "userName": "Paulo"
    },
    "attributes": {
        "creationDate": "2024-04-10T14:06:08Z",
        "mfaAuthenticated": "false"
    }
},
"invokedBy": "q.amazonaws.com"
},
"eventTime": "2024-04-10T14:07:55Z",
"eventSource": "s3.amazonaws.com",
"eventName": "ListBuckets",
"awsRegion": "us-east-1",
"sourceIPAddress": "q.amazonaws.com",
"userAgent": "q.amazonaws.com",
"requestParameters": {
    "Host": "s3.amazonaws.com"
},
"responseElements": null,
"additionalEventData": {
    "SignatureVersion": "SigV4",
    "CipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "bytesTransferredIn": 0,
    "AuthenticationMethod": "AuthHeader",
    "x-amz-id-2": "ExampleRequestId123456789",
    "bytesTransferredOut": 4054
},
"requestID": "ecd94349-b36f-44bf-b6f5-EXAMPLE9c463",
"eventID": "2939ba50-1d26-4a5a-83bd-EXAMPLE85850",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "555555555555",
"vpcEndpointId": "vpce-EXAMPLE1234",
"eventCategory": "Management"
}
```

# Monitoring Amazon Q Developer with Amazon CloudWatch

## Note

The metrics discussed here only pertain to using [Amazon Q in your IDE](#).

You can monitor Amazon Q Developer using CloudWatch, which collects raw data and processes it into readable, near real-time metrics. These statistics are kept for 15 months, so that you can access historical information and gain a better perspective on how Amazon Q is performing. You can also set alarms that watch for certain thresholds, and send notifications or take actions when those thresholds are met. For more information, see the [Amazon CloudWatch User Guide](#).

The Amazon Q Developer service reports the following metrics in the AWS/CodeWhisperer namespace.

Dimensions	Metric	Use case or explanation
Count	Invocations	You want to determine how many invocations have been counted over time.
UserCount	DailyActiveUserTrend	You want to determine the number of active users per day.
SubscriptionUserCount	SubscriptionCount	You want to determine the number of users with paying subscriptions.
UniqueUserCount	MonthlyActiveUniqueUsers	You want to determine the number of users who are active in a given month.
ProgrammingLanguage, SuggestionState,	GeneratedLineCount	You want to determine the number of lines generated by CodeWhisperer.

Dimensions	Metric	Use case or explanation
CompletionType		
ProgrammingLanguage, SuggestionState, CompletionType	SuggestionReferenceCount	You want to determine the number of recommendation triggers with references that have taken place.
ProgrammingLanguage	CodeScanCount	You want to determine the number of code scans that have taken place.
ProgrammingLanguage	TotalCharacterCount	The number of characters in your file, including all suggestions from CodeWhisperer.
ProgrammingLanguage	CodeWhispererCharacterCount	The number of characters generated by CodeWhisperer.

To aggregate Invocations, use the Sum statistic.

To aggregate DailyActiveUserTrend, use the Sum statistic, and use "1 Day" as the period.

To aggregate SubscriptionCount, use the Sum statistic.

To aggregate MonthlyActiveUniqueUsers use the Sum statistic, and use "30 Days" as the period.

## Tracking Amazon Q Developer usage across your organization

Your business may operate many different AWS accounts that are all part of one AWS organization. In that case, you may want to create a separate Amazon Q Developer instance for each of your

AWS accounts. Then, you can assign a different Amazon Q administrator, and a different (or overlapping) set of developers to each account.

When a Amazon Q administrator views the dashboard, they will only see information about the account to which they have been assigned.

Billing for Amazon Q Developer Pro usage is per AWS organization. If the same developer uses Amazon Q Developer in multiple accounts within the same organization, you will not be double-billed.

## Tracking subscription expenses per user

You can identify the cost of Amazon Q subscriptions for specific users with resource IDs through AWS Billing and Cost Management.

To do so, in the Billing and Cost Management console under [Data Exports](#), create either a standard data export or a legacy CUR export with the **Include resource IDs** option selected.

To learn more, refer to [Creating data exports](#) in the AWS Data Exports user guide.

## Identifying actions by specific users with Amazon CloudWatch Logs

It's possible to get user-level metrics on your Amazon Q Developer usage. To figure out which user has taken a particular action, look for the events called SendTelemetryEvent, and examine the JSON object of type SendTelemetryEventRequestthat they contain. Within that object, the schema appears as follows.

```
http://json-schema.org/draft-07/schema#",
  "definitions": {
    "ProgrammingLanguage": {
      "type": "object",
      "properties": {
        "languageName": {
          "type": "string",
          "enum": [
            "python",
            "javascript",
            "java",
            "csharp",
            "typescript",
            "go",
            "c",
            "rust",
            "kotlin",
            "php",
            "sql"
          ]
        }
      }
    }
  }
}
```

```
        "c",
        "cpp",
        "go",
        "kotlin",
        "php",
        "ruby",
        "rust",
        "scala",
        "shell",
        "sql",
        "json",
        "yaml",
        "vue",
        "tf",
        "tsx",
        "jsx",
        "plaintext"
    ],
    "description": "Programming Languages supported by Q"
}
}

},
"Dimension": {
    "type": "object",
    "properties": {
        "name": {
            "type": "string",
            "description": "must match ^[-a-zA-Z0-9._]*$ and be between 1 and
255 characters"
        },
        "value": {
            "type": "string",
            "description": "must match ^[-a-zA-Z0-9._]*$ and be between 1 and
1024 characters"
        }
    }
},
"telemetryEvents": {
    "UserTriggerDecisionEvent": {
        "type": "object",
        "properties": {
            "sessionId": {
                "type": "string",

```

```
        "description": "UUID for the session"
    },
    "requestId": {
        "type": "string",
        "description": "UUID for the request"
    },
    "customizationArn": {
        "type": "string",
        "description": "ARN of the customization matching pattern: ^arn:[-.a-zA-Z0-9]{1,63}:codewhisperer:[-.a-zA-Z0-9]{0,63}:[{2}([a-zA-Z0-9-_:/])]{1,1023}$"
    },
    "programmingLanguage": {
        "$ref": "#/definitions/ProgrammingLanguage"
    },
    "completionType": {
        "type": "string",
        "enum": [
            "BLOCK",
            "LINE"
        ]
    },
    "suggestionState": {
        "type": "string",
        "enum": [
            "ACCEPT",
            "REJECT",
            "DISCARD",
            "EMPTY"
        ]
    },
    "recommendationLatencyMilliseconds": {
        "type": "number"
    },
    "timestamp": {
        "type": "string",
        "description": "datetime, example: Jul 23, 2024, 12:11:02 AM"
    },
    "triggerToResponseLatencyMilliseconds": {
        "type": "number"
    },
    "suggestionReferenceCount": {
        "type": "integer"
    },
    "generatedLine": {
```

```
        "type": "integer"
    },
    "numberOfRecommendations": {
        "type": "integer"
    }
},
"required": [
    "sessionId",
    "requestId",
    "programmingLanguage",
    "completionType",
    "suggestionState",
    "recommendationLatencyMilliseconds",
    "timestamp"
]
},
"CodeCoverageEvent": {
    "type": "object",
    "properties": {
        "customizationArn": {
            "type": "string",
            "description": "ARN of the customization matching pattern: ^arn:[-.a-zA-Z0-9]{1,63}:codewhisperer:([-.a-zA-Z0-9]{0,63}):{2}([a-zA-Z0-9-_:/]){1,1023}$"
        },
        "programmingLanguage": {
            "$ref": "#/definitions/ProgrammingLanguage"
        },
        "acceptedCharacterCount": {
            "type": "integer"
        },
        "totalCharacterCount": {
            "type": "integer"
        },
        "timestamp": {
            "type": "string",
            "description": "datetime, example: Jul 23, 2024, 12:11:02 AM"
        },
        "unmodifiedAcceptedCharacterCount": {
            "type": "integer"
        }
},
"required": [
    "programmingLanguage",
    "acceptedCharacterCount",

```

```
        "totalCharacterCount",
        "timestamp"
    ],
},
"UserModificationEvent": {
    "type": "object",
    "properties": {
        "sessionId": {
            "type": "string",
            "description": "UUID for the session"
        },
        "requestId": {
            "type": "string",
            "description": "UUID for the request"
        },
        "programmingLanguage": {
            "$ref": "#/definitions/ProgrammingLanguage"
        },
        "modificationPercentage": {
            "type": "number",
            "description": "This is the percentage of AI generated code which has been modified by the user"
        },
        "customizationArn": {
            "type": "string",
            "description": "ARN of the customization matching pattern: ^arn: [-.a-zA-Z0-9]{1,63}:codewhisperer:([-.a-zA-Z0-9]{0,63}:[{2}([a-zA-Z0-9-_:/])]{1,1023}$"
        },
        "timestamp": {
            "type": "string",
            "description": "datetime, example: Jul 23, 2024, 12:11:02 AM"
        }
    },
    "required": [
        "sessionId",
        "requestId",
        "programmingLanguage",
        "modificationPercentage",
        "timestamp"
    ]
},
"CodeScanEvent": {
    "type": "object",
    "properties": {
```

```
        "programmingLanguage": {
            "$ref": "#/definitions/ProgrammingLanguage"
        },
        "codeScanJobId": {
            "type": "string"
        },
        "timestamp": {
            "type": "string",
            "description": "datetime, example: Jul 23, 2024, 12:11:02 AM"
        },
        "codeAnalysisScope": {
            "type": "string",
            "enum": [
                "FILE",
                "PROJECT"
            ]
        }
    },
    "required": [
        "programmingLanguage",
        "codeScanJobId",
        "timestamp"
    ]
},
"CodeScanRemediationsEvent": {
    "type": "object",
    "properties": {
        "programmingLanguage": {
            "$ref": "#/definitions/ProgrammingLanguage"
        },
        "CodeScanRemediationsEventType": {
            "type": "string",
            "enum": [
                "CODESCAN_ISSUE_HOVER",
                "CODESCAN_ISSUE_APPLY_FIX",
                "CODESCAN_ISSUE_VIEW_DETAILS"
            ]
        },
        "timestamp": {
            "type": "string",
            "description": "datetime, example: Jul 23, 2024, 12:11:02 AM"
        },
        "detectorId": {
            "type": "string"
        }
    }
}
```

```
        },
        "findingId": {
            "type": "string"
        },
        "ruleId": {
            "type": "string"
        },
        "component": {
            "type": "string"
        },
        "reason": {
            "type": "string"
        },
        "result": {
            "type": "string"
        },
        "includesFix": {
            "type": "boolean"
        }
    }
},
"MetricData": {
    "type": "object",
    "properties": {
        "metricName": {
            "type": "string",
            "description": "must match pattern ^[-a-zA-Z0-9._]*$ and be between 1 and 1024 characters"
        },
        "metricValue": {
            "type": "number"
        },
        "timestamp": {
            "type": "string",
            "description": "datetime, example: Jul 23, 2024, 12:11:02 AM"
        },
        "product": {
            "type": "string",
            "description": "must match pattern ^[-a-zA-Z0-9._]*$ and be between 1 and 128 characters"
        },
        "dimensions": {
            "type": "array",
            "items": {

```

```
        "$ref": "#/definitions/Dimension"
    },
    "description": "maximum size of 30"
}
},
"required": [
    "metricName",
    "metricValue",
    "timestamp",
    "product"
]
},
"ChatAddMessageEvent": {
    "type": "object",
    "properties": {
        "conversationId": {
            "type": "string",
            "description": "ID which represents a multi-turn conversation, length between 1 and 128"
        },
        "messageId": {
            "type": "string",
            "description": "Unique identifier for the chat message"
        },
        "customizationArn": {
            "type": "string",
            "description": "ARN of the customization matching pattern: ^arn: [-.a-zA-Z0-9]{1,63}:codewhisperer:([-.a-zA-Z0-9]{0,63}):){2}([a-zA-Z0-9-_:/]){1,1023}$"
        },
        "userIntent": {
            "type": "string",
            "enum": [
                "SUGGEST_ALTERNATE_IMPLEMENTATION",
                "APPLY_COMMON_BEST_PRACTICES",
                "IMPROVE_CODE",
                "SHOW_EXAMPLES",
                "CITE_SOURCES",
                "EXPLAIN_LINE_BY_LINE",
                "EXPLAIN_CODE_SELECTION",
                "GENERATE_CLOUDFORMATION_TEMPLATE"
            ]
        },
        "hasCodeSnippet": {
            "type": "boolean"
        }
    }
}
```

```
        },
        "programmingLanguage": {
            "$ref": "#/definitions/ProgrammingLanguage"
        },
        "activeEditorTotalCharacters": {
            "type": "integer"
        },
        "timeToFirstChunkMilliseconds": {
            "type": "number"
        },
        "timeBetweenChunks": {
            "type": "array",
            "items": {
                "type": "number"
            },
            "description": "maximum size of 100"
        },
        "fullResponseLatency": {
            "type": "number"
        },
        "requestLength": {
            "type": "integer"
        },
        "responseLength": {
            "type": "integer"
        },
        "numberOfCodeBlocks": {
            "type": "integer"
        },
        "hasProjectLevelContext": {
            "type": "boolean"
        }
    },
    "required": [
        "conversationId",
        "messageId"
    ]
},
"ChatInteractWithMessageEvent": {
    "type": "object",
    "properties": {
        "conversationId": {
            "type": "string",

```

```
        "description": "ID which represents a multi-turn conversation,  
length between 1 and 128"  
    },  
    "messageId": {  
        "type": "string",  
        "description": "Unique identifier for the chat message"  
    },  
    "customizationArn": {  
        "type": "string",  
        "description": "ARN of the customization matching pattern: ^arn:  
[-.a-z0-9]{1,63}:codewhisperer:([-.a-z0-9]{0,63}):){2}([a-zA-Z0-9-_:/]){1,1023}$"  
    },  
    "interactionType": {  
        "type": "string",  
        "enum": [  
            "INSERT_AT_CURSOR",  
            "COPY_SNIPPET",  
            "COPY",  
            "CLICK_LINK",  
            "CLICK_BODY_LINK",  
            "CLICK_FOLLOW_UP",  
            "HOVER_REFERENCE",  
            "UPVOTE",  
            "DOWNVOTE"  
        ],  
        "description": "Chat Message Interaction Type"  
    },  
    "interactionTarget": {  
        "type": "string",  
        "description": "Target of message interaction"  
    },  
    "acceptedCharacterCount": {  
        "type": "integer"  
    },  
    "acceptedLineCount": {  
        "type": "integer"  
    },  
    "acceptedSnippetHasReference": {  
        "type": "boolean"  
    },  
    "hasProjectLevelContext": {  
        "type": "boolean"  
    }  
},  
},
```

```
        "required": [
            "conversationId",
            "messageId"
        ],
    },
    "ChatUserModificationEvent": {
        "type": "object",
        "properties": {
            "conversationId": {
                "type": "string",
                "description": "ID which represents a multi-turn conversation, length between 1 and 128"
            },
            "customizationArn": {
                "type": "string",
                "description": "ARN of the customization matching pattern: ^arn: [-.a-zA-Z0-9]{1,63}:codewhisperer:([-.a-zA-Z0-9]{0,63}:[{2}([a-zA-Z0-9-_:/])]{1,1023}$"
            },
            " messageId": {
                "type": "string",
                "description": "Unique identifier for the chat message"
            },
            "programmingLanguage": {
                "$ref": "#/definitions/ProgrammingLanguage"
            },
            "modificationPercentage": {
                "type": "number",
                "description": "This is the percentage of AI generated code which has been modified by the user"
            },
            "hasProjectLevelContext": {
                "type": "boolean"
            }
        },
        "required": [
            "conversationId",
            "messageId",
            "modificationPercentage"
        ]
    },
    "SuggestionState": {
        "type": "string",
        "enum": [
            "ACCEPT",
            "REJECT"
        ]
    }
}
```

```
        "REJECT",
        "DISCARD",
        "EMPTY"
    ],
},
"TerminalUserInteractionEvent": {
    "type": "object",
    "properties": {
        "terminalUserInteractionEventType": {
            "type": "string",
            "enum": [
                "CODEWHISPERER_TERMINAL_TRANSLATION_ACTION",
                "CODEWHISPERER_TERMINAL_COMPLETION_INSERTED"
            ],
            "description": "Terminal User Interaction Event Type"
        },
        "terminal": {
            "type": "string"
        },
        "terminalVersion": {
            "type": "string"
        },
        "shell": {
            "type": "string"
        },
        "shellVersion": {
            "type": "string"
        },
        "duration": {
            "type": "integer"
        },
        "timeToSuggestion": {
            "type": "integer"
        },
        "isCompletionAccepted": {
            "type": "boolean"
        },
        "cliToolCommand": {
            "type": "string"
        }
    }
},
"FeatureDevEvent": {
    "type": "object",

```

```
        "properties": {
            "conversationId": {
                "type": "string",
                "description": "ID which represents a multi-turn conversation, length between 1 and 128"
            }
        },
        "required": [
            "conversationId"
        ]
    }
},
"SendTelemetryEventRequest": {
    "type": "object",
    "properties": {
        "clientToken": {
            "type": "string",
            "description": "The client's authentication token"
        },
        "telemetryEvent": {
            "properties": {
                "oneOf": [
                    {
                        "_comment": "This event is emitted when a user accepts or rejects an inline code suggestion",
                        "$ref": "#/definitions/userTriggerDecisionEvent"
                    },
                    {
                        "_comment": "This event is emitted every five minutes. It details how much code is written by inline code suggestion and in total during that period",
                        "$ref": "#/definitions/codeCoverageEvent"
                    },
                    {
                        "_comment": "This event is emitted when a code snippet from inline code suggestion has been edited by a user. It details the percentage of that code snippet modified by the user",
                        "$ref": "#/definitions/userModificationEvent"
                    },
                    {
                        "_comment": "This field is emitted when a security scan is requested by a user",
                        "$ref": "#/definitions/codeScanEvent"
                    },
                ]
            }
        }
    }
}
```

```
{  
    "_comment": "This field is emitted when a security scan  
recommended remediation is accepted by a user",  
    "$ref": "#/definitions/codeScanRemediationsEvent"  
},  
{  
    "_comment": "This event is deprecated but may still occur  
in telemetry. Do not use this.",  
    "$ref": "#/definitions/metricData"  
},  
{  
    "_comment": "This event is emitted when Q adds an AI  
generated message to the chat window",  
    "$ref": "#/definitions/chatAddMessageEvent"  
},  
{  
    "_comment": "This event is emitted when a user interacts  
with a chat message",  
    "$ref": "#/definitions/chatInteractWithMessageEvent"  
},  
{  
    "_comment": "This event is emitted when a user modifies a  
code snippet sourced from chat. It gives a percentage of the code snippet which has  
been modified",  
    "$ref": "#/definitions/chatUserModificationEvent"  
},  
{  
    "_comment": "This event is emitted when a user interacts  
with a terminal suggestion",  
    "$ref": "#/definitions/terminalUserInteractionEvent"  
},  
{  
    "_comment": "This event is emitted when a user first  
prompts the /dev feature.",  
    "$ref": "#/definitions/featureDevEvent"  
}  
]  
}  
},  
"optOutPreference": {  
    "type": "string",  
    "enum": [  
        "OPTIN",  
        "OPTOUT"
```

```
        ],
        "description": "OPTOUT and telemetry is only provided to the account of
purchasing enterprise, OPTIN and telemetry may also be used for product improvement"
    },
    "userContext": {
        "type": "object",
        "properties": {
            "ideCategory": {
                "type": "string",
                "enum": [
                    "JETBRAINS",
                    "VSCODE",
                    "CLI",
                    "JUPYTER_MD",
                    "JUPYTER_SM"
                ]
            },
            "operatingSystem": {
                "type": "string",
                "description": "The operating system being used"
            },
            "product": {
                "type": "string",
                "description": "The name of the product being used"
            },
            "clientId": {
                "type": "string",
                "description": "A UUID representing the individual client being
used"
            },
            "ideVersion": {
                "type": "string",
                "description": "The version of the Q plugin"
            }
        },
        "required": [
            "ideCategory",
            "operatingSystem",
            "product",
            "clientId",
            "ideVersion"
        ]
    },
    "profileArn": {
```

```
"type": "string",
"description": "The arn of the Q Profile used to configure individual
user accounts."
```

Observe that a SendTelemetryEvent may contain one of a number of telemetry events. Each of these describes a specific interaction between the development environment.

A more detailed description of each event appears below.

## UserTriggerDecisionEvent

This event is triggered when a user interacts with a suggestion made by Amazon Q. It captures whether the suggestion was accepted, rejected, or modified, along with relevant metadata.

- completionType: Whether the completion was a block or a line.
- suggestionState: Whether the user accepted, rejected, or discarded the suggestion.

## CodeScanEvent

This event is logged when a code scan is performed. It helps track the scope and result of the scan, providing insights into security and code quality checks.

- codeScanJobId: The unique identifier for the code scan job.
- codeAnalysisScope: Whether the scan was performed at the file level or the project level.
- programmingLanguage: The language being scanned.

## CodeScanRemediationsEvent

This event captures user interactions with Amazon Q's remediation suggestions, such as applying fixes or viewing issue details.

- CodeScanRemediationsEventType: The type of remediation action taken (e.g., viewing details or applying a fix).
- includesFix: A boolean indicating whether the user applied a fix.

## ChatAddMessageEvent

This event is triggered when a new message is added to an ongoing chat conversation. It captures the user's intent and any code snippets involved.

- `conversationId`: The unique identifier for the conversation.
- `messageId`: The unique identifier for the chat message.
- `userIntent`: The user's intent, such as improving code or explaining code.
- `programmingLanguage`: The language related to the chat message.

## ChatInteractWithMessageEvent

This event captures when users interact with chat messages, such as copying code snippets, clicking links, or hovering over references.

- `interactionType`: The type of interaction (for example, copy, hover, click).
- `interactionTarget`: The target of the interaction (for example, a code snippet or a link).
- `acceptedCharacterCount`: The number of characters from the message that were accepted.
- `acceptedSnippetHasReference`: A boolean indicating if the accepted snippet included a reference.

## TerminalUserInteractionEvent

This event logs user interactions with terminal commands or completions in the terminal environment.

- `terminalUserInteractionEventType`: The type of interaction (for example, terminal translation or code completion).
- `isCompletionAccepted`: A boolean indicating whether the completion was accepted by the user.
- `duration`: The time taken for the interaction.

# Accessing customization-related messages in Amazon CloudWatch Logs

You can store information about the creation of your [customization](#) in [Amazon CloudWatch Logs](#).

You can authorize your CodeWhisperer administrator to view those logs with the following permission set.

To learn more about the permissions required to delivery logs to multiple resources, see [Logging that requires additional permissions \[V2\]](#) in the *Amazon CloudWatch Logs User Guide*.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowLogDeliveryActions",  
            "Effect": "Allow",  
            "Action": [  
                "logs:PutDeliverySource",  
                "logs:GetDeliverySource",  
                "logs>DeleteDeliverySource",  
                "logs:DescribeDeliverySources",  
                "logs:PutDeliveryDestination",  
                "logs:GetDeliveryDestination",  
                "logs>DeleteDeliveryDestination",  
                "logs:DescribeDeliveryDestinations",  
                "logs>CreateDelivery",  
                "logs:GetDelivery",  
                "logs>DeleteDelivery",  
                "logs:DescribeDeliveries",  
                "firehose>ListDeliveryStreams",  
                "firehose:DescribeDeliveryStream",  
                "s3>ListAllMyBuckets",  
                "s3>ListBucket",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": [  
                "arn:aws:logs:us-east-1:account number:log-group:*",  
                "arn:aws:firehose:us-east-1:account number:deliverystream/*",  
                "arn:aws:s3:::*"  
            ]  
        }  
    ]  
}
```

```
]  
}
```

# Supported Regions for Amazon Q Developer

## Note

If you have enabled cross-region calls in Amazon Q, Amazon Q might make calls to opt-in Regions that aren't listed on this page. To manage access to Regions Amazon Q can make calls to, see [Allow Amazon Q permission to perform actions on your behalf in specific regions](#).

This topic describes the AWS Regions where you can use Amazon Q Developer. For more information about AWS Regions, see [Specify which AWS Regions your account can use](#) in the *AWS Account Management Reference Guide*.

Regardless of where you use Amazon Q Developer, your data is processed in a US Region. Even if you use Amazon Q in a US Region, your data may be processed in a different US Region to support cross-region inference. For more information, see [Cross region inference in Amazon Q Developer](#). For information on where data is stored during processing, see [Data protection](#).

## Supported Regions (enabled by default)

Amazon Q Developer is available in the AWS Management Console, AWS Console Mobile Application, AWS website, AWS Documentation website, and AWS Chatbot in the following AWS Regions. These Regions are enabled by default, meaning you don't need to enable them before use. For more information, see [Regions that are enabled by default](#).

## Note

The Amazon Q Developer console is only available in US East (N. Virginia). To set up Amazon Q Developer, your AWS Management Console must be set to US East (N. Virginia). Certain features of Amazon Q might not be available in all of these Regions. Check the topic for the feature you're using to verify availability.

- US East (Ohio)
- US East (N. Virginia)

- US West (N. California)
- US West (Oregon)
- Asia Pacific (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seoul)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- Canada (Central)
- Europe (Frankfurt)
- Europe (Ireland)
- Europe (London)
- Europe (Paris)
- Europe (Stockholm)
- South America (São Paulo)

## Supported opt-in Regions

To use an opt-in Region with Amazon Q Developer, you must enable the Region manually. For more information, see [Opt-in Regions](#).

Opt-in Regions are only supported in the Amazon Q Developer Free tier. The following opt-in Regions are supported.

- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asia Pacific (Hyderabad)
- Asia Pacific (Jakarta)
- Asia Pacific (Melbourne)
- Europe (Milan)
- Europe (Spain)
- Europe (Zurich)

- Middle East (Bahrain)
- Middle East (UAE)
- Israel (Tel Aviv)

# Quotas for Amazon Q Developer Pro tier subscriptions

Amazon Q Pro tier subscriptions are governed by the usage quotas outlined on the [Amazon Q Developer pricing page](#). Some of the features on the pricing page are subject to specific quotas listed in the following tables. These quotas ensure fair usage and optimal performance for all subscribers.

For information about pricing, see the [Amazon Q Developer pricing page](#).

## Quotas for Amazon Q (in IDEs)

Feature	Quota
<a href="#">Amazon Q Developer Agent for software development</a>	30 invocations per month
<a href="#">Amazon Q Developer Agent for code transformation</a>	4000 lines of code per month per user pooled at the account level. Extra lines of code available. For more information, see <a href="#">Amazon Q Developer Pro pricing</a> .
<a href="#">Security vulnerability and code quality scanning</a>	500 project scans per month, auto scanning included
<a href="#">Customize Amazon Q Developer inline code recommendations and chat responses</a>	Create up to 8 customizations, and keep up to 2 active at the same time

## Quotas for Amazon Q (in the AWS Management Console)

Feature	Quota
<a href="#">Generative SQL</a>	1000 queries per user per month
<a href="#">Analyze network reachability</a>	20 requests per day

## Quotas for Amazon Q (in Amazon CodeCatalyst)

Feature	Quota
<a href="#"><u>Amazon Q Developer Agent for software development</u></a>	30 per month
<a href="#"><u>Pull request summaries</u></a>	20 per month

# Amazon Q Developer rename - Summary of changes

On April 30, 2024, Amazon CodeWhisperer became a part of Amazon Q Developer. This section points you to the parts of this guide where you can find documentation for features that you are accustomed to using through CodeWhisperer.

As you transition from using CodeWhisperer to using Amazon Q Developer, you may consider the following changes to be most significant:

- The [administrative setup](#) at the professional tier (Amazon Q Developer Pro) is different than it was for CodeWhisperer Professional.
- You can [chat with Amazon Q Developer](#) in the AWS Management Console, and on the AWS documentation and marketing websites.

The following familiar features of CodeWhisperer are available as part of Amazon Q Developer, with a few changes:

- Coding suggestions [in a third-party IDE](#)
- Coding suggestions [in the context of another AWS service](#)
- Suggestions [at the command line](#)
- [Customizations](#)
- [Security scans](#)
- [Dashboard](#)

# Document history for Amazon Q Developer User Guide

The following table describes the document history for the *Amazon Q Developer User Guide*. For notifications about updates to this documentation, you can subscribe to the RSS feed.

Change	Description	Date
<a href="#">Transformation on the command line</a>	You can transform Java applications <a href="#">on the command line</a> , available in preview.	November 27, 2024
<a href="#">Multiple diffs for transformation in the IDE</a>	You can choose to receive transformation changes from Amazon Q <a href="#">in multiple diffs</a> .	November 27, 2024
<a href="#">Amazon Q in Eclipse</a>	The <a href="#">Amazon Q plugin</a> is available in preview in Eclipse.	November 27, 2024
<a href="#">Cost analysis</a>	The <a href="#">cost analysis</a> capability, previously available in preview, is now generally available.	November 26, 2024
<a href="#">Transformation for embedded SQL code</a>	You can convert <a href="#">embedded SQL code in your Java applications</a> with Amazon Q transformation in the IDE.	November 22, 2024
<a href="#">Dashboard update</a>	The <a href="#">Amazon Q Developer Pro dashboard</a> has been updated with new metrics.	November 22, 2024
<a href="#">CodeConnections repositories</a>	When <a href="#">creating a customization</a> using a CodeConnections connection, you can now choose the repositories you want to use.	November 22, 2024

<a href="#"><u>Amazon Q command line now supports Linux</u></a>	<a href="#"><u>Amazon Q command line</u></a> supports Linux environments. It supports Ubuntu 22 and 24, and may otherwise work with GNOME v42+ or environments where the display server is Xorg and the input method framework is IBus.	November 21, 2024
<a href="#"><u>Subscribing users</u></a>	The instructions for subscribing users in <a href="#"><u>Setting up access to the Amazon Q Developer Pro tier</u></a> have been updated to reflect new user interface (UI) elements.	November 20, 2024
<a href="#"><u>Changes to customizations</u></a>	The <a href="#"><u>Customization in chat</u></a> feature is now generally available. Also, customizations can now be trained with the following file types: .md, .mdx, .rst, and .txt.	November 20, 2024
<a href="#"><u>Supported IAM Identity Center Regions</u></a>	A section has been added with information about the <a href="#"><u>Regions where you can set up IAM Identity Center instances</u></a> for Amazon Q Developer Pro subscriptions.	November 18, 2024
<a href="#"><u>Languages added</u></a>	<a href="#"><u>Support</u></a> has been added for Dart, Lua, R, Swift, SystemVerilog, and Powershell, as well as expanded support for JSON and YAML.	November 18, 2024

<a href="#"><u>Customer managed key support</u></a>	Information about using customer managed keys and the features that can be encrypted with them has been added to the <a href="#"><u>Data encryption</u></a> topic.	November 18, 2024
<a href="#"><u>Cross-region inference</u></a>	A topic on <a href="#"><u>cross-region inference in Amazon Q Developer</u></a> has been added.	November 18, 2024
<a href="#"><u>Amazon Q Developer Pro quotas</u></a>	A <a href="#"><u>Pro tier quotas</u></a> section has been added.	November 18, 2024
<a href="#"><u>Updated managed policy: AmazonQFullAccess</u></a>	Additional permissions have been added to the <a href="#"><u>AmazonQFullAccess</u></a> policy.	November 13, 2024
<a href="#"><u>Updated managed policy: AmazonQDeveloperAccess</u></a>	Additional permissions have been added to the <a href="#"><u>AmazonQDeveloperAccess</u></a> policy.	November 13, 2024
<a href="#"><u>Amazon Q plugins</u></a>	<a href="#"><u>Plugins</u></a> enable users to chat with Amazon Q about metrics provided by third party tools.	November 13, 2024
<a href="#"><u>User activity reports</u></a>	You can now <a href="#"><u>enable user activity reports</u></a> .	November 8, 2024
<a href="#"><u>Customizations section update</u></a>	The <a href="#"><u>Preparing your data</u></a> section now describes file and directory naming limitations.	November 5, 2024
<a href="#"><u>Clarified the Amazon Q Developer Pro section</u></a>	The instructions for <a href="#"><u>subscribing users to Amazon Q Developer Pro</u></a> have been clarified.	November 1, 2024

<a href="#">Inline chat</a>	You can transform code using the new <a href="#">inline chat</a> feature.	October 29, 2024
<a href="#">Updated managed policies: AmazonQFullAccess and AmazonQDeveloperAccess</a>	Additional permissions have been added to the <a href="#">AmazonQFullAccess</a> policy and <a href="#">AmazonQDeveloperAccess</a> policy.	October 28, 2024
<a href="#">Customizations section correction</a>	The <a href="#">Creating your customization</a> section now indicates that your codebase must reside in a folder in Amazon S3, not the bucket's root.	October 28, 2024
<a href="#">Prompt logging section clarification</a>	The <a href="#">Enabling prompt logging</a> section's wording was clarified.	October 24, 2024
<a href="#">Amazon S3 bucket policy fix</a>	The Amazon S3 bucket policy shown in <a href="#">Enabling prompt logging</a> contained a JSON syntax error that was fixed.	October 22, 2024
<a href="#">Expanded features chapter</a>	The chapter <a href="#">describing various Amazon Q Developer features</a> has been significantly expanded.	October 3, 2024
<a href="#">Console-to-Code</a>	Console-to-Code, previously available in preview as a feature of Amazon EC2, <a href="#">is now generally available</a> as a feature of Amazon Q Developer. It integrates with Amazon EC2, Amazon VPC, and Amazon RDS.	October 3, 2024

<a href="#"><u>New policy: Use Amazon Q CLI with AWS CloudShell</u></a>	Identity-based policy <a href="#">allows users to use Amazon Q CLI with AWS CloudShell.</a>	October 2, 2024
<a href="#"><u>Prompt logging</u></a>	You can <a href="#">log your users' IDE prompts</a> in an Amazon S3 bucket.	September 16, 2024
<a href="#"><u>Setup content updated</u></a>	The <a href="#">Getting started</a> chapter has been significantly simplified and restructured.	August 15, 2024
<a href="#"><u>CodeWhisperer endpoint needed for IDE VPC access</u></a>	<a href="#">Access from a Amazon VPC</a> must include both q and codewhisperer endpoints.	July 18, 2024
<a href="#"><u>New endpoint</u></a>	Endpoints can now <a href="#">use the string q instead of codewhisperer</a> .	July 12, 2024
<a href="#"><u>Customizations are GA</u></a>	The <a href="#">customizations</a> feature is generally available.	July 10, 2024
<a href="#"><u>Chatting about customizations (Preview)</u></a>	In Preview, you can use the <a href="#">customizations</a> feature to ask questions related to your codebase.	July 10, 2024
<a href="#"><u>Updated managed policy: AmazonQFullAccess</u></a>	Additional permissions have been added to the <a href="#">AmazonQFullAccess</a> policy.	July 9, 2024
<a href="#"><u>New managed policy: AmazonQDeveloperAccess</u></a>	The <a href="#">AmazonQDeveloperAccess</a> managed policy provides full access to enable interactions with Amazon Q Developer, without administrator access.	July 9, 2024

<u><a href="#">Updated Amazon Q Developer admin policy</a></u>	The <a href="#">policy for empowering Amazon Q Developer administrators has been updated to include sso:ListProfiles</a> .	June 19, 2024
<u><a href="#">Trusted access section</a></u>	A <a href="#">new section</a> more clearly explains how a Amazon Q Developer administrator can share settings with member accounts.	June 19, 2024
<u><a href="#">Updated setup procedures</a></u>	There's an improved <a href="#">Getting started</a> chapter that includes support for <a href="#">account instances</a> .	June 6, 2024
<u><a href="#">Updated code examples</a></u>	The <a href="#">code examples</a> now include C and C++, and have improved examples for C#.	June 6, 2024
<u><a href="#">Updated managed policy: AmazonQFullAccess</a></u>	Additional permissions have been added to the <a href="#">AmazonQFullAccess</a> policy.	April 30, 2024
<u><a href="#">New service-linked role: AWSServiceRoleForUserSubscriptions</a></u>	The <a href="#">AWSServiceRoleForUserSubscriptions</a> service-linked role provides access for User Subscriptions to your IAM Identity Center resources to automatically update your subscriptions.	April 30, 2024
<u><a href="#">New service-linked role: AWSServiceRoleForAmazonQDeveloper</a></u>	The <a href="#">AWSServiceRoleForAmazonQDeveloper</a> service-linked role grants permission to access and emit data, and to create reports.	April 30, 2024

<a href="#"><u>New managed policy: AWSServiceRoleForUserSubscriptionPolicy</u></a>	The <a href="#"><u>AWSServiceRoleForUserSubscriptionPolicy</u></a> allows principals to track IAM Identity Center directory and AWS Organizations changes.	April 30, 2024
<a href="#"><u>New managed policy: AWSServiceRoleForAmazonQDeveloperPolicy</u></a>	The <a href="#"><u>AWSServiceRoleForAmazonQDeveloperPolicy</u></a> allows Amazon Q Developer to call CloudWatch and CodeGuru on your behalf.	April 30, 2024
<a href="#"><u>GA release</u></a>	Amazon Q Developer is available for general audiences.	April 30, 2024
<a href="#"><u>Amazon CodeWhisperer merge</u></a>	Amazon CodeWhisperer is now a part of Amazon Q Developer.	April 30, 2024
<a href="#"><u>New guide name</u></a>	This service and accompanying user guide have been renamed Amazon Q Developer.	March 29, 2024
<a href="#"><u>New permission</u></a>	The <a href="#"><u>ListConversations action</u></a> is required to chat with Amazon Q in the console.	March 5, 2024
<a href="#"><u>New data protection topic</u></a>	Amazon Q now uses content for <a href="#"><u>service improvement purposes</u></a> .	January 25, 2024
<a href="#"><u>New topic</u></a>	Added instructions for how to <a href="#"><u>add Amazon Q to Slack and Microsoft Teams channels</u></a> that are configured with AWS Chatbot.	January 18, 2024

Preview release

This is the initial preview  
release of the *Amazon Q*  
*Developer User Guide*.

November 28, 2023