

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> .....	<b>1</b>
Key features of Amazon Q .....	1
Get started .....	2
Amazon Q Developer pricing .....	3
Features .....	3
Get answers in context to your conversation .....	4
Diagnose common console errors .....	4
Chat about code .....	4
Get inline code suggestions .....	4
Transform code .....	4
Scan your code for security vulnerabilities .....	4
Contact AWS Support .....	5
Use Amazon Q in AWS Chatbot .....	5
Use Amazon Q in the AWS Console Mobile Application .....	5
<b>Setting up</b> .....	<b>6</b>
Prerequisites .....	6
Step 1: Setting up your AWS account .....	6
Step 2: Setting up AWS Organizations .....	7
Step 3: Setting up IAM Identity Center with Amazon Q Developer .....	7
Step 4: Creating an administrator .....	9
Step 5: Creating an administrator for Amazon Q Subscriptions .....	15
Tiers of service .....	18
Professional tier .....	18
Free tier .....	18
Subscriptions .....	18
Unsubscribing .....	25
Troubleshooting subscriptions .....	26
<b>On AWS</b> .....	<b>28</b>
Add IAM permissions .....	28
Assign permissions .....	30
Chat about AWS .....	30
Working with Amazon Q on AWS websites .....	31
Example questions .....	32
Chat about your resources .....	32

Chat about your costs .....	35
Diagnose console errors .....	36
Diagnose common errors in the console .....	37
Chat with AWS Support .....	37
Prerequisites .....	38
Specify the right service .....	38
Create a support case .....	38
Leave feedback .....	41
Chat with Amazon Q in AWS Chatbot .....	41
Add Amazon Q to a chat channel .....	41
Ask Amazon Q questions in your channel .....	34
<b>In your IDE .....</b>	<b>43</b>
Install the Amazon Q plugin .....	43
Choose a supported version of your IDE .....	44
With Amazon Q Developer Pro .....	44
With Builder ID .....	45
With Visual Studio .....	45
IAM principals in your AWS console .....	45
Understanding AWS coding environments .....	46
Chat about code .....	46
Working with Amazon Q in your IDE .....	47
Example topics and questions .....	48
Reporting issues with responses .....	48
Explain and update code .....	48
Get inline suggestions .....	49
Pausing suggestions .....	50
Amazon Q code completion in action .....	52
Suggestions in AWS coding environments .....	58
Shortcut keys .....	69
Code references .....	75
Code examples .....	85
Transform your code .....	103
Step 1: Prerequisites .....	104
Step 2: Configure your project .....	105
Step 3: Transform your code .....	105
How it works .....	107

Troubleshooting .....	109
Develop software .....	114
Develop features with <code>/dev</code> .....	114
Best practices .....	116
Example tasks .....	116
Scan your code .....	117
Types of scans .....	118
Quotas .....	118
Start a scan .....	119
Address findings .....	123
Supported languages .....	124
Inline suggestions .....	124
Code Transformation .....	125
Security scans .....	125
<b>On your command line .....</b>	<b>127</b>
Installing .....	127
Supported environments .....	128
Verifying .....	128
Chat .....	129
Context integration .....	129
Context modifiers .....	129
Usage .....	129
CLI Completions .....	130
Popular settings .....	131
Over SSH .....	132
Translation .....	137
Debugging .....	139
Expected output .....	139
Modifying .....	140
<b>Customize .....</b>	<b>141</b>
Preparing .....	141
Accessing Amazon CloudWatch Logs .....	144
Creating .....	146
Connecting to your data source .....	146
Customizations and your data .....	148
Troubleshooting .....	148

Deleting .....	149
Evaluating .....	150
Evaluating .....	150
Optimizing .....	150
Troubleshooting .....	151
Setting up log delivery .....	151
Understanding Amazon CloudWatch Logs .....	152
Understanding console error messages .....	154
Activating .....	156
Activating a version .....	156
Updating .....	157
Creating a new version .....	158
Adding users and groups .....	159
Using .....	160
<b>Dashboard .....</b>	<b>161</b>
User activity .....	162
Code impact .....	162
Security scans .....	162
<b>Amazon Q and other services .....</b>	<b>163</b>
Amazon Q Business .....	163
Amazon Q in AWS Chatbot .....	163
Amazon Q in Amazon CodeCatalyst .....	163
Amazon Q in Amazon Connect .....	164
Amazon Q in Amazon EC2 .....	164
Amazon Q in AWS Glue .....	164
Amazon Q in VPC Reachability Analyzer .....	164
Amazon Q in Amazon QuickSight .....	164
<b>Security .....</b>	<b>166</b>
Data protection .....	166
Data encryption .....	168
Service improvement .....	168
Opt out of data sharing in the IDE .....	169
Identity and access management .....	175
Audience .....	176
Authenticating with identities .....	176
Managing access using policies .....	179

---

How Amazon Q works with IAM .....	182
Manage access to Amazon Q .....	189
Amazon Q permissions reference .....	198
AWS managed policies for Amazon Q .....	201
Using service-linked roles .....	206
Compliance validation .....	213
Resilience .....	214
Infrastructure security .....	214
VPC endpoints (AWS PrivateLink) .....	215
Considerations for Amazon Q VPC endpoints .....	215
Prerequisites .....	215
Creating an interface VPC endpoint for Amazon Q .....	216
Using an on-premises computer to connect to a Amazon Q endpoint .....	216
Using an in-console coding environment to connect to a Amazon Q endpoint .....	216
Connecting to Amazon Q through AWS PrivateLink from a third-Party IDE on an Amazon EC2 instance .....	217
<b>Monitoring .....</b>	<b>218</b>
CloudTrail logs .....	218
Amazon Q Developer information in CloudTrail .....	218
Understanding Amazon Q Developer log file entries .....	219
Monitoring with CloudWatch .....	224
Across your organization .....	225
<b>Supported Regions .....</b>	<b>227</b>
Opt-in Regions .....	228
<b>Amazon Q Developer service rename .....</b>	<b>229</b>
<b>Document history .....</b>	<b>230</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 net 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](#).

## Key features of Amazon Q Developer

Amazon Q Developer includes the following capabilities:

- **Chat** – Amazon Q answers natural language questions in English about AWS, including questions about AWS service selection, AWS Command Line Interface (AWS CLI) usage, documentation,



best practices, and your AWS resources. Amazon Q responds with information summaries or step-by-step instructions, and includes links to its information sources.

- **Conversation memory** – Amazon Q uses the context of your conversation to inform future responses for the duration of your conversation.
- **Code improvements and advice** – Within IDEs, Amazon Q can answer questions about software development, improve your code, and generate new code.
- **Code completion** – As you write code, Amazon Q automatically generates suggestions, in real time, based on your existing code and comments.
- **Troubleshoot and support** – Amazon Q can help you understand errors in the AWS Management Console and provides access to live AWS Support agents to address your AWS questions and issues.

For more information about these and other Amazon Q features, see [Amazon Q Developer features](#).

## 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

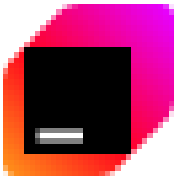
Add the [necessary permissions](#) to your IAM identity, and then choose the Amazon Q icon to start chatting in the AWS Management Console, AWS Documentation website, AWS website, or AWS Console Mobile Application. For more information, see [Amazon Q Developer on AWS apps and websites](#).

### IDEs

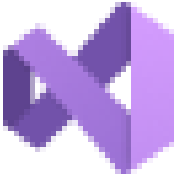
Download the Amazon Q extension and use your AWS Builder ID (no AWS account required) to sign in for free.



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



[Download Amazon Q in JetBrains IDEs](#)



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

From the Amazon Q extension, choose **Open Chat Panel** to start chatting or initiate a development workflow. For more information, see [Install Amazon Q Developer in your IDE](#).

### Command line

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

Beyond answering questions in English about AWS, Amazon Q Developer offers additional functionality and assistance. The availability of these features depends on whether you're using Amazon Q on AWS apps and websites or in an integrated development environment (IDE). This section describes the different features of Amazon Q that you might encounter across your AWS experience.

## Get answers in context to your conversation

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.

## Diagnose common console errors

With Amazon Q, you can diagnose console error messages directly in the AWS Management Console. Amazon Q provides information about the error and an overview of a potential solution. For more information, see [Diagnose common errors in the console 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 [Chat with Amazon Q Developer about code](#).

## 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 [Get inline suggestions with Amazon Q](#).

## Transform 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 [Upgrade language versions with the Amazon Q Developer Agent for Code Transformation](#).

## 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 [Scan your code with Amazon Q](#).

## Contact AWS Support

You can create AWS Support cases and chat with a live support agent within Amazon Q to address issues you're having with AWS services. You can access AWS Support with Amazon Q in the AWS Management Console. For more information, see [Use Amazon Q Developer to chat with AWS Support](#).

## Use Amazon Q in AWS Chatbot

You can add Amazon Q to your Slack and Microsoft Teams channels that are configured with AWS Chatbot. Amazon Q in AWS Chatbot can answer questions about building at AWS, including those about best practices for building solutions, troubleshooting issues, and identifying next steps. For more information, see [Chat with Amazon Q Developer in AWS Chatbot](#).

## 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 [Setting up Amazon Q Developer](#).

# Setting up Amazon Q Developer

This section relates to setting up Amazon Q Developer as an administrator for your organization. To start using Amazon Q Developer as a non-administrative user, see the following topics:

- [Add permissions for Amazon Q on AWS apps and websites](#)
- [Install the Amazon Q extension in your IDE](#)
- [Set up AWS coding environments for inline suggestions](#)

For administrators setting up Amazon Q Developer, complete the topics in this chapter.

## Topics

- [Account, IAM Identity Center, and Organization setup](#)
- [Understanding tiers of service for Amazon Q Developer](#)
- [Subscriptions](#)

### Note

In the AWS Management Console, AWS Console Mobile Application, AWS website, and AWS Documentation website, the Amazon Q icon appears regardless of whether Amazon Q is enabled.

If a user, group, or role has the [AdministratorAccess](#) policy attached, Amazon Q is enabled. For information about how to manage access to Amazon Q, see [Manage access to Amazon Q Developer with policies](#).

## Account, IAM Identity Center, and Organization setup

### Step 1: Setting up your AWS account

If you do not have an AWS account, complete the following steps to create one.

#### To sign up for an AWS account

1. Open <https://portal.aws.amazon.com/billing/signup>.
2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, [assign administrative access to an administrative user](#), and use only the root user to perform [tasks that require root user access](#).

## Step 2: Setting up AWS Organizations

In order to use Amazon Q Developer Pro, you must also use Organizations. Organizations is an account management service that enables you to consolidate multiple AWS accounts into an organization that you create and centrally manage.

For more information, refer to [What is AWS Organizations](#) in *The AWS Organizations User Guide*.

## Step 3: Setting up IAM Identity Center with Amazon Q Developer

In order to use Amazon Q Professional, you must use IAM Identity Center. IAM Identity Center is the recommended AWS service for managing human user access to AWS resources. It is through IAM Identity Center that you will assign developers access to Amazon Q Professional. For more information, see [What is IAM Identity Center?](#) in *The AWS IAM Identity Center User Guide*.

When you subscribe to Amazon Q Developer, Amazon Q will connect to your [organization instance](#) of IAM Identity Center. You should not have an [account instance](#) of IAM Identity Center set up in the same organization. Doing so may cause the identity data in the account instance to be overwritten.

### Warning

For compatibility with Amazon Q, you cannot set up IAM Identity Center in an [opt-in Region](#).

## Understanding identity-aware sessions

When you set up Amazon Q Developer Pro, you will have the option to enable identity-aware sessions. An identity-aware session is a session in which you are authenticated with IAM Identity

Center while you are using the AWS console, or when you have your web browser open to an AWS documentation or marketing page.

In order to understand identity-aware sessions, it may be useful to consider the difference between authenticating with IAM and authenticating with IAM Identity Center.

The main difference between the two is that IAM users are granted long-term credentials to AWS resources, while users in IAM Identity Center have temporary credentials that are established each time the user signs-in to AWS.

When you log in to the AWS console as an IAM user, your session might be short. Your session may also last long enough to expire, in which case you will have to re-authenticate with your IAM credentials. In such cases, it may seem as if your IAM credentials are not intended for long-term use.

However, in a more general sense, IAM credentials are also used by non-human elements such as Amazon Elastic Compute Cloud instances and AWS Lambda functions, through roles. If you use the AWS CLI from your local computer, then you authenticate with access keys, stored in your local configuration files, that function as your IAM credentials. IAM credentials have many uses, and most of them take place in long-term contexts, with non-human principals.

IAM Identity Center credentials, however, are always used in a short-term context with a human user. They are completely identity-based.

In order to chat with Amazon Q in the AWS console, you must:

- Authenticate to the AWS management console, where you can use whatever AWS resources your administrator has made available to you. One of these resources could be the Amazon Q Developer administrative console. Another might be the Amazon S3 console. Depending on how your administrator has given you access, you might authenticate to the console with IAM or with IAM Identity Center.
- Authenticate with Amazon Q through your IAM Identity Center workforce identity. This gives you access to chat with Amazon Q inside the AWS console. It may be easier to conceptualize this distinction if you consider that Amazon Q is not really part of the AWS console. The Amazon Q chat panel happens to be available in the console, but Amazon Q is more of a companion that helps you navigate your AWS experience in general. It is not part of the console; it is in the console *with you*.

## Step 4: Creating an administrator for Amazon Q Developer

The following policies empower a user to take all administrative actions related to Amazon Q Developer, including access to the Amazon Q Developer console and all the steps in the procedures throughout the rest of this chapter. However, depending on the structure of your enterprise, you might choose to divide these powers over multiple personas.

You might need one of two policies to take administrative actions related to Amazon Q Developer and use the Amazon Q Developer console. For new administrators of Amazon Q Developer, use the following 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",
      "Action": [
        "sso-directory:GetUserPoolInfo"
      ],
      "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": [
    "*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "codeguru-security:UpdateAccountConfiguration"
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
```

```

    "iam:CreateServiceLinkedRole"
  ],
  "Resource": [
    "arn:aws:iam::*:role/AWSServiceRoleForQDeveloper"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "codewhisperer:UpdateProfile",
    "codewhisperer:ListProfiles",
    "codewhisperer:TagResource",
    "codewhisperer:UntagResource",
    "codewhisperer:ListTagsForResource",
    "codewhisperer:CreateProfile"
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "cloudwatch:GetMetricData",
    "cloudwatch:ListMetrics"
  ],
  "Resource": [
    "*"
  ]
}
]
}

```

For legacy Amazon CodeWhisperer profiles, the following policy will enable an IAM principal to administer a CodeWhisperer application.

### 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: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": [

```

```
"Resource": [
  "*"
],
},
{
  "Effect": "Allow",
  "Action": [
    "codeguru-security:UpdateAccountConfiguration"
  ],
  "Resource": [
    "*"
  ],
},
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole"
  ],
  "Resource": [
    "arn:aws:iam::*:role/AWSServiceRoleForCodeWhisperer"
  ],
},
{
  "Effect": "Allow",
  "Action": [
    "codewhisperer:UpdateProfile",
    "codewhisperer:ListProfiles",
    "codewhisperer:TagResource",
    "codewhisperer:UntagResource",
    "codewhisperer:ListTagsForResource",
    "codewhisperer:CreateProfile"
  ],
  "Resource": [
    "*"
  ],
},
{
  "Effect": "Allow",
  "Action": [
    "cloudwatch:GetMetricData",
    "cloudwatch:ListMetrics"
  ],
  "Resource": [
    "*"
  ],
}
```

```
    ]
  }
]
}
```

**Note**

If you are using Customizations, then your Amazon Q Developer administrator will require additional permissions. For more information, see [Prerequisites for customizations](#).

To learn more about IAM policies, see [Access Management](#) in the *IAM User Guide*.

## Step 5: Creating an administrator for Amazon Q Developer Subscriptions

The following policy empowers a user to view and manage subscriptions for Amazon Q Developer and use the Amazon Q Subscriptions 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": [
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:DisableAWSServiceAccess",
        "organizations:EnableAWSServiceAccess",
        "organizations:DescribeOrganization"
      ],
      "Resource": [
        "*"
      ]
    }
  ],
}
```

```

{
  "Effect": "Allow",
  "Action": [
    "sso:ListApplications",
    "sso:ListInstances",
    "sso:DescribeRegisteredRegions",
    "sso:GetSharedSsoConfiguration",
    "sso:DescribeInstance",
    "sso:CreateApplication",
    "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"
  ],
  "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/AWSServiceRoleForUserSubscriptions"  
    ]  
}  
]  
}
```

## Understanding tiers of service for Amazon Q Developer

### Understanding Amazon Q Developer Pro tier

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. As a user with an Amazon Q Developer Pro subscription, your limits are based on an individual user level in the Amazon Q Console, Q in the IDE, and Q in Code Catalyst.

Subscriptions are not available in the AWS Console Mobile App (ACMA), AWS Chatbot, or on AWS marketing pages. Interactions with Amazon Q through those channels are governed exclusively by free tier limits.

### Understanding the free tier

Amazon Q Developer offers a perpetual free tier with monthly limits available to anyone logged in as an AWS Identity and Access Management user or AWS Builder ID user. Exactly which features you can use depends on your interface, and on how you authenticate.

To see the capabilities and limits for Amazon Q Developer Pro and the Amazon Q Developer free tier, visit the [Amazon Q Developer pricing page](#).

## Subscriptions

1. Go to the Amazon Q landing page.
2. Choose **Getting Started**.

Generative AI

# Amazon Q

Your generative AI-powered assistant designed for work that can be tailored to your business

Boost employee productivity with Amazon Q, a generative AI-powered application that revolutionizes how you get work done.

**Start using Amazon Q features**  
[Pricing](#)  
[Getting started](#)

**Getting started**  
[What is Amazon Q?](#)  
[Setting up Amazon Q](#)

**Benefits and features**

### 3. Confirm your connection to IAM Identity Center.

Amazon Q < Amazon Q > Getting started

## Getting started with Amazon Q info

### Connect to Identity Center

**Amazon Q connected to IAM Identity Center**  
 Manage access to Amazon Q by assigning users and groups in IAM Identity Center to this application. [Learn more](#)

### Select a bundle to get started info

To learn more about bundle pricing, [view pricing documentation](#)

**Amazon Q Business Lite**

This plan includes access to Q&A feature in Amazon Q Business that enables users to ask questions and get answers from enterprise data sources. [Learn more](#)

Pricing available on April 30, 2024.

[Subscribe in Q Business](#)

**Amazon Q Business Pro**

This plan includes access to Amazon Q Business, that can help you solve problems, generate content, and find insights in data, and Amazon Quicksight, a generative BI assistant to help consume insights.

Pricing available on April 30, 2024.

Subscribe in [Q Business](#) or [QuickSight](#).

**Amazon Q Developer Pro**

This plan includes access to Amazon Q Developer, which can transform the way that developers and IT professionals build, optimize, and operate applications. [Learn more](#)

Includes:  
 Amazon Q in the IDE, Amazon Q in the AWS Console, and Amazon CodeCatalyst

Pricing available on April 30, 2024.

[Subscribe](#)

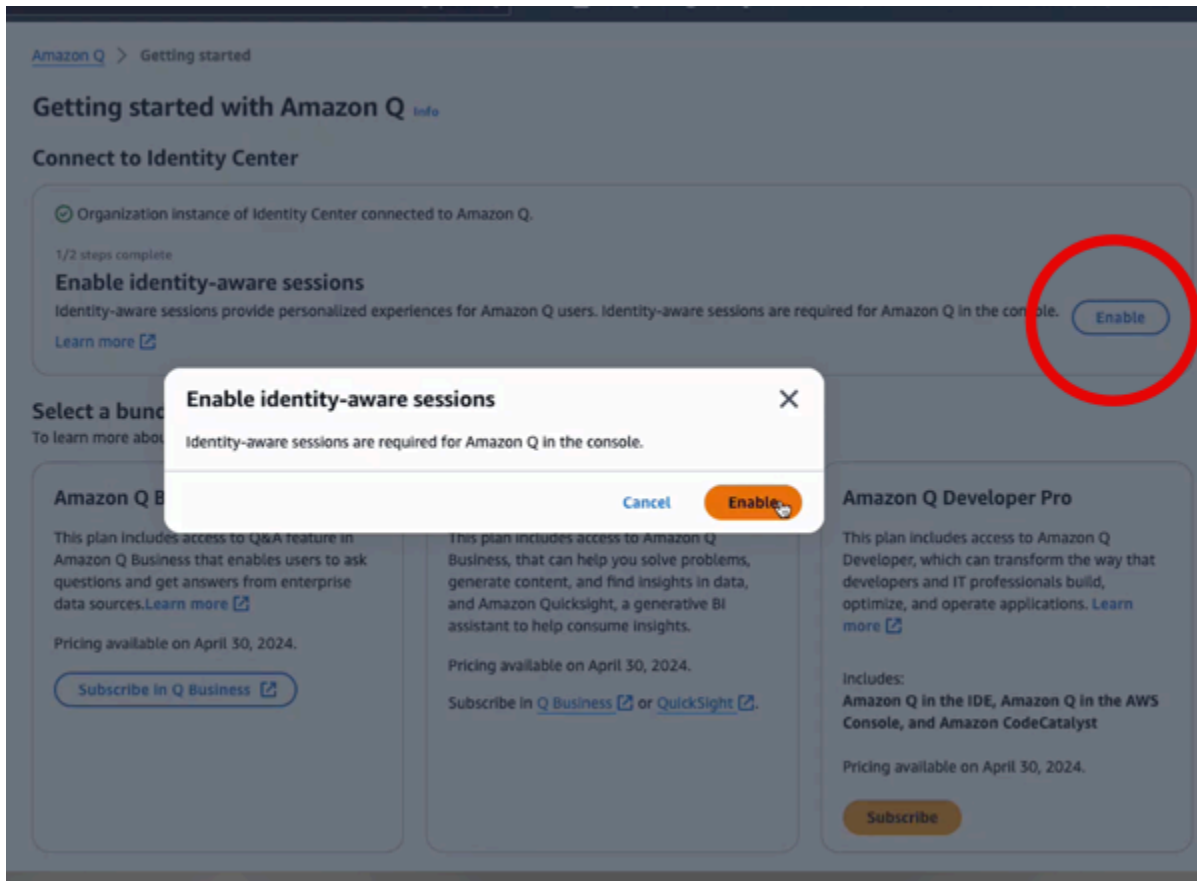
#### Warning

If, as AWS advises, you are planning to connect Amazon Q Developer Pro with an organization instance of IAM Identity Center, but you already have an [account instance](#) set up, you should delete it before continuing. For more information, see [Setting up IAM Identity Center with Amazon Q Developer](#).

If you do not already have an organization instance connected, you should create one now.

The screenshot shows the Amazon Q Developer console interface. On the left is a navigation sidebar with 'Amazon Q' at the top, followed by 'Get started', 'Subscriptions', and 'Settings'. Below these are links for 'Q Business Pro', 'Q Developer Pro', and 'Documentation'. The main content area is titled 'Get started with Amazon Q' and includes a sub-section 'Connect to IAM Identity Center'. Under this section, there is a heading 'Connect Amazon Q to IAM Identity Center' and an 'AWS Region' dropdown set to 'US West (Oregon) | us-west-2'. A checkbox for 'Specify tags for IAM Identity Center' is present. Below this, the 'IAM Identity Center' section explains that an instance must be created. Two buttons are shown: 'Create organization instance' (which is circled in red) and 'Create account instance'. A blue information banner at the bottom states: 'Amazon Q requires an organization instance. To connect this application to IAM Identity Center, cancel setup, and choose Create organization instance.'

4. (optional) Enable identity-aware sessions. This will enable your developers to use Amazon Q in the AWS console, and on the AWS marketing and documentation pages. Your session becomes identity-aware when your login as an IAM principal becomes associated with your identity as registered in IAM Identity Center.



5. Under Amazon Q Developer Pro, choose **Subscribe**.

**Note**

Read about how [access tiers](#) differ.

6. (optional) Share your profile with your managed accounts.


You have the option of sharing your Amazon Q settings with your member accounts. These settings relate to which features of Amazon Q may be used.

## Prerequisite: Create managed application ✕

The first time you subscribe a user we will create a managed application instance on your behalf. You can use a managed app instance to assign specific settings for Q Developer Pro features.

By default, this managed app will be available to all associated member accounts. You can turn this setting off here, or change it later in the **Settings** page.

Share settings profile with member accounts

 You can proceed to subscribe users and groups to Q Developer Pro, or skip to create only the managed application and configure settings.

Cancel

Create only

Create and subscribe to Q Developer Pro

### 7. Subscribe (or not).

If you are not ready to subscribe, you can choose **Create only**, which will save the Amazon Q Developer managed application that you have been setting up with this process.

If you are ready to subscribe, choose **Create and subscribe to Amazon Q Developer Pro**.

### 8. On the Amazon Q subscriptions page, select the users and groups that should be end users of Amazon Q Developer, and choose **Assign**.

**Note**

When you add a group, as defined in IAM Identity Center, to a subscription, or add a user to a subscribed group, it may take up to 24 hours for that user, or users in that group, to be subscribed.

## Assign users and groups



×

People
Alice Jackson alice.jackson@amazon.com
Alice Kingman alice.kingman@amazon.com
Alice Collins alice.collins@amazon.com

Groups
Accounting Group description
A-Team Group description
Analytics Group description

Cancel

Assign

9. Open the Amazon Q Developer console.

The screenshot displays the Amazon Q Developer console interface. At the top, a green notification banner states: "You have successfully created a Q for Developers subscription for 3 users and 3 groups. Visit Q Developer console to learn more about features." A red circle highlights the "Q Developer console" button in the notification. Below the notification, the breadcrumb "Amazon Q > Q Developer Pro" is visible. The main heading is "Q Developer Pro Info", followed by a "Learn more" link. A section titled "Complete your setup in the Amazon Q Developer console" contains a paragraph of placeholder text and a red circle around the "Amazon Q for Developers console" button. Below this, the "Included in the bundle" section features three cards: "Amazon Q in AWS Console", "Amazon Q in IDE", and "Amazon Q in CodeCatalyst". Each card includes a "Get started" section with a list of steps and a "Learn more" button. The "Amazon Q in CodeCatalyst" card also includes an "Open CodeCatalyst" button.

## 10. Manage your Amazon Q Developer settings.

On this page, you can select the features of Amazon Q Developer that will be available to your users.

### Note

Currently, the only elements you can change here are the encryption key type and the tags.

[Amazon Q Developer](#) > Settings

## Settings Info

Delete profile

### Q Developer account details Edit

#### Enabled Amazon Q Developer features

##### Q in the IDE

Amazon Q code transformation  
Amazon Q task assistance  
Amazon Q completions  
Amazon Q feature dev  
Amazon Q conversations  
Amazon Q analysis

##### Q in the AWS Console

Amazon Q assistant  
Amazon Q troubleshooting  
Amazon Q actions

##### Amazon CodeCatalyst

Amazon Q feature dev  
Pull request comment summary

##### Q ChatBots

AWS console mobile application  
AWS chatbot in Slack and Teams

#### Include suggestions with code references

True

#### Encryption key

Default AWS managed key

#### Region

US East (N. Virginia)

#### Start URL

Give this URL to your users. This is required for them to access Amazon Q Developer in their IDEs. [Learn more](#)

<https://b-5632de3745.awsapps.com/start>

### Member account settings Edit

#### Q Developer managed application and settings profile

To share Q Developer Settings Profile set in the Management account with your Member accounts, ensure this is turned on.

On (default)

### Tags Info

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key

Value

## 11. Copy the Start URL.

Your administrative setup is complete.

Pass the Start URL along to your developers, and direct them to [Using Amazon Q Developer in the IDE](#).

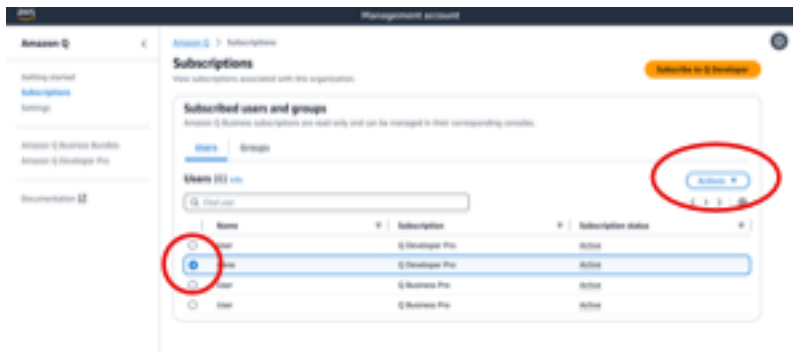
## Unsubscribing

Administrators can unsubscribe users from Amazon Q Developer Pro subscriptions through the subscriptions page in the console by de-selecting user names in the Amazon Q console.



To unsubscribe a user:

1. Go to the Subscriptions page of the Amazon Q Developer console.
2. Select the user.
3. Under Actions, choose **Unsubscribe**.
4. A pop-up window will ask you to confirm. Choose **Unsubscribe**.



However, the process varies based on the creator of the subscription:

- *Member account administrators* can only view and unsubscribe users subscribed within their account, and only if they created that specific subscription.
- *Organization management account administrators* can unsubscribe users if they created that subscription. They can also view other accounts the user is subscribed through (by selecting **Share settings profile with member accounts** on the **Settings** page), allowing them to coordinate with respective member account admins for unsubscription. Alternatively, they can log in as a member account administrator and unsubscribe the user directly.

## Troubleshooting subscriptions

You may encounter the following error message when trying to access Amazon Q Developer in the AWS Management Console and on the AWS Documentation website:

An error occurred when attempting to access Amazon Q  
Review the Amazon Q in console common errors documentation for potential solutions.

This could be due to one of the following reasons:

- **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 [Add IAM permissions](#). After you obtain the necessary permissions, reload the console page to access Amazon Q.

# 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, and also about your AWS resources and costs. 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, in addition to asking questions, you can use Amazon Q to contact AWS Support directly and to diagnose common console errors.

## Topics

- [Add IAM permissions](#)
- [Chat about AWS](#)
- [Diagnose common errors in the console with Amazon Q Developer](#)
- [Use Amazon Q Developer to chat with AWS Support](#)
- [Chat with Amazon Q Developer in AWS Chatbot](#)

## Add IAM permissions

To use Amazon Q Developer features on AWS apps and websites, you must attach the appropriate AWS Identity and Access Management (IAM) permissions. Following is an example policy you can use to access most Amazon Q features on AWS apps and websites:

### Note

The availability of Amazon Q features depends on the environment where you use Amazon Q. See the topic for the feature you want to use for more information about where you can access it and the specific permissions you need to use it.

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

```

    "Effect": "Allow",
    "Action": [
      "q:SendMessage",
      "q:StartConversation",
      "q:GetConversation",
      "q:ListConversations",
      "q:GetIdentityMetaData",
      "q:StartTroubleshootingAnalysis",
      "q:GetTroubleshootingResults",
      "q:StartTroubleshootingResolutionExplanation",
      "q:UpdateTroubleshootingCommandResult",
      "q:PassRequest"
    ],
    "Resource": [
      "*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "ce:GetCostAndUsage"
    ],
    "Resource": [
      "*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "sts:setContext"
    ],
    "Resource": [
      "arn:aws:sts::*:self"
    ]
  }
]
}

```

Alternatively, you can add permissions for the individual Amazon Q features you want to use. For examples of policies you can create to allow access to specific features, see [Identity-based policy examples for Amazon Q Developer](#).

For more information about configuring IAM permissions with Amazon Q, see [Manage access to Amazon Q Developer with policies](#).

For more information about what these permissions do, see the [Amazon Q Developer permissions reference](#).

## 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 [Creating 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 [Creating 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*.

## Chat about AWS

Chat with Amazon Q in the AWS Management Console, AWS Console Mobile Application, AWS website, and AWS Documentation website 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 also generates short scripts or code snippets to help you get started using the AWS SDKs and AWS CLI.

### Topics

- [Working with Amazon Q on AWS websites](#)
- [Example questions](#)

- [Chat about your resources](#)
- [Chat about your costs](#)

## 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 context of 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?

## Chat about your resources

Asking Amazon Q about your AWS resources is in preview, and is subject to change.

You can ask Amazon Q about your AWS resources in the AWS Management Console. 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 6 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, you need the following permissions:

- `q:SendMessage` to chat with Amazon Q
- `q:StartConversation` to chat with Amazon Q
- `q:GetConversation` to chat with Amazon Q
- `q:ListConversations` to chat with Amazon Q
- `q:PassRequest` to allow Amazon Q to call the APIs required to view your resources
- Permissions to perform the actions required to view your resources

Your IAM identity must have permissions to perform the actions required to access 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 perform an action that you don't have permission to perform yourself.

For an example IAM policy that grants the necessary permissions, see [Allow Amazon Q to perform actions on your behalf](#). For more information about the `q:PassRequest` action, see [Using q:PassRequest](#).



### 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>

## Chat about your costs

Amazon Q's cost analysis capability is in preview and can make mistakes. Please verify your cost data with AWS Cost Explorer. Use the thumb icon in Amazon Q to provide feedback and help us improve.

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.

### Prerequisites

#### Permissions

To chat about your costs, you need the following permissions:

- `q:SendMessage` to chat with Amazon Q
- `q:StartConversation` to chat with Amazon Q
- `q:GetConversation` to chat with Amazon Q
- `q:ListConversations` to chat with Amazon Q
- `q:PassRequest` to allow Amazon Q to call the APIs required to view your resources
- `ce:GetCostAndUsage` to get Cost Explorer data

For more information about the `q:PassRequest` action, see [Using q:PassRequest](#).

#### 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?

## Diagnose 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

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 [Use 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.

You might need additional permissions to access this feature. For more information, 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.

## Use 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 permissions to start conversations with Amazon Q. For more information, see [Add IAM permissions](#).
- 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	<div style="border: 1px solid #ccc; border-radius: 15px; padding: 5px; display: inline-block;">  <b>Attach Files</b> </div> <p>You can attach up to 3 files. Each file can be up to 5 MB.</p>	
<div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid #ccc; border-radius: 15px; padding: 5px 15px; background-color: #f0f0f0;">Cancel</div> <div style="border: 1px solid #ccc; border-radius: 15px; padding: 5px 15px; background-color: #e0e0e0;">Edit</div> <div style="border: 1px solid #ccc; border-radius: 15px; padding: 5px 15px; background-color: #ff9900; color: white;">Submit</div> </div>		

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**.

## Chat 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 [AmazonQFullAccess](#) managed policy, and then add the policy as a channel guardrail.

1. Add the AmazonQFullAccess 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 **AmazonQFullAccess** in the search.
  - f. Select **AmazonQFullAccess**.
  - g. Choose **Add permissions**.
2. Add the AmazonQFullAccess 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 **AmazonQFullAccess** in the search.
    - f. Select **AmazonQFullAccess**.
    - 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 the following capabilities:

- Answering questions about building at AWS
- Answering general questions about software development and code
- Generating code
- Updating your code language version
- Developing new code features
- Scanning your code for security vulnerabilities
- Explaining, refactoring, or optimizing code snippets

Amazon Q is currently available as an extension in Visual Studio Code and in JetBrains. Inline suggestions from Amazon Q is additionally available in the AWS Toolkit for Visual Studio. To install Amazon Q in your IDE, see [Install Amazon Q Developer in your IDE](#).

## Topics

- [Install Amazon Q Developer in your IDE](#)
- [Chat with Amazon Q Developer about code](#)
- [Get inline suggestions with Amazon Q](#)
- [Upgrade language versions with the Amazon Q Developer Agent for Code Transformation](#)
- [Develop software with the Amazon Q Developer Agent for software development](#)
- [Scan your code with Amazon Q](#)
- [Supported languages for Amazon Q Developer in the IDE](#)

## Install Amazon Q Developer in your IDE

To set up Amazon Q Developer in your integrated development environment (IDE), complete the following steps. After installing the Amazon Q 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 plugin for your IDE:

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

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

## Steps

- [Prerequisite: Choose a supported version of your IDE](#)
- [Authenticating to VS Code and JetBrains](#)
- [Authenticating with AWS Builder ID](#)
- [Authenticating to Visual Studio](#)
- [If you are using an IAM principal in your AWS console](#)
- [Understanding AWS coding environments](#)

## Prerequisite: Choose a supported version of your IDE

Amazon Q is available as an extension in Visual Studio Code and a plugin in JetBrains. Amazon Q is also available in the AWS Toolkit for Visual Studio.

- The minimum version of Visual Studio Code supported by Amazon Q is 1.85.0.
- 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 supported by Amazon Q is 1.68.

## Authenticating to VS Code and JetBrains

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

1. Install the Amazon Q IDE extension in your Visual Studio Code or JetBrains IDE:

- [Download Amazon Q in Visual Studio Code](#)
  - [Download Amazon Q in JetBrains IDEs](#)
2. Choose the Amazon Q icon from the sidebar in your IDE.
  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 [account 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. Begin using Amazon Q in your IDE.

## Authenticating with AWS 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 IDE extension or plugin in your Visual Studio Code or JetBrains IDE:
  - [Download Amazon Q in Visual Studio Code](#)
  - [Download Amazon Q in JetBrains IDEs](#)

2. Choose the Amazon Q icon in your IDE.

In VS Code and JetBrains, 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. Begin using Amazon Q in your IDE.

## Authenticating to Visual Studio

Refer to [the AWS Toolkit for Visual Studio documentation](#) for setup instructions.

## If you are using an IAM principal in your AWS console

Depending on how you use AWS, you may be accustomed to using your IAM credentials to log 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.

## Understanding AWS coding environments

If you're developing software, then the way to get the most out of Amazon Q Developer is to subscribe to Amazon Q Developer Pro, and to use Amazon Q through the Amazon Q extension in Visual Studio, Visual Studio Code, or a JetBrains IDE. Most of the features described in this chapter pertain to that setup.

However, there are many other aspects of your AWS experience in which you might end up writing code. For example, you might write code directly in the AWS Lambda interface. In some such cases, inline code suggestions from Amazon Q can be available to you.

Environments in which you may enable code suggestions include:

- Amazon SageMaker Studio Classic
- JupyterLab
- Amazon EMR Studio
- AWS Glue Studio
- AWS Lambda
- AWS Cloud9

To enable code suggestions in coding environments within the AWS console, your administrator must assign you [the appropriate permissions](#).

## Chat 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](#)
- [Explain and update code with Amazon Q Developer](#)

## 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 ask a question, choose the Amazon Q icon from the navigation bar in your IDE, and then choose **Open Chat Panel**. You can enter your question in the text bar.

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 inform future responses. You can ask follow up questions or refer to previous questions and responses throughout the duration of your session. You can start a new conversation with Amazon Q by opening 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:

- **/transform** - Use this command to update the code language version of an entire project. For more information, see [Upgrade language versions with the Amazon Q Developer Agent for Code Transformation](#).
- **/dev** - Use this command to get an implementation plan to develop a feature with Amazon Q. For more information, see [Develop software with the Amazon Q Developer Agent for software development](#).
- **/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.

## Explain and update code with Amazon Q Developer

Amazon Q Developer can explain and update specific lines of code in your integrated development environment (IDE). To get your code updated, 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 **Send to 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.

## Get inline suggestions with Amazon Q

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.

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 [Customize suggestions](#).



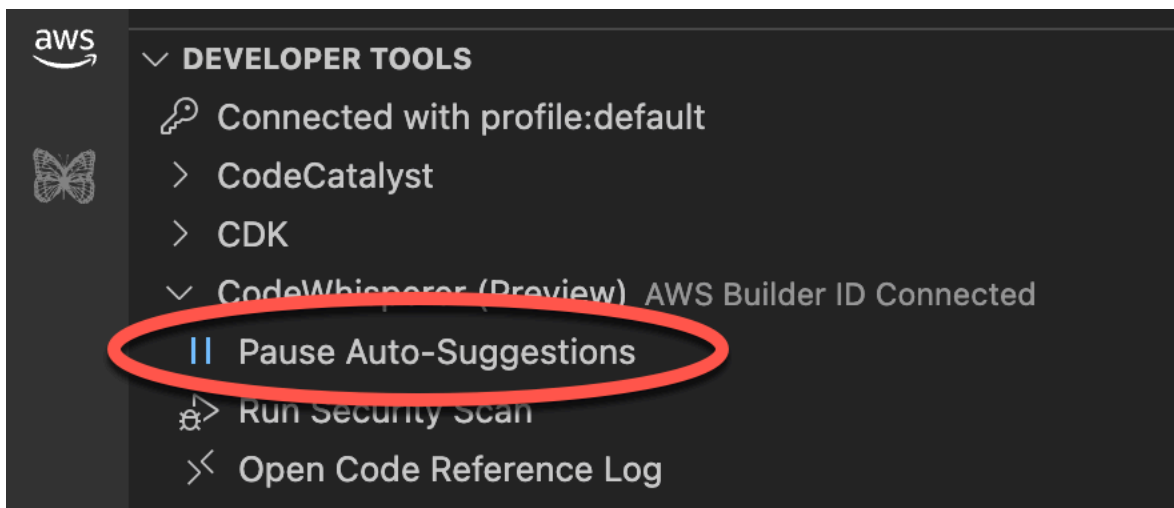
## Topics

- [Pausing suggestions with Amazon Q](#)
- [Amazon Q code completion in action](#)
- [Get coding suggestions from Amazon Q within other AWS services](#)
- [Use shortcut keys](#)
- [Code references in IDEs](#)
- [Code examples](#)

## Pausing suggestions with Amazon Q

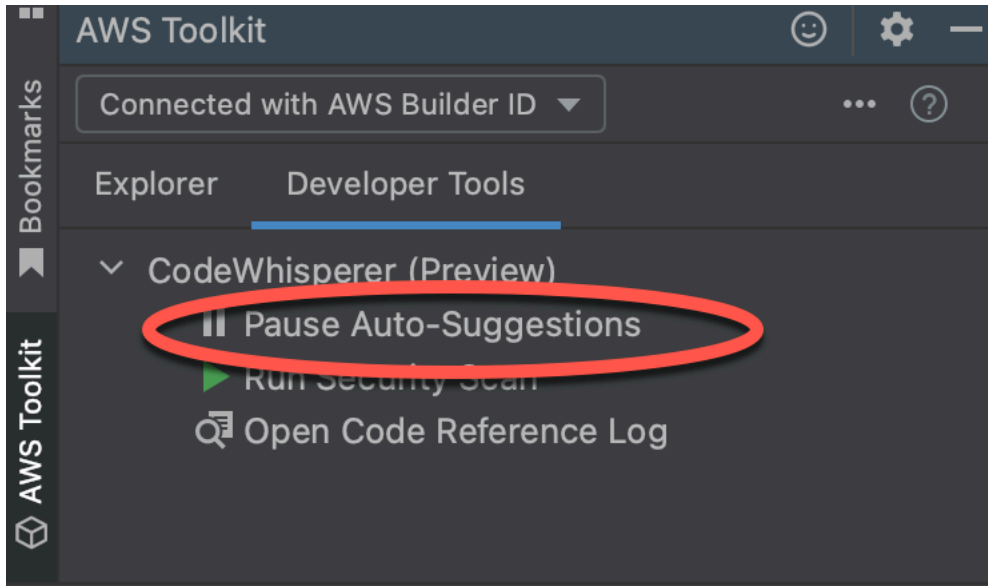
Choose your IDE to see steps for pausing and resuming inline code suggestions in Amazon Q.

### Visual Studio Code



1. In VS Code, choose the Amazon Q logo from the left sidebar.
2. Near the bottom of the VS Code window expand the **Developer Tools** section.
3. Expand the Amazon Q section.
4. Choose **Pause Auto-Suggestions** or **Resume Auto-Suggestions**.

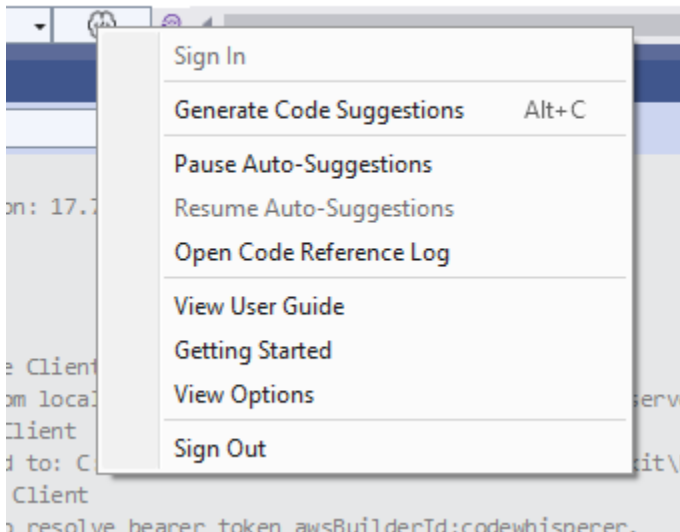
## JetBrains



1. In JetBrains, choose the Amazon Q logo from the left sidebar.
2. In the Amazon Q window, select the **Developer Tools** tab.
3. Expand the Amazon Q section.
4. 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**



## 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 DEFAULT REGION = '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)
```



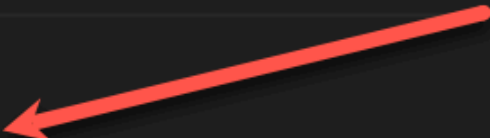
The developer has merely written the name of the function that will create the bucket. But based on that (and the context), Amazon Q offers a full function, complete with try/except clauses.

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):
        self.s3 = session.client('s3', region_name=DEFAULT_REGION)
        self.dynamodb = session.client('dynamodb', region_name=DEFAULT_REGION)
        self.s3_resource = session.resource('s3', region_name=DEFAULT_REGION)
        self.dynamodb_resource = session.resource('dynamodb', region_name=DEFAULT_REGION)

    def tearDown(self):
        self.s3.delete_bucket(Bucket=TEST_BUCKET_NAME)
        self.dynamodb.delete_table(TableName=TEST_TABLE_NAME)

    def test_create_s3_bucket(self):
        self.assertTrue(create_s3_bucket(TEST_BUCKET_NAME, DEFAULT_REGION))

    def test_create_dynamodb_table(self):
        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.

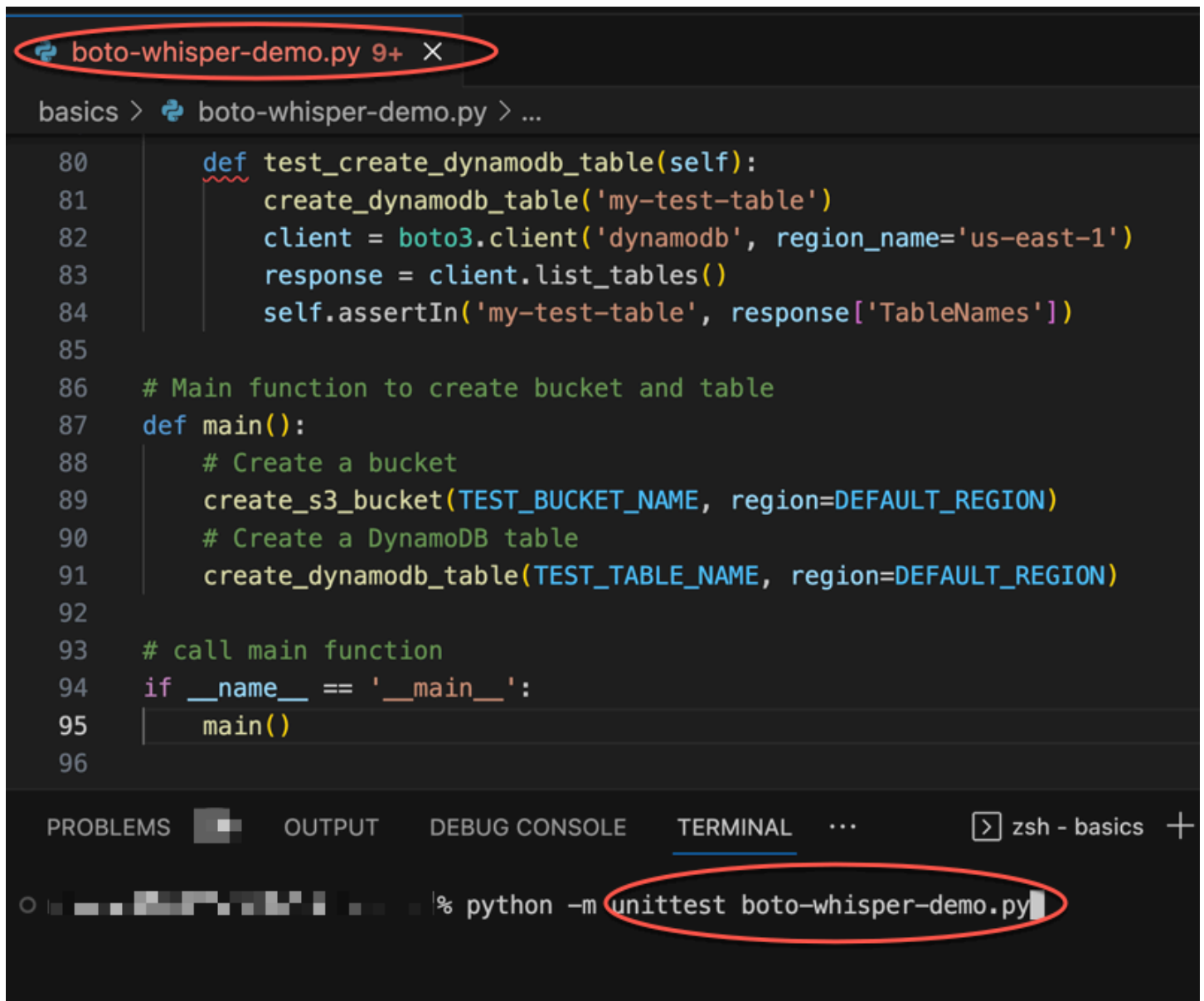
Based only on a comment and the context, Amazon Q supplies the entire main function.

```
# 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.





The screenshot shows an IDE window titled "boto-whisper-demo.py 9+ X". The code editor displays Python code for a test function and a main function. The terminal window at the bottom shows the command `python -m unittest boto-whisper-demo.py` being entered.

```
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
```

## Get coding suggestions from Amazon Q within other AWS services

The following sections describe how to set up Amazon Q inline code suggestions within AWS services that provide their own coding environments.

### Note

If you are using Amazon Q, on behalf of your organization, 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 [Setting up 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

This page describes how to set up and activate Amazon Q for Amazon SageMaker Studio. 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 SageMaker Studio.

### 1. **Set up Amazon SageMaker prerequisites.**

The prerequisites for using SageMaker include creating an AWS account and creating an administrative user.


For more information, see [Set up Amazon SageMaker prerequisites](#) in the *Amazon SageMaker User Guide*.

### 2. **Set up a Amazon SageMaker Domain.**

To use Amazon SageMaker Studio, you must complete the Amazon SageMaker Domain onboarding process using the SageMaker console or the AWS CLI. For more information, see [Onboard to Amazon SageMaker Domain](#) in the *Amazon SageMaker User Guide*.

### 3. **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**

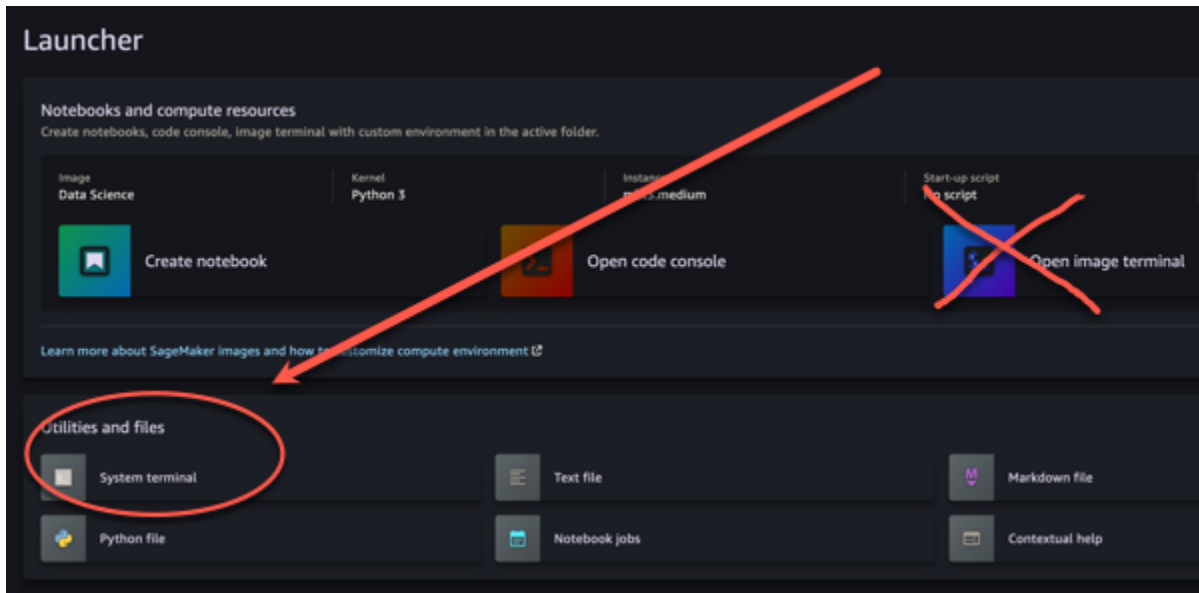
The `codewhisperer` prefix is a legacy name from a service that merged with Amazon Q Developer. For more information, see [Amazon Q Developer rename - Summary of changes](#).

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

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

**4. Enable the Amazon Q extension in your SageMaker Studio domain.**

Open the Launcher tab. Then, in the system terminal (not the image terminal) inside SageMaker Studio, run the following commands.



```
conda activate studio
pip install amazon-codewhisperer-jupyterlab-ext~=1.0
jupyter server extension enable amazon_codewhisperer_jupyterlab_ext
conda deactivate
restart-jupyter-server
```

For more information about this step and the next, see [Use the Amazon SageMaker Studio Launcher](#) in the Amazon SageMaker Developer Guide.

## 5. Open a new notebook.

### **Note**

Code completions with Amazon Q only appear in code cells. They do not appear in markdown cells.

Now you should be ready to code with Amazon Q in SageMaker Studio. (You may need to refresh your browser first.)

For keyboard shortcuts, see [Use shortcut keys](#).

## 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 $\geq 4.0$

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

```
# JupyterLab 4  
pip install amazon-codewhisperer-jupyterlab-ext
```

### Installation using pip for Jupyter Lab version $\geq 3.6$ and $< 4.0$

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

```
# JupyterLab 3  
pip install amazon-codewhisperer-jupyterlab-ext~=1.0  
jupyter server extension enable amazon_codewhisperer_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 [Use shortcut keys](#).

The screenshot shows a JupyterLab notebook with the following content:

```
clinical['patient_id'] = clinical['patient_id'].astype(object)
```

Create a TimeStamp for each Record

```
[ ]: import time

current_time_sec = int(round(time.time()))
# append EventTime feature
clinical['EventTime'] = pd.Series([current_time_sec]*len(clinical), dtype="float64")
```

Check data types for each column  
 Suggestion 2 of 2 from CodeWhisperer

```
[ ]: def cast_object_to_string(data_frame):
    for column in data_frame.columns:
        if data_frame[column].dtype == 'object':
            data_frame[column] = data_frame[column].astype(str)
    return data_frame

# cast object dtype to string. The SageMaker Feature Store Python SDK will then map the string dtype to String feature type.
cast_object_to_string(clinical)
```

```
[ ]: clinical.dtypes
```

## 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 [Use 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 **Enable Amazon Q**.

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*.

## Use shortcut keys

Choose the integrated development environment (IDE) where you are developing code to see keyboard shortcuts for common actions you take while getting inline suggestions from Amazon Q

### Toolkit for Visual Studio

Action	Keyboard shortcut
Manually trigger Amazon Q <code>AWSToolkit.CodeWhisperer.GetSuggestion</code> in the keybindings	Alt + C
Accept a recommendation	Tab
Next recommendation	Alt + .

Action	Keyboard shortcut
Edit.NextSuggestion in the keybindings	
Previous recommendation Edit.PreviousSuggestion in the keybindings	Alt + ,
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.

#### AWS Toolkit for Visual Studio Code

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

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.

## AWS Toolkit for JetBrains

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

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

## Amazon SageMaker

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

Action	Keyboard shortcut
Reject a recommendation	ESC

## JupyterLab

Action	Keyboard shortcut
Manually trigger 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

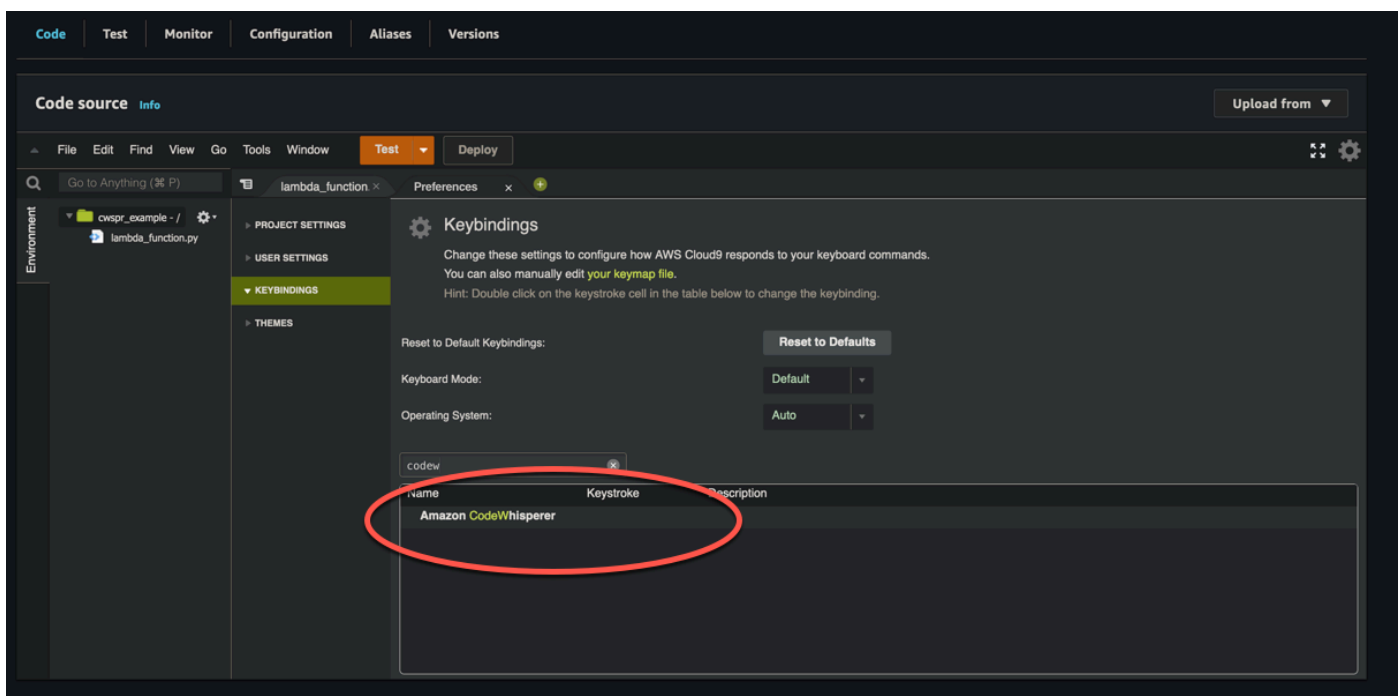
Action	Keyboard shortcut
Manually trigger Amazon Q	MacOS: Option + C Windows: Alt + C
Accept a recommendation	Tab
Next recommendation	Down arrow
Previous recommendation	Up arrow
Reject a recommendation	ESC

## Lambda

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

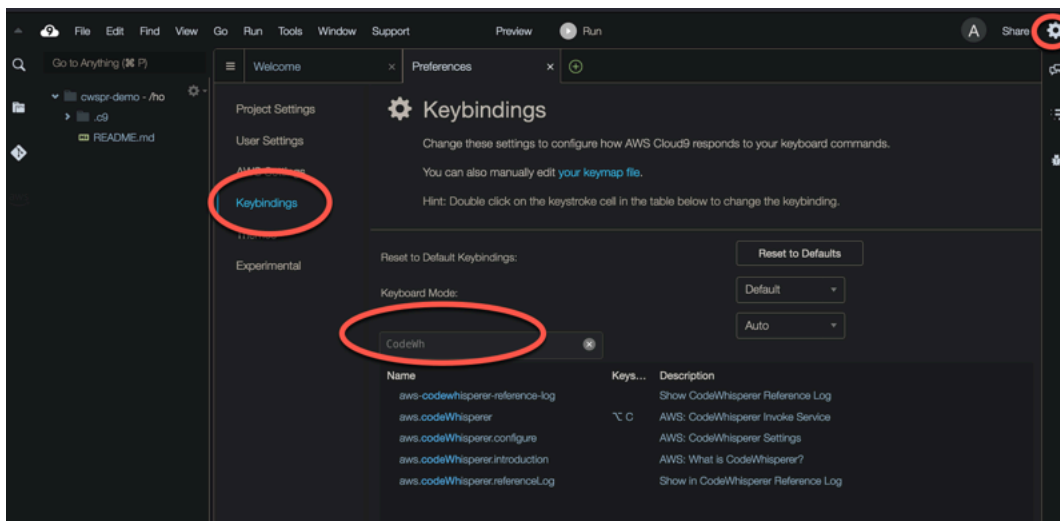




## AWS Cloud9

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



# Code references in IDEs

## Topics

- [Viewing code references](#)
- [Turning code references off and on](#)
- [Opting out of code with references](#)

## Viewing 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.

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.

This chapter explains how to view the code references.

### Toolkit for Visual Studio

When Amazon Q suggests code that contains a reference in 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'.
```

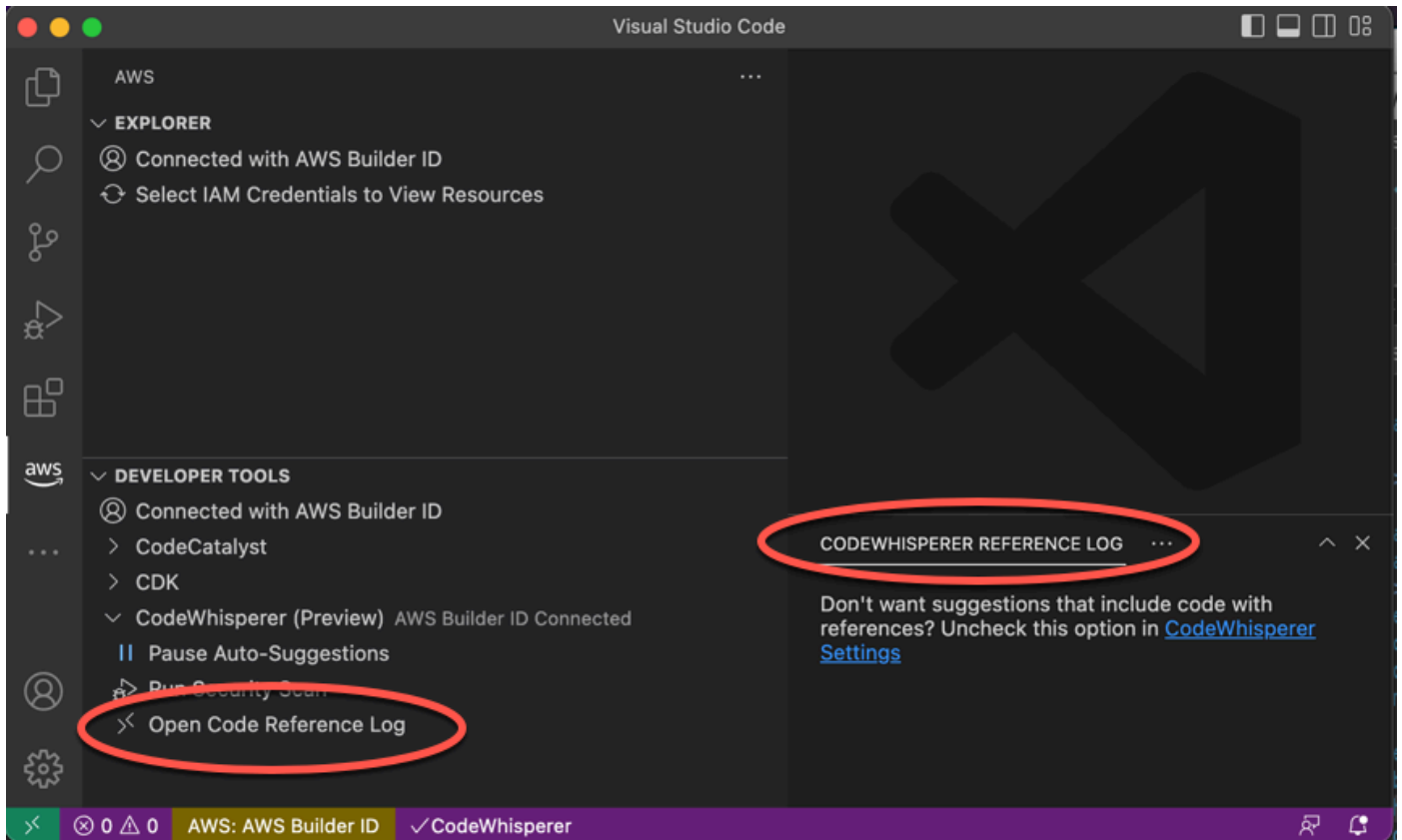
All accepted suggestions that contain references are captured in the reference log.

To access the reference log, choose the Amazon Q 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

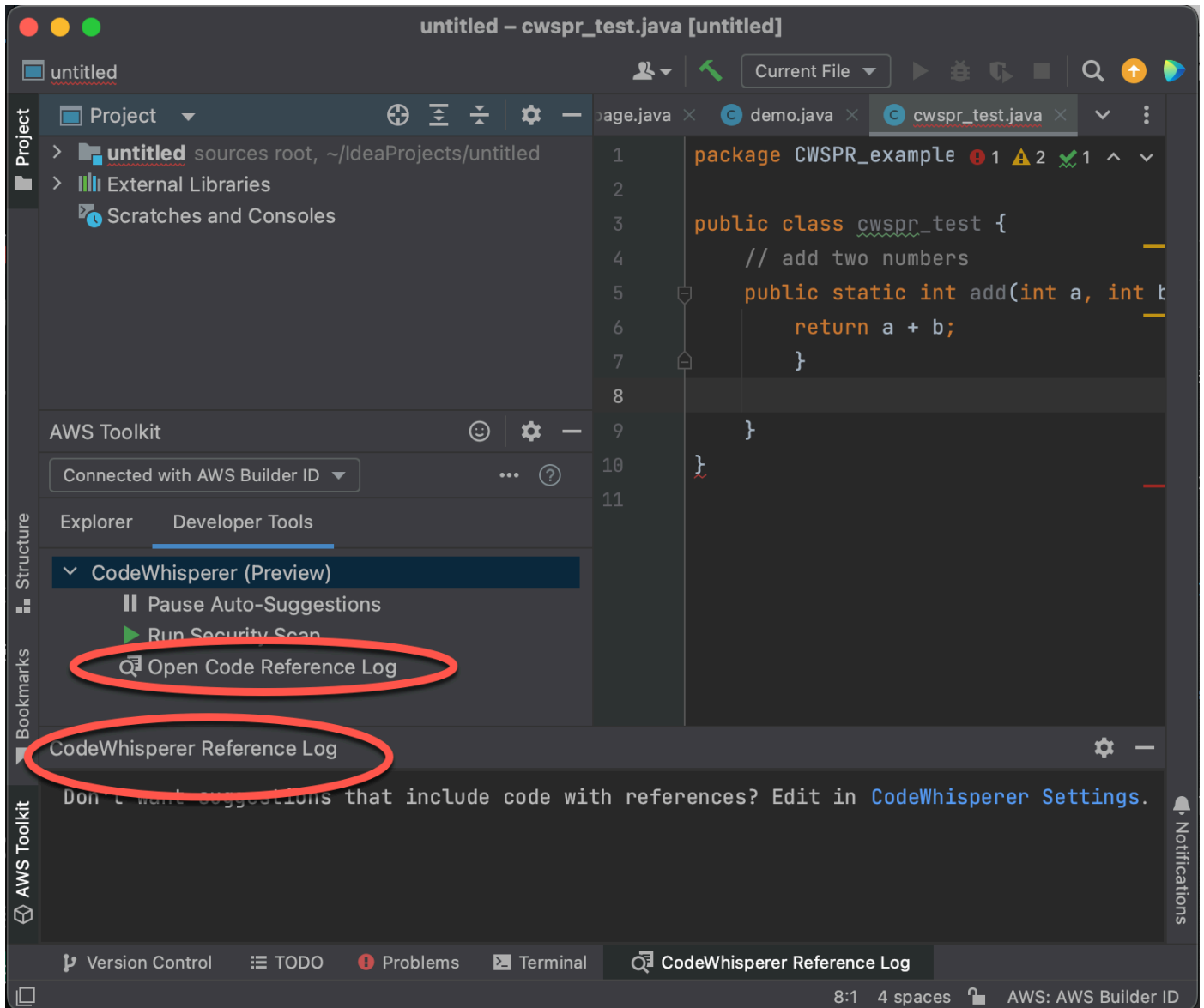




## AWS Toolkit for JetBrains

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

1. Make sure you are using the latest version of both JetBrains and the Amazon Q extension.
2. On the left side of the JetBrains window, choose the Amazon Q logo.
3. In the Amazon Q extension panel, select the **Developer Tools** tab.
4. In the Amazon Q dropdown, 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**.

## Turning code references off and on

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

This section explains how to use the code reference options.

## AWS Toolkit for Visual Studio Code

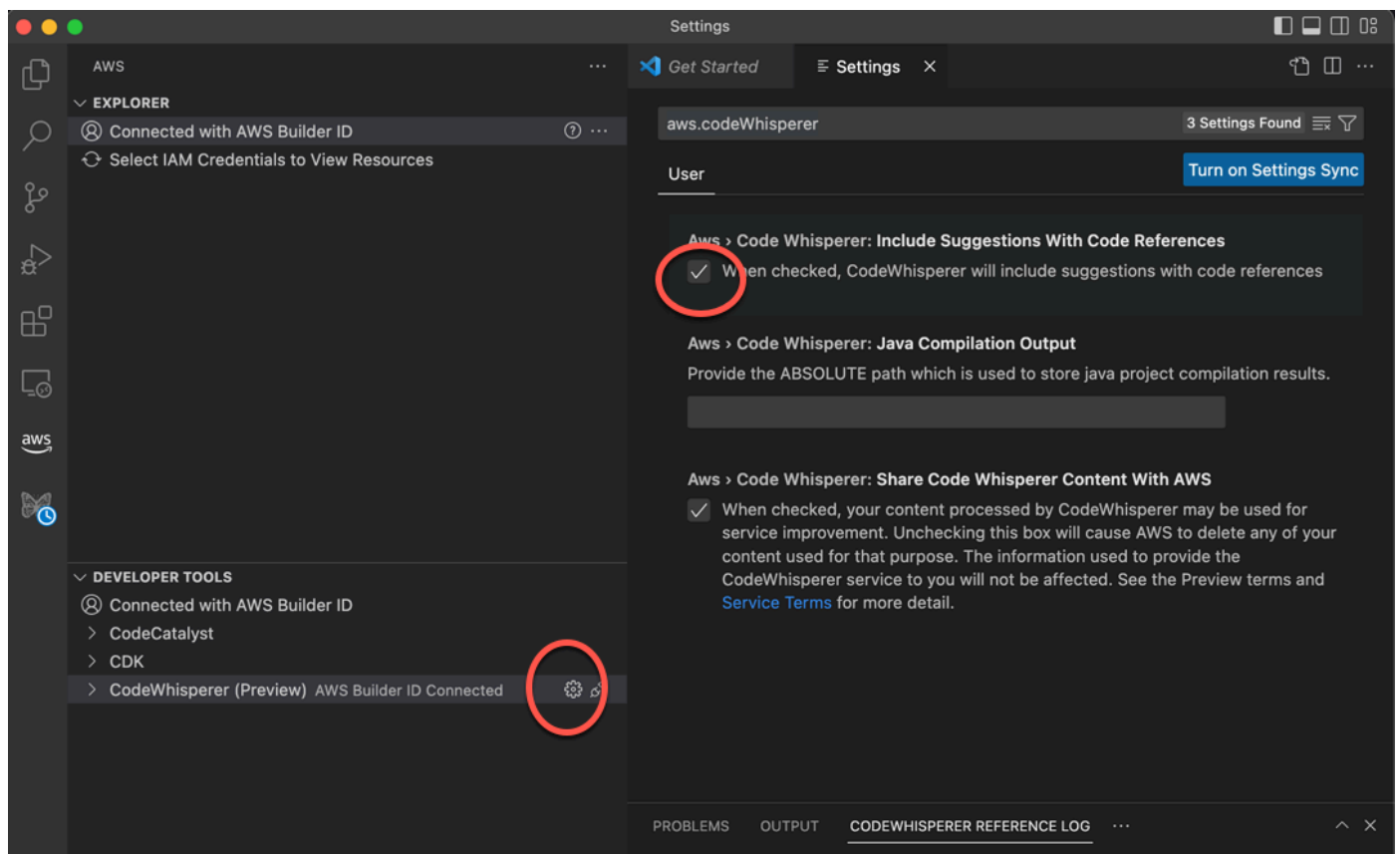
When you use Amazon Q with VS Code, code references are on by default.

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

1. Make sure you are using the latest version of both VS Code and Amazon Q extension.
2. In VS Code, choose the AWS logo from the left side of the window.
3. Open the **Developer Tools** dropdown menu.
4. Open the **Developer Tools** dropdown menu.
5. Next to the **Amazon Q** option, choose the gear icon.

On the side of the VS Code window, the **Settings** tab will open, with the options related to Amazon Q displayed.

6. Select or deselect the box under **Include Suggestions with Code References**.



## AWS Toolkit for JetBrains

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

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

1. Make sure you are using the latest version of both JetBrains and the Amazon Q extension.
2. In JetBrains, open **Preferences**.
3. In the **Preferences** window, under **Tools**, under **AWS**, select Amazon Q.
4. In the Amazon Q panel on the right, select or deselect the box labeled **Include suggestions with code references**.

## AWS Cloud 9

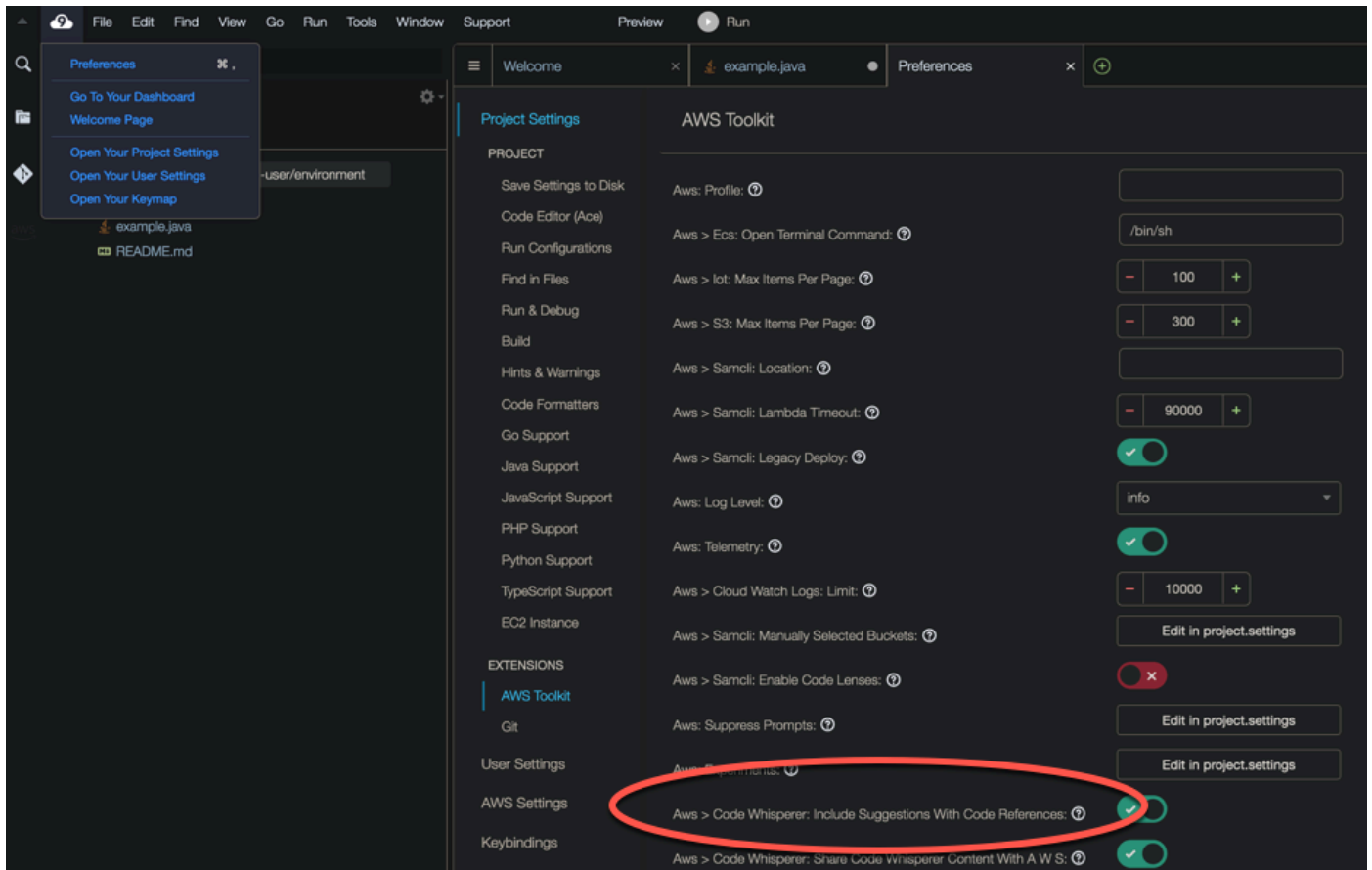
When you use Amazon Q with AWS Cloud 9, code references are on by default.

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.

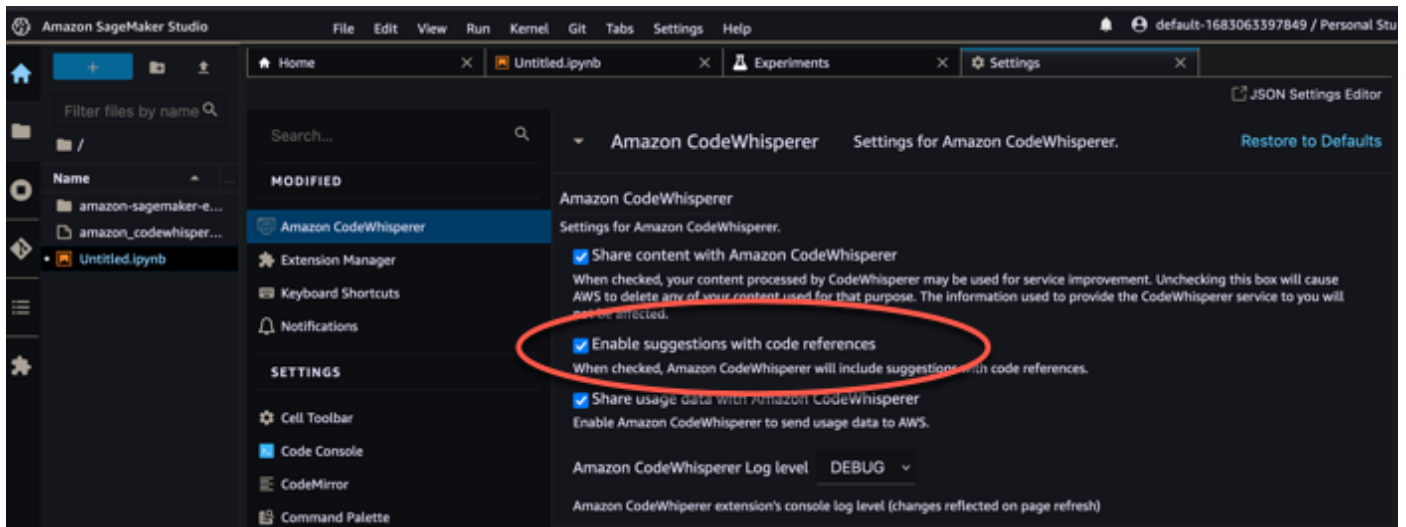
## SageMaker Studio

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**.



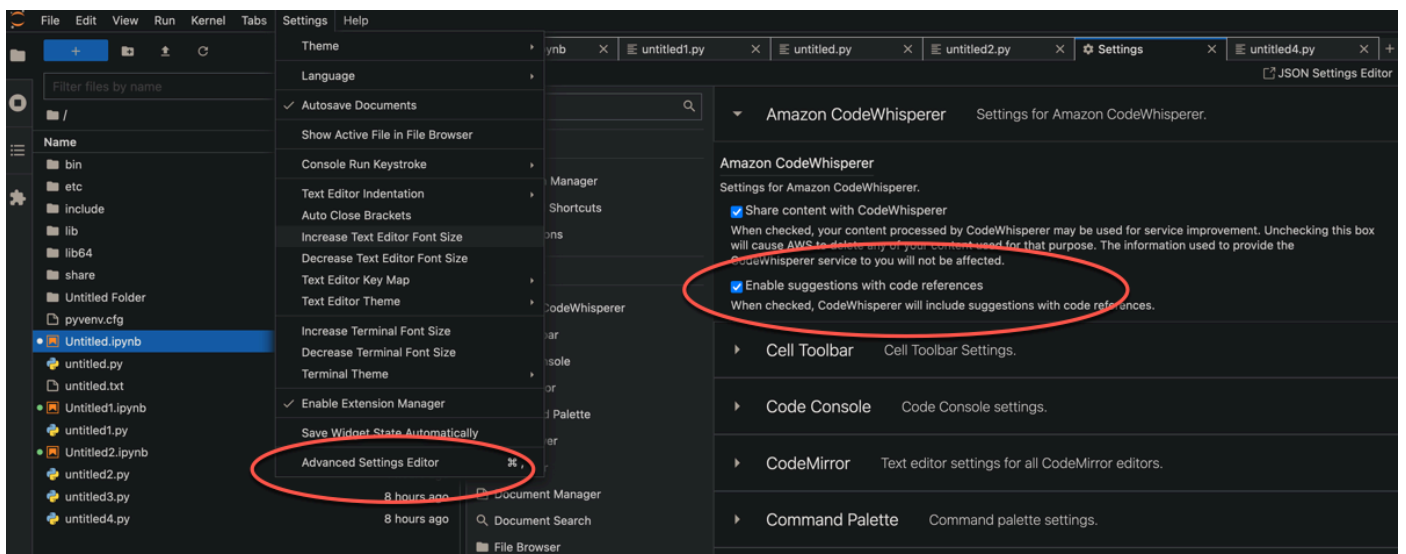


## JupyterLab

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**.



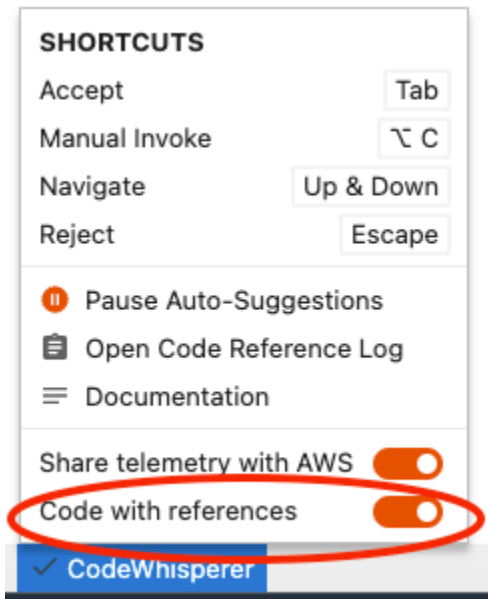
## AWS Glue Studio Notebook

1. From the bottom of the AWS Glue Studio Notebook window choose **Amazon Q**.

- 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.



## Opting out of code with references

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

This section explains how to do so.

### Toolkit for Visual Studio

You can toggle code references off and on in one of two ways:

- Choose the Amazon Q icon at the edge of the window, then select **Options...**
- Go to **Tools** -> **AWS Toolkit, Amazon Q**

Then change the toggle to **True** or **False**, depending on whether you want to include suggestions with references.

## AWS Toolkit for 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 procedure above. But if you have opted out at the enterprise level, that individual action will have no effect.

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

1. In the Amazon Q console, choose **Settings** and then **Edit details**.
2. On the Edit details page, under **Suggestions** deselect **Include suggestions with code references**.
3. Choose **Save changes**.

## AWS Toolkit for 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 procedure above. But if you have opted out at the enterprise level, that individual action will have no effect.

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

1. In the Amazon Q console, choose **Settings** and then **Edit details**.
2. On the Edit details page, under **Suggestions** deselect **Include suggestions with code references**.
3. Choose **Save changes**.

## 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

View the following code examples to see how Amazon Q suggests code in different scenarios.

## Topics

- [Single-line code completion](#)
- [Full function generation](#)
- [Block completion](#)
- [Docstring, JSDoc, and Javadoc completion](#)
- [Line-by-line recommendations](#)

## Single-line code completion

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

### 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  |
```

## TypeScript

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

```
TS index.ts ×
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.

```
// Get the attributes of an SNS topic by its ARN.
0 references
public async Task<Dictionary<string, string>> GetTopicAttributesByArn(string topicArn)
{
    var result = await _amazonSNSClient.GetTopicAttributesAsync(
        new GetTopicAttributesRequest()
        {
            TopicArn = topicArn
        });
}
```

## 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

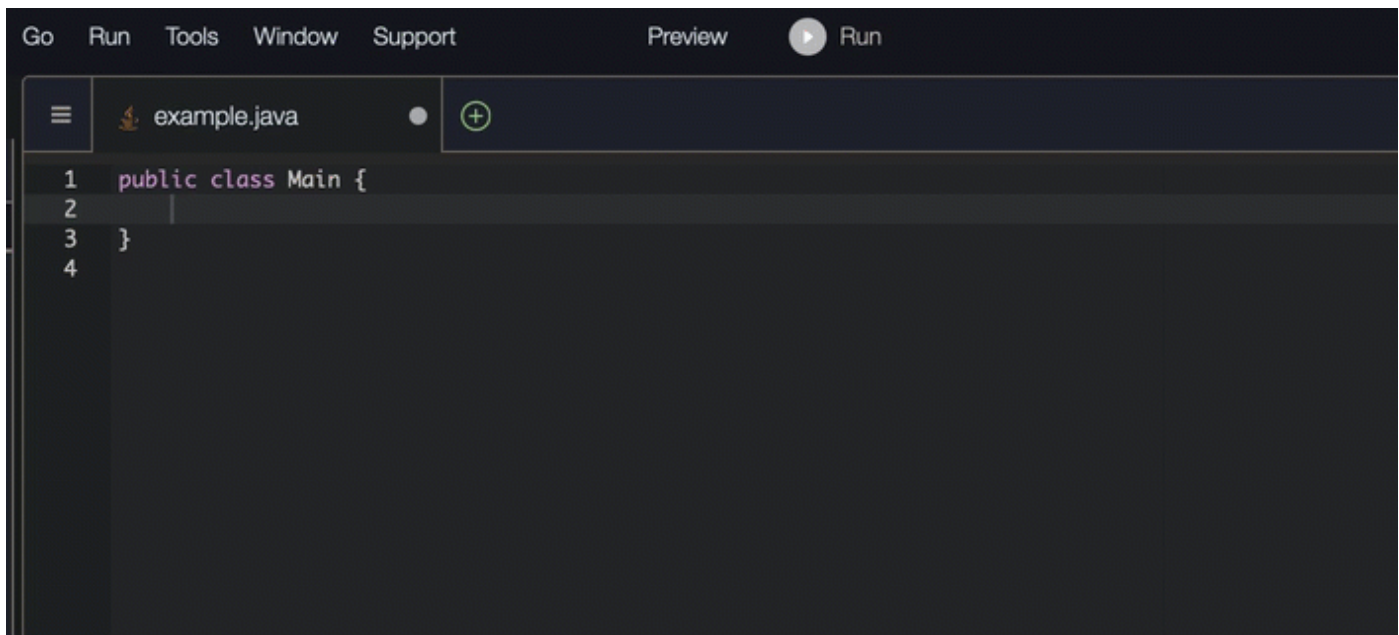
|
```

## Java

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

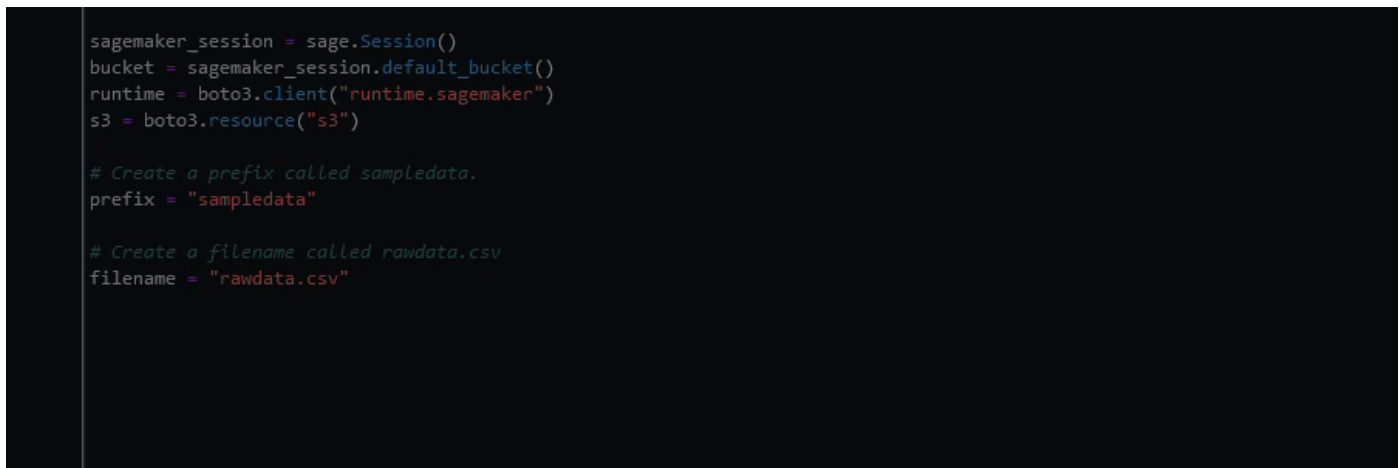
In the example below, in Java, a user enters the string `public` into an existing class.

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



## Python

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

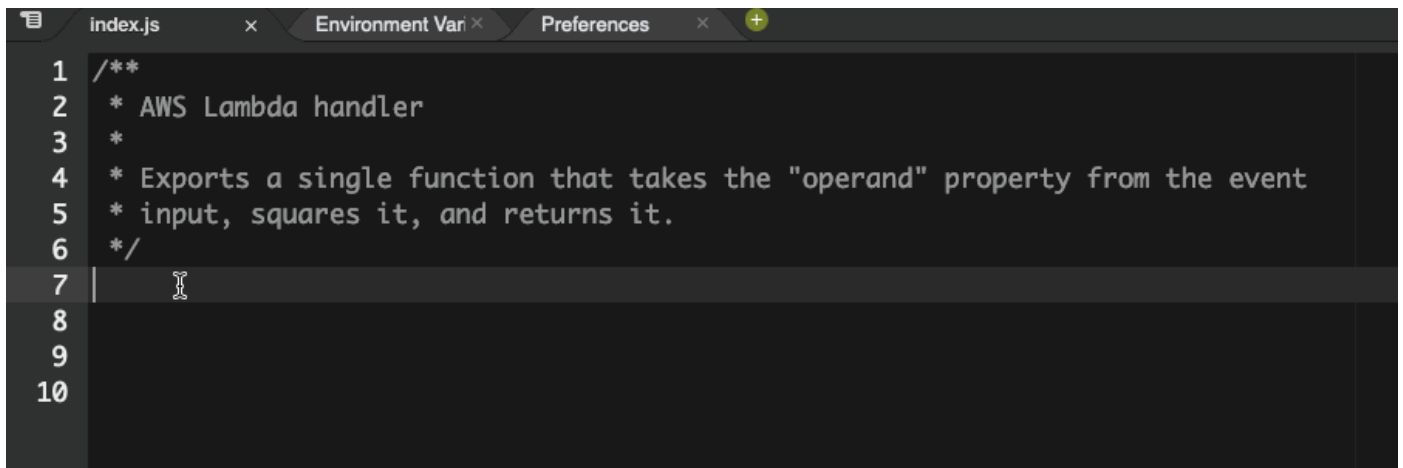


## Full function generation

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.


## JavaScript

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



```
index.js x Environment Var x Preferences x +
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.



```
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.

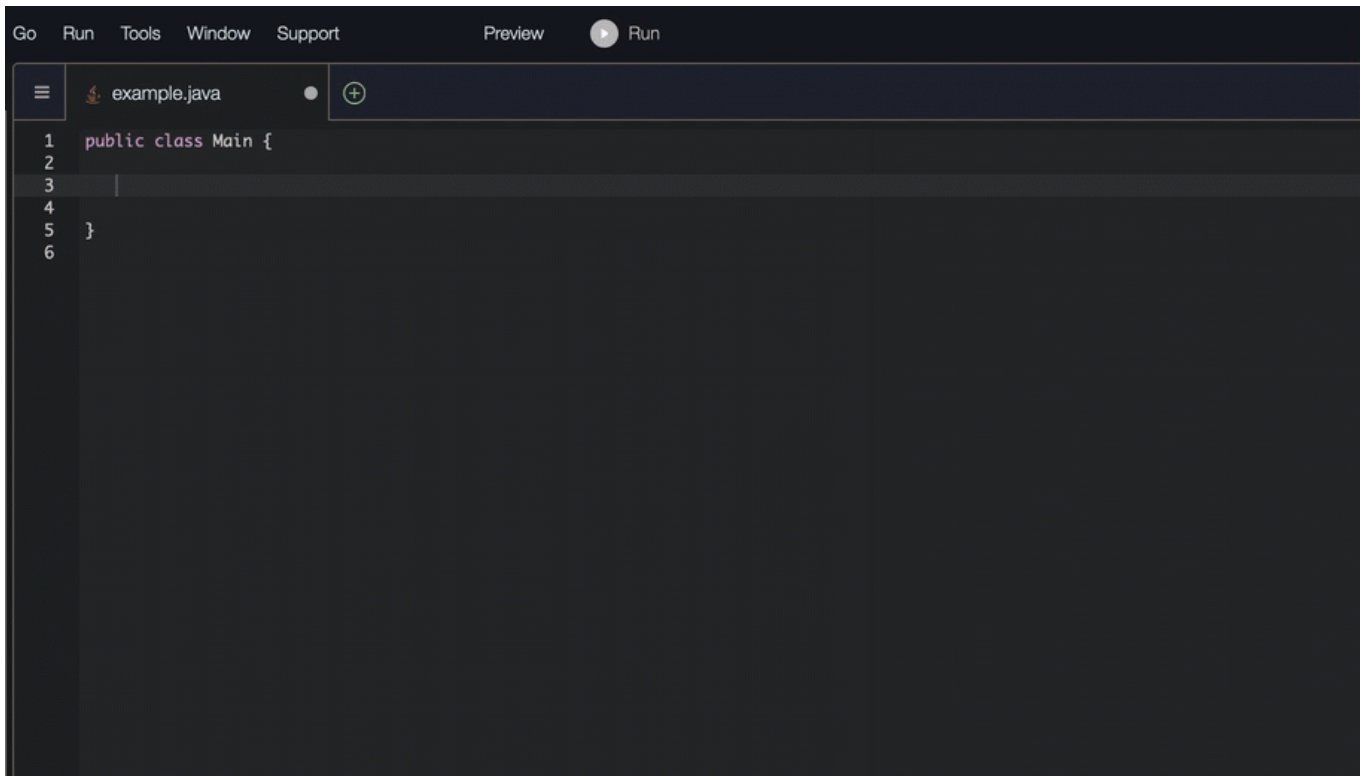
```
# Binary search function
def binary_search(arr, l, r, x):
    while l <= r:
        mid = l + (r - l) // 2
        if arr[mid] == x:
            return mid
        elif arr[mid] < x:
            l = mid + 1
        else:
            r = mid - 1
```

## Java

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

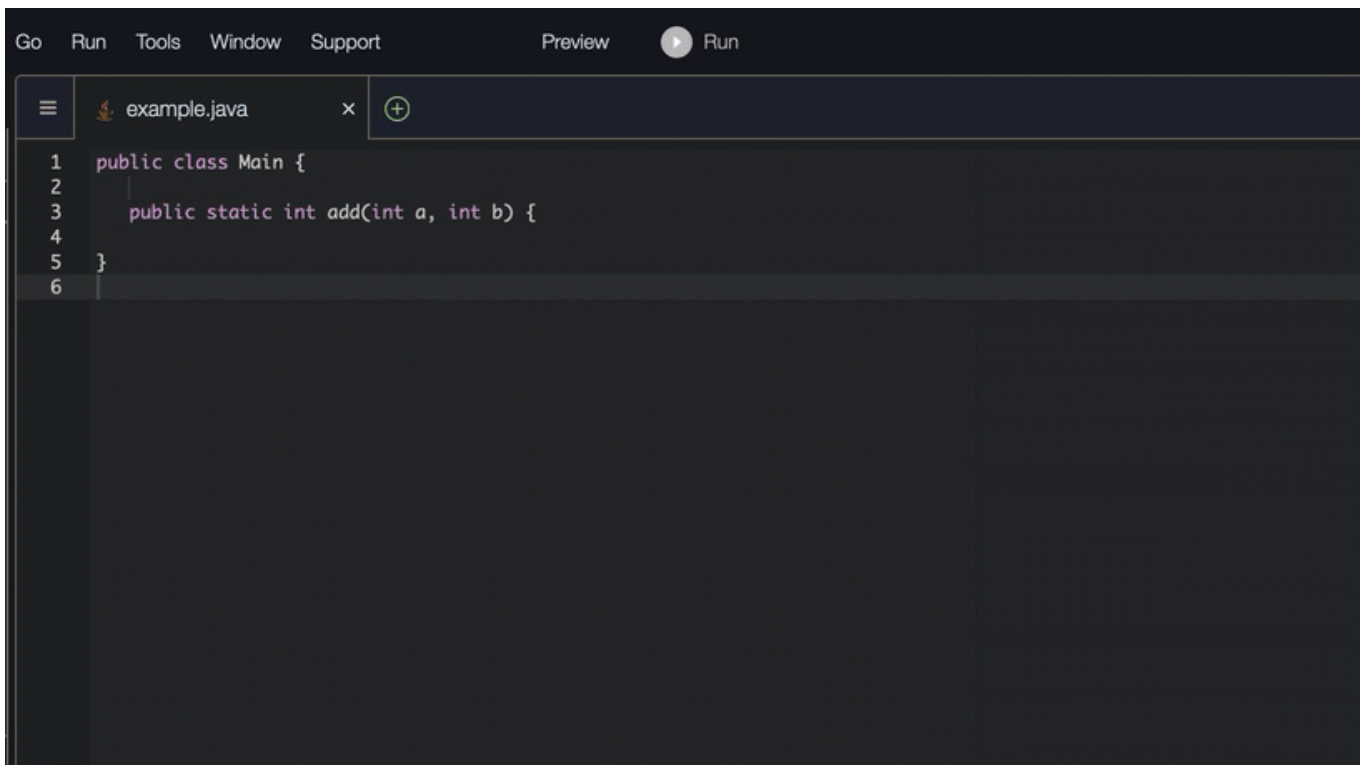
1. In the following example, a user inputs a comment. Amazon Q suggests a function signature.

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



```
Go Run Tools Window Support Preview Run
example.java
1 public class Main {
2
3
4
5 }
6
```

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.



```
Go Run Tools Window Support Preview Run
example.java x (+)
1 public class Main {
2
3     public static int add(int a, int b) {
4
5     }
6
```

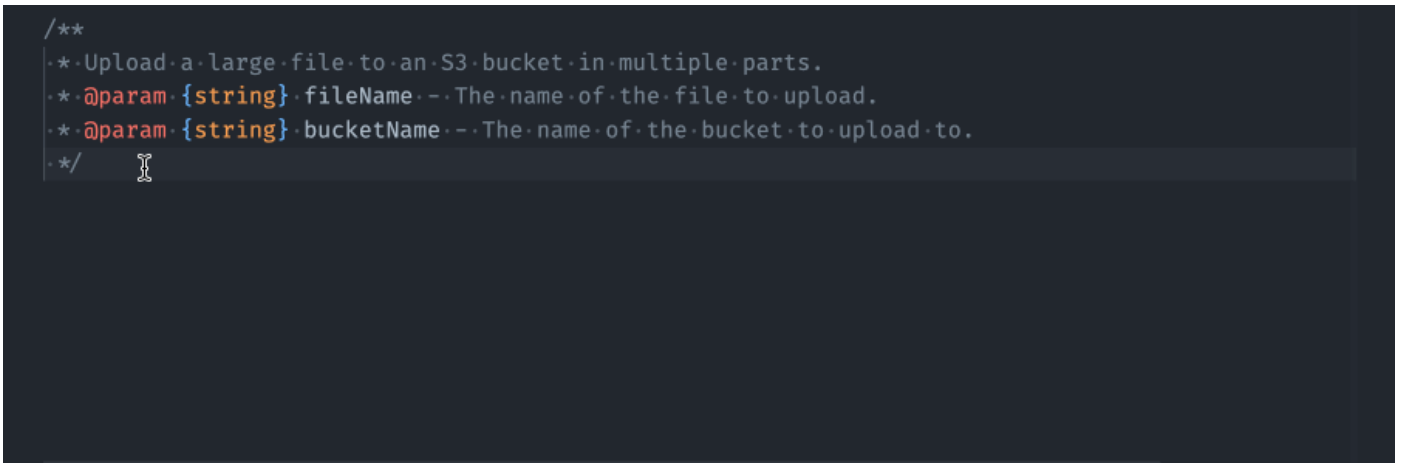
## C#

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



## TypeScript

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



## Python

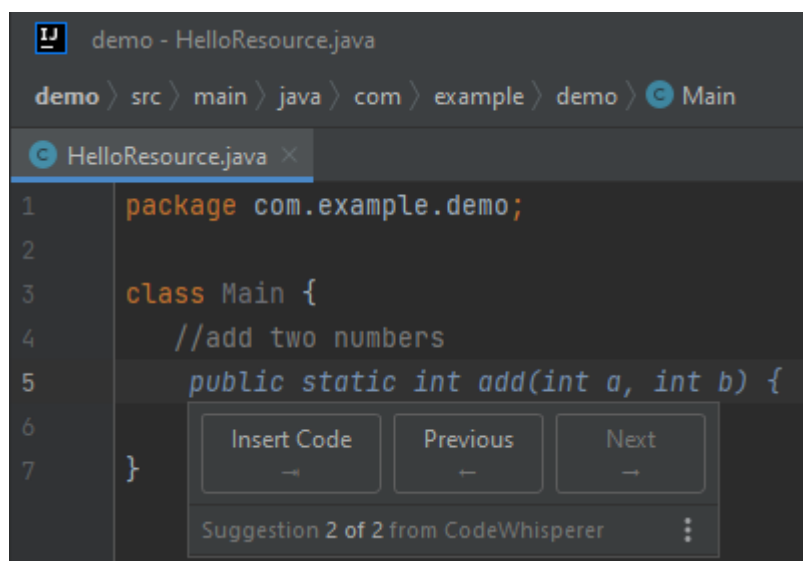
Amazon Q can generate an entire function based on a comment that you've written. As you finish your comment, Amazon Q will suggest a function signature. If you accept the suggestion, Amazon Q automatically advances your cursor to the next part of the function and makes a suggestion. Even if you enter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

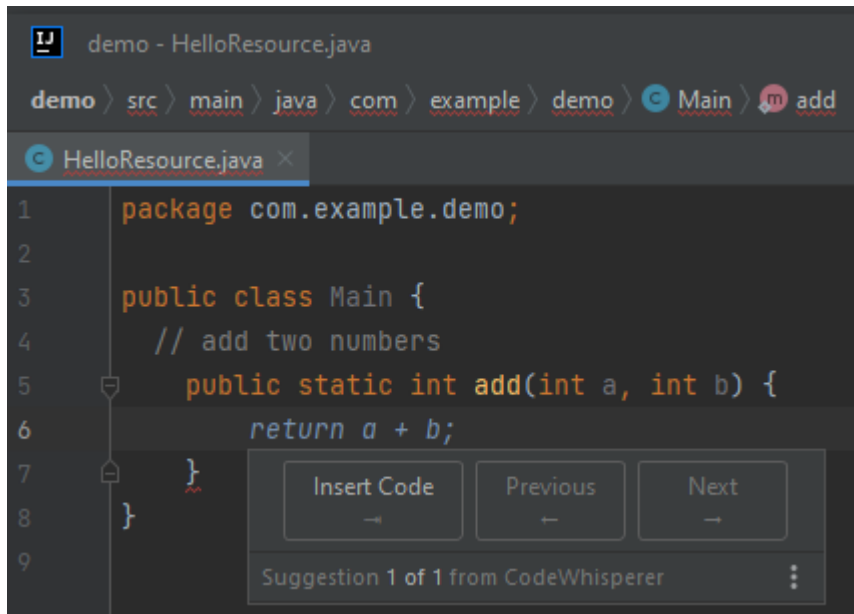


The screenshot shows an IDE window titled "demo - HelloResource.java". The breadcrumb navigation is "demo > src > main > java > com > example > demo > Main". A tab for "HelloResource.java" is open. The code editor shows the following code:

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

A suggestion box is visible over the code, containing three buttons: "Insert Code", "Previous", and "Next". Below the buttons, it says "Suggestion 2 of 2 from CodeWhisperer".

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.

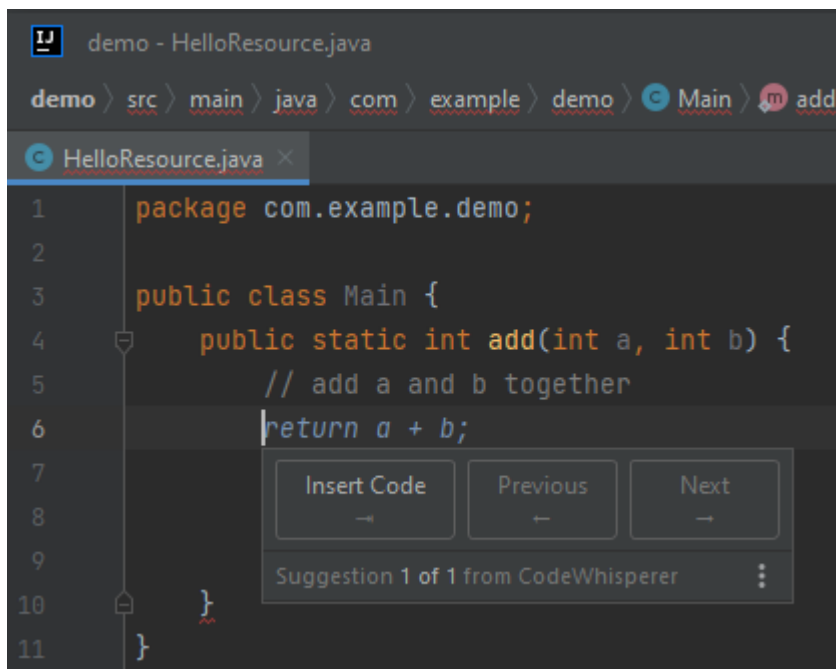


The screenshot shows an IDE window titled "demo - HelloResource.java". The breadcrumb navigation is "demo > src > main > java > com > example > demo > Main > add". The file "HelloResource.java" is open, showing the following code:

```
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
```

A CodeWhisperer suggestion box is visible, containing the text "Suggestion 1 of 1 from CodeWhisperer" and three buttons: "Insert Code", "Previous", and "Next".

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.



The screenshot shows the same IDE window as above, but with a comment added to the code:

```
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
10
11 }
```

The CodeWhisperer suggestion box is still present, showing "Suggestion 1 of 1 from CodeWhisperer" and the same three buttons: "Insert Code", "Previous", and "Next".

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

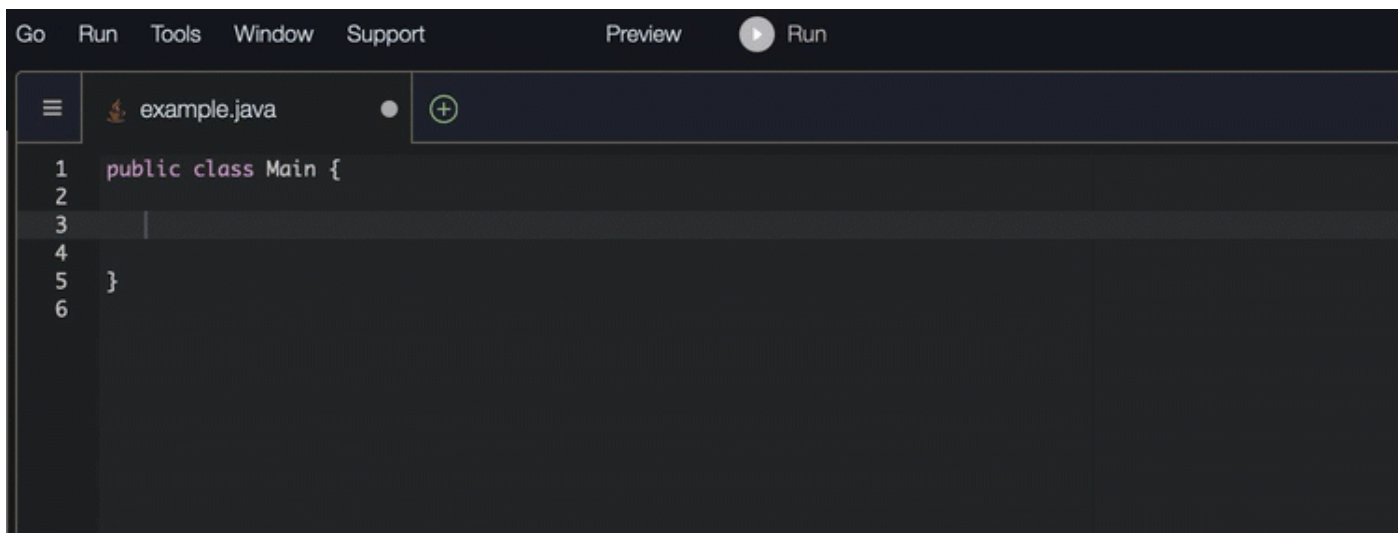
```
examplebucketname = "example-bucket-1"
```

## Block completion

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

### 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.



The screenshot shows an IDE window titled "example.java" with a menu bar containing "Go", "Run", "Tools", "Window", "Support", "Preview", and "Run". The code editor displays the following Java code:

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

### TypeScript

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

```
TS index.ts 2 x
TS index.ts > [e] 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 (
```

## AWS Toolkit for JetBrains

In the image below, a user has input the signature of an if statement. The body of the statement, `System.out.println("negative");` is a suggestion from Amazon Q.

```
demo - HelloResource.java
demo > src > main > java > com > example > demo > Main
HelloResource.java x
1 package com.example.demo;
2
3 class Main {
4     if(number < 0)
5
6     {
7         System.out.println("Number is negative")
8
9     }
10 }
```

## Python

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



```
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)
```

## Docstring, JSDoc, and Javadoc completion

### Javascript

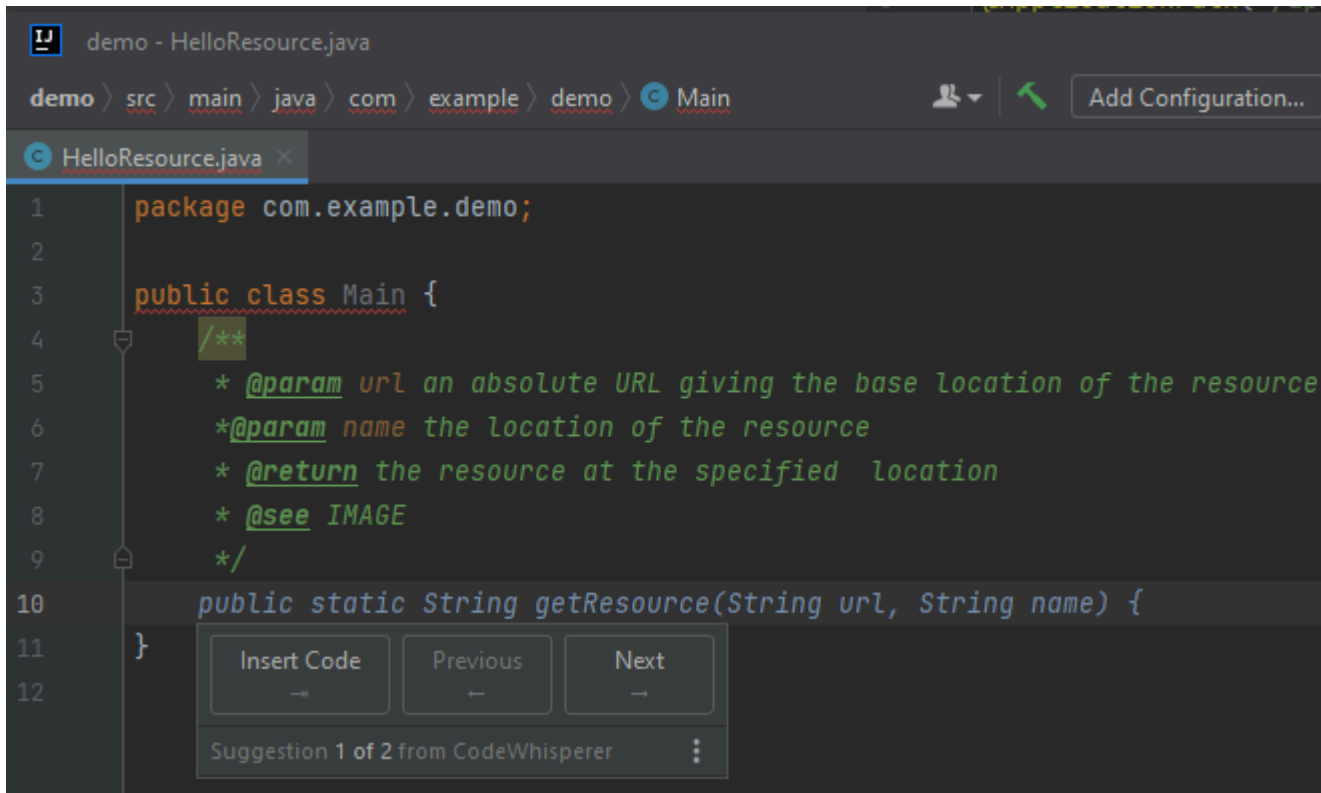
In this example, Amazon Q fills in JSDoc parameters based on existing constants.

```
1 import {PutObjectCommand, S3Client} from "@aws-sdk/client-s3";
2
3 const client = new S3Client({});
4
5 /**
6  *
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));
```

### 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.



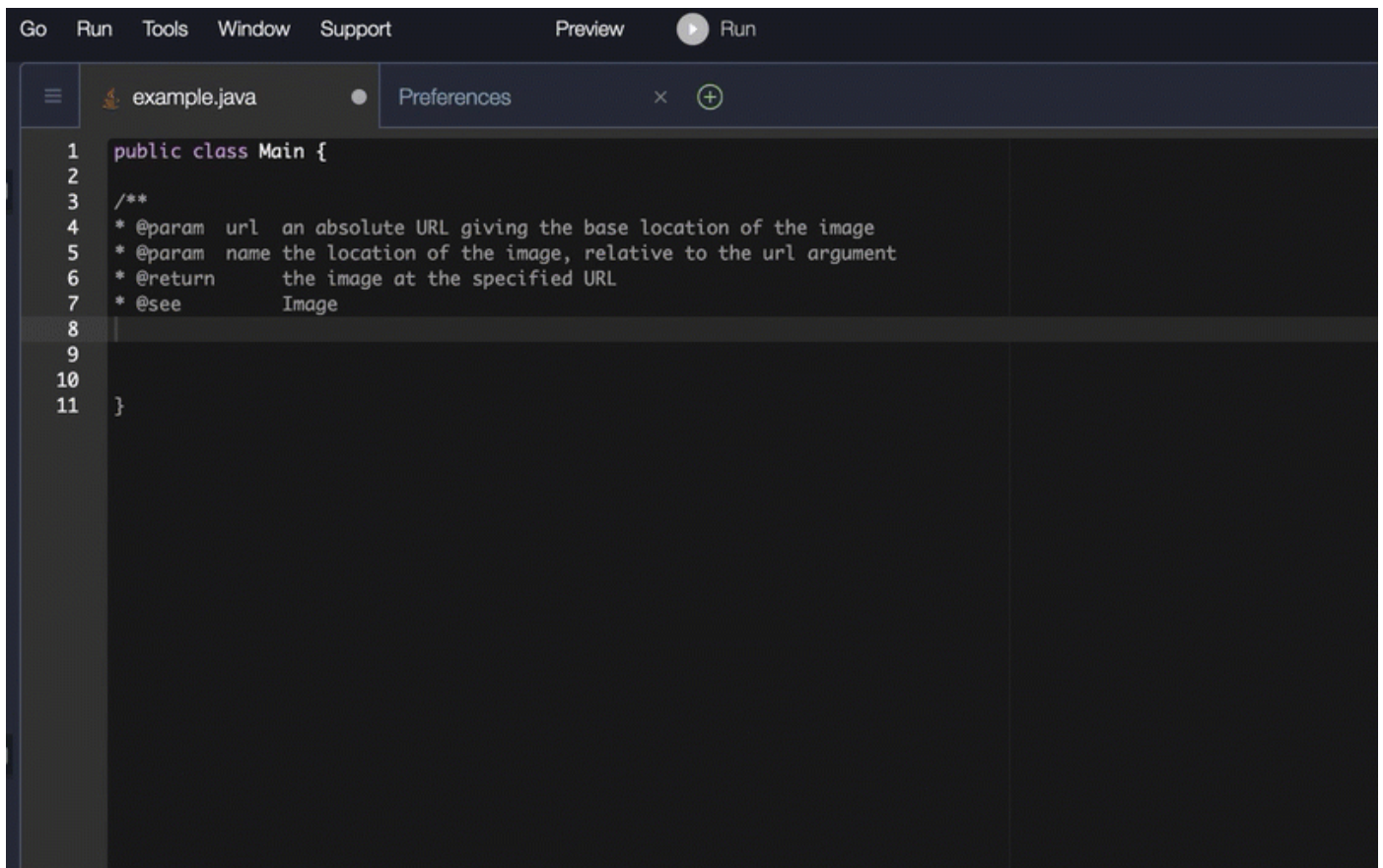
The screenshot shows an IDE window titled "demo - HelloResource.java". The breadcrumb navigation is "demo > src > main > java > com > example > demo > Main". The file name "HelloResource.java" is shown in the tab. The code is as follows:

```
1 package com.example.demo;
2
3 public class Main {
4     /**
5      * @param url an absolute URL giving the base location of the resource
6      * @param name the location of the resource
7      * @return the resource at the specified location
8      * @see IMAGE
9      */
10    public static String getResource(String url, String name) {
11    }
12
```

A CodeWhisperer suggestion box is visible at the bottom of the code editor, containing the text "Suggestion 1 of 2 from CodeWhisperer" and three buttons: "Insert Code", "Previous", and "Next".

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.

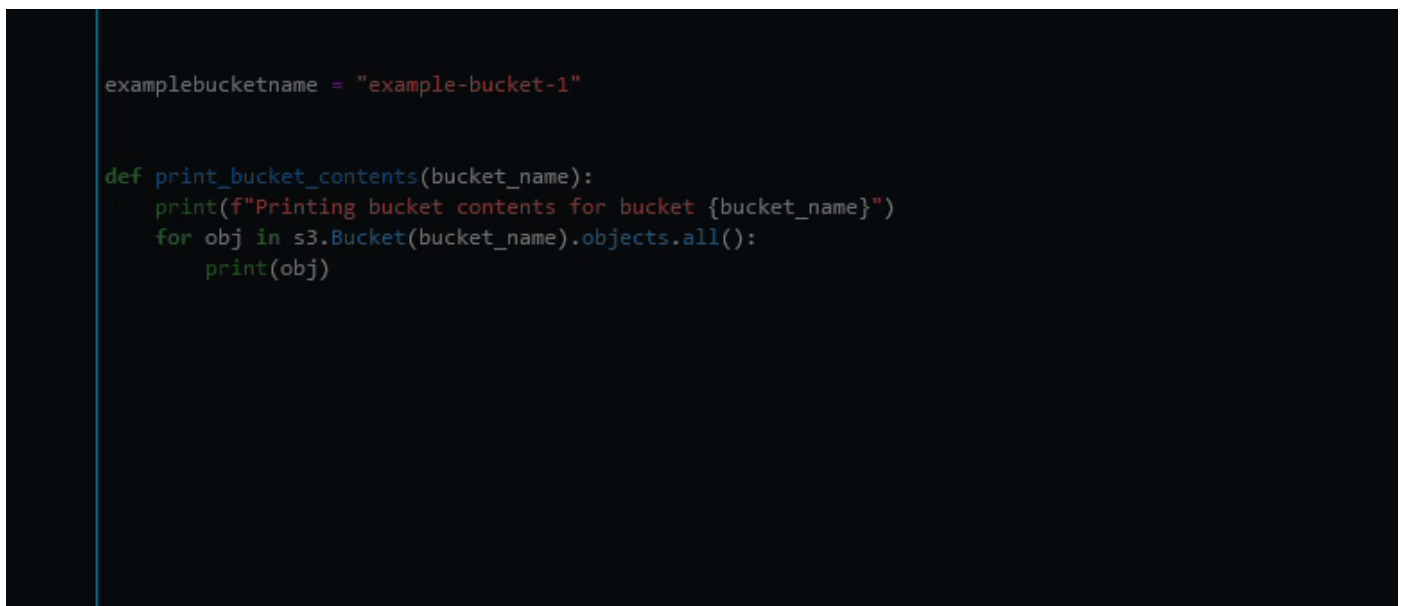


The screenshot shows an IDE window with a dark theme. The menu bar at the top includes 'Go', 'Run', 'Tools', 'Window', 'Support', 'Preview', and 'Run'. The active file is 'example.java'. The code in the editor is as follows:

```
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 a Python code snippet in a dark-themed IDE. The code is as follows:

```
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)
```

## 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.

```
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 }
18
```

Here is another example of line-by-line recommendations, this time with a unit test.

```
3     import "testing"
4
5     func Add(a, b int) int { no usages
6         return a + b
7     }
8
9
10
11
12
13
14
15
16
17
```

### 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")
```

## Upgrade language versions with the Amazon Q Developer Agent for Code Transformation

The Amazon Q Developer Agent for Code Transformation can upgrade the code language version of your files in place. You can transform a module in JetBrains and a project or entire workspace in Visual Studio Code.

Amazon Q first builds your code in the source language version and verifies that it has the information necessary to transform your code. After Amazon Q successfully transforms your code, you verify and accept the changes in your integrated development environment (IDE). For more information, see [How the Amazon Q Developer Agent for Code Transformation works](#).

Currently, Amazon Q can upgrade Java 8 and Java 11 code to Java 17 code.

## Topics

- [Step 1: Prerequisites](#)
- [Step 2: Configure your project](#)
- [Step 3: Transform your code](#)
- [How the Amazon Q Developer Agent for Code Transformation works](#)
- [Troubleshooting issues with the Amazon Q Developer Agent for Code Transformation](#)

## 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 a Java 8 or Java 11 project built on Maven.
- Your project successfully builds with Maven in your IDE. Amazon Q Code Transformation supports Maven 2.8, but we recommend Maven 3.9.5 or later.
- Your project source JDKs are available locally.
- 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 project doesn't use pom.xml file dependency version ranges. For example, if your pom.xml file includes `<version>[1.0.0, )</version>`, the transformation will fail. For more information, see [Version Range Specification](#) in the Apache Maven Project documentation.
- 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](#).

## 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 for Code Transformation in JetBrains, specify the JDK version for your project in the **Project Structure** settings. For more information, see [Change the JDK version in a Maven project](#) in the JetBrains IDEA documentation and follow the instructions for *Change the JDK version in the Project Structure*.

### Configure a project in VS Code

To configure your project for Code Transformation 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/dhasani23/QCT-sample-app>.

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 your project build time exceeds 55 minutes or the project build artifact is greater than 1 GB, the transformation will fail. For information about addressing build time and build artifact size, see [Troubleshooting](#).

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 menu, 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 the Amazon Q Developer Agent for Code Transformation works

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. This build artifact must be under 1 GB for Amazon Q to transform your code.

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.

## 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:

- Upgrade popular libraries and frameworks to a version compatible with Java 17. This includes updating Spring, Spring Boot, Junit, JakartaEE, Mockito, Hibernate, and Log4j to their latest available major versions.
- Update deprecated code components according to Java 17 recommendations

## 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.

# Troubleshooting issues with the Amazon Q Developer Agent for Code Transformation

The following information can help you troubleshoot common issues with the Amazon Q Developer Agent for Code Transformation.

## Topics

- [Why does my project exceed size limits?](#)
- [Why are my Maven commands failing?](#)
- [How do I add Maven to my PATH?](#)
- [Why did my transformation fail after 55 minutes?](#)
- [Why can't I see my upgraded code?](#)
- [How do I access logs emitted by Code Transformation?](#)
- [How do I find my transformation job ID?](#)

## Why does my project exceed size limits?

To transform your code, Amazon Q generates a build artifact that contains your source code, project dependencies, and build logs. The maximum build artifact size for a transformation job is 1 GB. If your build artifact exceeds the limit, you see the following error:

Your project size exceeds the Amazon Q Code Transformation upload limit of 1GB.

Even if the size of your code project is under 1 GB, its dependencies can result in a larger artifact file. To ensure that the artifact file is under 1 GB, you might need to reduce the dependencies of your project or the size of your source code.

## 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.

## 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 menu, 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 menu, 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.

## 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 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 see my upgraded code?

After the transformation is complete, Amazon Q generates an output artifact with a maximum size of 1 GB. The output artifact contains a diff with your upgraded code and a transformation summary. The artifact is only available for 24 hours after the transformation has completed.

There is a small chance that the output artifact exceeds 1 GB. In that case, the IDE can't download the output artifact and an error occurs when you try to view your upgraded files or transformation summary.

If this issue occurs after the transformation is complete, try transforming your code again. If the issue persists, contact AWS Support. For information about contacting AWS Support with Amazon Q, see [Use Amazon Q Developer to chat with AWS Support](#).

## How do I access logs emitted by Code Transformation?

### JetBrains

For information about how to access JetBrains log files, see [Locating IDE log files](#) in the JetBrains documentation.

To find logs emitted by Code Transformation in JetBrains, search the IDE logs for the following string:

```
software.aws.toolkits.jetbrains.services.codemodernizer
```

Logs emitted by Code Transformation 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.

### Visual Studio Code

To find logs emitted by 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**.



Logs emitted by Code Transformation 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?

### 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.

### VS Code

To find a transformation job ID in VS Code, go to the **Transformation Hub** and choose the **Show Job Status** (clock) icon.

## Develop 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 feature you want to develop, and Amazon Q uses the context of your current project to generate an implementation plan and the accompanying code to implement the feature. 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 files in your workspace root as context to develop a plan and generate code.

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 Amazon Q to generate an implementation plan and new code for your feature.

### Develop features with `/dev`

To develop a feature 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 extension, and then choose **Open Chat Panel**. Enter `/dev` in the Amazon Q chat panel that opens. You can also type your feature description directly after `/dev`.

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.
```

3. A new tab opens. Enter a description of the feature that you want to develop or the issue that 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 come up with an implementation plan.
4. Amazon Q responds with an implementation plan, including the code updates that it would make to complete the task you described.
5. Review the implementation plan that Amazon Q generates. If you want to change anything about the plan, enter modifications or more details to clarify what you want to achieve.
6. When you're satisfied with the implementation plan, choose **Generate code**. Amazon Q generates new code and proposes updates to your files based on the implementation plan. This might take a few minutes.
7. After Amazon Q has generated the code for your feature, it provides a list of files with proposed changes. Choose a file to view a file diff with the changes.
8. To accept the changes and update your file, choose **Accept**. To reject the changes to the file, choose **Reject**.

If you aren't satisfied with the proposed changes, you can provide feedback in the chat panel on what can be improved. Amazon Q generates new code based on your feedback. When the generated code meets your requirements, choose **Accept** to update the code in your workspace.

9. To develop another feature with Amazon Q in the same chat tab, choose **Discuss a new plan** and repeat these steps. Your previous conversation and implementation plan aren't used as context for the new plan.

## Best practices

To make the most out of Amazon Q feature 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 create a comprehensive implementation plan and code changes that can be implemented. See Examples
- 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 Amazon Q to perform, with the corresponding description you might provide.

- **Update CSS Styles for Responsive Layout:** 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.
- **Fix User Profile Image Upload:** 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.
- **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.
- **Improve Accessibility for Form Fields:** Enhance accessibility by adding appropriate labels and aria attributes to form fields. Ensure compatibility with screen readers and assistive technologies for users with disabilities.
- **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.

## Scan 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](#)
- [Start a scan](#)
- [Address 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:

Resource	Auto scans	Project scans
Maximum input artifact size	200 KB	5 GB
Maximum source code size	200 KB	50 MB

## Start 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 not available with the free tier of Amazon Q.

Amazon Q auto-scans automatically scan 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.

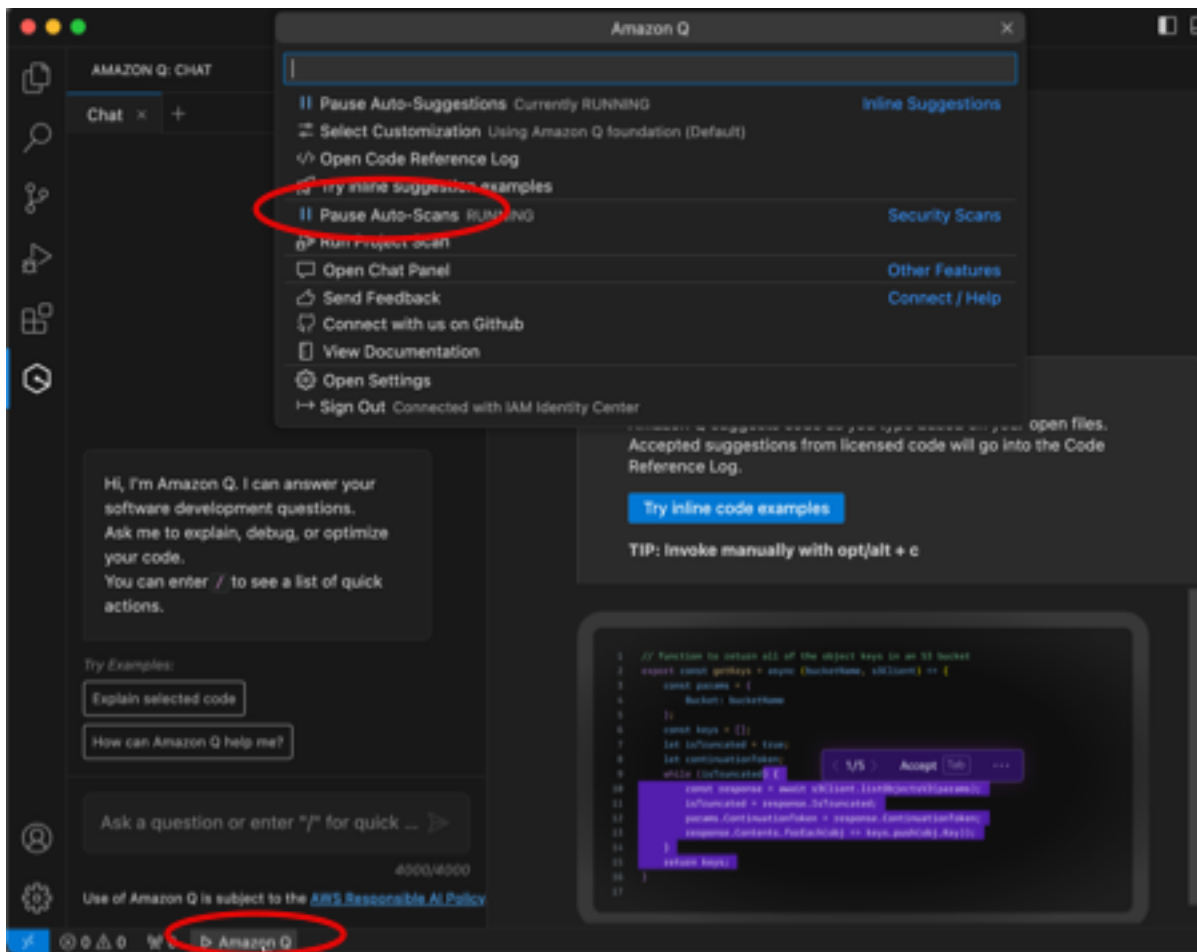
### Pausing auto-scans

To pause auto scans, complete the following steps.

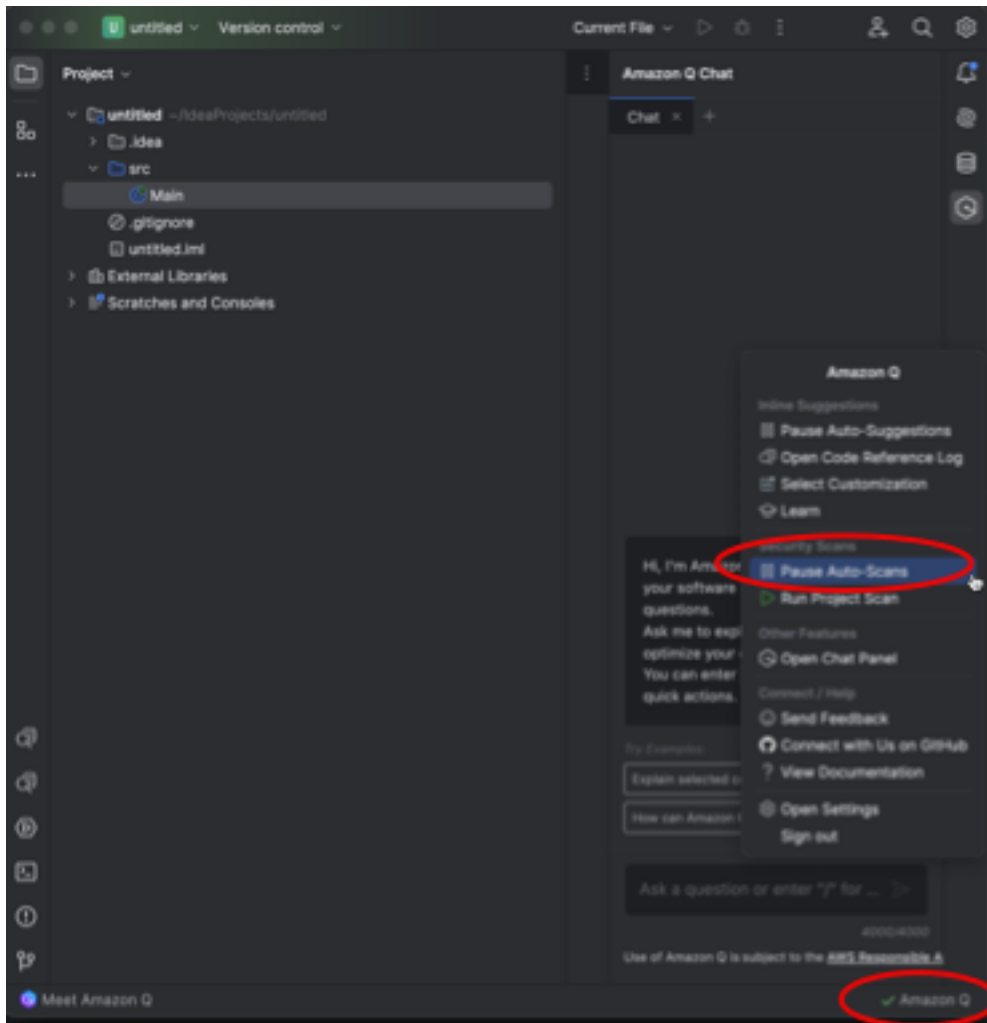
1. Choose the text, **Amazon Q**, from the rectangular area at the bottom of the IDE window. In VS Code, this area is called the *component tray*. In JetBrains IDEs, this area is called the *status bar*.

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. You can scan up to 100 MB of code at a time.

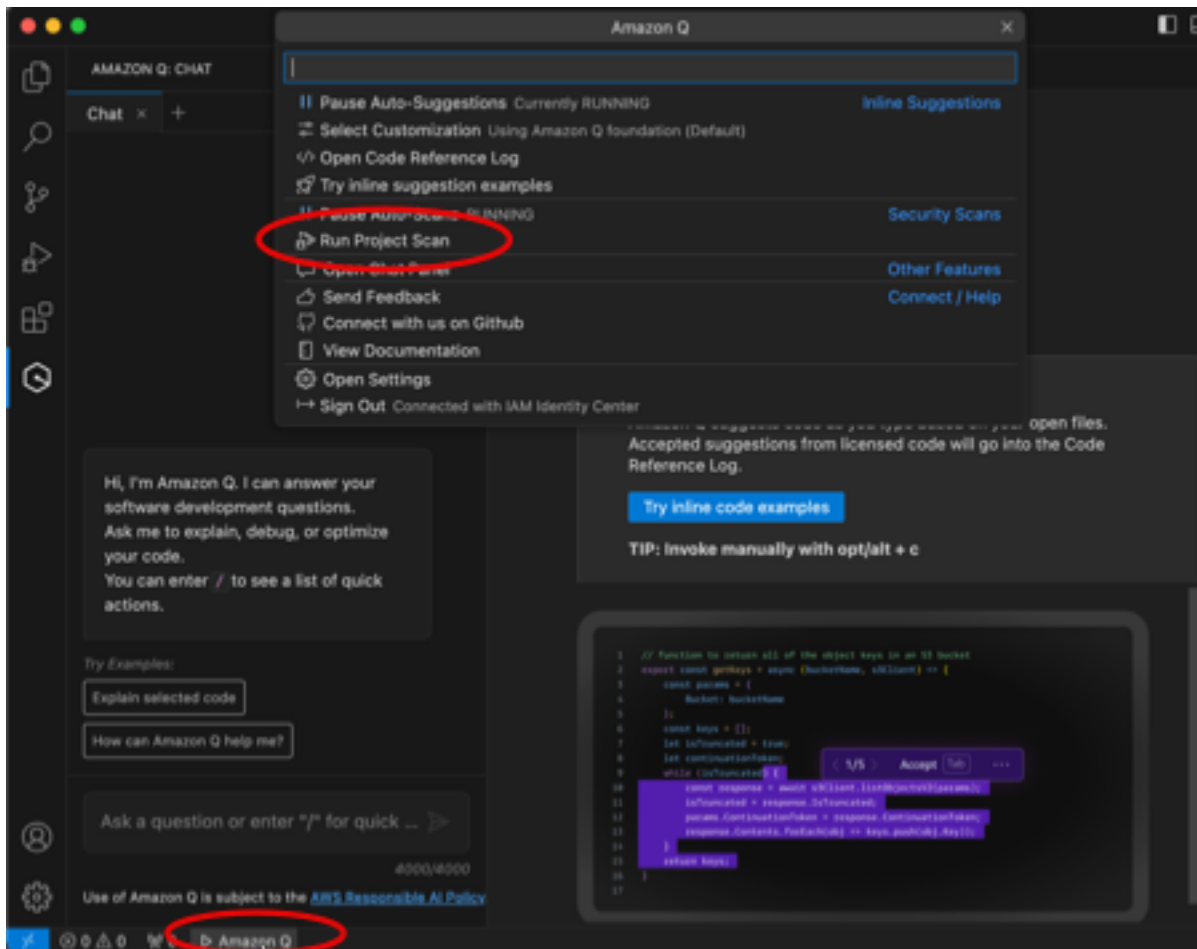
To manually run a scan, complete the following steps.

1. Open up a file from the project you want to scan in your IDE.
2. 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*.

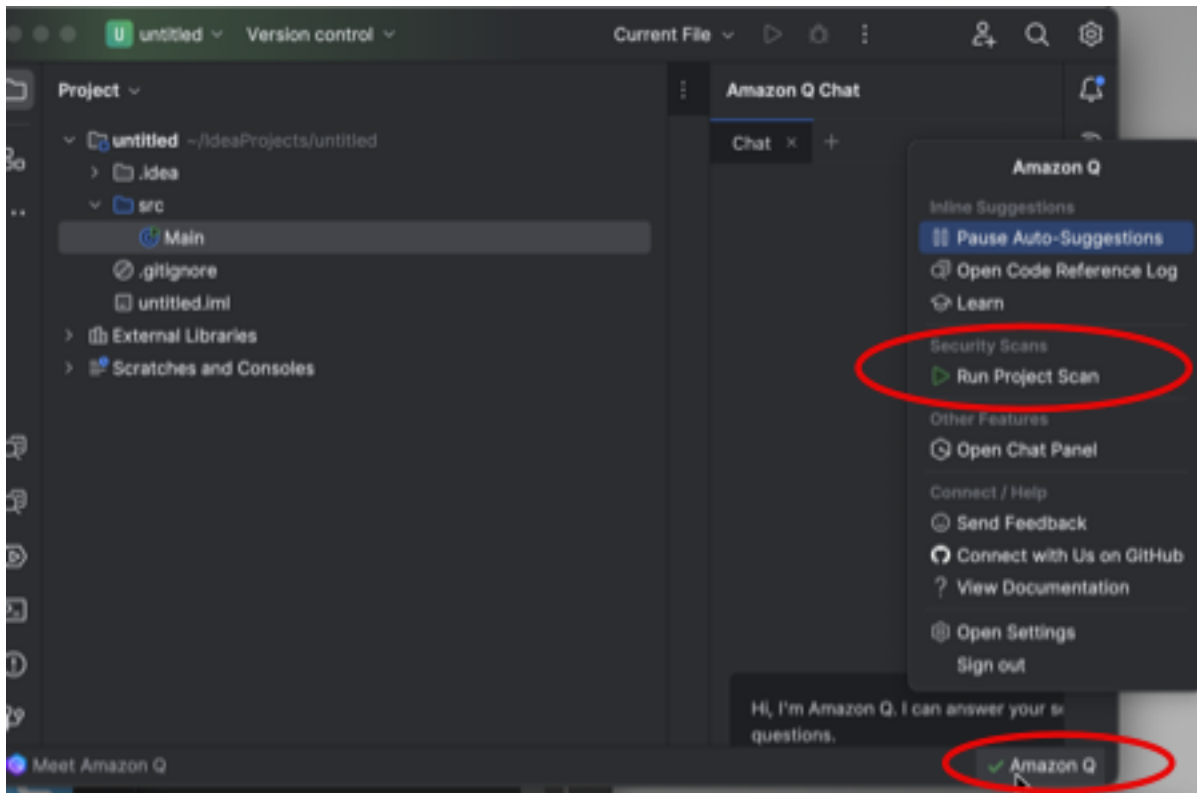
After you choose the Amazon Q text the Amazon Q task bar opens at the top of the IDE window.



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 **Run Project Scan**.



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 **Run Project Scan**.



3. Choose **Run Project Scan**. Amazon Q begins scanning your project.
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 [Address findings](#).

## Address 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.

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.

## Supported languages for Amazon Q Developer in the IDE

### 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:

- Java
- Python
- JavaScript
- TypeScript
- C#
- Go
- PHP
- Rust
- Kotlin
- SQL

The Infrastructure as Code (IaC) languages with the most support are:

- JSON (AWS CloudFormation)
- YAML (AWS CloudFormation)
- HCL (Terraform)
- CDK (Typescript, Python)

Amazon Q also supports code generation for:

- Ruby
- C++
- C
- 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
- **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

# Amazon Q Developer for command line

Amazon Q Developer in the command line adds 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.

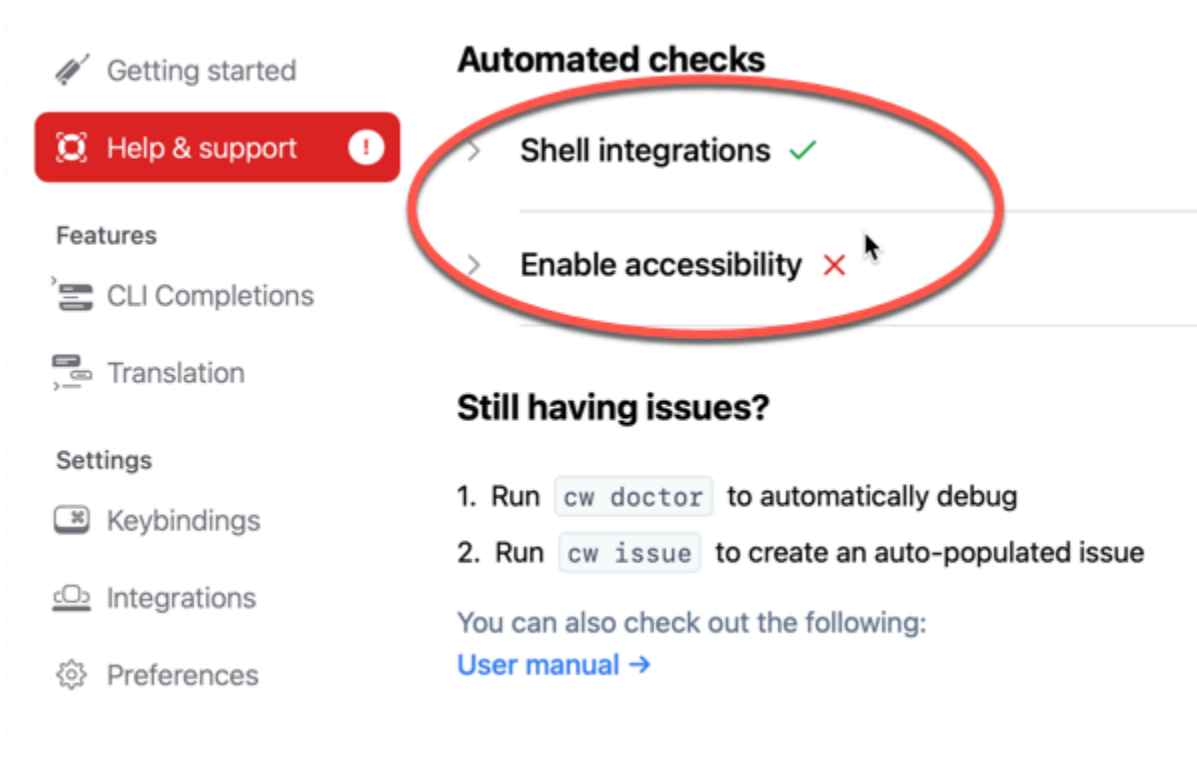
## Topics

- [Installing Amazon Q for command line](#)
- [Amazon Q chat in the CLI](#)
- [CLI Completions](#)
- [Natural language to bash translation](#)
- [Debugging Amazon Q Developer for the command line](#)
- [Adding your own completion specs to Amazon Q Developer](#)

## Installing Amazon Q for command line

To install Amazon Q for command line, follow the steps below.

1. [Download Amazon Q for command line \(macOS only\)](#)
2. Authenticate with [Builder ID](#), or with [IAM Identity Center](#) using the start URL given to you by your account administrator.
3. Follow the instructions to install the shell integrations, and to grant macOS accessibility permissions.



## Supported command line environments

Amazon Q for command line integrates with the following environments:

- Operating systems: macOS
- Shells: bash, zsh, fish
- Terminal emulators: iTerm2, macOS terminal, Hyper, Alacritty, Kitty, wezTerm
- IDEs: VS Code terminal, JetBrains terminals (except Fleet)
- CLIs: 500+ of the most popular CLIs such as git, aws, docker, npm, yarn

## Verifying your download

After you download Amazon Q for command line, you can verify its code signature as follows:

```
codesign -v /Applications/Amazon\ Q.app
```

If there is no output, then the app's code signature is valid, and it has not been tampered with since it was signed.

For more verbose information about the app signature, run:

```
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.

## Amazon Q chat in the CLI

The Amazon Q CLI is a command-line interface that allows you to interact with the Amazon Q. With the Amazon Q CLI, 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 CLI 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 CLI 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

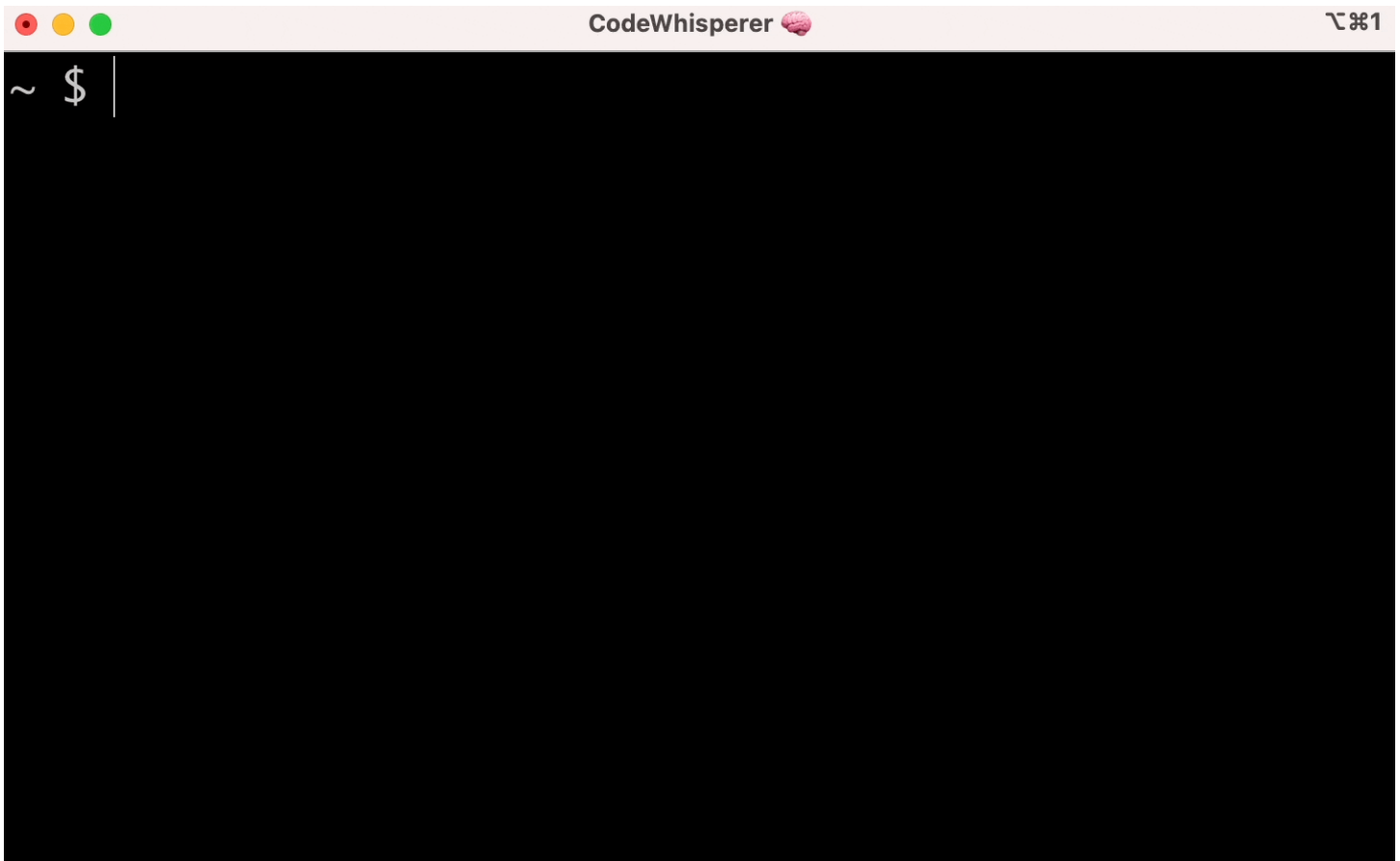
To start using the Amazon Q CLI, follow these steps:



1. Install the Amazon Q CLI by following the installation instructions provided in the official documentation.
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.

## CLI Completions

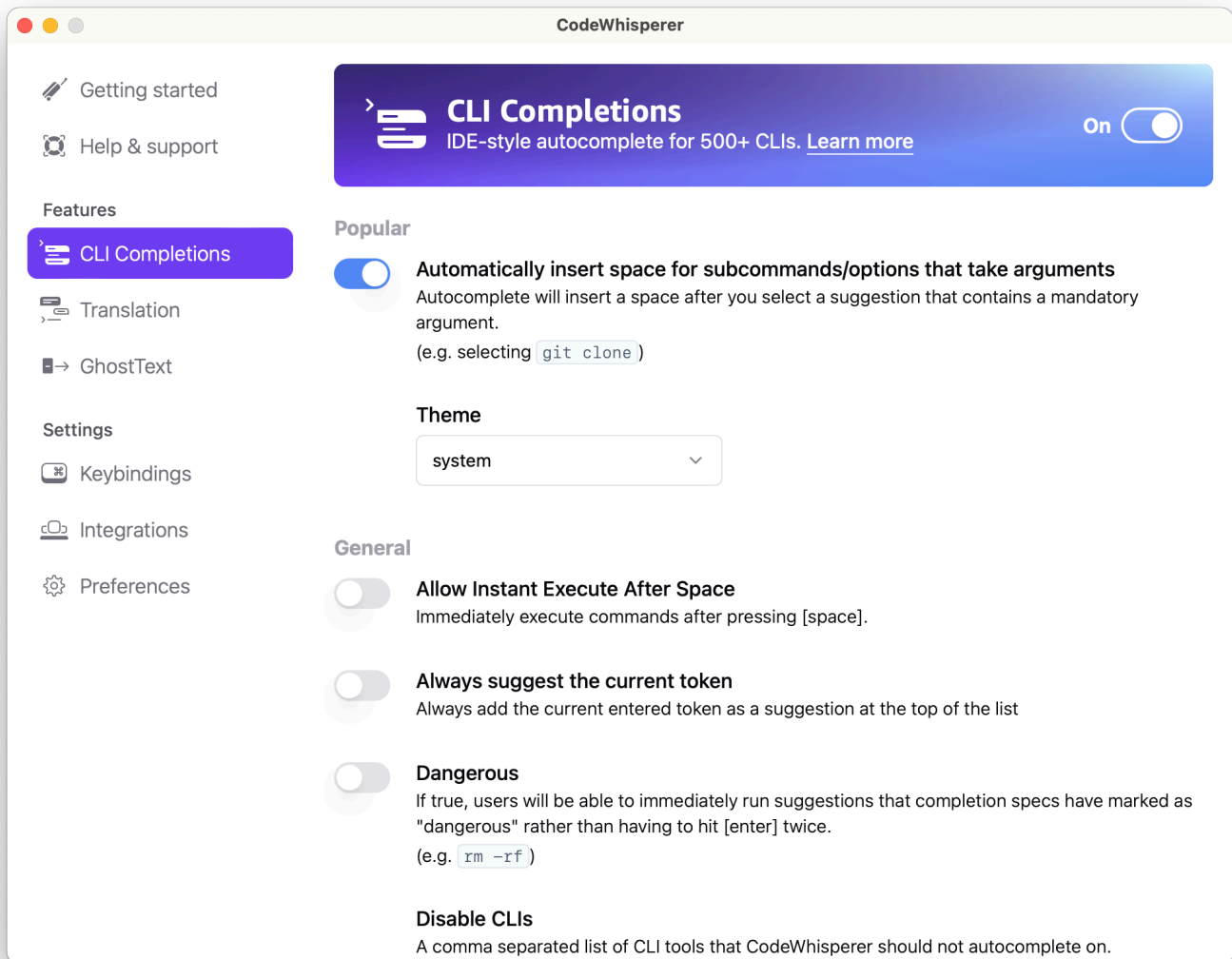
Amazon Q Developer for command line adds IDE-style completions for hundreds of popular CLIs like `git`, `npm`, `docker`, and `aws`. Start typing, and Amazon Q will pop up contextually relevant subcommands, options and arguments.



## 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 CLI 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 CLI Autocomplete on a remote machine with SSH

By default, the Amazon Q CLI adds autocomplete for over 500 CLI tools to your existing terminal in your local environment. By installing the SSH integration, CLI autocomplete will be 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 CLI 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 CLI 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 CLI.

### Install the Amazon Q CLI

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 CLI 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 CLI 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
BKNG0m0JEftYXpvbiBRIENSSBUZWFtIDxxLWNsaUBhbWF6b24uY29tPoiZBBMW
CgBBFiEEmvYEF+gnQskUPgPsUNx6jcJMVmcfAmYo0tECGwMFCQPCZwAFCwkIBwIC
IglIGFQoJCAsCBBYCAwECHgcCF4AACgkQUNx6jcJMVmef5QD/QWWEgg/c0nbDnp68
SJXuFkwiNw1H2rPw9ZRIQMnfAS0A/0V6ZsGB4k0y1Bfc7CNfzRFGtovdBBgHqA6P
zQ/PNscGuDgEZig60RIKKwYBBAGXVQEFaQEHQC4qle0NMBCq3+wJwbZSr0vbuRba
D1xr4wUPn4Avn4AnAwEIB4h+BBgWCgAmFiEEmvYEF+gnQskUPgPsUNx6jcJMVmcf
AmYo0tECGwMFCQPCZwAACgkQUNx6jcJMVmchMgEA6l3RveCM0YHAGQaSFmkguoAo
vK6Fg0kDawgP0NPIP2oA/jIA04gsAntuQgM0sPunEdDeji2t+AhV02+DQIsXZpoB
=f8yY
-----END PGP PUBLIC KEY BLOCK-----
```

- c. Import the Amazon Q CLI 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 CLI Team <q-cli@amazon.com>"
imported
gpg: Total number processed: 1
gpg:             imported: 1
```

- d. Download the Amazon Q CLI 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 CLI, 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 CLI, 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 -sSf "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 CLI, 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 CLI, 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"
```

For a specific version of the Amazon Q CLI, 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 CLI Team <q-cli@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](#).

- 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
```

- 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
```

- 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
```

- 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/u1vx4xfmvqav0oxfa4zfknaixwmsbr/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.

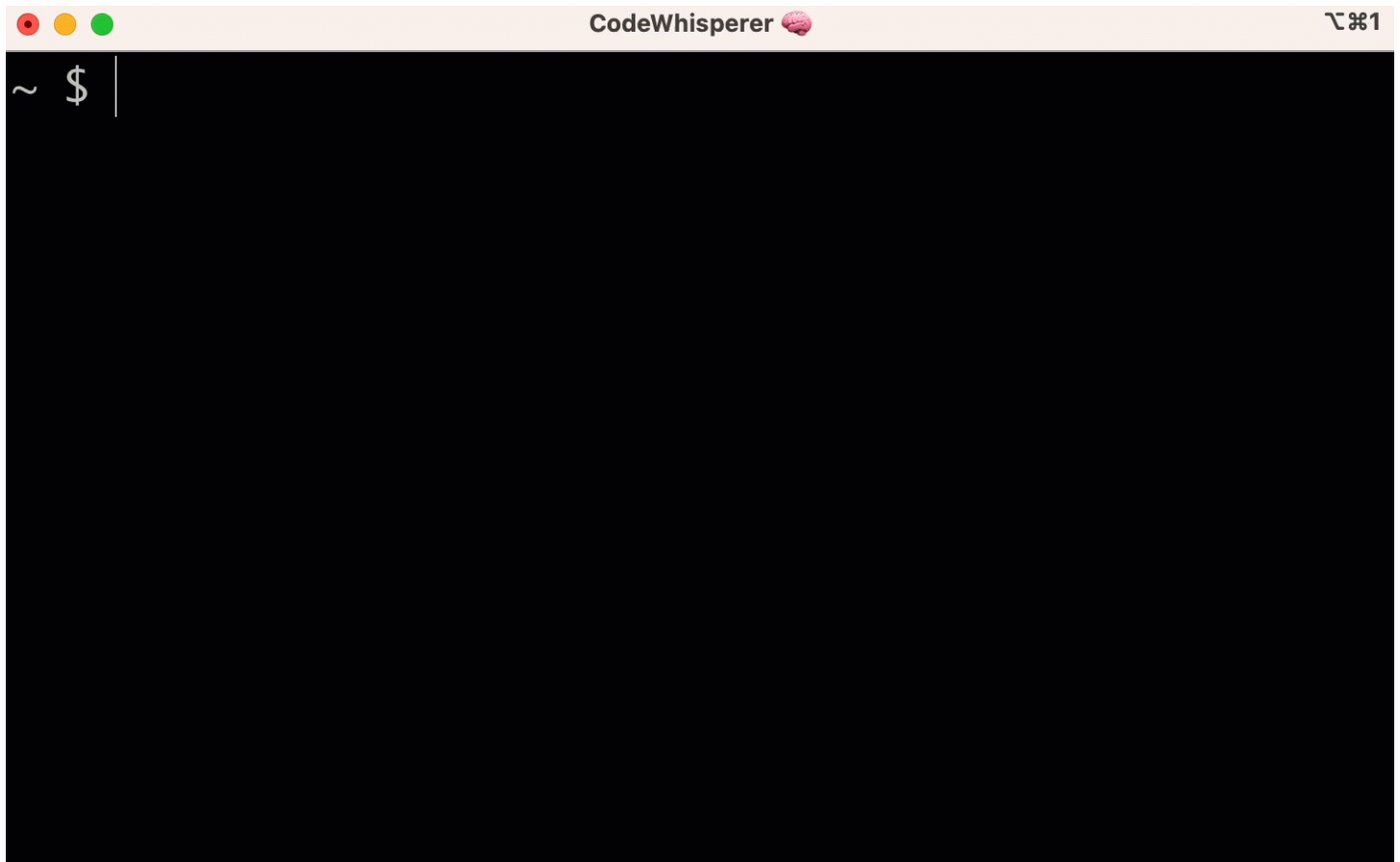
## Natural language to bash translation

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`.

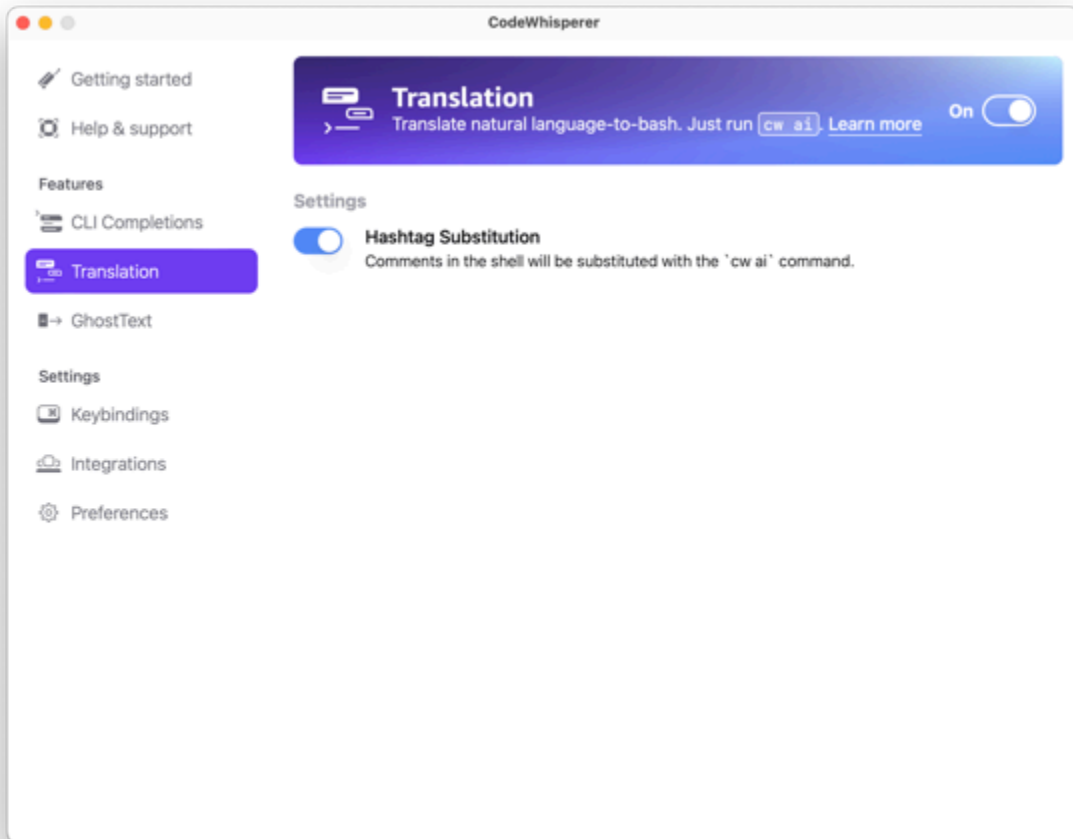
To get started, run either of the following

- `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.

## Adding your own completion specs to Amazon Q Developer

This section discusses about how to build and contribute to your own completion specs.

A completion spec is a declarative schema that specifies the subcommands, options and args for a CLI tool. Amazon Q Developer for command line uses these schemas to generate suggestions.

To edit an existing spec or contribute your own, see <https://fig.io/docs>.

# Customize suggestions

The Amazon Q Customizations feature is in preview, and is subject to change.

## Note

Currently, customizations are only supported for inline code suggestions from Amazon Q.

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.

## 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

The Amazon Q Customizations feature is in preview, and is subject to change.

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 [Setting up 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.

When you use Amazon Q Customizations, your Amazon Q administrator must be authorized to access your codebase, which you may 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.

### 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.

Therefore, before you use Amazon Q Customizations, you must add the following permissions to your Amazon Q administrator's role:

### 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": [
        "*"
    ]
}
]
}

```

## Accessing customization-related messages in Amazon CloudWatch Logs

Amazon Q stores information about the creation of your customization in [Amazon CloudWatch Logs](#).

You can authorize your Amazon Q 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:::*"
    ]
}
]
```

**Note**

The encryption key that you set up for Amazon Q Developer Pro is also used for customizations.

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.

Your data source must contain at least 2 MB, and at most 20 GB, of source code files from supported languages. There is no limit on the number of files, but you must include at least 10 files for each language that you want your customization to support. In the Amazon S3 data source, ensure that all source code is placed within a directory and not at the root level. Any files at the root level will be ignored.

**Note**

Amazon Q Customizations supports the following languages and file extensions:

- Java (.java)
- JavaScript (.js, .jsx)
- Python (.py)
- TypeScript (.ts, .tsx)



# Creating your customization

The Amazon Q Customizations feature is in preview, and is subject to change.

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 VS Code or JetBrains through the AWS plugin.

## Connecting to your data source

The Amazon Q Customizations feature is in preview, and is subject to change.

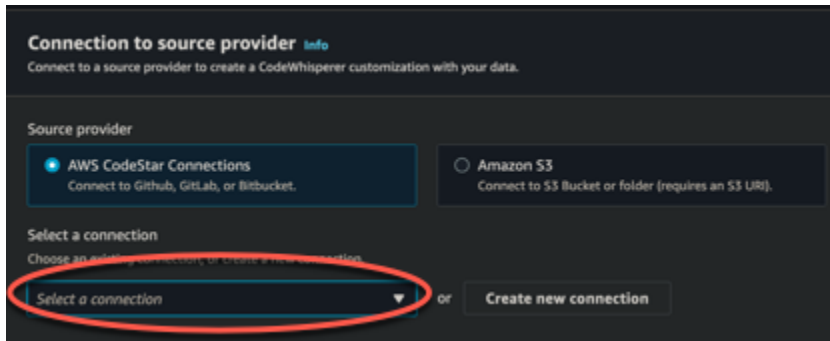
Before you create a customization, you must connect to the data source that contains your codebase. How you do this depends on where your data source is.

If your data source is in Github, GitLab, or Bitbucket, then you must connect to it with AWS CodeConnections. Otherwise, use Amazon S3.

To learn more about CodeConnections, see [What are connections?](#) in the *Developer Tools console User Guide*

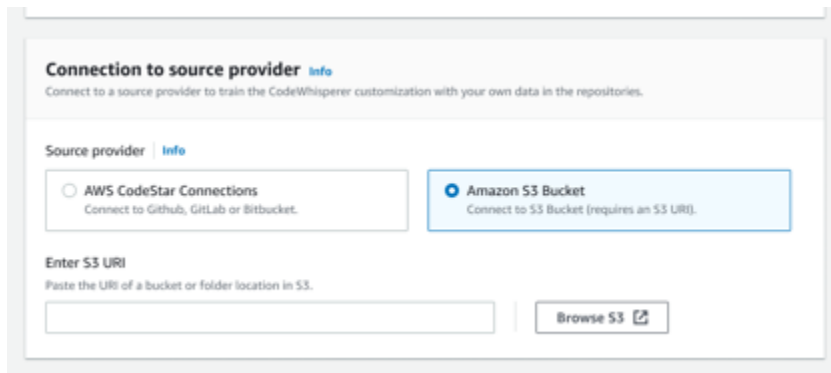
To connect to your data source through CodeConnections, follow this procedure:

1. Under **Connection to source provider**, select CodeConnections.
2. If you are using an existing connection, choose **Select existing connection**. Then, under **Select a connection**, select your connection from the dropdown.



Otherwise, choose **Create a new connection**.

3. In the pop-up window that opens, navigate to your data source and follow the instructions in the console.
4. After you create your data source, return to the **Create customization** page.
5. 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 the bucket or folder containing your codebase and make a note of the URI.

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. The total size of all code files (containing one or more languages) in your data source must be at least 10 MB.

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.

## Deleting your customization

The Amazon Q Customizations feature is in preview, and is subject to change.

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

The Amazon Q Customizations feature is in preview, and is subject to change.

## 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:** This customization is not likely to be useful. 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.

**Note**

Amazon Q automatically filters out non-code files, such as configuration files and text files.

- 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 10 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

Log message	Log level
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
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



Log message	Log level
<p>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.</p>	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
<p>You have activated the maximum number of customizations.</p>	<p>Deactivate an active customization and try again.</p>
<p>You have exceeded the maximum number of group permissions limit of <i>limit</i>.</p>	<p>Remove a group and retry.</p>
<p>You have exceeded the maximum number of user permissions limit of <i>limit</i>.</p>	<p>Remove a user and retry.</p>
<p>Maximum active jobs reached.</p>	<p>Wait until an in-progress job in the same account has finished. Retry the operation.</p>
<p>Encountered an unexpected error when processing the request.</p>	<p>Retry the operation. If it continues to fail, contact customer support.</p>
<p>Access denied when attempting to reach the provided AWS CodeConnections connection.</p>	<p>Validate permissions on your connection and on your third-party provider. Then retry the operation.</p>

Error message	Suggested action
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.
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 <i>size</i> 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

The Amazon Q Customizations feature is in preview, and is subject to change.

## 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**.

The screenshot shows the Amazon CodeWhisperer Customizations console. The top navigation bar includes "Amazon CodeWhisperer" and "Customizations". A blue badge indicates "Active version: 2". The "Details" section shows the customization's name, description, ARN, and latest version (3). The status is "Updated". An evaluation bar shows a score of 4, categorized as "Fair". The evaluation score is 4, and the text states: "This customization's overall effectiveness score is considered fair. AWS recommends that you activate this customization. However, if you see a low acceptance rate after activation, consider adding more data." Below the details are tabs for "Versions", "Groups and users", "Tags", and "Log deliveries". The "Versions" table lists two versions:

Version	Last updated	Source provider	Data reference	Status	Evaluation
3	Nov 8, 2023, 2:40:30 PM	Amazon S3	[Redacted]	Updated	4
2	Nov 8, 2023, 2:23:53 PM	Amazon S3	[Redacted]	Activated	3

To deactivate a customization, choose **Deactivate** from the dropdown.

The screenshot shows the Amazon CodeWhisperer Customizations console. The top navigation bar includes "Amazon CodeWhisperer" and "Customizations". A blue badge indicates "Active version: 2". The "Details" section shows the customization's name, description, ARN, and latest version (2). The status is "Activated". An evaluation bar shows a score of 3, categorized as "Poor". The evaluation score is 3, and the text states: "This customization's overall effectiveness score is considered poor. AWS recommends creating a new customization with more data prior to activation." Below the details are tabs for "Versions", "Groups and users", "Tags", and "Log deliveries". The "Versions" table lists one version:

Version	Last updated	Source provider	Data reference	Status	Evaluation
2	Nov 8, 2023, 2:23:53 PM	Amazon S3	[Redacted]	Activated	3

The "Actions" dropdown menu is open, and the "Deactivate" option is highlighted with a red circle.

## Updating your Amazon Q customizations

The Amazon Q Customizations feature is in preview, and is subject to change.

This section explains how to update a customization with Amazon Q.

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

The screenshot shows the Amazon CodeWhisperer console interface. The main content area displays details for a customization, including its name, description, and ARN. A 'Details' section shows the latest version (1) with a status of 'Created'. An 'Evaluation info' section features a horizontal bar chart with a score of 6, categorized as 'Fair'. Below the chart, the evaluation score is 6, and a note states: 'This customization's overall effectiveness score is considered fair. AWS recommends that you activate this customization. However, if you see a low acceptance rate after activation, consider adding more data.' The 'Versions' table below shows one version (1) with a last updated date of Nov 21, 2023, 4:03:47 PM, source provider of Amazon S3, and status of 'Created'. The 'Actions' dropdown menu is open, showing options: 'Create new version', 'Delete', and 'Deactivate'. On the right, an 'Evaluation' sidebar explains the score: 'Very good 8-10' (AWS recommends activation), 'Fair 4-7' (attempt to optimize), and 'Poor 1-3' (not effective).

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. Select **Create new version** from the **Actions** dropdown.
5. If applicable, 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.

[Browse S3](#)

[Cancel](#) [Create](#)

6. Choose **Create**.

If you receive error messages, see [Troubleshooting the creation of your customization](#).

## Adding users and groups to your Amazon Q customizations

The Amazon Q Customizations feature is in preview, and is subject to change.

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 [Subscriptions](#).

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

The Amazon Q Customizations feature is in preview, and is subject to change.

This section contains information about how to use customizations as a developer.

Amazon Q only supports customizations in VS Code and JetBrains IDEs.

### AWS Toolkit for Visual Studio Code

To use customizations with VS Code:

1. Authenticate to Amazon Q Developer Pro with IAM Identity Center using the steps under [Install Amazon Q Developer 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.

### AWS Toolkit for JetBrains

To use customizations with JetBrains:

1. Authenticate to Amazon Q Developer Pro with IAM Identity Center using the steps under [Install Amazon Q Developer 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**.

# Dashboard

The screenshot shows the Amazon Q Developer dashboard with the following components:

- Navigation:** AWS logo, Services menu, search bar, [Alt+S] shortcut, region (N. Virginia), and user profile (MyRole/AWSUser @ 0123-4567-8901).
- Dashboard Header:** "Amazon Q in the IDE > Dashboard" and "Dashboard" title with a close button.
- Filters:** "Active customization" (All) and "Programming language" (All) dropdowns, a "Clear filters" button, and a "Data range" selector (2023-10-27 — 2023-11-10).
- User activity:** A line chart showing "Active subscriptions" (blue) and "Daily active users" (red) from 10/29/2023 to 11/10/2023. The y-axis is labeled "Count" and ranges from 0 to 20.
- Code impact:**
  - Lines of code generated by Amazon Q in the IDE: **144.85 million**
  - Acceptance rate: **45.32%** (highlighted with a red box)
  - Accepted recommendations with references: **4.83 million**
- Security scans:** A table showing the number of scans for different languages.
 

Language	Number of scans
Java	1.2 million
Javascript	-
Python	-
- Right Sidebar:**
  - User activity:** Text explaining that the section indicates how many Amazon Q in the IDE seats are being used on a daily basis.
  - Code impact:** Text explaining that the lines of code generated simply indicates how many lines of code were suggested and accepted.
  - Security scans:** Text explaining that the security scan tool helps identify security vulnerabilities in developers' code.
  - Learn more:** Links to "Amazon Q in the IDE dashboard", "Security scan", and "Code references".
- Footer:** CloudShell, Feedback, Language, and copyright information (© 2024, Amazon Web Services, Inc. or its affiliates. Privacy, Terms, Cookie preferences).

Available only for Amazon Q Developer administrators, the Amazon Q Developer dashboard summarizes useful data about how your developers use the service. Among the useful metrics is the acceptance rate, which indicates how often you take inline suggestions from Amazon Q.

You can filter the data in the dashboard by date range. The minimum range is two weeks and the maximum is one year. You can also filter by programming language.

To view metrics on the dashboard, you must have the `cloudwatch:GetMetricData` and `cloudwatch:listMetrics` permissions. This permission is granted to administrators as part of [assigning Amazon Q administration rights](#).



## User activity

The **User activity** section indicates how many Amazon Q seats you are paying for, and how many of those seats are being used on a daily basis. The difference between the two is the number of subscriptions you are paying for that are not being used.

## 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.

# Amazon Q Developer and other AWS services

Amazon Q Developer is available as a feature in several other AWS services. This section explains some of the ways you can access Amazon Q within AWS.

## AWS services

- [Amazon Q Business](#)
- [Amazon Q in AWS Chatbot](#)
- [Amazon Q in Amazon CodeCatalyst](#)
- [Amazon Q in Amazon Connect](#)
- [Amazon Q in Amazon EC2](#)
- [Amazon Q in AWS Glue](#)
- [Amazon Q in VPC Reachability Analyzer](#)
- [Amazon Q in Amazon QuickSight](#)

## Amazon Q Business

Amazon Q Business is a fully managed, generative AI-powered enterprise chat assistant that you can deploy within your organization. For more information, see the [Amazon Q Business User Guide](#).

## 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 [Chat 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*.

## Amazon Q in Amazon CodeCatalyst

Amazon Q feature development capability in Amazon CodeCatalyst is a generative AI assistant to which you can assign issues. After an issue is assigned to Amazon Q, it analyzes the issue based on its title and description and reviews the code in the specified repository. If Amazon Q can create an approach, it creates a draft solution for users to evaluate in a pull request. For more information, see [Tutorial: Using CodeCatalyst generative AI features to speed your development work](#) in the *Amazon CodeCatalyst User Guide*.

## Amazon Q in Amazon Connect

Amazon Q in Amazon Connect is a generative AI customer service assistant that is a large language model (LLM)-enhanced evolution of Amazon Connect Wisdom. Amazon Q in Amazon Connect delivers real-time recommendations to help contact center agents resolve customer issues quickly and accurately. For more information, see [Use Amazon Q in Amazon Connect](#) in the *Amazon Connect Administrator Guide*.

## Amazon Q in Amazon EC2

In the Amazon Elastic Compute Cloud (Amazon EC2) console, you can use the Amazon Q EC2 instance type selector to provide guidance and suggestions for EC2 instance types that are best suited to your new workloads. For more information, see [Get instance type recommendations for a new workload](#) in the *Amazon EC2 User Guide*.

## Amazon Q in AWS Glue

Amazon Q data integration in AWS Glue is a generative AI-powered assistant that enables you to build data integration jobs in AWS Glue using natural language. With Amazon Q data integration, you can author data integration jobs, troubleshoot issues, and get answers to AWS Glue data integration questions. For more information, see [Amazon Q data integration in AWS Glue](#) in the *AWS Glue User Guide*.

## Amazon Q in VPC Reachability Analyzer

When you ask Amazon Q to help you fix a network connectivity problem, it works with VPC Reachability Analyzer to check your connections, inspect your network configuration, and identify potential issues. Then Amazon Q provides guidance on how to resolve or further diagnose the issue. Using the conversational capabilities of Amazon Q and Reachability Analyzer together creates an intuitive way for you to fix network connectivity problems quickly. For more information, see [Amazon Q network troubleshooting](#) in the *Reachability Analyzer User Guide*.

## Amazon Q in Amazon QuickSight

Amazon Q in Amazon QuickSight is optimized to understand natural language that you use to ask questions about your data. You can ask Amazon Q questions that use common phrases related to

sales, marketing, and retail to meet your business needs quickly. For more information, see [Amazon Q in Amazon QuickSight](#) in the *Amazon QuickSight User Guide*.

# 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

Regardless of where you use Amazon Q Developer, data is sent to and stored in an AWS Region in the US. Your conversations with Amazon Q are stored in the US East (N. Virginia) 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. Data processed during interactions with Amazon Q in integrated development environments (IDEs) is stored in the US East (N. Virginia) Region.

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. 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](#).

## Topics

- [Data encryption in Amazon Q Developer](#)
- [Amazon Q Developer service improvement](#)
- [Opt out of data sharing in the IDE](#)

## 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 data stored by Amazon Q in integrated development environments (IDEs), you can create your own customer managed AWS KMS key to encrypt your data at rest. 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. For information on creating your own KMS key, see [Creating keys](#) in the *AWS Key Management Service Developer Guide*.

## 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, and does not contain personally identifiable information (PII) such as your IP address.

Choose an environment 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.

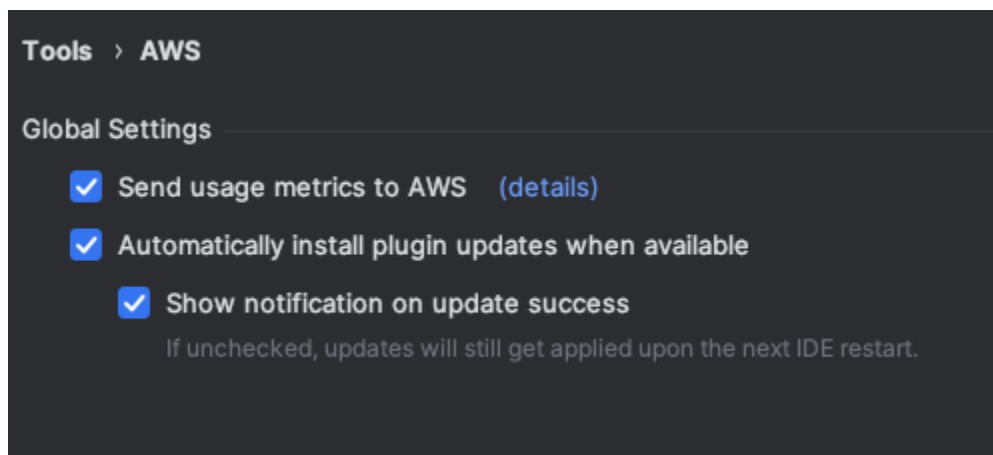
**Note**

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.

## 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.

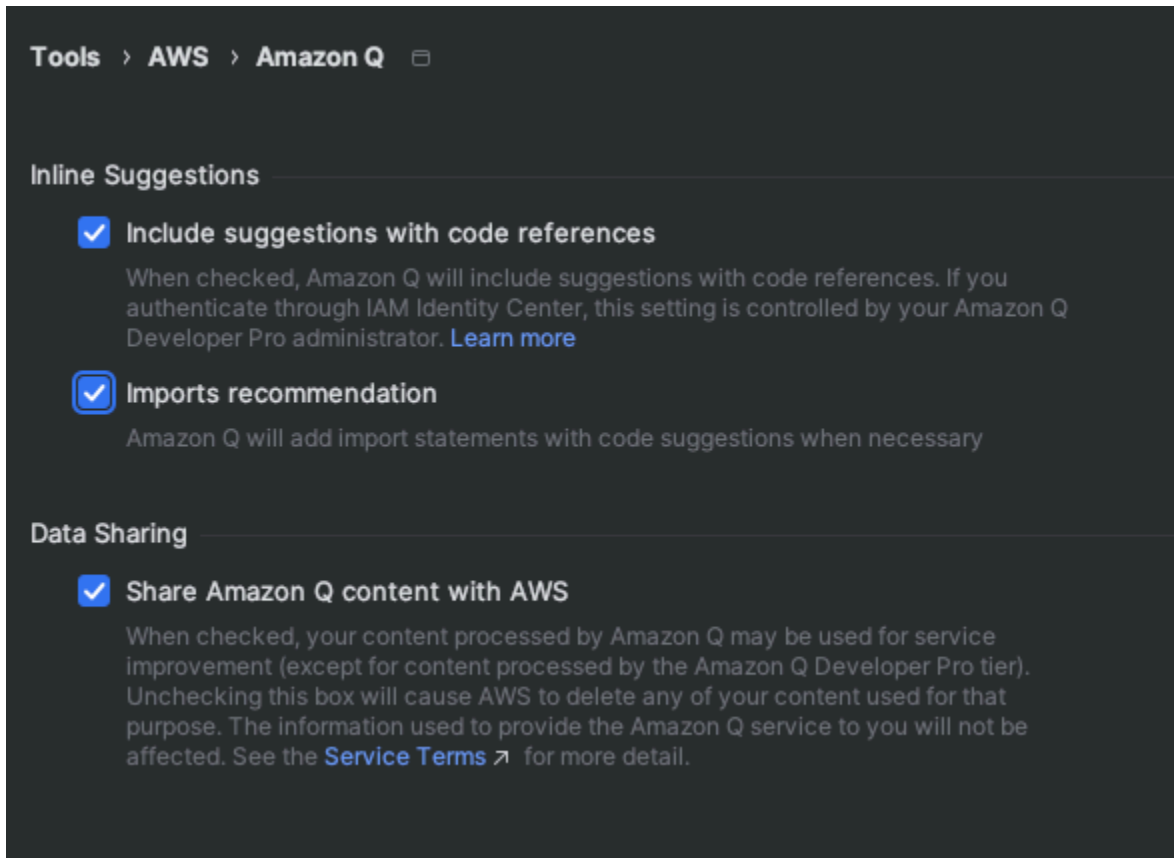
1. Open **Settings** in VS Code.
2. If you are using VS Code workspaces, switch to the **Workspace** sub-tab. In VS Code, workspace settings override user settings.
3. In the Settings search bar, enter Amazon Q: Share Content.
4. Deselect the box.

## 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**.



## 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**.

# 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
<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 [Creating IAM 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 [How IAM roles differ from resource-based policies](#) 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 [What is ABAC?](#) 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 [Switching to a 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 [Creating 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, AWS Chatbot, and in IDEs. Other services integrated with Amazon Q might require different policies or settings. 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 [Amazon Q Developer permissions reference](#).

### Topics

- [Policy best practices](#)
- [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 [IAM Access Analyzer policy validation](#) 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 [Configuring MFA-protected API access](#) in the *IAM User Guide*.

For more information about best practices in IAM, see [Security best practices in IAM](#) 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": [
        "q:*"
      ],
      "Resource": "*"
    }
  ]
}
```

## Data perimeters for Amazon Q resources

For some features, Amazon Q uploads artifacts to AWS service-owned Amazon S3 buckets. If you are using data perimeters 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 describes each of the Amazon S3 buckets that Amazon Q requires access to and the features that use each bucket.

Amazon S3 bucket ARN	Description
<code>arn:aws:s3:::amazonq-code-scan-us-east-1-29121b44f7b</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>	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>	An Amazon S3 bucket used to upload 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.

For a list of all Amazon Q permissions you can control with policies, see the [Amazon Q Developer permissions reference](#).

### 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 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"
      ],
      "Resource": "*"
    }
  ]
}

```

## Allow Amazon Q to perform actions on your behalf

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 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: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 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: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 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: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": "*"
  }
]
}

```

## Deny access to Amazon Q

The following example policy denies all permissions to use 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": [
        "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:ListGroupsWithUser",
      "iam:ListAttachedUserPolicies",
      "iam:ListUserPolicies",
      "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
  },
  {
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
      "iam:GetGroupPolicy",
      "iam:GetPolicyVersion",
      "iam:GetPolicy",
      "iam:ListAttachedGroupPolicies",
      "iam:ListGroupPolicies",
      "iam:ListPolicyVersions",
      "iam:ListPolicies",
      "iam:ListUsers"
    ],
    "Resource": "*"
  }
]
}

```

## Amazon Q Developer permissions reference

You can use the following table 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).

### Important

To chat with Amazon Q, an IAM identity needs permissions for the following actions:

- StartConversation

- `SendMessage`
- `GetConversation` (console only)
- `ListConversations` (console only)

To troubleshoot console errors with Amazon Q, an IAM identity needs permissions for the following actions:

- `StartTroubleshootingAnalysis`
- `GetTroubleshootingResults`
- `StartTroubleshootingResolutionExplanation`
- `UpdateTroubleshootingCommandResult`

If one of these actions isn't explicitly allowed by an attached policy, an IAM permissions error is returned when you try to use Amazon Q.

The following table shows the Amazon Q Developer permissions that you can allow or deny access to in policies.

## 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](#)

- [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](#)

## Other useful permissions

In addition to Amazon Q permissions, you might need the following permissions to monitor or grant access to Amazon Q features.

- `codewhisperer:GenerateRecommendations` - Gets code suggestions in Amazon Q for AWS Cloud9 and AWS Lambda.
- `codewhisperer:GenerateCompletions` - Gets code suggestions in Amazon Q for VS Code and JetBrains.
- `codewhisperer:StartCodeAnalysis` - Starts a security scan in Amazon Q for VS Code and JetBrains.
- `codewhisperer:GetCodeAnalysis` - Gets the status of an ongoing security scan.
- `codewhisperer:ListCodeAnalysisFindings` - Called after `GetCodeAnalysis` signals job completions. Returns the list of all security issues in the files scanned.
- `codewhisperer>CreateUploadUrl` - Creates the URL to upload the code files that will be scanned in Amazon Q for VS Code and JetBrains.
- `codewhisperer>CreateProfile` - Called when a CodeWhisperer administrator creates a new CodeWhisperer application.
- `codewhisperer:UpdateProfile` - Called when the CodeWhisperer administrator updates CodeWhisperer profiles.
- `codewhisperer:ListProfiles` - Called when the CodeWhisperer administrator lists CodeWhisperer profiles.
- `codewhisperer:TagResource` - Called when the CodeWhisperer administrator adds or creates a tag on the CodeWhisperer resource.
- `codewhisperer:UntagResource` - Called when the CodeWhisperer administrator removes a tag from the CodeWhisperer resource.
- `codewhisperer:ListTagsForResource` - Called by CodeWhisperer on console page load to list tags on the CodeWhisperer resource.
- `codewhisperer:StartDataCollection` - Used for starting a data collection from customer's data source to use in creating a Customization.

- `codewhisperer:GetDataCollectionStatus` - Used for polling the status of a data collection job.
- `codewhisperer:CreateCustomization` - Used for creating a Customization from collected customer data.
- `codewhisperer>DeleteCustomization` - Used for deleting a Customization from collected customer data.
- `codewhisperer:ListCustomizations` - Used for listing Customizations based on their state.
- `codewhisperer:UpdateCustomization` - Activates or deactivates a customization.
- `codewhisperer:ListCustomizationVersions` - Lists the versions of a customization.
- `codewhisperer:GetCustomization` - Used for describing a Customization.

## 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 policy for Amazon Q Developer can be attached to IAM identities:

- `AmazonQFullAccess` provides full access to enable interactions with Amazon Q Developer.

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 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAmazonQFullAccess",
      "Effect": "Allow",
      "Action": [
        "q:*"
      ],
      "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 [AWSServiceRoleForUserSubscriptionPolicy](#).

## 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="#">AmazonQFullAccess</a> - Updated policy	Additional permissions have been added to enable subscriptions checks for Amazon Q Developer.	April 30, 2024
<a href="#">AWSServiceRoleForSubscriptionPolicy</a> - New policy	Allows Amazon Q Subscriptions to automatically update subscriptions from changes in AWS IAM Identity Center, AWS IAM Identity Center directory and AWS Organizations on your behalf.	April 30, 2024
<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

Region name	Region identity	Support in Amazon Q Developer
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
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

Region name	Region identity	Support in User Subscriptions
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.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, `codewhisperer.us-east-1.amazonaws.com`.

For more information, see [Accessing a service through an interface endpoint](#) in the *Amazon VPC User Guide*.

### 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 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": "AIDA60N6E4XEGIEEXAMPLE",
    "arn": "arn:aws:iam::555555555555:user/Mary",
    "accountId": "555555555555",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",

```

```

    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDA60N6E4XEGIEEXAMPLE",
        "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": "AIDA60N6E4XEGIEEXAMPLE",
    "arn": "arn:aws:iam::555555555555:user/Paulo",
    "accountId": "555555555555",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",

```

```
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDA60N6E4XEGIEEXAMPLE",
        "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.

Dimension	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 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 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 in multiple accounts within the same organization, you will only be billed for their seat in the first account in which they use the service.

# 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](#).

Regardless of where you use Amazon Q Developer, data is sent to and stored in an AWS Region in the US. Your conversations with Amazon Q are stored in the US East (N. Virginia) 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. Data processed during interactions with Amazon Q in integrated development environments (IDEs) is stored in the US East (N. Virginia) Region.

Amazon Q Developer in the AWS Management Console, AWS Console Mobile Application, AWS website, AWS Documentation website, and AWS Chatbot is available in the following AWS Regions. Even if you set your console to one of these Regions, your data will still be processed in a US Region.

- 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)

## Opt-in Regions

You can chat with Amazon Q Developer [on AWS apps and websites](#) in an *opt-in Region* (Region that is disabled by default). To use an opt-in Region with Amazon Q Developer, you must enable the Region manually. For more information about AWS Regions, see [Specify which AWS Regions your account can use](#) in the *AWS Account Management Reference Guide*. The following opt-in Regions are supported. Even if you set your console to one of these Regions, your data will still be processed in a US Region.

- 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)

# 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="#">Updated managed policy: AmazonQFullAccess</a>	Additional permissions have been added to the <a href="#">AmazonQFullAccess</a> policy.	April 30, 2024
<a href="#">New service-linked role: AWSServiceRoleForUserSubscriptions</a>	The <a href="#">AWSServiceRoleForUserSubscriptions</a> provides access for User Subscriptions to your IAM Identity Center resources to automatically update your subscriptions.	April 30, 2024
<a href="#">New service-linked role: AWSServiceRoleForAmazonQDeveloper</a>	The <a href="#">AWSServiceRoleForAmazonQDeveloper</a> grants permission to access and emit data, and to create reports.	April 30, 2024
<a href="#">New managed policy: AWSServiceRoleForUserSubscriptionPolicy</a>	The <a href="#">AWSServiceRoleForUserSubscriptionPolicy</a> Allows principals to track IAM Identity Center directory and AWS Organizations changes.	April 30, 2024
<a href="#">New managed policy: AWSServiceRoleForAmazonQDeveloperPolicy</a>	The <a href="#">AWSServiceRoleForAmazonQDeveloperPolicy</a> allows Amazon Q Developer to call CloudWatch and CodeGuru on your behalf.	April 30, 2024

---

<a href="#">GA release</a>	Amazon Q Developer is available for general audiences.	April 30, 2024
<a href="#">Amazon CodeWhisperer merge</a>	Amazon CodeWhisperer is now a part of Amazon Q Developer.	April 30, 2024
<a href="#">New guide name</a>	This service and accompanying user guide have been renamed Amazon Q Developer.	March 29, 2024
<a href="#">New permission</a>	The <a href="#">ListConversations action</a> is required to chat with Amazon Q in the console.	March 5, 2024
<a href="#">New data protection topic</a>	Amazon Q now uses content for <a href="#">service improvement purposes</a> .	January 25, 2024
<a href="#">New topic</a>	Added instructions for how to <a href="#">add Amazon Q to Slack and Microsoft Teams channels</a> that are configured with AWS Chatbot.	January 18, 2024
<a href="#">Preview release</a>	This is the initial preview release of the <i>Amazon Q Developer User Guide</i> .	November 28, 2023