
CodeWhisperer

User Guide



CodeWhisperer: User Guide

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

What is CodeWhisperer?	1
Setting up	2
Choosing your IDE	2
Installing or updating your IDE	3
Installing the AWS Toolkit	3
Choosing your authentication method	3
Setting up your authentication method	3
Get an AWS account and your root user credentials	4
CodeWhisperer Professional	4
With Organizations	5
With IAM Identity Center administration	6
Administering end users	10
CodeWhisperer profiles	14
Getting started	18
With VS Code and JetBrains	18
Professional tier	18
Individual tier	22
Authenticating	27
With Amazon SageMaker Studio	28
With JupyterLab	29
With AWS Glue Studio	31
With AWS Lambda	31
Enabling IAM permissions for CodeWhisperer	31
Activating Amazon CodeWhisperer with Lambda	32
With AWS Cloud9	32
Enabling IAM permissions for CodeWhisperer	32
Activating Amazon CodeWhisperer with AWS Cloud9	33
Features	34
Dashboard	35
User activity	36
Code impact	36
Security scans	36
User actions	36
Useful APIs	39
Languages and IDEs	40
Languages	40
IDE support	41
Pausing suggestions	43
Security scans	46
Activating security scans	48
Security scan data limits	51
Code references	52
Viewing code references	52
Toggling code references	55
Opting out	58
Types of users	60
Root user	60
IAM Identity Center administrator	60
CodeWhisperer administrator	60
Professional-tier developer	60
Individual-tier developer	61
In-console developer	61
Billing	62
Individual tier	62

Professional tier	62
Inside the AWS console	63
Monitoring	64
Monitoring with CloudWatch	64
Code examples	66
Single-line code completion	66
Full function generation	67
Block completion	74
Docstring and Javadoc completion	75
Line-by-line recommendations	76
Single-line comment completion	77
Sharing data	79
Opting out of sharing your client-side telemetry	79
Opting out of sharing your content	82
Quotas	85
Security	86
Resilience	86
Vulnerability analysis and management	87
Administrative security	87
Data protection	87
Data Protection for Amazon CodeWhisperer	87
Compliance validation	89
Security best practices	89
Infrastructure security	89
Identity and Access Management	90
Audience	90
Authenticating with identities	90
Managing access using policies	93
How Amazon CodeWhisperer works with IAM	94
Identity-based policy examples	99
AWS managed policies	101
Troubleshooting	104
Using service-linked roles	105
VPC endpoints (AWS PrivateLink)	108
Considerations for CodeWhisperer VPC endpoints	108
Prerequisites	108
Creating an interface VPC endpoint for CodeWhisperer	108
Using an on-premises computer to connect to a CodeWhisperer endpoint	109
Using an in-console IDE to connect to a CodeWhisperer endpoint	109
Connecting to CodeWhisperer through AWS PrivateLink from a third-Party IDE on an Amazon EC2 instance	109
Document history	111

What is CodeWhisperer?

Note

The fastest way to start using CodeWhisperer is to [authenticate with AWS Builder ID as an individual developer \(p. 22\)](#). You don't need an AWS account to do this.

Amazon CodeWhisperer is a general purpose, machine learning-powered code generator that provides you with code recommendations in real time. As you write code, CodeWhisperer 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.

CodeWhisperer can also scan your code to highlight and define security issues.

For information about which programming languages CodeWhisperer supports, see [Language support \(p. 40\)](#).

Setting up

The following sections describe the steps you need to take before using CodeWhisperer **as a developer** for the first time.

If you are an **administrator** who is setting up CodeWhisperer Professional for your organization, see [Setting up Amazon CodeWhisperer for administrators \(p. 4\)](#).

Before you use CodeWhisperer for the first time, you must follow the following steps:

1. Choose your IDE.
2. Install or update your IDE (if applicable).
3. Install or update the AWS Toolkit (if applicable).
4. Choose your authentication method.
5. Set up your Builder ID, IAM Identity Center, or IAM credentials.

Choosing your IDE

CodeWhisperer acts as an enhancement to certain integrated development environments (IDEs). You can use it within any of the following services.

- [Amazon SageMaker notebooks](#) serve as an essential component of the SageMaker Studio interactive development environment, offering a managed JupyterLab environment to create, share, and collaborate on Jupyter notebooks. Designed to support machine learning workflows within AWS, Studio Notebooks provide built-in version control and collaboration functionalities. They facilitate integration with SageMaker and other AWS services, allowing users to build, train, and deploy models directly from their notebooks. Additionally, SageMaker Studio notebooks automatically scale underlying resources according to workload requirements, ensuring efficient resource utilization.
- JupyterLab is an IDE that allows you to work with data and code in a flexible, open-source platform. With JupyterLab, you can create and edit Jupyter notebooks, run code in various programming languages, and visualize and manipulate data using a range of libraries and tools. JupyterLab is widely used in data science, machine learning, and scientific research, and is supported by a vibrant community of contributors and users.
- The [AWS Toolkit for Visual Studio Code](#) is an open source plug-in for Visual Studio Code that makes it easier to create, debug, and deploy applications on Amazon Web Services. With the AWS Toolkit for Visual Studio Code, you will be able to get started faster and be more productive when building applications with Visual Studio Code on AWS. The toolkit provides an integrated experience for developing serverless applications, including assistance for getting started, ML-powered code recommendations, step-through debugging, and deploying from the IDE.
- The [AWS Toolkit for JetBrains](#) is an open source plug-in for the IDEs from JetBrains that makes it easier for developers to develop, debug, and deploy serverless applications that use Amazon Web Services. It includes features like credentials management and AWS Region management that simplify writing applications for Amazon Web Services.
- [AWS Cloud9](#) is a cloud-based IDE that lets you write, run, and debug your code with just a browser. It includes a code editor, debugger, and terminal. AWS Cloud9 comes prepackaged with essential tools for popular programming languages, including JavaScript, Python, and PHP.

- [AWS Lambda](#) is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. You can trigger Lambda from over 200 AWS services and software as a service (SaaS) applications, and only pay for what you use.

Installing or updating your IDE

To install VS Code for the first time, use [the VS Code download page](#).

If you already have VS Code installed, update to the latest version as follows:

- On MacOS, choose Code -> Check for Updates.
- On Windows and Linux, choose Help -> Check for Updates.

To install JetBrains for the first time, use [the JetBrains download page](#).

If you already have JetBrains installed, update to the latest version as follows:

- On MacOS, from the IDE's main dropdown menu, choose **Check for Updates**.
- On Windows and Linux, choose **Help -> Check for Updates**.

Installing the AWS Toolkit

In order to use CodeWhisperer with VS Code or JetBrains, you must first download and install the AWS Toolkit.

For information about installing the AWS Toolkit for VS Code, see [Setting Up the AWS Toolkit for Visual Studio Code](#) in the [AWS Toolkit for Visual Studio Code user guide](#).

For information about installing the AWS Toolkit for JetBrains, see [Setting Up the AWS Toolkit for JetBrains](#) in the [AWS Toolkit for JetBrains user guide](#).

Choosing your authentication method

If you plan to use CodeWhisperer with VS Code (through AWS Toolkit) or JetBrains (through AWS Toolkit), you will have to authenticate using either AWS Builder ID or IAM Identity Center.

If you plan to use CodeWhisperer with AWS Cloud9 or AWS Lambda, you will have to authenticate using IAM.

For information about authenticating with CodeWhisperer, see [Authenticating with CodeWhisperer and AWS Toolkit \(p. 27\)](#).

Setting up your authentication method

[Builder ID](#) (used with AWS Toolkit and VS Code or JetBrains) requires only an email address. To use it, you don't even need an AWS account.

[IAM Identity Center](#) requires setup by your enterprise administrator.

[IAM](#) credentials are used within your AWS account to regulate access to, and between, various AWS services.

Get an AWS account and your root user credentials

To access AWS, you must sign up for an AWS account.

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](#).

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to <https://aws.amazon.com/> and choosing **My Account**.

Setting up Amazon CodeWhisperer for administrators

This section describes the setup procedures that are necessary before a developer can use CodeWhisperer Professional.

In this context, a *professional* developer is a developer who works for a business (enterprise) that has an AWS account.

Here is a summary of the procedures described on this page. If you are a regular AWS user, then you may have already completed one or more of these procedures in connection with another AWS service.

- The root user comes built-in with your AWS account.
- The root user creates the permission set for the AWS Organizations administrator.
- The root user adds that permission set to the Organizations administrator.
- The Organizations administrator adds users.
- The Organizations administrator authorizes the CodeWhisperer administrator to manage CodeWhisperer.
- The CodeWhisperer administrator authorizes the enterprise developers to use CodeWhisperer.

For more information about the different personas that may use CodeWhisperer see [Types of users for CodeWhisperer \(p. 60\)](#).

Setting up CodeWhisperer Professional with AWS Organizations administration

The root user comes built-in with your AWS account

The root user is the user that comes with your account. The root user has access to all services and account configurations.

Because the root user is so powerful, it is a best practice to use it as seldom as possible. However, one useful function of the root user is to create a powerful administrative user.

In this case, we will use the root user to create the Organizations administrator.

The root user creates the Organizations administrator

Permission sets are stored in IAM Identity Center and define the level of access that users and groups have to an AWS account. Perform the following steps to create a permission set that grants administrative permissions.

1. Sign in to the [AWS Management Console](#) as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.
2. Open the [IAM Identity Center console](#).
3. If this is the first time you're using IAM Identity Center, choose **Enable**. Then choose **Create AWS organization**. If you have previously enabled IAM Identity Center, then you can skip this step.
4. In the IAM Identity Center navigation pane, under **Multi-account permissions**, choose **Permission sets**.
5. Choose **Create permission set**.
6. On the **Select permission set type** page, keep the default settings and choose **Next**. The default settings grant full access to AWS services and resources using the **AdministratorAccess** predefined permission set.

Note

The predefined **AdministratorAccess** permission set uses the **AdministratorAccess** AWS managed policy.

7. On the **Specify permission set details** page, keep the default settings and choose **Next**. The default setting limits your session to one hour.
8. On the **Review and create** page, do the following:
 1. Review the permission set type and confirm that it is **AdministratorAccess**.
 2. Review the AWS managed policy and confirm that it is **AdministratorAccess**.
 3. Choose **Create**.

The root user assigns the Organizations administrator permission set to a user

In the last section, you created the **AdministratorAccess** permission set. Now you must assign that permission set to a user.

1. In the the [IAM Identity Center console](#), on the **AWS accounts** page, a tree view list of your organization appears. Select the check box next to the AWS account to which you want to assign administrative access. If you have multiple accounts in your organization, select the check box next to the management account.
2. Choose **Assign users or groups**.

3. If necessary, select the **Users** tab.
4. Choose **Create users**. A new browser tab will open with the **Users** page.
5. Choose **Add user**.
6. On the **Specify user details** page, fill out the fields with information about the user who will be your account administrator. An example username might be *account_admin*.
Then choose **Next**.
7. On the **Add user to groups** page, add this user to a group if you like, and then choose **Next**.
8. On the **Review and add user** page, review the information that you have entered, and select **Add user**.
9. If you chose to use a one-time password, then a pop-up window will display your one-time password.

Copy this password to a secure location on your local computer.

Choose **Close**.

10. Return to the previous browser tab. with the **Assign users and groups to "AWS-account-name"** at the top of the page.
11. Choose the refresh button or refresh the browser tab. The user you recently created should appear in the list.
12. Select the checkbox next to the name of the user who will become the account administrator.
13. Choose **Next**.
14. On the **Assign permission sets to "AWS-account-name"** page, under **Permission sets**, select the **AdministratorAccess** permission set.
15. Choose **Next**.
16. On the **Review and submit assignments to "AWS-account-name"** page, choose **Submit**.

Important

The user assignment process might take a few minutes to complete. Leave this page open until the process successfully completes.

17. While you're still in the IAM Identity Center, from the navigation bar on the left, choose **Dashboard**.
18. From the **Settings summary** on the right side of the page, copy the AWS access portal URL.

This URL will be used by the account administrator and the CodeWhisperer administrator when they log in to IAM Identity Center.

It will also be used by the CodeWhisperer Professional developer when they authenticate through VS Code or JetBrains. In that context, it is called the Start URL, as discussed in [Setting up CodeWhisperer for professional developers \(p. 18\)](#).

Setting up CodeWhisperer Professional with IAM Identity Center

Delegate IAM Identity Center administration to a non-management account

You must administer CodeWhisperer from the same account that you use to administer IAM Identity Center. And as a matter of best practices, you should not administer IAM Identity Center from your management account.

Therefore, you should use [Delegated administration](#) to designate a non-management account for administering IAM Identity Center. Then, you should also use that account to administer CodeWhisperer.

If you only have one account in your AWS organization, then that is the management account. You should create another account to use for administering IAM Identity Center and CodeWhisperer. You can learn about best practices for creating and maintaining multiple AWS accounts in the [AWS Account Management Reference Guide](#).

After you choose the account that will become your delegated administrator account, follow the steps under [Register a member account](#) in the *IAM Identity Center User Guide*.

Assigning CodeWhisperer administration rights

Warning

In this procedure, you are acting as the Organizations administrator, logged into the delegator administrator account. Depending on how you were logged in for the previous procedures, you may need to switch users, accounts, and/or roles before continuing.

The administrator of your CodeWhisperer profile is a special user with the right to change the settings in the CodeWhisperer profile, and to manage the access of, or add, users and groups to CodeWhisperer.

To promote a user to CodeWhisperer administrator, the account administrator uses the following procedures.

Note

This procedure assumes that you already have a user whom you want to promote to CodeWhisperer administrator. If you don't, then create one through the procedures described in [Assign users and groups to IAM Identity Center \(p. 11\)](#).

Setting up the policies for a CodeWhisperer administrator

1. Open a browser tab with the access portal URL given to you by the root user, and log in as the account administrator.
2. Under **Multi-account permissions**, choose **Permission sets**.
3. Choose **Create permission set**.
4. Under **Permission set type**, select **Custom permission set**.
5. Choose **Next**.
6. Expand the **Inline policy** window.
7. Erase the brackets in the box.
8. Paste the following text into the box:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "sso-directory:SearchUsers",  
                "sso-directory:SearchGroups",  
                "sso-directory:GetUserPoolInfo",  
                "sso-directory:DescribeDirectory",  
                "sso>ListApplicationInstances",  
                "sso-directory>ListMembersInGroup"  
            ],  
            "Resource": [  
                "*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iam>ListRoles"  
            ]  
        }  
    ]  
}
```

```
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "pricing:GetProducts"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "sso>ListProfileAssociations",
            "sso>ListProfiles",
            "sso>GetSharedSsoConfiguration",
            "sso>ListDirectoryAssociations",
            "sso>DescribeRegisteredRegions",
            "sso>GetSsoConfiguration",
            "sso>GetApplicationInstance",
            "sso>GetManagedApplicationInstance",
            "sso>UpdateSsoConfiguration",
            "sso>AssociateDirectory",
            "sso>AssociateProfile",
            "sso>DisassociateProfile",
            "sso>GetProfile",
            "sso>GetSSOStatus",
            "sso>CreateManagedApplicationInstance",
            "sso>DeleteManagedApplicationInstance"
        ],
        "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>GetEncryptionContextSharing"
        ],
        "Resource": [
            "*"
        ]
    }
]
```

```
        "kms:RetireGrant",
        "kms:DescribeKey"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "codeguru-security:UpdateAccountConfiguration"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "iam>CreateServiceLinkedRole"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "codewhisperer>CreateProfile",
        "codewhisperer>UpdateProfile",
        "codewhisperer>ListProfiles",
        "codewhisperer>DeleteProfile",
        "codewhisperer>TagResource",
        "codewhisperer>UnTagResource",
        "codewhisperer>ListTagsForResource"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:GetMetricData"
    ],
    "Resource": [
        "*"
    ]
}
]
```

9. Choose **Next**.
10. Under **Permission set name**, enter **CodeWhisperer_administrator**.
11. Choose **Next**.
12. On the **Review and create** page, choose **Create**.

Attaching the policies for a CodeWhisperer administrator to a user

Warning

In this procedure, you are acting as the Organizations administrator, logged into the delegated administrator account. Depending on how you were logged in for the previous procedures, you may need to switch users, accounts, and/or roles before continuing.

1. Open a browser tab with the access portal URL given to you by the root user, and log in as the account administrator.
2. From the main console page, choose **IAM Identity Center**.
3. In the navigation pane, under **Multi-account permissions**, choose **AWS accounts**.
4. On the **AWS accounts** page, a tree view list of your organization appears. Select the name of your account.
5. Choose **Assign users or groups**.
6. On the **Assign users and groups** page, select the **Users** tab.
7. Select the checkbox next to name of the user that will become the CodeWhisperer administrator.
8. Choose **Next**.
9. On the **Assign permission sets** page, select the checkbox next to **CodeWhisperer_administrator**.
10. Choose **Next**.
11. On the **Review and submit assignments** page, choose **Submit**.

Now the CodeWhisperer administrator has the proper access.

The next step is for the CodeWhisperer administrator to authorize a professional developer to use CodeWhisperer Professional through an IDE.

Administering end users

If CodeWhisperer has already been set up for your organization

The procedures below will only be effective if CodeWhisperer has not already been set up anywhere in your organization.

You should set up CodeWhisperer in [a delegated administrator account](#) that you use primarily for IAM Identity Center and CodeWhisperer administration.

To delete CodeWhisperer from an account, choose **Delete** from the top of the CodeWhisperer settings page.

Adding the CodeWhisperer application to IAM Identity Center

Warning

In this procedure, you are acting as the CodeWhisperer administrator. In the previous procedure, you were acting as the AWS Organizations administrator. If necessary, log out of the AWS console, and log back in as the CodeWhisperer administrator.

To add the CodeWhisperer application to IAM Identity Center, complete the following steps:

1. Open a browser tab with the access portal URL, and log in as the CodeWhisperer administrator.
2. At the top of the next screen, choose the orange cube representing your AWS account. If your account is the only account in its AWS organization, then there will only be one choice.
3. The name of your account will appear in a bar, along with your account number and the associated email address.

Choose the bar.

4. The bar will expand to show **CodeWhisperer_administrator**. It may also show other access profiles, depending on how your account is configured.

On the same row as **Administrator Access**, choose **Management console**.

5. From the console homepage choose Amazon CodeWhisperer.
6. From the CodeWhisperer console page, choose **Set up CodeWhisperer**.

Note

After your initial setup, the **Set up** page becomes the **Settings** page.

7. On the **Set up** page, under **Details**, the option **Include suggestions with code references** is selected by default. If appropriate, leave it selected.

To learn more about this option, see [Code references \(p. 52\)](#).

Assign users and groups to IAM Identity Center

Warning

In this procedure, you are acting as the Organizations administrator, logged into the delegator administrator account. Depending on how you were logged in for the previous procedures, you may need to switch users, accounts, and/or roles before continuing.

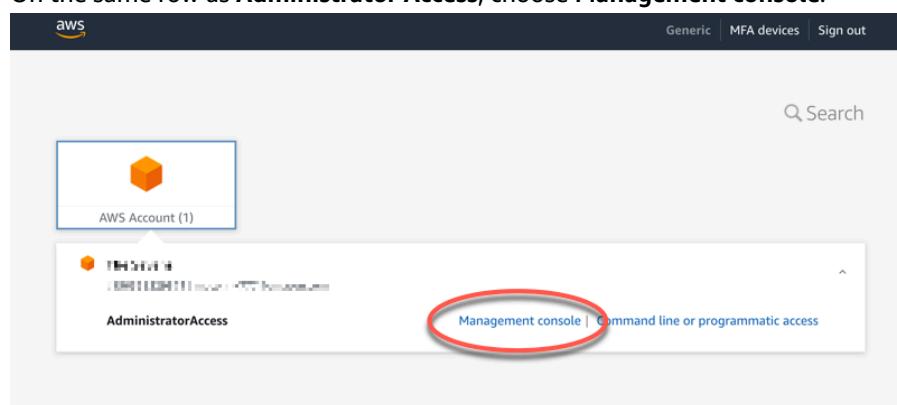
Your organization's contributors authenticate through AWS IAM Identity Center (successor to AWS Single Sign-On). To authorize developers in your organization to work with CodeWhisperer Professional, you must first create or import them as users in IAM Identity Center.

1. Open a browser tab with the access portal URL given to you by the root user.
2. Log in to the account that the root user created for you. Either the root user provided you with a one-time password that you now must change, or you received an email with directions for setting up your own password.
3. At the top of the next screen, choose the orange cube representing your AWS account. If your account is the only account in its AWS organization, then there will only be one choice.
4. The name of your account will appear in a bar, along with your account number and the associated email address.

Choose the bar.

5. The bar will expand to show **Administrator Access**. It may also show other access profiles, depending on how your account is configured.

On the same row as **Administrator Access**, choose **Management console**.



6. From the console home page, choose IAM Identity Center.
7. From the dashboard, choose **Choose your identity source**.

Your default identity source is Identity Center directory. With Identity Center directory, you manage users and groups completely inside IAM Identity Center.

8. (optional) If you want to choose a different identity source, then under the **Actions** dropdown, choose **Change identity source**.

On the **Choose identity source** page, the other two options are:

- **Active Directory:** Select this option if you already have your users and groups configured in Active Directory.
- **External identity provider:** Select this option if you already have your users and groups configured in an external system that is not Active Directory.

The rest of the process for adding users and groups to IAM Identity Center is beyond the scope of this guide. For additional information about IAM Identity Center and how to set it up, see the [AWS IAM Identity Center User Guide](#). Be sure to set create or import at least two more users: one for the CodeWhisperer administrator, and one for the professional developer. Then, return to this guide for [Assigning CodeWhisperer administration rights \(p. 7\)](#).

Authorizing professional developers to use CodeWhisperer

Warning

In this procedure, you are acting as the CodeWhisperer administrator. If necessary, log out of the AWS console, and log back in as the CodeWhisperer administrator.

To authorize specific users to work with CodeWhisperer, complete the following steps:

1. From the [CodeWhisperer console](#) choose **Settings** to open the **Settings** menu.
2. If you want to, from the **Details** section, select **Include suggestions with code references**.
After you make this selection, individual developers will not be able to change it in the IDE.
3. Under **Users** view, select the individuals that require authorization to use CodeWhisperer.

The users that you select will appear under **Selected users and groups**.

Note

Before a user can be chosen here, they must first be added in the IAM Identity Center by [the IAM Identity Center administrator \(p. 10\)](#). For more information, see [Assign users and groups to IAM Identity Center \(p. 11\)](#)

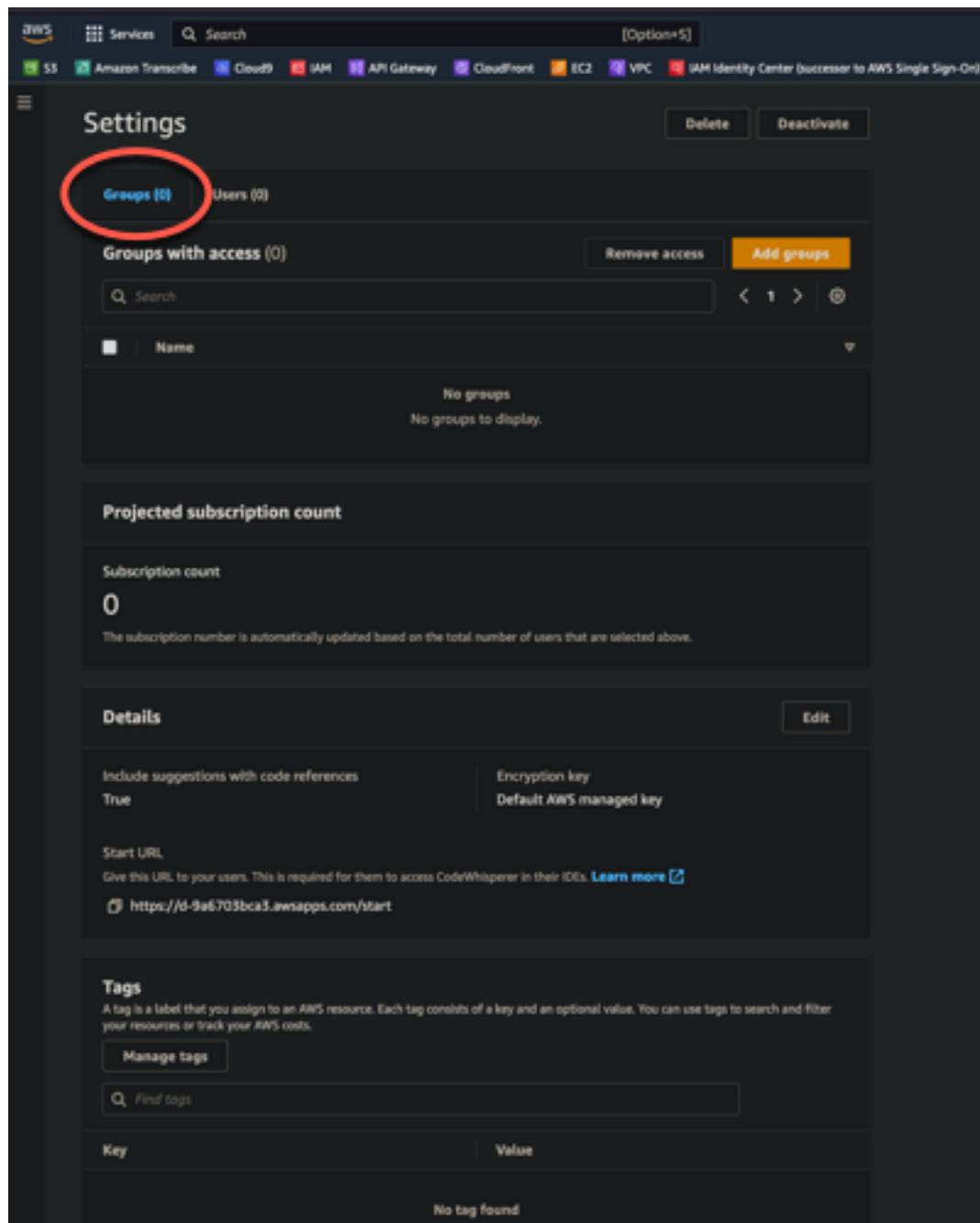
4. Choose **Set up CodeWhisperer**.

To authorize groups of users to use CodeWhisperer, complete the following steps:

1. From the [CodeWhisperer console](#) choose **Settings** to open the **Settings** menu.
2. In the upper right corner of the console window, confirm that the region is set to US East (N. Virginia).

This step is necessary, regardless of which region you used when adding the CodeWhisperer application to IAM Identity Center, or which region the account administrator used when adding or creating users and groups in IAM Identity Center.

3. If you want to, from the **Details** section, select **Include suggestions with code references**.
After you make this selection, individual developers will not be able to change it in the IDE.
4. From the **Groups** tab, choose **Add groups** to open the **Add groups** view.



5. From the **Add groups** view, choose the groups that require authorization to use CodeWhisperer.
6. Choose **Add groups** to authorize CodeWhisperer access for the selected groups.

The CodeWhisperer administrator removes access to CodeWhisperer

You can remove CodeWhisperer access for users and groups of individual users.

To remove CodeWhisperer access from individual users, complete the following steps:

1. From the CodeWhisperer console choose **Settings** to open the **Settings** menu.
2. From the **Users** tab, choose **Remove access**.
3. When prompted, choose **Remove** to confirm that you want to remove CodeWhisperer access for the user.

To remove CodeWhisperer access from a group of users, complete the following steps:

1. From the CodeWhisperer console, choose **Settings** to open the **Settings** menu.
2. From the **Groups** tab, choose **Remove Access**.
3. When prompted, choose **Remove** to confirm that you want to remove CodeWhisperer access for the group of users.

CodeWhisperer profiles

A CodeWhisperer profile is the configuration for your company's CodeWhisperer application. It includes decisions you make about the account (such as whether to include suggestions with code references), as well as the users and groups you give access to CodeWhisperer.

The concept of the CodeWhisperer profile can be important if you are changing IAM permissions related to CodeWhisperer. In that situation, the profile is the resource upon which CodeWhisperer acts.

For more information, see [Controlling access to AWS resources using policies](#) in the *IAM User Guide*.

Choosing your encryption key

By default, data collected by CodeWhisperer for the purpose of [Security scans \(p. 46\)](#) is stored using [Amazon S3](#) and [Amazon DynamoDB](#). This data is only stored as long as it's needed for that purpose. The data is encrypted using the data-at-rest encryption capabilities of Amazon S3 and Amazon DynamoDB, with a Builder ID-owned key.

However, administrators of CodeWhisperer Professional have the option of encrypting their company's data (used by CodeWhisperer for the purpose of security scans) with the AWS Key Management Service.

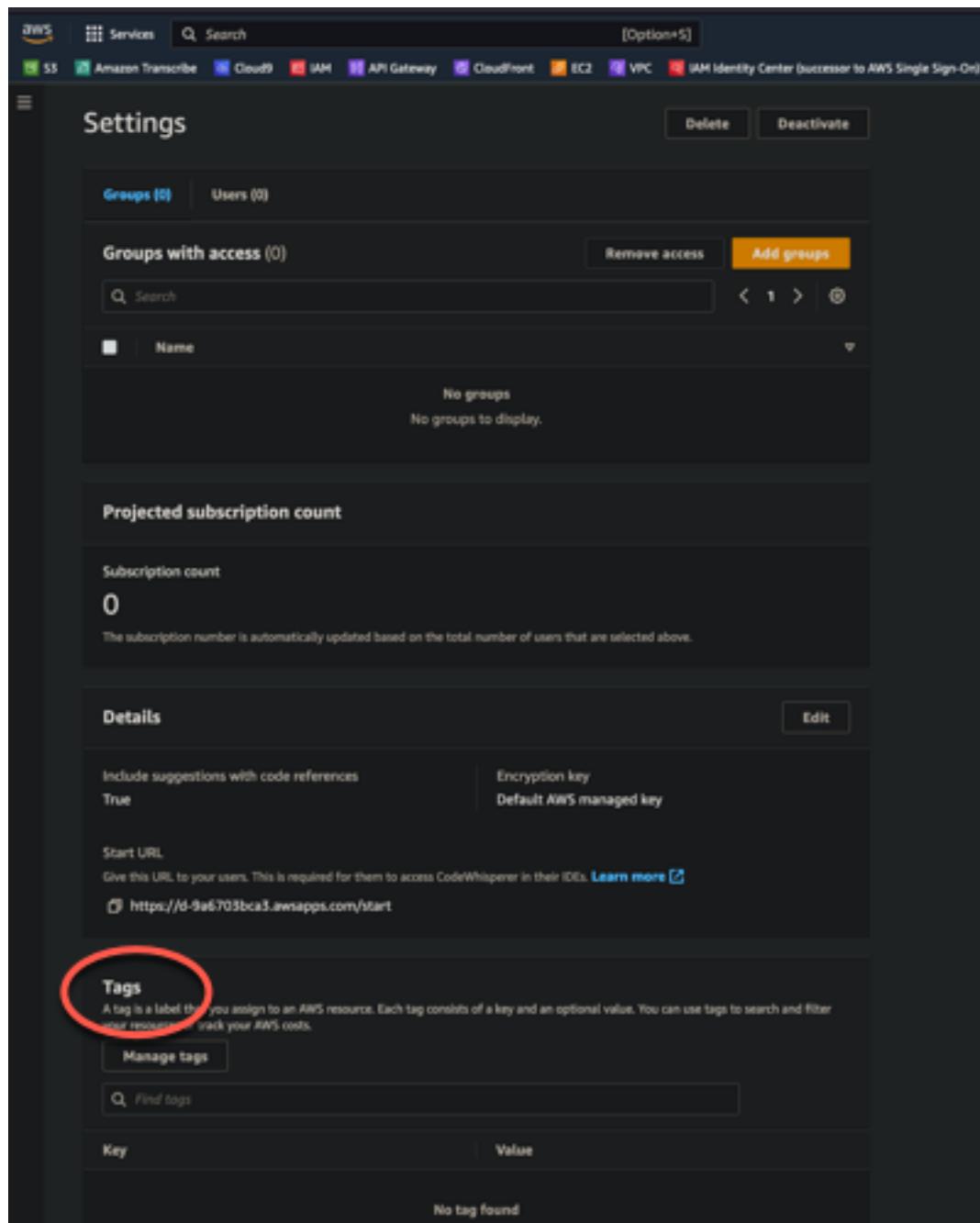
To learn more about AWS KMS, see [AWS Key Management Service concepts](#) in the *AWS Key Management Service Developer Guide*.

The screenshot shows the AWS CodeWhisperer settings interface. At the top, there's a navigation bar with the AWS logo, a search bar, and a 'Services' dropdown. Below the navigation is a breadcrumb trail: Amazon CodeWhisperer > Settings > Edit details. The main section is titled 'Edit details' and contains a 'Advanced settings' panel. This panel includes a note that these settings apply to all users and groups. It has two sections: 'CodeWhisperer suggestions' and 'Encryption key - optional'. In 'CodeWhisperer suggestions', there's a checked checkbox for 'Include suggestions with code references' with a descriptive text below it. In 'Encryption key - optional', there's another checked checkbox for 'Customize encryption settings (advanced)' with a note about using the default key. There's also a text input field for choosing an AWS KMS key or entering an ARN, and a button to 'Create an AWS KMS key'. At the bottom right are 'Cancel' and 'Save changes' buttons.

Understanding CodeWhisperer profile tags

You may want to add tags to CodeWhisperer profile in order to more easily track expenses, or to grant IAM permissions.

For more information, see [Tagging your AWS resources](#) in the *Tagging AWS resources User Guide*.



Activating and deactivating the CodeWhisperer application

You can control access to CodeWhisperer for your organization by activating or deactivating the application.

To disable the CodeWhisperer application, complete the following steps:

1. From the CodeWhisperer console choose **Settings** to open the **Settings** menu.
2. Choose **Disable CodeWhisperer**.
3. When prompted, choose **Disable in IAM Identity Center** to open the **IAM Identity Center**.

4. From **Configured applications** in the **IAM Identity Center**, choose **CodeWhisperer**.
5. From the **Actions** list, choose **Disable application** to disable CodeWhisperer.

To re-enable the CodeWhisperer application, complete the following steps:

1. From the CodeWhisperer console choose **Settings** to open the **Settings** menu.

Note

The console displays an alert indicating that CodeWhisperer has been disabled.

2. When prompted, choose **IAM Identity Center**.
3. From the **Groups** tab, choose **Add access**.

Getting started

The following sections describe how to set up CodeWhisperer for use with each of four possible IDEs: AWS Toolkit for JetBrains, AWS Toolkit for Visual Studio Code, Lambda, and AWS Cloud9.

With Lambda and AWS Cloud9, the setup simply involves activating CodeWhisperer within the IDE.

If you are using CodeWhisperer, on behalf of your organization, with VS Code or JetBrains, then you are using CodeWhisperer Professional. In that case, administrators at your organization must complete additional steps before you can start coding. For more information, see [Setting up CodeWhisperer for professional developers \(p. 18\)](#).

If you are using CodeWhisperer, on your own behalf, with VS Code or JetBrains, then you are using CodeWhisperer Individual. In that case, you can go directly to [Setting up CodeWhisperer for individual developers \(p. 22\)](#).

Topics

- [With VS Code and JetBrains \(p. 18\)](#)
- [Setting up CodeWhisperer with Amazon SageMaker Studio \(p. 28\)](#)
- [Setting up CodeWhisperer with JupyterLab \(p. 29\)](#)
- [Setting up CodeWhisperer with AWS Glue Studio \(p. 31\)](#)
- [Setting up Amazon CodeWhisperer with AWS Lambda \(p. 31\)](#)
- [Setting up CodeWhisperer with AWS Cloud9 \(p. 32\)](#)

With VS Code and JetBrains

Topics

- [Setting up CodeWhisperer for professional developers \(p. 18\)](#)
- [Setting up CodeWhisperer for individual developers \(p. 22\)](#)
- [Authenticating with CodeWhisperer and AWS Toolkit \(p. 27\)](#)

Setting up CodeWhisperer for professional developers

You use CodeWhisperer Professional if your organization has set it up with IAM Identity Center and provided you a Start URL (an authentication URL for CodeWhisperer).

If you have acquired the tool independent of your team or organization and you will use AWS Builder ID to login, see [Setting up CodeWhisperer for individual developers \(p. 22\)](#).

Important

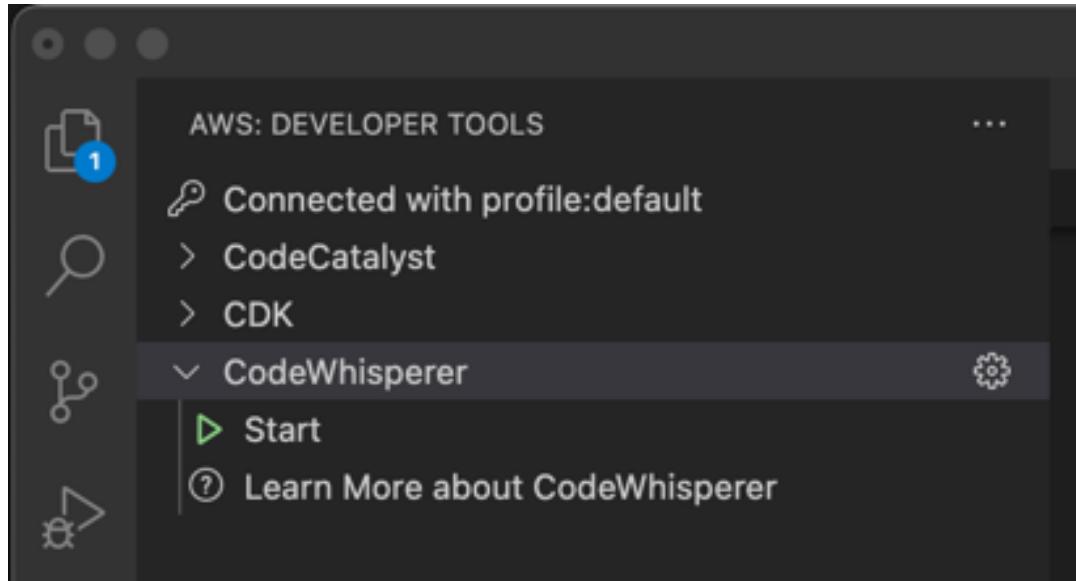
Before you proceed, make sure that you are using the latest version of both your IDE and the AWS Toolkit.

Note

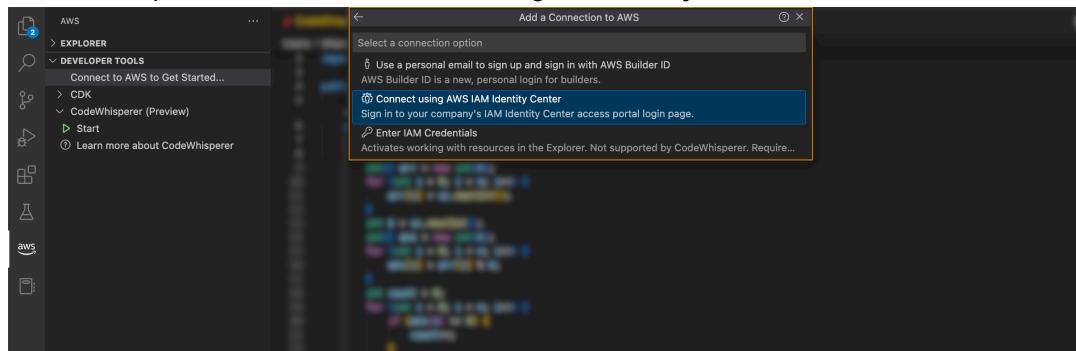
AWS recommends that, before using CodeWhisperer, you disable any other extensions that provide code completion functionality.

VS Code

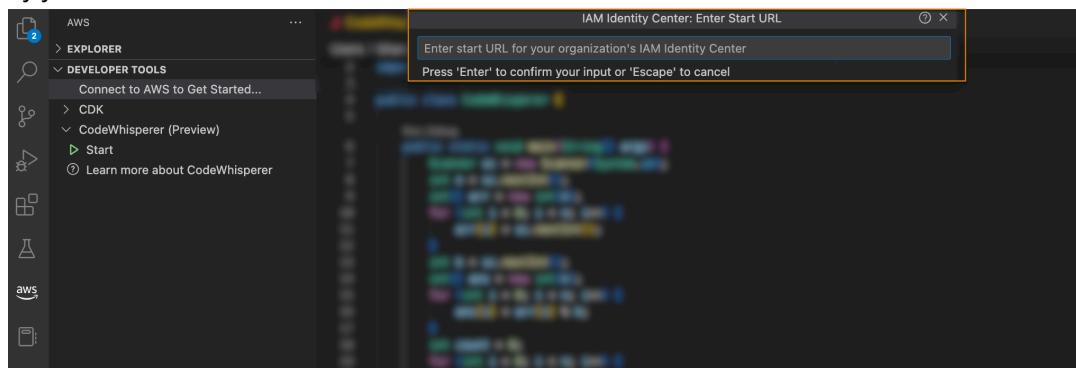
1. Open your local instance of VS Code (with the Toolkit for Visual Studio already installed).
2. In the Toolkit for Visual Studio pane, under **Developer tools**, under CodeWhisperer, select **Start**. A dropdown menu will appear at the top of VS Code.



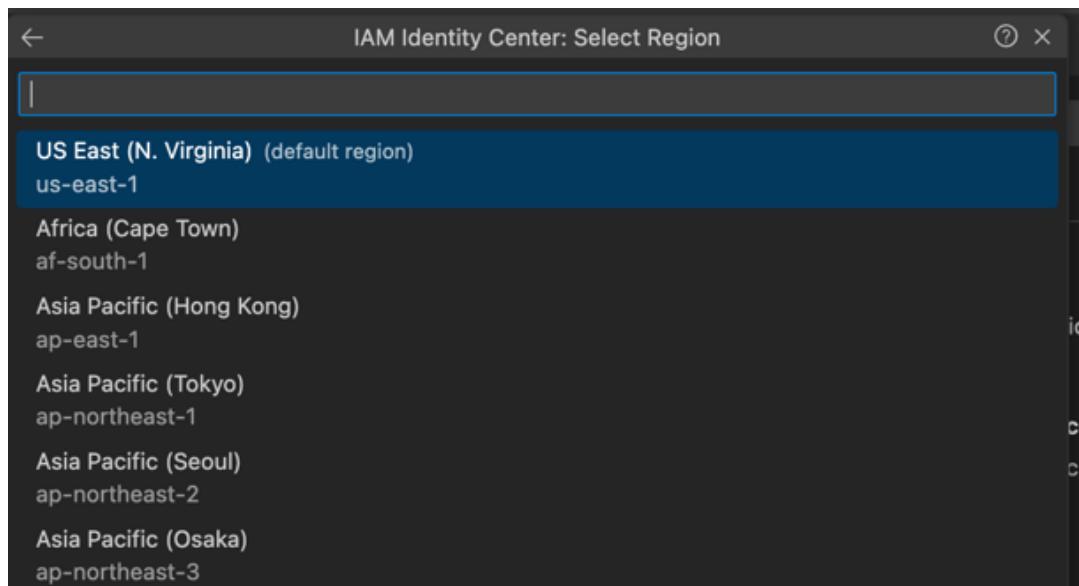
3. From the dropdown menu, select **Connect using IAM Identity Center**.



4. In the **IAM Identity Center: Enter Start URL** request window, enter the start URL given to you by your account administrator.



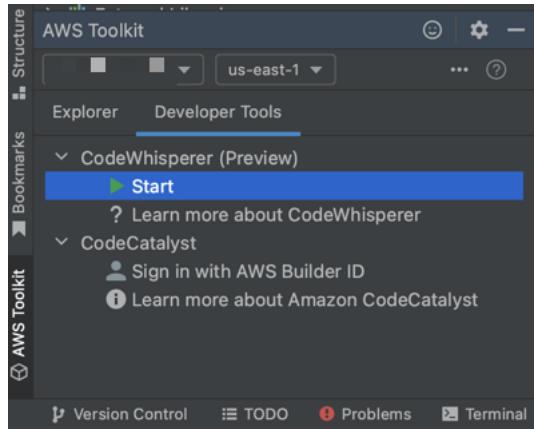
5. In the **IAM Identity Center: Select Region** request window, enter the region in which your account administrator created your organization's IAM Identity Center profile.



6. From the pop-up, choose **Copy Code and Proceed**.
 7. When prompted, confirm that you want to open the sign in page in your preferred web browser.
 8. From your web browser, in the **Authorize request** window, paste the code from your buffer into the empty field.
 9. When prompted, enter your username.
 10. When prompted, enter your password and choose **Sign in**.
 11. When prompted, confirm that you want to allow AWS Toolkit to access your data.
 12. When the connection is complete, a notification indicates that it is safe to close your browser.
- Close the browser tab and return to VS Code.
13. In VS Code, choose **Accept and Activate CodeWhisperer**.

JetBrains

1. From the AWS Toolkit for JetBrains, choose the **Developer Tools** tab to open the **Developer Tools** explorer.
2. From the **Developer Tools** explorer, expand **CodeWhisperer** and choose **Start**.

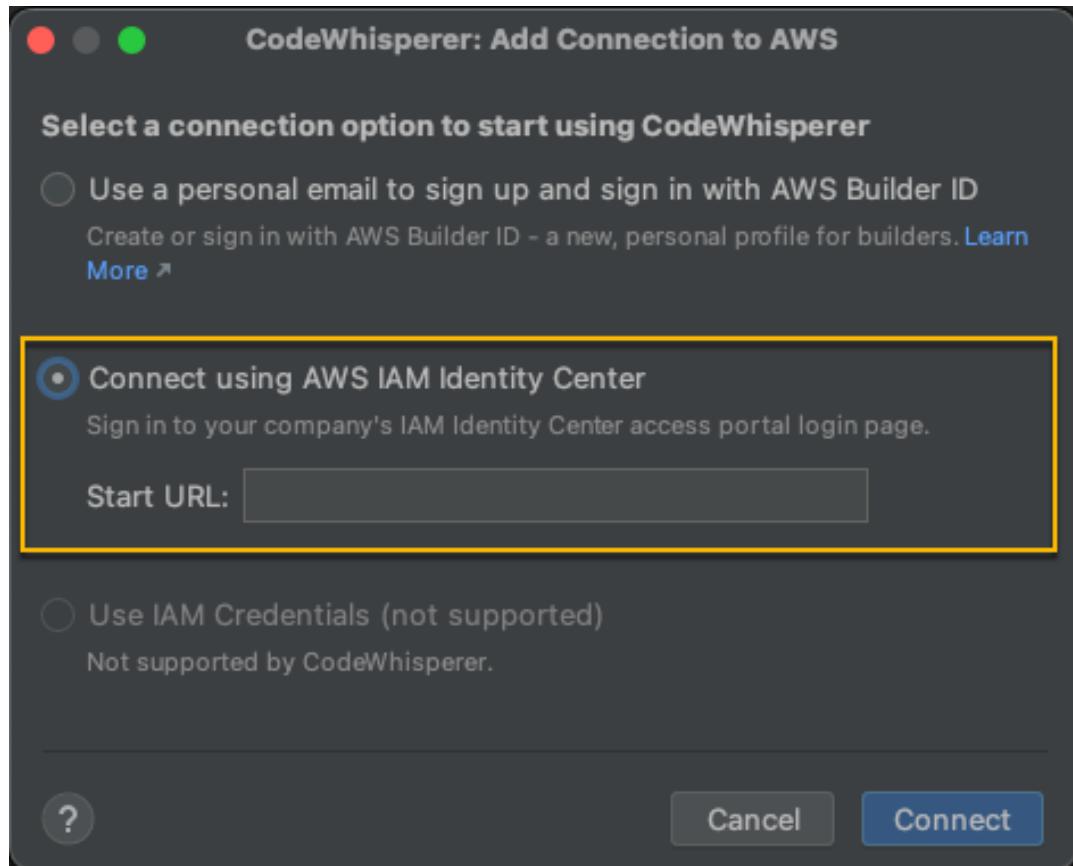


3. A pop-up window will appear, titled **CodeWhisperer: Add a Connection to AWS**. Select the **Connect Using IAM Identity Center** option.

Enter the start URL provided by your administrator.

Enter the region in which your account administrator set up IAM Identity Center (you have may to ask).

Then choose **Connect**.



4. A pop-up window will display an access code.

Choose **Open and Copy Code**.

5. Back in your browser in the Authorize request window, paste the access code that was automatically added to your copy/paste buffer on the last step. Then choose **Next**.
6. In the browser, on the sign-in page, enter your username and choose **Next**.
7. On the password page, enter your password and choose **Next**.

Your account administrator may have given you a one-time password, which you must then reset. Or, you may have received an account creation email, including instructions for setting your own password.

8. If necessary, create a new password.
9. Choose **Allow** to confirm that you want to allow AWS Toolkit for JetBrains to access your data.
10. Return to JetBrains and begin coding.

Setting up CodeWhisperer for individual developers

You use CodeWhisperer Individual if you have acquired the tool independent of your team or organization and you will use AWS Builder ID to login.

If your organization is using IAM Identity Center and has provided you a Start URL (an authentication URL), see [Setting up CodeWhisperer for professional developers \(p. 18\)](#).

Note

AWS recommends that, before using CodeWhisperer, you disable any other extensions that provide code completion functionality.

Important

Before you proceed, make sure that you are using the latest version of both your IDE and the AWS Toolkit.

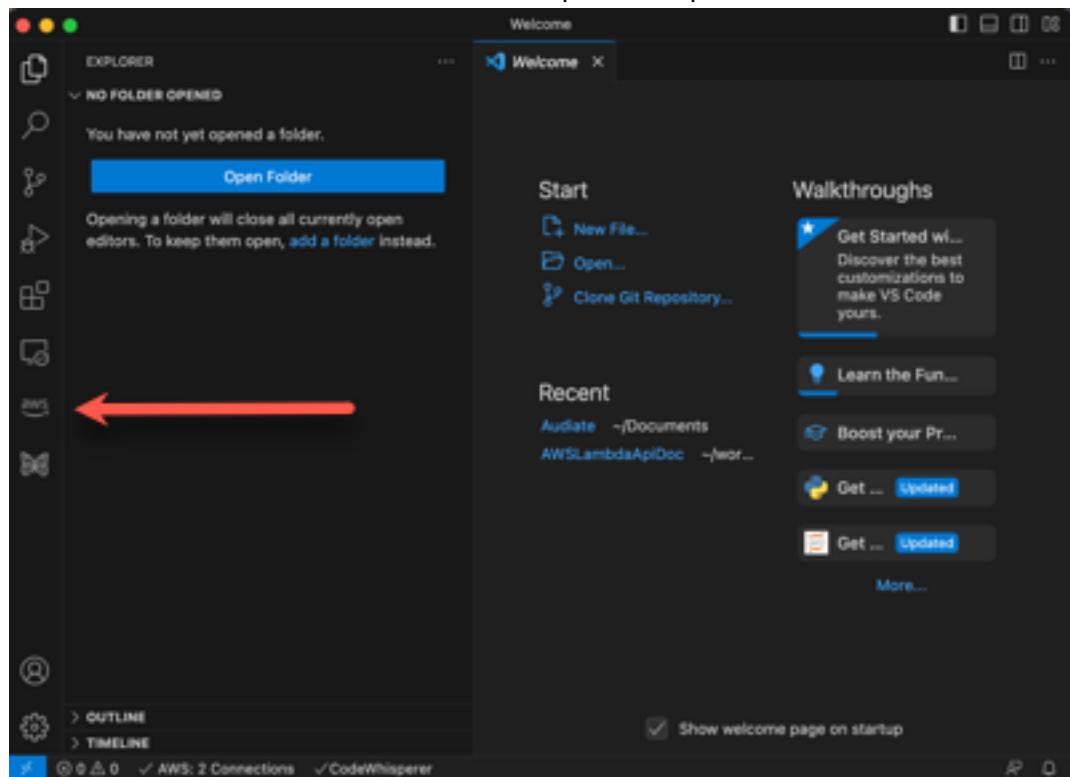
VS Code

Note

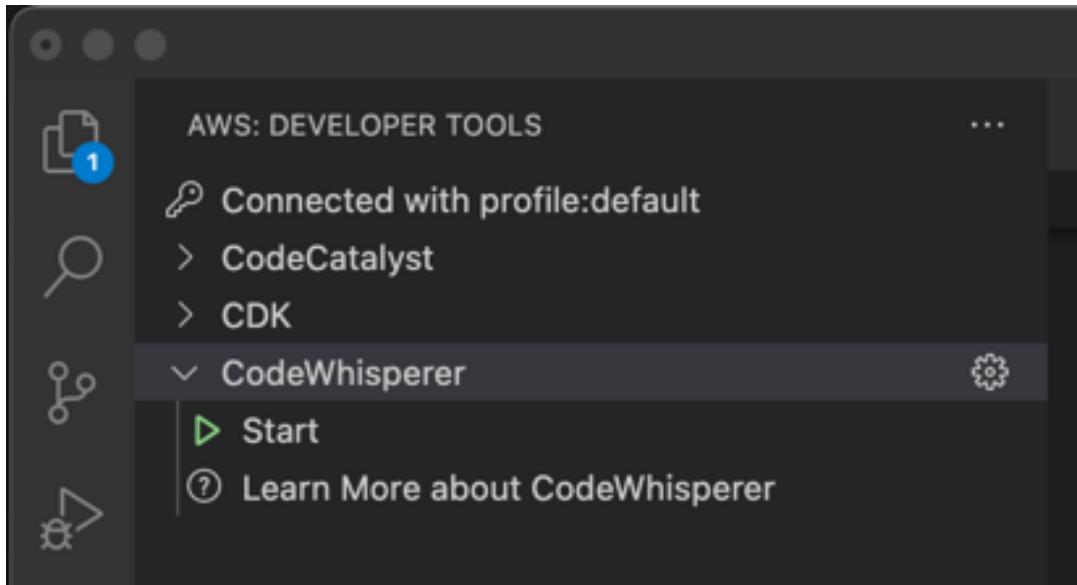
In order to use CodeWhisperer, all individual developers must sign up through the procedure below. Access to CodeWhisperer, for individual developers, is not tied to your AWS account access. Having an individual AWS account that is already connected to VS Code does not automatically give you access to CodeWhisperer in VS Code.

Your VS Code may already be connected to AWS through IAM Identity Center or IAM. In that case, during the following procedure, you will be prompted to decide whether to continue while using simultaneous methods of authentication. For more information, see [Authenticating with CodeWhisperer and AWS Toolkit \(p. 27\)](#).

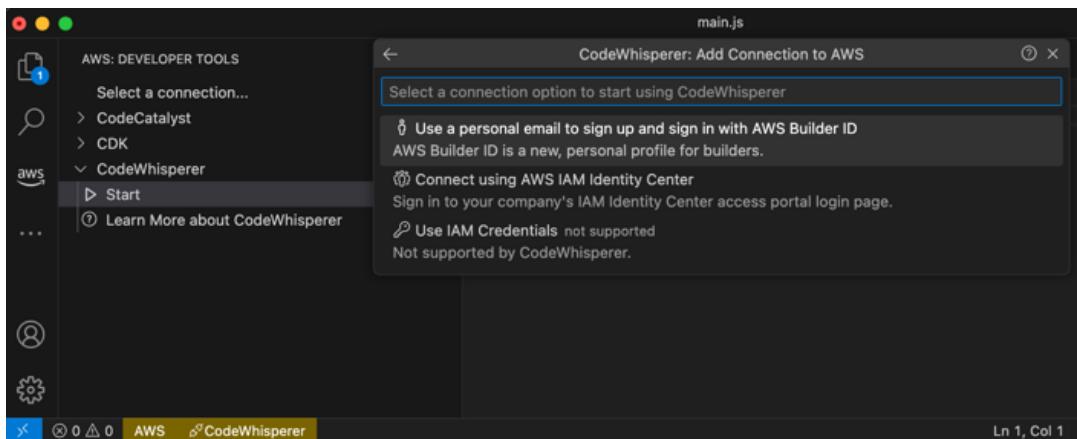
1. If necessary, [install the AWS Toolkit for Visual Studio Code](#).
2. In VS Code, with the AWS Toolkit for Visual Studio Code installed, choose the AWS logo from the side of the VS Code window. The AWS Toolkit panel will open.



3. on the AWS Toolkit panel, under **AWS: Developer tools**, under CodeWhisperer, select **Start**. A dropdown menu will appear at the top of VS Code.



4. From the dropdown menu, select **Use a personal email to sign up and sign in with AWS Builder ID**



5. At the prompt **Copy Code for AWS Builder ID** choose **Copy Code and Proceed**.
6. At the prompt **Do you want Code to open the external website?** choose **Open**.
7. A browser tab will open, displaying the **Authorize request** window.

The code should have been already added to your copy/paste buffer. Paste it in, and choose **Next**.

8. A browser tab will open to the **Create AWS Builder ID** page.

Enter your email address, and choose **Next**.

9. A field for **Your name** will appear. Enter your name and choose **Next**.
 10. AWS will send an confirmation code to the email address that you submitted.
- On the email verification screen, enter the code and choose **Verify**.
11. On the **Choose your password** screen, enter a password, confirm it, and choose **Create AWS Builder ID**.

12. A browser tab will open with a message asking you to allow AWS Toolkit for Visual Studio Code to access your data.

Choose **Allow**.

13. Return to VS Code. If you have already authenticated into other AWS tools using IAM, you will be asked whether you want to use Builder ID for all AWS services. Choose the option that best suits your situation.

If you're not sure what to do, choose **Yes, keep using AWS Builder ID with CodeWhisperer while using....**

JetBrains

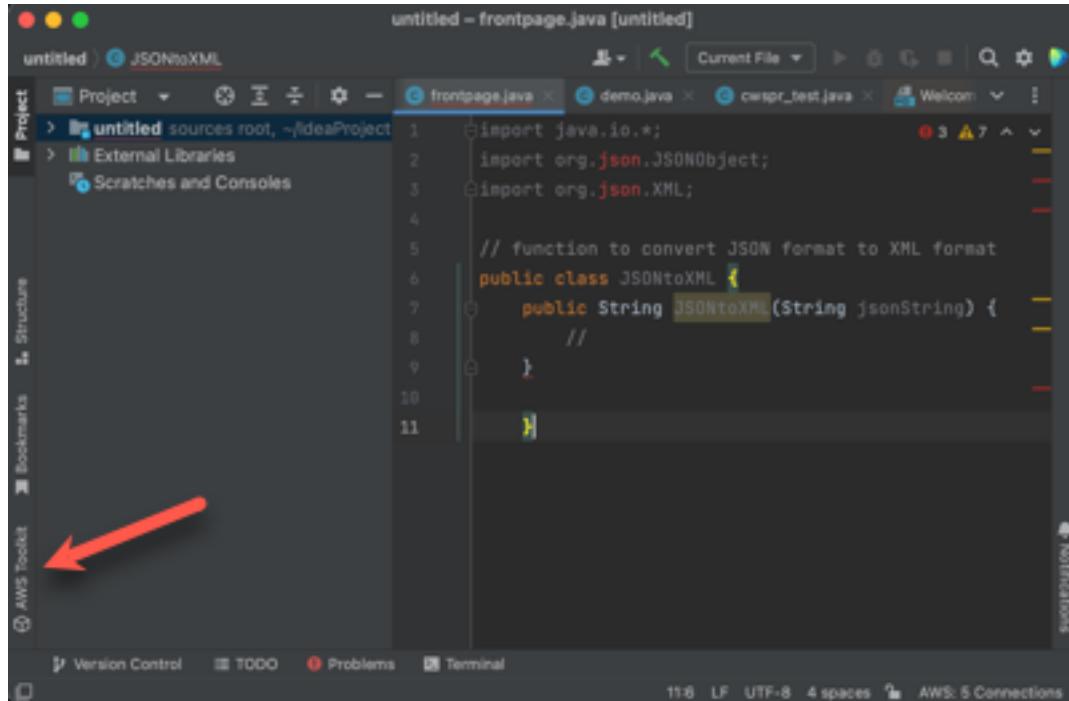
[Getting started with Amazon CodeWhisperer with IntelliJ](#)

Note

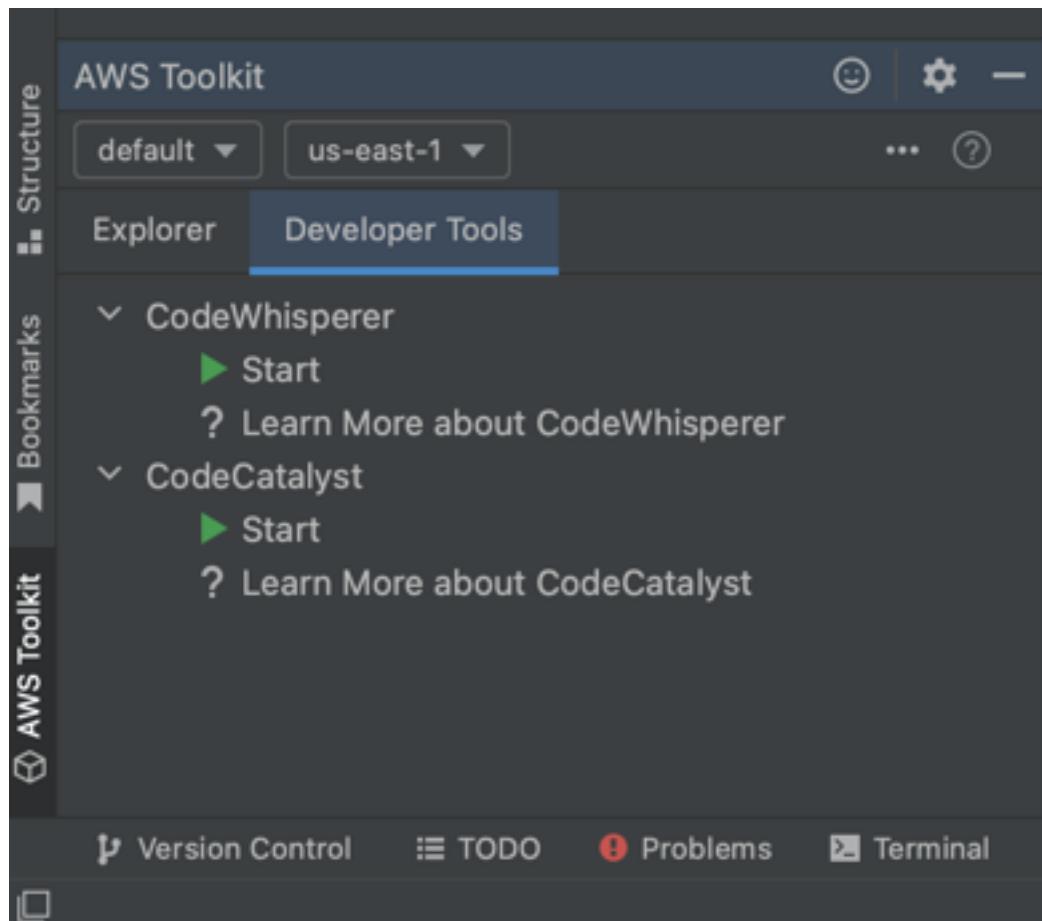
In order to use CodeWhisperer, all individual developers must sign up through the procedure below. Access to CodeWhisperer, for individual developers, is not tied to your AWS account access. Having an individual AWS account that is already connected to JetBrains does not automatically give you access to CodeWhisperer in JetBrains.

To access CodeWhisperer from your JetBrains IDE, complete the following steps:

1. If necessary, [install the AWS Toolkit for JetBrains](#).
2. in JetBrains, choose **AWS Toolkit** from the edge of the JetBrains window. The AWS Toolkit for JetBrains panel will open.

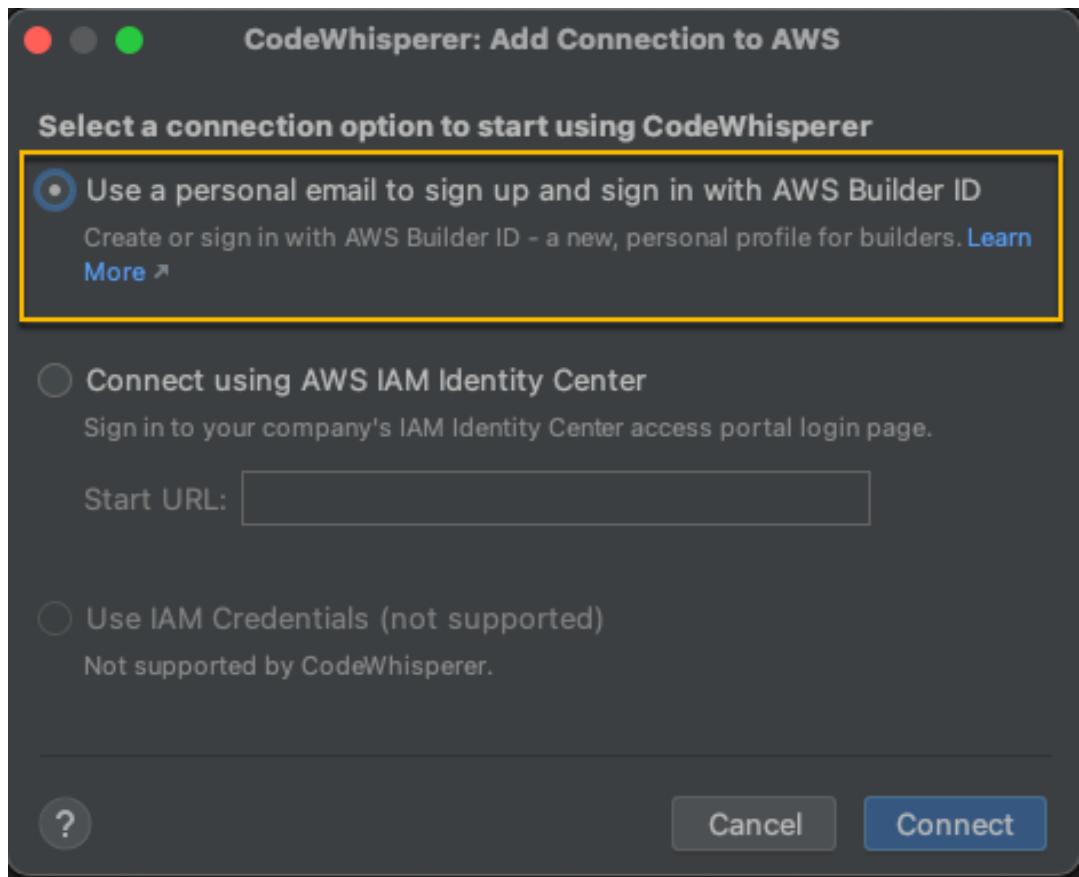


3. From the AWS Toolkit for JetBrains panel, choose the **Developer Tools** tab to open the **Developer Tools** explorer.
4. From the **Developer Tools** explorer, expand **CodeWhisperer** and choose **Start**.



5. A pop-up window will appear, titled **CodeWhisperer: Add a Connection to AWS**.

Select **Use a personal email to sign up and sign in with AWS Builder ID** and choose **Connect**.



6. A pop-up will open titled **Sign in with AWS Builder ID**. Choose **Open and Copy Code**.
7. A browser tab will open, displaying the **Authorize request** window.

The code should have been already added to your copy/paste buffer. Paste it in, and choose **Next**.
8. A browser tab will open to the **Create AWS Builder ID** page.

Enter your email address, and choose **Next**.
9. A field for **Your name** will appear. Enter your name and choose **Next**.
10. AWS will send an confirmation code to the email address that you submitted.

On the email verification screen, enter the code and choose **Verify**.
11. On the **Choose your password** screen, enter a password, confirm it, and choose **Create AWS Builder ID**.
12. A browser tab will open with a message asking you to allow JetBrains to access your data.

Choose **Allow**.
13. A browser tab will open. A message will appear in the new browser tab, asking if you want to allow AWS Toolkit for JetBrains to access your data.

If you agree, choose **Allow**.
14. Return to JetBrains.

Now you should be logged into CodeWhisperer with Builder ID.

To begin coding, see [User actions \(p. 36\)](#).

Authenticating with CodeWhisperer and AWS Toolkit

To use CodeWhisperer with the AWS Toolkit for Visual Studio Code, you must establish an authenticated connection to AWS (but you don't need an AWS account). This page describes each method of authenticating with the AWS Toolkit, and how each one relates to CodeWhisperer.

AWS IAM Identity Center (successor to AWS Single Sign-On)

IAM Identity Center expands the capabilities of IAM to provide a central place that brings together administration of users and their access to AWS accounts and cloud applications. Users in IAM Identity Center are managed by a corporate IT or cloud administrator, or by the administrator of the organization's identity provider, such as Okta, Ping, or Azure.

When using CodeWhisperer, you should authenticate with IAM Identity Center if you are an enterprise developer. That is, you are working with CodeWhisperer as an employee of an organization that has an AWS account. Before you can authenticate using IAM Identity Center, your administrator must add you as a user.

[Learn more about IAM Identity Center](#)

[Learn about setting up CodeWhisperer for enterprise developers \(p. 18\)](#)

Builder ID

AWS Builder ID is a personal profile for builders. It represents you as a person, outside the scope of your company or school. You can sign up for AWS Builder ID with your name and email.

When using CodeWhisperer, you should authenticate with Builder ID if you are an individual developer. That is, you are working on a personal project, or if your organization does not authenticate to AWS using IAM Identity Center.

[Learn about Builder ID](#)

[Learn about setting up CodeWhisperer for individual developers \(p. 22\)](#)

AWS Identity and Access Management

AWS Identity and Access Management is a web service that helps you securely control access to AWS resources. Using IAM, you manage access in AWS by creating policies and attaching them to IAM identities (users, groups of users, or roles) or AWS 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 uses an IAM entity (user or role) to make a request. CodeWhisperer, when used with AWS Toolkit, does not support authentication with IAM. However, IAM credentials are required to use CodeWhisperer with [Lambda](#) or [AWS Cloud9](#).

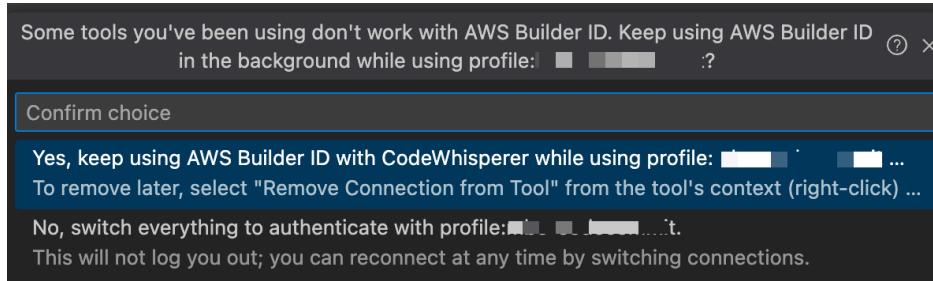
[Learn about IAM](#)

Switching between authentication methods

Although CodeWhisperer does not support authentication with IAM, you may use IAM to access other AWS services from inside the same IDE. However, in such cases, your access to CodeWhisperer will still be managed through either IAM Identity Center or Builder ID.

For example, suppose that you are using CodeWhisperer in your JetBrains IDE, and you are authenticated with Builder ID. Then you decide to switch tasks, but without leaving JetBrains. Now you want to [invoke a Lambda function](#) in your AWS account. However, access to Lambda requires IAM credentials. Therefore, you must switch profiles within JetBrains, from your Builder ID profile to another profile that authenticates using your IAM credentials.

In such cases, the IDE presents an alert, reminding you that you are switching to a service with a different method of authentication. You will also have the option to stay connected to CodeWhisperer (using Builder ID or IAM Identity Center) while simultaneously using another service that you are connected to using IAM.



Setting up CodeWhisperer with Amazon SageMaker Studio

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

Note

Python is the only programming language that CodeWhisperer 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 CodeWhisperer-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.

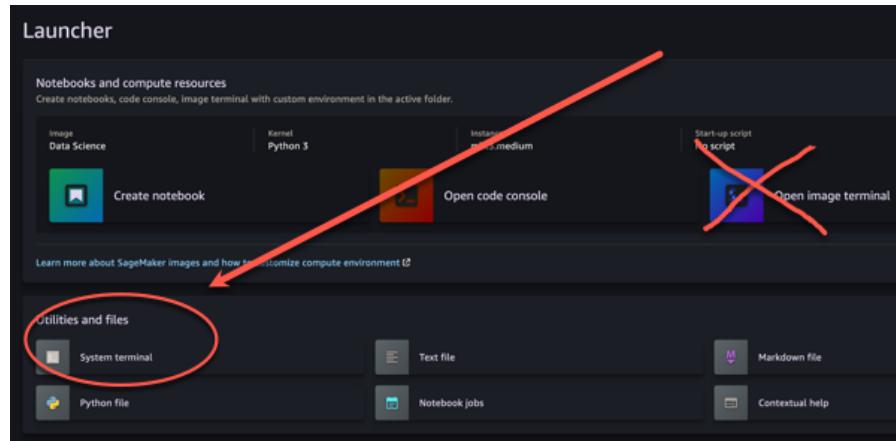
```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "CodeWhispererPermissions",  
            "Effect": "Allow",  
            "Action": ["codewhisperer:GenerateRecommendations"],  
            "Resource": "*"  
        }  
    ]  
}
```

```
}
```

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

4. Enable the CodeWhisperer extension in your SageMaker Studio domain.

In the system terminal (not the image terminal) inside SageMaker Studio, run the following commands.



```
conda activate studio
pip install amazon-codewhisperer-jupyterlab-ext
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 CodeWhisperer only appear in code cells. They do not appear in markdown cells.

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

For keyboard shortcuts, see [User actions \(p. 36\)](#).

Setting up CodeWhisperer with JupyterLab

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

Note

Python is the only programming language that CodeWhisperer supports in JupyterLab.

In the following procedure, you will install CodeWhisperer in JupyterLab.

1. Install [JupyterLab 3.6](#) on your computer.

Warning

CodeWhisperer does not currently support JupyterLab 4.

2. Start JupyterLab from the command line.

```
jupyter lab
```

JupyterLab will open in your browser.

3. Enable the CodeWhisperer extension in JupyterLab, as follows.

In your terminal, run the following commands.

```
pip install amazon-codewhisperer-jupyterlab-ext
jupyter server extension enable amazon_codewhisperer_jupyterlab_ext
```

4. In the JupyterLab GUI, refresh the list of installed extensions. The CodeWhisperer extension should appear.

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

1. Refresh the browser tab on which you are using JupyterLab.
2. From the CodeWhisperer panel at the bottom of the window, choose **Start CodeWhisperer**.
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 CodeWhisperer to access your data.

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

To begin coding, see [User actions \(p. 36\)](#).

The screenshot shows the JupyterLab interface with a code cell containing the following Python code:

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

Below the code cell, a suggestion from CodeWhisperer is shown:

```
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
```

A note below the suggestion states: "# cast object dtype to string. The SageMaker Feature Store Python SDK will then map the string dtype to String feature type." The final line of the code cell is `cast_object_to_string(clinical)`.

Setting up CodeWhisperer with AWS Glue Studio

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

Note

CodeWhisperer 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 CodeWhisperer.

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

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "CodeWhispererPermissions",  
            "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 CodeWhisperer
8. Choose **Start notebook**.

Setting up Amazon CodeWhisperer with AWS Lambda

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

Note

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

AWS Identity and Access Management permissions for Lambda

For CodeWhisperer 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:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "CodeWhispererPermissions",  
            "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 CodeWhisperer with Lambda

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

Note

CodeWhisperer 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 **CodeWhisperer code suggestions**. This immediately activates the CodeWhisperer service, and a check mark appears next to this option. To deactivate, choose this option again.

Setting up CodeWhisperer with AWS Cloud9

AWS Identity and Access Management permissions for AWS Cloud9

For CodeWhisperer 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:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "CodeWhispererPermissions",  
            "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 CodeWhisperer with AWS Cloud9

To activate CodeWhisperer 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 **CodeWhisperer** dropdown.
3. Choose **Enable CodeWhisperer**.

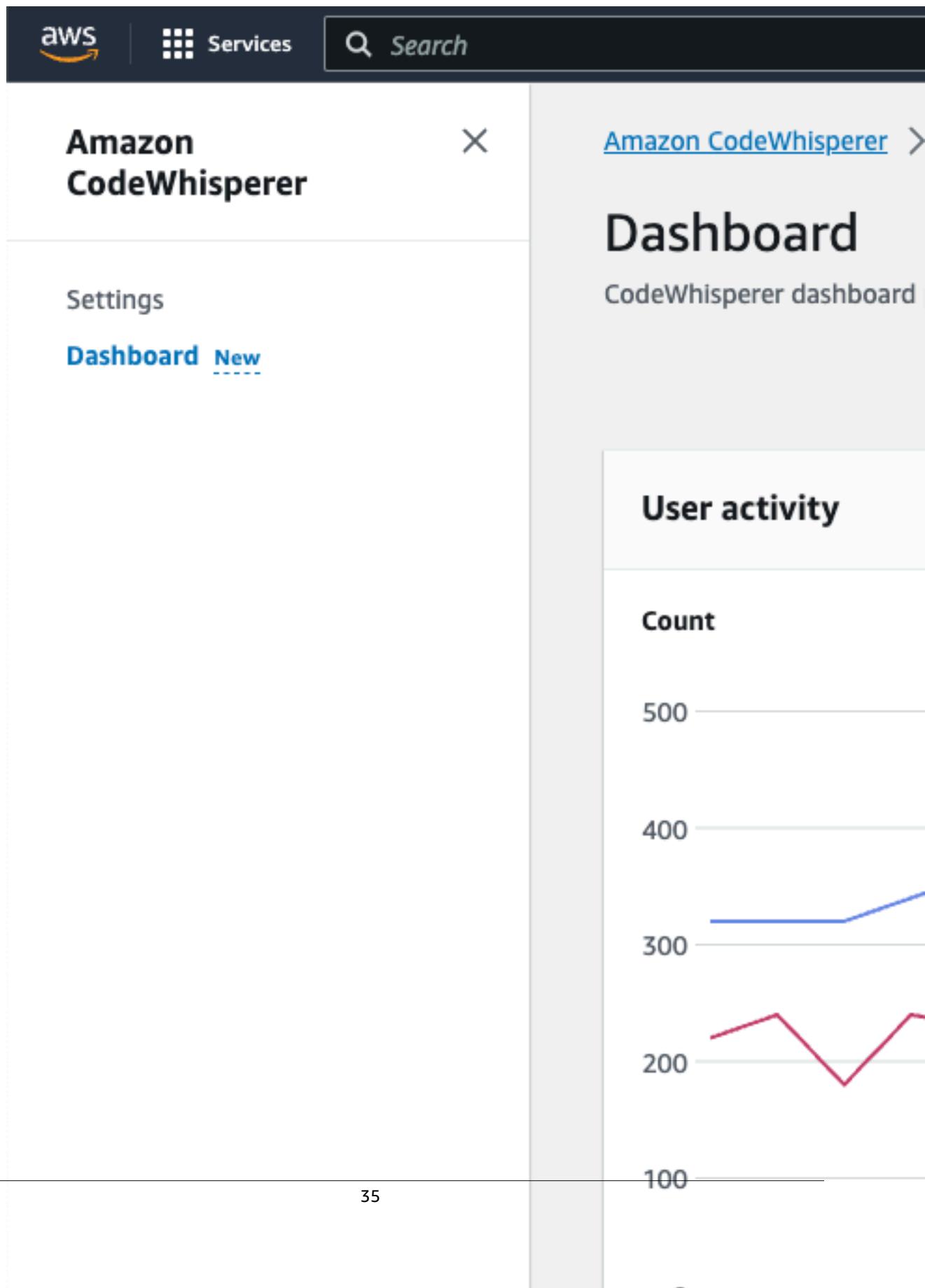
For examples of how CodeWhisperer integrates with AWS Cloud9 and displays code suggestions in the AWS Cloud9 IDE, see [Code examples \(p. 66\)](#).

Features

Topics

- [Dashboard \(p. 35\)](#)
- [User actions \(p. 36\)](#)
- [Language and IDE support in Amazon CodeWhisperer \(p. 40\)](#)
- [Pausing suggestions with Amazon CodeWhisperer \(p. 43\)](#)
- [Security scans \(p. 46\)](#)
- [Code references \(p. 52\)](#)

Dashboard



Available only for administrators, and only at the Professional tier, the CodeWhisperer dashboard summarizes useful data about how your developers use the service.

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

To view metrics on the dashboard, you must have the `cloudwatch:GetMetricData` permission. This permission is granted to administrators as part of [Assigning CodeWhisperer administration rights \(p.\)](#).

User activity

The **User activity** section indicates how many CodeWhisperer 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 CodeWhisperer simply indicates how many lines of code were suggested by CodeWhisperer and accepted by your developers.

The percentage of code written by CodeWhisperer indicates how much of your total code is made up of suggestions from CodeWhisperer.

Accepted recommendations with references indicates the number of suggestions from CodeWhisperer that are based on open-source projects, the references to which CodeWhisperer makes available to you.

Security scans

CodeWhisperer security scan is a tool that helps identify security vulnerabilities in your developers' code.

The data shown indicates how many scans your developers have successfully run in their IDEs.

User actions

Amazon SageMaker

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

JupyterLab

Action	Keyboard shortcut
Manually trigger CodeWhisperer	MacOS: Option + C

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

AWS Toolkit for Visual Studio Code

Action	Keyboard shortcut
Manually trigger CodeWhisperer	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 CodeWhisperer	MacOS: Option + C

Action	Keyboard shortcut
	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.

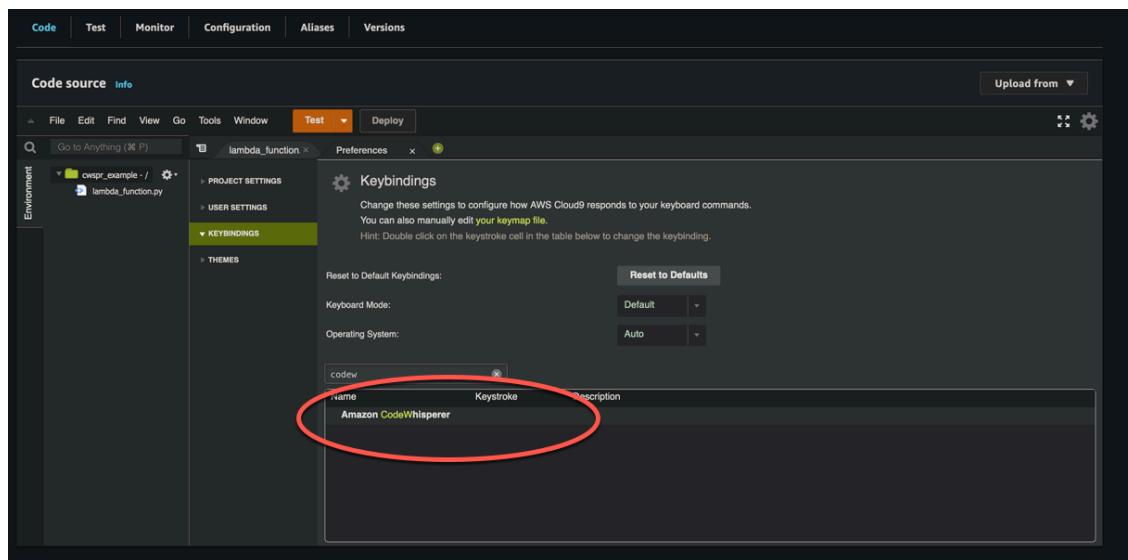
To change keybindings in IntelliJ, see [IntelliJ IDEA keyboard shortcuts](#) on the JetBrains website.

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.

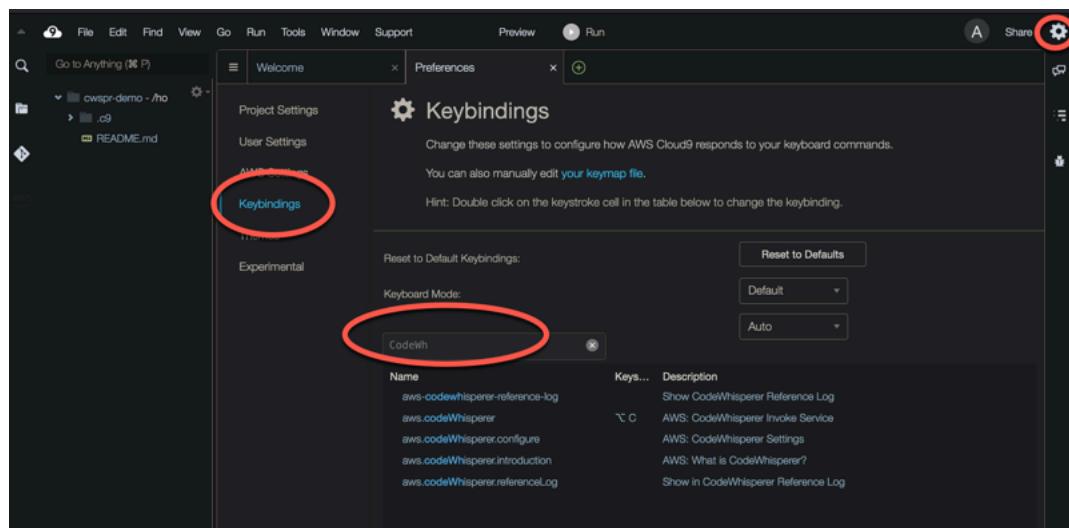
1. While viewing a particular function, choose the gear icon to open the **Preferences** tab.
2. On the **Preferences** tab, select **Keybindings**.
3. In the keybindings search box, enter CodeWhisperer.



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 CodeWhisperer.
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.



Useful APIs

- **GenerateRecommendations** - Gets code suggestions in CodeWhisperer for AWS Cloud9 and CodeWhisperer for Lambda Console.
- **GenerateCompletions** - Gets code suggestions in CodeWhisperer for VS Code and JetBrains.
- **StartCodeAnalysis** - Starts a security scan in CodeWhisperer for VS Code and JetBrains.
- **GetCodeAnalysis** - Gets the status of an ongoing security scan.
- **ListCodeAnalysisFindings** - Called after GetCodeAnalysis signals job completions. Returns the list of all security issues in the files scanned.
- **CreateUploadUrl** - Creates the URL to upload the code files that will be scanned in CodeWhisperer for VS Code and JetBrains.

- CreateProfile - Called when the CodeWhisperer administrator creates a new CodeWhisperer application.
- UpdateProfile - Called when the CodeWhisperer administrator updates CodeWhisperer profiles.
- ListProfiles - Called when the CodeWhisperer administrator lists CodeWhisperer profiles.
- TagResource - Called when the CodeWhisperer administrator adds or creates a tag on the CodeWhisperer resource.
- UnTagResource - Called when the CodeWhisperer administrator removes a tag from the CodeWhisperer resource.
- ListTagsForResource - Called by CodeWhisperer on console page load to list tags on the CodeWhisperer resource.
- StartDataCollection - Used for starting a data collection from customer's data source to use in creating a Customization.
- GetDataCollectionStatus - Used for polling the status of a data collection job.
- CreateCustomization - Used for creating a Customization from collected customer data.
- ListCustomizations - Used for listing Customizations based on their state.
- UpdateCustomization - Activates or deactivates a customization.
- GetCustomization - Used for describing a Customization.

Language and IDE support in Amazon CodeWhisperer

Language support in Amazon CodeWhisperer

CodeWhisperer supports code generation for 15 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#

CodeWhisperer also supports code generation for:

- Ruby
- Go
- PHP
- C++
- C
- Shell
- Scala
- Rust
- Kotlin
- SQL

IDEs supported by Amazon CodeWhisperer

CodeWhisperer can be used in the following interactive development environments (IDEs).

Amazon Sagemaker Studio

Amazon Sagemaker Studio is a web-based IDE designed for machine learning workflows with AWS.

Python is the only language that CodeWhisperer supports with SageMaker.

JupyterLab

JupyterLab is an open-source, web-based IDE that provides an interactive environment for working with notebooks, code, and data.

Python is the only language that CodeWhisperer supports with JupyterLab.

Visual Studio Code

Visual Studio Code is a free, extensible code editor, created by Microsoft. Most of the source code for VS Code is available on GitHub under the MIT License, which puts limited restriction on reuse.

CodeWhisperer runs in VS Code through the AWS Toolkit, which runs as a VS Code extension.

VS Code is expected to support all programming languages that CodeWhisperer supports. In some cases, it may be necessary to install a VS Code extension related to the language that you are expecting to use.

Learn more:

- [Setting up CodeWhisperer for individual developers \(p. 22\)](#)
- [Billing at the individual tier \(p. 22\)](#)
- [Billing at the professional tier \(p. 22\)](#)
- VS Code tabs under [Billing for CodeWhisperer \(p. 79\)](#)
- [The AWS Toolkit for VS Code User Guide](#)

JetBrains

JetBrains is a software company that makes multiple IDEs.

CodeWhisperer supports all JetBrains IDEs, with the exception of JetBrains Gateway.

In the context of CodeWhisperer, all JetBrains IDEs are expected to behave the same way. Therefore, this guide does not make a distinction between the different types of JetBrains IDE. We use the name *JetBrains* to mean whichever supported JetBrains IDE that you happen to be using.

CodeWhisperer runs in JetBrains through the AWS Toolkit, which runs as a JetBrains extension.

The JetBrains IDEs that CodeWhisperer supports include, but are not limited to, the following:

- Clion (for C & C++ development)
- GoLand (for Go development)
- IntelliJ (for Java development)
- WebStorm (for Node.js development)

- Rider (for .NET development)
- PhpStorm (for PHP development)
- PyCharm (for Python development)
- RubyMine (for Ruby development)

Learn more:

- [Setting up CodeWhisperer for individual developers \(p. 22\)](#)
- [Billing at the individual tier \(p. 63\)](#)
- [Billing at the professional tier \(p. 63\)](#)
- JetBrains tabs under [Billing for CodeWhisperer \(p. 79\)](#)
- [The AWS Toolkit for JetBrains User Guide](#)

AWS Cloud9

AWS Cloud9 is an IDE that AWS offers as a service. You use it inside your web browser.

CodeWhisperer runs inside AWS Cloud9. You need an AWS account.

Learn more:

- [Setting up CodeWhisperer with AWS Cloud9 \(p. 32\)](#)
- [Billing for CodeWhisperer when used with services inside the AWS console \(p. 63\)](#)
- AWS Cloud9 tabs under [Billing for CodeWhisperer \(p. 79\)](#)
- [The AWS Cloud9 user guide](#)

AWS Lambda

Lambda is a compute service that lets you run code without provisioning or managing servers. You need an AWS account to use it, and you use it inside the AWS console. You pay for the compute time that you consume.

You organize your code into Lambda functions. You can use CodeWhisperer to help you write those functions.

Learn more:

- [Setting up Amazon CodeWhisperer with AWS Lambda \(p. 31\)](#)
- [Billing for CodeWhisperer when used with services inside the AWS console \(p. 63\)](#)
- Lambda tabs under [Billing for CodeWhisperer \(p. 79\)](#)
- [The Lambda user guide](#)

AWS Glue Studio

AWS Glue Studio is a graphical interface that makes it easy to create, run, and monitor extract, transform, and load (ETL) jobs.

You can use CodeWhisperer to help you write ETL scripts in AWS Glue Studio Notebook.

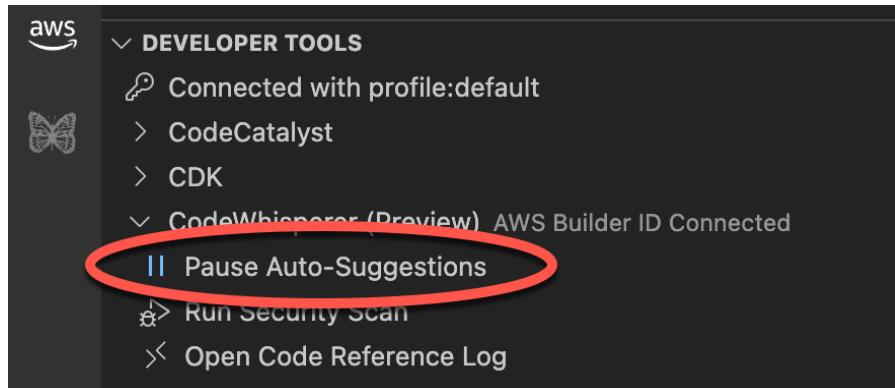
Learn more:

- [Setting up CodeWhisperer with AWS Glue Studio \(p. 31\)](#)
- [Billing for CodeWhisperer when used with services inside the AWS console \(p. 63\)](#)
- AWS Glue tabs under [Billing for CodeWhisperer \(p. 79\)](#)
- [The AWS Glue user guide](#)

Pausing suggestions with Amazon CodeWhisperer

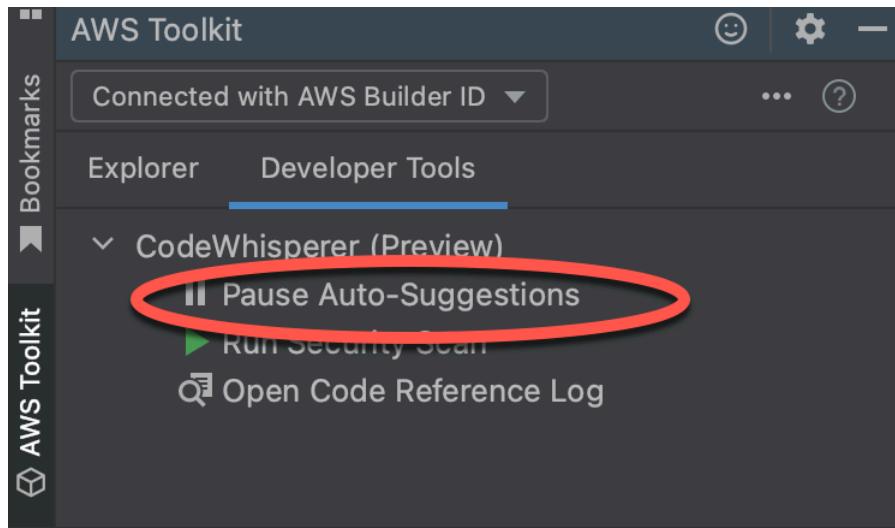
This chapter describes how to pause and resume automatic suggestions in CodeWhisperer.

VS Code



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

JetBrains



1. In JetBrains, choose the AWS logo from the left sidebar.
2. In the AWS Toolkit window, select the **Developer Tools** tab.
3. Expand the CodeWhisperer section.

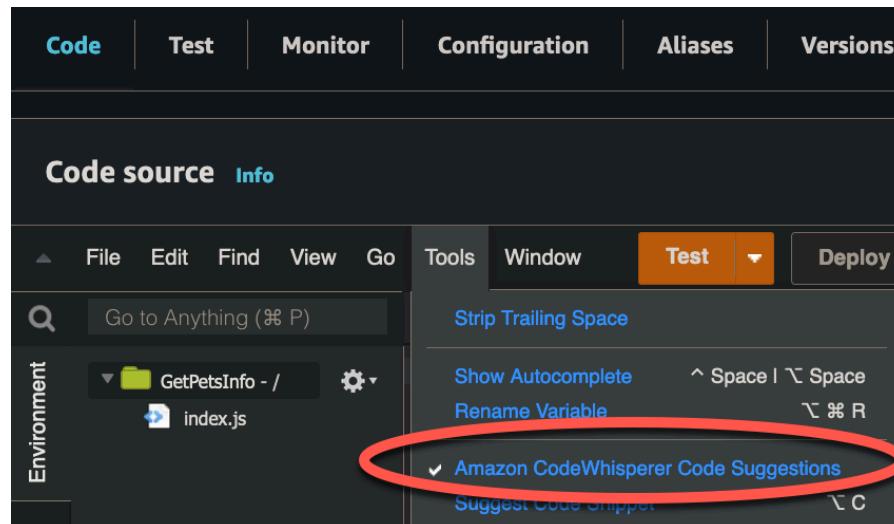
4. Choose **Pause Auto-Suggestions** or **Resume Auto-Suggestions**.

AWS Cloud9

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

To stop receiving CodeWhisperer suggestions in AWS Cloud9, remove IAM policy that gives CodeWhisperer access to AWS Cloud9 from the role or user that you are using to access AWS Cloud9. For more information, see [AWS Identity and Access Management permissions for AWS Cloud9 \(p. 32\)](#)

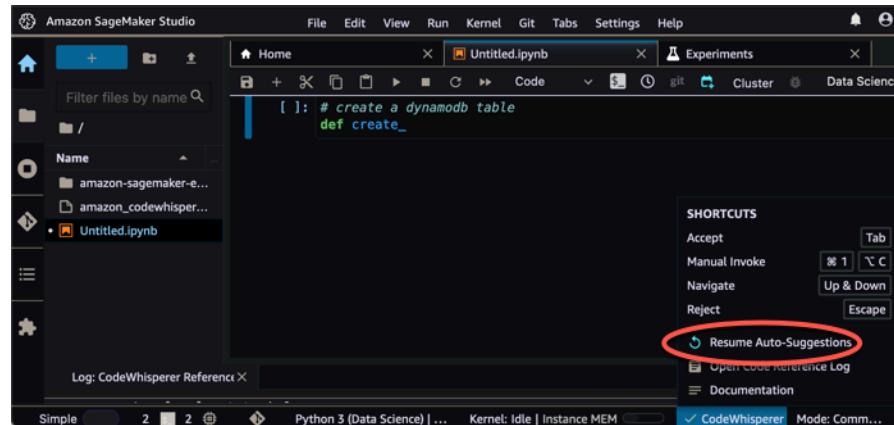
Lambda



To deactivate or re-activate CodeWhisperer code suggestions in Lambda:

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

Amazon SageMaker Studio

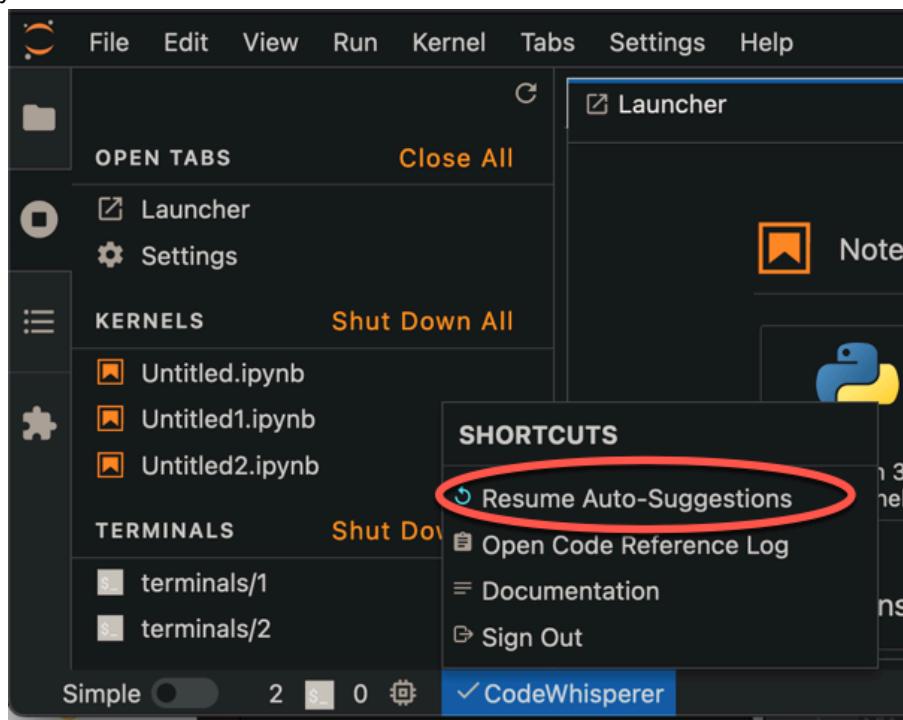


1. In the SageMaker Studio console, choose CodeWhisperer from the bottom of the window.

The CodeWhisperer panel will open.

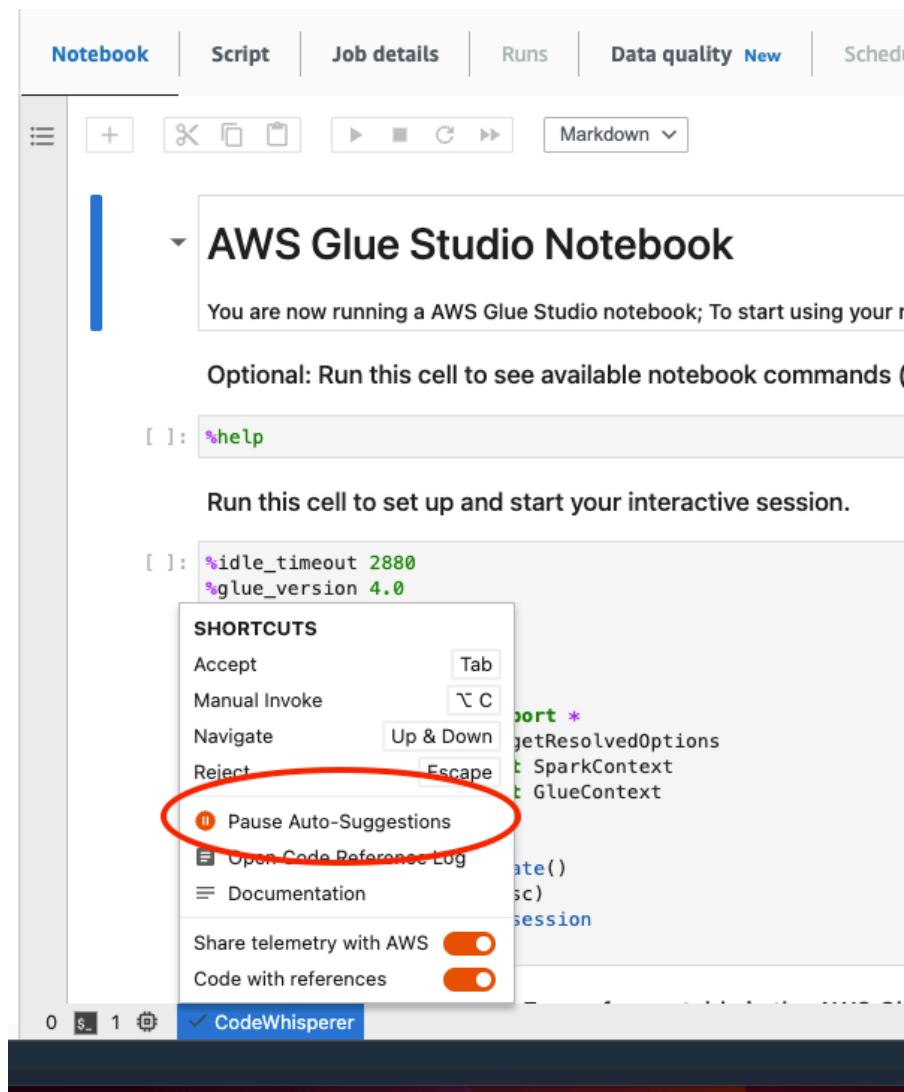
2. Choose **Pause Auto-Suggestions** or **Resume Auto-Suggestions**.

JupyterLab



1. In the JupyterLab console, choose CodeWhisperer from the bottom of the window.
The CodeWhisperer panel will open.
2. Choose **Pause Auto-Suggestions** or **Resume Auto-Suggestions**.

AWS Glue Studio Notebook



1. In the AWS Glue Studio Notebook console, choose CodeWhisperer from the bottom of the window.
The CodeWhisperer panel will open.
2. Choose **Pause Auto-Suggestions** or **Resume Auto-Suggestions**.

Security scans

Run Security Scan performs a security scan on currently active file in the IDE editor, and its dependent files from the project. After the scan is finished, security issues in the scanned files are highlighted in the **Problems** panel in VSC. Note that for JetBrains, security issues are highlighted in a separate **CodeWhisperer Security Issues** tab in the **Problems** panel.

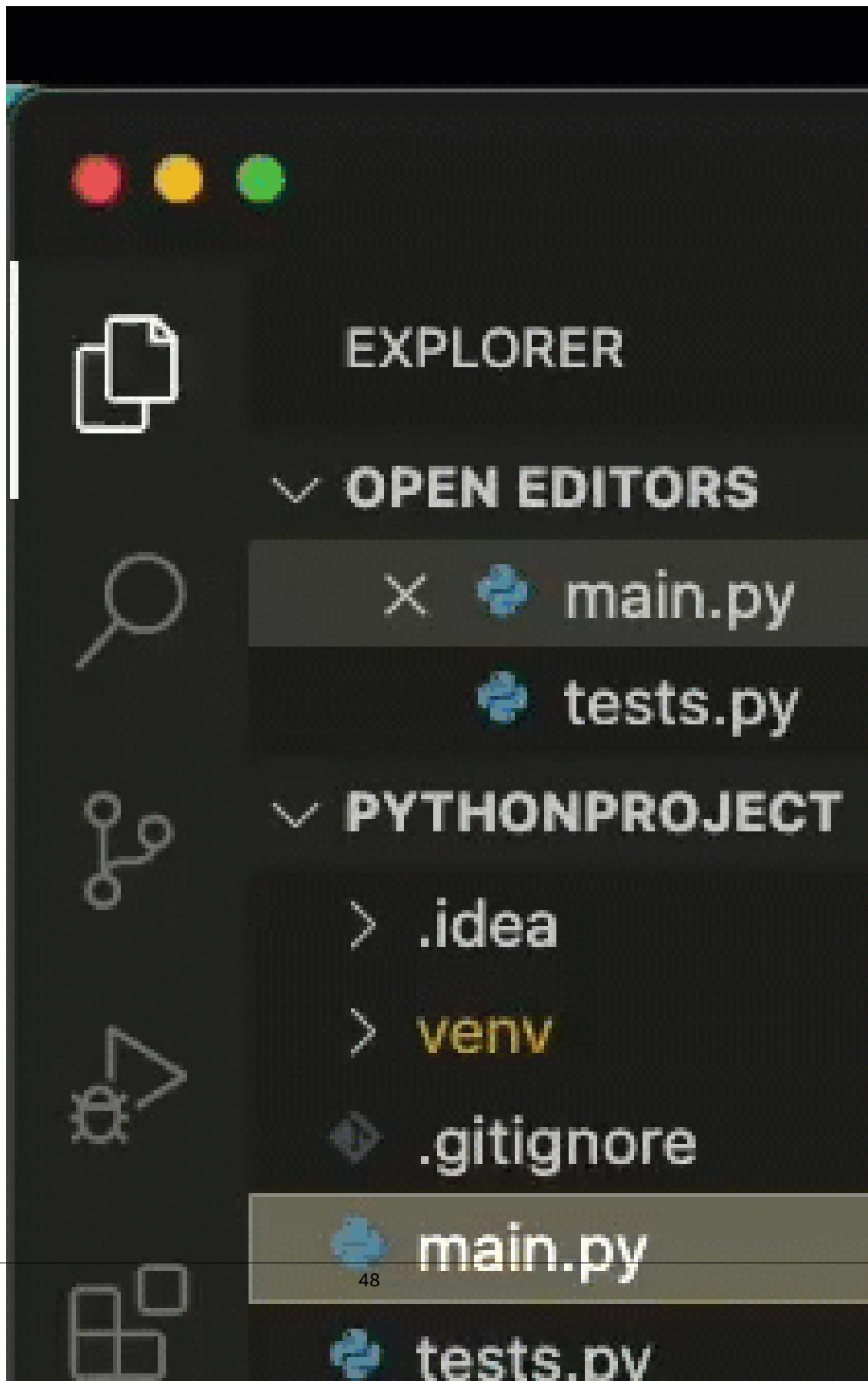
CodeWhisperer's security scan is integrated with [Amazon CodeGuru](#). CodeGuru does multiple layers of filtering before scanning code to ensure that you can focus on the most critical issues. As part of that, CodeGuru filters unsupported languages, and also tests code, including open source code, before scanning for security issues.

Topics

- [Activating security scans \(p. 48\)](#)
- [Security scan data limits \(p. 51\)](#)

Activating security scans

AWS Toolkit for Visual Studio Code



To begin a security scan in VS Code, use the following procedure.

1. In VS Code, choose the AWS logo on the left side of the window. The AWS Toolkit panel will open.
2. In the AWS Toolkit panel, under **Developer Tools**, under **CodeWhisperer**, choose **Start Security Scan**.

A scan can take up to 60 seconds. You may choose to stop an ongoing security scan by selecting **Stop Security Scan**. Note that, once started, a scan is counted towards your monthly (per user) security scans usage limits. For more information, see [Security scan data limits \(p. 51\)](#).

Note

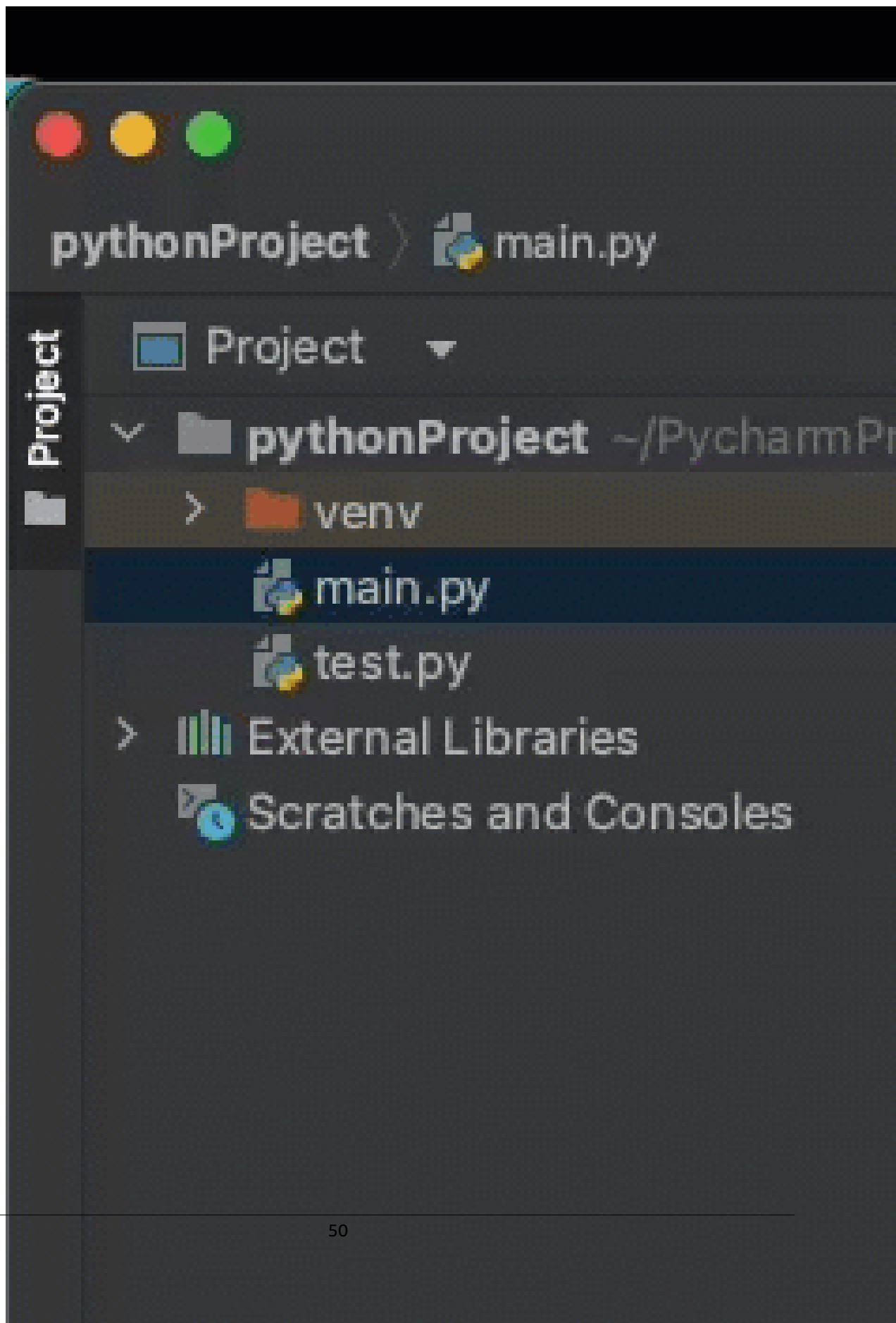
If you are running a security scan on a Java file or project, the build artifacts (.class files) are required. If you are running into issues with scanning your Java file or project, check the following:

1. Make sure your project structure is valid for the build system that you are using.
2. Build your project in VS Code before a running security scan, to ensure that CodeWhisperer has access to your build artifacts.

Note

If your project has built successfully in VS Code, but the Security Scan fails with an error message: Cannot find build artifacts for the project, troubleshoot the error by specifying the location of your build artifacts in the compiler output path.

AWS Toolkit for JetBrains



To begin a security scan in JetBrains, use the following procedure.

1. In JetBrains, choose the AWS logo on the left side of the window. The AWS Toolkit panel will open.
2. In the AWS Toolkit panel, under **Developer Tools**, under **CodeWhisperer**, choose **Start Security Scan**.

A scan can take up to 60 seconds. You may choose to stop an ongoing security scan by selecting **Stop Security Scan**. Note that, once started, a scan is counted towards your monthly (per user) security scans usage limits. For more information, see [Security scan data limits \(p. 51\)](#).

Run security scan performs a security scan on the project or file, and its dependencies, that are currently active in IntelliJ. After the scan is finished, security issues are displayed in the **CodeWhisperer Security Issues** tab of the **Problems** panel.

Note

To run a security scan on a Java file or project, the build artifacts (.class files) are required.

1. Make sure your project structure is valid for the build system that you are using.
2. Build your project in IntelliJ before running a security scan, to ensure that CodeWhisperer has access to your build artifacts.

If your project has built successfully in IntelliJ, but the Security Scan fails with an error message: Can not find build artifacts for the project, troubleshoot the error by specifying the location of your build artifacts in the compiler output path, as described below:

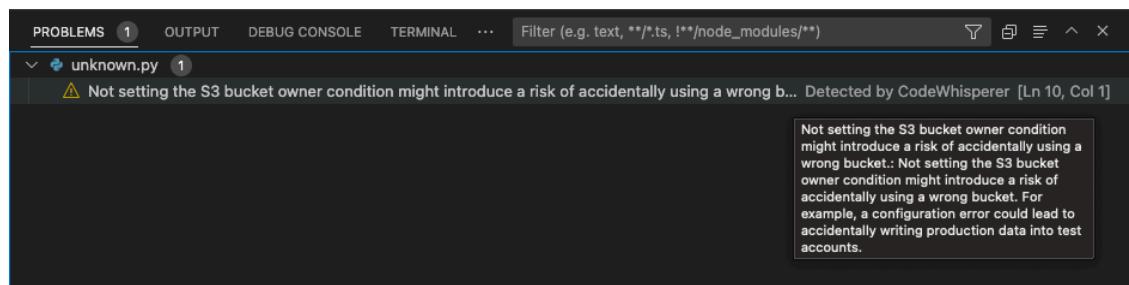
1. From the IntelliJ main menu, expand **File** (Windows) or open **Preferences** (Mac).
2. Choose **Project Structure** to open the **Project Structure** navigation pane.
3. Choose **Project** to open the **Project** pane.
4. Enter or select the location of your project's artifact files from the **Compiler output** field.

Security scan data limits

Each security scan may include more than one file. However, the amount of data that can be scanned, per scan, is limited. The limits are subject to regular change by AWS, and they also vary by programming language. If your project exceeds this data limit, then not all of your files will be scanned. After a scan, you can check the log to see the files that were scanned by selecting **Show Scanned Files**. If the file you are interested in is not scanned because of the data limits, open the file in IDE and start another scan to ensure that this file is included in the scan.

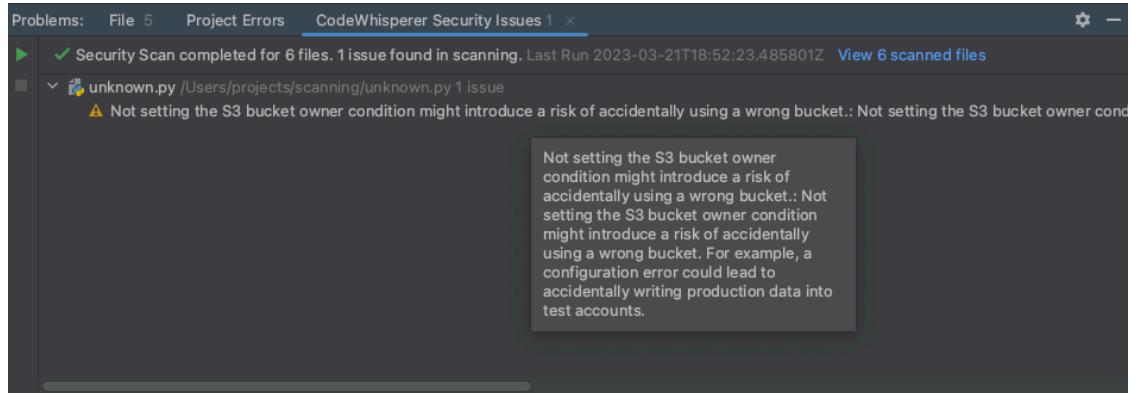
AWS Toolkit for Visual Studio Code

This screenshot shows what the list of scanned files looks like in VS Code.



AWS Toolkit for JetBrains

This screenshot shows what the list of scanned files looks like in JetBrains.



Code references

Topics

- [Viewing code references \(p. 52\)](#)
- [Turning code references off and on \(p. 55\)](#)
- [Opting out of code with references \(p. 58\)](#)

Viewing code references

CodeWhisperer 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 CodeWhisperer.

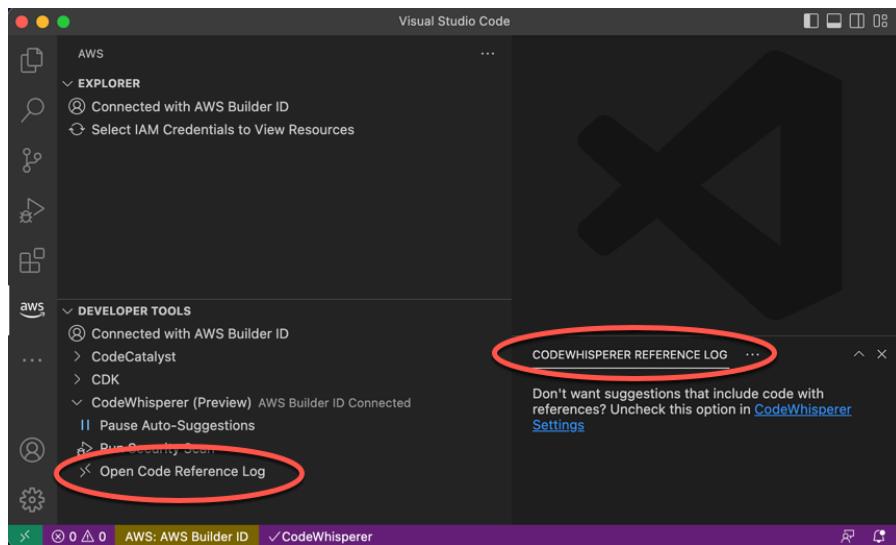
This chapter explains how to view the code references.

AWS Toolkit for Visual Studio Code

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

1. Make sure you are using the latest version of both VS Code and the AWS Toolkit.
2. In VS Code, choose the AWS logo from the left side of the window.
3. Open the **Developer Tools** dropdown menu.
4. Choose **Open Code Reference Log**.

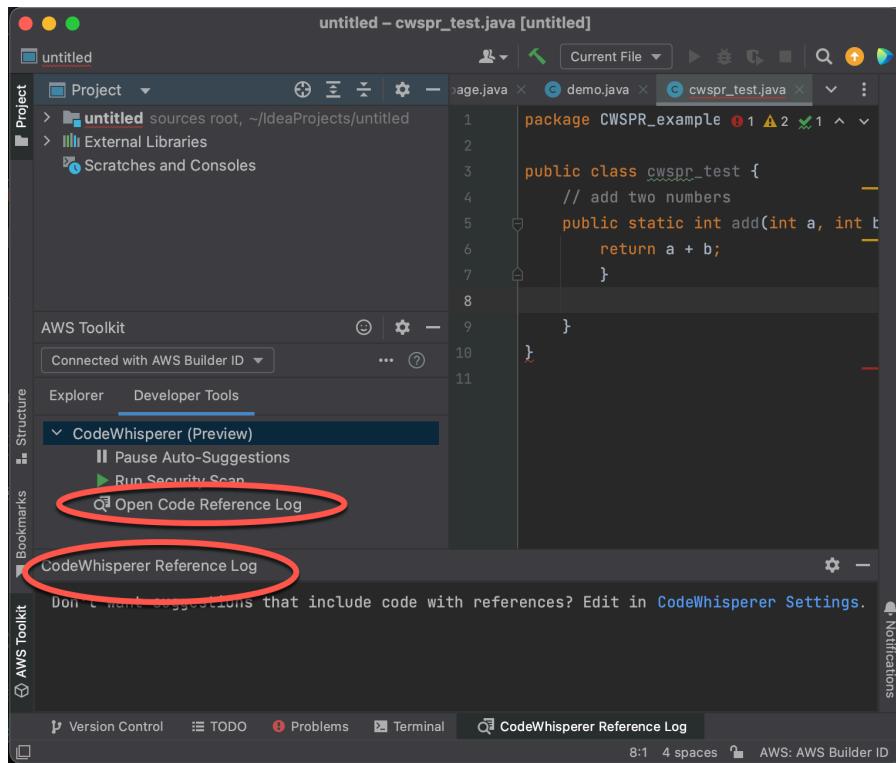
The code reference log will appear in the lower right part of the VS Code window.



AWS Toolkit for JetBrains

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

1. Make sure you are using the latest version of both JetBrains and the AWS Toolkit.
2. On the left side of the JetBrains window, choose the AWS logo.
3. In the AWS Toolkit panel, select the **Developer Tools** tab.
4. In the CodeWhisperer dropdown, choose **Open Code Reference Log**.

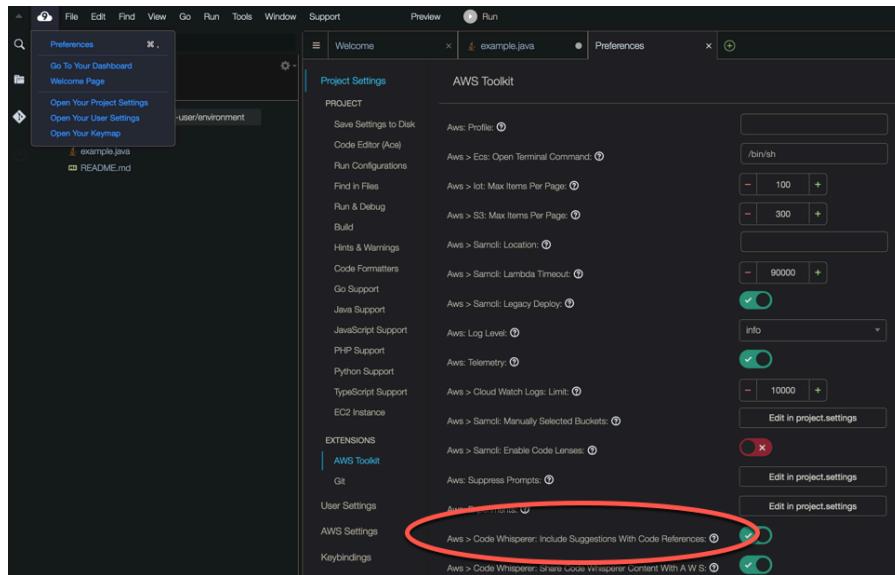


AWS Cloud 9

When you use CodeWhisperer 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 **Aws > Code Whisperer: Include Suggestions With Code References**.



Lambda

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

SageMaker Studio

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

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

JupyterLab

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

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

AWS Glue Studio Notebook

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

1. At the bottom of the AWS Glue Studio Notebook window, open the CodeWhisperer 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 CodeWhisperer.

This section explains how to use the code reference options.

AWS Toolkit for Visual Studio Code

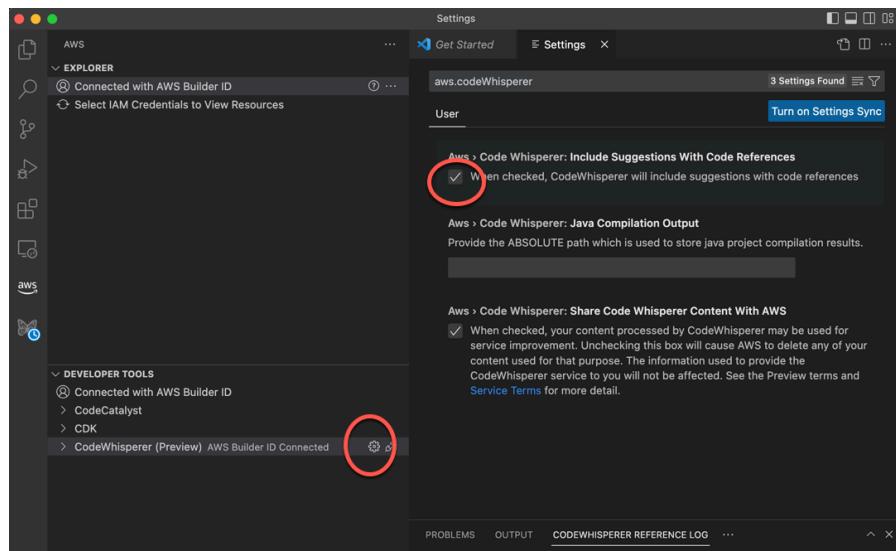
When you use CodeWhisperer with VS Code, code references are on by default.

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 the AWS Toolkit.
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 **CodeWhisperer** option, choose the gear icon.

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

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



AWS Toolkit for JetBrains

When you use CodeWhisperer 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 AWS Toolkit.
2. In IntelliJ, open **Preferences**.
3. In the **Preferences** window, under **Tools**, under **AWS**, select CodeWhisperer.

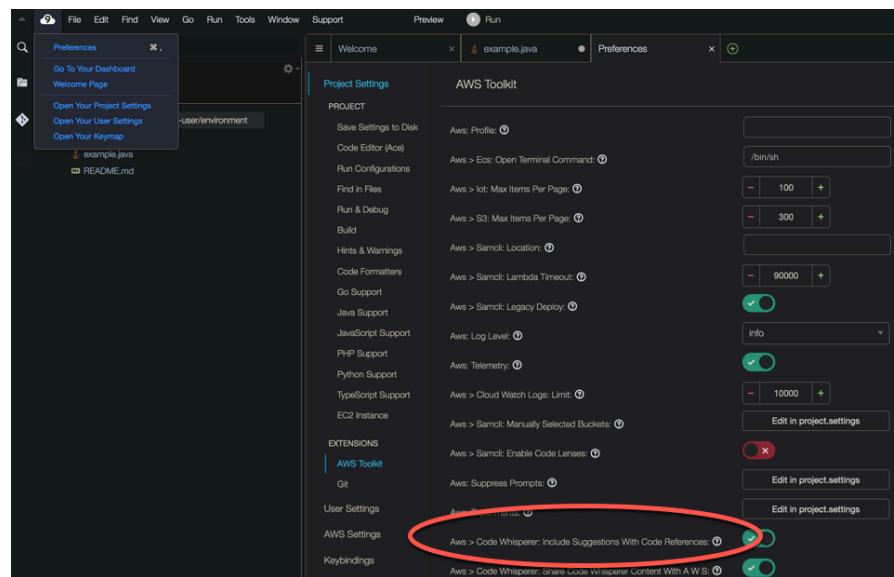
4. In the CodeWhisperer panel on the right, select or deselect the box labeled **Include suggestions with code references**.

AWS Cloud 9

When you use CodeWhisperer 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 **CodeWhisperer: Include Suggestions With Code References**.



Lambda

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

SageMaker Studio

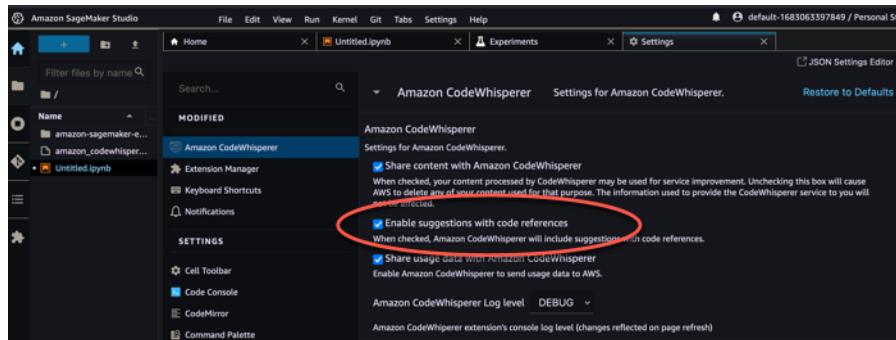
When you use CodeWhisperer with SageMaker Studio, code references are on by default.

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 CodeWhisperer dropdown, select or deselect the box next to **Enable suggestions with code references**.

CodeWhisperer User Guide

Toggling code references

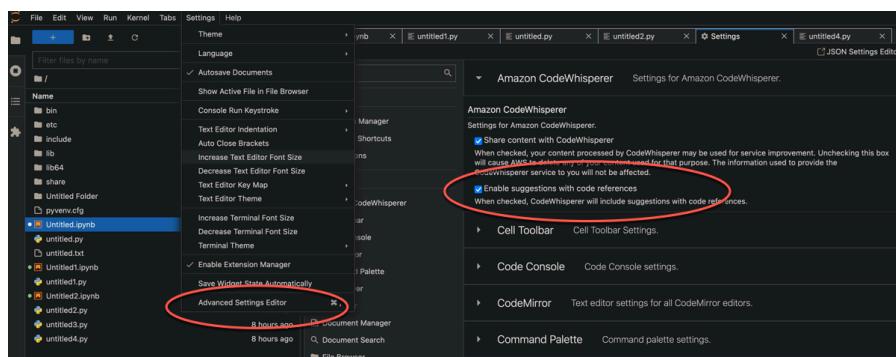


JupyterLab

When you use CodeWhisperer 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 CodeWhisperer dropdown, select or deselect the box next to **Enable suggestions with code references**.

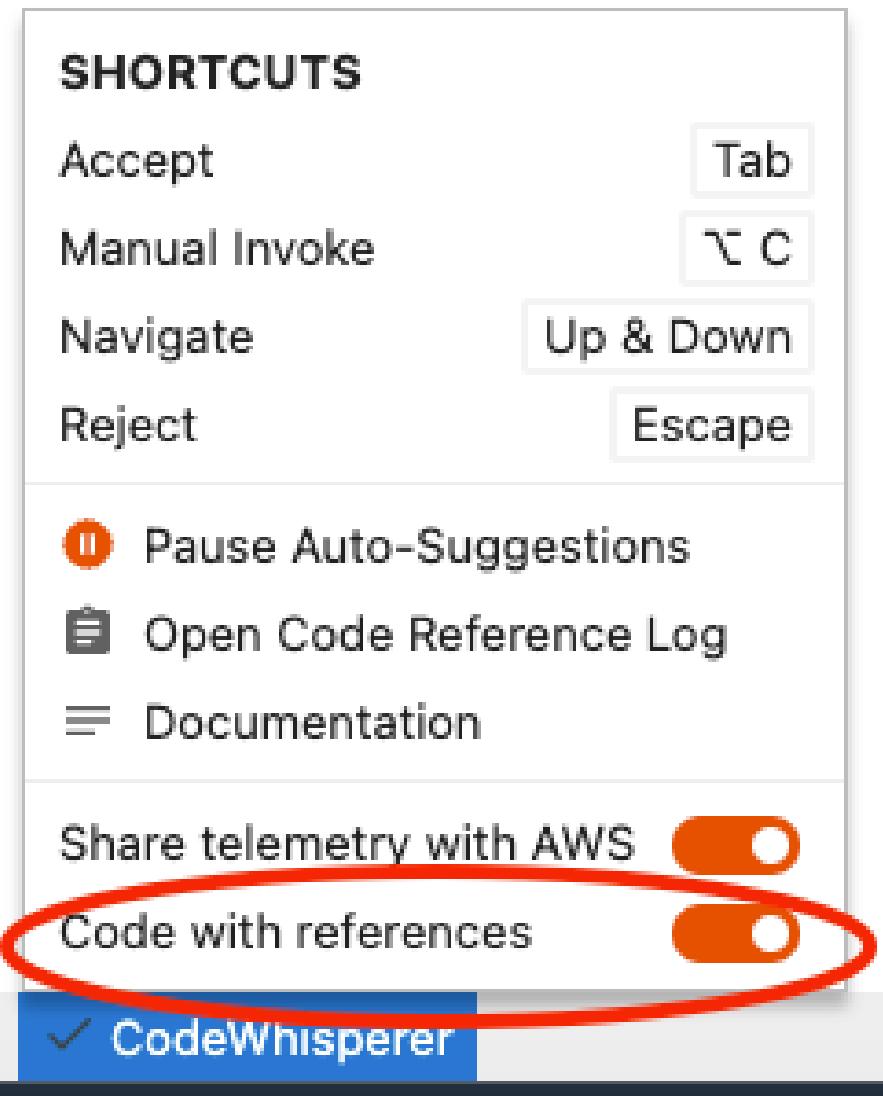


AWS Glue Studio Notebook

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

Note

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



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.

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. On the main CodeWhisperer console page, choose **Set up CodeWhisperer**.

2. On the setup page, under **Suggestions** deselect the box labeled **Include suggestions with code references**.
3. At the bottom of the console window, choose **Set up CodeWhisperer**.

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. On the main CodeWhisperer console page, choose **Set up CodeWhisperer**.
2. On the setup page, under **Suggestions** deselect the box labeled **Include suggestions with code references**.
3. At the bottom of the console window, choose **Set up CodeWhisperer**.

AWS Cloud 9

CodeWhisperer 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

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

SageMaker Studio

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

JupyterLab

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

AWS Glue Studio Notebook

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

Types of users for CodeWhisperer

There are multiple scenarios under which you may come to use CodeWhisperer. Understanding how your situation differs from the situation of other customers may help you understand issues related to authentication, IDE choices, and billing. This page explains the differences between the types of CodeWhisperer users.

Professional-tier developers are users who work for an enterprise (that is, a company), and it's the enterprise, not the individual, who has a financial relationship with AWS.

Root user (of a whole AWS account)

The root user is the most powerful user in the AWS account. When a customer first sets up an AWS account, the root user is the only user. Because the root user is so powerful, it should be used very infrequently. The root user should create administrative users, and then those administrative users should be used for the majority of account management tasks.

IAM Identity Center administrator

The root user creates the IAM Identity Center administrator. The IAM Identity Center administrator is in charge of adding users to the account through the IAM Identity Center. The person who logs in as the IAM Identity Center administrator may work in human resources. They may not have a direct relationship with CodeWhisperer. They also probably manage users for the same professional who are using AWS services other than CodeWhisperer. Some, but probably not all, of the users managed by the IAM Identity Center administrator will become CodeWhisperer professional developers.

CodeWhisperer administrator

The root user creates the CodeWhisperer administrator. The CodeWhisperer administrator decides which users should have access to CodeWhisperer as professional developers. The pool of users from which the CodeWhisperer administrator picks these users, is the pool of users created by the people administrator. The CodeWhisperer administrator might not be a developer, and they might not use CodeWhisperer themselves at all.

Professional-tier developer (using a third-party IDE)

The IAM Identity Center manager adds the professional-tier developer to the IAM Identity Center. Then the CodeWhisperer administrator gives the professional-tier developer access to CodeWhisperer. Then the professional-tier developer uses CodeWhisperer through the AWS Toolkit in either VS Code or JetBrains.

Individual-tier developer (using a third-party IDE)

The individual-tier developer does not use CodeWhisperer on behalf of a professional. Therefore, they are in charge of their own access. This developer authenticates with Builder ID, which does not require an AWS account.

In-console developer

An in-console developer uses CodeWhisperer inside AWS Cloud 9, Lambda, Sagemaker Studio, or AWS Glue Studio within the AWS console. This developer logs in as a user who is created in IAM (not IAM Identity Center). Typically, this developer is using their personal AWS account. This account owner may also act as their own administrator. In that case, they may have created the in-console developer IAM user themselves, while logged in as the root user (not recommended), or (best practice), a user that acts as a general AWS account administrator.

Billing for CodeWhisperer

This page describes the different tiers of CodeWhisperer usage from a billing perspective.

Individual tier

The individual tier is free and easy to set up, but does not include the benefits of organizational license management.

If you are using CodeWhisperer at the individual tier, then:

- You use CodeWhisperer with the AWS Toolkit in either VS Code or JetBrains, or with JupyterLab.
- You authenticate with Builder ID.
- You control your own reference tracker settings.
- You have access to code generation for all supported languages.
- By default, you share code fragment data with AWS. You can opt out of this in the IDE settings.
- By default, you share telemetry data with AWS. [You can opt out of this \(p. 79\)](#) in the IDE settings.
- You can run up to 50 security scans per month.

Professional tier

The professional tier includes a charge for additional features. Your employer pays the bill through their company AWS account.

The professional tier offers administrative capabilities to organizations that want to enable their developers to use CodeWhisperer. At the professional tier, the CodeWhisperer administrator is empowered by the organization to centrally manage which developers in the organization should have access to CodeWhisperer. The CodeWhisperer administrator also sets policies at the organizational level, such as whether developers are allowed to receive code recommendations that are similar to open source training data.

If you are using CodeWhisperer at the professional tier, then:

- You use CodeWhisperer with the AWS Toolkit in either VS Code or JetBrains.
- You authenticate with credentials set up by your employer's AWS account's IAM Identity Center administrator in IAM Identity Center.
- You don't use Builder ID.
- Your administrator controls the reference tracker settings.
- You have access to code generation for all supported languages.
- You do not share code fragment data with AWS.
- By default, you share telemetry data with AWS. [You can opt out of this \(p. 79\)](#) in the IDE settings.
- You can run up to 500 security scans per month.

For pricing details, see [the CodeWhisperer pricing page](#).

Billing for CodeWhisperer when used with services inside the AWS console

The following services work with CodeWhisperer inside the AWS console (as opposed to a third-party IDE):

- AWS Cloud9
- AWS Lambda
- SageMaker Studio
- AWS Glue Studio

If you are using CodeWhisperer with any of those services, then:

- You are *not* using CodeWhisperer with the AWS Toolkit in either VS Code or JetBrains.
- You authenticate by logging directly into the AWS console using IAM credentials set up by your employer's AWS account's IAM Identity Center administrator. (If you are using a personal AWS account, then you can set up those credentials yourself.)
- You don't use Builder ID.
- There is no additional charge for using CodeWhisperer.
- You cannot run CodeWhisperer-related security scans from inside AWS Cloud9, Lambda, SageMaker Studio, or AWS Glue Studio Notebook.

Monitoring Amazon CodeWhisperer

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

- *Amazon CloudWatch* monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track the number of times that CodeWhisperer has been invoked on your account, or the number of daily active users. For more information, see the [Amazon CloudWatch User Guide](#).

Monitoring CodeWhisperer with Amazon CloudWatch

You can monitor CodeWhisperer 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 your web application or service 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 CodeWhisperer 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, CompletionType	GeneratedLineCount	You want to determine the number of number of lines generated by CodeWhisperer.
ProgrammingLanguage, SuggestionState, CompletionType	SuggestionReferenceCount	You want to determine the number of 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.

Dimensions	Metric	Use case or explanation
Programming Language	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.

Code examples

Topics

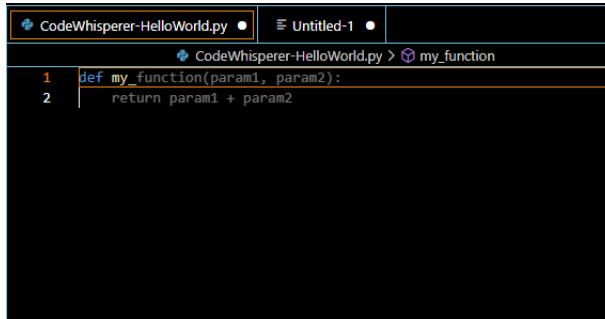
- [Single-line code completion \(p. 66\)](#)
- [Full function generation \(p. 67\)](#)
- [Block completion \(p. 74\)](#)
- [Docstring and Javadoc completion \(p. 75\)](#)
- [Line-by-line recommendations \(p. 76\)](#)
- [Single-line comment completion \(p. 77\)](#)

Single-line code completion

AWS Toolkit for Visual Studio Code

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

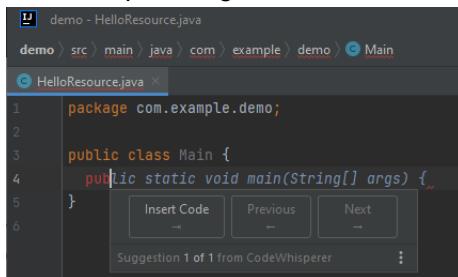
In the image below, a user has started to type out a line of code. Based on the input, CodeWhisperer has generated suggestions to complete the line. The user can cycle through the suggestions using the arrow keys.



AWS Toolkit for JetBrains

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

In the image below, a user has started to input the string `public` (shown in red text). Based on the input, CodeWhisperer has generated a list of suggestions. The current suggestion (`public`) is in line with the input string.



Lambda

When you start typing out single lines of code, CodeWhisperer makes suggestions based on your current and previous inputs. In the image below, a user has begun to define a variable for an Amazon S3 client. Based on this, CodeWhisperer then suggests a way to complete this line of code.

```
s3_client = |  
            boto3.client('s3')
```

As another example, in the image below, a user has already written some code, and now wants to send a message to an Amazon SQS queue. CodeWhisperer suggests a way to complete this final line of code.

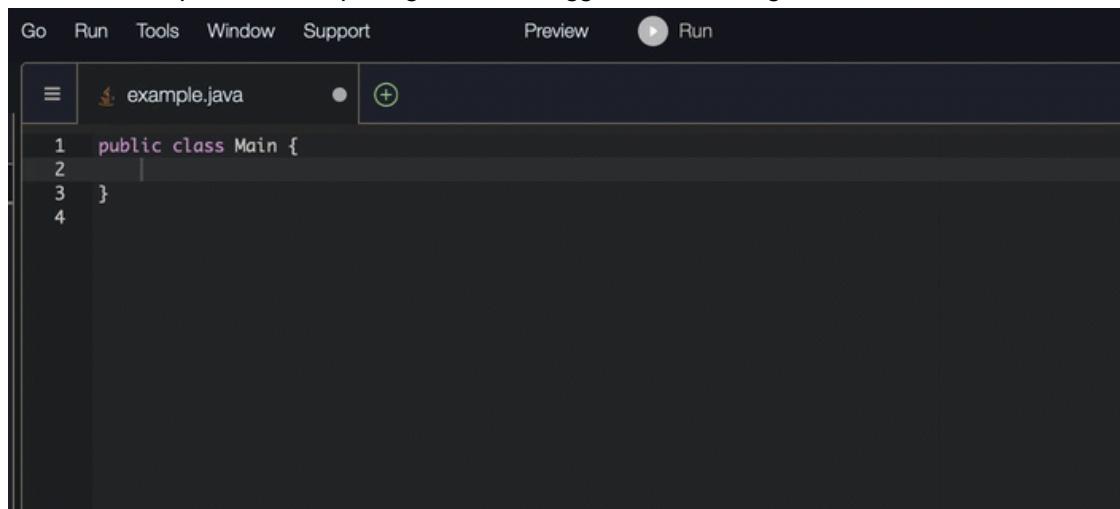
```
    sqs = boto3.client('sns')  
    message = "message"  
    queue_url = "https://example.com"  
    sqs.sendMessage|  
        (QueueUrl=queue_url, MessageBody=message)
```

AWS Cloud9

When you start typing out single lines of code, CodeWhisperer 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, CodeWhisperer generates a suggestion for the signature of the main method.



Full function generation

Lambda

CodeWhisperer can generate an entire function based on your function signature or code comments. In the following image, a user has written a function signature for reading a file from Amazon S3. Amazon CodeWhisperer 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, CodeWhisperer 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. CodeWhisperer 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)
```

CodeWhisperer 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. CodeWhisperer 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]

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

In the following image, a user has written a comment. Based on this comment, CodeWhisperer 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. CodeWhisperer 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
```

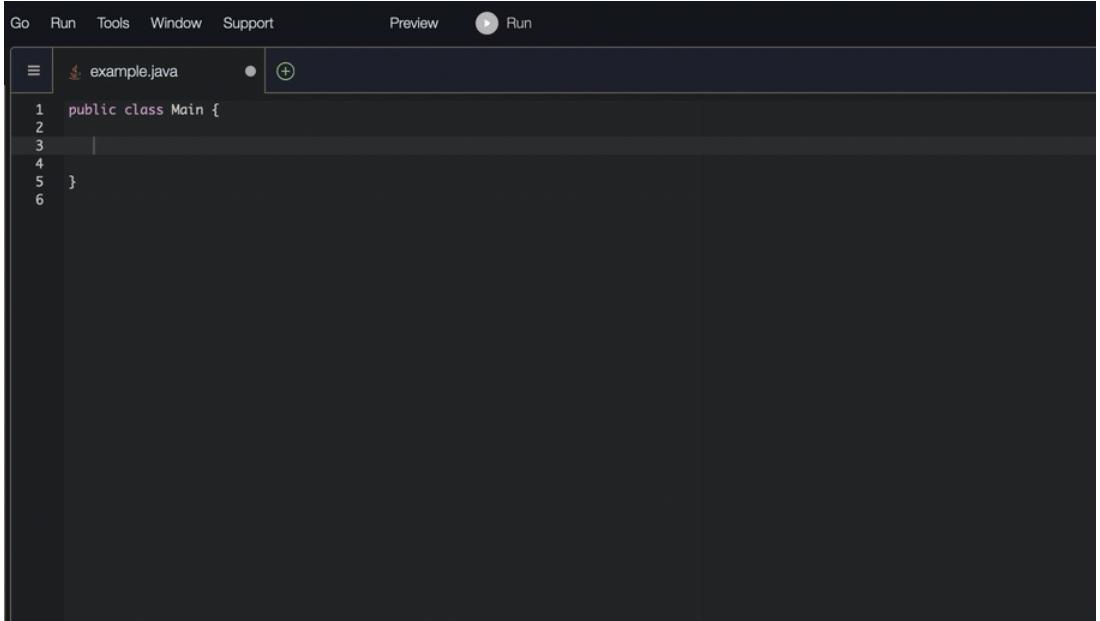
AWS Cloud9

CodeWhisperer can generate an entire function based on a comment that you've written. As you finish your comment CodeWhisperer will suggest a function signature. If you accept the suggestion, CodeWhisperer 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, CodeWhisperer will refactor based on your input.

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

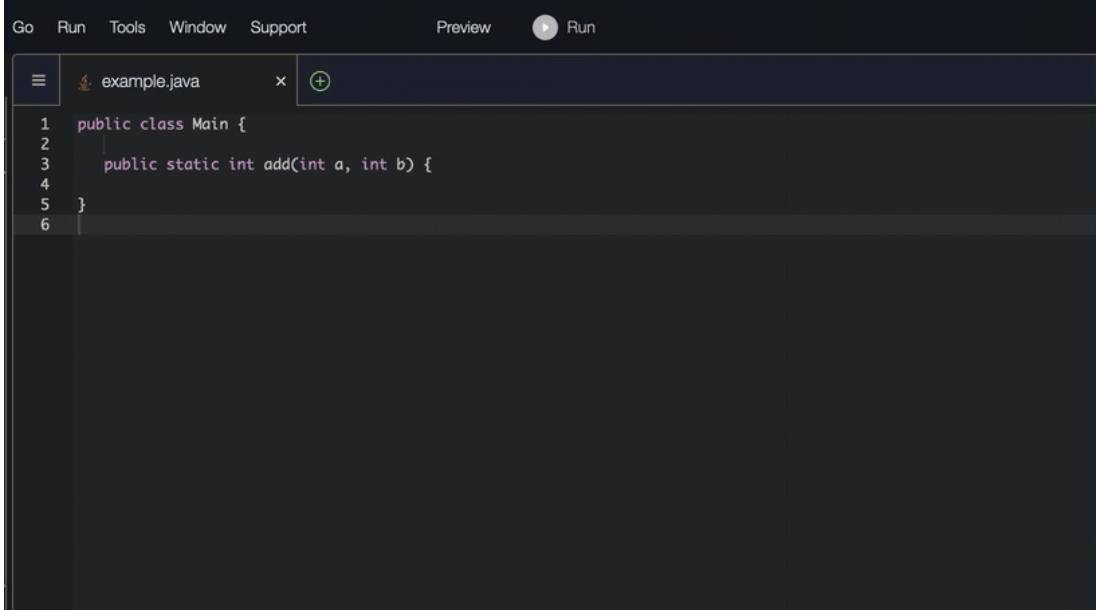
1. In the example below, in Java, a user inputs a comment. CodeWhisperer suggests a function signature.

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



```
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 CodeWhisperer. On the following line, CodeWhisperer generates a suggestion based on the comment.



```
1 public class Main {  
2  
3  public static int add(int a, int b) {  
4  
5 }  
6
```

AWS Toolkit for Visual Studio Code

CodeWhisperer can generate an entire function based on a comment that you've written. As you finish your comment CodeWhisperer will suggest a function signature. If you accept the suggestion, CodeWhisperer 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, CodeWhisperer will update recommendations based on your input.

The following list contains examples of how CodeWhisperer 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 below the comment is a suggestion from CodeWhisperer.

The screenshot shows a code editor window with two tabs: 'CodeWhisperer-HelloWorld.py 5' and 'Untitled-1'. The 'CodeWhisperer-HelloWorld.py 5' tab contains the following Python code:

```
1 # add two numbers
2 def add(self, a, b):
```

The cursor is positioned at the end of the word 'add' in the function definition. A tooltip or suggestion box is visible above the code, indicating a function signature.

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

The screenshot shows the same code editor window after accepting the suggestion. The code now includes a return statement:

```
1 # add two numbers
2 def add(self, a, b):
3     return a + b
```

The cursor is now at the end of the 'return' statement. A tooltip or suggestion box is visible above the code, indicating a new suggestion for the function body.

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

The screenshot shows the code editor with the following Python code:

```
1 x = 1
2 y = 2
3 z = 3
4
5 print(
6     #add x to z and -y
7     x + z - y
8 )
```

A user comment '#add x to z and -y' is present in the code. A tooltip or suggestion box is visible above the code, indicating a new suggestion based on the user's comment.

AWS Toolkit for JetBrains

CodeWhisperer can generate an entire function based on a comment that you've written. As you finish your comment CodeWhisperer will suggest a function signature. If you accept the suggestion, CodeWhisperer 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, CodeWhisperer will refactor based on your input.

The following list contains examples of how CodeWhisperer 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 CodeWhisperer.

A screenshot of a Java code editor showing a file named `HelloResource.java`. The code defines a class `Main` with a static method `add`. A tooltip box is open over the method signature, containing three buttons: `Insert Code`, `Previous`, and `Next`. Below the buttons, the text "Suggestion 2 of 2 from CodeWhisperer" is displayed.

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

A screenshot of a Java code editor showing the same `HelloResource.java` file. The function `add` now contains the code `return a + b;`. A tooltip box is open over the closing brace of the function body, containing three buttons: `Insert Code`, `Previous`, and `Next`. Below the buttons, the text "Suggestion 1 of 1 from CodeWhisperer" is displayed.

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

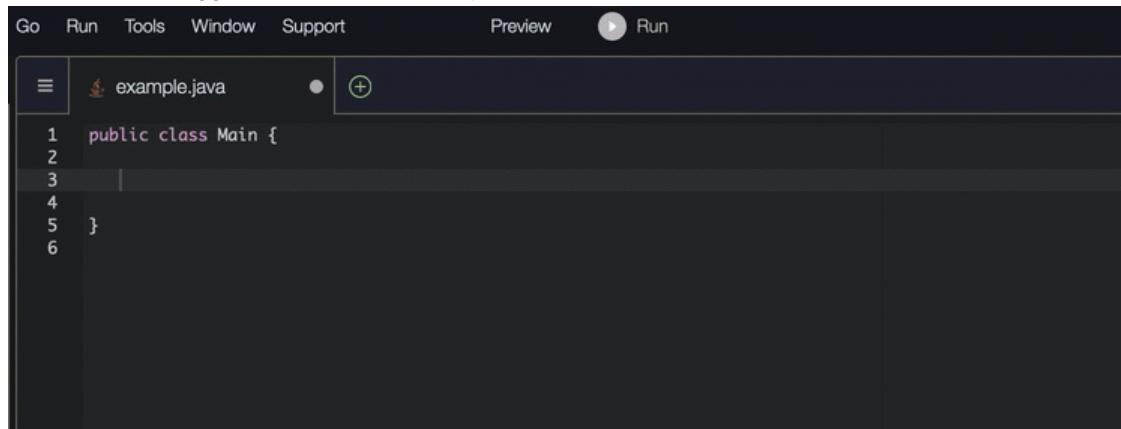
A screenshot of a Java code editor showing the same `HelloResource.java` file. The function `add` now contains the code `// add a and b together` followed by the generated suggestion `return a + b;`. A tooltip box is open over the generated code, containing three buttons: `Insert Code`, `Previous`, and `Next`. Below the buttons, the text "Suggestion 1 of 1 from CodeWhisperer" is displayed.

Block completion

AWS Cloud9

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

In the example below, in Java, a user enters the signature of an if statement. The body of the statement is a suggestion from CodeWhisperer.



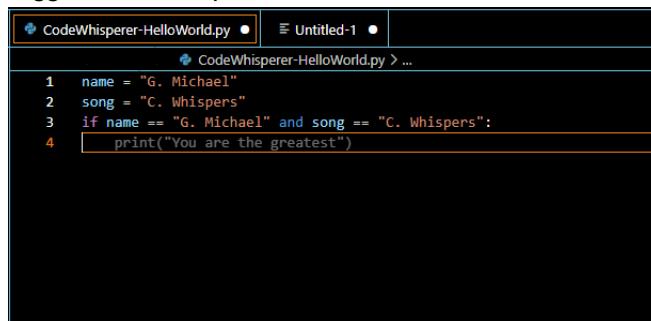
The screenshot shows the AWS Cloud9 IDE interface. The top navigation bar includes 'Go', 'Run', 'Tools', 'Window', 'Support', 'Preview', and 'Run' buttons. A file tab labeled 'example.java' is open. The code editor displays the following Java code:

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

AWS Toolkit for Visual Studio Code

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

In the image below, a user has input the signature of an if statement. CodeWhisperer is making a suggestion to complete the function.



The screenshot shows the AWS Toolkit for Visual Studio Code interface. The top navigation bar includes tabs for 'CodeWhisperer-HelloWorld.py' and 'Untitled-1'. The code editor displays the following Python code:

```
1 name = "G. Michael"  
2 song = "C. Whispers"  
3 if name == "G. Michael" and song == "C. Whispers":  
4     print("You are the greatest")
```

AWS Toolkit for JetBrains

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

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 CodeWhisperer.

A screenshot of the AWS Cloud9 IDE interface. The file being edited is `HelloResource.java`. The code contains a simple if-statement. A tooltip from CodeWhisperer is displayed over the closing brace of the if-block, suggesting the addition of a docstring. The tooltip includes buttons for "Insert Code", "Previous", and "Next", and a message "Suggestion 1 of 3 from CodeWhisperer".

Docstring and Javadoc completion

AWS Cloud9

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

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

A screenshot of the AWS Toolkit for JetBrains IDE. The file `example.java` contains a class definition with a constructor that has a parameter annotated with `@param url`. A tooltip from CodeWhisperer is shown, suggesting the addition of a docstring for the `url` parameter. The tooltip includes buttons for "Insert Code", "Previous", and "Next", and a message "Suggestion 1 of 3 from CodeWhisperer".

AWS Toolkit for JetBrains

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

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

The screenshot shows a Java code editor with a file named `HelloResource.java`. The code defines a class `Main` with a method `getResource`. A docstring is being completed, starting with `/**` and followed by several annotations: `@param url`, `@param name`, `@return`, and `@see IMAGE`. A tooltip box is open at the bottom of the screen, containing three buttons: `Insert Code`, `Previous`, and `Next`. Below the buttons, it says "Suggestion 1 of 2 from CodeWhisperer".

Line-by-line recommendations

Lambda

Depending on your use case, CodeWhisperer may not be able to generate an entire function block in one recommendation. However, CodeWhisperer can still provide line-by-line recommendations. 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, CodeWhisperer is only able to suggest the client initialization code in its first recommendation, as shown in the following image.

The screenshot shows a code editor with a comment at the top: "# Publish a message to a CloudWatch Logs". Below the comment, there is a line of code: `client = boto3.client('logs')`. This line is highlighted with a light gray background, indicating it is a recommended line of code.

However, if the user continues to request line-by-line recommendations, CodeWhisperer 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 CodeWhisperer makes suggestions, remember to rename any constants as required.

CodeWhisperer 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

Single-line comment completion

AWS Toolkit for Visual Studio Code

In the image below, the user has started to input a comment, CodeWhisperer has generated suggestions to complete the comment.

The screenshot shows a code editor interface with two tabs: "CodeWhisperer-HelloWorld.py" and "Untitled-1". The "CodeWhisperer-HelloWorld.py" tab is active and displays the following Python code:

```
1 x = 1
2 y = 2
3 z = 3
4
5 print(
6     #add three numbers
7
8 )
9
```

The cursor is positioned at the start of the sixth line, where the code "print(" is followed by a multi-line comment "#add three numbers". A tooltip or completion suggestion box is visible over the comment line, containing the text "#add three numbers".

Sharing your data with AWS

When you use CodeWhisperer, AWS may, for service improvement purposes, store data about your usage and content. This page explains how to opt out of sharing that data.

The data that AWS may collect with CodeWhisperer includes your client-side telemetry and your content.

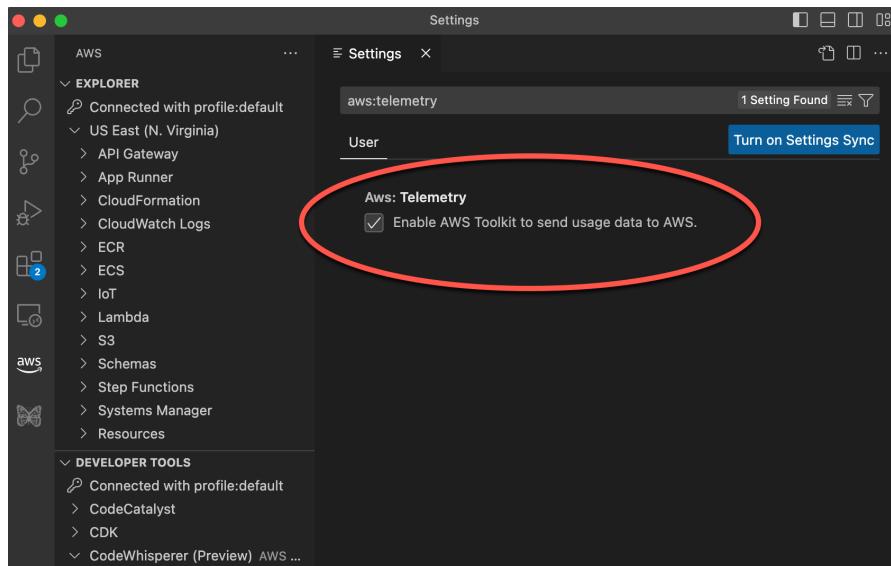
Your content includes the parts of your code that CodeWhisperer uses to generate suggestions, as well as the content of the suggestions themselves. **At the professional tier, CodeWhisperer does not collect your content for service improvement purposes.**

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.

Opting out of sharing your client-side telemetry

AWS Toolkit for Visual Studio Code

1. In VS Code, choose the AWS logo from the side of the window. The AWS panel will open.
2. Under **Developer tools** choose the gear icon next to **CodeWhisperer**.
3. If you are using VS Code workspaces, switch to the Workspace sub-tab. In VS Code, workspace settings override user settings.
4. In the **Settings** tab search for **aws:telemetry**.
5. Uncheck the box.



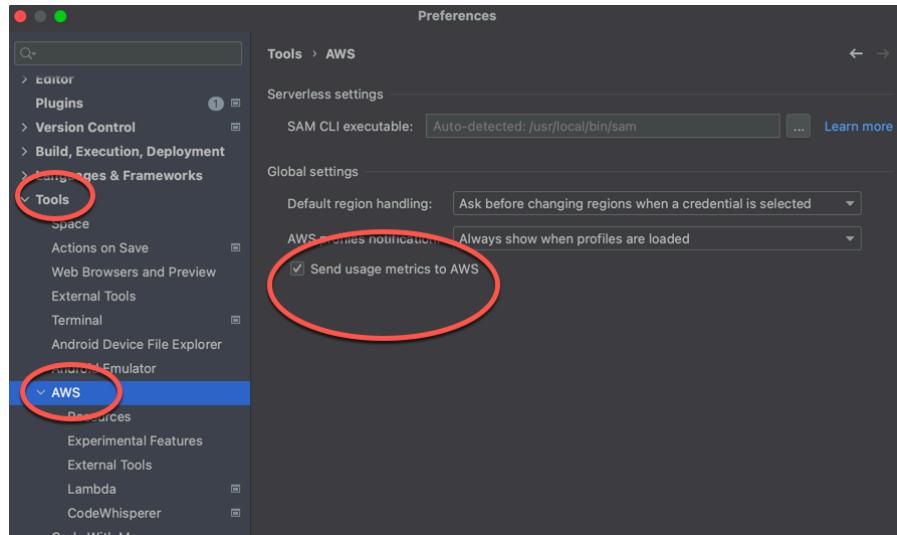
Note

This is a decision for each developer to make inside their own IDE. If you are using CodeWhisperer as part of an enterprise, your administrator will not be able to change this setting for you.

AWS Toolkit for JetBrains

1. In JetBrains, choose **AWS Toolkit** logo from the side of the window. The AWS panel will open.

2. Near the **AWS Toolkit** heading, choose the gear icon.
3. From the pop-up menu choose **Show AWS Settings**.
4. In the Preferences window, under Tools -> AWS, next to **Send usage metrics to AWS**, uncheck the box.

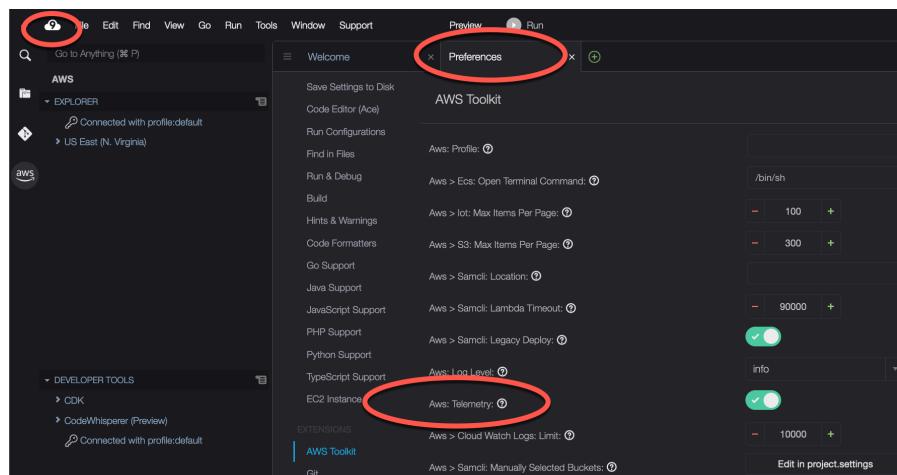


Note

This is a decision for each developer to make inside their own IDE. If you are using CodeWhisperer as part of an enterprise, your administrator will not be able to change this setting for you.

AWS Cloud 9

1. From inside your AWS Cloud 9 IDE, choose the AWS Cloud 9 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

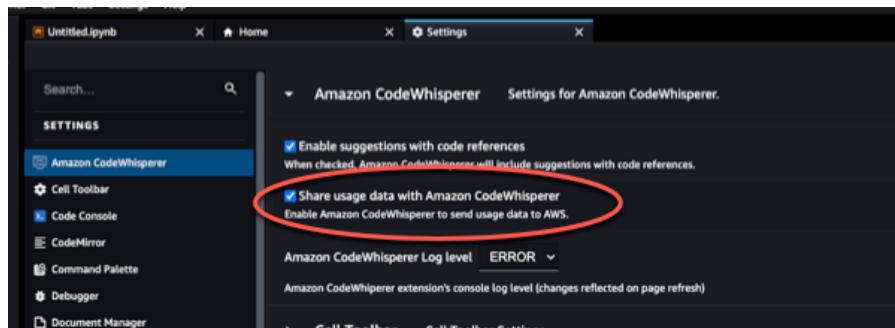
This setting affects whether or not you share your AWS Cloud 9 client-side telemetry in general, not just for CodeWhisperer.

Lambda

When you use CodeWhisperer with Lambda, CodeWhisperer does not share your client-side telemetry with AWS.

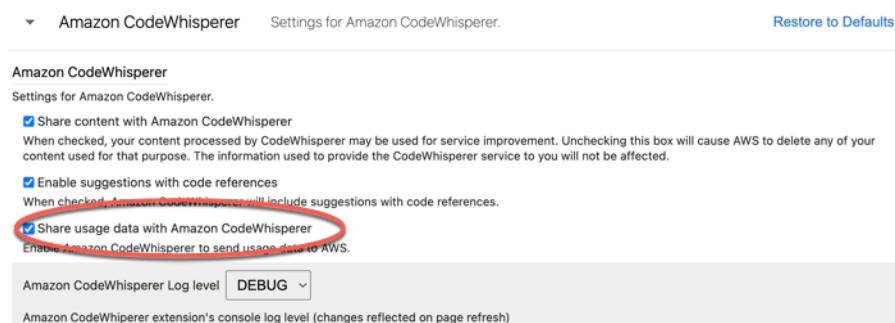
SageMaker Studio

1. From the top of the SageMaker Studio window choose **Settings**.
2. From the **Settings** dropdown, choose **Advanced Settings Editor**.
3. In the Amazon CodeWhisperer dropdown, select or deselect the box next to **Share usage data with Amazon CodeWhisperer**.



JupyterLab

1. From the top of the JupyterLab window choose **Settings**.
2. From the **Settings** dropdown, choose **Advanced Settings Editor**.
3. In the Amazon CodeWhisperer dropdown, select or deselect the box next to **Share usage data with Amazon CodeWhisperer**.

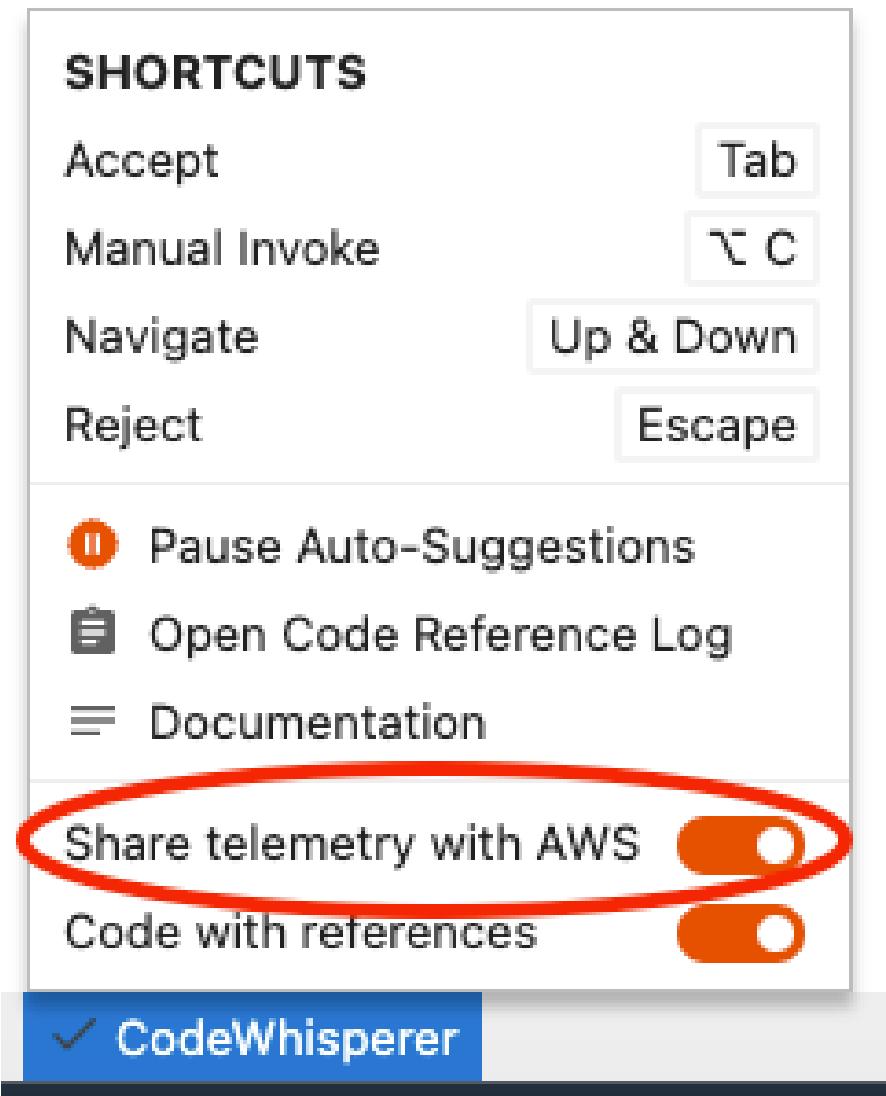


AWS Glue Studio Notebook

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



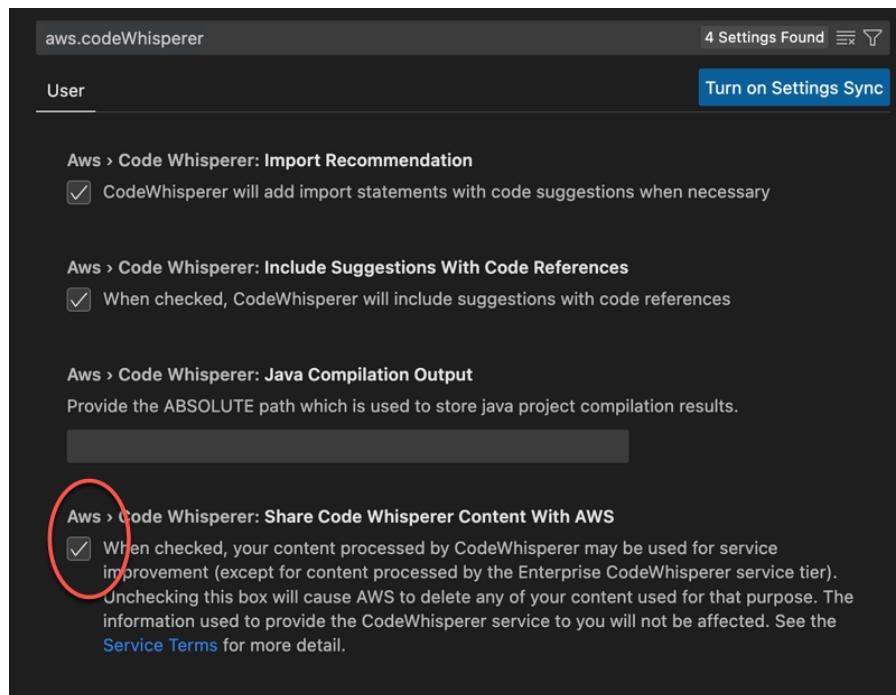
Opting out of sharing your content

AWS Toolkit for Visual Studio Code

At [the professional tier \(p. 60\)](#), CodeWhisperer does not collect your content.

At [the individual tier \(p. 61\)](#), to opt out of sharing your content in VS Code, use the following procedure.

1. In VS Code, choose the AWS logo from the side of the window. The AWS panel will open.
2. Under **Developer tools** choose the gear icon next to **CodeWhisperer**.
3. If you are using VS Code workspaces, switch to the Workspace sub-tab. In VS Code, workspace settings override user settings.
4. Uncheck the box near **Share CodeWhisperer Content With AWS**.

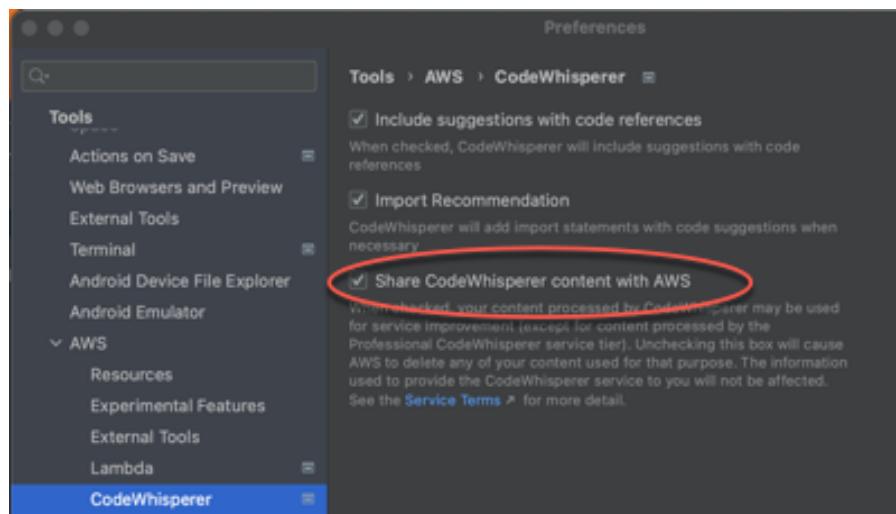


AWS Toolkit for JetBrains

At the professional tier, CodeWhisperer does not collect your content.

To opt out of sharing CodeWhisperer data in JetBrains, use the following procedure.

1. Make sure you are using the latest version of both JetBrains and the AWS Toolkit.
 2. In JetBrains, open **Preferences** (on a Mac, this will be under **Settings**).
 3. In the **Preferences** window, under **Tools**, under **AWS**, select **CodeWhisperer**.
- The CodeWhisperer preferences pane will open on the right.
4. In the CodeWhisperer preferences pane, deselect **Share CodeWhisperer content with AWS**.



AWS Cloud 9

When you use CodeWhisperer with AWS Cloud 9, CodeWhisperer does not share your content with AWS.

Lambda

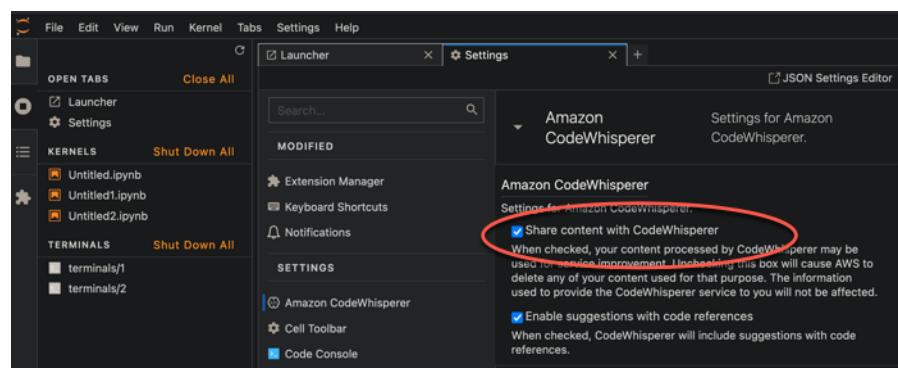
When you use CodeWhisperer with Lambda, CodeWhisperer does not share your content with AWS.

SageMaker Studio

When you use CodeWhisperer with SageMaker Studio, CodeWhisperer 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 CodeWhisperer dropdown, select or deselect the box next to **Share content with Amazon CodeWhisperer**.



AWS Glue Studio Notebook

When you use CodeWhisperer with AWS Glue Studio Notebook, CodeWhisperer does not share your content with AWS.

Quotas for Amazon CodeWhisperer

CodeWhisperer does not maintain any service quotas.

To learn about the differences in usage available per service tier, see [???](#) (p. 62).

Security in Amazon CodeWhisperer

Note

The CodeWhisperer infrastructure is located in US East (N. Virginia).

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

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

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the [AWS Compliance Programs](#). To learn about the compliance programs that apply to Amazon CodeWhisperer, 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 CodeWhisperer. The following topics show you how to configure CodeWhisperer to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your CodeWhisperer resources.

Note

Topics

- [Resilience in Amazon CodeWhisperer \(p. 86\)](#)
- [Vulnerability analysis and management in Amazon CodeWhisperer \(p. 87\)](#)
- [Best practices for administrative security with IAM Identity Center and CodeWhisperer \(p. 87\)](#)
- [Data protection \(p. 87\)](#)
- [Compliance validation for Amazon CodeWhisperer \(p. 89\)](#)
- [Security best practices in Amazon CodeWhisperer \(p. 89\)](#)
- [Infrastructure security in Amazon CodeWhisperer \(p. 89\)](#)
- [Identity and Access Management for Amazon CodeWhisperer \(p. 90\)](#)
- [Amazon CodeWhisperer and interface VPC endpoints \(AWS PrivateLink\) \(p. 108\)](#)

Resilience in Amazon CodeWhisperer

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](#).

In addition to the AWS global infrastructure, CodeWhisperer offers several features to help support your data resiliency and backup needs.

Vulnerability analysis and management in Amazon CodeWhisperer

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

Best practices for administrative security with IAM Identity Center and CodeWhisperer

This section describes some suggestions for simplifying your security decisions as you administer IAM Identity Center in connection with CodeWhisperer.

- [Activate MFA on your management console root user](#), and also on your external identity provider.
- If you are using a multi-account environment, [configure delegated administration](#).
- [Use an existing identity source](#) and enable it when you first use IAM identity center
- [Delegate administrative permissions](#) to a particular user
- [Create an administrative permission set](#).
- [Use service control policies \(SCPs\)](#) to control which applications can access the information in which AWS Organizations accounts.
- [Create a permissions boundary](#).

For more information, see [the IAM Identity Center user guide](#).

Data protection

Data Protection for Amazon CodeWhisperer

The [AWS shared responsibility model](#) applies to data protection in Amazon CodeWhisperer. 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. This content includes 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 user accounts with AWS Identity and Access Management. 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 personal data that is stored in Amazon Simple Storage Service.
- 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 variables. This includes when you work with CodeWhisperer or other AWS services using the console, API, AWS CLI, or AWS SDKs.

AWS CloudTrail and CodeWhisperer APIs

If you use CloudTrail, you may find that your CloudTrail logs contain API calls related to CodeWhisperer. These calls are:

- CreateProfile
- ListProfiles
- UpdateProfile
- DeleteProfile
- GenerateRecommendations
- GetCodeAnalysis
- ListCodeAnalysisFindings
- StartCodeAnalysis
- CreateUploadUrl
- GenerateCompletions
- CreateCustomization
- ListCustomizations
- UpdateCustomization
- GetCustomization

To learn more about how these APIs may be called from the console, and related IAM permissions, see [???](#) (p. 4).

For explanations of specific APIs, see [???](#) (p. 39).

Data encryption in Amazon CodeWhisperer

Encryption is an important part of CodeWhisperer security. Data in transit and at rest is encrypted by default as part of Amazon CodeWhisperer and doesn't require you to do anything.

- Encryption of data at rest – By default data collected by CodeWhisperer is stored using Amazon Simple Storage Service and Amazon DynamoDB. The data is encrypted using their data-at-rest encryption capabilities with a AWS-owned key.

However, [enterprise users have the option \(p. 14\)](#) of encrypting their data using an AWS KMS key.
- Encryption of data in transit – All communication between customers and CodeWhisperer, and between CodeWhisperer and its internal dependencies is protected using TLS (Transport Layer Security) to encrypt data in transit. All CodeWhisperer endpoints use SHA-256 certificates that are managed by the AWS Private Certificate Authority. For more information, see [What is AWS Private CA?](#) in the [AWS Private CA User Guide](#).

Compliance validation for Amazon CodeWhisperer

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

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

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. 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 baseline environments on AWS that are security and compliance focused.
- [Architecting for HIPAA Security and Compliance on Amazon Web Services](#) – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

Note

Not all AWS services are HIPAA eligible. For more information, see the [HIPAA Eligible Services Reference](#).

- [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* – The AWS Config service 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. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see [Security Hub controls reference](#).
- [AWS Audit Manager](#) – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Security best practices in Amazon CodeWhisperer

For information on best practices in administrative security, see [Best practices for administrative security with IAM Identity Center and CodeWhisperer \(p. 87\)](#).

For information on best practices in infrastructure security, see [Infrastructure security in Amazon CodeWhisperer \(p. 89\)](#).

Infrastructure security in Amazon CodeWhisperer

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

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

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

Identity and Access Management for Amazon CodeWhisperer

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 CodeWhisperer resources. IAM is an AWS service that you can use with no additional charge.

Topics

- [Audience \(p. 90\)](#)
- [Authenticating with identities \(p. 90\)](#)
- [Managing access using policies \(p. 93\)](#)
- [How Amazon CodeWhisperer works with IAM \(p. 94\)](#)
- [Identity-based policy examples for Amazon CodeWhisperer \(p. 99\)](#)
- [AWS managed policies for Amazon CodeWhisperer \(p. 101\)](#)
- [Troubleshooting Amazon CodeWhisperer identity and access \(p. 104\)](#)
- [Using service-linked roles for CodeWhisperer \(p. 105\)](#)

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in CodeWhisperer.

Service user – If you use the CodeWhisperer service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more CodeWhisperer 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. If you cannot access a feature in CodeWhisperer, see [Troubleshooting Amazon CodeWhisperer identity and access \(p. 104\)](#).

Service administrator – If you're in charge of CodeWhisperer resources at your company, you probably have full access to CodeWhisperer. It's your job to determine which CodeWhisperer 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 CodeWhisperer, see [How Amazon CodeWhisperer works with IAM \(p. 94\)](#).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to CodeWhisperer. To view example CodeWhisperer identity-based policies that you can use in IAM, see [Identity-based policy examples for Amazon CodeWhisperer \(p. 99\)](#).

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, as 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 (successor to AWS Single Sign-On) (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](#).

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see [Signing AWS API requests](#) in the [IAM User Guide](#).

Regardless of the authentication method that you use, you might 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 \(successor to AWS Single Sign-On\) User Guide](#) and [Using multi-factor authentication \(MFA\) in AWS](#) in the [IAM User Guide](#).

AWS account root user

When you create an AWS account, you begin with one 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 the 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 [AWS Account Management Reference 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 (successor to AWS Single Sign-On). 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 \(successor to AWS Single Sign-On\) 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. To learn more, 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. It 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 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 (successor to AWS Single Sign-On) User Guide*.
- **Temporary IAM user permissions** – An IAM user or role 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). To learn 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. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. 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. To see whether an action requires additional dependent actions in a policy, see [Actions, resources, and condition keys for Amazon CodeWhisperer](#) in the *Service Authorization Reference*.
- **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 EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an 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

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*.

To learn whether to use IAM roles or IAM users, 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**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

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, 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*.

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. To learn 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. 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.

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

Access control lists (ACLs)

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

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

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 service control policies (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 CodeWhisperer works with IAM

Before you use IAM to manage access to CodeWhisperer, learn what IAM features are available to use with CodeWhisperer.

IAM features you can use with Amazon CodeWhisperer

IAM feature	CodeWhisperer support
Identity-based policies (p. 95)	Yes
Resource-based policies (p. 95)	Yes
Policy actions (p. 96)	Partial
Policy resources (p. 96)	No
Policy condition keys (service-specific) (p. 97)	No
ACLs (p. 97)	No

IAM feature	CodeWhisperer support
ABAC (tags in policies) (p. 97)	Yes
Temporary credentials (p. 98)	Yes
Principal permissions (p. 98)	Yes
Service roles (p. 98)	Yes
Service-linked roles (p. 99)	Yes

To get a high-level view of how CodeWhisperer 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 CodeWhisperer

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 CodeWhisperer

To view examples of CodeWhisperer identity-based policies, see [Identity-based policy examples for Amazon CodeWhisperer \(p. 99\)](#).

Resource-based policies within CodeWhisperer

Supports resource-based policies	Yes
----------------------------------	-----

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 CodeWhisperer

Supports policy actions	Partial
-------------------------	---------

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 CodeWhisperer actions, see [Actions defined by Amazon CodeWhisperer](#) in the *Service Authorization Reference*.

Policy actions in CodeWhisperer use the following prefix before the action:

```
codewhisperer
```

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

```
"Action": [  
    "codewhisperer:action1",  
    "codewhisperer:action2"  
]
```

To view examples of CodeWhisperer identity-based policies, see [Identity-based policy examples for Amazon CodeWhisperer \(p. 99\)](#).

Policy resources for CodeWhisperer

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 see a list of CodeWhisperer resource types and their ARNs, see [Resources defined by Amazon CodeWhisperer](#) in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see [Actions defined by Amazon CodeWhisperer](#).

To view examples of CodeWhisperer identity-based policies, see [Identity-based policy examples for Amazon CodeWhisperer \(p. 99\)](#).

Policy condition keys for CodeWhisperer

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 see a list of CodeWhisperer condition keys, see [Condition keys for Amazon CodeWhisperer](#) in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see [Actions defined by Amazon CodeWhisperer](#).

To view examples of CodeWhisperer identity-based policies, see [Identity-based policy examples for Amazon CodeWhisperer \(p. 99\)](#).

ACLs in CodeWhisperer

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 CodeWhisperer

Supports ABAC (tags in policies)	Yes
----------------------------------	-----

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 CodeWhisperer

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 CodeWhisperer

Supports principal permissions	Yes
--------------------------------	-----

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. To see whether an action requires additional dependent actions in a policy, see [Actions, resources, and condition keys for Amazon CodeWhisperer](#) in the *Service Authorization Reference*.

Service roles for CodeWhisperer

Supports service roles	Yes
------------------------	-----

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 CodeWhisperer functionality. Edit service roles only when CodeWhisperer provides guidance to do so.

Service-linked roles for CodeWhisperer

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 service-linked roles, see [AWS services that work with IAM](#). Find a service in the table that includes a Yes in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for Amazon CodeWhisperer

By default, users and roles don't have permission to create or modify CodeWhisperer resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see [Creating IAM policies](#) in the *IAM User Guide*.

For details about actions and resource types defined by CodeWhisperer, including the format of the ARNs for each of the resource types, see [Actions, resources, and condition keys for Amazon CodeWhisperer](#) in the *Service Authorization Reference*.

Topics

- [Policy best practices \(p. 99\)](#)
- [Using the CodeWhisperer console \(p. 100\)](#)
- [Permissions required for the CodeWhisperer console \(p. 100\)](#)
- [Allow users to view their own permissions \(p. 100\)](#)

Policy best practices

Identity-based policies determine whether someone can create, access, or delete CodeWhisperer 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*.

Using the CodeWhisperer console

To access the Amazon CodeWhisperer console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the CodeWhisperer resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

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

Permissions required for the CodeWhisperer console

The CodeWhisperer console uses the following API actions.

- codewhisperer:CreateProfile
- codewhisperer>ListProfiles
- codewhisperer:UpdateProfile
- codewhisperer:DeleteProfile

The CreateProfile, ListProfiles, UpdateProfile, and DeleteProfile API actions are not intended to be called by your code. Therefore, these API actions are not included in the AWS CLI and AWS SDKs.

Allow users to view their own permissions

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

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ViewOwnUserInfo",  
            "Effect": "Allow",  
            "Action": "iam:GetUser",  
            "Resource": "arn:aws:iam::  
                ${AWS::AccountId}:user/  
                ${AWS::Username}"  
        }  
    ]  
}
```

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

AWS managed policies for Amazon CodeWhisperer

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

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

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

For more information, see [AWS managed policies](#) in the *IAM User Guide*.

AWS managed policy: AWSServiceRoleForCodeWhispererPolicy

This AWS managed policy grants permissions commonly needed to use Amazon CodeWhisperer. The policy is added to the AWSServiceRoleForCodeWhisperer that is created when you onboard to CodeWhisperer.

You can't attach AWSServiceRoleForCodeWhispererPolicy to your IAM entities. This policy is attached to a [service-linked role \(p. 105\)](#) that allows CodeWhisperer to perform actions on your behalf. For more information, see [Using service-linked roles for CodeWhisperer \(p. 105\)](#).

This policy grants **administrator** permissions that allow code artifacts to be scanned for security purposes, and that allow usage metrics to be collected in order to track billing.

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 CodeWhisperer in CloudWatch.
- **codeguru-security** – Allows principals to upload code artifacts, perform codescans and list codescan findings with Amazon CodeGuru. This is required so that CodeWhisperer can conduct a security scan on your code from within the JetBrains and Visual Studio Code IDEs.
- **sso** – Allows principals to retrieve all details of the CodeWhisperer application as represented in IAM Identity Center. This is required to enable billing of CodeWhisperer usage.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "sso-directory>ListMembersInGroup"  
            ],  
            "Resource": [  
                "*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "sso>ListProfileAssociations",  
                "sso>ListProfiles",  
                "sso>ListDirectoryAssociations",  
                "sso>DescribeRegisteredRegions",  
                "sso:GetProfile",  
                "sso:GetManagedApplicationInstance"  
            ],  
            "Resource": [  
                "*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "codeguru-security>CreateUploadUrl"  
            ],  
            "Resource": [  
                "*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "codeguru-security>CreateScan",  
                "codeguru-security>GetScan",  
                "codeguru-security>ListFindings",  
                "codeguru-security>GetFindings"  
            ],  
            "Resource": [  
                "*"  
            ]  
        }  
    ]  
}
```

```

        "arn:aws:codeguru-security:*.*:scans/CodeWhisperer-*"
    ],
},
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:PutMetricData"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "cloudwatch:namespace": [
                "AWS/CodeWhisperer"
            ]
        }
    }
}
]
}

```

To view this policy in the context of other AWS managed policies, see [AWSServiceRoleForCodeWhispererPolicy](#).

CodeWhisperer updates to AWS managed policies

View details about updates to AWS managed policies for CodeWhisperer since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on [the CodeWhisperer Document history page \(p. 111\)](#).

Change	Description	Date
Updated AWSServiceRoleForCodeWhispererPolicy	Added GetFindings and GetManagedApplicationInstance permissions. GetFindings permissions simplify cross-service interactions, but do not impact your experience with the service. GetManagedApplicationInstance prevents you from being billed for disabled CodeWhisperer application instances.	June 19, 2023
Updated AWSServiceRoleForCodeWhispererPolicy	Added permissions for getting User and group information for billing purposes.	May 31, 2023
AWSServiceRoleForCodeWhispererPolicy – New policy	<p>Added (p. 101) policy to allow CodeWhisperer to call CloudWatch and CodeGuru on your behalf.</p> <p>This policy is added to the AWSServiceRoleForCodeWhisperer that is created when you onboard to Amazon CodeWhisperer.</p>	March 29, 2023

Change	Description	Date
CodeWhisperer started tracking changes	CodeWhisperer started tracking changes for its AWS managed policies.	March 29, 2023

Troubleshooting Amazon CodeWhisperer identity and access

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

Topics

- [I am not authorized to perform an action in CodeWhisperer \(p. 104\)](#)
- [I am not authorized to perform iam:PassRole \(p. 104\)](#)
- [I want to allow people outside of my AWS account to access my CodeWhisperer resources \(p. 105\)](#)

I am not authorized to perform an action in CodeWhisperer

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the mateojackson IAM user tries to use the console to view details about a fictional *my-example-widget* resource but doesn't have the fictional codewhisperer:*GetWidget* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: codewhisperer:GetWidget on resource: my-example-widget
```

In this case, the policy for the mateojackson user must be updated to allow access to the *my-example-widget* resource by using the codewhisperer:*GetWidget* action.

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

I am not authorized to perform iam:PassRole

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

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

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

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

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

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

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

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

To learn more, consult the following:

- To learn whether CodeWhisperer supports these features, see [How Amazon CodeWhisperer works with IAM \(p. 94\)](#).
- To learn how to provide access to your resources across AWS accounts that you own, see [Providing access to an IAM user in another AWS account that you own](#) in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see [Providing access to AWS accounts owned by third parties](#) in the *IAM User Guide*.
- To learn how to provide access through identity federation, see [Providing access to externally authenticated users \(identity federation\)](#) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.

Using service-linked roles for CodeWhisperer

Amazon CodeWhisperer 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 CodeWhisperer. Service-linked roles are predefined by CodeWhisperer and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up CodeWhisperer easier because you don't have to manually add the necessary permissions. CodeWhisperer defines the permissions of its service-linked roles, and unless defined otherwise, only CodeWhisperer 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 CodeWhisperer 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 CodeWhisperer \(p. 101\)](#).

Service-linked role permissions for CodeWhisperer

CodeWhisperer uses the service-linked role named **AWSServiceRoleForCodeWhisperer** – This role grants permissions to CodeWhisperer 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 AWSServiceRoleForCodeWhisperer service-linked role trusts the following services to assume the role:

- codewhisperer.amazonaws.com

The role permissions policy named AWSServiceRoleForCodeWhispererPolicy allows CodeWhisperer to complete the following actions on the specified resources:

- Action: cloudwatch:PutMetricData on AWS/CodeWhisperer CloudWatch namespace
- Action: codeguru-security>CreateUploadUrl on *
- Action: codeguru-security>CreateScan on arn:aws:codeguru-security:*:scans/CodeWhisperer-*
- Action: codeguru-security>GetScan on arn:aws:codeguru-security:*:scans/CodeWhisperer-*
- Action: codeguru-security>ListFindings on arn:aws:codeguru-security:*:scans/CodeWhisperer-*
- Action: sso>ListProfiles on *
- Action: sso>ListProfileAssociations on *
- Action: sso-directory>ListMembersInGroup on *
- Action: sso>ListDirectoryAssociations on *
- Action: sso>DescribeRegisteredRegions on *
- Action: sso>GetProfile 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 CodeWhisperer

You don't need to manually create a service-linked role. When you Set up CodeWhisperer in the AWS Management Console, CodeWhisperer 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, CodeWhisperer 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 codewhisperer.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 CodeWhisperer

CodeWhisperer does not allow you to edit the AWSServiceRoleForCodeWhisperer 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 CodeWhisperer

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 CodeWhisperer 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 AWSServiceRoleForCodeWhisperer service-linked role. For more information, see [Deleting a service-linked role](#) in the *IAM User Guide*.

Supported Regions for CodeWhisperer service-linked roles

CodeWhisperer supports using service-linked roles in all of the Regions where the service is available. For more information, see [AWS Regions and endpoints](#).

CodeWhisperer does not support using service-linked roles in every Region where the service is available. You can use the AWSServiceRoleForCodeWhisperer role in the following Regions.

Region name	Region identity	Support in CodeWhisperer
US East (N. Virginia)	us-east-1	Yes
US East (Ohio)	us-east-2	No
US West (N. California)	us-west-1	No
US West (Oregon)	us-west-2	No
Africa (Cape Town)	af-south-1	No
Asia Pacific (Hong Kong)	ap-east-1	No
Asia Pacific (Jakarta)	ap-southeast-3	No
Asia Pacific (Mumbai)	ap-south-1	No
Asia Pacific (Osaka)	ap-northeast-3	No
Asia Pacific (Seoul)	ap-northeast-2	No
Asia Pacific (Singapore)	ap-southeast-1	No
Asia Pacific (Sydney)	ap-southeast-2	No
Asia Pacific (Tokyo)	ap-northeast-1	No
Canada (Central)	ca-central-1	No
Europe (Frankfurt)	eu-central-1	No
Europe (Ireland)	eu-west-1	No
Europe (London)	eu-west-2	No
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

Region name	Region identity	Support in CodeWhisperer
AWS GovCloud (US-West)	us-gov-west-1	No

Amazon CodeWhisperer and interface VPC endpoints (AWS PrivateLink)

You can establish a private connection between your VPC and Amazon CodeWhisperer by creating an *interface VPC endpoint*. Interface endpoints are powered by [AWS PrivateLink](#), a technology that enables you to privately access CodeWhisperer 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 CodeWhisperer APIs. Traffic between your VPC and CodeWhisperer 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*.

Note

CodeWhisperer does not support endpoint policies.

Considerations for CodeWhisperer VPC endpoints

Before you set up an interface VPC endpoint for CodeWhisperer, ensure that you review [Interface endpoint properties and limitations](#) in the *Amazon VPC User Guide*.

CodeWhisperer supports making calls to all of its API actions from your VPC, in the context of services that are configured to work with CodeWhisperer.

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 CodeWhisperer.

Creating an interface VPC endpoint for CodeWhisperer

You can create a VPC endpoint for the CodeWhisperer 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 CodeWhisperer using the following service name:

- com.amazonaws.*region*.codewhisperer

If you enable private DNS for the endpoint, you can make API requests to CodeWhisperer 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*.

Using an on-premises computer to connect to a CodeWhisperer endpoint

This section describes the process of using an on-premises computer to connect to CodeWhisperer 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 CodeWhisperer. \(p. 108\)](#)
3. [Set up an inbound Amazon Route 53 endpoint.](#) This will enable you to use the DNS name of your CodeWhisperer endpoint from your on-premises device.

Using an in-console IDE to connect to a CodeWhisperer endpoint

This section describes the process of using an in-console IDE to connect to a CodeWhisperer 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 CodeWhisperer. \(p. 108\)](#)
2. Set up CodeWhisperer with the in-console IDE.
 - [AWS Cloud9 \(p. 32\)](#)
 - [SageMaker Studio \(p. 28\)](#)
 - [AWS Glue Studio \(p. 31\)](#)
3. Configure the IDE to use the CodeWhisperer endpoint.
 - [AWS Cloud9](#)
 - [SageMaker Studio](#)
 - [AWS Glue Studio](#)

Connecting to CodeWhisperer 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 CodeWhisperer using AWS PrivateLink.

1. [Create an interface VPC endpoint for CodeWhisperer. \(p. 108\)](#)
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 AWS Toolkit, using one of the following procedures:
 - [Installing the AWS Toolkit for JetBrains.](#)
 - [Installing the AWS Toolkit for Visual Studio Code.](#)
6. Configure the IDE to connect via AWS PrivateLink.
 - [Network connections in Visual Studio Code](#)
 - [JetBrains remote development](#)

Document history for the CodeWhisperer User Guide

The following table describes the documentation releases for CodeWhisperer.

Change	Description	Date
Dashboard (p. 111)	A dashboard is available for administrators at the professional tier.	September 8, 2023
AWS PrivateLink integration (p. 111)	You can establish a private connection between your VPC and CodeWhisperer by creating an interface VPC endpoint .	July 26, 2023
AWS Glue integration (p. 111)	You can use CodeWhisperer with AWS Glue Studio Notebook .	July 26, 2023
Updated managed policy (p. 111)	Added GetFindings and GetManagedApplicationInstance.	June 28, 2023
Added permissions to policy for administrators (p. 111)	Added related new permissions iamadmin>ListRolesByPrincipal and pricing:GetProducts, needed for CodeWhisperer administration.	June 27, 2023
Non-management accounts can now manage CodeWhisperer (p. 111)	AWS recommends that CodeWhisperer and IAM Identity Center be administered through a non-management account . Also, added related new permission sso>ListApplicationInstances, needed for CodeWhisperer administration.	June 12, 2023
Added API calls to be tracked with AWS CloudTrail (p. 111)	The following APIs can now be tracked in CloudTrail: DeleteProfile, GetCodeAnalysis, ListCodeAnalysisFindings, StartCodeAnalysis, CreateUploadUrl, GenerateCompletions.	June 6, 2023
Change to service-linked role (p. 111)	Added permissions for getting user and group information for billing purposes by updating AWSServiceRoleForCodeWhispererPolicy, which is associated with the service-linked role .	May 30, 2023

<u>Added two Amazon CloudWatch metrics (p. 111)</u>	Added Subscriptions and MonthlyActiveUniqueUsers as CloudWatch metrics.	May 30, 2023
<u>Amazon SageMaker Studio and JupyterLab support (p. 111)</u>	Added sections explaining setup for integration with SageMaker Studio and JupyterLab .	May 9, 2023
<u>New service-linked role (p. 111)</u>	Added sso>ListDirectoryAssociations as a service-linked role .	May 1, 2023
<u>Monitoring chapter (p. 111)</u>	Added information about monitoring CodeWhisperer with CloudWatch.	April 24, 2023
<u>Security scans of multiple files (p. 111)</u>	The security scan section has been updated to clarify that a scan can include more than one file.	April 20, 2023
<u>Region-based restrictions (p. 111)</u>	The CodeWhisperer administrator section has been updated to clarify the specific situation in which actions must be taken in a particular region.	April 19, 2023
<u>Useful APIs (p. 111)</u>	The Useful APIs section has been added to User actions. Although CodeWhisperer does not have a public API, these API calls may be useful in the context of creating or editing IAM policies.	April 13, 2023
<u>Types of users (p. 111)</u>	The Types of users chapter has been added to help clarify the different personas who use CodeWhisperer in different ways.	April 13, 2023
<u>Setting up (p. 111)</u>	Some content moved from Getting started to Setting up. Setup instructions for administrators broken out into three parts: Root user, AWS account admin, and CodeWhisperer admin.	April 13, 2023
<u>Service-linked roles (p. 111)</u>	Added information about service-linked roles.	April 13, 2023

<u>Security content (p. 111)</u>	New content has been added to clarify security issues related to CodeWhisperer. This update includes the addition of the Identity and Access Management section, as well as Infrastructure security, Compliance validation, Security best practices, Infrastructure security, and updates to the Data protection section.	April 13, 2023
<u>Quotas (p. 111)</u>	The Quotas chapter has been updated to clarify that CodeWhisperer does not maintain any quotas.	April 13, 2023
<u>Language support (p. 111)</u>	The Language support section has been updated to include languages that are newly supported as of CodeWhisperer general availability.	April 13, 2023
<u>IDE screenshots (p. 111)</u>	Various screenshots have been updated to reflect the appearance of the CodeWhisperer interface inside the AWS Toolkit once CodeWhisperer was no longer in preview.	April 13, 2023
<u>Features (p. 111)</u>	The following sections have been folded into the Features chapter: User actions, Language support, Pausing suggestions, Security scans, and Code references.	April 13, 2023
<u>Data opt-out screenshots (p. 111)</u>	The screenshots in the "opting out" section of the data sharing chapter have been updated to reflect the current CodeWhisperer settings in both VS Code and JetBrains.	April 13, 2023
<u>Billing (p. 111)</u>	The Billing chapter has been added to provide information about how you may be charged for using CodeWhisperer.	April 13, 2023
<u>Initial release (p. 111)</u>	Initial release of the CodeWhisperer User Guide. Some materials included here were previously available in other guides.	February 21, 2023

New policy:

AWSServiceRoleForAmazonCodeWhisperer
(p. 111)

Added a new policy to
allow CodeWhisperer to call
CloudWatch and CodeGuru on
your behalf.

February 17, 2023