User Guide

CodeWhisperer

CodeWhisperer: User Guide

Table of Contents

CodeWhisperer's features are becoming a part of Amazon Q Developer. Learn more

CodeWhisperer is becoming a part of Amazon Q Developer

The purpose of this section is to explain the relationship between CodeWhisperer and Amazon Q Developer.

Amazon Q Developer is a generative artificial intelligence (AI) powered conversational assistant that can help you understand, build, extend, and operate AWS applications. To learn more, see The Amazon Q Developer User Guide.

All of the features of CodeWhisperer are moving to Amazon Q Developer.

Administrators of CodeWhisperer Professional can activate the feature development and code transformation functionality of Amazon Q Developer with a toggle in the CodeWhisperer console.

- With Amazon Q Developer, you can authenticate with a personal identity through IAM Identity Center when you chat with Amazon Q in the AWS console, or on the documentation or marketing pages.
- Amazon Q Developer is part of an integrated family of services that includes Amazon Q Business and Amazon QuickSight.

How to switch to Amazon Q Developer

To switch from CodeWhisperer Professional to Amazon Q Developer Pro, use the following procedure.

1. Delete any existing customizations.
2. Delete your CodeWhisperer profile.
3. Subscribe to Amazon Q Developer Pro.

Trying Amazon Q features inside CodeWhisperer

You can try out some features of Amazon Q in the IDE through CodeWhisperer.

To do this, select **Enable Amazon Q Developer features** in your CodeWhisperer settings.

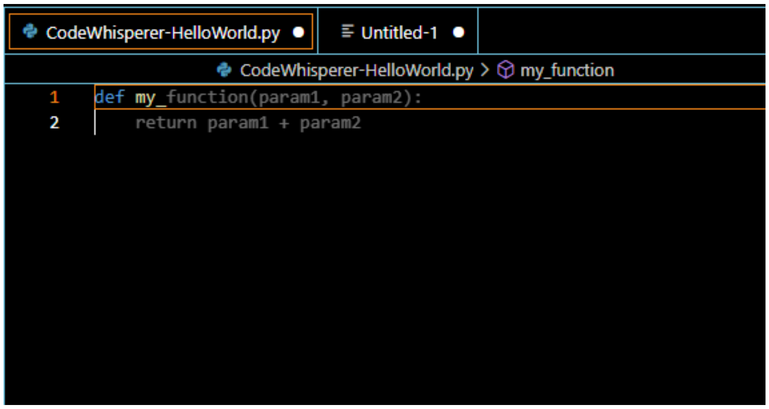
What is CodeWhisperer?

> **Note**
>
> The fastest way to start using CodeWhisperer is to authenticate with AWS Builder ID as an individual developer. 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.

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

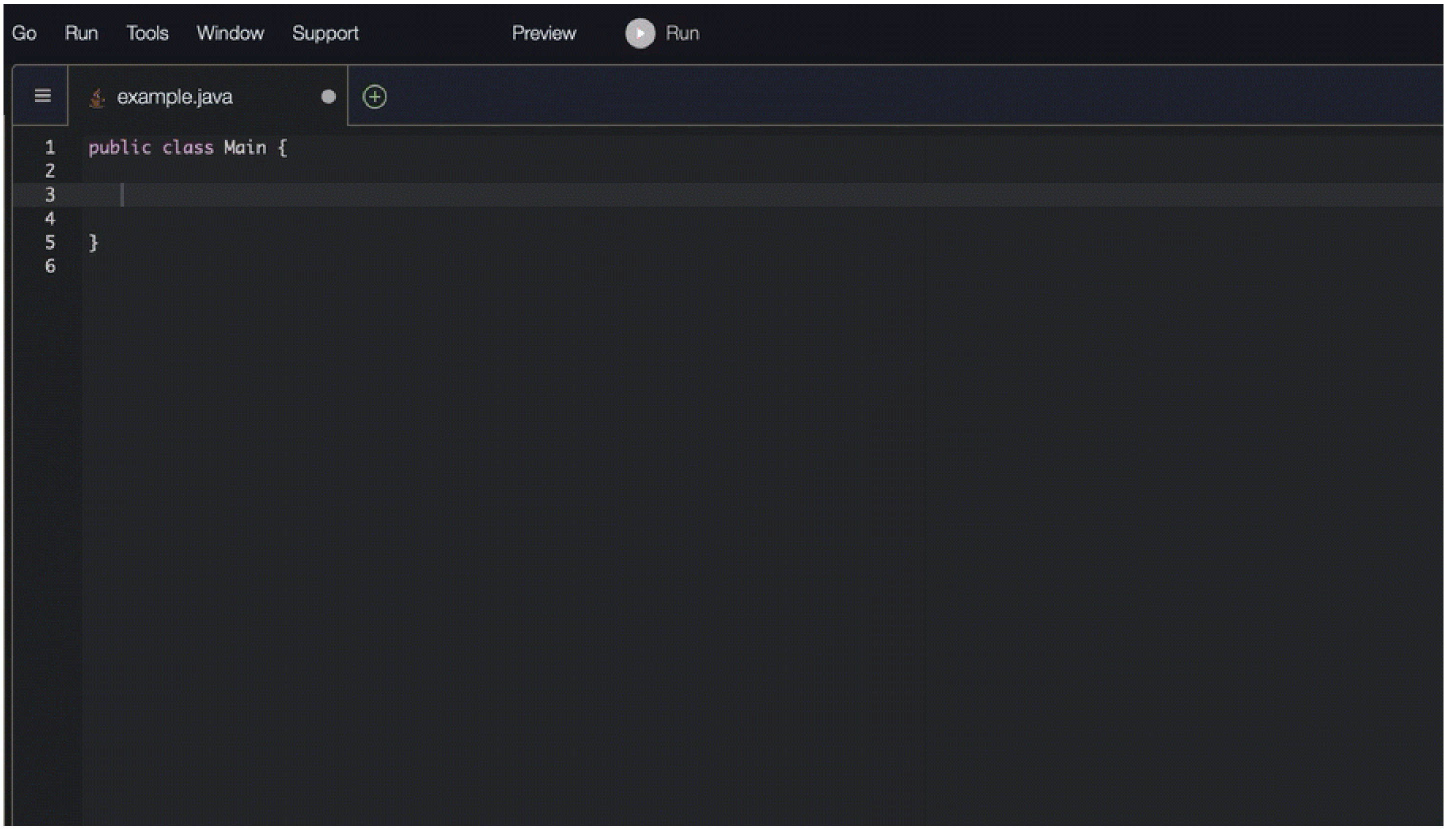
In the image below, a user has started to type out a line of code. Based on the input, CodeWhisperer has generated suggestions to complete the line. The user can cycle through the suggestions using the arrow keys.


CodeWhisperer-HelloWorld.py
Untitled-1
CodeWhisperer-HelloWorld.py > my_function

```
def my_function(param1, param2):
    return param1 + param2
```

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

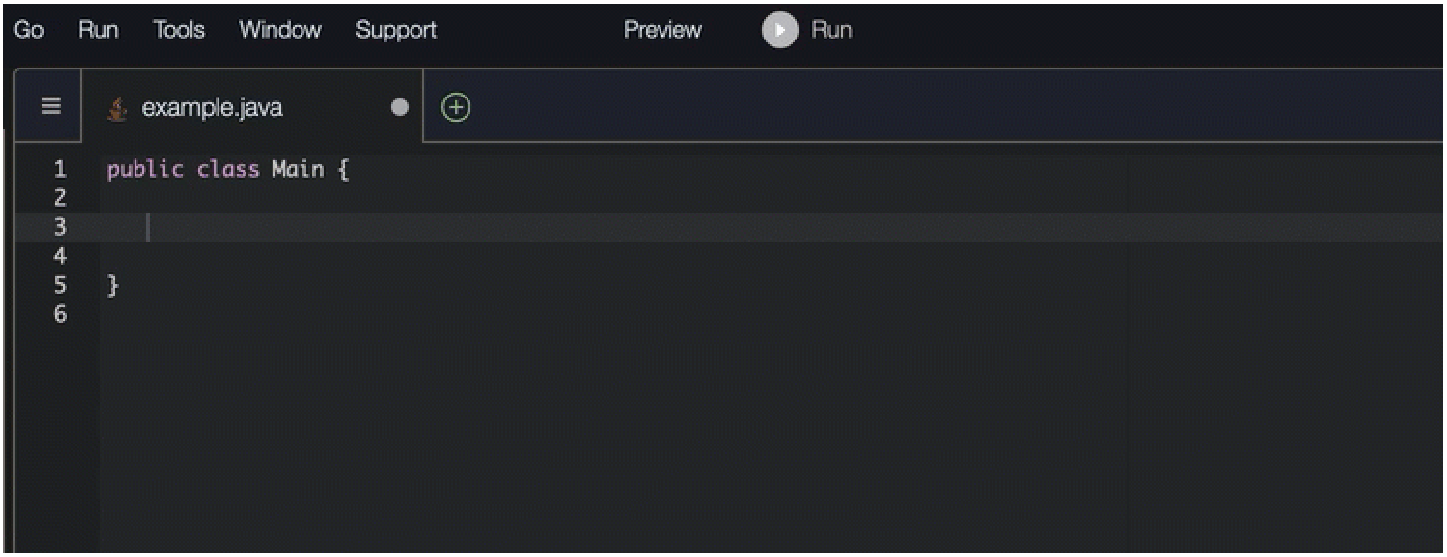
After the user accepts that suggestion, CodeWhisperer suggests a function body.



```
public class Main {



}
```

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.

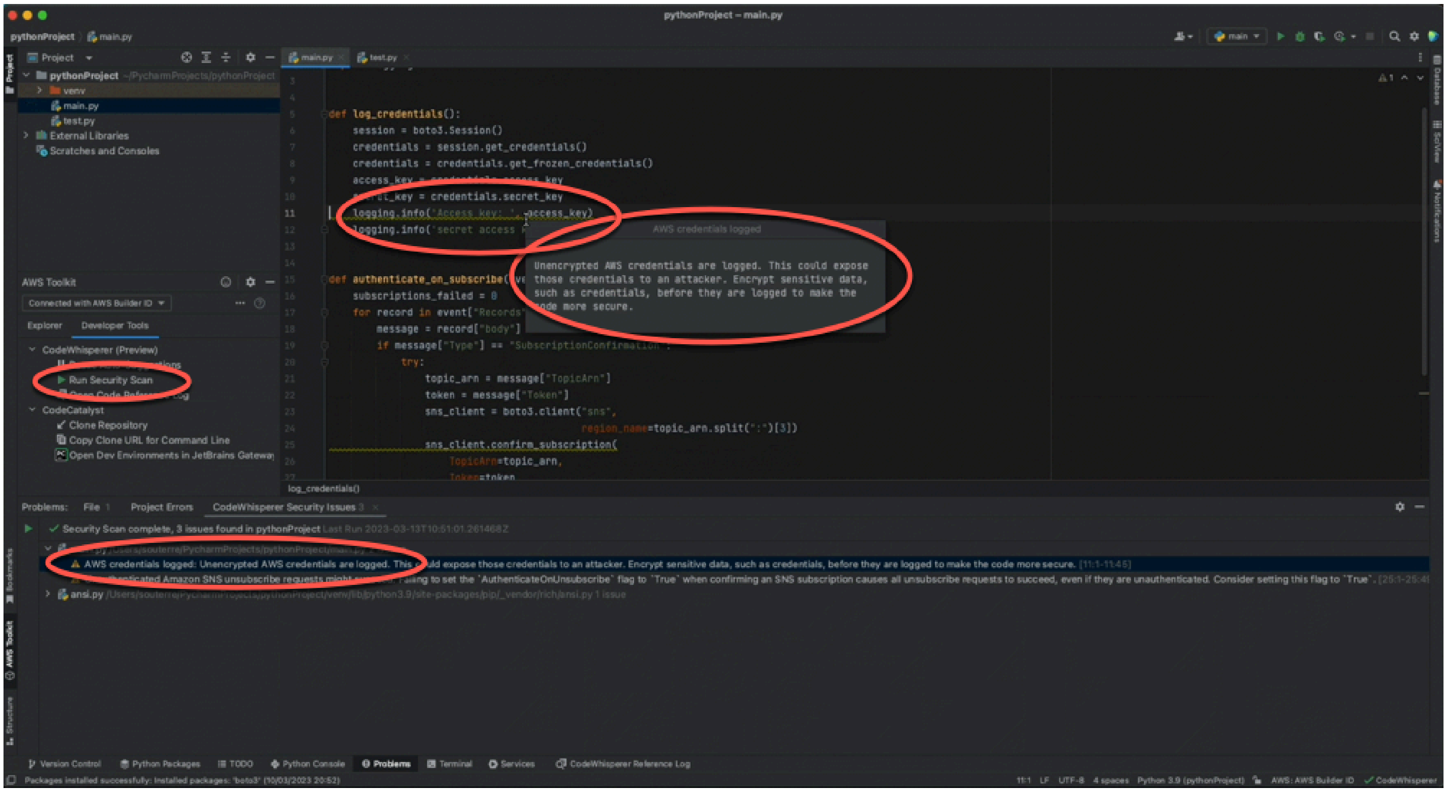


```
public class Main {



}
```

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

In this example using Python and JetBrains, the user has written code that would write unencrypted AWS credentials to a log; a bad security practice.

Fortunately, the user has also run a security scan. CodeWhisperer has identified the problem, and raised an alert.



```
def log_credentials():
    session = boto3.Session()
    credentials = session.get_credentials()
    credentials = credentials.get_frozen_credentials()
    access_key = credentials.access_key
    secret_key = credentials.secret_key
    logging.info('Access key: ', access_key)
    logging.info('secret access k

def authenticate_on_subscribe(ev
    subscriptions_failed = 0
    for record in event["Records"
        message = record["body"]
        if message["Type"] == "SubscriptionConfirmation":
            try:
                topic_arn = message["TopicArn"]
                token = message["Token"]
                sns_client = boto3.client("sns",
                                          region_name=topic_arn.split(":")[3])
                sns_client.confirm_subscription(
                    TopicArn=topic_arn,
                    Token=token
```

AWS credentials logged

Unencrypted AWS credentials are logged. This could expose those credentials to an attacker. Encrypt sensitive data, such as credentials, before they are logged to make the code more secure.

Problems: File 1 Project Errors CodeWhisperer Security Issues 3

Security Scan complete, 3 issues found in pythonProject Last Run 2023-03-13T10:51:01.261468Z

AWS credentials logged: Unencrypted AWS credentials are logged. This could expose those credentials to an attacker. Encrypt sensitive data, such as credentials, before they are logged to make the code more secure. [11:1-11:45]

Unauthenticated Amazon SNS unsubscribe requests might succeed. Failing to set the `AuthenticateOnUnsubscribe` flag to `True` when confirming an SNS subscription causes all unsubscribe requests to succeed, even if they are unauthenticated. Consider setting this flag to `True`. [25:1-25:4

ansi.py /Users/souterre/PycharmProjects/pythonProject/venv/lib/python3.9/site-packages/pip/_vendor/rich/ansi.py 1 issue

Packages installed successfully: Installed packages: 'boto3' (10/03/2023 20:52)

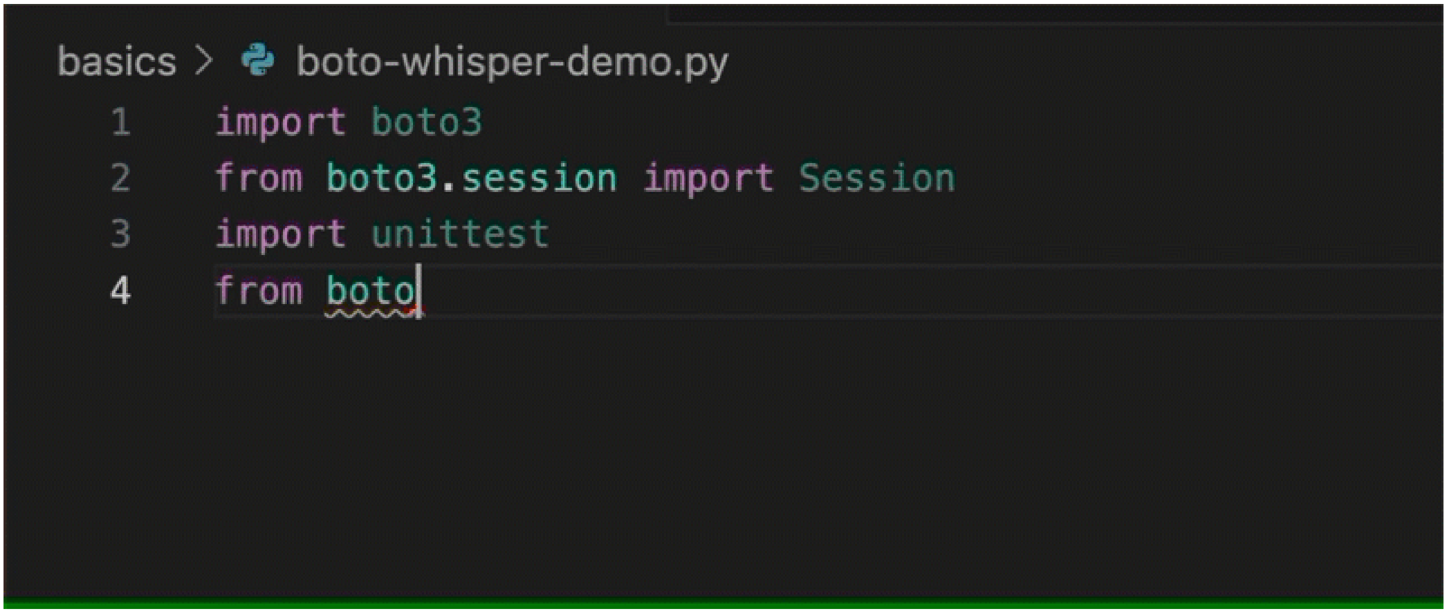
11:1 LF UTF-8 4 spaces Python 3.9 (pythonProject) AWS: AWS Builder ID CodeWhisperer

For information about which programming languages CodeWhisperer supports, see Language support.

CodeWhisperer in action

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

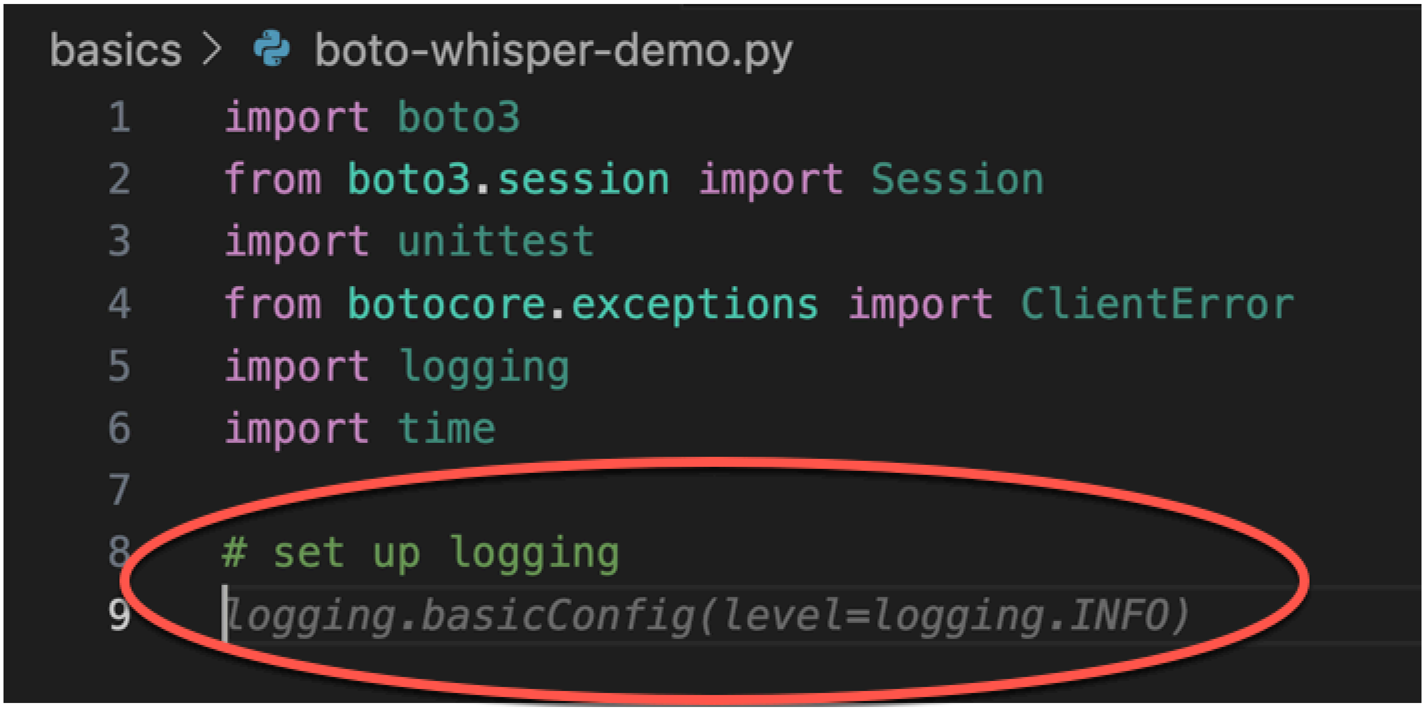
Here, CodeWhisperer helps the developer choose which libraries to import. Using the arrow keys, the developer toggles through multiple suggestions.



```python
import boto3
from boto3.session import Session
import unittest
from boto
```

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

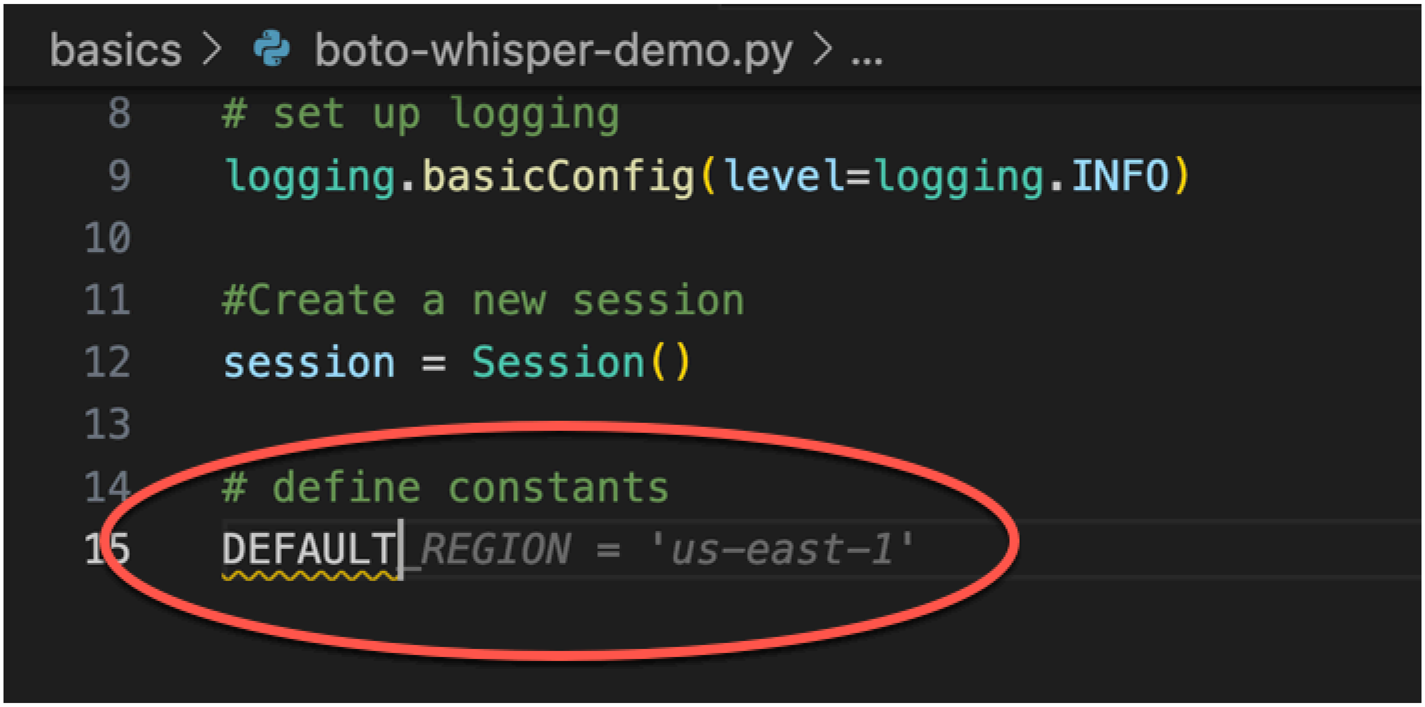
CodeWhisperer correctly anticipates the method to be called. The developer can accept the suggestion with the tab key.



```python
import boto3
from boto3.session import Session
import unittest
from botocore.exceptions import ClientError
import logging
import time

# set up logging
logging.basicConfig(level=logging.INFO)
```

Here, the developer prepares to define constants.

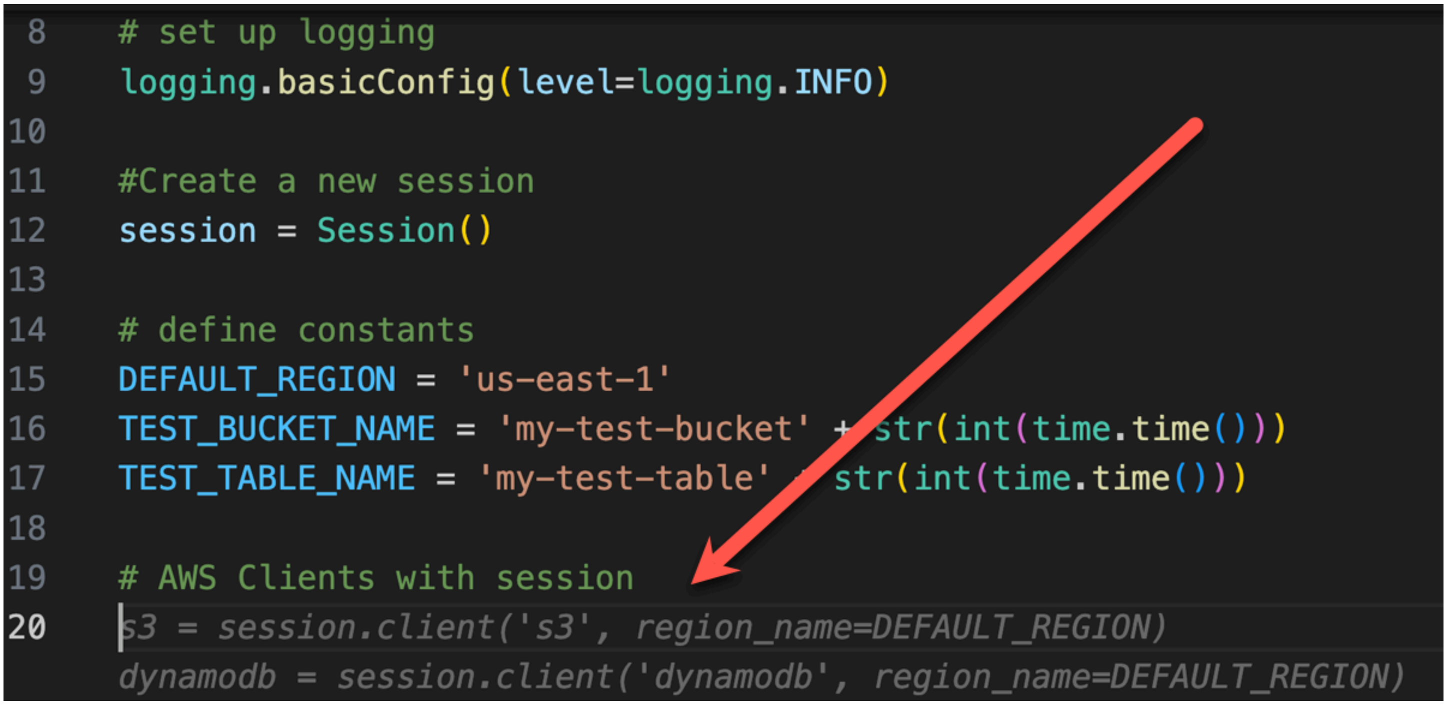
CodeWhisperer correctly anticipates that the first constant will be `REGION` and that its value will be `us-east-1`, which is the default.



```
# set up logging
logging.basicConfig(level=logging.INFO)

#Create a new session
session = Session()

# define constants
DEFAULT_REGION = 'us-east-1'
```

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

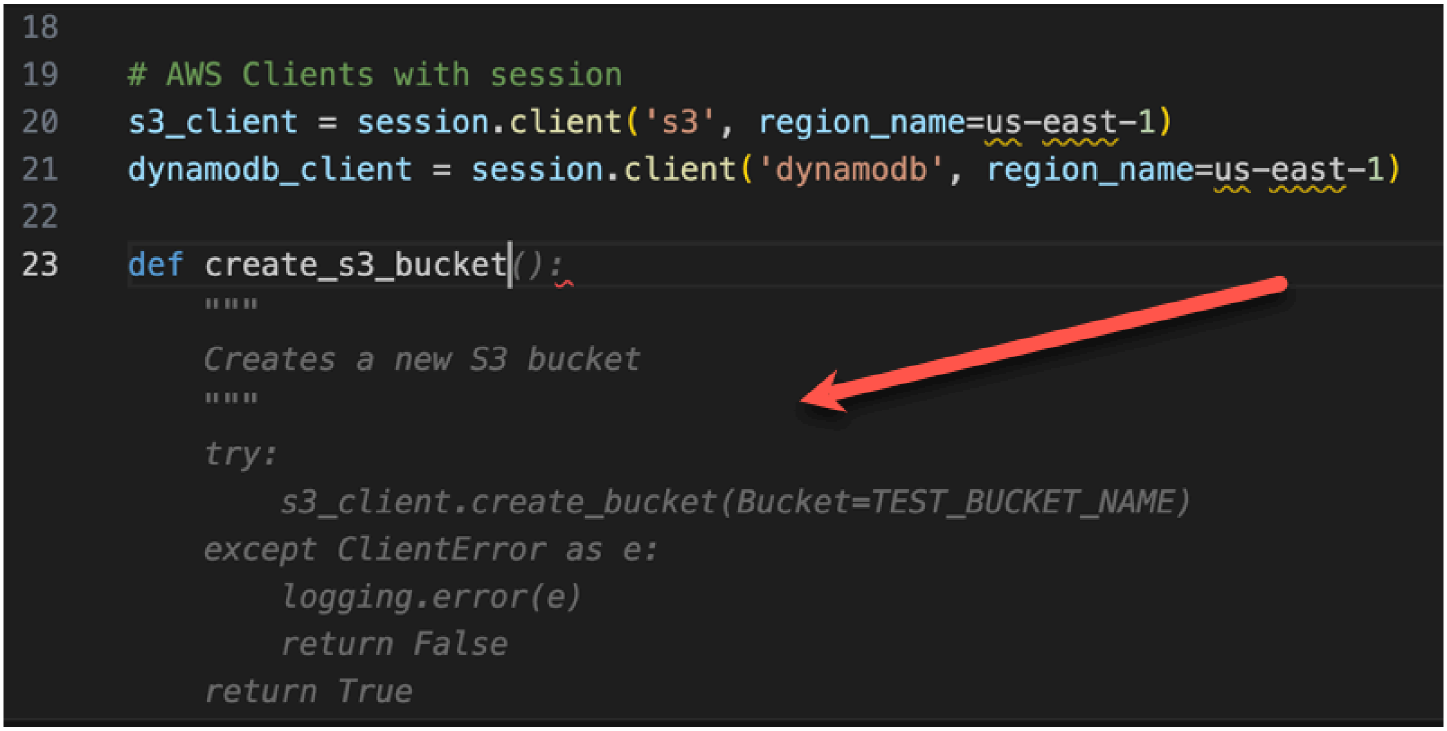
CodeWhisperer, familiar with AWS APIs and SDKs, suggests the correct format.



```
# set up logging
logging.basicConfig(level=logging.INFO)

#Create a new session
session = Session()

# define constants
DEFAULT_REGION = 'us-east-1'
TEST_BUCKET_NAME = 'my-test-bucket' + str(int(time.time()))
TEST_TABLE_NAME = 'my-test-table' + str(int(time.time()))

# AWS Clients with session
s3 = session.client('s3', region_name=DEFAULT_REGION)
dynamodb = session.client('dynamodb', region_name=DEFAULT_REGION)
```

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

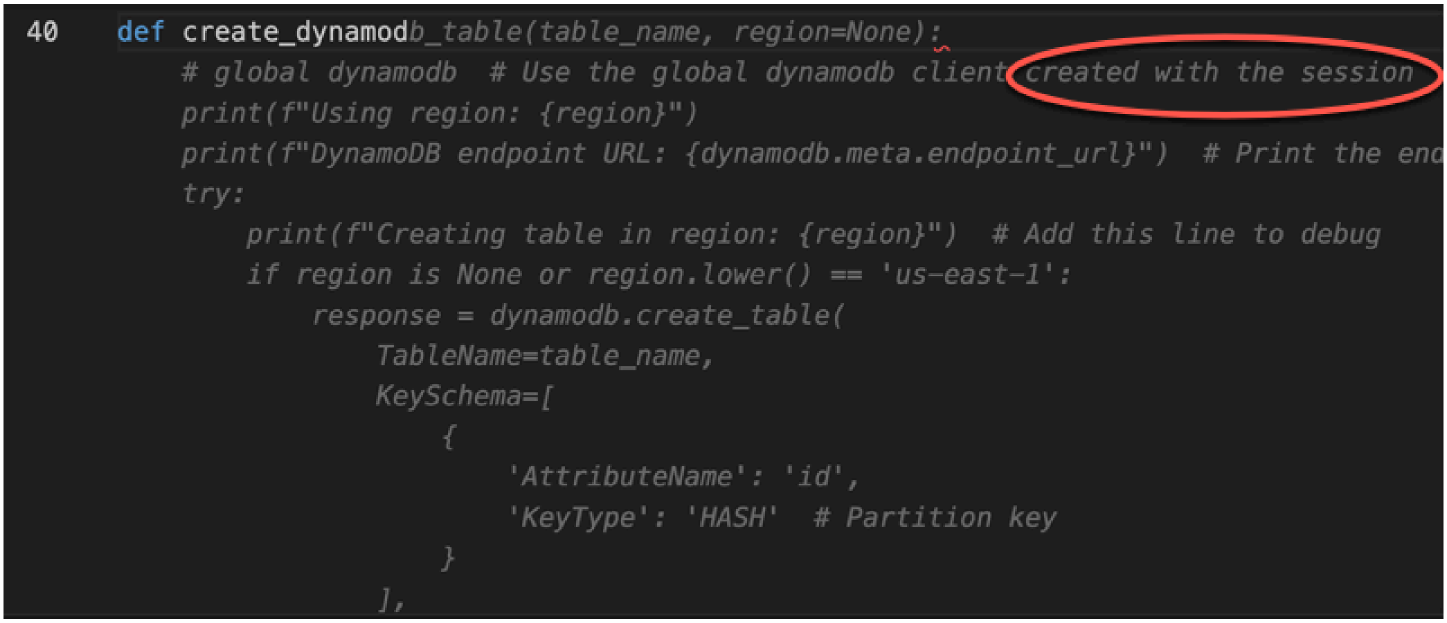
Notice the use of `TEST_BUCKET_NAME`, which is a constant declared earlier in the same file.



```
# AWS Clients with session
s3_client = session.client('s3', region_name=us-east-1)
dynamodb_client = session.client('dynamodb', region_name=us-east-1)

def create_s3_bucket():
    """
    Creates a new S3 bucket
    """
    try:
        s3_client.create_bucket(Bucket=TEST_BUCKET_NAME)
    except ClientError as e:
        logging.error(e)
        return False
    return True
```

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

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



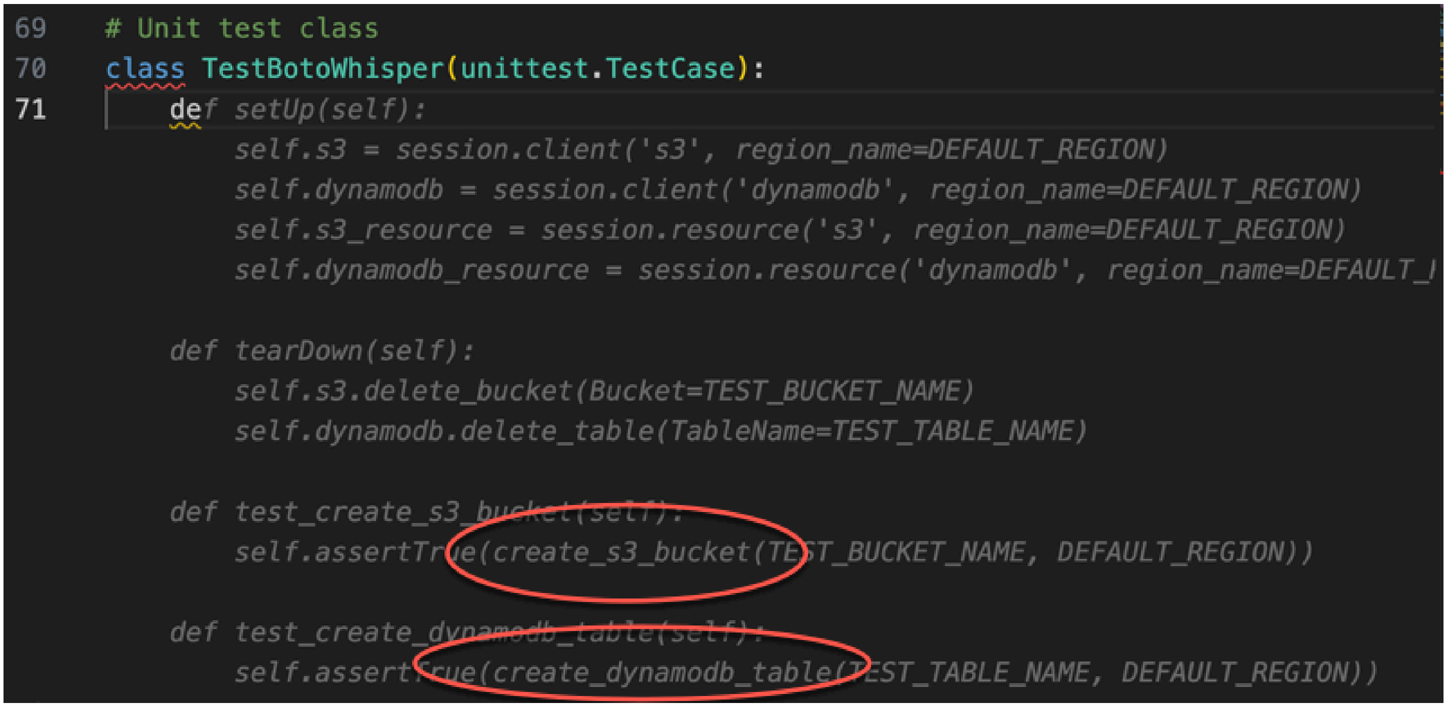
```
def create_dynamodb_table(table_name, region=None):
    # global dynamodb  # Use the global dynamodb client created with the session
    print(f"Using region: {region}")
    print(f"DynamoDB endpoint URL: {dynamodb.meta.endpoint_url}")  # Print the end
    try:
        print(f"Creating table in region: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

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

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

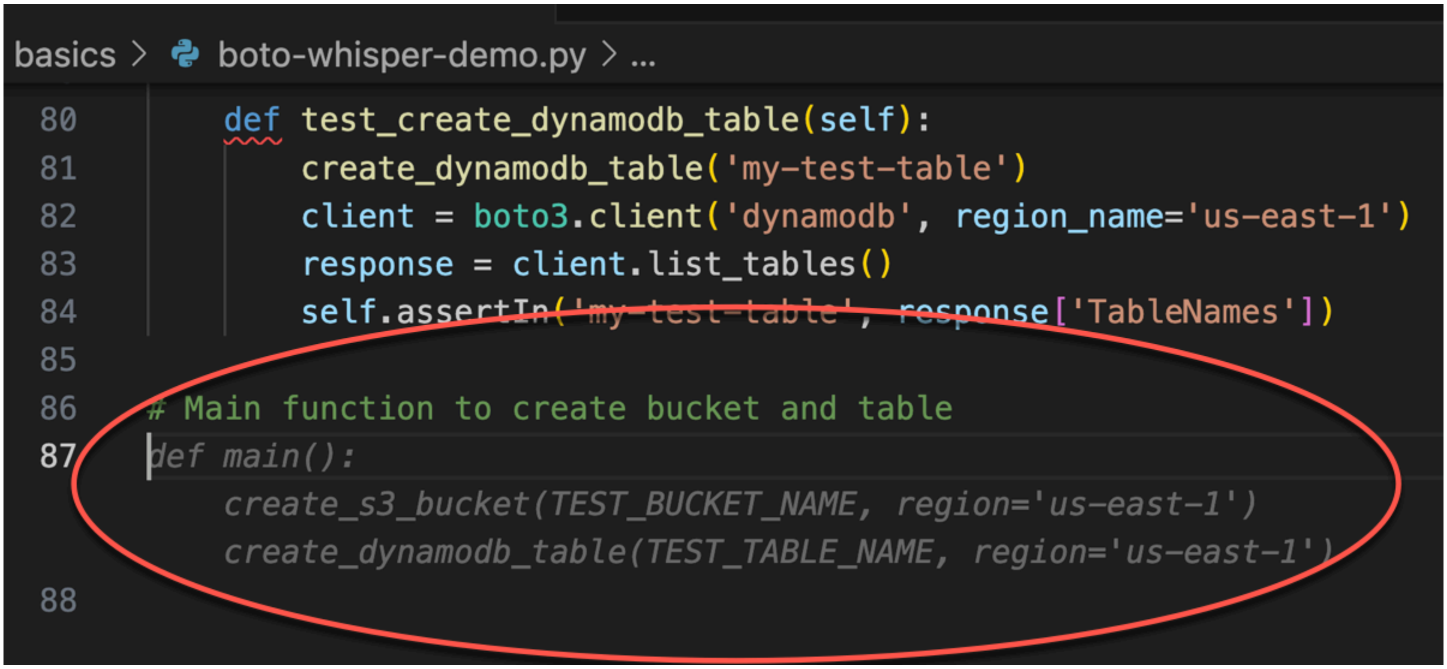
The developer has only just begun to type in the name of the function that will create a DynamoDB table. But CodeWhisperer can tell where this is going.

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



```
# Unit test class
class TestBotoWhisper(unittest.TestCase):
    def setUp(self):
        self.s3 = session.client('s3', region_name=DEFAULT_REGION)
        self.dynamodb = session.client('dynamodb', region_name=DEFAULT_REGION)
        self.s3_resource = session.resource('s3', region_name=DEFAULT_REGION)
        self.dynamodb_resource = session.resource('dynamodb', region_name=DEFAULT_

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

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

    def test_create_dynamodb_table(self):
        self.assertTrue(create_dynamodb_table(TEST_TABLE_NAME, DEFAULT_REGION))
```

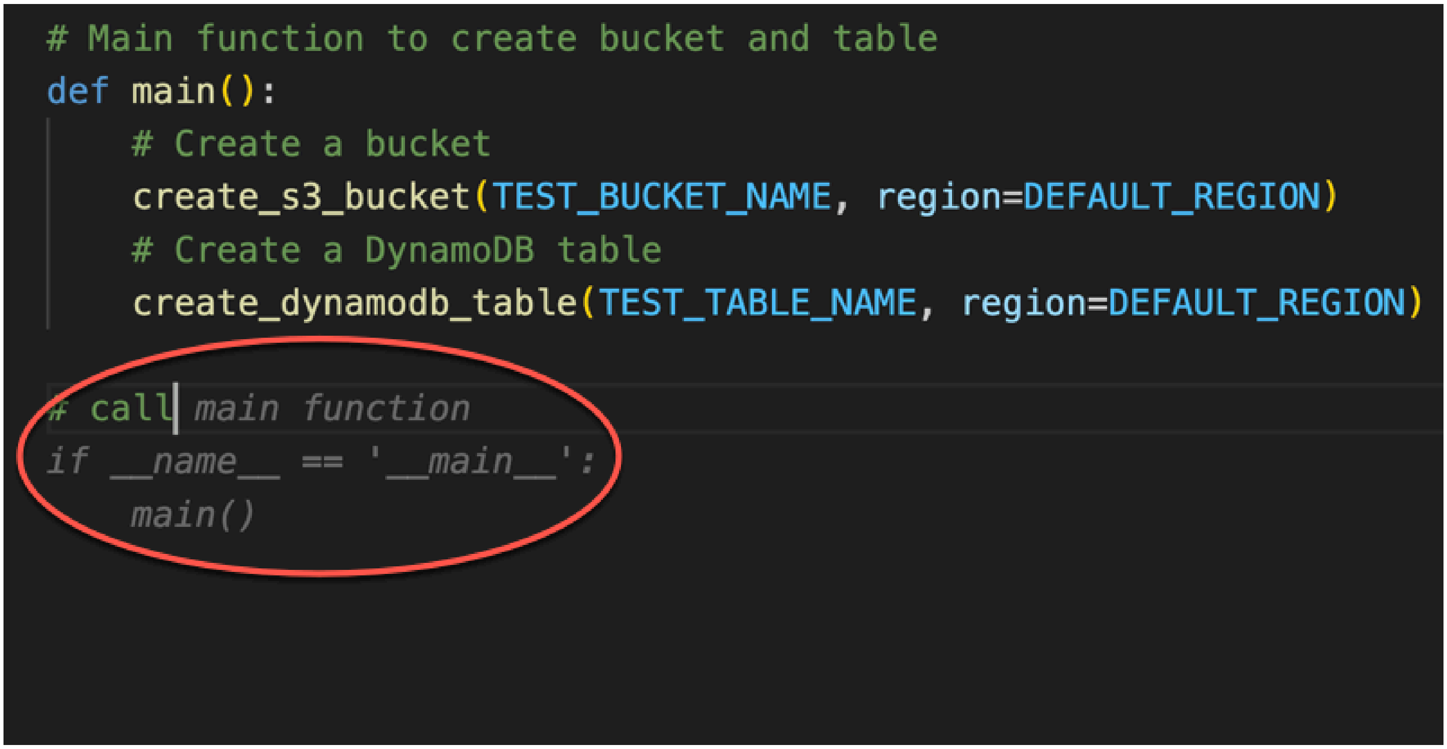
Based only on a comment and the context, CodeWhisperer supplies the entire main function.



```
    def test_create_dynamodb_table(self):
        create_dynamodb_table('my-test-table')
        client = boto3.client('dynamodb', region_name='us-east-1')
        response = client.list_tables()
        self.assertIn('my-test-table', response['TableNames'])

# Main function to create bucket and table
def main():
    create_s3_bucket(TEST_BUCKET_NAME, region='us-east-1')
    create_dynamodb_table(TEST_TABLE_NAME, region='us-east-1')

```

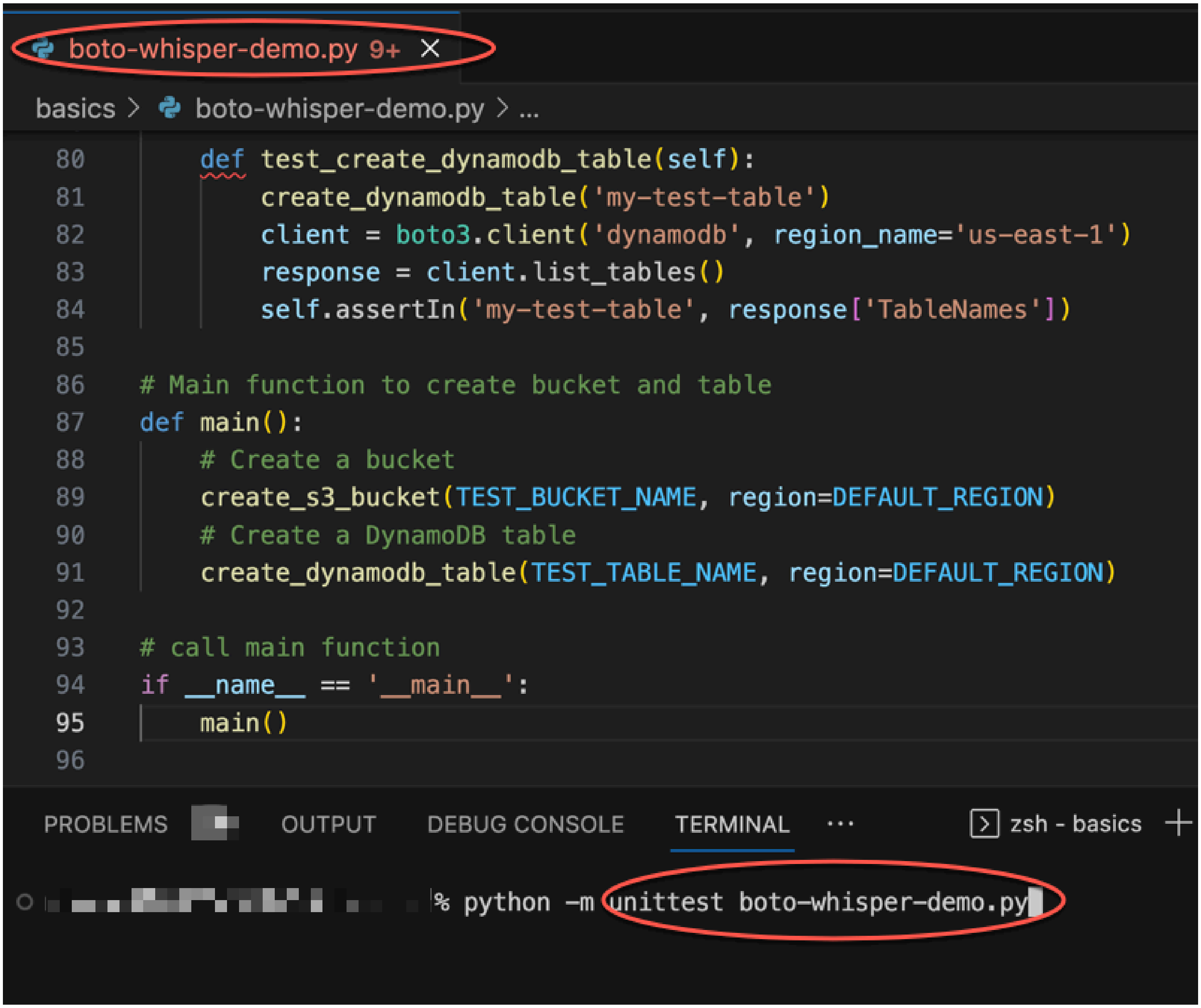
All that's left is the main guard, and CodeWhisperer knows it.

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



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

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

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



```python
    def test_create_dynamodb_table(self):
        create_dynamodb_table('my-test-table')
        client = boto3.client('dynamodb', region_name='us-east-1')
        response = client.list_tables()
        self.assertIn('my-test-table', response['TableNames'])

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

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

```
○ [redacted] % python -m unittest boto-whisper-demo.py
```

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.

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 AWS Lambda, SageMaker Studio, JupyterLab, or AWS Glue Studio, you will have to authenticate using IAM.

For information about authenticating with CodeWhisperer, see Authenticating with CodeWhisperer and AWS Toolkit.

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 a 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.

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 special permissions related to CodeWhisperer

In this section, you will add an inline policy to the permission set that you just created. This policy will allow the IAM Identity Center administrator to create and remove instances of the CodeWhisperer application.

1. In IAM Identity Center, from the left nav, under **Multi-account permissions**, choose **Permission sets**.
2. Choose the **AdministratorAccess** permission set that you created in the previous section.
3. Under **Inline policy**, choose **Edit**.
4. Delete the code in the code window, and paste this in:

```
{
 "Version": "2012-10-17",
 "Statement": [
     {
  "Sid": "Statement1",
  "Effect": "Allow",
  "Action": [
   "sso:CreateManagedApplicationInstance",
   "sso:DeleteManagedApplicationInstance",
   "codewhisperer:CreateProfile",
   "codewhisperer:DeleteProfile"
  ],
  "Resource": [
   "*"
  ]
     }
 ]
}
```

5. At the bottom of the page, choose **Save changes**.

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 Getting started with CodeWhisperer in VS Code and JetBrains.

Setting up CodeWhisperer Professional with IAM Identity Center

Delegate IAM Identity Center administration to a non-management account

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.

If you only have one account in your AWS organization, then that is the management account. You should create additional accounts 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 administer account, follow the steps under Register a member account in the *IAM Identity Center User Guide*.

You do *not* have to administer CodeWhisperer from the same account that you use to administer IAM Identity Center.

Administration of CodeWhisperer occurs on an account-by-account basis within your Organization.

> ⚠ **Warning**
>
> For compatibility with CodeWhisperer, you cannot set up IAM Identity Center in an opt-in Region.

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.

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",
                "sso:CreateManagedApplicationInstance"
            ],
            "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:AssociateProfile",
                "sso:DisassociateProfile",
                "sso:GetProfile",
                "sso:GetSSOStatus"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "identitystore:ListUsers",
                "identitystore:ListGroups"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "organizations:DescribeAccount",
                "organizations:DescribeOrganization"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases",
                "kms:CreateGrant",
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey*",
                "kms:RetireGrant",
                "kms:DescribeKey"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "codeguru-security:UpdateAccountConfiguration"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/aws-service-role/codewhisperer.amazonaws.com/
AWSServiceRoleForCodeWhisperer"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "codewhisperer:UpdateProfile",
                "codewhisperer:ListProfiles",
                "codewhisperer:TagResource",
                "codewhisperer:UnTagResource",
                "codewhisperer:ListTagsForResource",
                "codewhisperer:CreateProfile"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:ListMetrics"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

> **Note**
>
> If you are using CodeWhisperer Customizations, then your CodeWhisperer administrator will require additional permissions. See Prerequisites for CodeWhisperer customizations.

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.

Useful APIs

CodeWhisperer does not have public APIs, in the sense that you cannot call any CodeWhisperer APIs programmatically, and they are not provided by any SDK. However, you can still reference the following APIs in IAM policies or IAM Identity Center permission sets.

- 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.
- DeleteCustomization - Used for deleting a Customization from collected customer data.
- ListCustomizations - Used for listing Customizations based on their state.
- UpdateCustomization - Activates or deactivates a customization.
- ListCustomizationVersions - Lists the versions of a customization.
- GetCustomization - Used for describing a Customization.

Administering end users

If CodeWhisperer has already been set up for your organization

You can set up CodeWhisperer in any member account in your organization. As a matter of best practice it should not be the Organizations management account. It does *not* have to be the same account that you use for IAM Identity Center 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.

Note

The timeout period for your CodeWhisperer or Amazon Q session is either the timeout period that you set in IAM Identity Center, or the timeout period of your third-party identity provider, whichever is lower.

In order to change the timeout period in IAM Identity Center, on the settings page, select the **Authentication** tab. Then, under **Session settings**, choose **Configure**.

Assign users and groups to IAM Identity Center

Warning

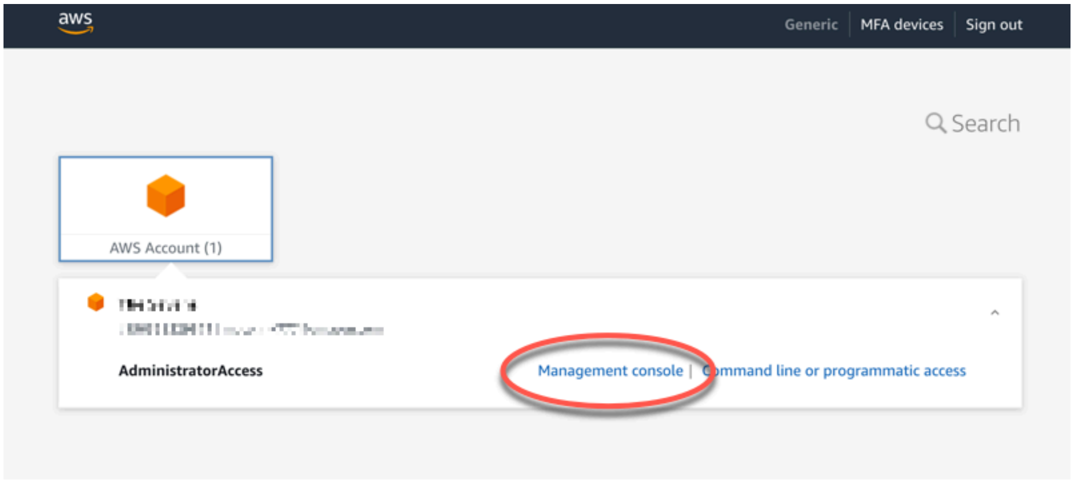
In this procedure, you are acting as the Organizations administrator, logged into the delegator administrator account. Depending on how you were logged in for the previous procedures, you may need to switch users, accounts, and/or roles before continuing.

Your organization's contributors authenticate through AWS IAM Identity Center. 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.

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. For more information, see Assign users and groups to IAM Identity Center

   > **Note**
   >
   > Even if the same user acts as a CodeWhisperer developer in two different accounts within the same organization, your organization will only be billed for that user once per billing cycle.
4. Choose **Set up CodeWhisperer**.

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

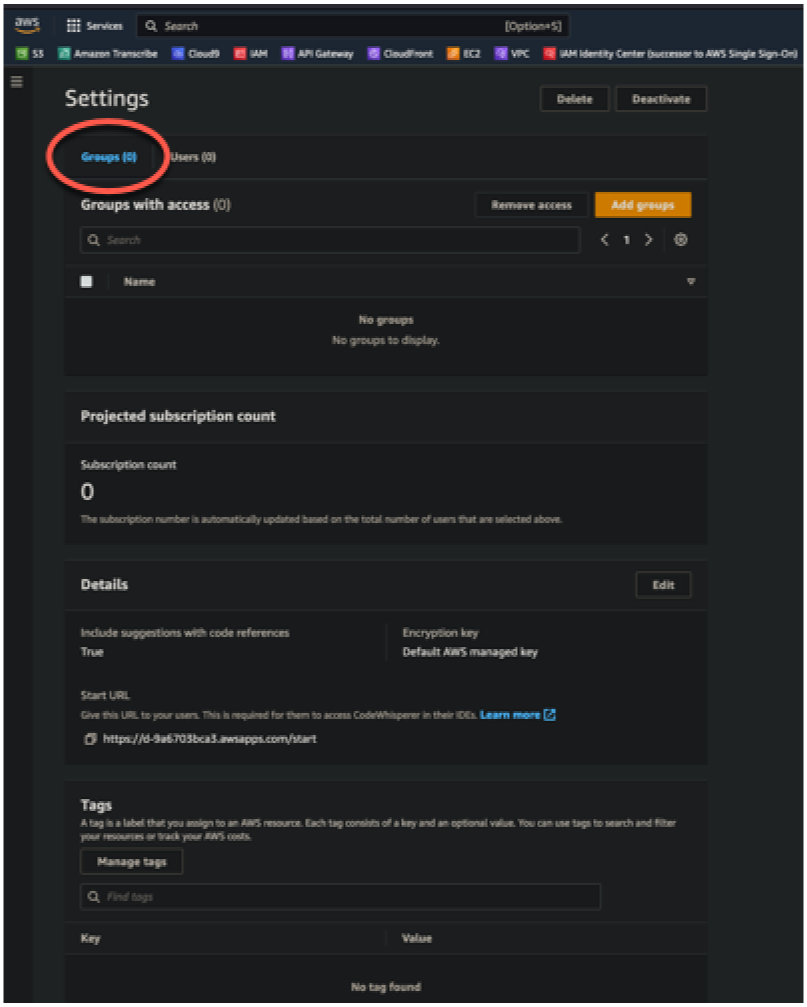
1. From the CodeWhisperer console choose **Settings** to open the **Settings** menu.
2. In the upper right corner of the console window, confirm that the region is set to US East (N. Virginia).

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.



No groups
No groups to display.

Projected subscription count

Subscription count
0
The subscription number is automatically updated based on the total number of users that are selected above.

Details

Include suggestions with code references
True

Encryption key
Default AWS managed key

Start URL
Give this URL to your users. This is required for them to access CodeWhisperer in their IDEs.
https://d-9a6703bca3.awsapps.com/start

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

Key	Value

No tag found

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 your 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 your 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.

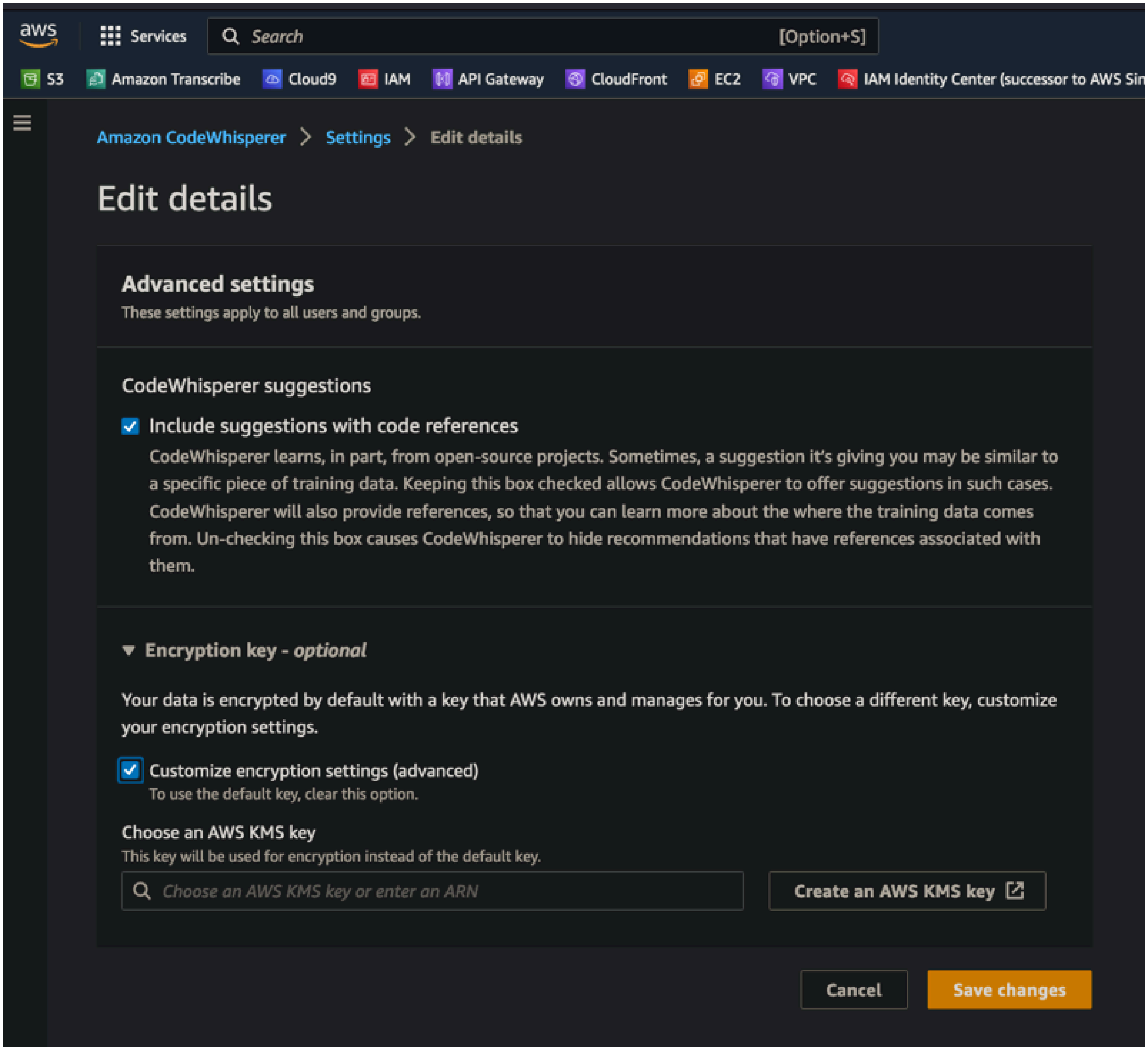
For more information, see Controlling access to AWS resources using policies in the *IAM User Guide*.

Choosing your encryption key

By default, data collected by CodeWhisperer for the purpose of Security scans 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*.


aws
Services
Search
[Option+S]
S3
Amazon Transcribe
Cloud9
IAM
API Gateway
CloudFront
EC2
VPC
IAM Identity Center (successor to AWS Sin
Amazon CodeWhisperer > Settings > Edit details

Edit details

Advanced settings

These settings apply to all users and groups.

CodeWhisperer suggestions

☑ Include suggestions with 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. Keeping this box checked allows CodeWhisperer to offer suggestions in such cases. CodeWhisperer will also provide references, so that you can learn more about the where the training data comes from. Un-checking this box causes CodeWhisperer to hide recommendations that have references associated with them.

▼ Encryption key - *optional*

Your data is encrypted by default with a key that AWS owns and manages for you. To choose a different key, customize your encryption settings.

☑ Customize encryption settings (advanced)

To use the default key, clear this option.

Choose an AWS KMS key

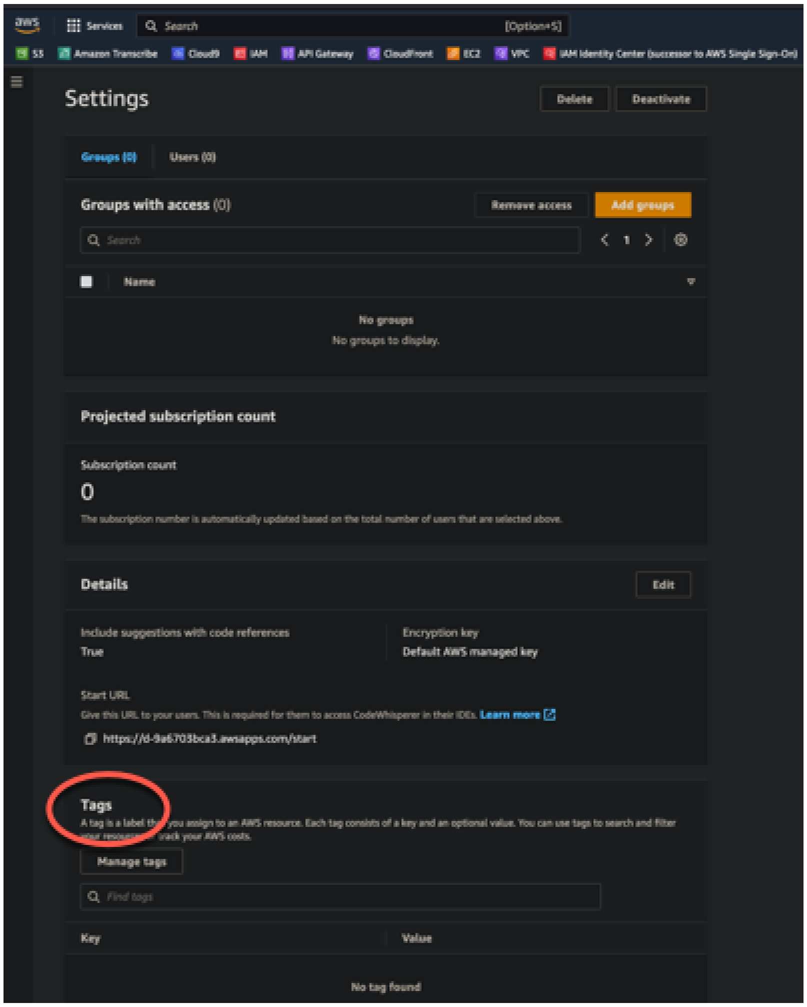
This key will be used for encryption instead of the default key.

Choose an AWS KMS key or enter an ARN

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*.



Settings

Groups with access (0)

No groups
No groups to display.

Projected subscription count

Subscription count
0
The subscription number is automatically updated based on the total number of users that are selected above.

Details

Include suggestions with code references
True

Encryption key
Default AWS managed key

Start URL
Give this URL to your users. This is required for them to access CodeWhisperer in their IDEs. Learn more
https://d-9a6703bca3.awsapps.com/start

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

Key	Value
No tag found	

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**.

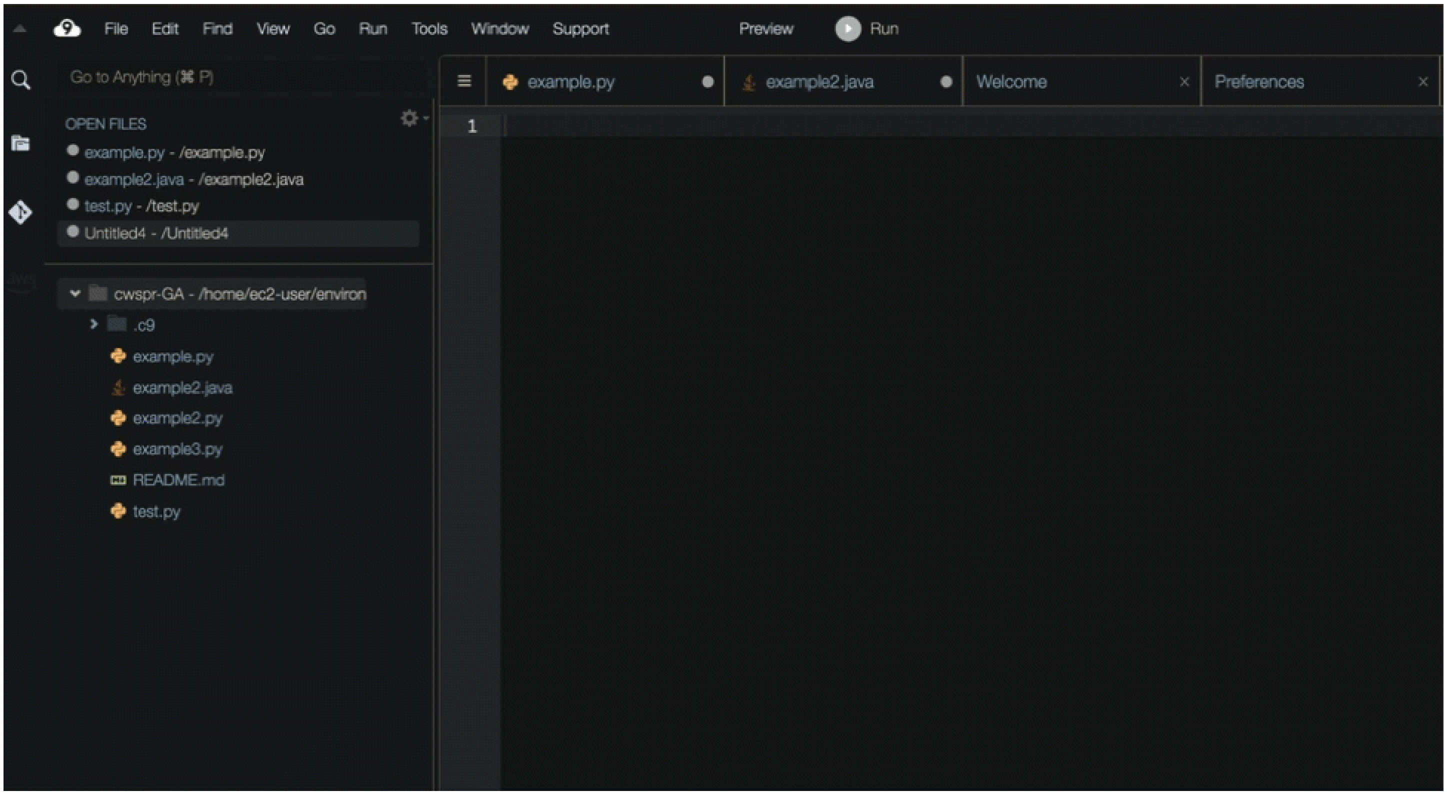
If you are transitioning to Amazon Q Developer, your next step is subscribing to Amazon Q Developer Pro.

Getting started

Note

The fastest way to get started with CodeWhisperer is by using CodeWhisperer for individual developers with VS Code or JetBrains.
If you are already logged into the AWS console, you can also get started quickly using CodeWhisperer with AWS Cloud9.

Here is an example of CodeWhisperer working in AWS Cloud9, using comment completion, single-line completion, line-by-line recommendations, and function completion.



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 Amazon CodeWhisperer for administrators.

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 Getting started with CodeWhisperer in VS Code and JetBrains.

Topics

- CodeWhisperer for command line
- Getting started with CodeWhisperer in VS Code and JetBrains
- Using CodeWhisperer with Visual Studio
- Using CodeWhisperer with Amazon SageMaker Studio
- Using CodeWhisperer with JupyterLab
- Getting started with CodeWhisperer and Amazon EMR Studio
- Using CodeWhisperer with AWS Glue Studio
- Using Amazon CodeWhisperer with AWS Lambda
- Using CodeWhisperer with AWS Cloud9
- Using CodeWhisperer with other services

CodeWhisperer for command line

Topics

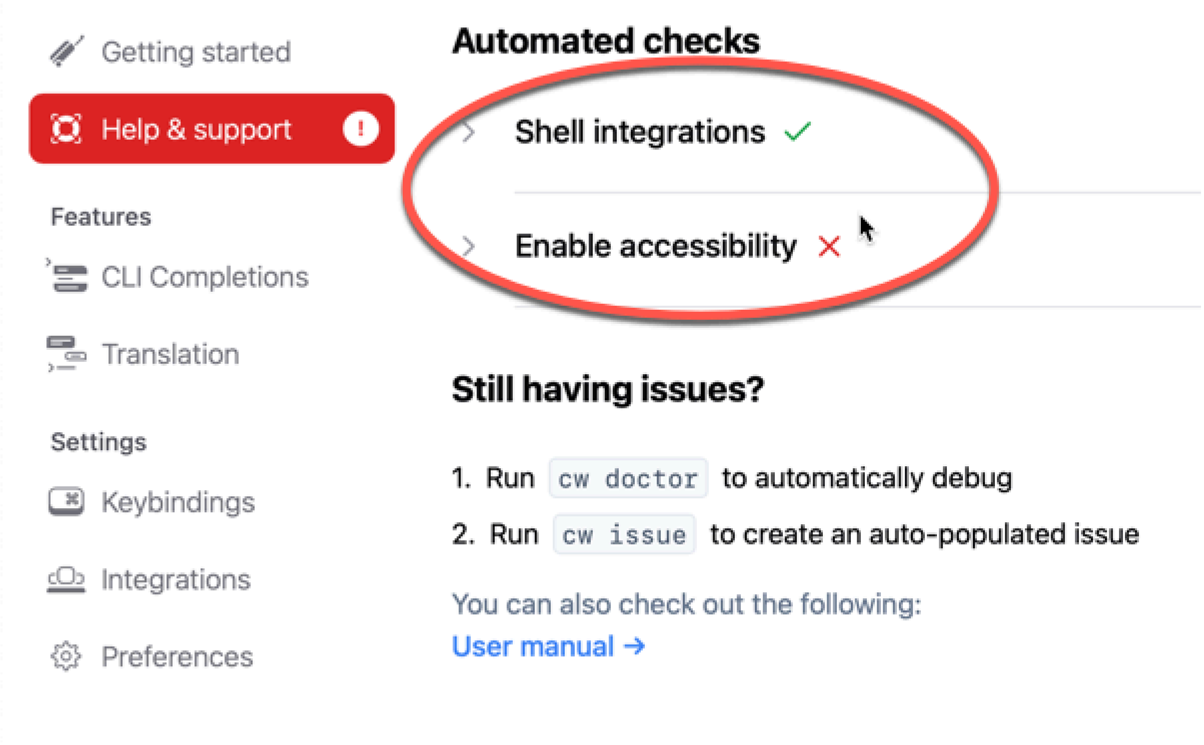
- Installing CodeWhisperer for command line
- CLI Completions
- Natural language to bash translation
- Debugging CodeWhisperer for the command line
- Adding your own completion specs to CodeWhisperer

Installing CodeWhisperer for command line

To install CodeWhisperer for command line, follow the steps below.

1. Download CodeWhisperer for command line (macOS only)

2. Authenticate with Builder ID for CodeWhisperer Individual users, or IAM Identity Center for CodeWhisperer Professional users using the start URL given to you by your account administrator.
3. Follow the instructions to install the shell integrations, and to grant macOS accessiblity permissions.



Automated checks

Shell integrations ✓

Enable accessibility ✗

Still having issues?

1. Run `cw doctor` to automatically debug
2. Run `cw issue` to create an auto-populated issue

You can also check out the following:
User manual →

Supported command line environments

CodeWhisperer for command line integrates with the following environments:

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

Verifying your download

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

```
codesign -v /Applications/CodeWhisperer.app
```

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

For more verbose information about the app signature, run:

```
codesign -dv --verbose=4 /Applications/CodeWhisperer.app
```

To learn more about the macOS codesign utility, see the Code Signing Guide on the Apple developer website.

Uninstalling CodeWhisperer for command line

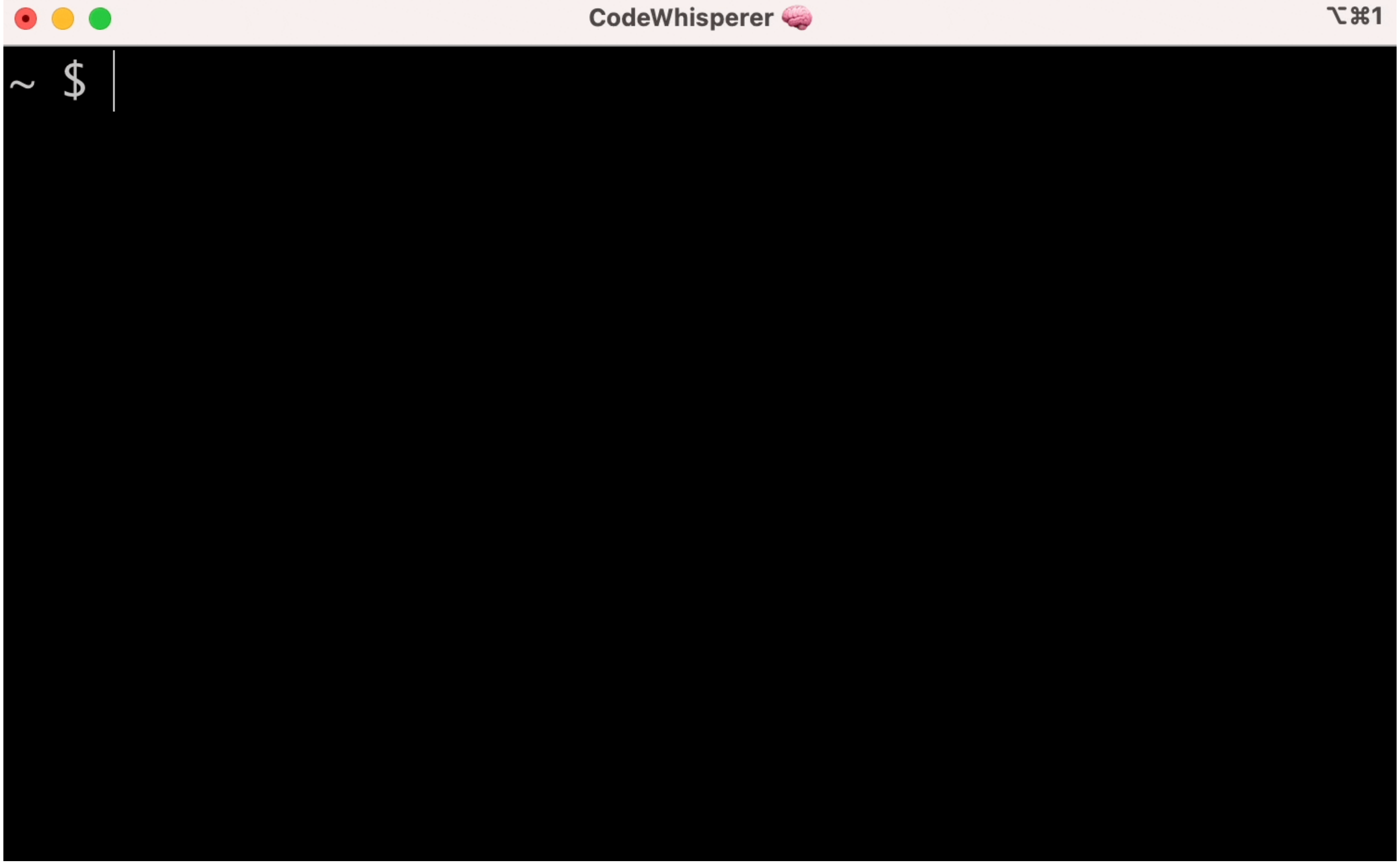
To uninstall CodeWhisperer for command line, complete the following steps.

1. Open a terminal window.
2. Run the following command:

   ```
   cw uninstall
   ```

CLI Completions

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

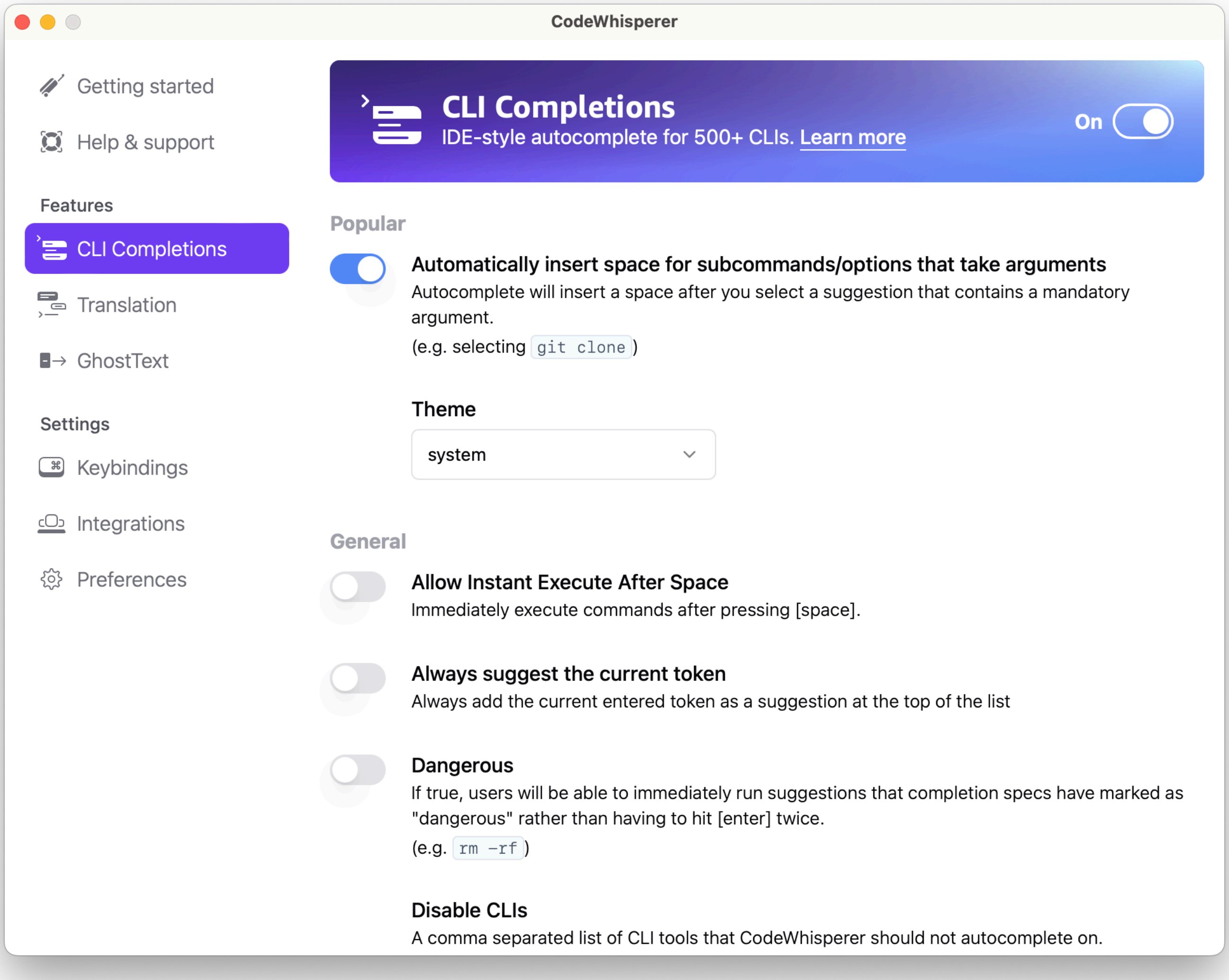


```
~ $
```

Popular settings

The default settings provided by CodeWhisperer for command line may not "feel right," and may disrupt your existing workflow. You can customize your settings at any time by running `cw` to open the settings dashboard. Here are a few popular settings

- Keybindings. Changing the `tab` keybinding to "Insert common prefix or navigate" may make CLI Completions feel more like traditional shell completions while "Insert common prefix or insert" will feel more like an IDE
- Theme. You know what this is. Choose your favorite.
- Instant execute after space. A lot developers habitually type a space character just before they execute it. Enable this setting to avoid CodeWhisperer blocking you
- First token completion. Enable this setting to get completions for CLIs themselves, not just the subcommands, options, and arguments



CLI Completions
IDE-style autocomplete for 500+ CLIs. Learn more

Popular

Automatically insert space for subcommands/options that take arguments
Autocomplete will insert a space after you select a suggestion that contains a mandatory argument.
(e.g. selecting `git clone`)

Theme
system

General

Allow Instant Execute After Space
Immediately execute commands after pressing [space].

Always suggest the current token
Always add the current entered token as a suggestion at the top of the list

Dangerous
If true, users will be able to immediately run suggestions that completion specs have marked as "dangerous" rather than having to hit [enter] twice.
(e.g. `rm -rf`)

Disable CLIs
A comma separated list of CLI tools that CodeWhisperer should not autocomplete on.

Natural language to bash translation

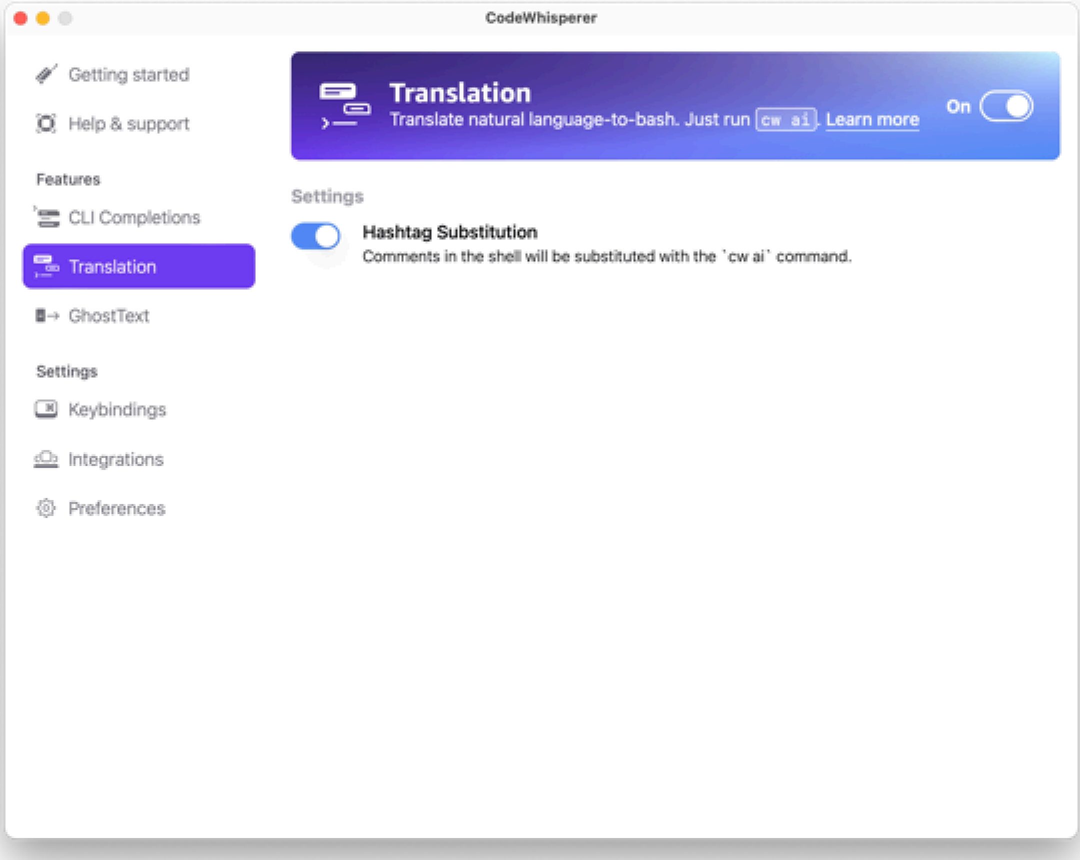
The `cw ai` command lets you write a natural language instruction such as “copy all files in my current directory to Amazon S3”. CodeWhisperer will then translate it to an instantly executable shell code snippet. The `cw ai` command is useful in those common situations where the correct bash syntax is easy to forget. Examples include reversing a `git` commit, finding strings inside files with `grep`, or compressing files with `tar`.

To get started, run either of the following

- `cw ai` *`prompt`*
- `#` *`prompt`*

```
~ $
```

To opt out of using # to invoke CodeWhisperer, go to **Settings -> Translate** and toggle off **Hashtag substitution**.



Translation

Translate natural language-to-bash. Just run `cw ai` Learn more

Settings

Hashtag Substitution

Comments in the shell will be substituted with the `cw ai` command.

Debugging CodeWhisperer for the command line

If you're having a problem with CodeWhisperer for command line, run `cw doctor`.

`cw doctor` identifies and fixes common issues. Most of the time, you won't need to do anything else.

Expected output

```
$ cw doctor

# Everything looks good!

CodeWhisperer still not working? Run cw issue to let us know!
```

If your output doesn't look like the output above, follow the prompts to resolve your issue. If it's still not working, run `cw issue` to report the bug.

Adding your own completion specs to CodeWhisperer

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

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

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

Getting started with CodeWhisperer in VS Code and JetBrains

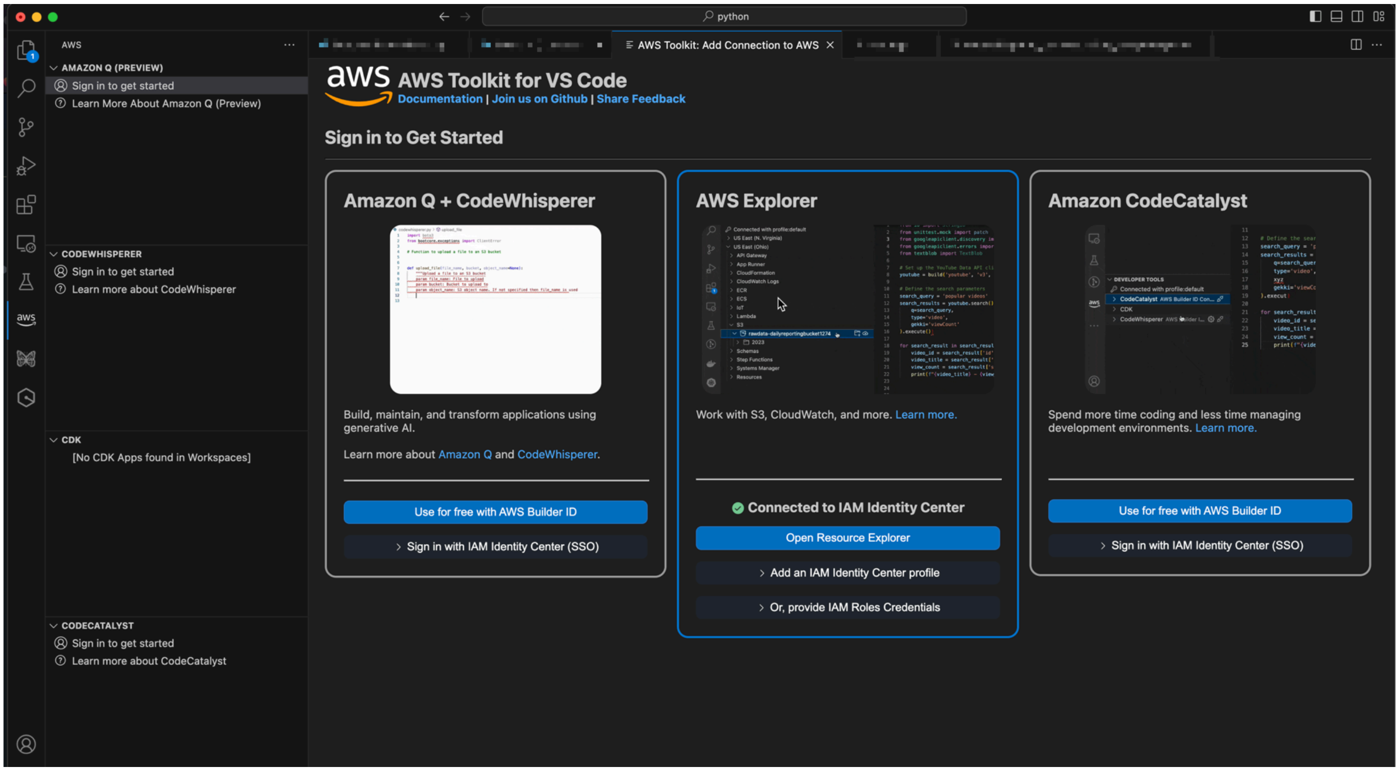
> ⚠ **Important**
>
> Before you proceed, make sure that you are using the latest version of both your IDE and the AWS Toolkit.

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

VS Code

1. From the AWS Toolkit for VS Code, in the AWS pane, under **CodeWhisperer**, choose **Sign in to get started**.

   The **AWS Toolkit: Add Connection to AWS** tab will open.
2. Select the **Amazon Q + CodeWhisperer** authentication panel.
3. Select the appropriate authentication method and log in.


python
AWS
AMAZON Q (PREVIEW)
Sign in to get started
Learn More About Amazon Q (Preview)
CODEWHISPERER
Sign in to get started
Learn more about CodeWhisperer
CDK
CODECATALYST
Sign in to get started
Learn more about CodeCatalyst
AWS Toolkit: Add Connection to AWS
aws AWS Toolkit for VS Code
Documentation | Join us on Github | Share Feedback
Sign in to Get Started
Amazon Q + CodeWhisperer
Use for free with AWS Builder ID
Sign in with IAM Identity Center (SSO)
AWS Explorer
Connected to IAM Identity Center
Open Resource Explorer
Add an IAM Identity Center profile
Or, provide IAM Roles Credentials
Amazon CodeCatalyst
Use for free with AWS Builder ID
Sign in with IAM Identity Center (SSO)

[No CDK Apps found in Workspaces]

Build, maintain, and transform applications using generative AI.

Learn more about Amazon Q and CodeWhisperer.

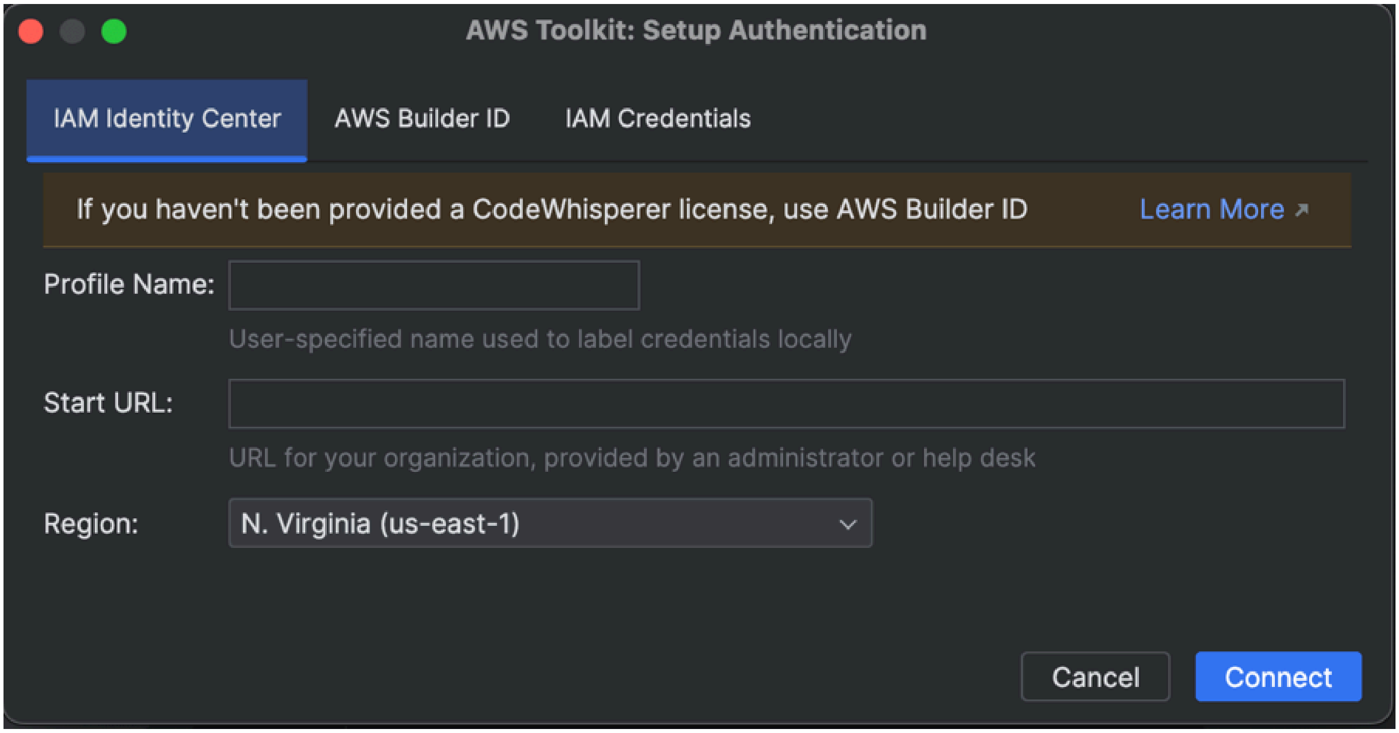
Work with S3, CloudWatch, and more. Learn more.

Spend more time coding and less time managing development environments. Learn more.

JetBrains

1. From the AWS Toolkit for JetBrains, select the **Amazon Q + CodeWhisperer** tab.
2. Under CodeWhisperer, choose **Sign in to get started**.

   The **AWS Toolkit: Setup Authentication** modal will open.
3. Select the appropriate authentication method and log in.



If you haven't been provided a CodeWhisperer license, use AWS Builder ID

Profile Name:

User-specified name used to label credentials locally

Start URL:

URL for your organization, provided by an administrator or help desk

Region: N. Virginia (us-east-1)

Authenticating with CodeWhisperer and AWS Toolkit

To use CodeWhisperer with the AWS Toolkit for Visual Studio Code or AWS Toolkit for JetBrains, 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

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 professional-tier developer. That is, you are working with CodeWhisperer as an employee of an organization that has an AWS account, and that is paying for a CodeWhisperer Professional license. Before you can authenticate using IAM Identity Center, your administrator must add you as a user. Your administrator will then provide you with the Start URL that you need to log in with IAM Identity Center.

At the professional tier, you can use CodeWhisperer to give you suggestions that conform to your team's internal libraries with customizations.

Learn more about IAM Identity Center

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.

If you have acquired the tool independent of your team or organization, you use CodeWhisperer Individual, and you will use AWS Builder ID to log in.

Learn about Builder ID

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.

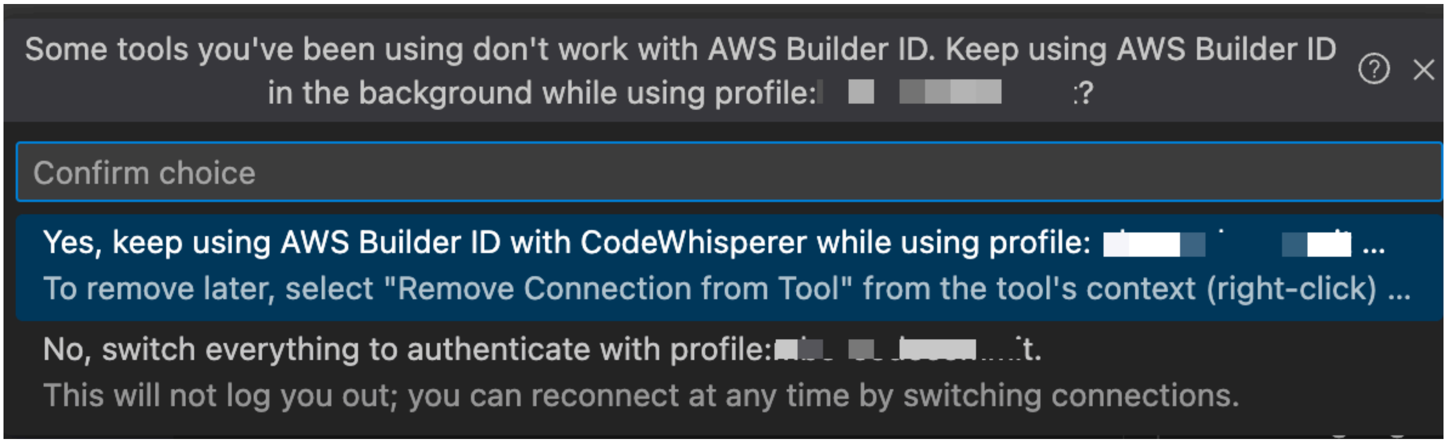
Learn about IAM

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.



Some tools you've been using don't work with AWS Builder ID. Keep using AWS Builder ID in the background while using profile: [redacted] :?

Yes, keep using AWS Builder ID with CodeWhisperer while using profile: [redacted] ...
To remove later, select "Remove Connection from Tool" from the tool's context (right-click) ...

No, switch everything to authenticate with profile: [redacted]t.
This will not log you out; you can reconnect at any time by switching connections.

Using CodeWhisperer with Visual Studio

The CodeWhisperer Visual Studio integration is in preview, and is subject to change.

This page descrbes how to set up and begin using CodeWhisperer with Visual Studio.

Note

The languages that CodeWhisperer supports with Visual Studio are: C, C++, and C#.

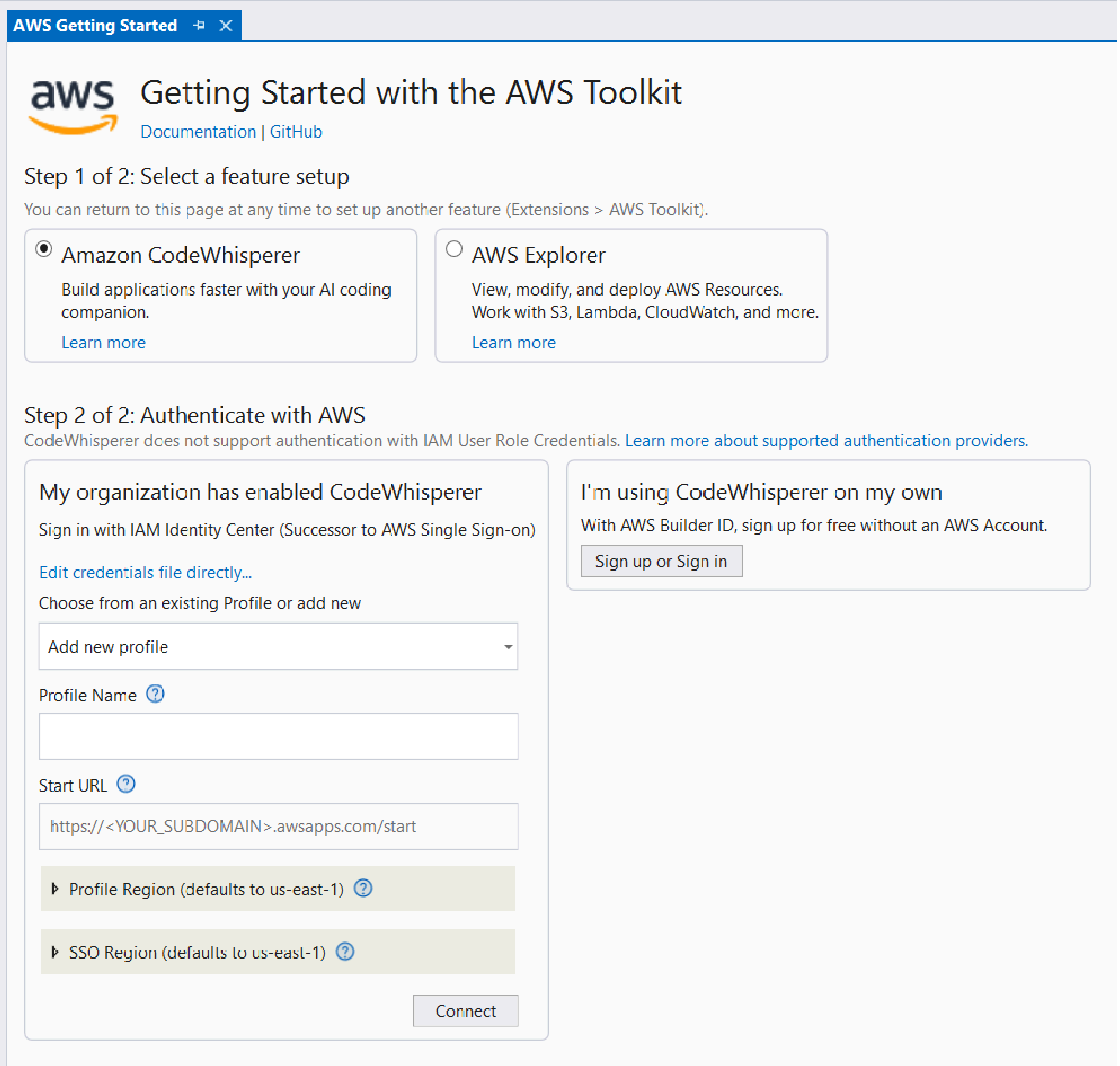
Note

The AWS Toolkit provides CodeWhisperer functionality through a standalone program called a language server. As you open a solution in Visual Studio, the Toolkit downloads and updates the language server in the background. The Toolkit then launches the language server as a separate process from Visual Studio. The language server is run with the same privilege level as Visual Studio, and is automatically closed when you close Visual Studio.

1. Install Visual Studio 2022.
2. Install the latest version of the AWS Toolkit for Visual Studio 2022.
3. On the Toolkit **Getting Started** page, select CodeWhisperer.

   You can return to the **Getting Started** page at any time with **Extensions** -> **AWS Toolkit** -> **Getting Started**.
4. Authenticate with either IAM Identity Center (for CodeWhisperer Professional) or AWS Builder ID (for CodeWhisperer Individual).

   Use CodeWhisperer for free at the individual tier by authenticating with AWS Builder ID. Alternatively, use the professional tier if your company has a license by authenticating with IAM Identity Center.



AWS Getting Started

Getting Started with the AWS Toolkit

Documentation | GitHub

Step 1 of 2: Select a feature setup

You can return to this page at any time to set up another feature (Extensions > AWS Toolkit).

● Amazon CodeWhisperer

Build applications faster with your AI coding companion.

Learn more

○ AWS Explorer

View, modify, and deploy AWS Resources. Work with S3, Lambda, CloudWatch, and more.

Learn more

Step 2 of 2: Authenticate with AWS

CodeWhisperer does not support authentication with IAM User Role Credentials. Learn more about supported authentication providers.

My organization has enabled CodeWhisperer

Sign in with IAM Identity Center (Successor to AWS Single Sign-on)

Edit credentials file directly...

Choose from an existing Profile or add new

Add new profile

Profile Name

Start URL

https://<YOUR_SUBDOMAIN>.awsapps.com/start

Profile Region (defaults to us-east-1)

SSO Region (defaults to us-east-1)

Connect

I'm using CodeWhisperer on my own

With AWS Builder ID, sign up for free without an AWS Account.

Sign up or Sign in

Using 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.**

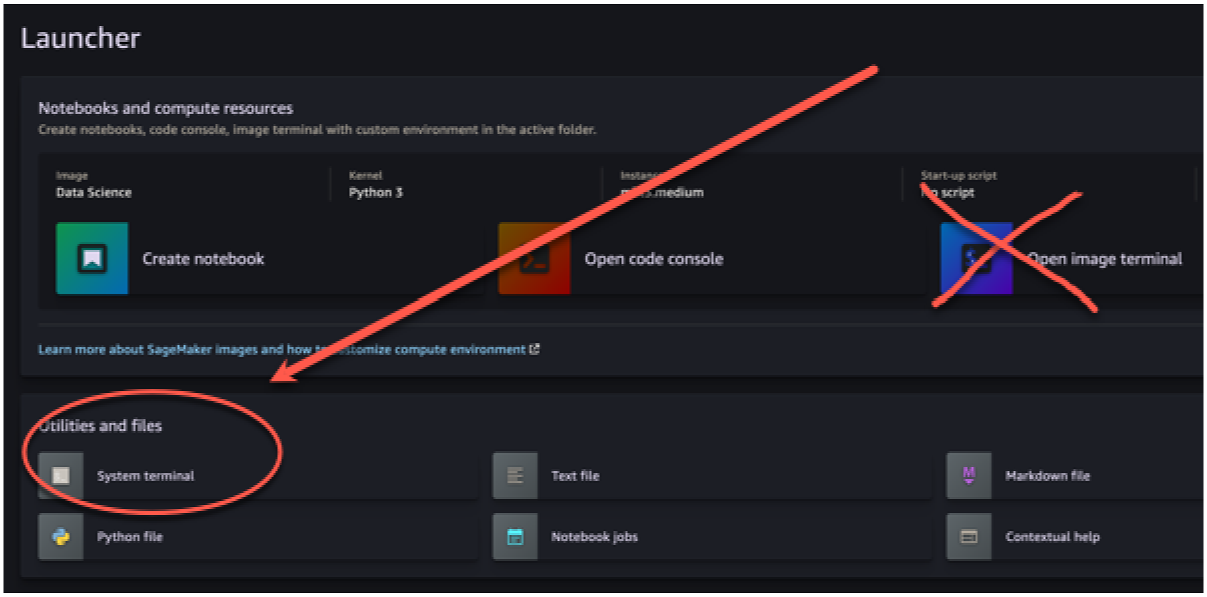
   Create an IAM policy containing the following statement. Then attach that policy to the execution role (IAM) or permission set (IAM Identity Center) associated with your user profile.

   ```
   {
     "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.**

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



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

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

5. **Open a new notebook.**

> **Note**
>
> Code completions with 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.

Using 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.

Installing JupyterLab itself

Install JupyterLab on your computer or if you already have JupyterLab installed, check it's version by running the following command.

```
pip show jupyterlab
```

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

Installation Using Pip for Jupyter Lab version >= 4.0

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

```
# JupyterLab 4
pip install amazon-codewhisperer-jupyterlab-ext
sudo systemctl restart jupyter-server
```

After running the code above, refresh your browser. You should be able to use CodeWhisperer.

If you encounter a permissions issue, run:

```
aws sts get-caller-identity
```

This will return the name of the identity that requires permission to use CodeWhisperer. If you are using SageMaker, then it will be the SageMaker execution role.

Add the appropriate permissions to the role, and refresh your browser again.

Installation Using Pip for Jupyter Lab version >= 3.6 and < 4.0

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

```
# JupyterLab 3
pip install amazon-codewhisperer-jupyterlab-ext~=1.0
jupyter server extension enable amazon_codewhisperer_jupyterlab_ext
sudo systemctl restart jupyter-server
```

After running the code above, refresh your browser. You should be able to use CodeWhisperer.

If you encounter a permissions issue, run:

```
aws sts get-caller-identity
```

This will return the name of the identity that requires permission to use CodeWhisperer. If you are using SageMaker, then it will be the SageMaker execution role.

Add the appropriate permissions to the role, and refresh your browser again.

Authenticating with AWS Builder ID

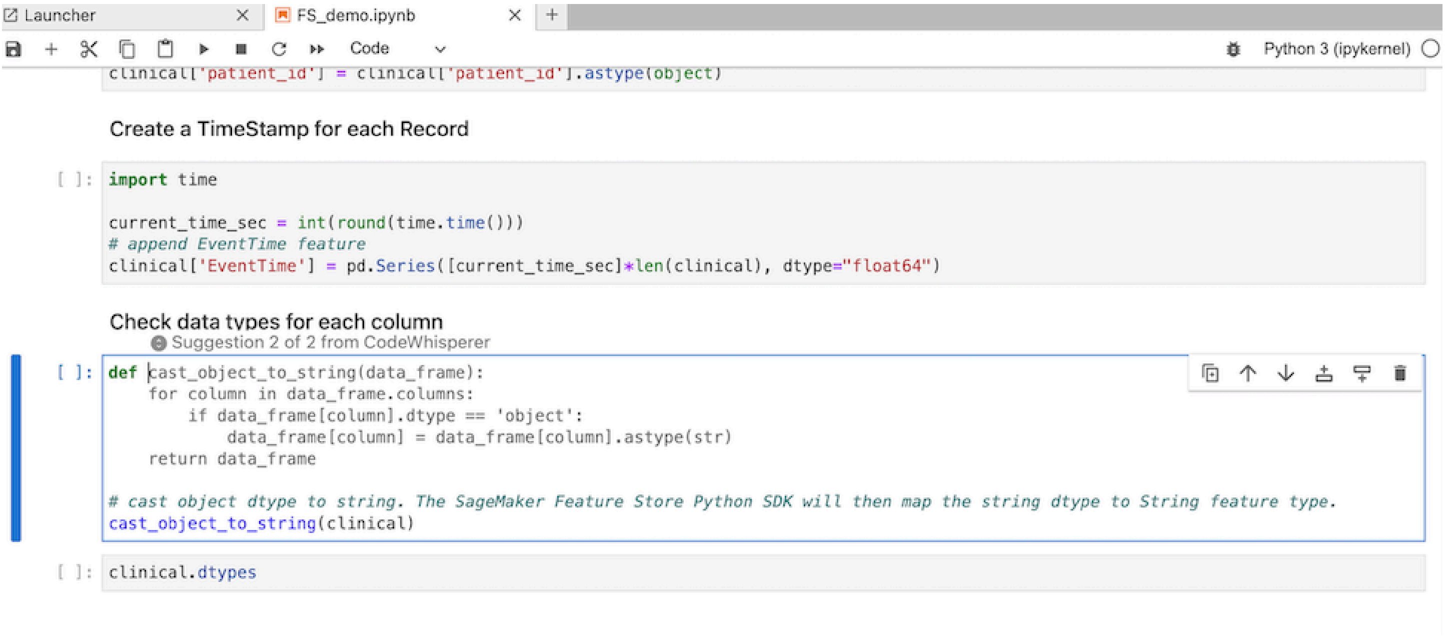
In the following procedure, you will set up Builder ID, which you will use to authenticate when you enable CodeWhisperer.

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.



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

Create a TimeStamp for each Record

```
[ ]: import time

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

Check data types for each column

Suggestion 2 of 2 from CodeWhisperer

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

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

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

Getting started with CodeWhisperer and Amazon EMR Studio

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

> **Note**
>
> CodeWhisperer supports Python, which can be used to code ETL scripts for Spark jobs in Amazon EMR Studio.

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

1. Set up Amazon EMR Studio Notebook.
2. Attach the following policy to the IAM user role for Amazon EMR Studio Notebook.

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

3. Open the Amazon EMR console.
4. Under Amazon EMR Studio, choose **Workspaces (Notebooks).**
5. Select your desired Workspace and choose **Quick launch**.

Using 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**.

Using 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.

For shortcut keys, see User actions.

Using 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.

Using CodeWhisperer with other services

AWS Identity and Access Management permissions for other services

For CodeWhisperer to provide recommendations in the context of another service, you must enable the correct IAM permissions for either your IAM user or role. You must add the `codewhisperer:GenerateRecommendations` permission, as outlined in the sample IAM policy below:

```
{
  "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, see Security best practices in the *IAM user guide*.

Features

CodeWhisperer's most prominent feature is its ability to provide you with suggestions while you're writing code.

CodeWhisperer anticipates how you're going to finish a line of code or a comment line. It can generate a full function for you, or it can complete a code block. It can suggest functions to complete docstrings, and it can provide line-by-line suggestions as you code at your own pace.

When you use CodeWhisperer with VS Code or JetBrains, it integrates with Amazon CodeGuru to perform security scans on both your active file and its dependents, highlighting any issues it finds.

Topics

Customizations

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

> **Note**
>
> Customizations are only available with CodeWhisperer Professional.

Every software development team has a different way of writing code. You may want CodeWhisperer to give you suggestions that conform to your team's internal libraries, proprietary algorithmic techniques, and enterprise code style.

In that case, CodeWhisperer customizations can help you. A customization is a set of elements that enables CodeWhisperer to provide you with suggestions based on your company's codebase.

Topics

Prerequisites for CodeWhisperer customizations

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

CodeWhisperer customizations build upon the foundation of CodeWhisperer Professional, and uses its features.

To use CodeWhisperer customizations you must first follow the CodeWhisperer Professional setup process under Setting up Amazon CodeWhisperer for administrators. This includes adding any users to your CodeWhisperer Professional profile that you also wish to grant access to CodeWhisperer Customizations.

When you use CodeWhisperer Customizations, your CodeWhisperer administrator must be authorized to access your codebase, which you may store on Amazon S3 or through AWS CodeStar. However, during the standard setup process for CodeWhisperer Professional, your AWS Organizations administrator does not provide the CodeWhisperer administrator with access to those services.

Note

If you are using GitHub as your data source, you can restrict usage to certain repositories. See Create a connection to GitHub in the Developer Tools Console User Guide.

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

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "sso-directory:DescribeUsers"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "codewhisperer:CreateCustomization",
                "codewhisperer:DeleteCustomization",
                "codewhisperer:ListCustomizations",
                "codewhisperer:UpdateCustomization",
                "codewhisperer:GetCustomization",
                "codewhisperer:ListCustomizationPermissions",
                "codewhisperer:AssociateCustomizationPermission",
                "codewhisperer:DisassociateCustomizationPermission"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "codestar-connections:ListConnections",
                "codestar-connections:ListOwners",
                "codestar-connections:ListRepositories",
                "codestar-connections:GetConnection"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "codestar-connections:UseConnection",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "codestar-connections:ProviderAction": [
                        "GitPull",
                        "ListRepositories",
                        "ListOwners"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject*",
                "s3:GetBucket*",
                "s3:ListBucket*"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Accessing customization-related messages in Amazon CloudWatch Logs

CodeWhisperer stores information about the creation of your customization in Amazon CloudWatch Logs.

You can authorize your CodeWhisperer administrator to view those logs with the following permission set.

To learn more about the permissions required to delivery logs to multiple resources, see Logging that requires additional permissions [V2] in the *Amazon CloudWatch Logs User Guide*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:PutDeliverySource",
                "logs:GetDeliverySource",
                "logs:DeleteDeliverySource",
                "logs:DescribeDeliverySources",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestination",
                "logs:DeleteDeliveryDestination",
                "logs:DescribeDeliveryDestinations",
                "logs:CreateDelivery",
                "logs:GetDelivery",
                "logs:DeleteDelivery",
                "logs:DescribeDeliveries",
                "firehose:ListDeliveryStreams",
                "firehose:DescribeDeliveryStream",
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:account number:log-group:*",
                "arn:aws:firehose:us-east-1:account number:deliverystream/*",
                "arn:aws:s3:::*"
            ]
        }
    ]
}
```

For more information about setting the permissions needed to administer CodeWhisperer Professional, see Assigning CodeWhisperer administration rights.

> **Note**
>
> The encryption key that you set up for CodeWhisperer Professional is also used for CodeWhisperer Customizations.

It's important to create your customization using the best possible source material. When preparing your data source, add code containing patterns that are encouraged on your team. Avoid code containing anti-patterns, bugs, security vulnerabilities, performance issues, and so forth.

Your data source must contain at least 20 MB, and at most 7 GB, of source code files from supported languages. There is no limit on the number of files, but you must include at least 10 files for each language that you want your customization to support. In the Amazon S3 data source, ensure that all source code is placed within a directory and not at the root level. Any files at the root level will be ignored.

> **Note**
>
> CodeWhisperer Customizations supports the following languages and file extensions:
>
> - Java (.java)
> - JavaScript (.js, .jsx)
> - Python (.py)
> - TypeScript (.ts, .tsx)

Creating your customization

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

This section explains how to create a customization with CodeWhisperer.

To create your customization, follow this procedure:

1. Complete your setup of CodeWhisperer Professional. This includes enabling IAM Identity Center and authorizing an administrator to CodeWhisperer, and activating the CodeWhisperer console.

2. Open the CodeWhisperer console.
3. From the navigation pane on the left, choose **Customizations**.
4. The customizations page will appear.
5. Choose **Create customization**.
6. Enter a customization name and (optional) description.

> **Note**
>
> Use both names and descriptions that will be informative to your developers. Developers from your organization who are authorized to use CodeWhisperer Enterprise will be able to see them in VS Code or JetBrains through the AWS plugin.

Connecting to your data source

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

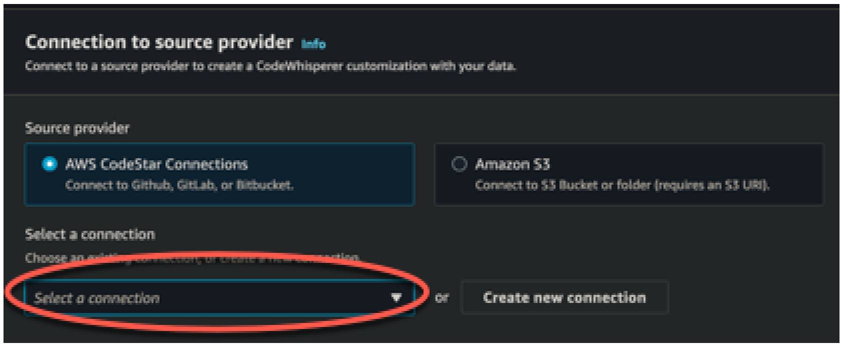
Before you create a customization, you must connect to the data source that contains your codebase. How you do this depends on where your data source is.

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

To learn more about CodeConnections, see What are connections? in the *Developer Tools console User Guide*

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

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



Connection to source provider Info

Connect to a source provider to create a CodeWhisperer customization with your data.

Source provider

● AWS CodeStar Connections
Connect to Github, GitLab, or Bitbucket.

○ Amazon S3
Connect to S3 Bucket or folder (requires an S3 URI).

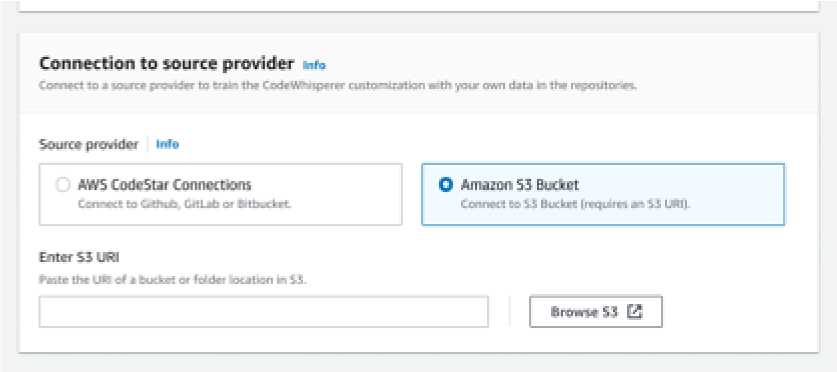
Select a connection

Choose an existing connection, or create a new connection.

Select a connection ▼ or Create new connection

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

3. In the pop-up window that opens, navigate to your data source and follow the instructions in the console.
4. After you create your data source, return to the **Create customization** page.
5. Under **Select a connection**, select your connection from the dropdown.



Connection to source provider Info

Connect to a source provider to train the CodeWhisperer customization with your own data in the repositories.

Source provider | Info

○ AWS CodeStar Connections
Connect to Github, GitLab or Bitbucket.

● Amazon S3 Bucket
Connect to S3 Bucket (requires an S3 URI).

Enter S3 URI

Paste the URI of a bucket or folder location in S3.

Browse S3

To connect to your data source through Amazon S3, follow this procedure:

1. Under **Connection to source provider**, select Amazon S3.
2. Choose **Browse Amazon S3**.
3. Navigate to the bucket or folder containing your codebase and make a note of the URI.

   For more information, see Creating, configuring, and working with Amazon S3 buckets and Access control best practices in the *Amazon S3 User Guide*.
4. Paste the URL into the field labeled **Enter Amazon S3 URI**.

Before you create your customization, you have the option of adding tags to it.

To learn more about tags, see the Tagging your AWS resources User Guide.

After following the procedures above, choose **Create customization**.

Customizations and your data

CodeWhisperer customizations use your content to present suggestions to you in the style of your organization's developers.

However, AWS will not store or use your content in any context that does not directly serve your enterprise.

AWS will not use your content to provide code suggestions to other customers.

CodeWhisperer will not reference security scans for other customers (or for you).

Troubleshooting the creation of your customization

- You may receive the error: `Total size of the provided repositories exceeds the maximum allowed size of *number* for a customization.`

  In that case, remove a repository from your data source and try again.
- You may receive the error: `Insufficient data to create a customization. Add more files from supported languages and retry.`

  In order for code written in a particular language to be used to create a customization, there must be at least 10 files containing code in that language in your data source. The total size of all code files (containing one or more languages) in your data source must be at least 20 MB.

  Some files, even if they are in the relevant language, will not count toward the 20 MB. For example, duplicate files and files in an unsupported format will not be counted.

  If you receive this error, add more files containing the programming language that is the focus of your customization, and try again.

Deleting your customization

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

This section explains how to delete a customization with CodeWhisperer.

> ⚠ **Warning**
>
> Deleting a customization will delete all versions associated with the resource.

To delete your customization, follow this procedure:

1. Open the CodeWhisperer console.
2. From the navigation pane on the left, choose **Customizations**.
3. The customizations page will appear.
4. If the customization that you want to delete is still active, choose **Deactivate**.
5. Choose **Delete**.

> ℹ **Note**
>
> You can also delete a customization from the page that gives the details of that customization.
> To do that, just choose **Delete** from the upper right corner of the customization detail page.

If you are transitioning to Amazon Q Developer, your next step is deactivating the CodeWhisperer application.

Evaluating and optimizing your customization

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

Evaluating your customization

This section explains how to evaluate your customization.

1. In the CodeWhisperer console, from the navigation panel, choose **Customizations**.

2. Choose the name of the customization to examine.
3. The right side of the window will display an evaluation score. This score indicates CodeWhisperer's evaluation of how effective your customization may be.

With your evaluation score in mind, you must now consider whether or not to activate your customization. In making this decision, take the following factors into consideration.

- **Very good 8-10:** CodeWhisperer recommends that you activate this customization.
- **Fair 5-7:** CodeWhisperer recommends that you activate this customization.

  If you do not see significant improvement, consider the optimization suggestions below. If those are not effective, consider switching to a different code source.
- **Poor 1-4:** This customization is not likely to be useful. Consider the optimization suggestions below. If those are not effective, consider switching to a different code source.

Optimizing your customization

This section contains suggestions for optimizing your suggestion in order to achieve a higher evaluation score.

- Consider expanding your data source to include more code repositories.
- If you primarily included data from limited programming languages, consider expanding to more languages.
- Remove auto-generated files and repositories, or those generated from templates. Training a customization to generate or complete such files is typically not valuable, and tends to just add noise.

  > **Note**
  >
  > CodeWhisperer automatically filters out non-code files, such as configuration files and text files.
- It is possible that your codebase does not frequently use internal libraries. If you know this to be true, then the core CodeWhisperer model may already have been performing as well as possible.

Optimizing for the languages you use

In order for code in a particular language to be used in a customization, you must include at least 10 data files containing that language, and all of your source files together must come to at least 20 MB. If your developers write code in a language that is not supported by your customization, CodeWhisperer's recommendations in that language will come from the CodeWhisperer base model (not your customization). In other words, they will be the same recommendations that you would receive if you did not have a customization. This, in turn, could affect the metrics on your dashboard. For example, the "Lines of code generated by CodeWhisperer" may be less than what it would have been if the language commonly used by your developers had been included in your customization.

Logging and troubleshooting

Setting up log delivery

CodeWhisperer can provide you with log files that will help you understand and troubleshoot issues with your customization.

You can have your log files sent to a Amazon CloudWatch Logs. group, an Amazon S3 bucket, an Amazon Data Firehose, or any combination.

To set up log delivery, select the Log deliveries tab on the console page for your customization. Follow the instructions in the interface to configure your log deliveries. Then choose **Create log deliveries**.

The prefix of logs delivered to an Amazon S3 bucket will be: `AWSLogs/account_id/codeWhispererCustomizationLogs/region/customization_id/year/month/day/hour/`

The files will be zipped, with the naming format: `account_id_codeWhispererCustomizationLogs_customization_id_date_file_id.log.gz`

> ⚠ **Warning**
>
> In order to get the most use out of customization logs, it's best to set up log delivery within five minutes of creating the customization.

To learn more about the permissions required to delivery logs to multiple resources, see Logging that requires additional permissions [V2] in the *Amazon CloudWatch Logs User Guide*.

Understanding customization-related log messages

The following table lists log messages that may help you understand issues with your customization.

Log message	Log level
Starting to ingest *number* repos from source *source*	Info
Downloading data from repo: *repo name*	Info
Received *amount* MB of supported data. *amount* MB required. Add more data and retry.	Error
The provided CodeStar Connection ARN: *Arn* is invalid.	Error
Access denied when attempting to reach the provided CodeStar Connection: *Arn*	Error
Failed to download with AWS CodeStar Connection: *Arn* probably deleted by customer	Error
ProviderThrottlingException from CodeStar Connection: *Arn* while cloning repository: *repository*	Error
Processing data from S3: *S3 URI*	Info
Invalid S3 path specified: *S3 Directory*	Error
Unable to access the provided S3 bucket: *bucket name*	Error

Log message	Log level
`The provided S3 bucket:` *`bucket name`* `does not exist.`	Error
`The provided S3 key` *`S3 URI`* `does not exist.`	Error
`Failed to ingest` *`number of failed repos / total number of repos`* `repositories`	Error
`Unable to process repository:` *`repo name`*`, with a size of` *`repo size`* `GB, exceeds the limit of` *`max size`* `GB.`	Warn
`Unable to process file:` *`file name`*`, with a size of` *`file size`*`, which exceeds the limit of` *`max file size`* `MB`	Error
`Unable to process collection:` *`collection name`* `, with total size of` *`total repo size`* `MB, which exceeds the limit of` *`max total repo size`* `MB`	Error
`The following languages will be used for customization:` *`list of languages`* `. Languages may be excluded from customization if they are not sufficiently represented in your files.`	Info

Understanding customization-related error messages in the console

The following table will help you understand customization-related messages in the CodeWhisperer console.

Error message	Suggested action
You have activated the maximum number of customizations.	Deactivate an active customization and try again.
You have exceeded the maximum number of group permissions limit of *`limit`*.	Remove a group and retry.
You have exceeded the maximum number of user permissions limit of *`limit`*.	Remove a user and retry.
Maximum active jobs reached.	Wait until an in-progress job in the same account has finished. Retry the operation.
Encountered an unexpected error when processing the request.	Retry the operation. If it continues to fail, contact customer support.
Access denied when attempting to reach the provided AWS CodeStar connection.	Validate permissions on your connection and on your third-party provider. Then retry the operation.
One or more repositories not found while accessing the provided AWS CodeStar connection.	Validate permissions and list of repos from the third-party provider. Then retry the operation.
The provided AWS CodeStar connection ARN is invalid.	Update the customization with a corrected Connection ARN.
The Host associated with the provided AWS CodeStar connection is unavailable.	Try again in 5 minutes.
Invalid Amazon S3 path specified.	Update the customization with a valid Amazon S3 URI.
Unable to access the provided Amazon S3 bucket.	Validate permissions for the admin's role. Retry after fixing any permission issues.
The provided Amazon S3 bucket does not exist.	Update the customization with a valid Amazon S3 URI.

Error message	Suggested action
The provided Amazon S3 key does not exist.	Update the customization with a valid Amazon S3 URI.
Insufficient data to create a customization. Add more files from supported languages and retry.	Add more data to the same data source, and update the customization with the same reference.
Total size of the provided repositories exceeds the maximum allowed size of *`size`* for a customization.	Remove some data from the provided data source. Update the customization with the same reference.
You have created the maximum number of customizations. Delete an existing customiza tion and try again.	Delete the current customization and retry.
Customizations exist within the account. You must delete all customizations prior to deleting the profile.	Delete all customizations associated with the account and retry.

Activating your CodeWhisperer customizations

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

Activating a version

This section describes how to activate and deactivate a version of your customization.

You can activate a new version of a customization, even while developers from your organization are using the previous version. After you activate the new version, the developers will seamlessly begin using it, with no adjustments needed on the development side.

You can also roll your customization back to a previously active state. However, CodeWhisperer does not actually re-activate a previously activated version. Instead, it creates a new version by copying a previous version and then activating the copy.

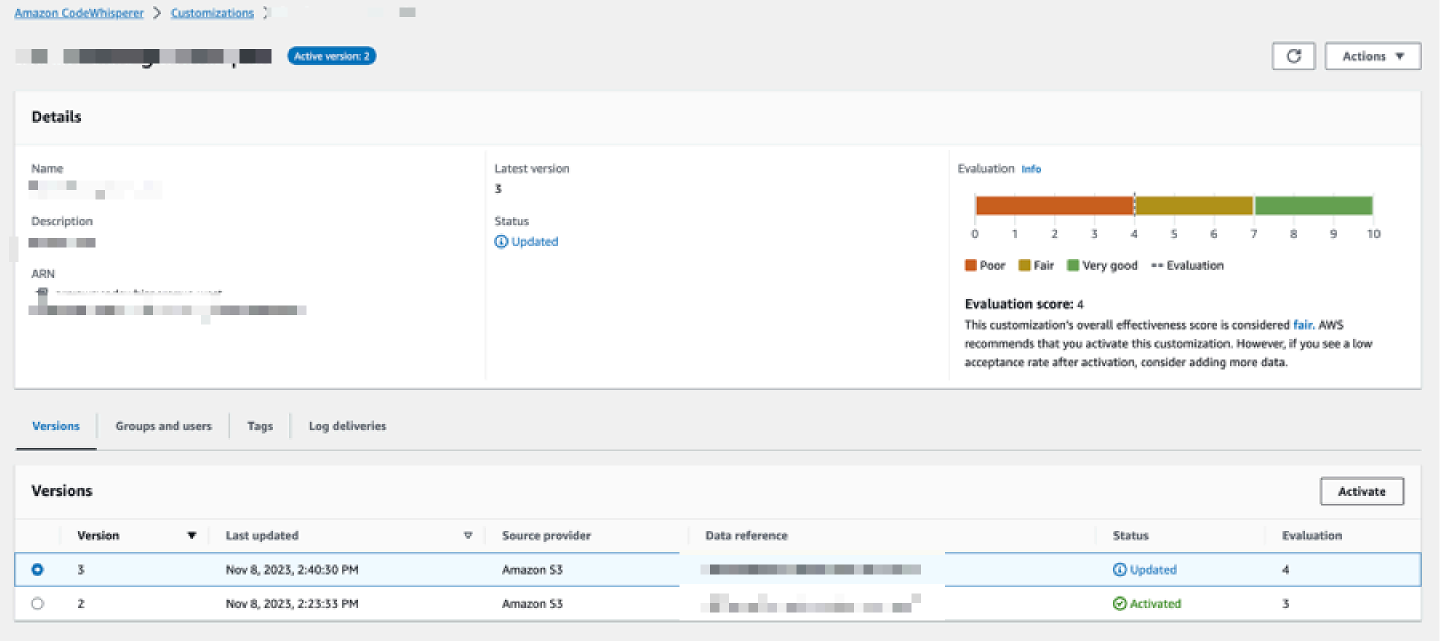
For example, suppose that you have three versions: 1, 2, and 3. The active version is 3. You decide to go back to version 1. But "re-activating" version 1 is actually just copying version 1 and creating version 4. That's the version you use: version 4, the new copy of the old version.

To activate a version of your customization, follow this procedure:

1. Open the CodeWhisperer console.
2. From the navigation pane on the left, choose **Customizations**.

   The customizations page will appear.
3. Choose the customization you want to activate a version for.

   The customization details page will appear.
4. Choose the version you want to activate from the **Versions** table.
5. Choose **Activate**.



[redacted]

Details

Name
[redacted]

Description
[redacted]

ARN
[redacted]

Latest version
3

Status
Updated

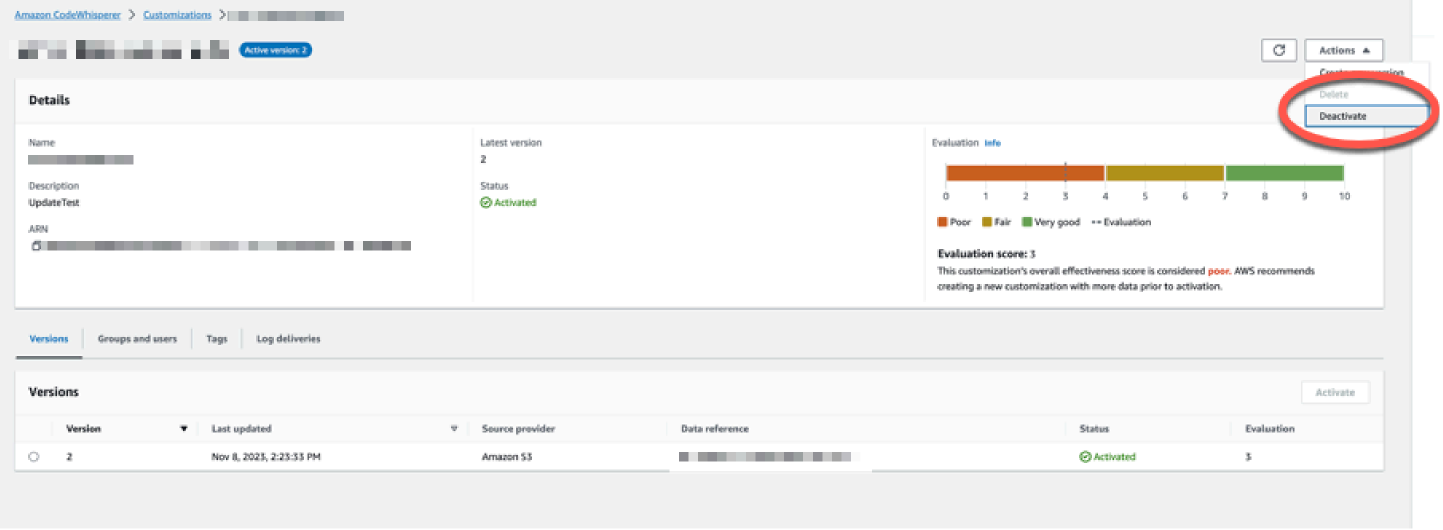
Evaluation Info

Evaluation score: 4
This customization's overall effectiveness score is considered fair. AWS recommends that you activate this customization. However, if you see a low acceptance rate after activation, consider adding more data.

Versions

	Version ▼	Last updated ▽	Source provider	Data reference	Status	Evaluation
●	3	Nov 8, 2023, 2:40:30 PM	Amazon S3	[redacted]	Updated	4
○	2	Nov 8, 2023, 2:23:33 PM	Amazon S3	[redacted]	Activated	3

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


Amazon CodeWhisperer > Customizations > [redacted]
[redacted] Active version: 2
Actions ▲
Delete
Deactivate
Versions
Groups and users
Tags
Log deliveries
Activate
0 1 2 3 4 5 6 7 8 9 10
🟥 Poor 🟨 Fair 🟩 Very good -- Evaluation

Details

Name
[redacted]

Description
UpdateTest

ARN
[redacted]

Latest version
2

Status
Activated

Evaluation Info

Evaluation score: 3

This customization's overall effectiveness score is considered poor. AWS recommends creating a new customization with more data prior to activation.

Versions

	Version	Last updated	Source provider	Data reference	Status	Evaluation
○	2	Nov 8, 2023, 2:23:33 PM	Amazon S3	[redacted]	Activated	3

Updating your CodeWhisperer customizations

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

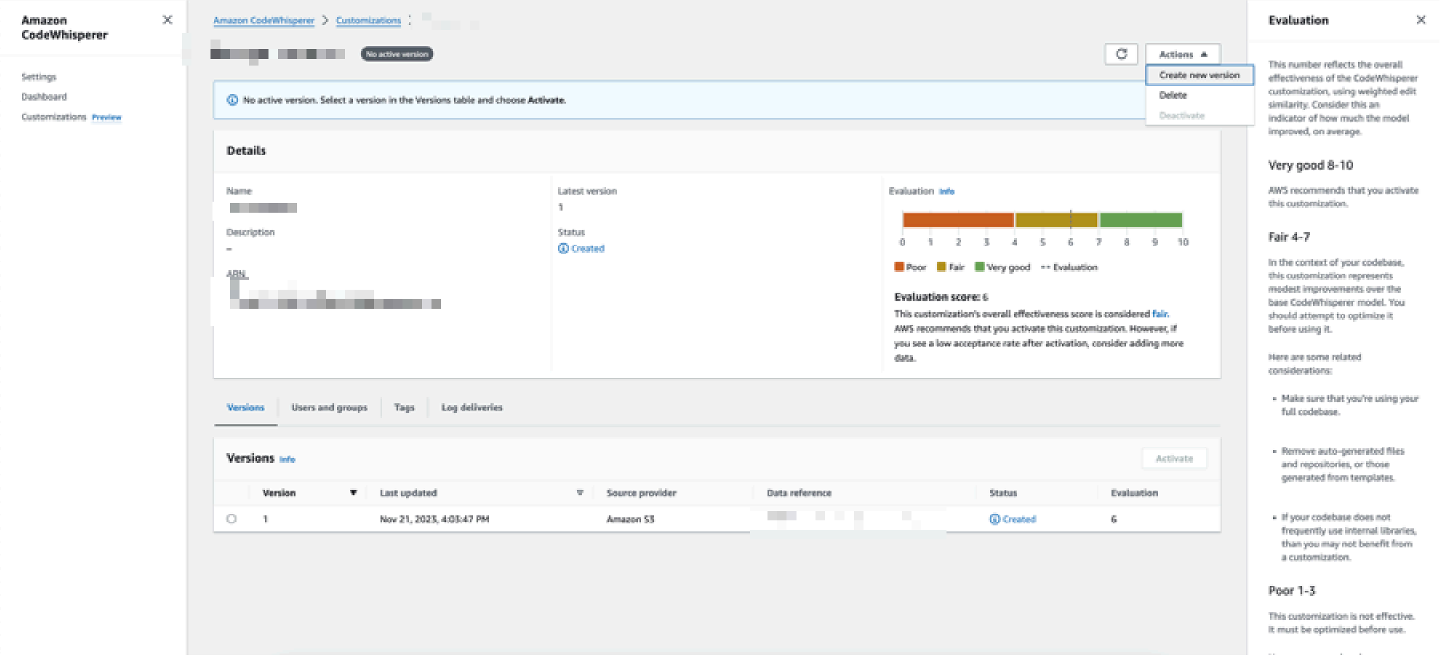
This section explains how to update a customization with CodeWhisperer.

A customization can have multiple versions.

CodeWhisperer administrators have access to a maximum of three versions for each customization:

- the latest version
- the currently active version
- the most recently active version that is not currently active

Creating a new version



[redacted]

No active version. Select a version in the Versions table and choose **Activate**.

Details

Name
[redacted]

Description
–

ARN
[redacted]

Latest version
1

Status
Created

Evaluation Info

Evaluation score: 6

This customization's overall effectiveness score is considered fair. AWS recommends that you activate this customization. However, if you see a low acceptance rate after activation, consider adding more data.

Versions Info

	Version	Last updated	Source provider	Data reference	Status	Evaluation
○	1	Nov 21, 2023, 4:03:47 PM	Amazon S3	[redacted]	Created	6

Evaluation

This number reflects the overall effectiveness of the CodeWhisperer customization, using weighted edit similarity. Consider this an indicator of how much the model improved, on average.

Very good 8-10

AWS recommends that you activate this customization.

Fair 4-7

In the context of your codebase, this customization represents modest improvements over the base CodeWhisperer model. You should attempt to optimize it before using it.

Here are some related considerations:

- Make sure that you're using your full codebase.
- Remove auto-generated files and repositories, or those generated from templates.
- If your codebase does not frequently use internal libraries, than you may not benefit from a customization.

Poor 1-3

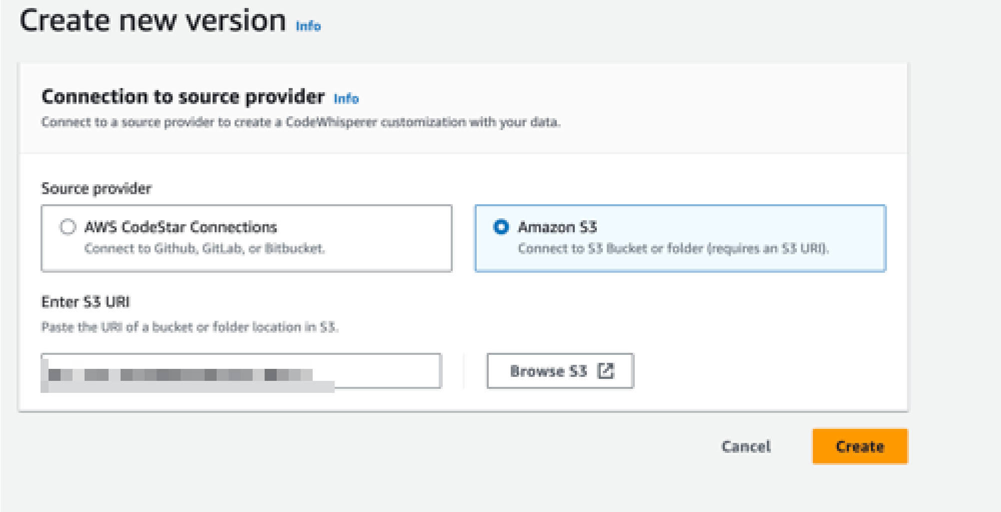
This customization is not effective. It must be optimized before use.

To create a new version of your customization, follow this procedure:

1. Open the CodeWhisperer console.
2. From the navigation pane on the left, choose **Customizations**.

   The customizations page will appear.
3. Choose the customization for which you want to create a new version.

   The customization details page will appear.
4. Select **Create new version** from the **Actions** dropdown.
5. If applicable, change the data source.



Create new version Info

Connection to source provider Info

Connect to a source provider to create a CodeWhisperer customization with your data.

Source provider

○ AWS CodeStar Connections
Connect to Github, GitLab, or Bitbucket.

● Amazon S3
Connect to S3 Bucket or folder (requires an S3 URI).

Enter S3 URI

Paste the URI of a bucket or folder location in S3.

[redacted]

6. Choose **Create**.

If you receive error messages, see Troubleshooting the creation of your customization.

Adding users and groups to your CodeWhisperer customizations

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

This section contains information about how to add users and groups to customizations.

> **Note**
>
> You must activate a customization before you can add users to it.

> **Note**
>
> You can only add a user or group to a customization if you have already added the user or group to your CodeWhisperer Professional profile. For more information, see Setting up Amazon CodeWhisperer for administrators

1. In the CodeWhisperer console, from the navigation panel, choose **Customizations**.
2. Choose the name of the customization to which you want to add users or groups.

3. In the bottom half of the window, if necessary, select the **Users and groups** tab. and then the **Users** or **Groups** sub-tab.
4. Select the users or groups that require access to your customization.
5. Choose **Add users** or **Add groups**.

Using CodeWhisperer customizations

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

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

CodeWhisperer only supports customizations in VS Code and JetBrains IDEs.

AWS Toolkit for Visual Studio Code

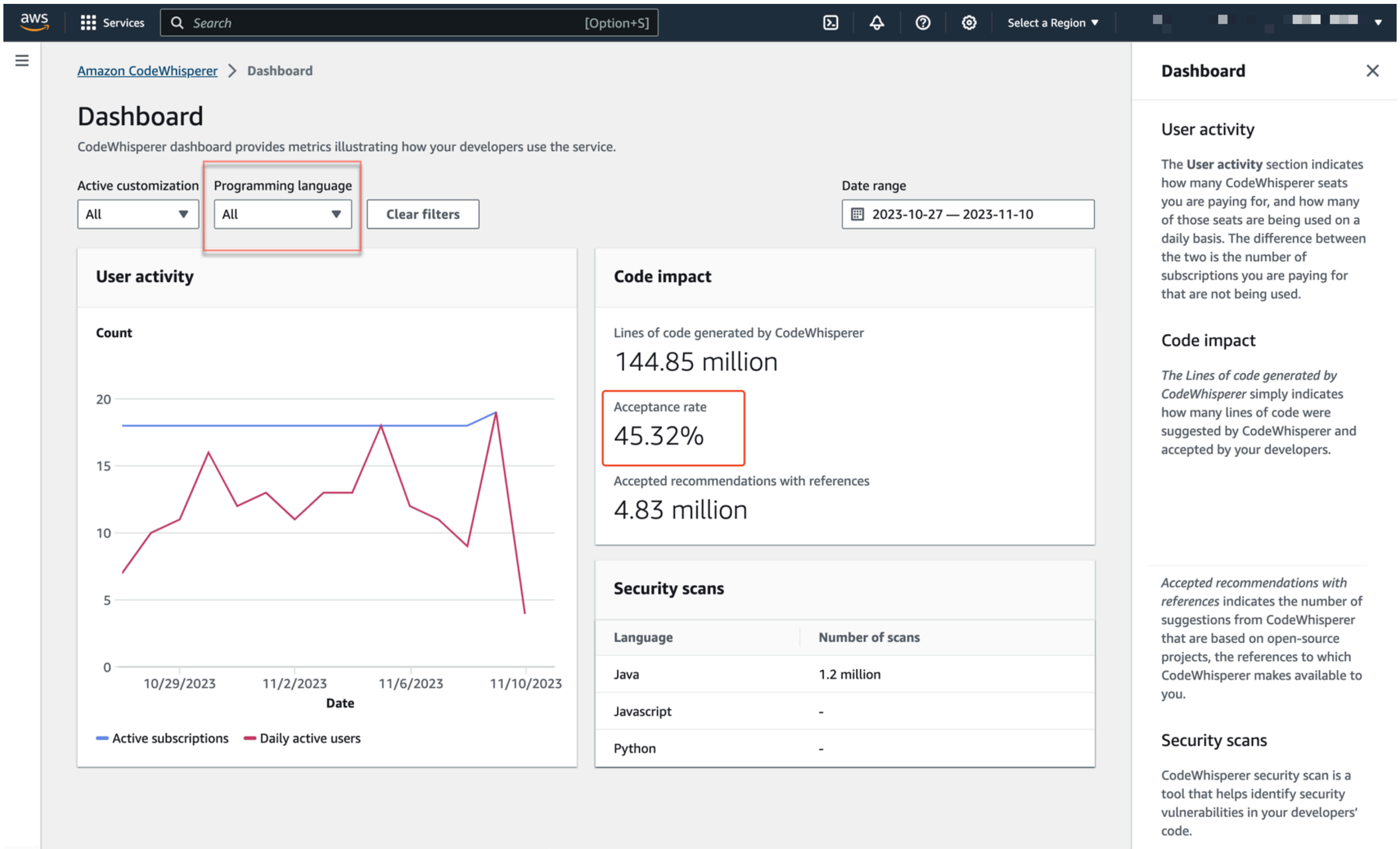
How to use customizations with VS Code.

1. Authenticate to CodeWhisperer Professional with IAM Identity Center using the steps under Getting started with CodeWhisperer in VS Code and JetBrains.
2. In the **Developer Tools** pane, under CodeWhisperer, choose **Select Customization**.
3. At the top of the window, from the dropdown menu, select the appropriate customization.

AWS Toolkit for JetBrains

1. Authenticate to CodeWhisperer Professional with IAM Identity Center using the steps under Getting started with CodeWhisperer in VS Code and JetBrains.
2. In the **Developer Tools** pane, under CodeWhisperer, choose **Select Customization**.
3. In the pop-up window, select the appropriate customization.
4. Choose **Connect**.

Dashboard



Amazon CodeWhisperer > Dashboard

Dashboard

CodeWhisperer dashboard provides metrics illustrating how your developers use the service.

Active customization: All ▼

Programming language: All ▼

Clear filters

Date range: 2023-10-27 — 2023-11-10

User activity

Code impact

Lines of code generated by CodeWhisperer
144.85 million

Acceptance rate
45.32%

Accepted recommendations with references
4.83 million

Security scans

Language	Number of scans
Java	1.2 million
Javascript	-
Python	-

Dashboard ✕

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.

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.

Available only for administrators, and only at the Professional tier, the CodeWhisperer dashboard summarizes useful data about how your developers use the service. Among the useful metrics is the acceptance rate, which indicates how often you take CodeWhisperer's suggestions.

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

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

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.

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.

If you use CodeWhisperer very little over a two-week period, the **Code impact** section will be affected as follows:

- If no recommendations are invoked for two weeks, then no data will appear in the **Code impact** section.
- If recommendations are invoked, but none are accepted or rejected, then no data will appear in the **Code impact** section.
- If recommendations are invoked, and none are accepted, but some are rejected, then the **Acceptance rate** (0%) will be displayed, but no data will appear for **Lines of code generated by CodeWhisperer** or **Accepted recommendations with references**.

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

Action	Keyboard shortcut
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 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

Action	Keyboard shortcut
Reject a recommendation	ESC

Toolkit for Visual Studio

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

See also Microsoft's Visual Studio default keyboard shortcuts.

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

AWS Toolkit for Visual Studio Code

Action	Keyboard shortcut
Manually trigger 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 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 Windows: Alt + C
Accept a recommendation	Tab
Next recommendation	Right arrow
Previous recommendation	Left arrow

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

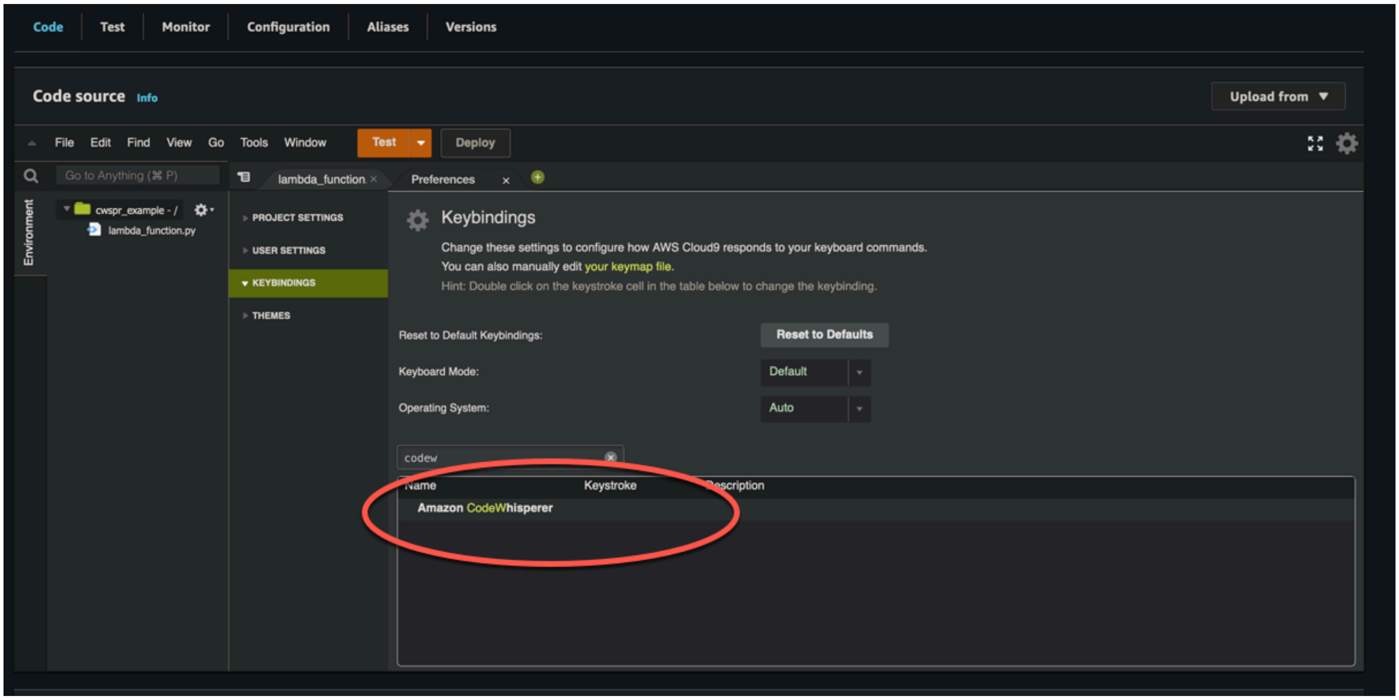
To change keybindings in IntelliJ, see IntelliJ IDEA keyboard shortcuts on the JetBrains website.

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.



Keybindings

Change these settings to configure how AWS Cloud9 responds to your keyboard commands.
You can also manually edit your keymap file.
Hint: Double click on the keystroke cell in the table below to change the keybinding.

Reset to Default Keybindings: Reset to Defaults

Keyboard Mode: Default

Operating System: Auto

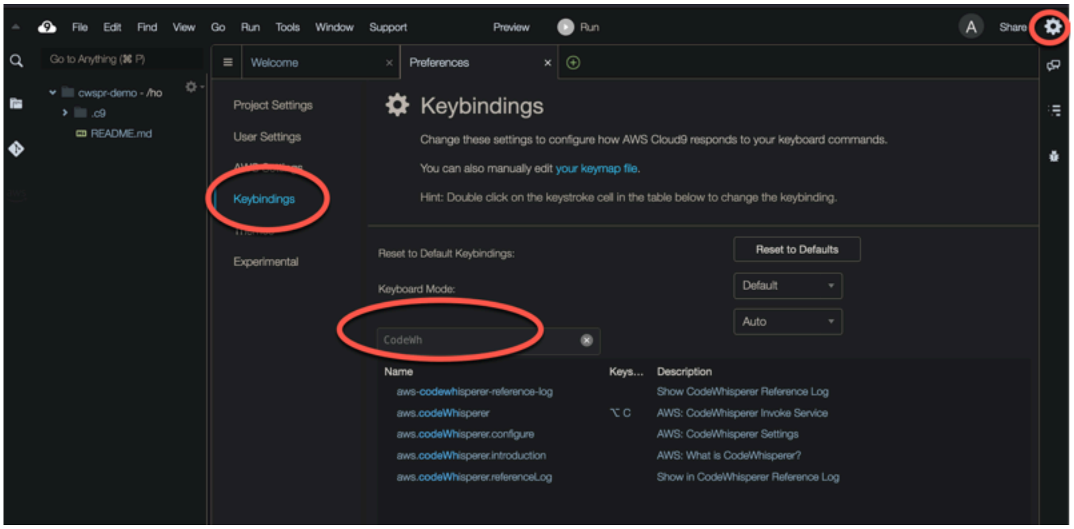
codew

Name	Keystroke	Description
Amazon 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.



Keybindings

Change these settings to configure how AWS Cloud9 responds to your keyboard commands.

You can also manually edit your keymap file.

Hint: Double click on the keystroke cell in the table below to change the keybinding.

Reset to Default Keybindings: Reset to Defaults

Keyboard Mode: Default

Auto

CodeWh

Name	Keys...	Description
aws-codewhisperer-reference-log		Show CodeWhisperer Reference Log
aws.codeWhisperer	⌥ C	AWS: CodeWhisperer Invoke Service
aws.codeWhisperer.configure		AWS: CodeWhisperer Settings
aws.codeWhisperer.introduction		AWS: What is CodeWhisperer?
aws.codeWhisperer.referenceLog		Show in CodeWhisperer Reference Log

Language support in Amazon CodeWhisperer

Language support in Amazon CodeWhisperer

CodeWhisperer supports code generation for multiple programming languages. The accuracy and quality of the code generation for a programming language depends on the size and quality of the training data.

In terms of the quality of the training data, the programming languages with the most support are:

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

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

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

CodeWhisperer also supports code generation for:

- Ruby
- C++
- C
- Shell
- Scala

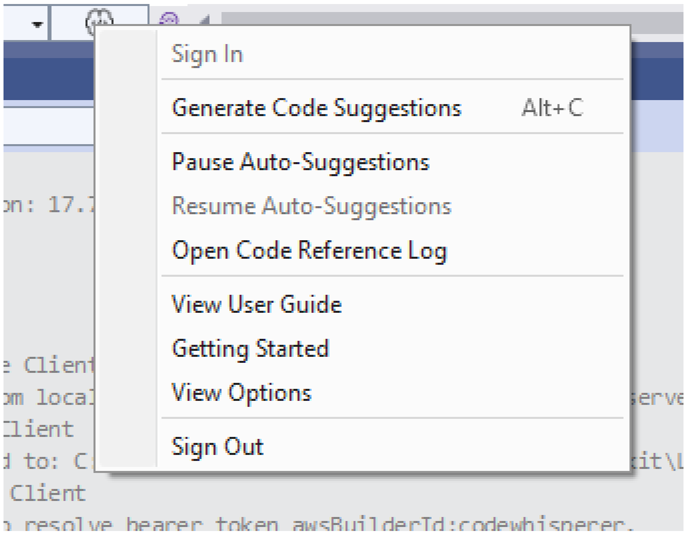
For a list of supported coding environments, refer to Getting started.

Pausing suggestions with Amazon CodeWhisperer

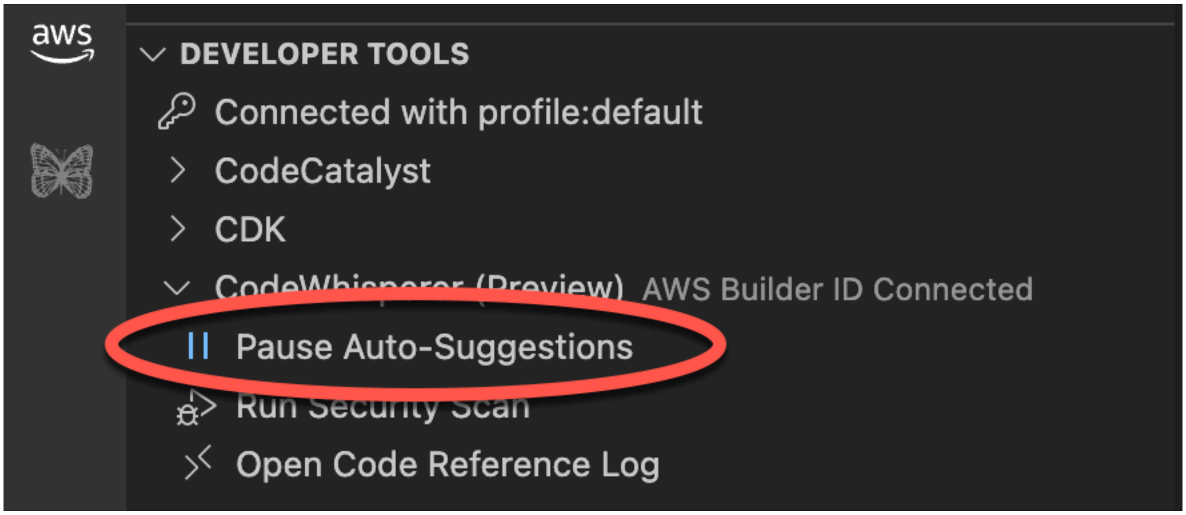
This chapter describes how to pause and resume automatic suggestions in CodeWhisperer.

Visual Studio

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

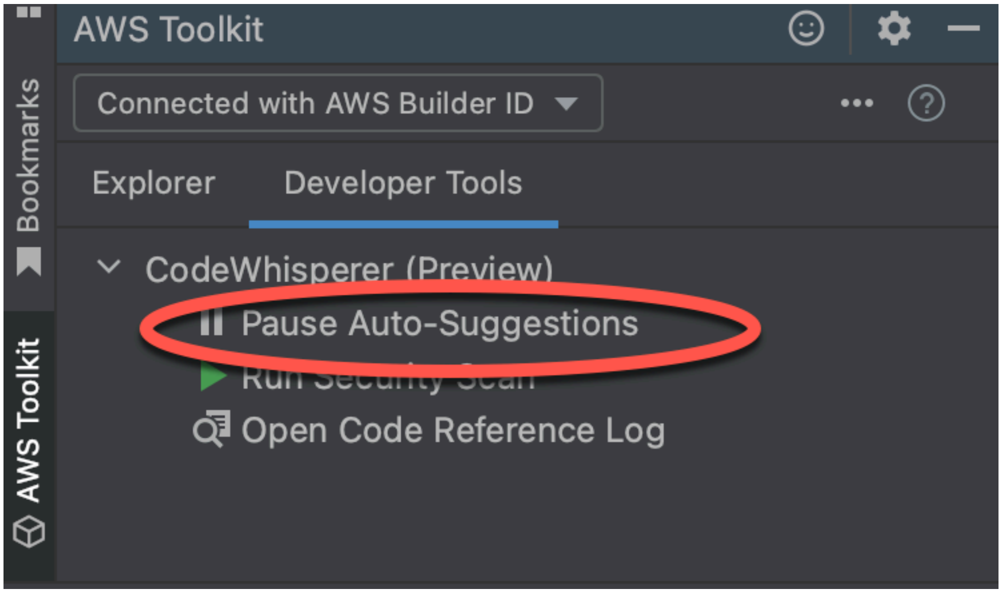


VS Code



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



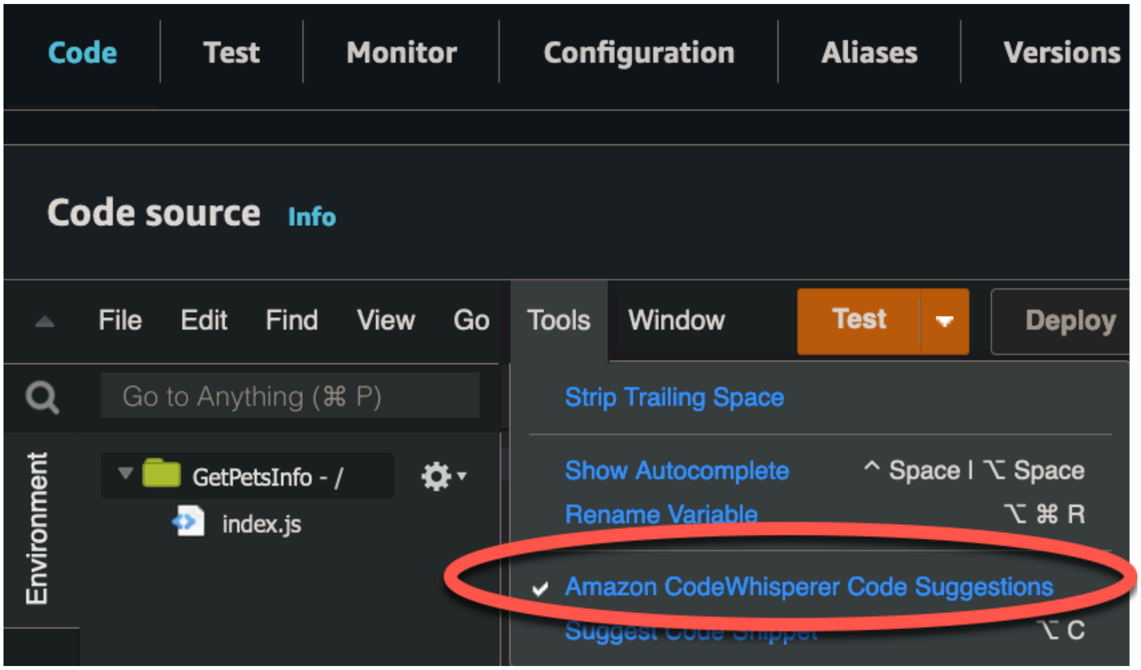
1. In JetBrains, choose the AWS logo from the left sidebar.
2. In the AWS Toolkit window, select the **Developer Tools** tab.
3. Expand the CodeWhisperer section.
4. Choose **Pause Auto-Suggestions** or **Resume Auto-Suggestions**.

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

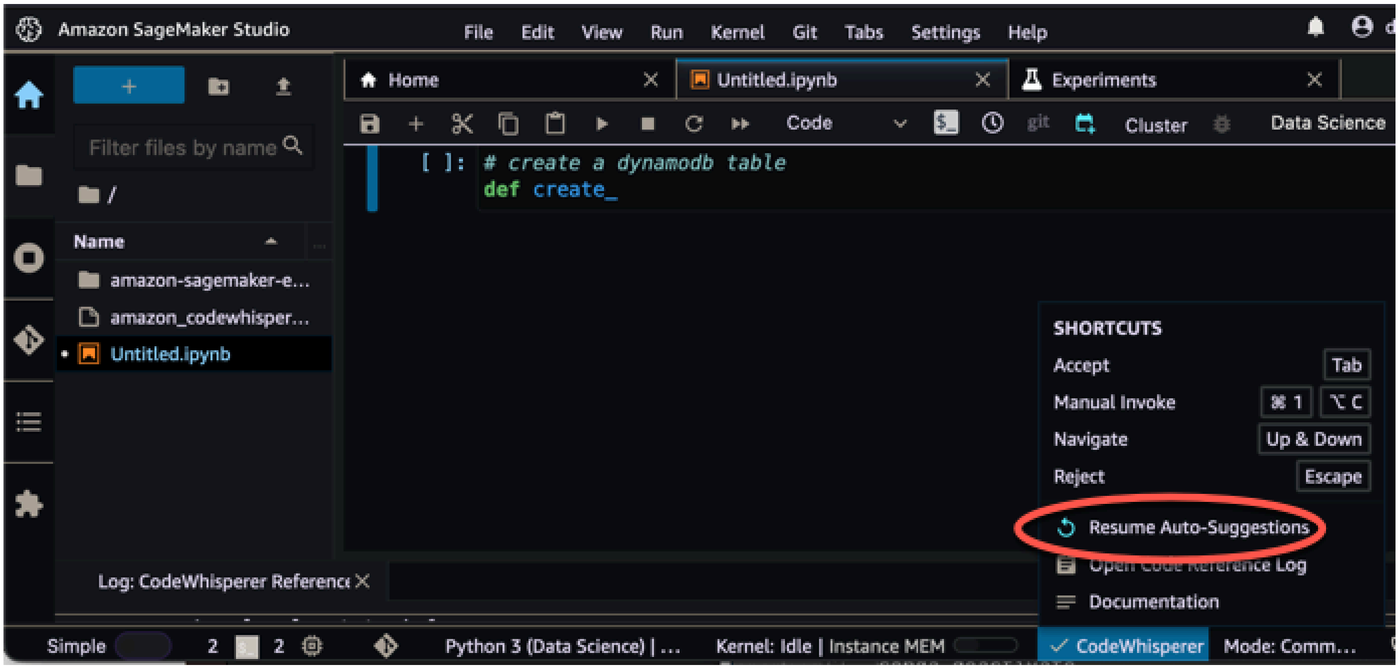
Lambda



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



```
[ ]: # create a dynamodb table
     def create_
```

SHORTCUTS

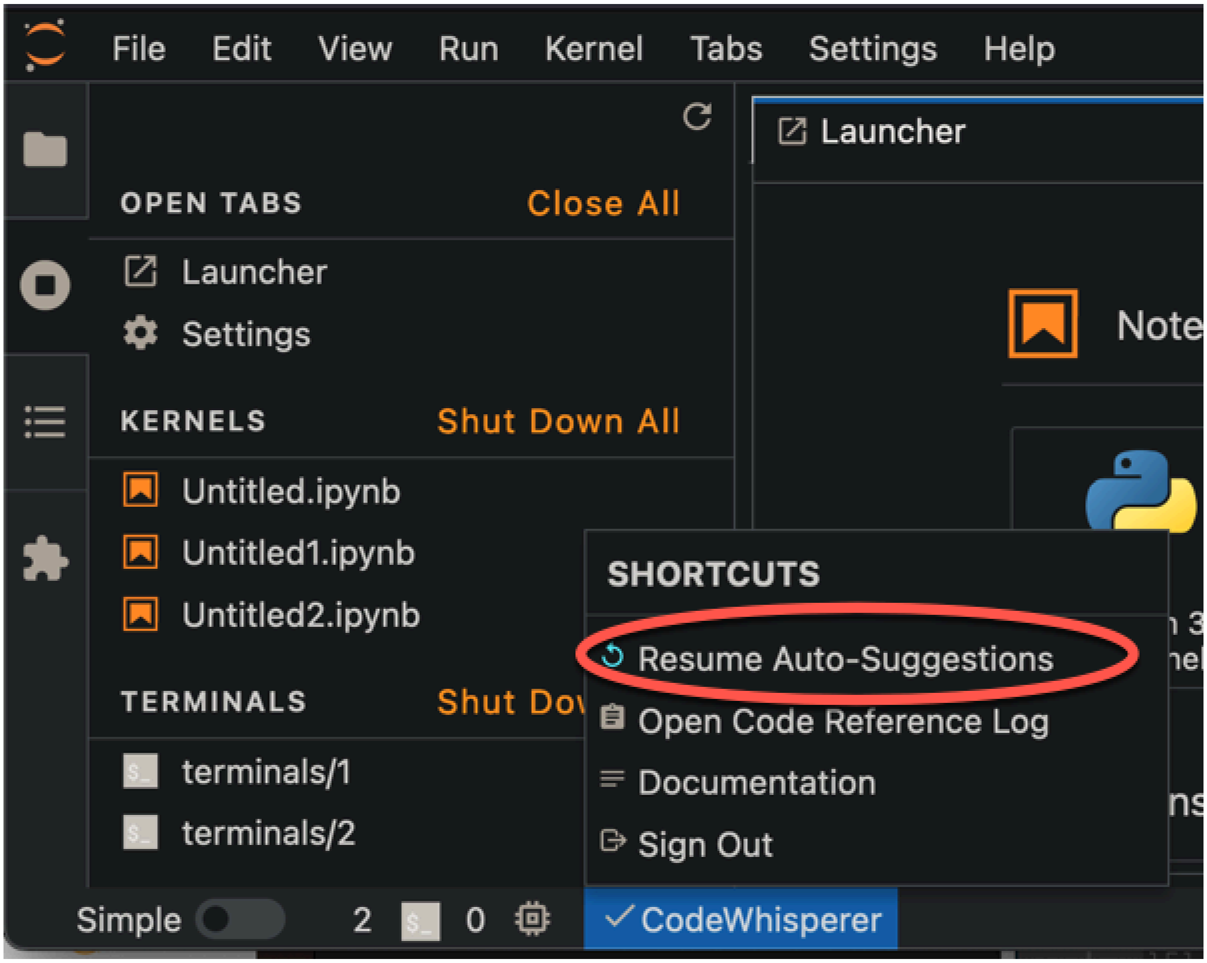
Accept	Tab
Manual Invoke	⌘ 1 ⌥ C
Navigate	Up & Down
Reject	Escape

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

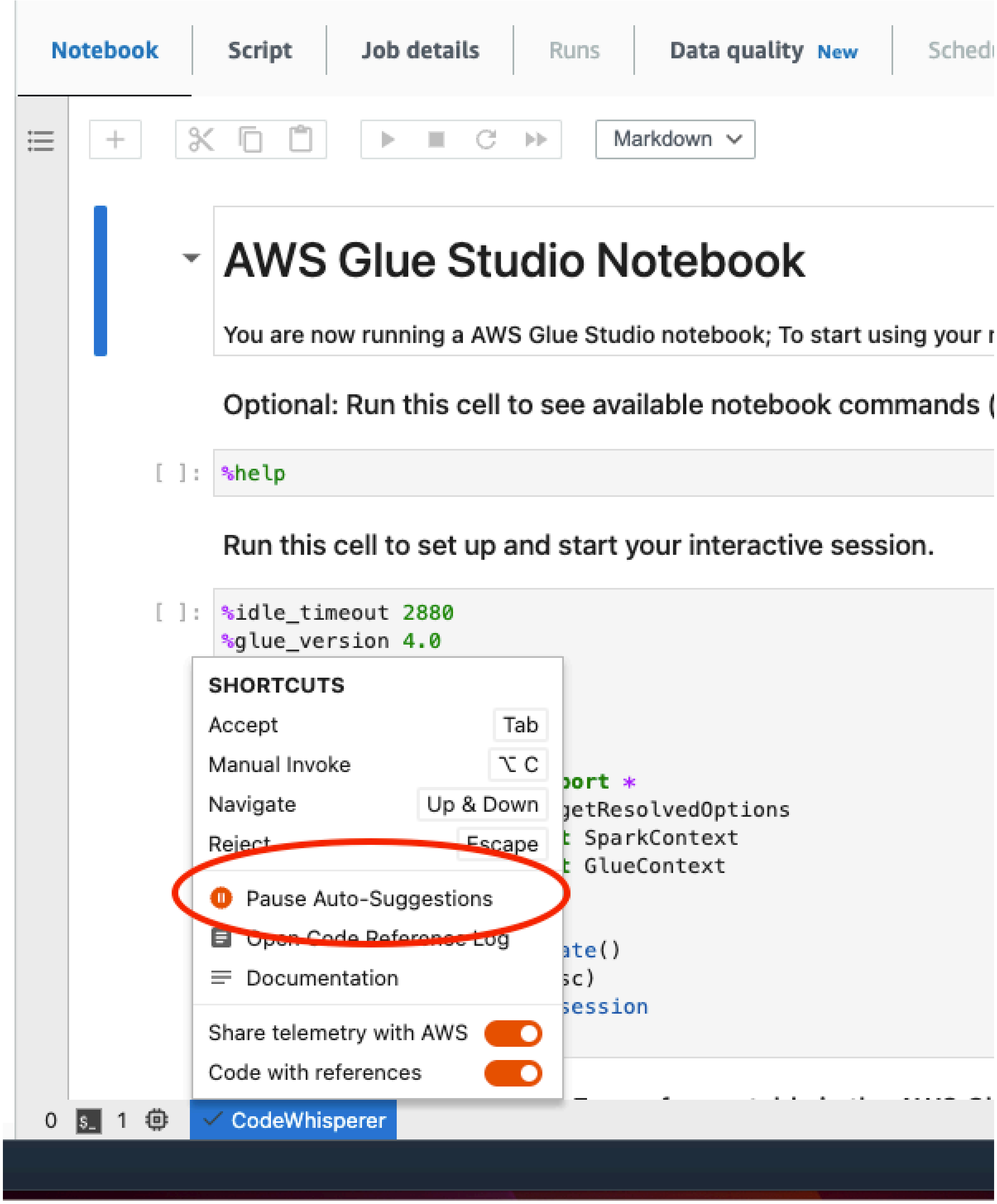


1. In the JupyterLab console, choose CodeWhisperer from the bottom of the window.

   The CodeWhisperer panel will open.

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

AWS Glue Studio Notebook



AWS Glue Studio Notebook

You are now running a AWS Glue Studio notebook; To start using your r

Optional: Run this cell to see available notebook commands (

```
%help
```

Run this cell to set up and start your interactive session.

```
%idle_timeout 2880
%glue_version 4.0
```

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

You can use CodeWhisperer to detect security policy violations and vulnerabilities in your code with static application security testing (SAST), secrets detection, and infrastructure as code (IaC) scanning. Security scans in CodeWhisperer identify security vulnerabilities and suggest how to improve your code. In some cases, CodeWhisperer provides code you can use to address those vulnerabilities.

Run Security Scan performs a security scan on the 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.

Security scans operate at the project level, analyzing files within a user's local project or workspace and then truncating them to create a payload for transmission to the server side. This payload has a size limit that differs per programming language.

CodeWhisperer's security scan is powered by detectors from the Amazon CodeGuru Detector Library. CodeGuru Security does multiple layers of filtering before scanning code to ensure that you can focus on the most critical issues. As part of that, CodeGuru Security filters unsupported languages, test code, and open source code, before scanning for security issues.

Topics

Languages that security scans work with

The CodeWhisperer security scan feature supports the following language versions:

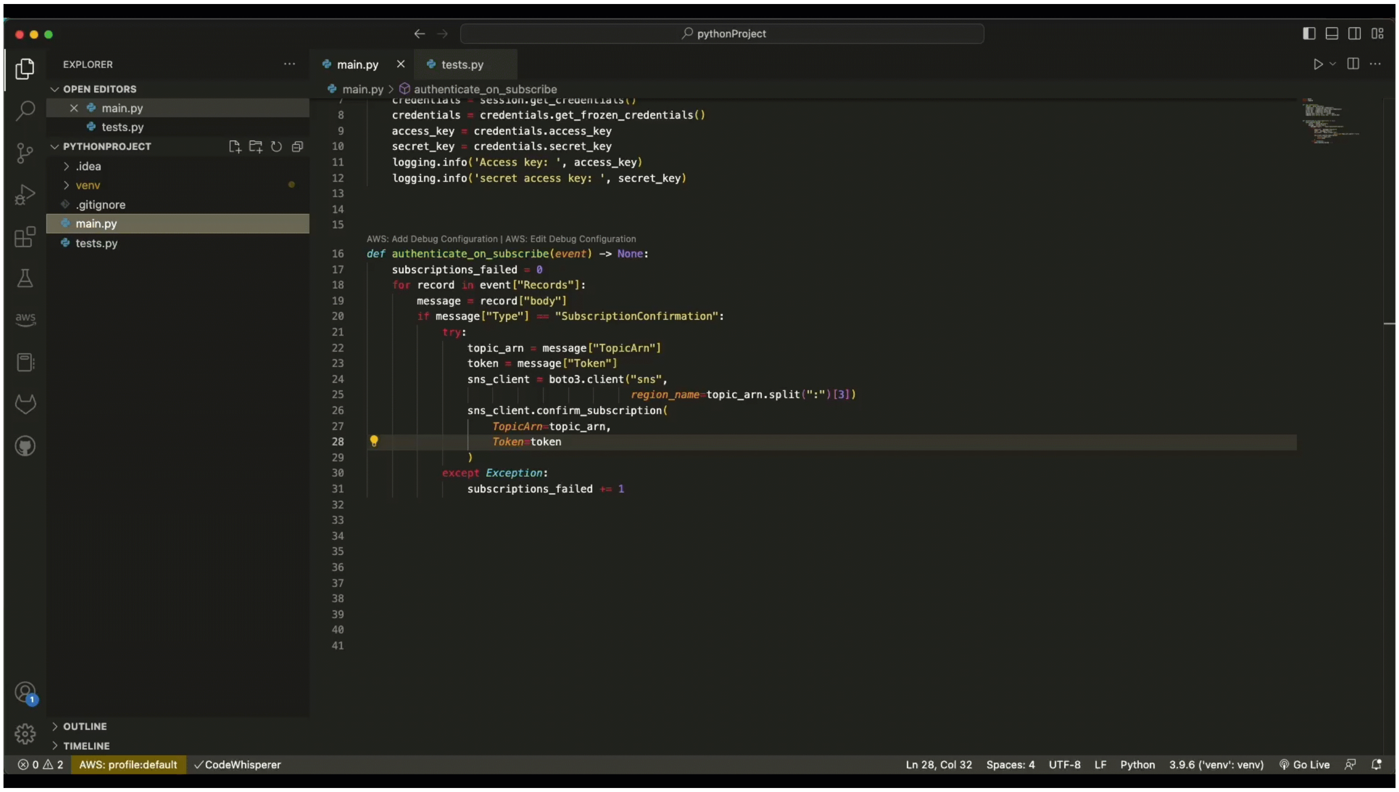
- Java - Java 17 and earlier
- JavaScript - ECMAScript 2021 and earlier
- Python - Python 3.11 and earlier, within the Python 3 series
- C# - All versions (.Net 6.0 and later recommended)

- TypeScript - All versions
- Ruby - Ruby 2.7 and 3.2
- Go - Go 1.18
- Infrastructure as Code (IaC) languages
    - AWS CloudFormation
    - Terraform - 1.6.2 and earlier
    - AWS CDK - TypeScript and Python

CodeWhisperer will only provide code remediation suggestions for code written in Java, Python, or JavaScript,

Running security scans

AWS Toolkit for Visual Studio Code



```
    credentials = session.get_credentials()
    credentials = credentials.get_frozen_credentials()
    access_key = credentials.access_key
    secret_key = credentials.secret_key
    logging.info('Access key: ', access_key)
    logging.info('secret access key: ', secret_key)



def authenticate_on_subscribe(event) -> None:
    subscriptions_failed = 0
    for record in event["Records"]:
        message = record["body"]
        if message["Type"] == "SubscriptionConfirmation":
            try:
                topic_arn = message["TopicArn"]
                token = message["Token"]
                sns_client = boto3.client("sns",
                                          region_name=topic_arn.split(":")[3])
                sns_client.confirm_subscription(
                    TopicArn=topic_arn,
                    Token=token
                )
            except Exception:
                subscriptions_failed += 1
```

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

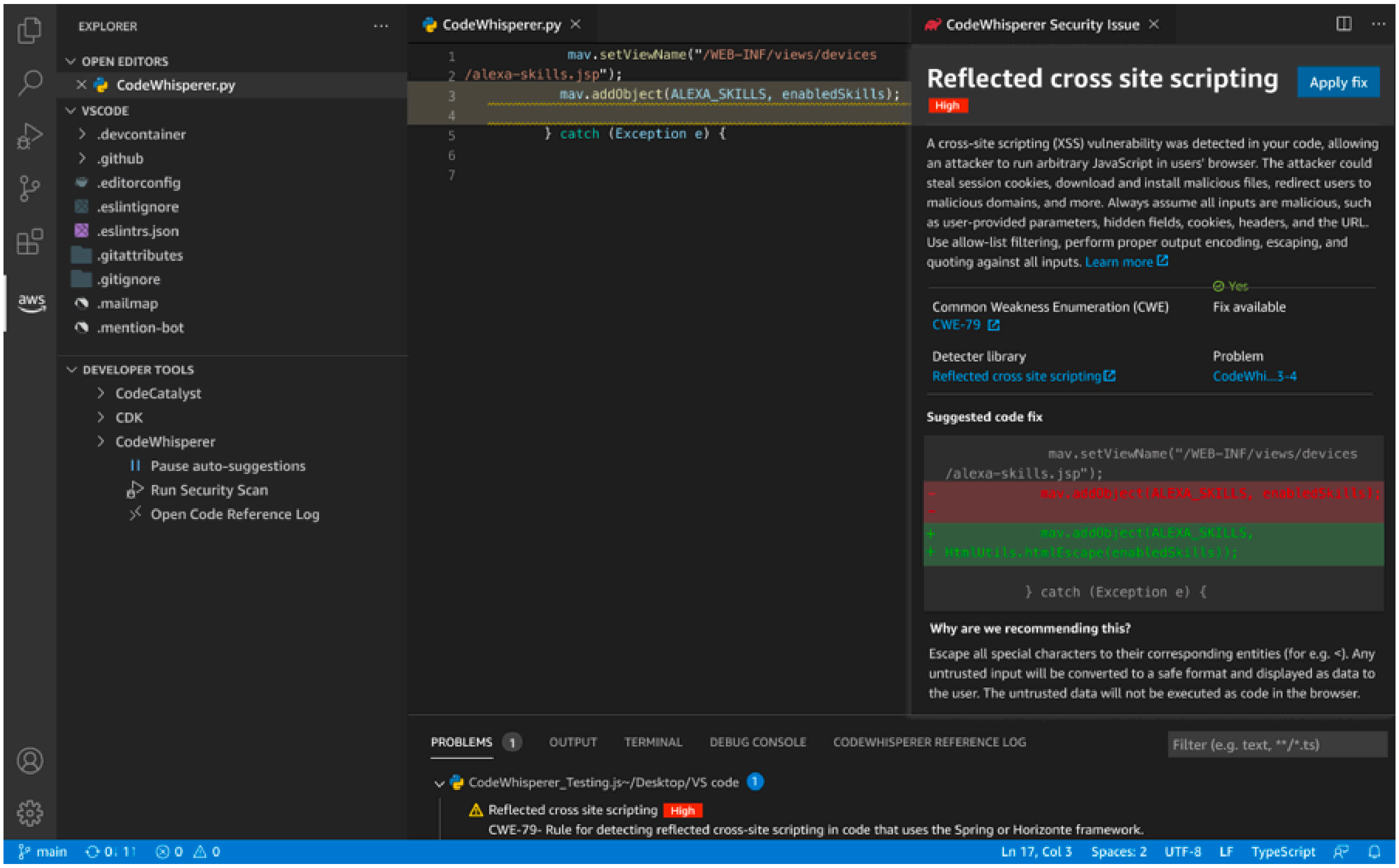
1. In VS Code, choose the AWS logo on the left side of the window. The AWS Toolkit panel will open.

2. In the AWS Toolkit panel, under **Developer Tools**, under **CodeWhisperer**, choose **Run Security Scan**.
3. After creating a scan, you can view findings in the **Problems** tab.

   To view information about the finding and suggested fix, hold your cursor over the underlined code.
4. If your code is written in Java, Python, or JavaScript, CodeWhisperer might provide a suggested code fix.

   - If it does provide a fix, and you want to implement that fix, choose **Apply fix**. The information about the finding will disappear.
   - If it does not provide a fix, update your code according to the information provided.

     Run another security scan to verify that the vulnerability was remediated.



```
               mav.setViewName("/WEB-INF/views/devices
/alexa-skills.jsp");
              mav.addObject(ALEXA_SKILLS, enabledSkills);

          } catch (Exception e) {
```

Reflected cross site scripting Apply fix

High

A cross-site scripting (XSS) vulnerability was detected in your code, allowing an attacker to run arbitrary JavaScript in users' browser. The attacker could steal session cookies, download and install malicious files, redirect users to malicious domains, and more. Always assume all inputs are malicious, such as user-provided parameters, hidden fields, cookies, headers, and the URL. Use allow-list filtering, perform proper output encoding, escaping, and quoting against all inputs. Learn more

Common Weakness Enumeration (CWE): CWE-79

Fix available: Yes

Detecter library: Reflected cross site scripting

Problem: CodeWhi...3-4

Suggested code fix

```
              mav.setViewName("/WEB-INF/views/devices
 /alexa-skills.jsp");
-             mav.addObject(ALEXA_SKILLS, enabledSkills);
-
+             mav.addObject(ALEXA_SKILLS,
+ HtmlUtils.htmlEscape(enabledSkills));

          } catch (Exception e) {
```

Why are we recommending this?

Escape all special characters to their corresponding entities (for e.g. <). Any untrusted input will be converted to a safe format and displayed as data to the user. The untrusted data will not be executed as code in the browser.

CodeWhisperer_Testing.js~/Desktop/VS code 1

⚠ Reflected cross site scripting High

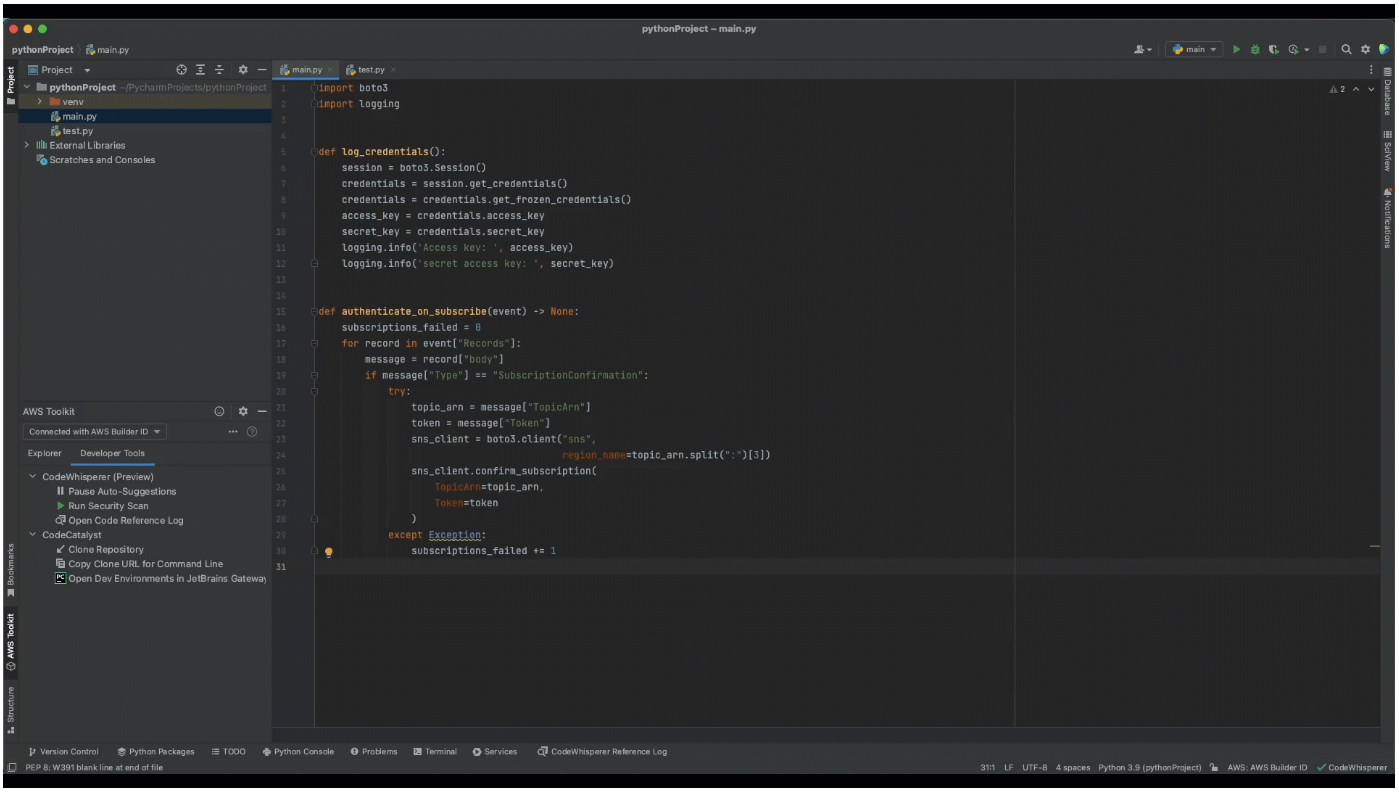
CWE-79- Rule for detecting reflected cross-site scripting in code that uses the Spring or Horizonte framework.

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.

> **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



```
import boto3
import logging


def log_credentials():
    session = boto3.Session()
    credentials = session.get_credentials()
    credentials = credentials.get_frozen_credentials()
    access_key = credentials.access_key
    secret_key = credentials.secret_key
    logging.info('Access key: ', access_key)
    logging.info('secret access key: ', secret_key)


def authenticate_on_subscribe(event) -> None:
    subscriptions_failed = 0
    for record in event["Records"]:
        message = record["body"]
        if message["Type"] == "SubscriptionConfirmation":
            try:
                topic_arn = message["TopicArn"]
                token = message["Token"]
                sns_client = boto3.client("sns",
                                          region_name=topic_arn.split(":")[3])
                sns_client.confirm_subscription(
                    TopicArn=topic_arn,
                    Token=token
                )
            except Exception:
                subscriptions_failed += 1

```

PEP 8: W391 blank line at end of file

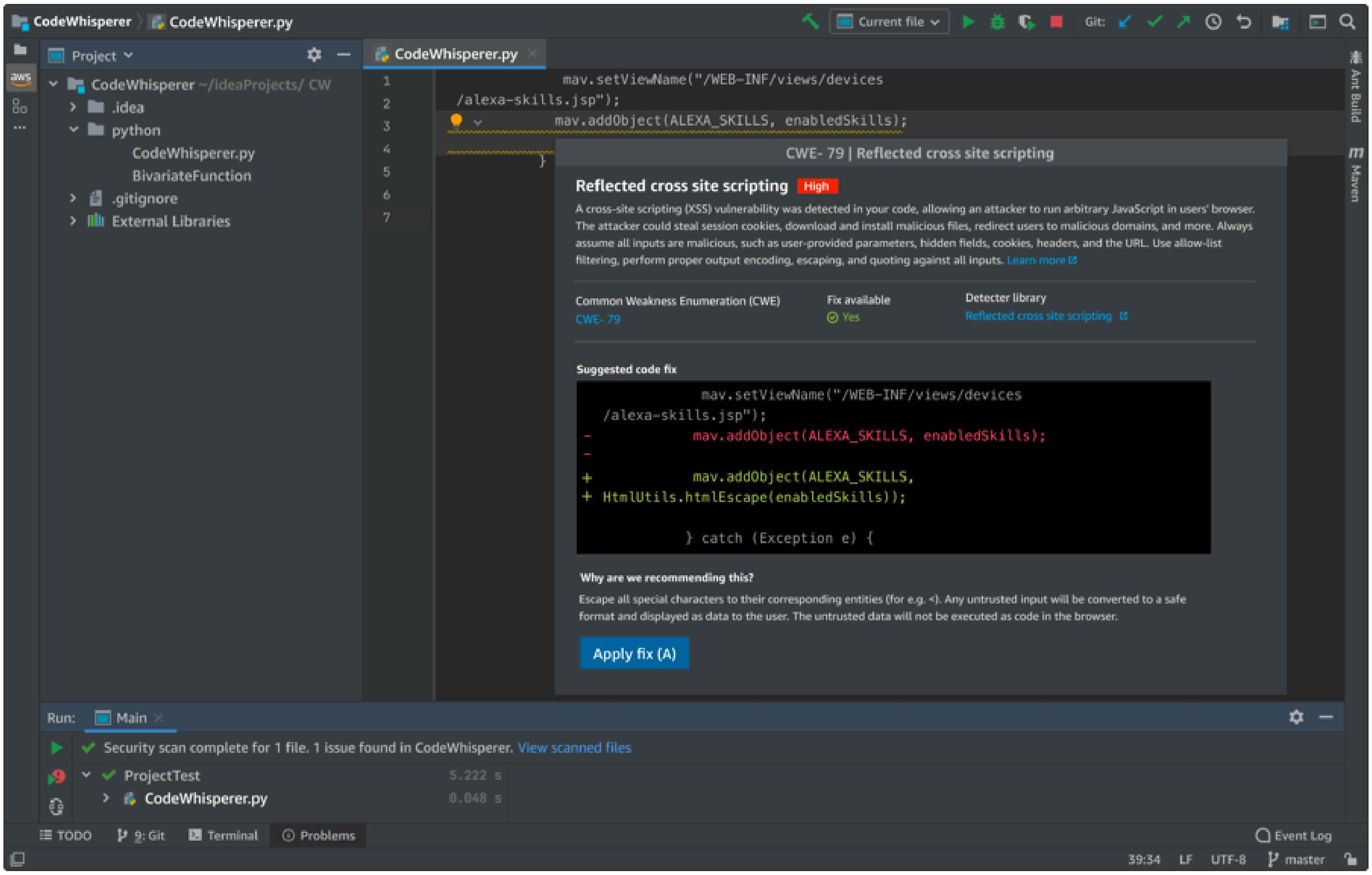
To begin a security scan in JetBrains, use the following procedure.

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**.
3. After creating a scan, you can view findings in the **CodeWhisperer Security Issues** tab of the **Problems** panel.

   To view information about the finding and suggested fix, hold your cursor over the underlined code.
4. CodeWhisperer may or may not provide a suggested code fix.

   - If it does provide a fix, and you want to implement that fix, choose **Apply fix**. The information about the finding will disappear.
   - If it does not provide a fix, update your code according to the information provided.

     Run another security scan to verify that the vulnerability was remediated.



```
        mav.setViewName("/WEB-INF/views/devices
/alexa-skills.jsp");
        mav.addObject(ALEXA_SKILLS, enabledSkills);
    }
```

CWE- 79 | Reflected cross site scripting

Reflected cross site scripting High

A cross-site scripting (XSS) vulnerability was detected in your code, allowing an attacker to run arbitrary JavaScript in users' browser. The attacker could steal session cookies, download and install malicious files, redirect users to malicious domains, and more. Always assume all inputs are malicious, such as user-provided parameters, hidden fields, cookies, headers, and the URL. Use allow-list filtering, perform proper output encoding, escaping, and quoting against all inputs. Learn more

Common Weakness Enumeration (CWE): CWE- 79

Fix available: Yes

Detecter library: Reflected cross site scripting

Suggested code fix

```
            mav.setViewName("/WEB-INF/views/devices
  /alexa-skills.jsp");
-           mav.addObject(ALEXA_SKILLS, enabledSkills);
-
+           mav.addObject(ALEXA_SKILLS,
+ HtmlUtils.htmlEscape(enabledSkills));

          } catch (Exception e) {
```

Why are we recommending this?

Escape all special characters to their corresponding entities (for e.g. <). Any untrusted input will be converted to a safe format and displayed as data to the user. The untrusted data will not be executed as code in the browser.

Apply fix (A)

```
Security scan complete for 1 file. 1 issue found in CodeWhisperer. View scanned files
ProjectTest            5.222 s
  CodeWhisperer.py     0.048 s
```

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.

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:

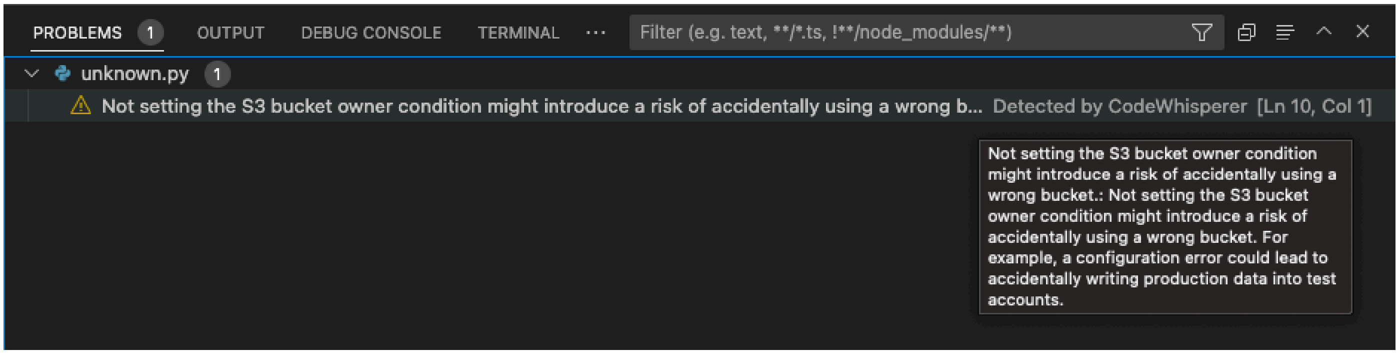
1. From the IntelliJ main menu, expand **File** (Windows) or open **Preferences** (Mac).
2. Choose **Project Structure** to open the **Project Structure** navigation pane.
3. Choose **Project** to open the **Project** pane.
4. Enter or select the location of your project's artifact files from the **Compiler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

AWS Toolkit for Visual Studio Code

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

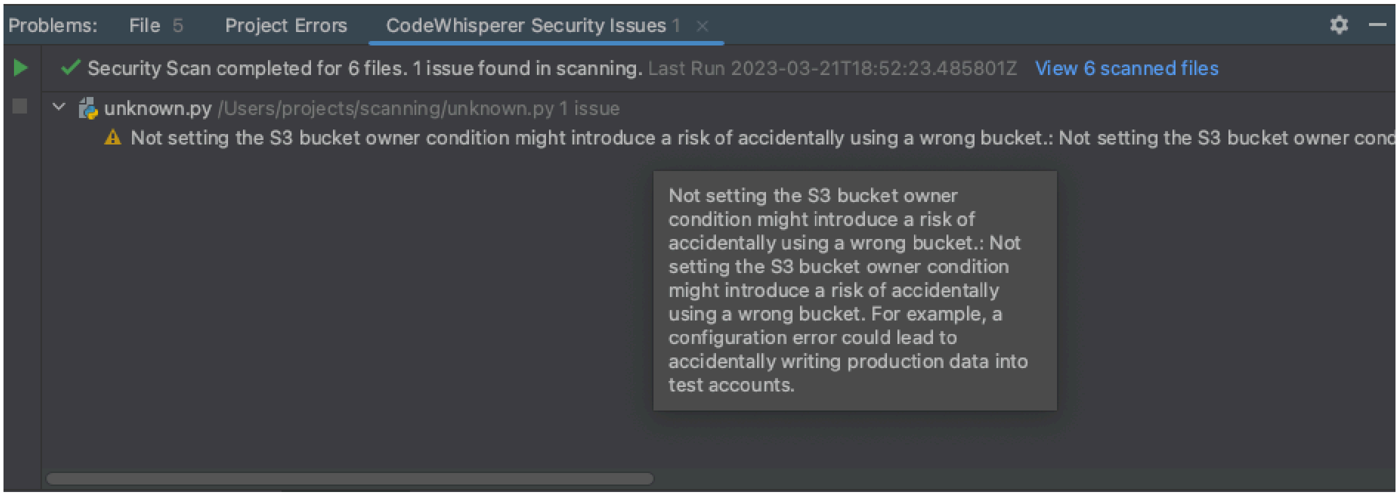


⚠ Not setting the S3 bucket owner condition might introduce a risk of accidentally using a wrong b... Detected by CodeWhisperer [Ln 10, Col 1]

Not setting the S3 bucket owner condition might introduce a risk of accidentally using a wrong bucket.: Not setting the S3 bucket owner condition might introduce a risk of accidentally using a wrong bucket. For example, a configuration error could lead to accidentally writing production data into test accounts.

AWS Toolkit for JetBrains

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



Problems: File 5 Project Errors CodeWhisperer Security Issues 1

Security Scan completed for 6 files. 1 issue found in scanning. Last Run 2023-03-21T18:52:23.485801Z View 6 scanned files

unknown.py /Users/projects/scanning/unknown.py 1 issue

Not setting the S3 bucket owner condition might introduce a risk of accidentally using a wrong bucket.: Not setting the S3 bucket owner cond

Not setting the S3 bucket owner condition might introduce a risk of accidentally using a wrong bucket.: Not setting the S3 bucket owner condition might introduce a risk of accidentally using a wrong bucket. For example, a configuration error could lead to accidentally writing production data into test accounts.

Code references

Topics

- Viewing code references
- Turning code references off and on
- Opting out of code with references

Viewing code references

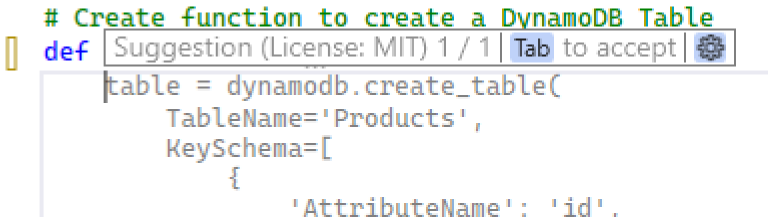
CodeWhisperer learns, in part, from open-source projects. Sometimes, a suggestion it's giving you may be similar to a specific piece of training data.

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.

Toolkit for Visual Studio

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



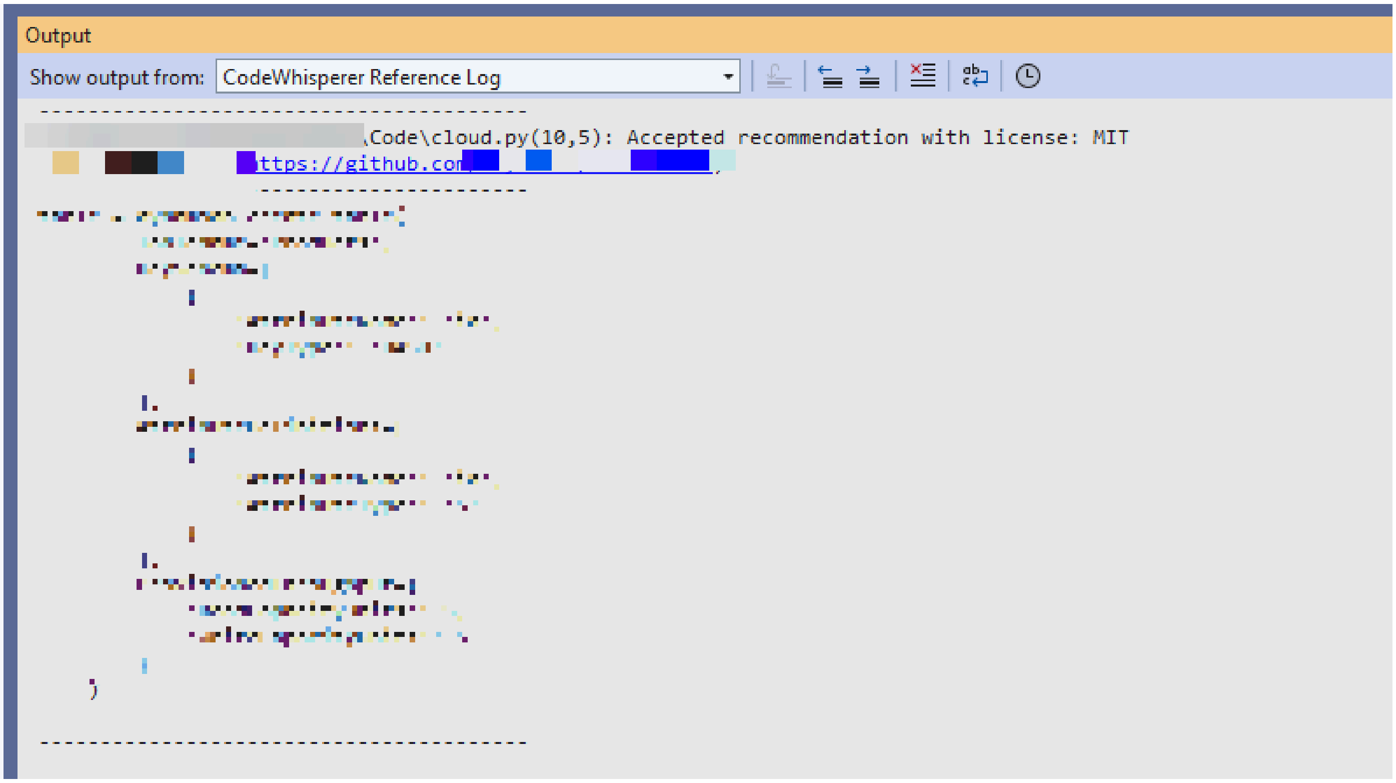
```
# Create function to create a DynamoDB Table
def Suggestion (License: MIT) 1 / 1 | Tab to accept
    table = dynamodb.create_table(
        TableName='Products',
        KeySchema=[
            {
                'AttributeName': 'id',
```

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

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

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

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



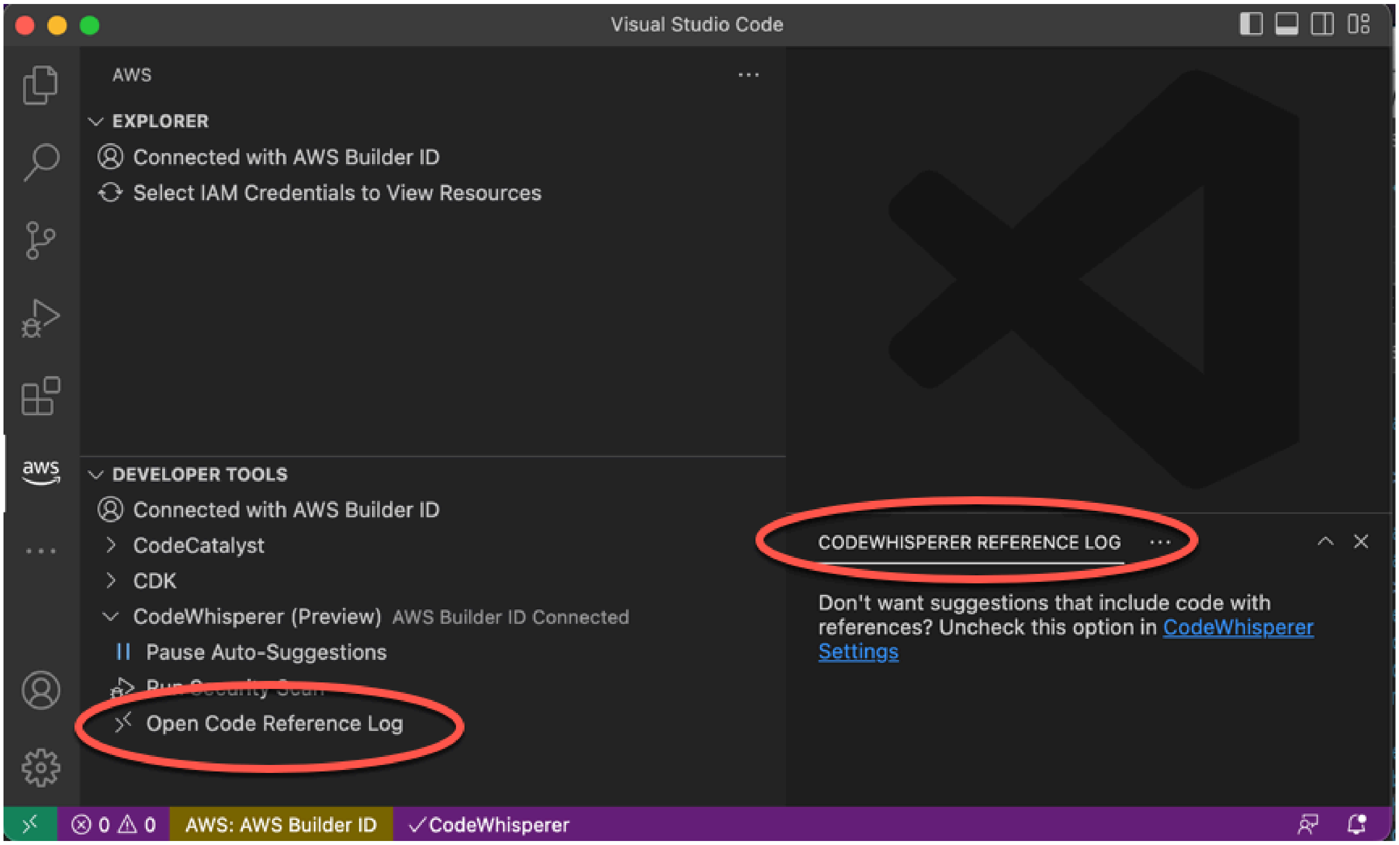
```
 ----------------------------------------
[redacted]\Code\cloud.py(10,5): Accepted recommendation with license: MIT
[redacted] https://github.com[redacted]
                    -----------------------
[redacted]
      }

 ----------------------------------------
```

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.

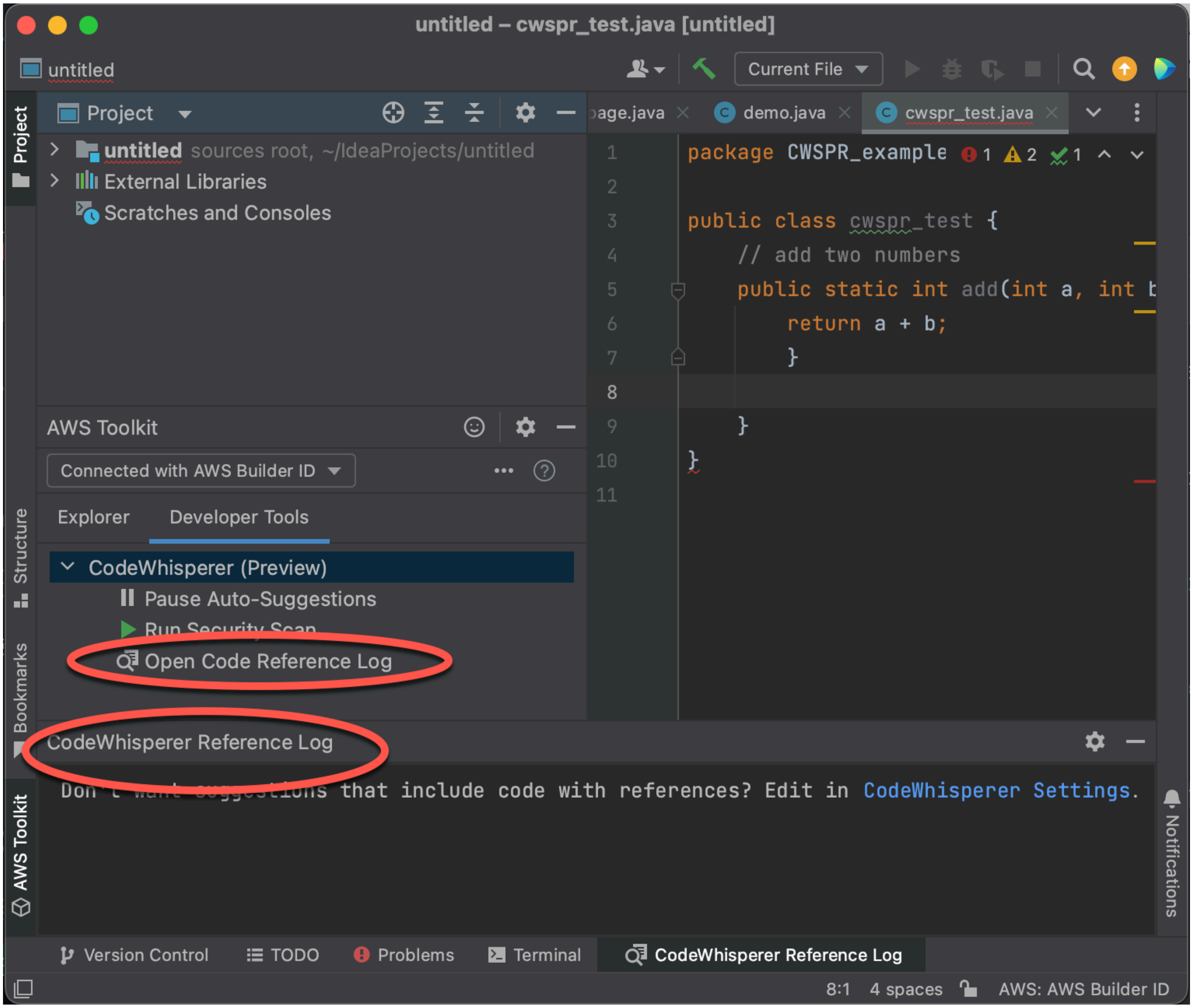

Visual Studio Code
AWS
EXPLORER
Connected with AWS Builder ID
Select IAM Credentials to View Resources
aws
DEVELOPER TOOLS
Connected with AWS Builder ID
CodeCatalyst
CDK
CodeWhisperer (Preview) AWS Builder ID Connected
Pause Auto-Suggestions
Run Security Scan
Open Code Reference Log
CODEWHISPERER REFERENCE LOG
0 0
AWS: AWS Builder ID
✓CodeWhisperer

Don't want suggestions that include code with references? Uncheck this option in CodeWhisperer Settings

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**.



```
package CWSPR_example

public class cwspr_test {
    // add two numbers
    public static int add(int a, int b
        return a + b;
        }

    }
}
```

Don't want suggestions that include code with references? Edit in CodeWhisperer Settings.

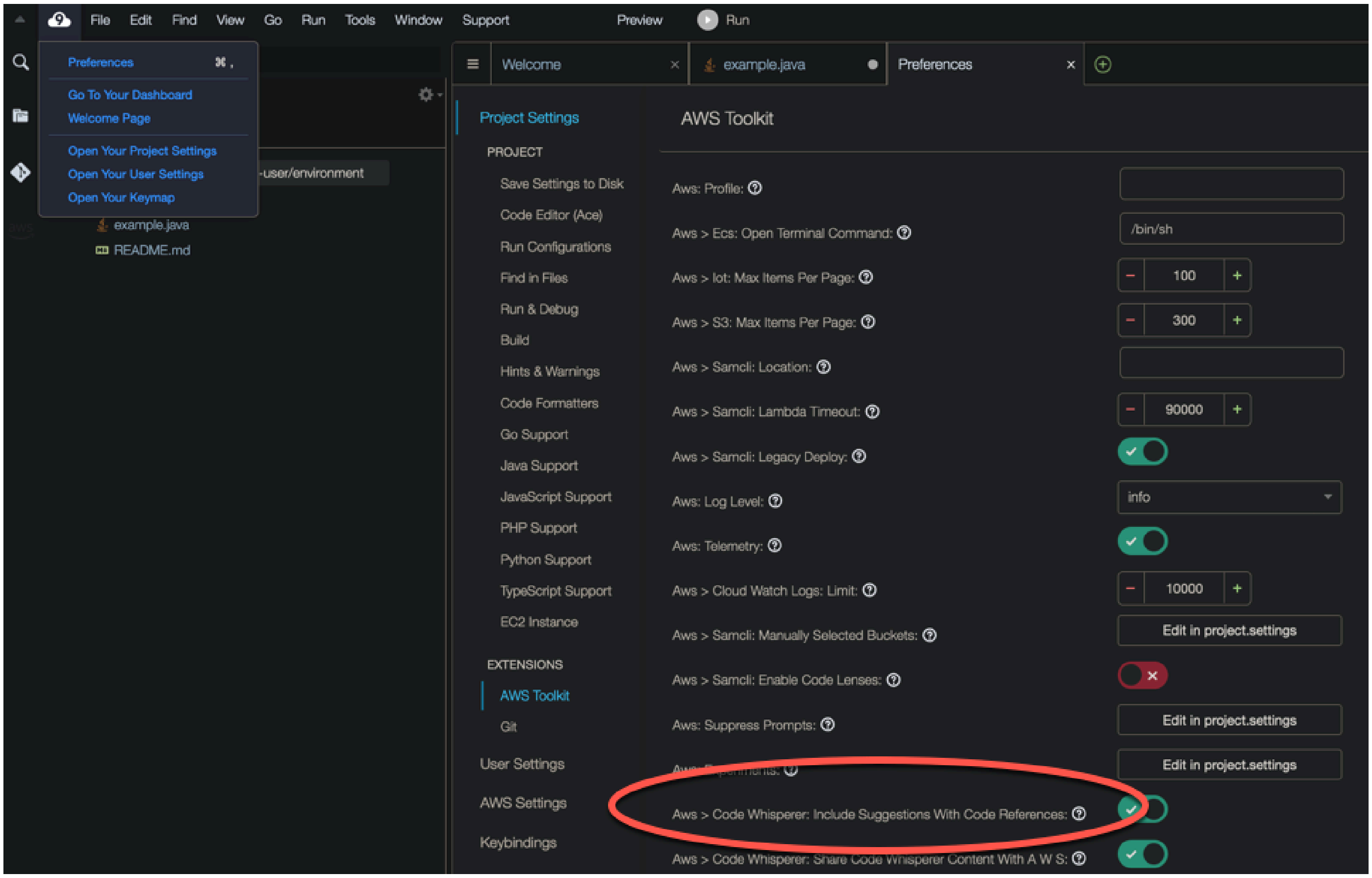
AWS Cloud 9

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**.



AWS Toolkit

Aws: Profile:	
Aws > Ecs: Open Terminal Command:	/bin/sh
Aws > Iot: Max Items Per Page:	100
Aws > S3: Max Items Per Page:	300
Aws > Samcli: Location:	
Aws > Samcli: Lambda Timeout:	90000
Aws > Samcli: Legacy Deploy:	✓
Aws: Log Level:	info
Aws: Telemetry:	✓
Aws > Cloud Watch Logs: Limit:	10000
Aws > Samcli: Manually Selected Buckets:	Edit in project.settings
Aws > Samcli: Enable Code Lenses:	✗
Aws: Suppress Prompts:	Edit in project.settings
Aws: Experiments:	Edit in project.settings
Aws > Code Whisperer: Include Suggestions With Code References:	✓
Aws > Code Whisperer: Share Code Whisperer Content With A W S:	✓

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.

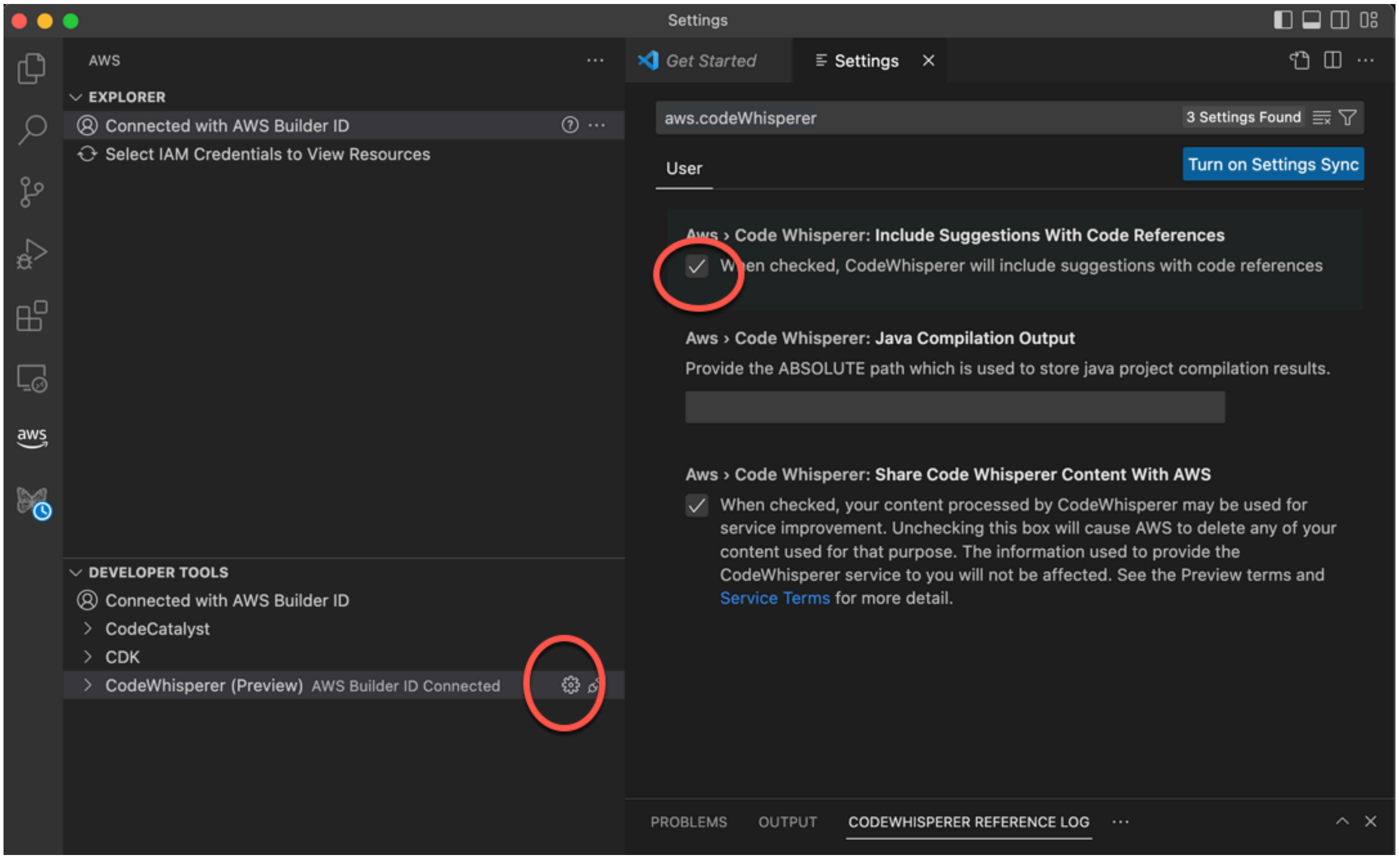
AWS Toolkit for Visual Studio Code

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**.


Settings
AWS
EXPLORER
Connected with AWS Builder ID
Select IAM Credentials to View Resources
DEVELOPER TOOLS
Connected with AWS Builder ID
CodeCatalyst
CDK
CodeWhisperer (Preview) AWS Builder ID Connected
Get Started
Settings
3 Settings Found
User
Turn on Settings Sync
PROBLEMS
OUTPUT
CODEWHISPERER REFERENCE LOG

aws.codeWhisperer

Aws › Code Whisperer: **Include Suggestions With Code References**

☑ When checked, CodeWhisperer will include suggestions with code references

Aws › Code Whisperer: **Java Compilation Output**

Provide the ABSOLUTE path which is used to store java project compilation results.

Aws › Code Whisperer: **Share Code Whisperer Content With AWS**

☑ When checked, your content processed by CodeWhisperer may be used for service improvement. Unchecking this box will cause AWS to delete any of your content used for that purpose. The information used to provide the CodeWhisperer service to you will not be affected. See the Preview terms and Service Terms for more detail.

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.

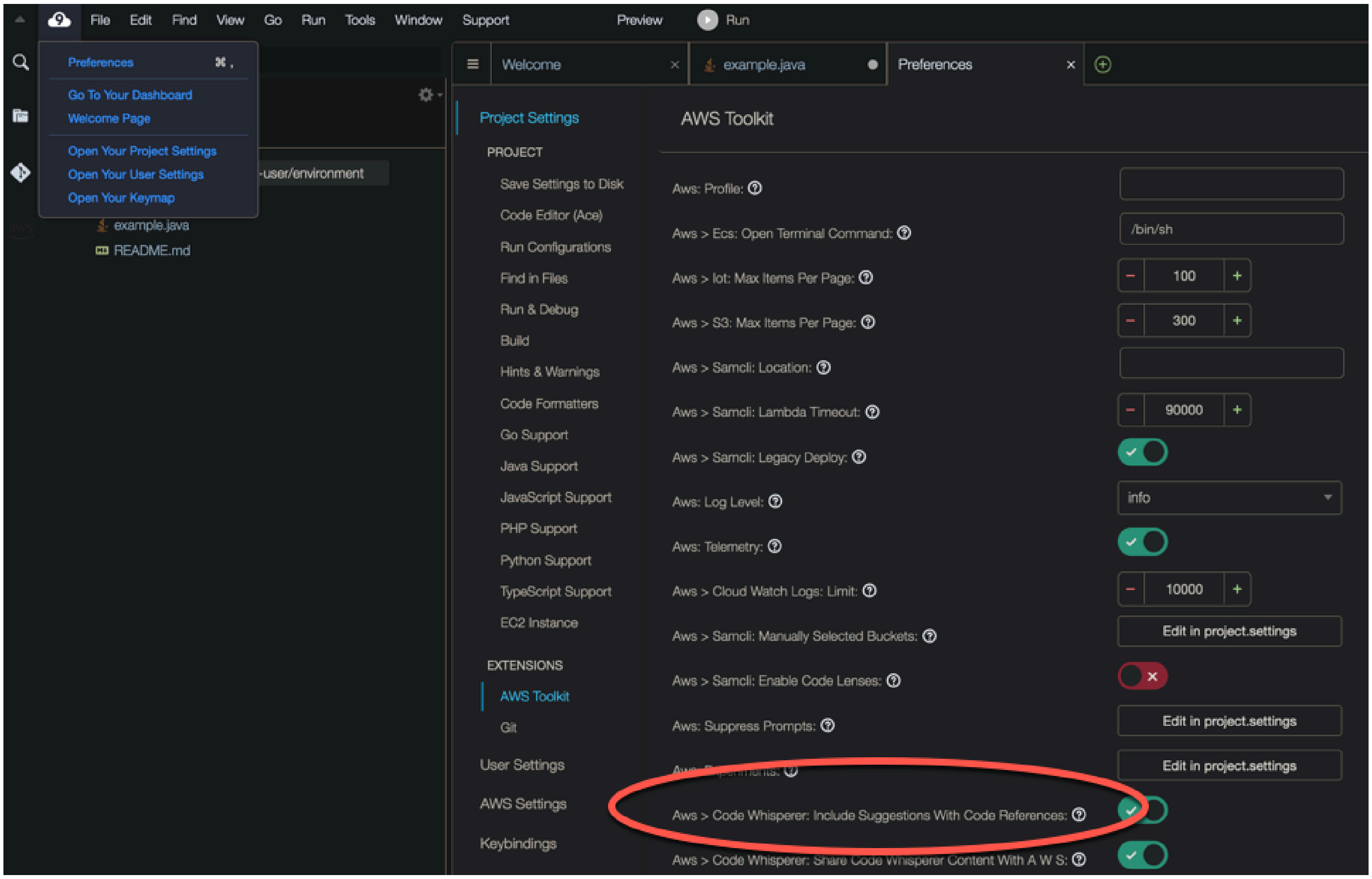
To turn them off, or to turn them back on later, use the following procedure.

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**.



AWS Toolkit

Aws: Profile:

Aws > Ecs: Open Terminal Command: /bin/sh

Aws > Iot: Max Items Per Page: 100

Aws > S3: Max Items Per Page: 300

Aws > Samcli: Location:

Aws > Samcli: Lambda Timeout: 90000

Aws > Samcli: Legacy Deploy: ✓

Aws: Log Level: info

Aws: Telemetry: ✓

Aws > Cloud Watch Logs: Limit: 10000

Aws > Samcli: Manually Selected Buckets: Edit in project.settings

Aws > Samcli: Enable Code Lenses: ✕

Aws: Suppress Prompts: Edit in project.settings

Aws: Experiments: Edit in project.settings

Aws > Code Whisperer: Include Suggestions With Code References: ✓

Aws > Code Whisperer: Share Code Whisperer Content With A W S: ✓

Lambda

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

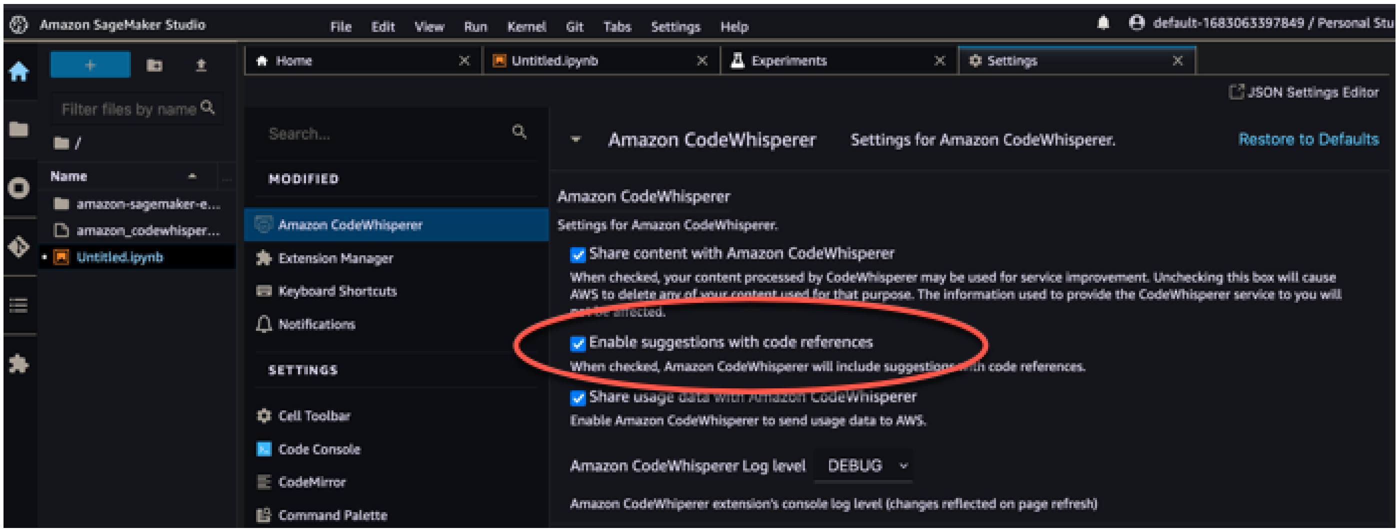
SageMaker Studio

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**.



Amazon CodeWhisperer

Settings for Amazon CodeWhisperer.

☑ Share content with Amazon CodeWhisperer

When checked, your content processed by CodeWhisperer may be used for service improvement. Unchecking this box will cause AWS to delete any of your content used for that purpose. The information used to provide the CodeWhisperer service to you will not be affected.

☑ Enable suggestions with code references

When checked, Amazon CodeWhisperer will include suggestions with code references.

☑ Share usage data with Amazon CodeWhisperer

Enable Amazon CodeWhisperer to send usage data to AWS.

Amazon CodeWhisperer Log level DEBUG

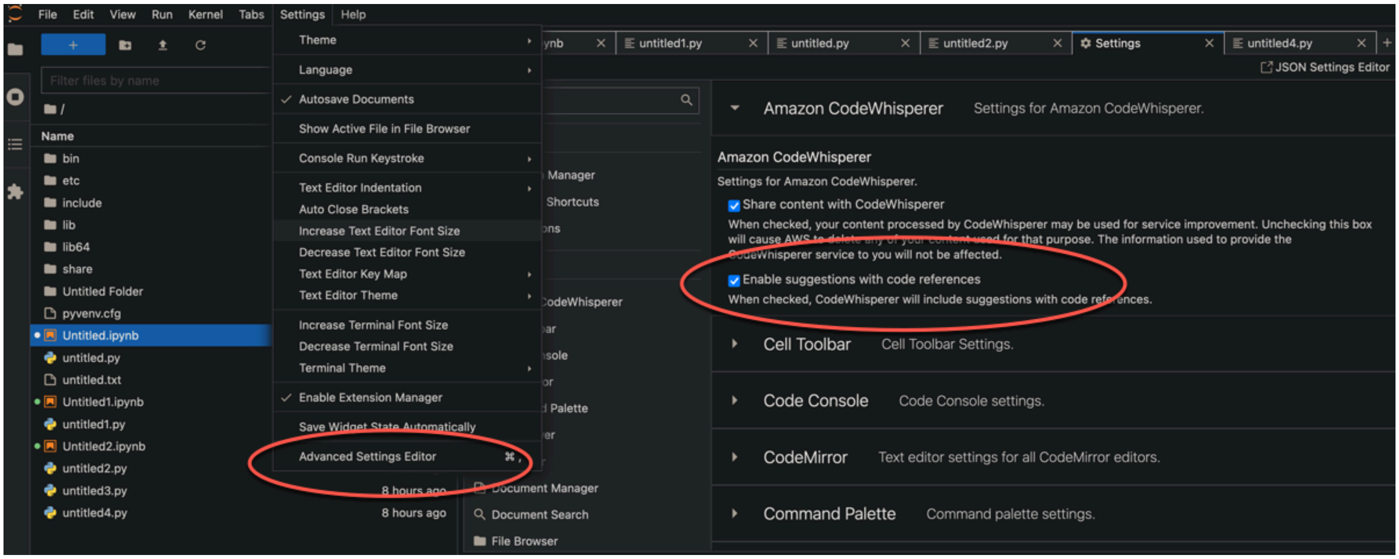
Amazon CodeWhiperer extension's console log level (changes reflected on page refresh)

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**.



Amazon CodeWhisperer Settings for Amazon CodeWhisperer.

Amazon CodeWhisperer

Settings for Amazon CodeWhisperer.

☑ Share content with CodeWhisperer

When checked, your content processed by CodeWhisperer may be used for service improvement. Unchecking this box will cause AWS to delete any of your content used for that purpose. The information used to provide the CodeWhisperer service to you will not be affected.

☑ Enable suggestions with code references

When checked, CodeWhisperer will include suggestions with code references.

Cell Toolbar Cell Toolbar Settings.

Code Console Code Console settings.

CodeMirror Text editor settings for all CodeMirror editors.

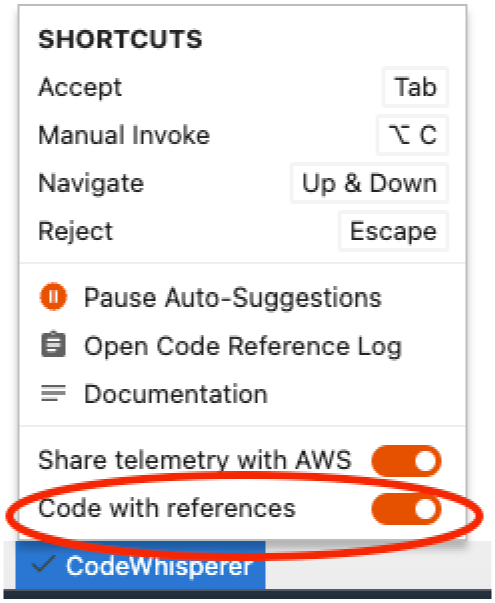
Command Palette Command palette settings.

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.



SHORTCUTS

Accept	Tab
Manual Invoke	⌥ C
Navigate	Up & Down
Reject	Escape

Share telemetry with AWS ●

Code with references ●

Opting out of code with references

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

This section explains how to do so.

Toolkit for Visual Studio

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

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

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

AWS Toolkit for Visual Studio Code

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

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

1. 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 professionals 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 IAM Identity Center 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 IDEs.

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.

Code examples

Topics

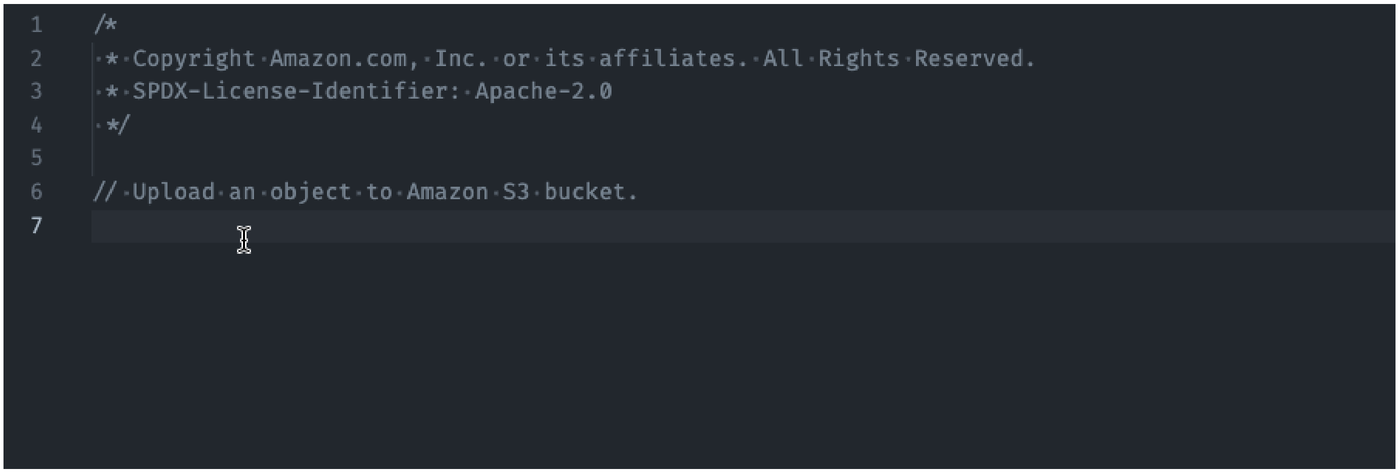
- Single-line code completion
- Full function generation
- Block completion
- Docstring, JSDoc, and Javadoc completion
- Line-by-line recommendations

Single-line code completion

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

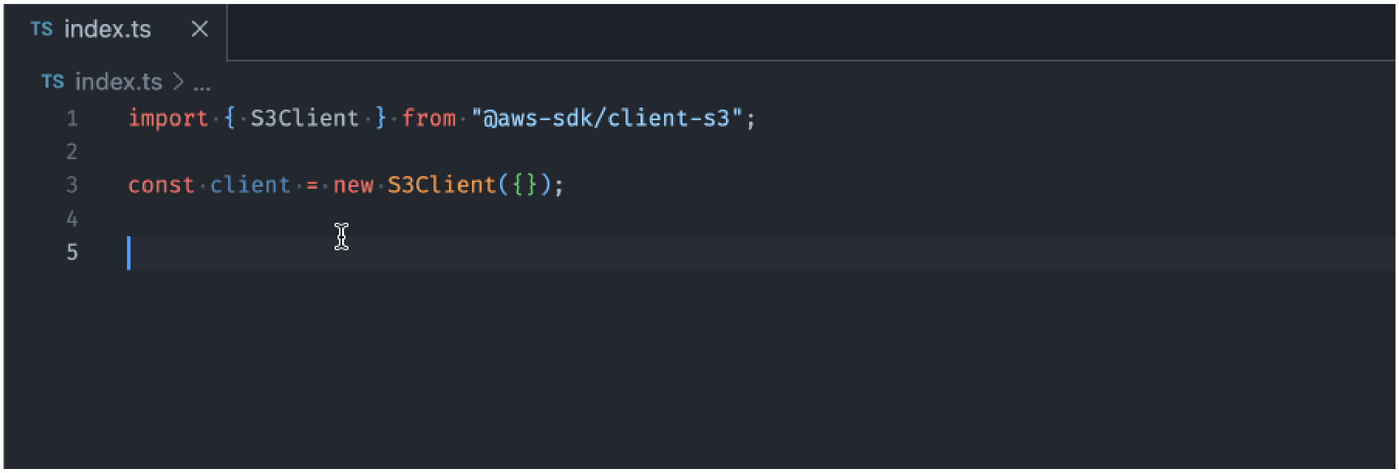
AWS Toolkit for Visual Studio Code

In this example using JavaScript and VS Code, CodeWhisperer completes a line of code that the developer begins.



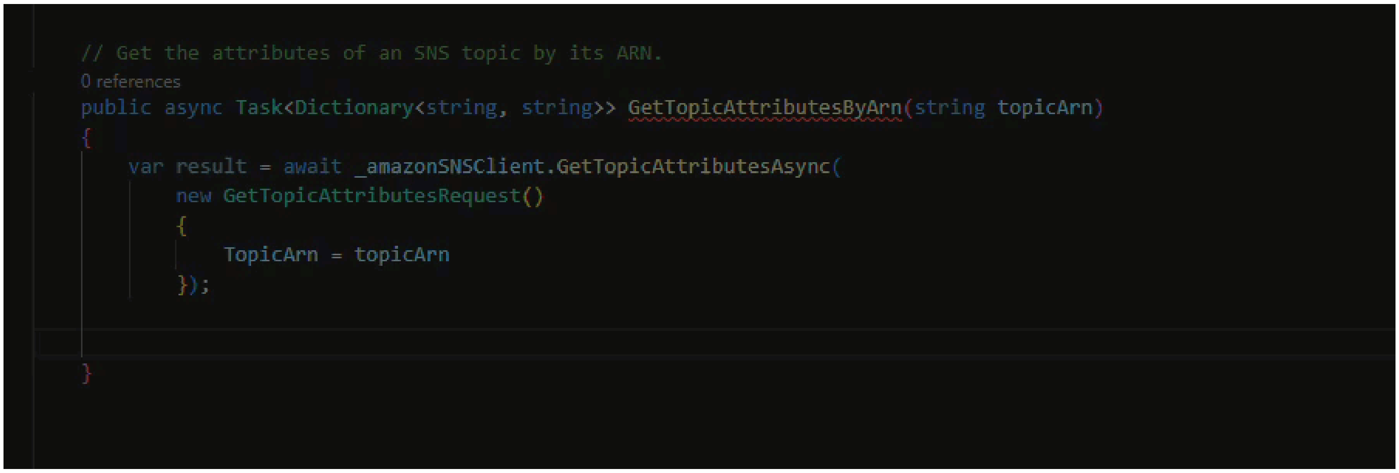
```
/*
 * Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
 * SPDX-License-Identifier: Apache-2.0
 */

// Upload an object to Amazon S3 bucket.

```

In this example using TypeScript and VS Code, the user enters a full comment, and then CodeWhisperer supplies the code that goes with it.



```
import { S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

```

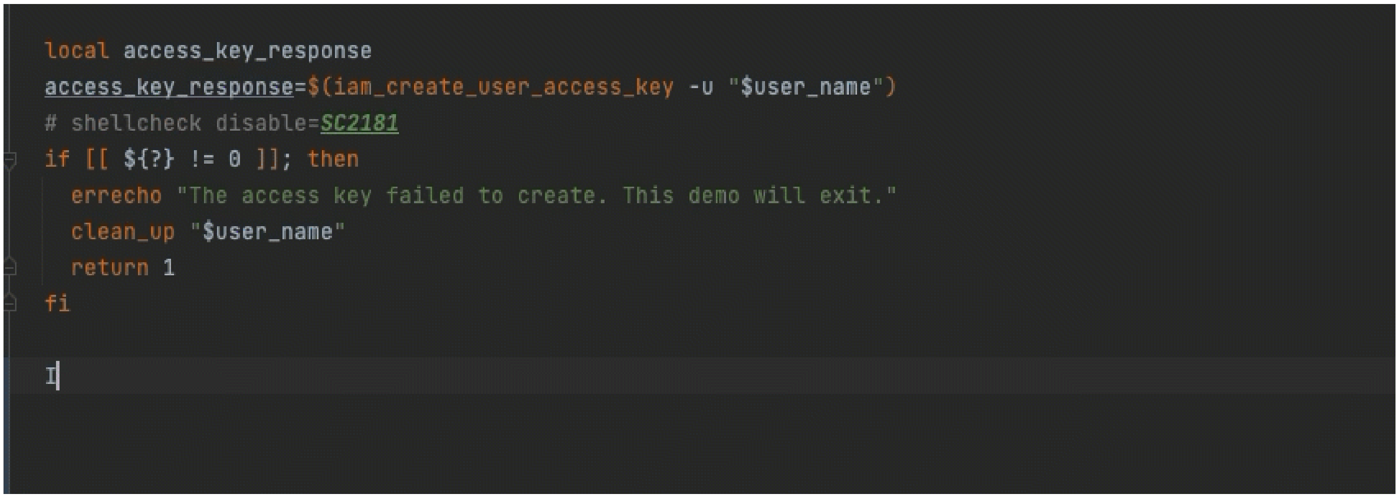
In this example, CodeWhisperer provides a single-line recommendation based on a comment, using C# and VS Code.



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


}
```

AWS Toolkit for JetBrains

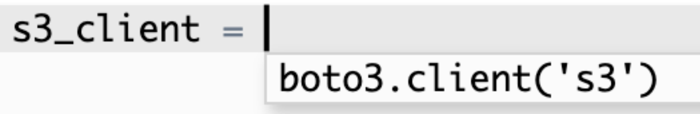
In the image below, using a shell script written in IntelliJ, CodeWhisperer offers recommendations on how to complete a single line of code.



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

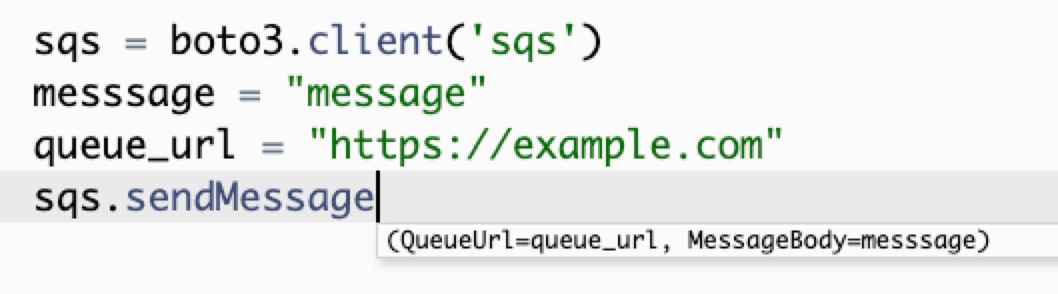
Lambda

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.



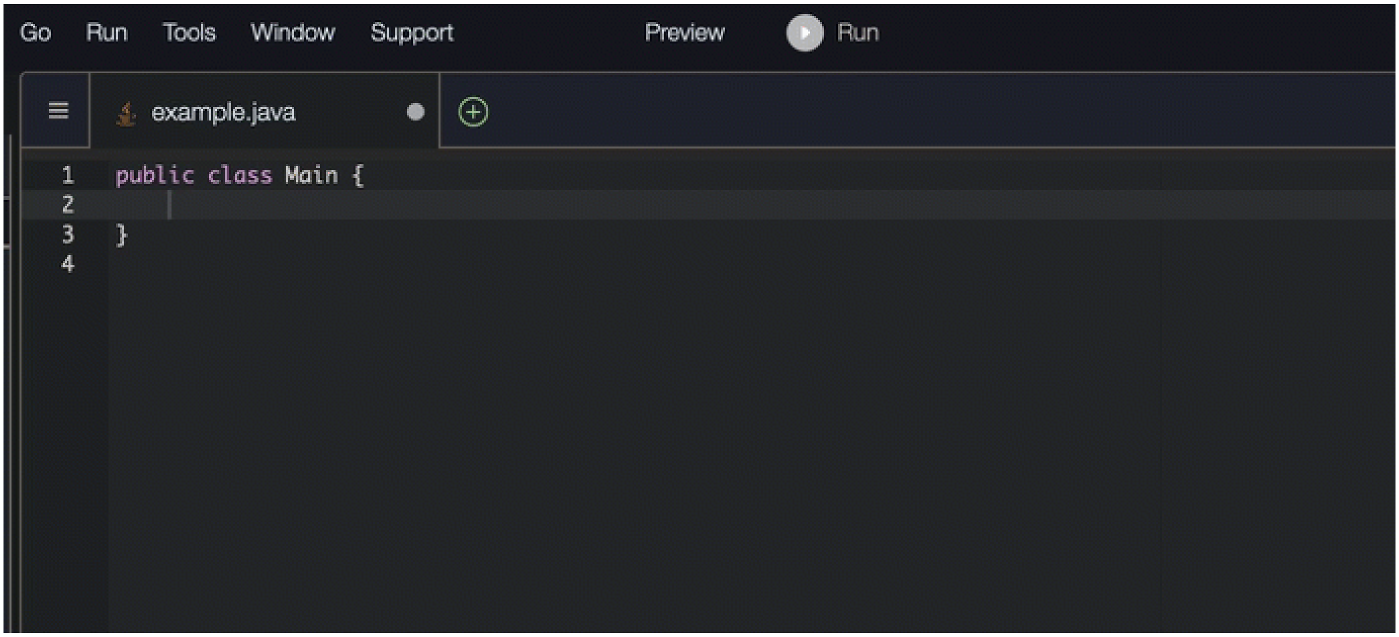
```
sqs = boto3.client('sqs')
messsage = "message"
queue_url = "https://example.com"
sqs.sendMessage
               (QueueUrl=queue_url, MessageBody=messsage)
```

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.

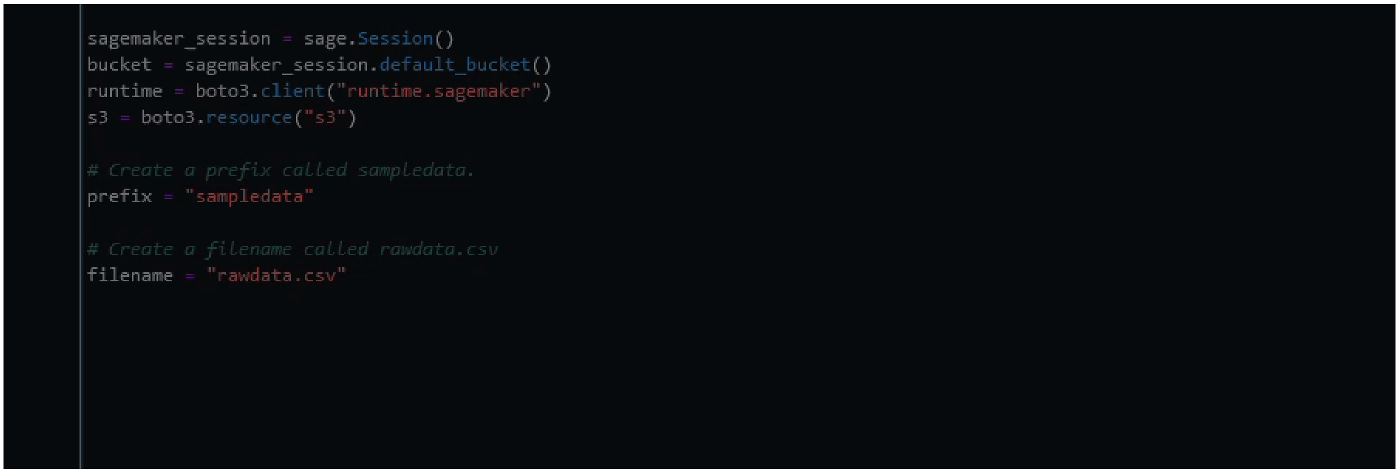
Based on the input, CodeWhisperer generates a suggestion for the signature of the main method.



```
public class Main {

}
```

SageMaker Studio

In this example using Python and SageMaker Studio, CodeWhisperer recommends a single line of code, based on the developer's comment.



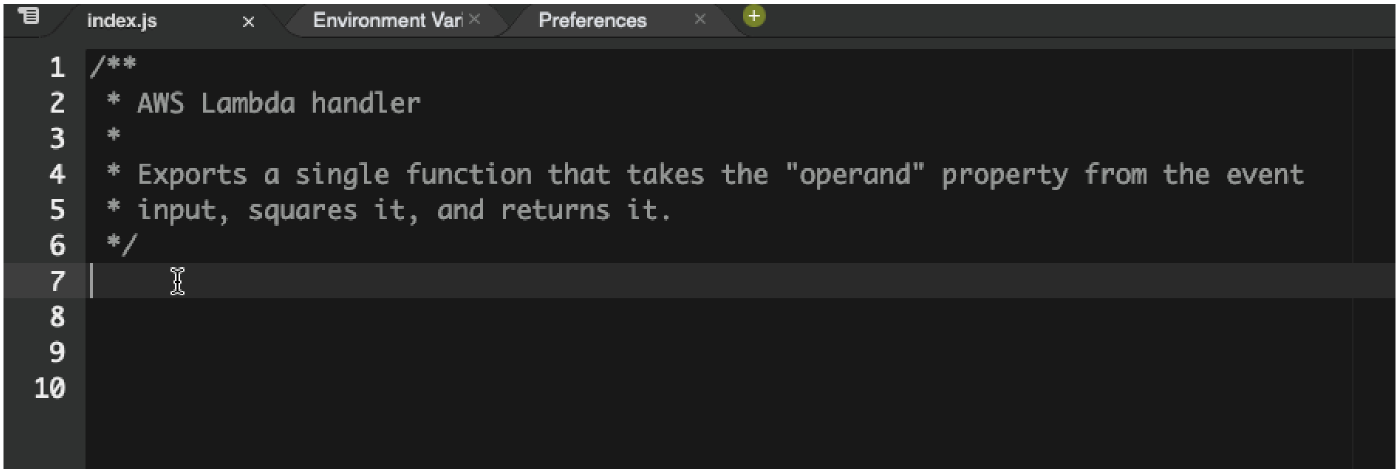
```
sagemaker_session = sage.Session()
bucket = sagemaker_session.default_bucket()
runtime = boto3.client("runtime.sagemaker")
s3 = boto3.resource("s3")

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

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

Full function generation

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.

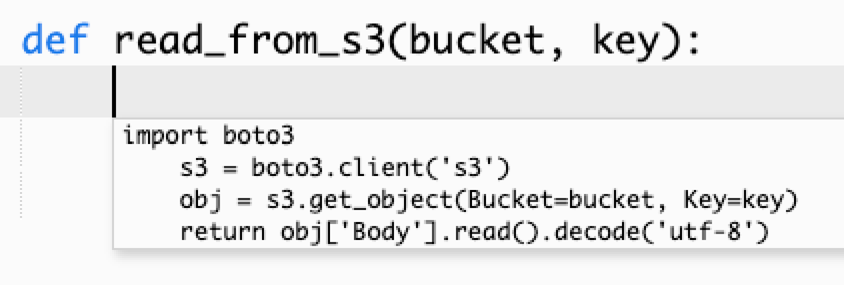
Lambda

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



```
/**
 * AWS Lambda handler
 *
 * Exports a single function that takes the "operand" property from the event
 * input, squares it, and returns it.
 */
```

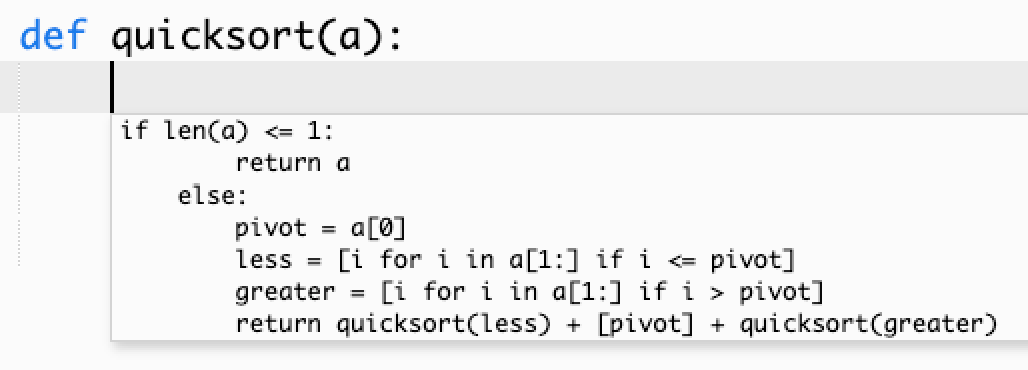
In the following image, a user has written a function signature for reading a file from Amazon S3. Amazon CodeWhisperer then suggests a full implementation of the `read_from_s3` method.



```
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')
```

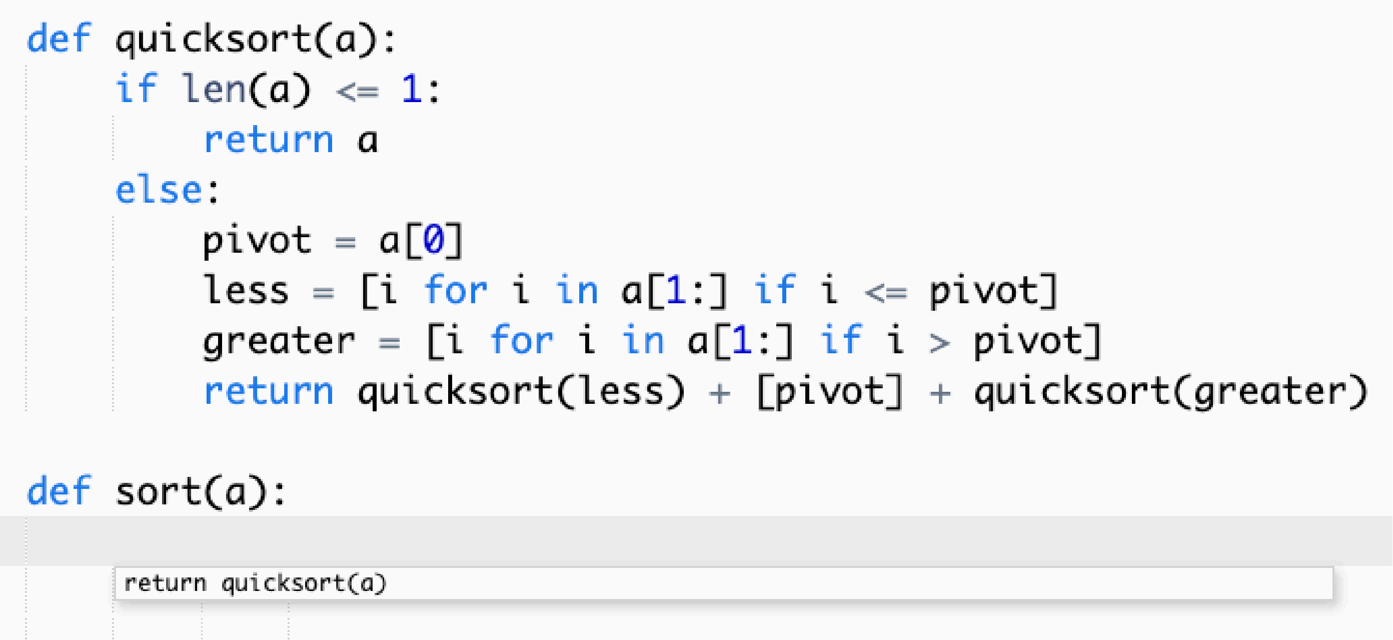
> **Note**
>
> Sometimes, as in the previous example, CodeWhisperer includes `import` statements as part of its suggestions. As a best practice, manually move these `import` statements to the top of your file.

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



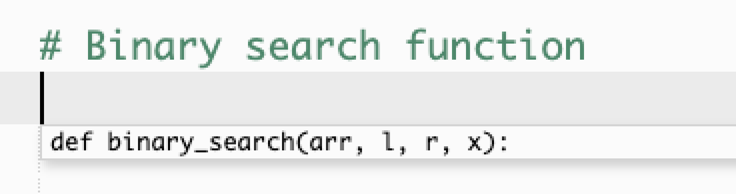
```
def quicksort(a):

    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.



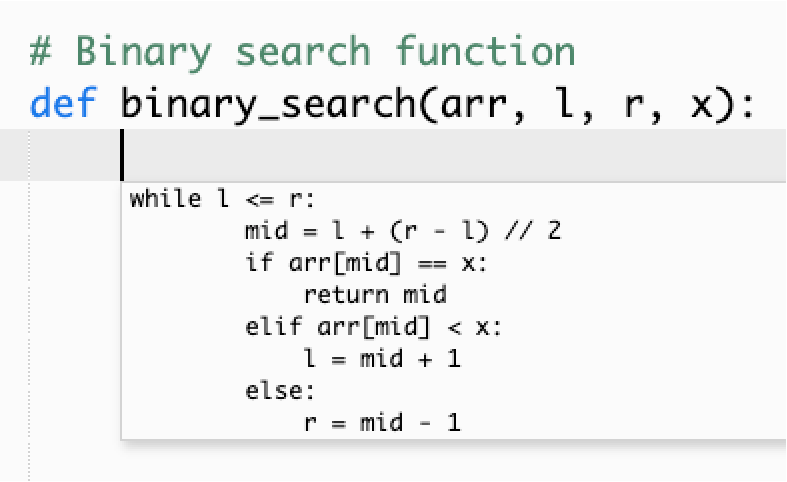
```
def quicksort(a):
    if len(a) <= 1:
        return a
    else:
        pivot = a[0]
        less = [i for i in a[1:] if i <= pivot]
        greater = [i for i in a[1:] if i > pivot]
        return quicksort(less) + [pivot] + quicksort(greater)

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.



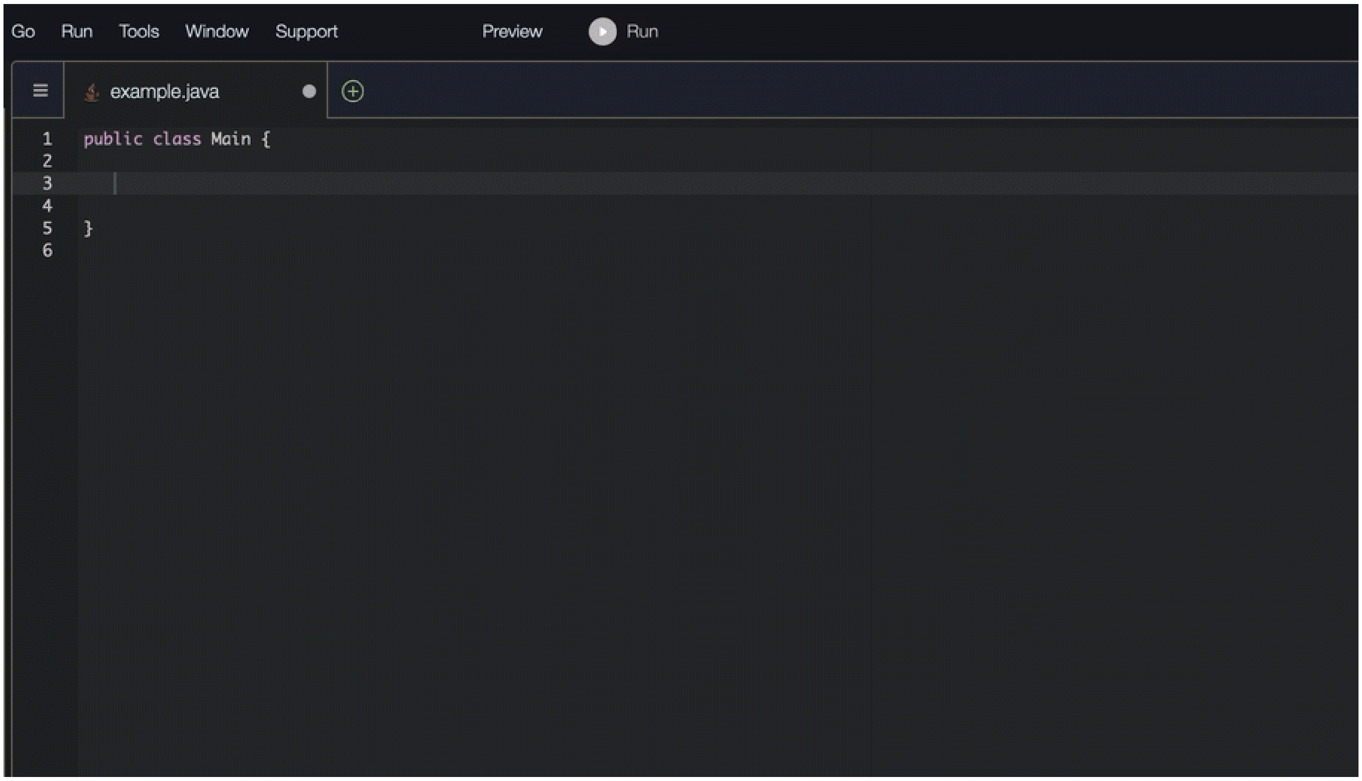
```
# Binary search function
def binary_search(arr, l, r, x):
        
        while l <= r:
                mid = l + (r - l) // 2
                if arr[mid] == x:
                    return mid
                elif arr[mid] < x:
                    l = mid + 1
                else:
              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

AWS Cloud9

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

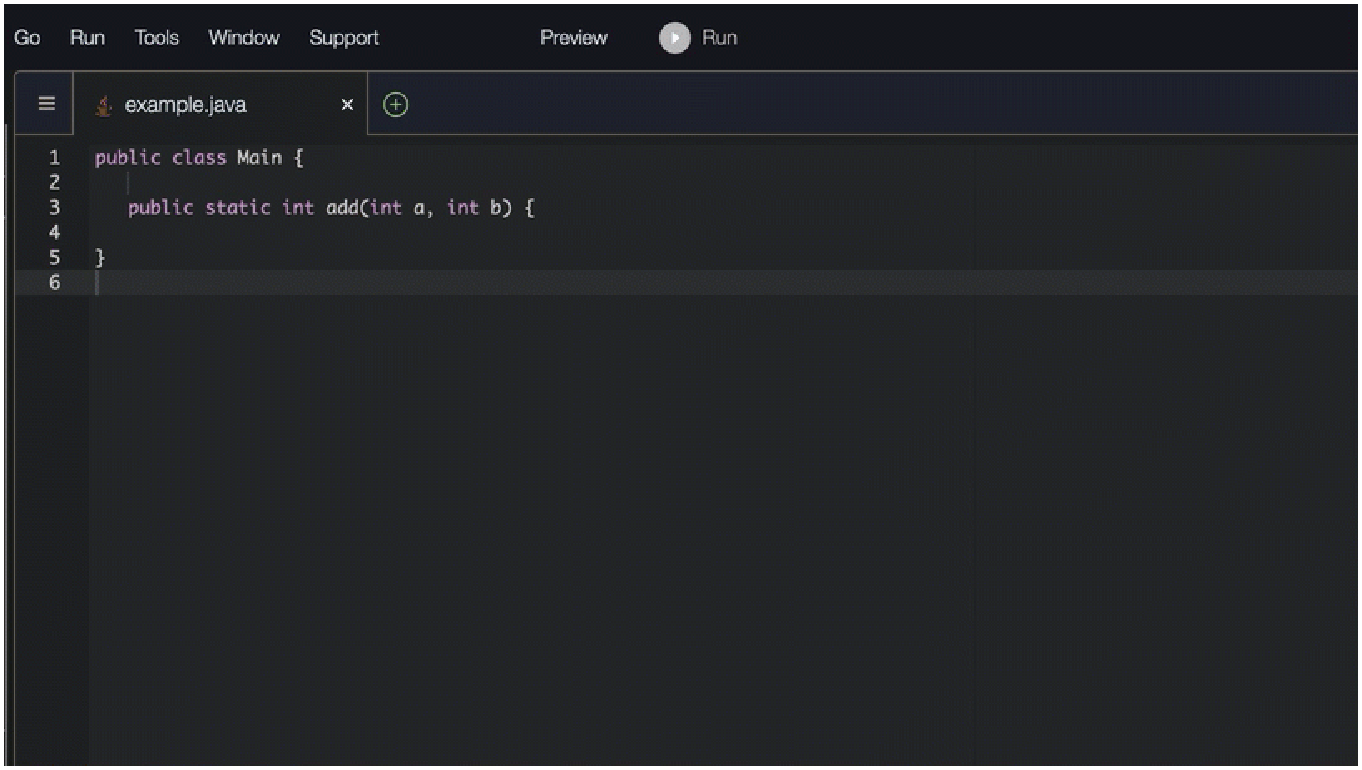
1. In the example below, in Java, a user inputs a comment. CodeWhisperer suggests a function signature.

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



```
public class Main {



}
```

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.

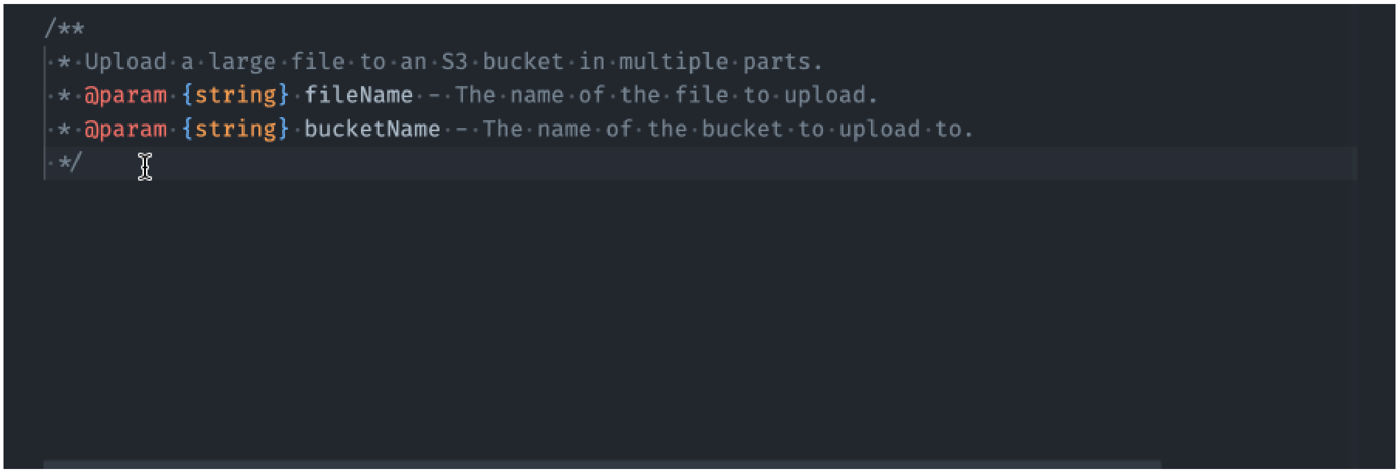


```
public class Main {

    public static int add(int a, int b) {

}
```

AWS Toolkit for Visual Studio Code

In the following example using C# and VS Code, CodeWhisperer recommends a full function.

In the following example, using TypeScript and VS Code, CodeWhisperer generates a function based on the user's docstrings.

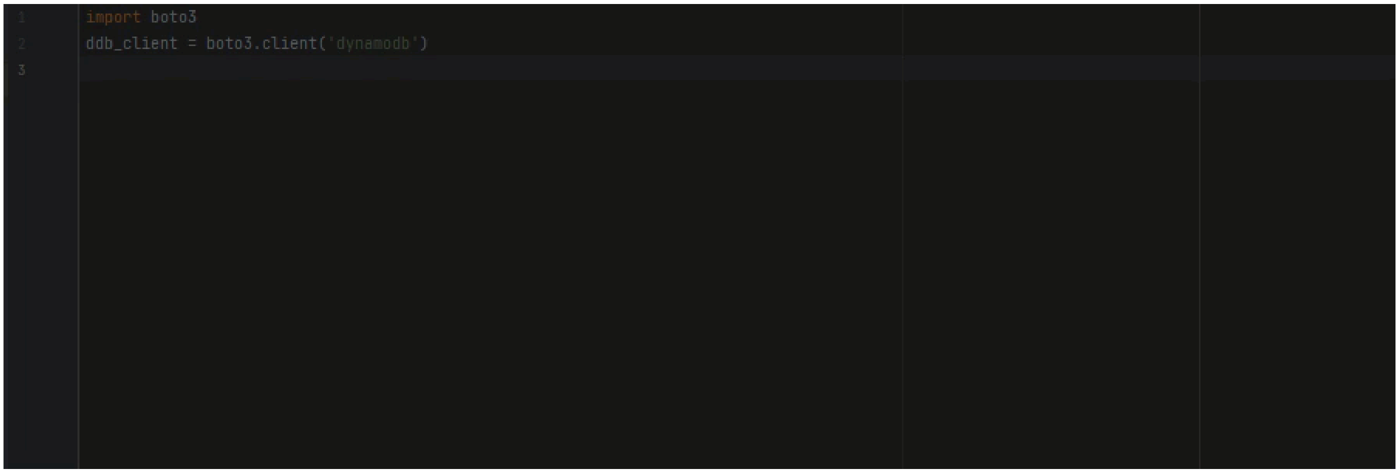


```
/**
 * Upload a large file to an S3 bucket in multiple parts.
 * @param {string} fileName - The name of the file to upload.
 * @param {string} bucketName - The name of the bucket to upload to.
 */
```

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.

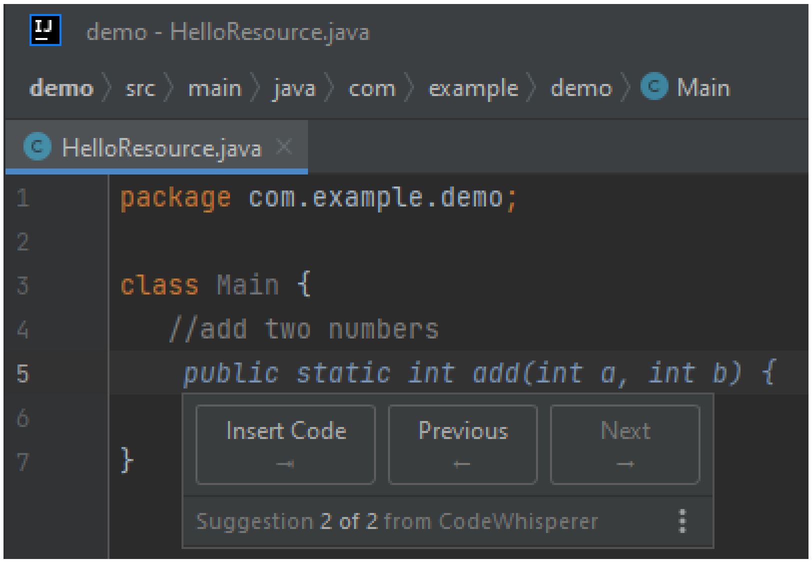
In the following example, using Python in Pycharm, CodeWhisperer generates both a full function and the corresponding unit test.



```
import boto3
ddb_client = boto3.client('dynamodb')
```

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

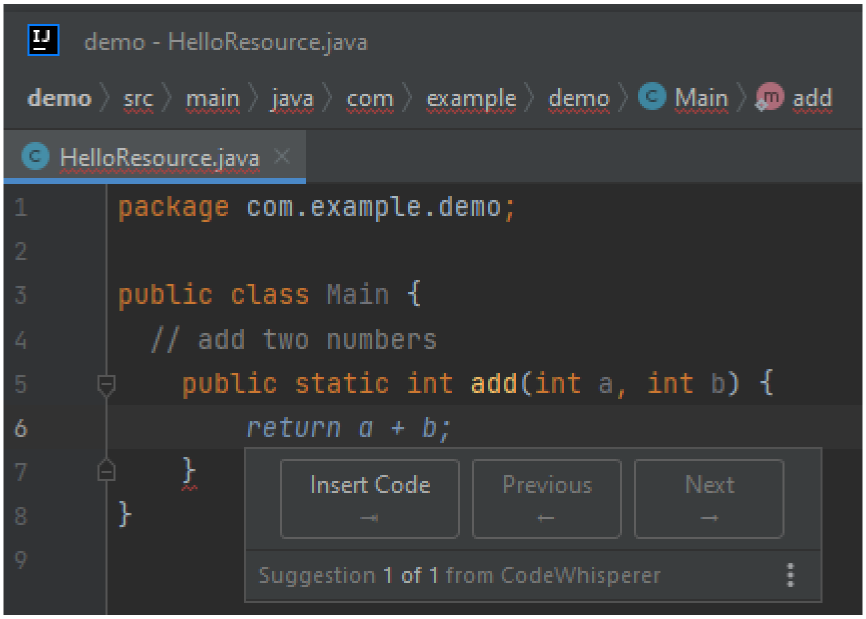
1. In the image below, a user has input a comment. The function signature, located below the comment, is a suggestion from CodeWhisperer.



demo - HelloResource.java
demo 〉 src 〉 main 〉 java 〉 com 〉 example 〉 demo 〉 Main
HelloResource.java
Insert Code ⇥
Previous ←
Next →
Suggestion 2 of 2 from CodeWhisperer

```
package com.example.demo;

class Main {
   //add two numbers
    public static int add(int a, int b) {

}
```

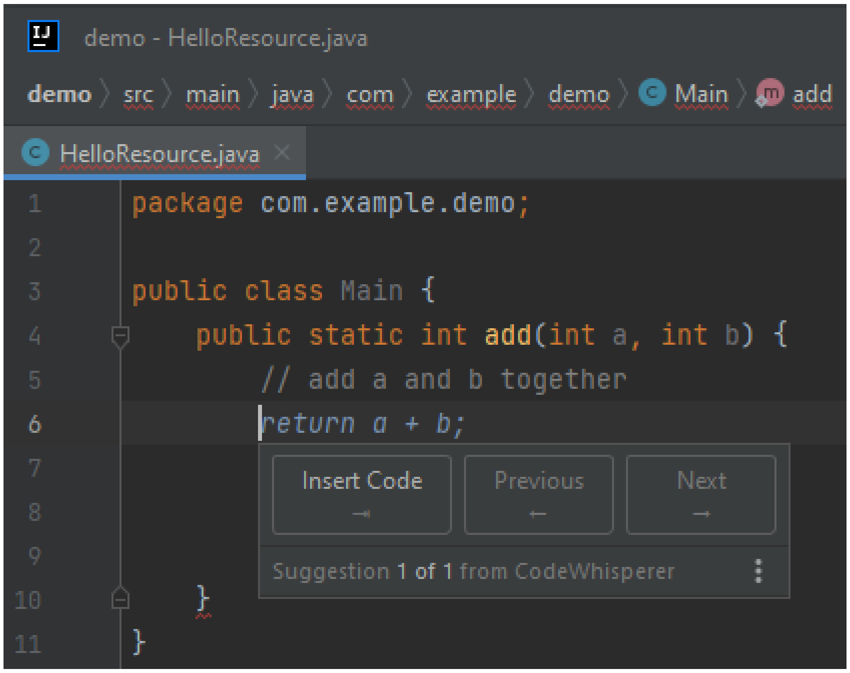
2. In the image below, the user has accepted the CodeWhisperer suggestion for a function signature. Accepting the suggestion automatically advanced the cursor and CodeWhisperer has made a new suggestion for the function body.



demo - HelloResource.java
demo 〉 src 〉 main 〉 java 〉 com 〉 example 〉 demo 〉 Main 〉 add
HelloResource.java
Insert Code
Previous
Next
Suggestion 1 of 1 from CodeWhisperer

```
package com.example.demo;

public class Main {
  // add two numbers
    public static int add(int a, int b) {
        return a + b;
    }
}
```

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.

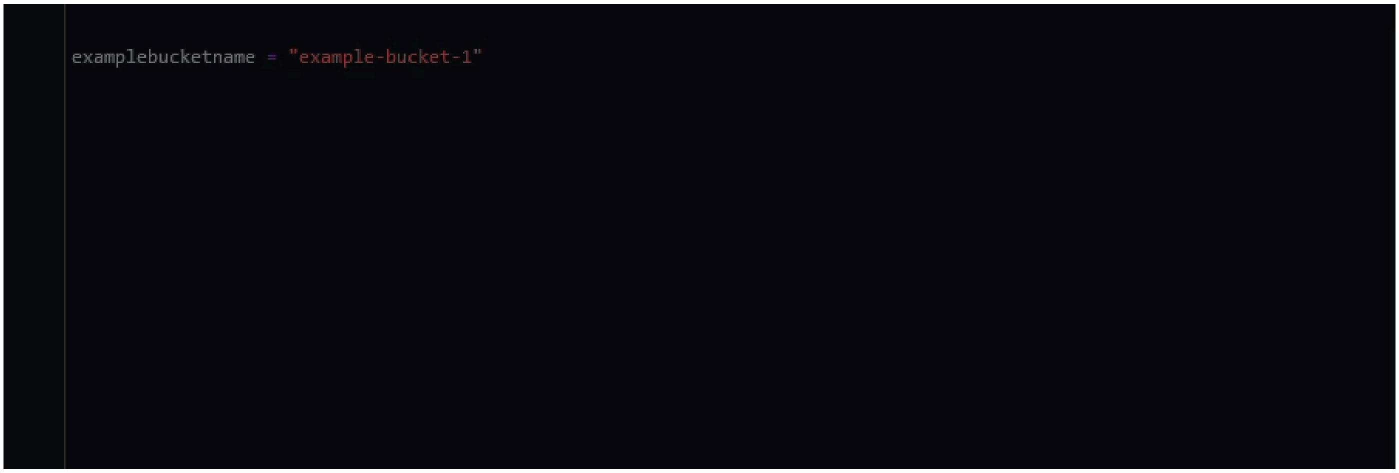


demo - HelloResource.java
demo 〉 src 〉 main 〉 java 〉 com 〉 example 〉 demo 〉 Main 〉 add
HelloResource.java
Insert Code
Previous
Next
Suggestion 1 of 1 from CodeWhisperer

```
package com.example.demo;

public class Main {
    public static int add(int a, int b) {
        // add a and b together
        return a + b;



    }
}
```

SageMaker Studio

In this example using Python and SageMaker Studio, CodeWhisperer recommends a full function after the user types part of the signature.



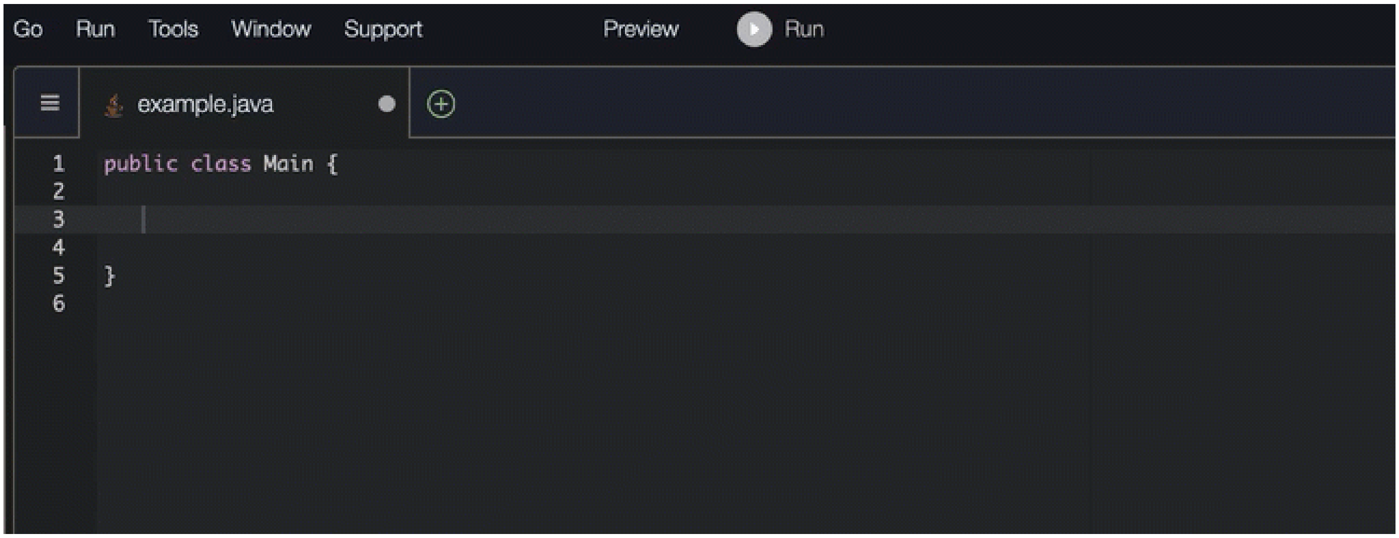
```
examplebucketname = "example-bucket-1"
```

Block completion

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

AWS Cloud9

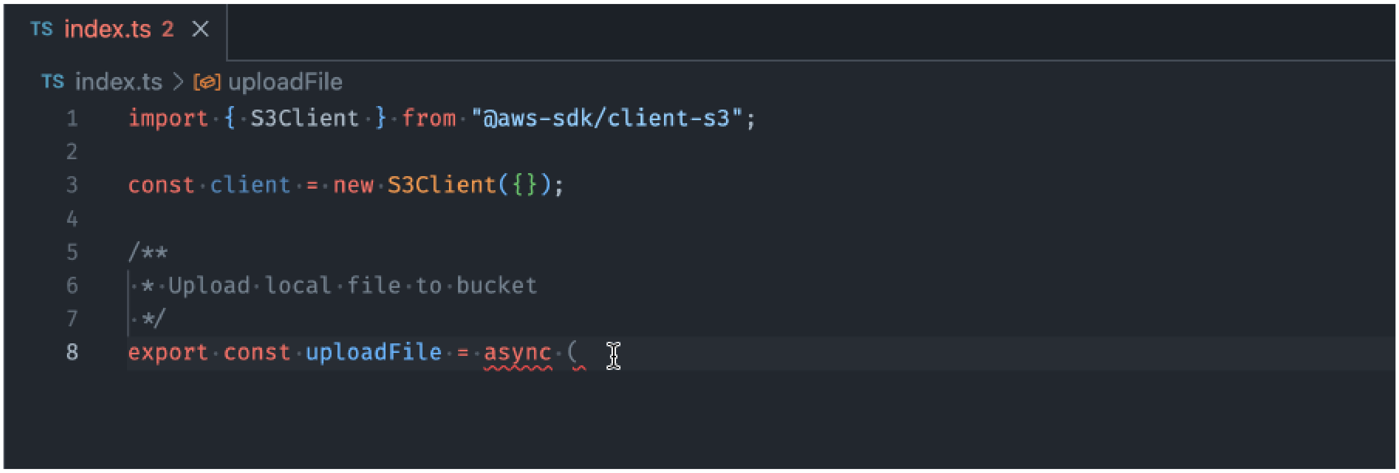
In the example below, in Java, a user enters the signature of an `if` statement. The body of the statement is a suggestion from CodeWhisperer.



```
public class Main {


}
```

AWS Toolkit for Visual Studio Code

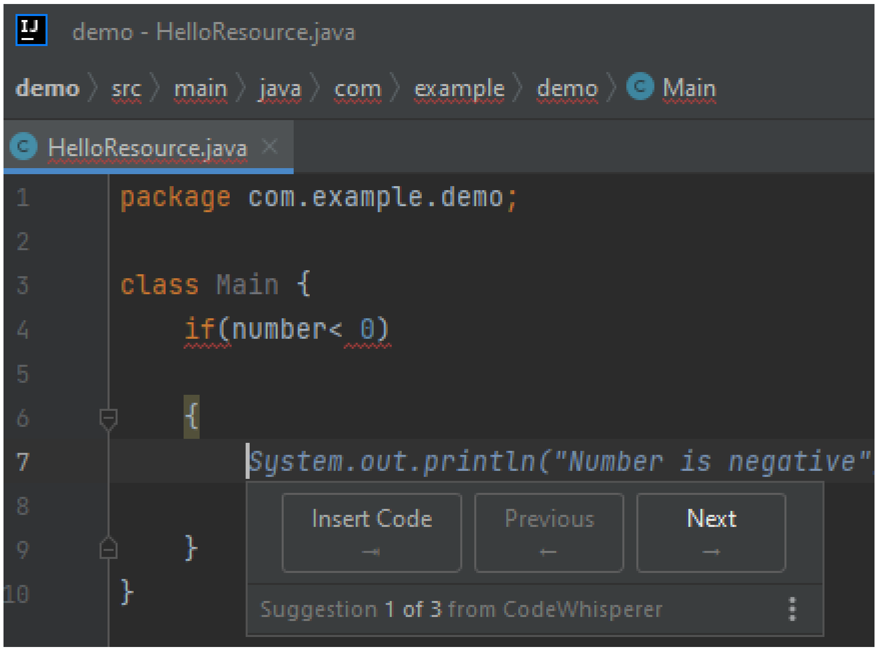
In the image below, using TypeScript and VS Code, CodeWhisperer recommends a way to complete the function.



```
import { S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

/**
 * Upload local file to bucket
 */
export const uploadFile = async (
```

AWS Toolkit for JetBrains

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

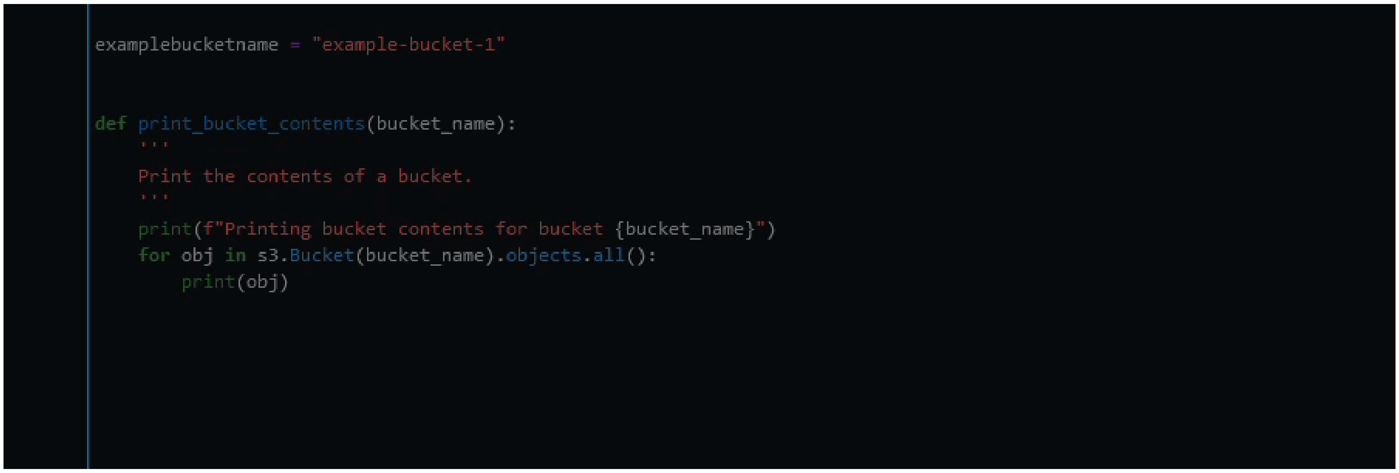

demo - HelloResource.java
demo > src > main > java > com > example > demo > Main
HelloResource.java
Insert Code
Previous
Next

```
package com.example.demo;

class Main {
    if(number< 0)

    {
        System.out.println("Number is negative"

    }
}
```

Suggestion 1 of 3 from CodeWhisperer

SageMaker Studio

In this example using Python and SageMaker Studio, CodeWhisperer recommends a block of code, based on the context.

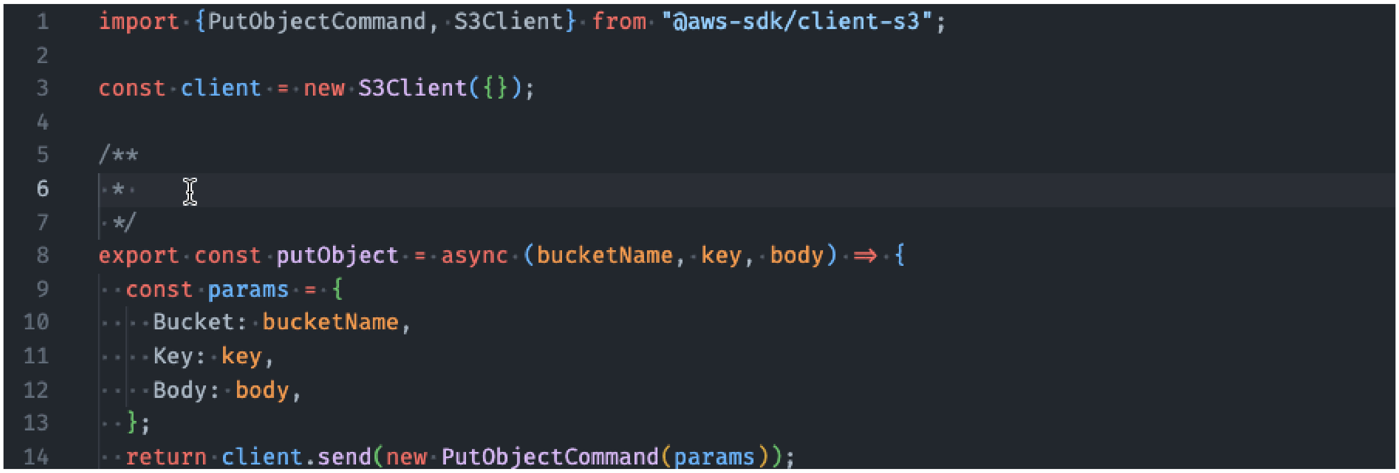


```
examplebucketname = "example-bucket-1"


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

Docstring, JSDoc, and Javadoc completion

Visual Studio Code

In this example using Javascript in VS Code, CodeWhisperer fills in JSDoc parameters based on existing constants.

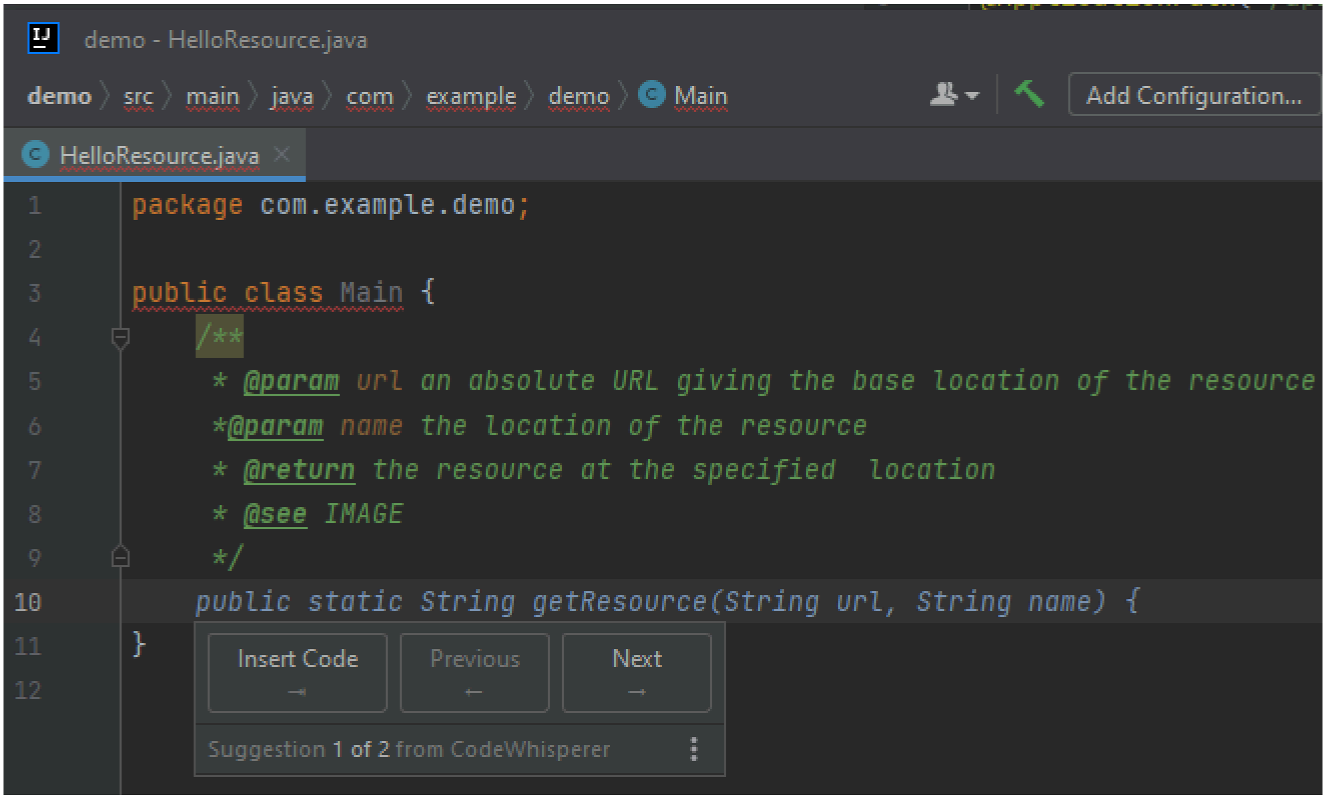


```
import {PutObjectCommand, S3Client} from "@aws-sdk/client-s3";

const client = new S3Client({});

/**
 * 
 */
export const putObject = async (bucketName, key, body) => {
  const params = {
    Bucket: bucketName,
    Key: key,
    Body: body,
  };
  return client.send(new PutObjectCommand(params));
```

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.


demo - HelloResource.java
demo 〉 src 〉 main 〉 java 〉 com 〉 example 〉 demo 〉 Main
Add Configuration...
HelloResource.java

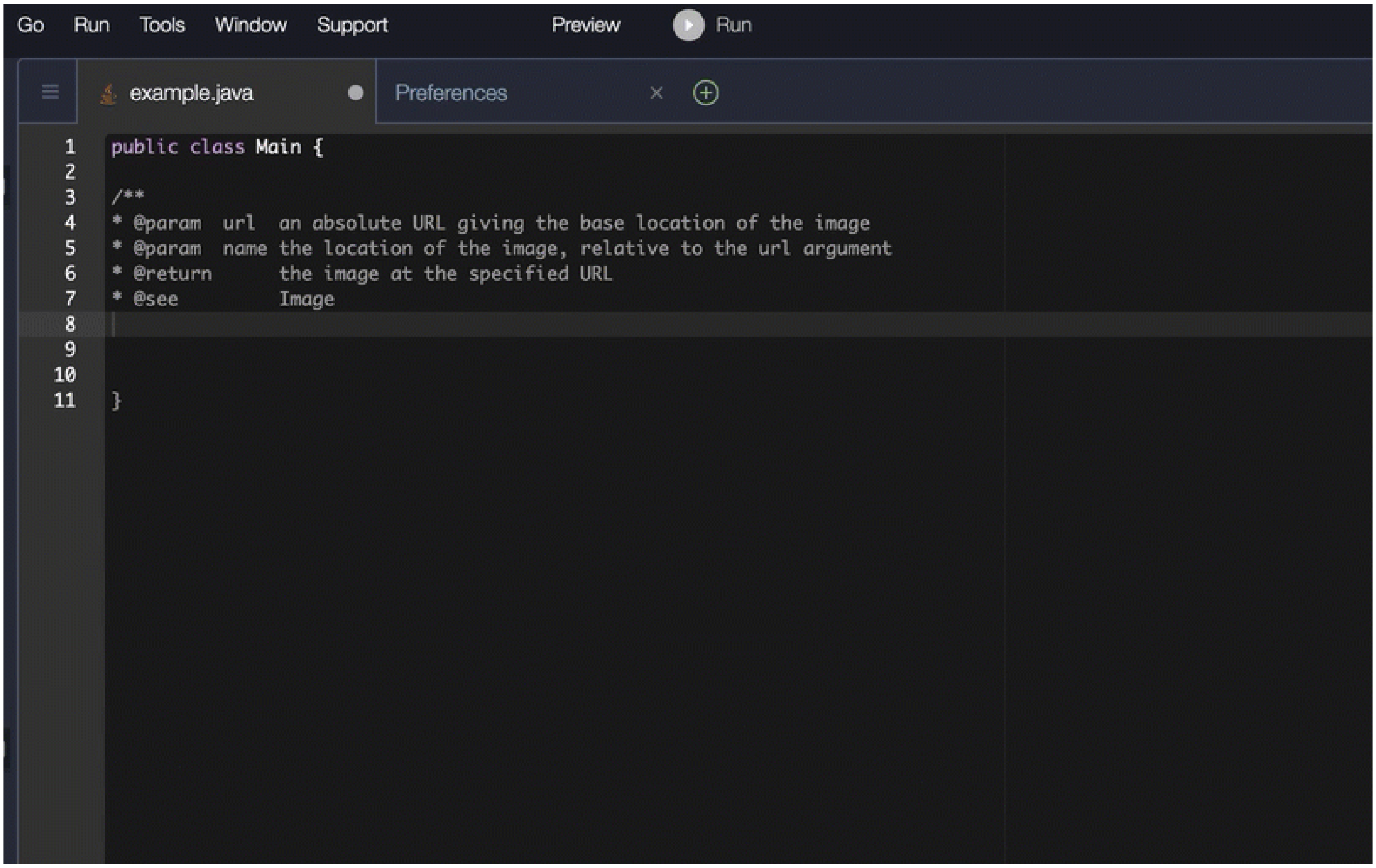
```
package com.example.demo;

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

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

AWS Cloud9

The following example is adapted from an example on the Oracle website.

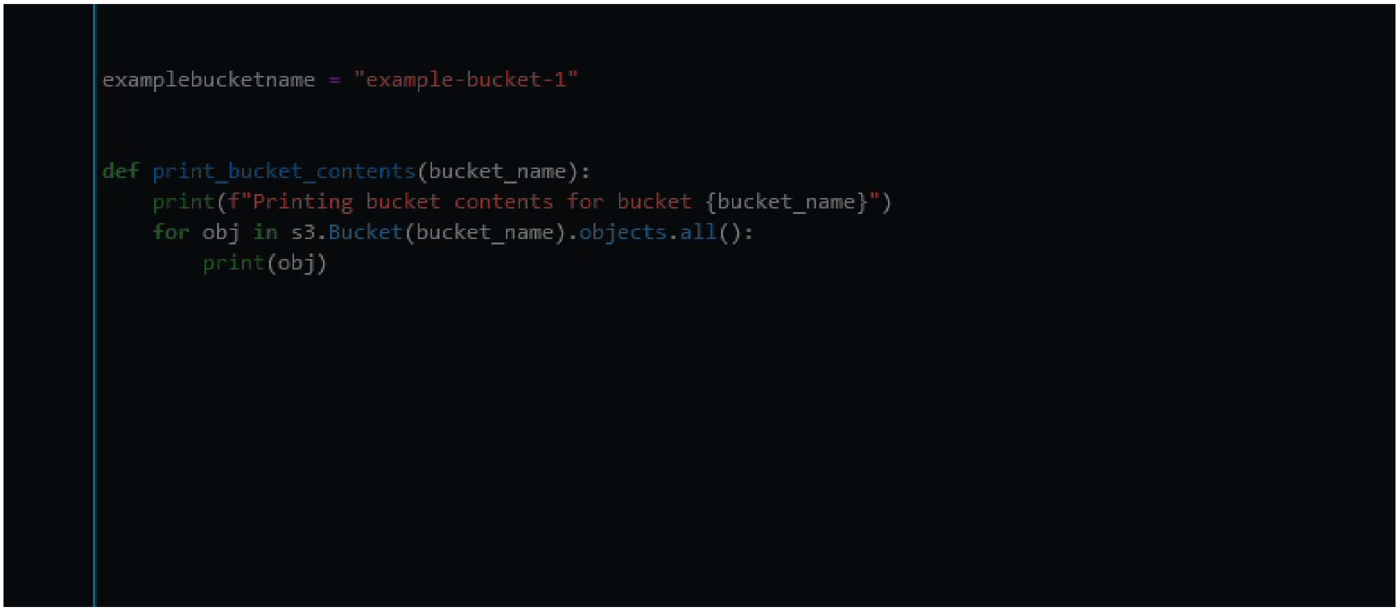
In the example below, in Java, the user enters a docstring. CodeWhisperer suggests a function to process the docstring.



```
public class Main {

/**
* @param  url  an absolute URL giving the base location of the image
* @param  name the location of the image, relative to the url argument
* @return      the image at the specified URL
* @see         Image



}
```

SageMaker Studio

In this example using Python and SageMaker Studio, CodeWhisperer recommends a Docstring, based on the surrounding context.



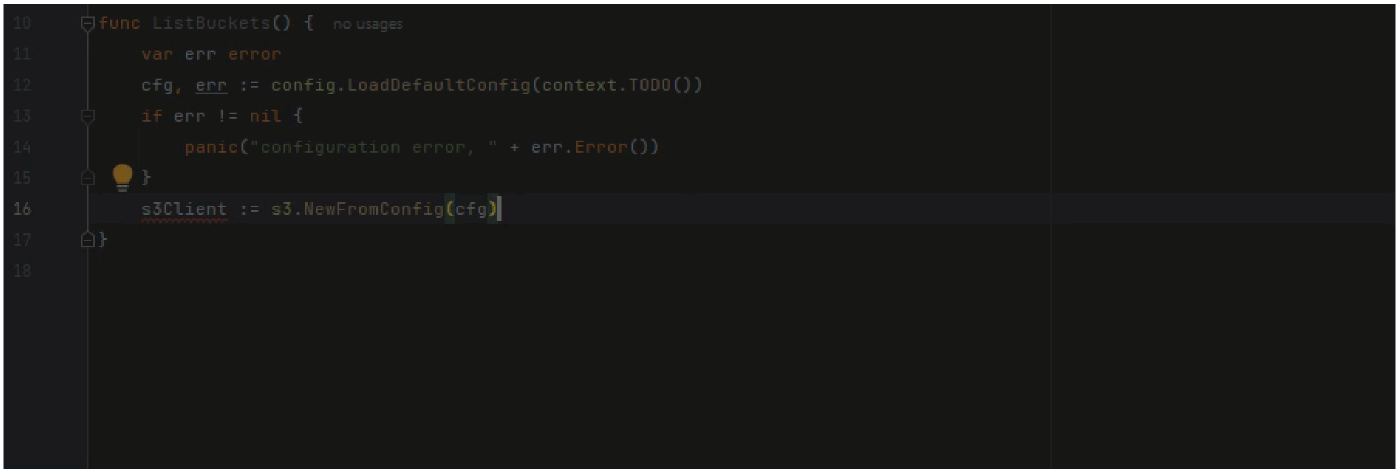
```
examplebucketname = "example-bucket-1"


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

Line-by-line recommendations

Depending on your use case, CodeWhisperer may not be able to generate an entire function block in one recommendation. However, CodeWhisperer can still provide line-by-line recommendations.

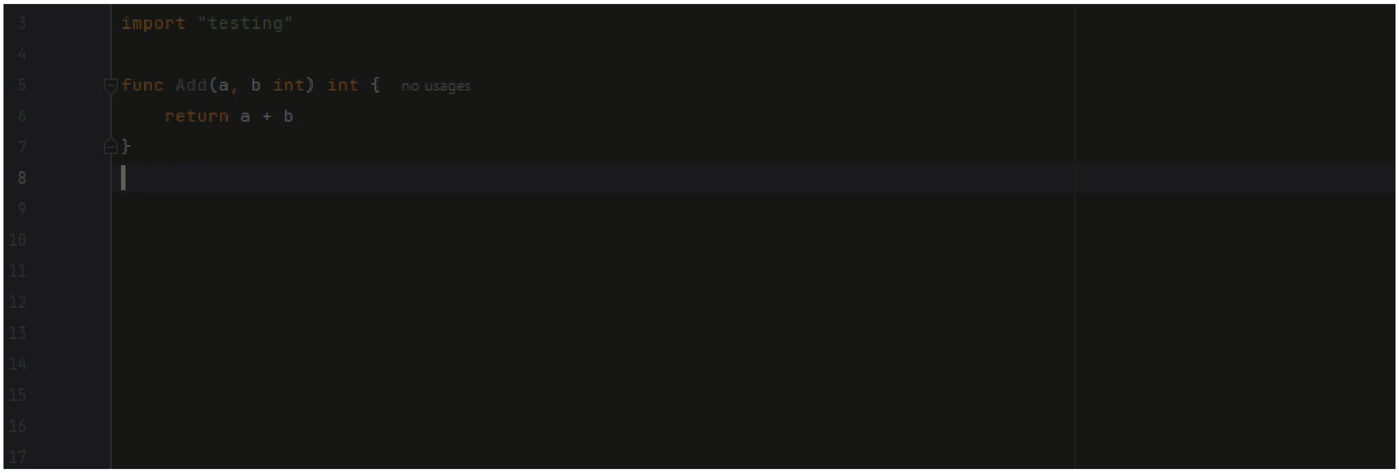
JetBrains

In this example using Go and GoLand, CodeWhisperer provides line-by-line recommendations.



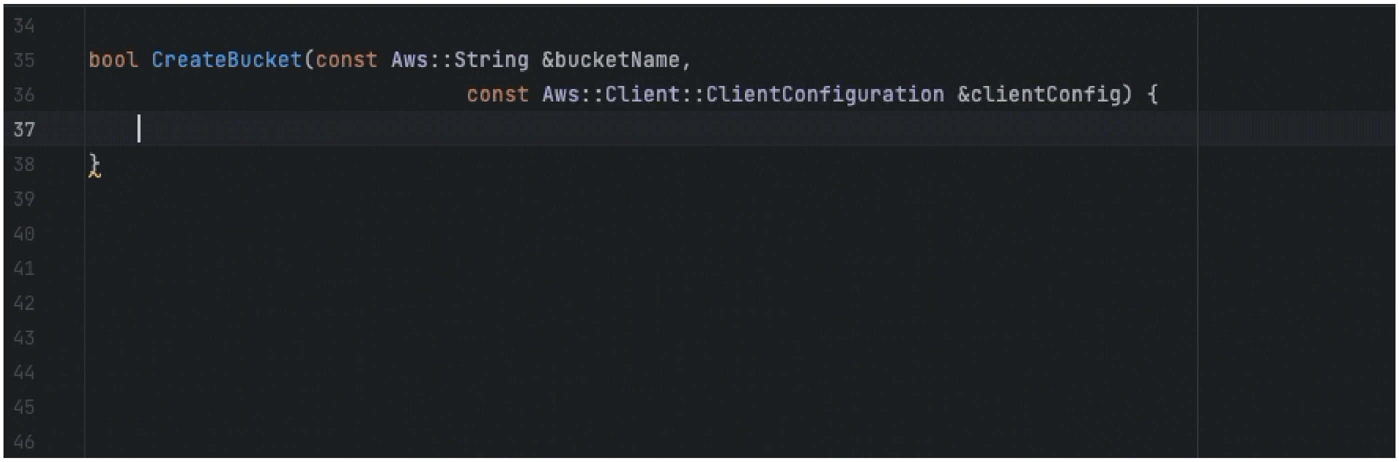
```
func ListBuckets() {
    var err error
    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
        panic("configuration error, " + err.Error())
    }
    s3Client := s3.NewFromConfig(cfg)
}
```

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



```
import "testing"

func Add(a, b int) int {
    return a + b
}

```

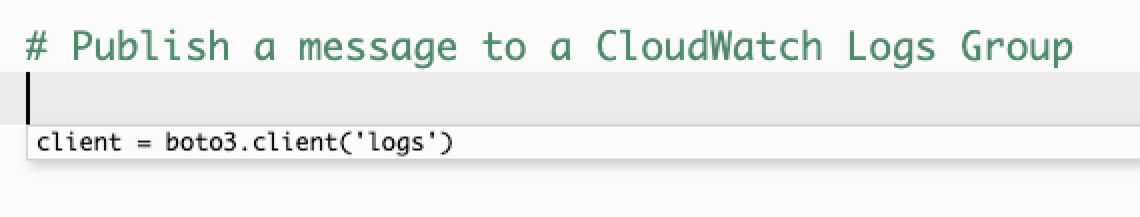
In this example, CodeWhisperer provides line-by-line recommendations using C++ and CLion.



```
bool CreateBucket(const Aws::String &bucketName,
                                    const Aws::Client::ClientConfiguration &clientConfig) {

}
```

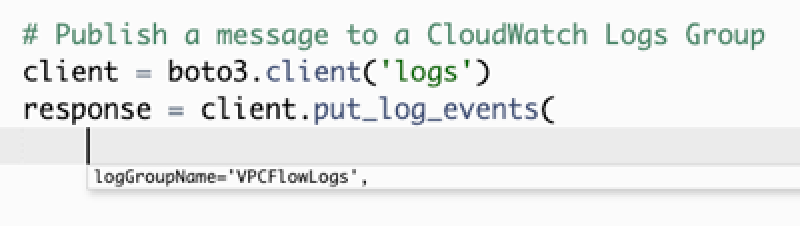
Lambda

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.



```
# Publish a message to a CloudWatch Logs Group

client = boto3.client('logs')
```

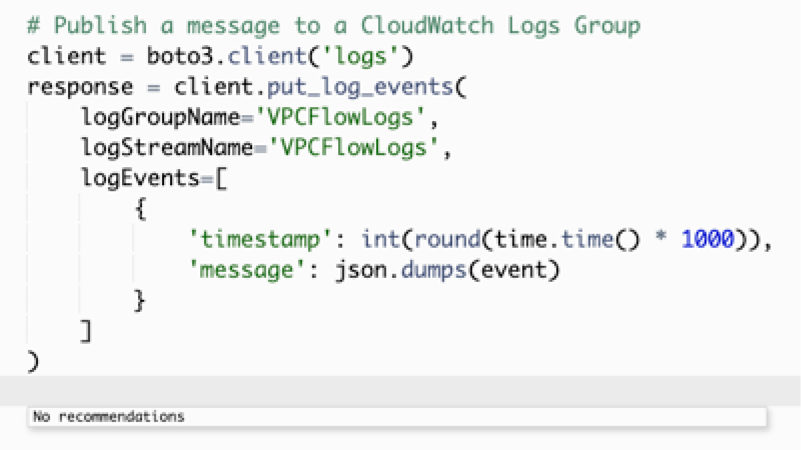
However, if the user continues to request line-by-line recommendations, CodeWhisperer also continues to suggest lines of code based on what's already been written.



```
# 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.

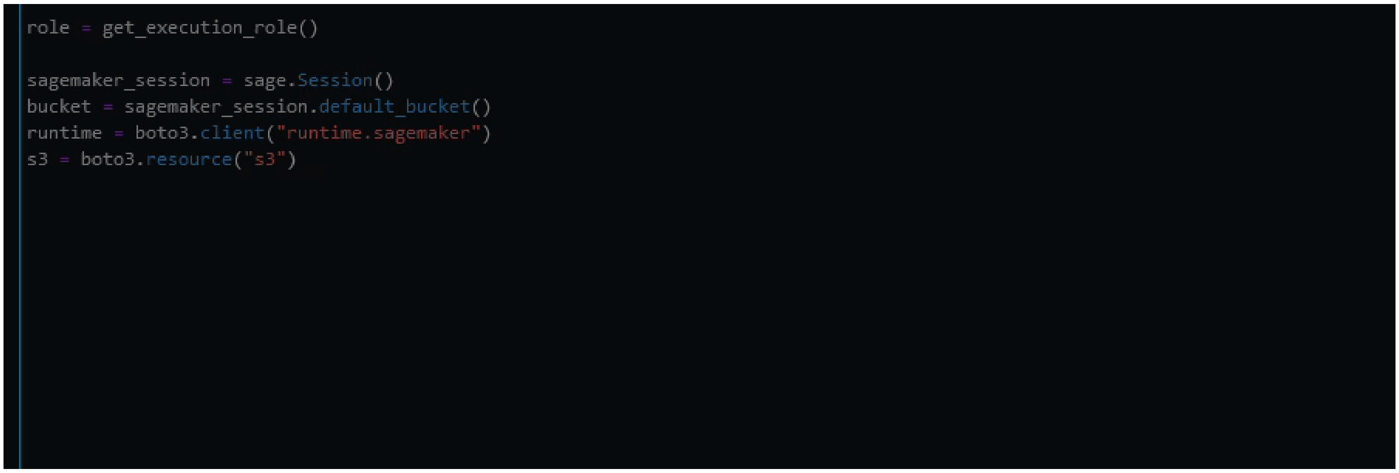


```
# Publish a message to a CloudWatch Logs Group
client = boto3.client('logs')
response = client.put_log_events(
    logGroupName='VPCFlowLogs',
    logStreamName='VPCFlowLogs',
    logEvents=[
        {
            'timestamp': int(round(time.time() * 1000)),
            'message': json.dumps(event)
        }
    ]
)
```

No recommendations

SageMaker Studio

In this example using Python and SageMaker Studio, CodeWhisperer provides recommendations, one line at at time.



```
role = get_execution_role()

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

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 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.

> **Note**
>
> The Customizations feature, which is in preview, is currently free. Pricing will become available at general availability.

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 in the IDE settings.
- You can run up to 500 security scans per month.

For pricing details, see the CodeWhisperer pricing page.

> **Note**
>
> Even if the same user acts as a CodeWhisperer developer in two different accounts within the same organization, your organization will only be billed for that user once per billing cycle.

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.

Billing for CodeWhisperer customizations

At the professional tier, users can take advantage of the new CodeWhisperer customization capability (in preview), which enables organizations to customize CodeWhisperer to generate more relevant recommendations by making it aware of an organization’s internal libraries, APIs, classes or methods.

During the preview, you can use the customization capability to create up to eight customizations based on their internal code bases. They can keep active up to two code customizations at the same time, for free. Pricing will become available at general availability

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.

> **Note**
>
> CloudWatch cannot be used to monitor activity at the individual tier, which does not require an AWS account.

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 CodeWhisperer 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.

Dimensions	Metric	Use case or explanation
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 lines generated by CodeWhisperer.
ProgrammingLanguage, SuggestionState, CompletionType	SuggestionReferenceCount	You want to determine the number of recommendation triggers with references that have taken place.
ProgrammingLanguage	CodeScanCount	You want to determine the number of code scans that have taken place.
ProgrammingLanguage	TotalCharacterCount	The number of characters in your file, including all suggestions from CodeWhisperer.

Dimensions	Metric	Use case or explanation
ProgrammingLanguage	CodeWhispererCharacterCount	The number of characters generated by CodeWhisperer.

To aggregate Invocations, use the Sum statistic.

To aggregate DailyActiveUserTrend, use the Sum statistic, and use "1 Day" as the period.

To aggregate SubscriptionCount, use the Sum statistic.

To aggregate MonthlyActiveUniqueUsers use the Sum statistic, and use "30 Days" as the period.

Tracking CodeWhisperer usage across your organization

Your business may operate many different AWS accounts that are all part of one AWS organization. In that case, you may want to create a separate CodeWhisperer instance for each of your AWS accounts. Then, you can assign a different CodeWhisperer administrator, and a different (or overlapping) set of developers to each account.

When a CodeWhisperer administrator views the dashboard, they will only see information about the account to which they have been assigned.

Billing for CodeWhisperer Professional usage is per AWS organization. If the same developer uses CodeWhisperer in multiple accounts within the same organization, you will only be billed for their seat in the first account in which they use the service.

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, and for in-console development, CodeWhisperer does not collect your content for service improvement purposes.** In-console development includes development in Amazon SageMaker Studio, AWS Glue Studio, AWS Lambda console, and AWS Cloud9. The professional tier, in this context, includes the code chat, feature development, and code transformation features of Amazon Q.

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

Toolkit for Visual Studio

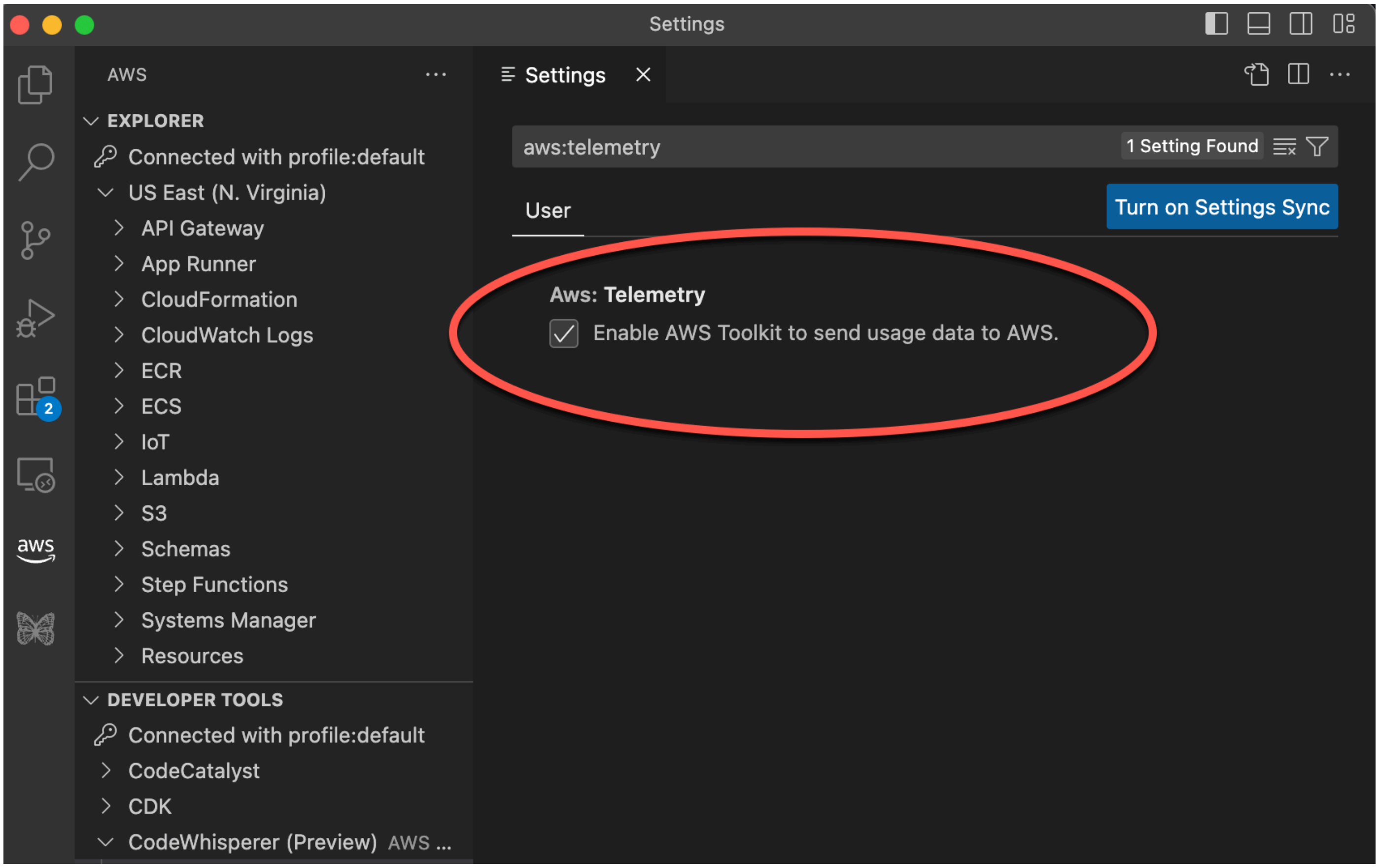
To opt out of sharing your telemetry data in Toolkit for Visual Studio, use this procedure:

1. Under **Tools**, select **Options**.
2. Select **AWS Toolkit**.
3. Under **Toolkit Usage**, 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 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.



aws:telemetry — 1 Setting Found

User

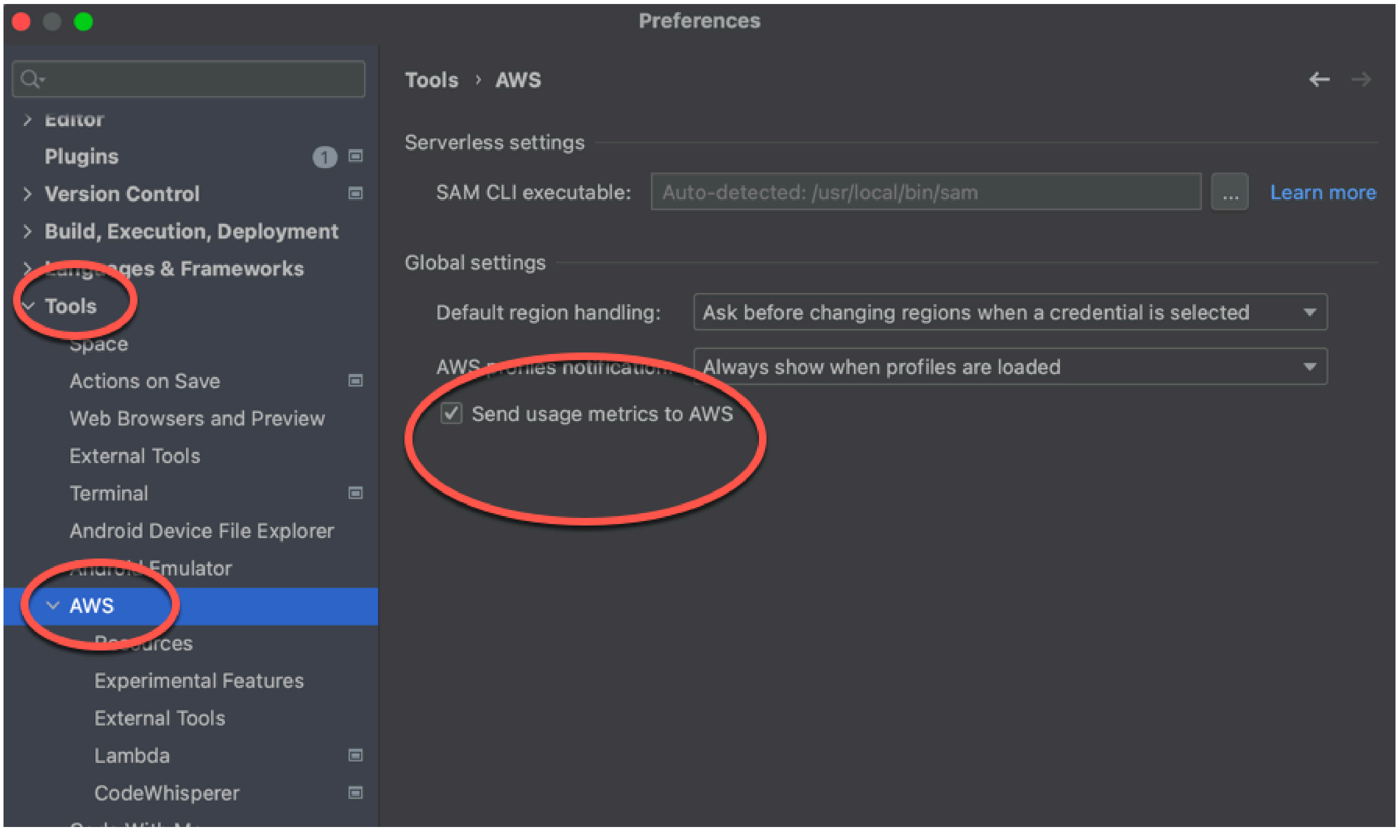
Aws: Telemetry

☑ Enable AWS Toolkit to send usage data to AWS.

> **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.



Serverless settings

SAM CLI executable: Auto-detected: /usr/local/bin/sam

Global settings

Default region handling: Ask before changing regions when a credential is selected

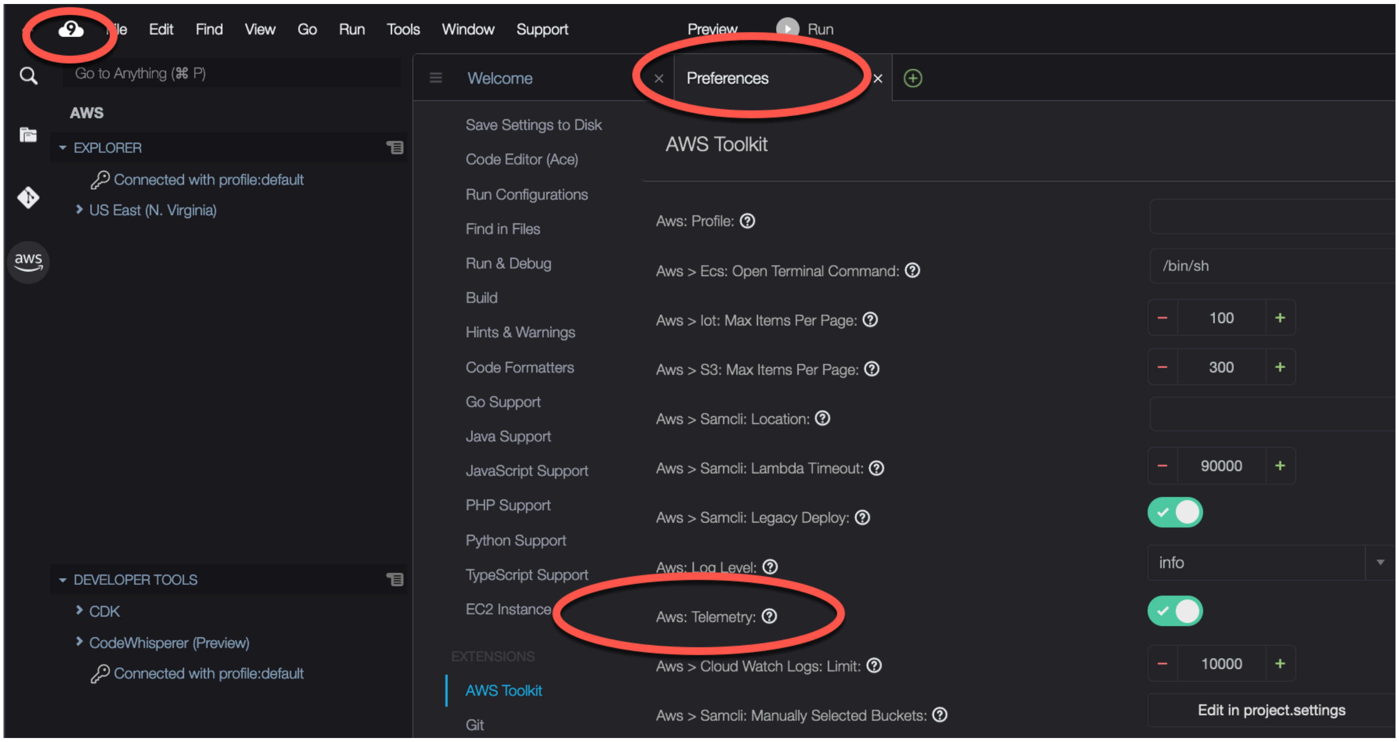
AWS profiles notification: Always show when profiles are loaded

☑ Send usage metrics to AWS

> **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.



Aws: Profile:

Aws > Ecs: Open Terminal Command: /bin/sh

Aws > Iot: Max Items Per Page: 100

Aws > S3: Max Items Per Page: 300

Aws > Samcli: Location:

Aws > Samcli: Lambda Timeout: 90000

Aws > Samcli: Legacy Deploy: ✓

Aws: Log Level: info

Aws: Telemetry: ✓

Aws > Cloud Watch Logs: Limit: 10000

Aws > Samcli: Manually Selected Buckets: Edit in project.settings

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

Lambda

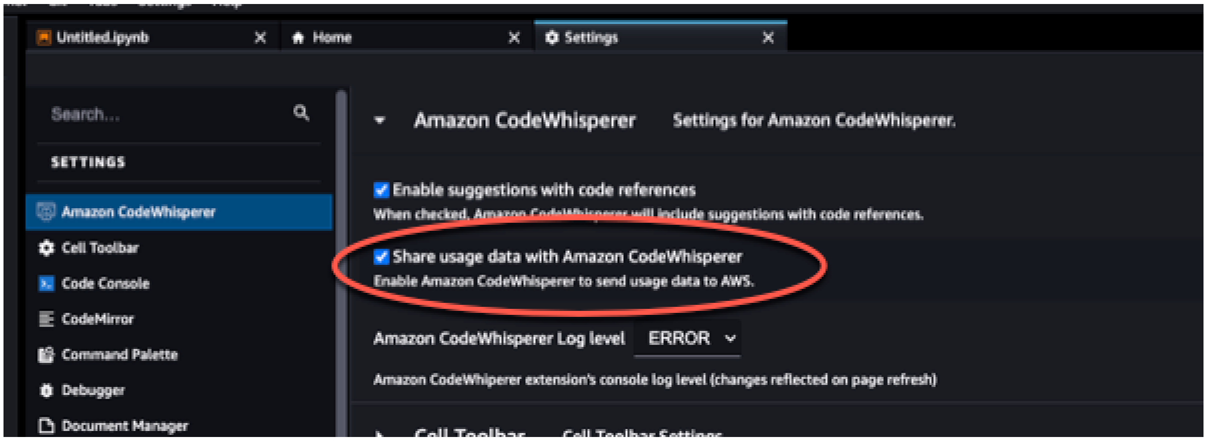
When you use CodeWhisperer with Lambda, CodeWhisperer does not share your client-side telemetry with AWS.

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**.



Amazon CodeWhisperer Settings for Amazon CodeWhisperer.

☑ Enable suggestions with code references

When checked, Amazon CodeWhisperer will include suggestions with code references.

☑ Share usage data with Amazon CodeWhisperer

Enable Amazon CodeWhisperer to send usage data to AWS.

Amazon CodeWhisperer Log level ERROR

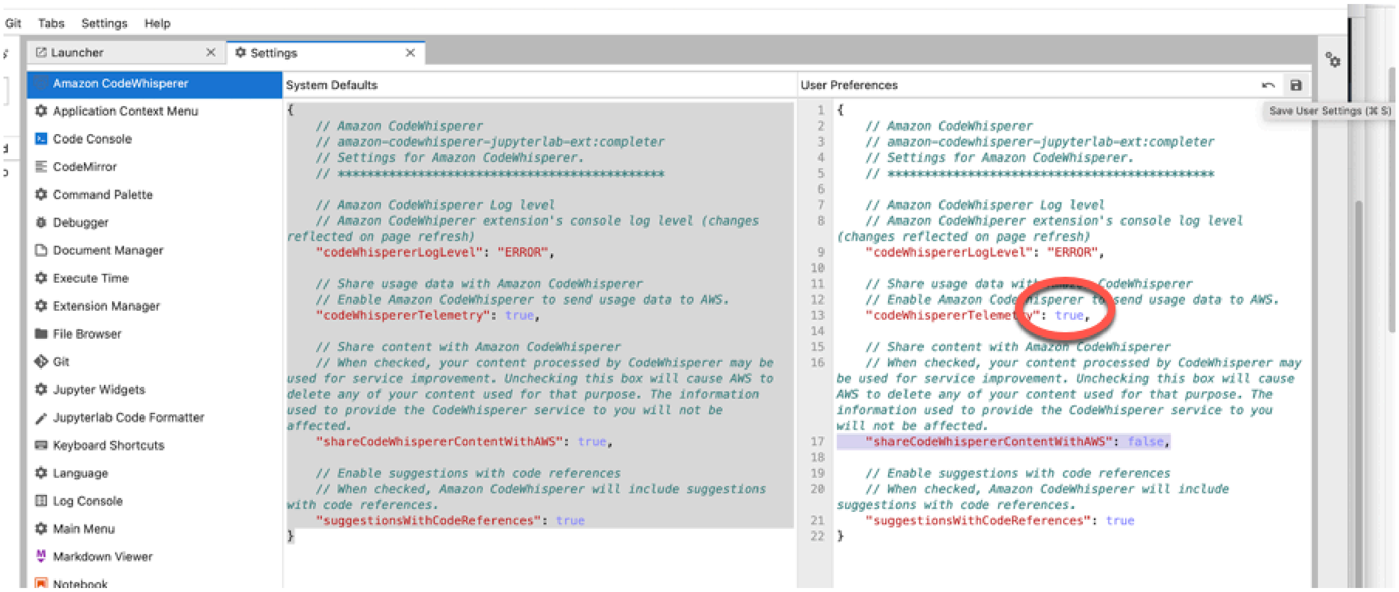
Amazon CodeWhiperer extension's console log level (changes reflected on page refresh)

Amazon EMR Studio

1. From the top of the Amazon EMR Studio window choose **Settings**.

2. From the **Settings** dropdown, choose **Advanced Settings Editor**.

3. Select **Amazon CodeWhisperer** in the dropdown. Set the value for `codeWhispererTelemetry` to **true** or **false**.



System Defaults

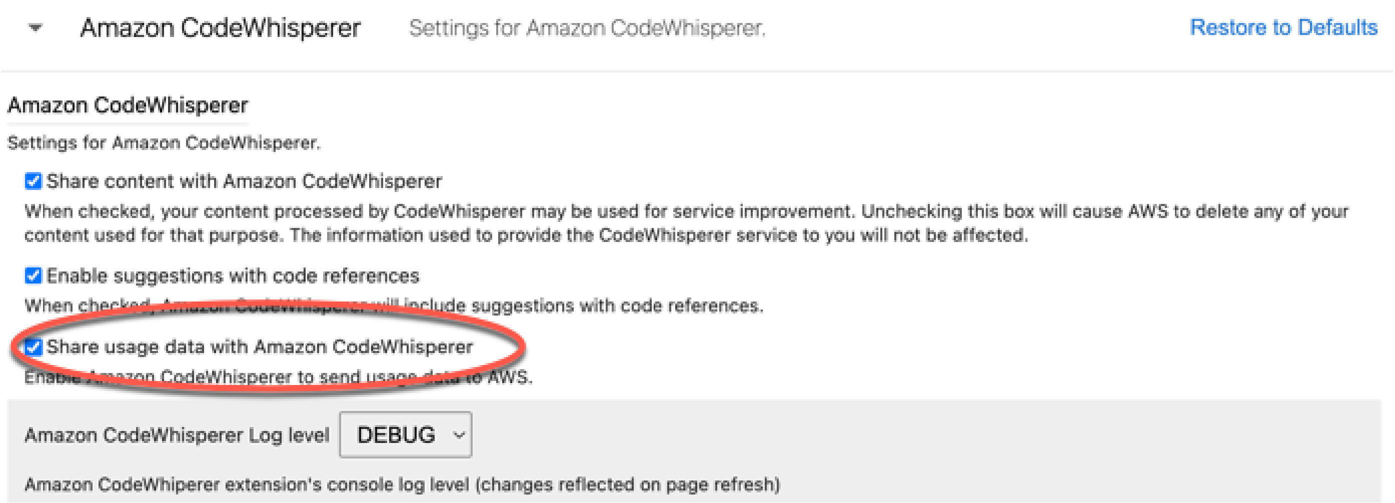
```
{
    // Amazon CodeWhisperer
    // amazon-codewhisperer-jupyterlab-ext:completer
    // Settings for Amazon CodeWhisperer.
    // *********************************************

    // Amazon CodeWhisperer Log level
    // Amazon CodeWhiperer extension's console log level (changes
reflected on page refresh)
    "codeWhispererLogLevel": "ERROR",

    // Share usage data with Amazon CodeWhisperer
    // Enable Amazon CodeWhisperer to send usage data to AWS.
    "codeWhispererTelemetry": true,

    // Share content with Amazon CodeWhisperer
    // When checked, your content processed by CodeWhisperer may be
used for service improvement. Unchecking this box will cause AWS to
delete any of your content used for that purpose. The information
used to provide the CodeWhisperer service to you will not be
affected.
    "shareCodeWhispererContentWithAWS": true,

    // Enable suggestions with code references
    // When checked, Amazon CodeWhisperer will include suggestions
with code references.
    "suggestionsWithCodeReferences": true
}
```

User Preferences

```
{
    // Amazon CodeWhisperer
    // amazon-codewhisperer-jupyterlab-ext:completer
    // Settings for Amazon CodeWhisperer.
    // *********************************************

    // Amazon CodeWhisperer Log level
    // Amazon CodeWhiperer extension's console log level
(changes reflected on page refresh)
    "codeWhispererLogLevel": "ERROR",

    // Share usage data with Amazon CodeWhisperer
    // Enable Amazon CodeWhisperer to send usage data to AWS.
    "codeWhispererTelemetry": true,

    // Share content with Amazon CodeWhisperer
    // When checked, your content processed by CodeWhisperer may
be used for service improvement. Unchecking this box will cause
AWS to delete any of your content used for that purpose. The
information used to provide the CodeWhisperer service to you
will not be affected.
    "shareCodeWhispererContentWithAWS": false,

    // Enable suggestions with code references
    // When checked, Amazon CodeWhisperer will include
suggestions with code references.
    "suggestionsWithCodeReferences": true
}
```

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**.



Amazon CodeWhisperer

Settings for Amazon CodeWhisperer.

☑ Share content with Amazon CodeWhisperer

When checked, your content processed by CodeWhisperer may be used for service improvement. Unchecking this box will cause AWS to delete any of your content used for that purpose. The information used to provide the CodeWhisperer service to you will not be affected.

☑ Enable suggestions with code references

When checked, Amazon CodeWhisperer will include suggestions with code references.

☑ Share usage data with Amazon CodeWhisperer

Enable Amazon CodeWhisperer to send usage data to AWS.

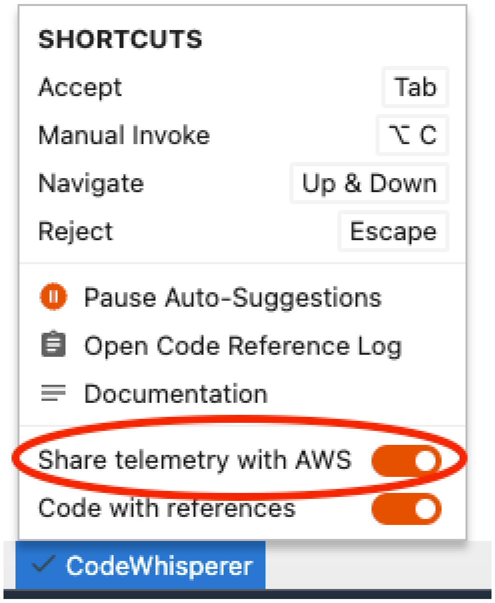
Amazon CodeWhisperer Log level: DEBUG

Amazon CodeWhiperer extension's console log level (changes reflected on page refresh)

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.

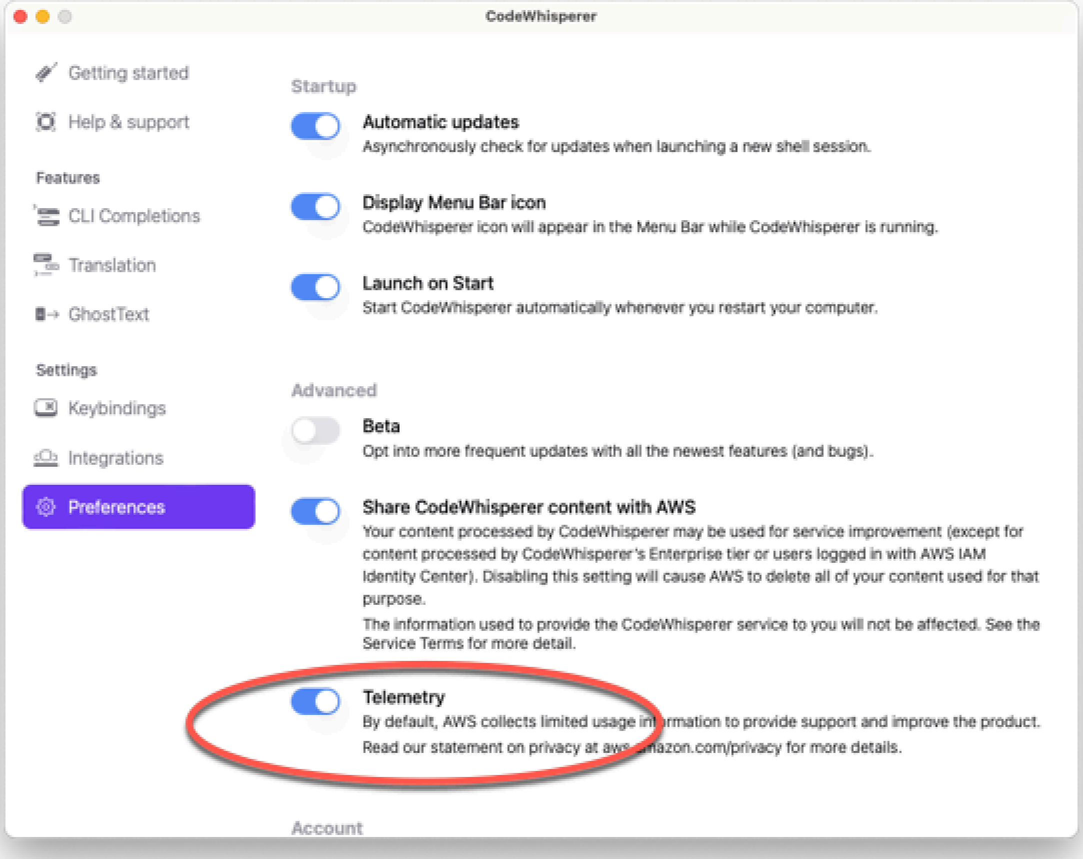


Share telemetry with AWS ●

Code with references ●

Command line

In the command line tool, under **Preferences**, toggle **Telemetry**.



Startup

● **Automatic updates**
Asynchronously check for updates when launching a new shell session.

● **Display Menu Bar icon**
CodeWhisperer icon will appear in the Menu Bar while CodeWhisperer is running.

● **Launch on Start**
Start CodeWhisperer automatically whenever you restart your computer.

Advanced

○ **Beta**
Opt into more frequent updates with all the newest features (and bugs).

● **Share CodeWhisperer content with AWS**
Your content processed by CodeWhisperer may be used for service improvement (except for content processed by CodeWhisperer's Enterprise tier or users logged in with AWS IAM Identity Center). Disabling this setting will cause AWS to delete all of your content used for that purpose.
The information used to provide the CodeWhisperer service to you will not be affected. See the Service Terms for more detail.

● **Telemetry**
By default, AWS collects limited usage information to provide support and improve the product.
Read our statement on privacy at aws.amazon.com/privacy for more details.

Account

Opting out of sharing your content

Toolkit for Visual Studio

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

At the individual tier, to opt out of sharing your content in Visual Studio, use the following procedure.

Bring up the CodeWhisperer options menu one of two ways:

- Choose the CodeWhisperer icon from the edge of the window, then select Options...
- Go to Tools -> Options -> AWS Toolkit -> CodeWhisperer

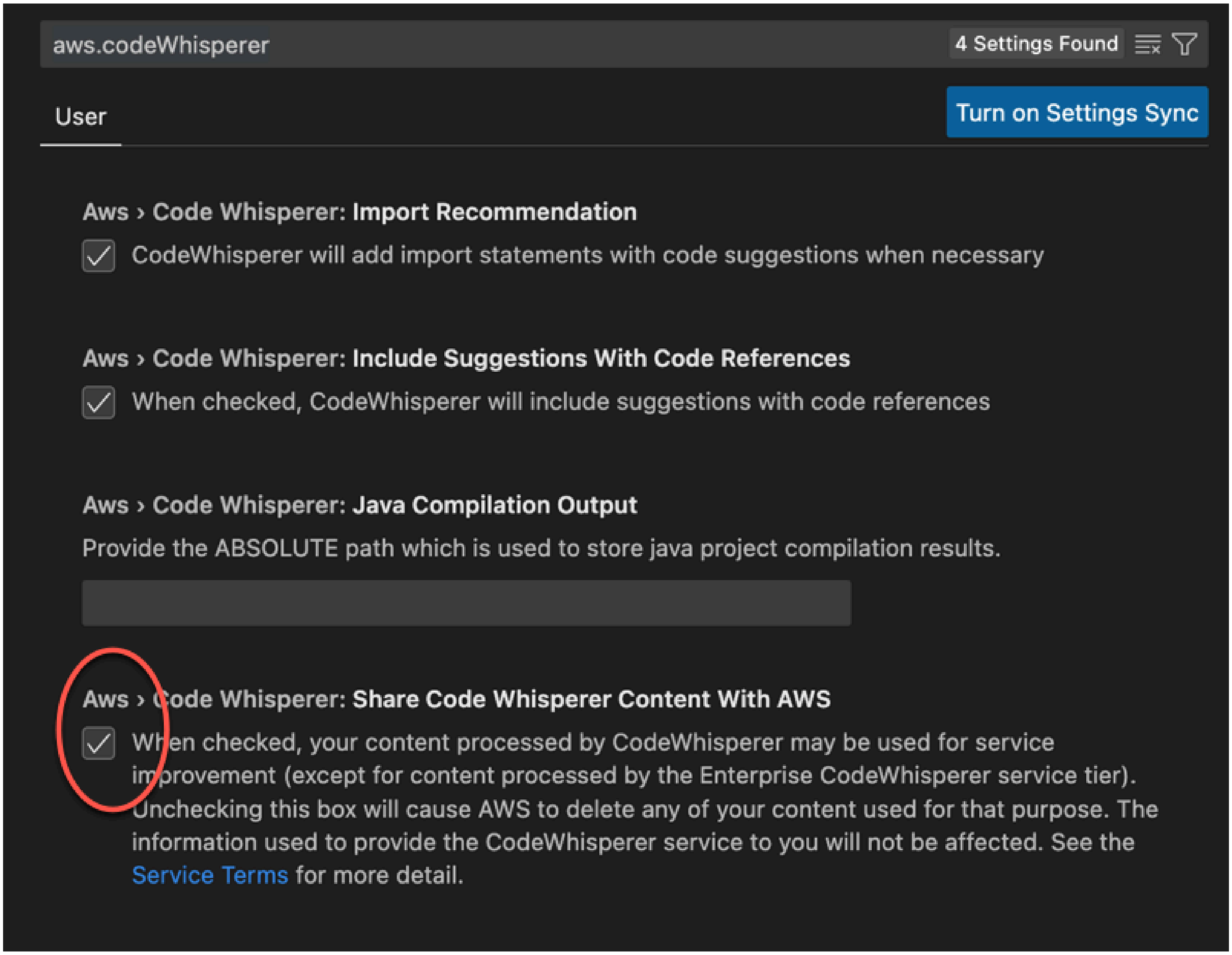
Then toggle **Share CodeWhisperer Content with AWS** to **True** or **False**.

AWS Toolkit for Visual Studio Code

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

At the individual tier, 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 › Code Whisperer: **Import Recommendation**

☑ CodeWhisperer will add import statements with code suggestions when necessary

Aws › Code Whisperer: **Include Suggestions With Code References**

☑ When checked, CodeWhisperer will include suggestions with code references

Aws › Code Whisperer: **Java Compilation Output**

Provide the ABSOLUTE path which is used to store java project compilation results.

Aws › Code Whisperer: **Share Code Whisperer Content With AWS**

☑ When checked, your content processed by CodeWhisperer may be used for service improvement (except for content processed by the Enterprise CodeWhisperer service tier). Unchecking this box will cause AWS to delete any of your content used for that purpose. The information used to provide the CodeWhisperer service to you will not be affected. See the Service Terms for more detail.

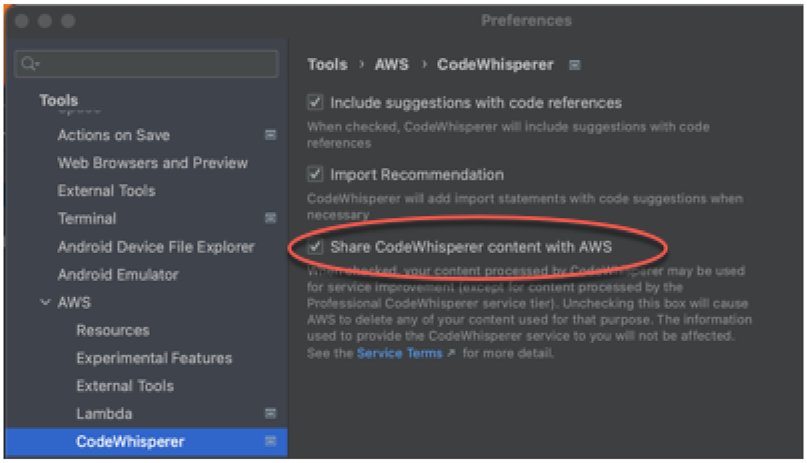
AWS Toolkit for JetBrains

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**.



☑ Include suggestions with code references

When checked, CodeWhisperer will include suggestions with code references

☑ Import Recommendation

CodeWhisperer will add import statements with code suggestions when necessary

☑ Share CodeWhisperer content with AWS

When checked, your content processed by CodeWhisperer may be used for service improvement (except for content processed by the Professional CodeWhisperer service tier). Unchecking this box will cause AWS to delete any of your content used for that purpose. The information used to provide the CodeWhisperer service to you will not be affected. See the Service Terms ↗ for more detail.

AWS Cloud 9

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

> **Note**
>
> The AWS Cloud 9 settings do contain a toggle switch for sharing CodeWhisperer content with AWS. But that switch is non-functional.

Lambda

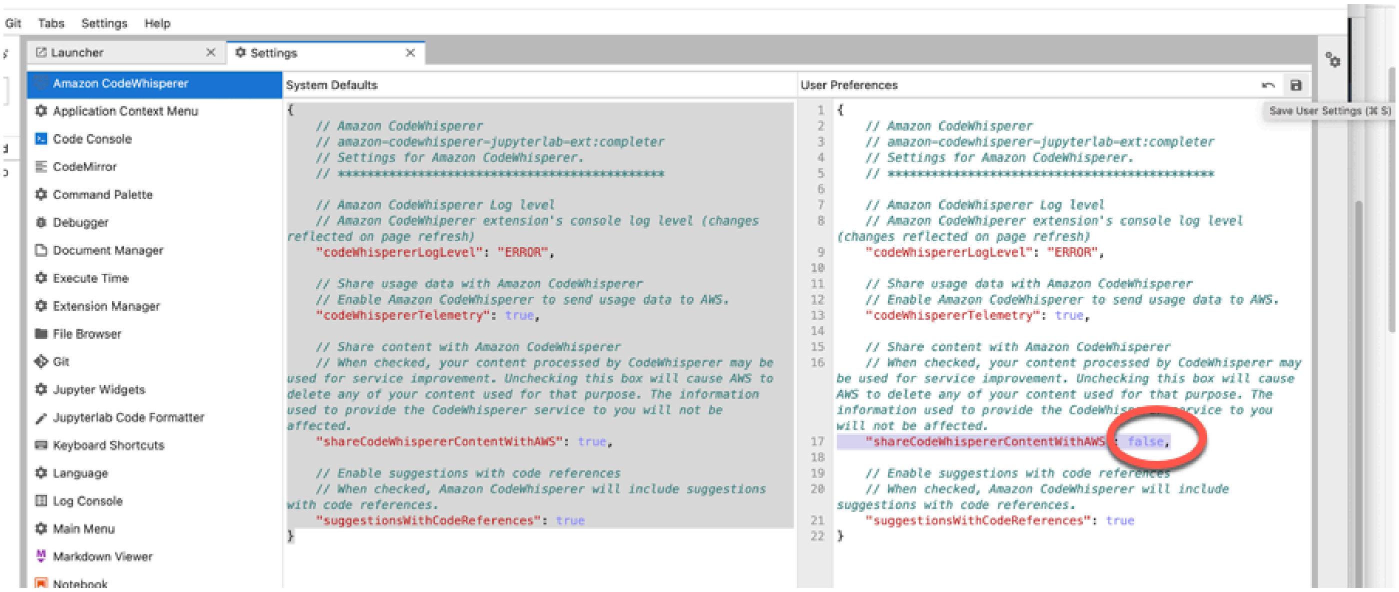
When you use CodeWhisperer with Lambda, CodeWhisperer does not share your content with AWS.

SageMaker Studio

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

Amazon EMR Studio

1. From the top of the Amazon EMR Studio window choose **Settings**.
2. From the **Settings** dropdown, choose **Advanced Settings Editor**.
3. Select **Amazon CodeWhisperer** in the dropdown. Set the value for `shareCodedeWhispererContentWithAWS` to **true** or **false**.



System Defaults

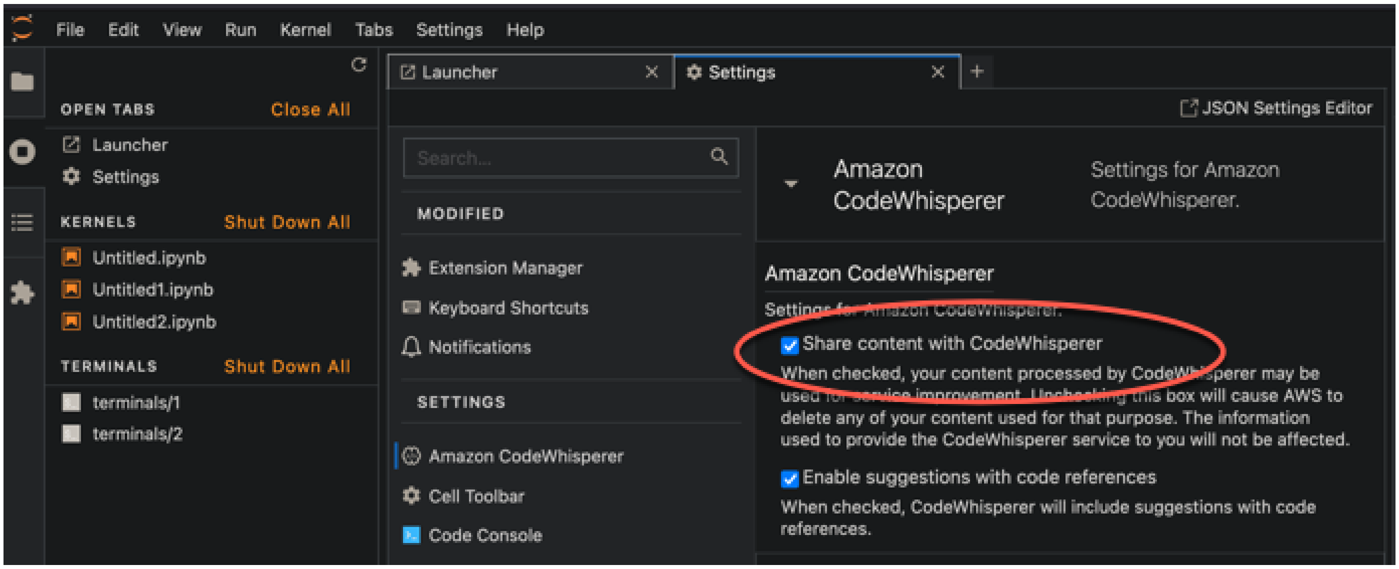
```
{
    // Amazon CodeWhisperer
    // amazon-codewhisperer-jupyterlab-ext:completer
    // Settings for Amazon CodeWhisperer.
    // ********************************************

    // Amazon CodeWhisperer Log level
    // Amazon CodeWhiperer extension's console log level (changes
reflected on page refresh)
    "codeWhispererLogLevel": "ERROR",

    // Share usage data with Amazon CodeWhisperer
    // Enable Amazon CodeWhisperer to send usage data to AWS.
    "codeWhispererTelemetry": true,

    // Share content with Amazon CodeWhisperer
    // When checked, your content processed by CodeWhisperer may be
used for service improvement. Unchecking this box will cause AWS to
delete any of your content used for that purpose. The information
used to provide the CodeWhisperer service to you will not be
affected.
    "shareCodeWhispererContentWithAWS": true,

    // Enable suggestions with code references
    // When checked, Amazon CodeWhisperer will include suggestions
with code references.
    "suggestionsWithCodeReferences": true
}
```

User Preferences

```
{
    // Amazon CodeWhisperer
    // amazon-codewhisperer-jupyterlab-ext:completer
    // Settings for Amazon CodeWhisperer.
    // ********************************************

    // Amazon CodeWhisperer Log level
    // Amazon CodeWhiperer extension's console log level
(changes reflected on page refresh)
    "codeWhispererLogLevel": "ERROR",

    // Share usage data with Amazon CodeWhisperer
    // Enable Amazon CodeWhisperer to send usage data to AWS.
    "codeWhispererTelemetry": true,

    // Share content with Amazon CodeWhisperer
    // When checked, your content processed by CodeWhisperer may
be used for service improvement. Unchecking this box will cause
AWS to delete any of your content used for that purpose. The
information used to provide the CodeWhisperer service to you
will not be affected.
    "shareCodeWhispererContentWithAWS": false,

    // Enable suggestions with code references
    // When checked, Amazon CodeWhisperer will include
suggestions with code references.
    "suggestionsWithCodeReferences": true
}
```

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**.



Amazon CodeWhisperer Settings for Amazon CodeWhisperer.

Amazon CodeWhisperer

Settings for Amazon CodeWhisperer.

☑ Share content with CodeWhisperer

When checked, your content processed by CodeWhisperer may be used for service improvement. Unchecking this box will cause AWS to delete any of your content used for that purpose. The information used to provide the CodeWhisperer service to you will not be affected.

☑ Enable suggestions with code references

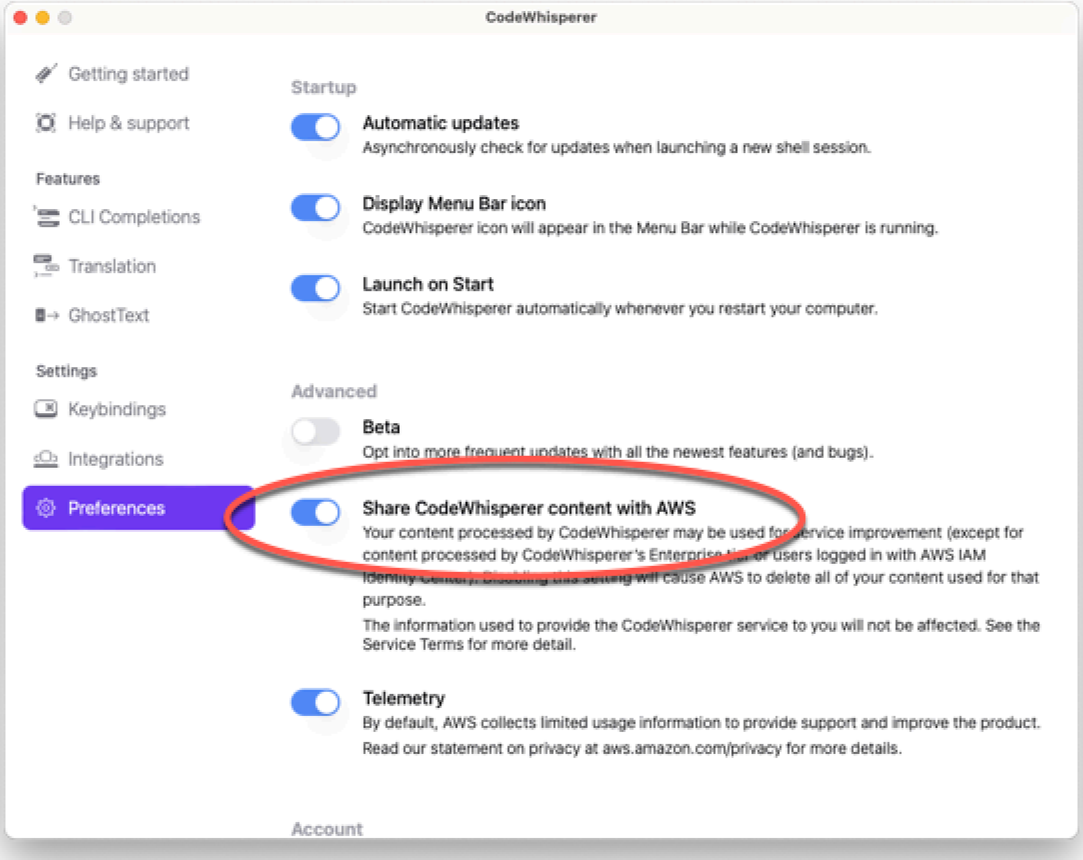
When checked, CodeWhisperer will include suggestions with code references.

AWS Glue Studio Notebook

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

Command line

In the command line tool, under **Preferences**, toggle **Share CodeWhisperer content with AWS**.



Startup

☑ **Automatic updates**
Asynchronously check for updates when launching a new shell session.

☑ **Display Menu Bar icon**
CodeWhisperer icon will appear in the Menu Bar while CodeWhisperer is running.

☑ **Launch on Start**
Start CodeWhisperer automatically whenever you restart your computer.

Advanced

☐ **Beta**
Opt into more frequent updates with all the newest features (and bugs).

☑ **Share CodeWhisperer content with AWS**
Your content processed by CodeWhisperer may be used for service improvement (except for content processed by CodeWhisperer's Enterprise tier or users logged in with AWS IAM Identity Center). Disabling this setting will cause AWS to delete all of your content used for that purpose.

The information used to provide the CodeWhisperer service to you will not be affected. See the Service Terms for more detail.

☑ **Telemetry**
By default, AWS collects limited usage information to provide support and improve the product.
Read our statement on privacy at aws.amazon.com/privacy for more details.

Account

Quotas for Amazon CodeWhisperer

CodeWhisperer does not maintain any service quotas.

To learn about the differences in usage available per service tier, see ???.

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

- Data protection for Amazon CodeWhisperer
- Compliance validation for Amazon CodeWhisperer
- Security best practices in Amazon CodeWhisperer
- Infrastructure security in Amazon CodeWhisperer
- Identity and Access Management for Amazon CodeWhisperer
- Amazon CodeWhisperer and interface VPC endpoints (AWS PrivateLink)

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 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

CodeWhisperer sends events to CloudTrail. The API calls are:

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

Your data will not be logged in CloudTrail. This includes both your content and your client-side telemetry.

To learn more about how these APIs may be called from the console, and related IAM permissions, see ???.

For explanations of specific APIs, see ???.

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 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.*

Data protection and CodeWhisperer customizations

When you create a customization, AWS protects your code files.

CodeWhisperer uploads your files to a CodeWhisperer-owned Amazon S3 bucket. Your files are encrypted in transit with HTTPS and TLS. They are encrypted at rest using a AWS KMS key, either supplied by you or, if you do not supply one, by AWS. Once your customization has been created, AWS permanently deletes your data from the bucket, and purges it from memory.

Your customizations are fully isolated from each other within your account. They are also isolated from the data of other customers.

Only users specified by the CodeWhisperer administrator have access to any specific customization. And before the CodeWhisperer administrator can specify which users can access which customizations, you must authorize that administrator permission to do so.

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.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- 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.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.

- 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.

For information on best practices in infrastructure security, see Infrastructure security in Amazon CodeWhisperer.

Infrastructure security in Amazon CodeWhisperer

As a managed service, Amazon CodeWhisperer is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well‐Architected Framework*.

You use AWS published API calls to access CodeWhisperer through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

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

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

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.

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.

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.

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 (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 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 *IAM User Guide*.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. 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 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 Cross account resource access in IAM 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.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.
  - **Service 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	Yes
Resource-based policies	Yes
Policy actions	Partial
Policy resources	No
Policy condition keys (service-specific)	No
ACLs	No
ABAC (tags in policies)	Yes
Temporary credentials	Yes
Principal permissions	Yes
Service roles	Yes
Service-linked roles	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.

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 Cross account resource access in IAM 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.

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.

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.

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 forward access sessions (FAS)	Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

Service roles for 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

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: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 that allows CodeWhisperer to perform actions on your behalf. For more information, see Using service-linked roles for CodeWhisperer.

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",
                 "sso:DescribeApplication"
            ],
            "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.

Change	Description	Date
Updated AWSServiceRoleForCodeWhispererPolicy	Added DescribeApplication permission, which allows you	March 29, 2024

Change	Description	Date
	to retrieve information about CodeWhisperer.	
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	Added a new policy to allow CodeWhisperer to call CloudWatch and CodeGuru on your behalf. This policy is added to the AWSServiceRoleForCodeWhisperer that is created when you onboard to Amazon CodeWhisperer.	March 29, 2023
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
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my CodeWhisperer resources

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.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Using service-linked roles for 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.

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 *
- Action: `sso:DescribeApplication` 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 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. For more information, see AWS Regions and endpoints.

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

Region name	Region identity	Support in CodeWhisperer
Asia Pacific (Seoul)	ap-northeast-2	No
Asia Pacific (Singapore)	ap-southeast-1	No
Asia Pacific (Sydney)	ap-southeast-2	No
Asia Pacific (Tokyo)	ap-northeast-1	No
Canada (Central)	ca-central-1	No
Europe (Frankfurt)	eu-central-1	No
Europe (Ireland)	eu-west-1	No
Europe (London)	eu-west-2	No
Europe (Milan)	eu-south-1	No
Europe (Paris)	eu-west-3	No
Europe (Stockholm)	eu-north-1	No
Middle East (Bahrain)	me-south-1	No
Middle East (UAE)	me-central-1	No
South America (São Paulo)	sa-east-1	No
AWS GovCloud (US-East)	us-gov-east-1	No
AWS GovCloud (US-West)	us-gov-west-1	No

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.
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-premesis 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.
2. Set up CodeWhisperer with the in-console IDE.
   - AWS Cloud9
   - SageMaker Studio
   - AWS Glue Studio
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.
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
Merge with Amazon Q Developer	CodeWhisperer is now a part of Amazon Q Developer.	April 30, 2024
DescribeApplication permission added	The service-linked role and the AWSServiceRoleForCodeWhispererPolicy managed policy have been updated to include the sso:DescribeApplication permission, which allows the retrieval of information about CodeWhisperer.	March 29, 2024
Updated JupyterLab setup instructions	The procedures for setting up CodeWhisperer with JuptyerLab have been updated and clarified.	March 6, 2024
Generic instructions for integration with other services	A new section provides the basic IAM policy necessary for using CodeWhisperer in the context of other services, in general.	March 6, 2024
Updated CodeWhisperer Professional administrator policy	Added permissions to the CodeWhisperer Professional administrator policy: sso:CreateManagedApplicationInstance and codewhisperer:CreateProfile. These permissions are needed to create a	March 5, 2024

	CodeWhisperer profile in a non-management account.	
Data sharing with Amazon EMR Studio	The data sharing page has information about sharing content and telemetry data with Amazon EMR Studio.	February 7, 2024
More dashboard permissions needed	Access to the customizations dashboard requires additional permissions. Also, when CodeWhisperer is not used for two weeks, the dashboard will look different.	January 26, 2024
Security scan support for Ruby and Go	The security scan feature supports Ruby and Go. Also, the security scan section has also been significantly updated for clarity.	January 12, 2024
Visual Studio support for C and C++	Visual Studio integration (in preview) works with C and C++, in addition to C#.	December 13, 2023
AWS CDK support	CodeWhisperer supports the AWS CDK with Typescript and Python.	December 13, 2023
Simplified Toolkit onboarding process	The description of the procedure for getting started with CodeWhisperer in VS Code and JetBrains IDEs has been significantly simplified.	November 28, 2023

Amazon Q authentication	In order to use some features of Amazon Q, you must authenticate with CodeWhisperer Professional.	November 28, 2023
Visual Studio support	CodeWhisperer works with Visual Studio.	November 26, 2023
Support for groups with customization	You can add groups of users to a customization.	November 26, 2023
Security scan support for more languages and frameworks	You can run a security scan, on code written in TypeScript, C#, Terraform, AWS CloudFormation, or the AWS CDK.	November 26, 2023
New permissions for listing customization versions	If you use customizations, you should add `codewhisperer:ListCustomizationVersions` to the customization policy attached to your CodeWhisperer administrator's role.	November 26, 2023
Dashboard updates	The dashboard displays the acceptance rate, and can be filtered by programming language.	November 26, 2023
Customization versioning	You can update your customizations by creating new versions.	November 26, 2023
Code generation support for more languages	Language support for JSON (AWS CloudFormation), YAML (AWS CloudFormation), and HCL (Terraform).	November 26, 2023

Assisted code remediation	After running a security scan, CodeWhisperer can help you fix your code.	November 26, 2023
Command line	You can use CodeWhisperer at the command line.	November 20, 2023
Amazon EMR integration	Integration with Amazon EMR.	November 17, 2023
Customization logs and console error messages	You can export log messages about your customiza tions, and a table provides information that will help you troubleshoot related console error messages.	November 13, 2023
JupyterLab version 4 support	Integration with JupyterLab version 4 is now supported.	November 7, 2023
Customizations	You can train CodeWhisperer on your own codebase.	October 17, 2023
More support for Go, SQL, PHP, Rust, and Kotlin	Go, SQL, PHP, Rust, and Kotlin are now included in the list of "most supported" languages.	October 13, 2023
Member accounts	You can now set up CodeWhisperer in more than one account within your organization.	September 12, 2023
Dashboard	A dashboard is available for adminstrators at the professio nal tier.	September 8, 2023

AWS PrivateLink integration	You can establish a private connection between your VPC and CodeWhisperer by creating an interface VPC endpoint.	July 26, 2023
AWS Glue integration	You can use CodeWhisperer with AWS Glue Studio Notebook.	July 26, 2023
Updated managed policy	Added GetFindings and GetManagedApplicationInstance.	June 28, 2023
Added permissions to policy for administrators	Added related new permissions iamadmin:ListRolesByPrincipal and pricing:GetProducts, needed for CodeWhisperer administration.	June 27, 2023
Non-management accounts can now manage CodeWhisperer	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	The following APIs can now be tracked in CloudTrail: DeleteProfile, GetCodeAnalysis, ListCodeAnalysisFindings, StartCodeAnalysis, CreateUploadUrl, GenerateCompletions.	June 6, 2023
Change to service-linked role	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
Added two Amazon CloudWatch metrics	Added Subscriptions and MonthlyActiveUniqueUsers as CloudWatch metrics.	May 30, 2023
Amazon SageMaker Studio and JupyterLab support	Added sections explaining setup for integration with SageMaker Studio and JupyterLab.	May 9, 2023
New service-linked role	Added sso:ListDirectoryAssociations as a service-linked role.	May 1, 2023
Monitoring chapter	Added information about monitoring CodeWhisperer with CloudWatch.	April 24, 2023
Security scans of multiple files	The security scan section has been updated to clarify that a scan can include more than one file.	April 20, 2023

Region-based restrictions	The CodeWhisperer administr ator section has been updated to clarify the specific situation in which actions must be taken in a particular region.	April 19, 2023
Useful APIs	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
Types of users	The Types of users chapter has been added to help clarify the different personas who use CodeWhisperer in different ways.	April 13, 2023
Setting up	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
Service-linked roles	Added information about service-linked roles.	April 13, 2023

Security content	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
Quotas	The Quotas chapter has been updated to clarify that CodeWhisperer does not maintain any quotas.	April 13, 2023
Language support	The Language support section has been updated to include languages that are newly supported as of CodeWhisperer general availability.	April 13, 2023
IDE screenshots	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
Features	The following sections have been folded into the Features chapter: User actions, Language support, Pausing suggestions, Security scans, and Code references.	April 13, 2023

Data opt-out screenshots	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
Billing	The Billing chapter has been added to provide information about how you may be charged for using CodeWhisperer.	April 13, 2023
Initial release	Initial release of the CodeWhisperer User Guide. Some materials included here were previously available in other guides.	February 21, 2023
New policy: AWSServiceRoleForAmazonCodeWhisperer	Added a new policy to allow CodeWhisperer to call CloudWatch and CodeGuru on your behalf.	February 17, 2023