



# Amazon CodeWhisperer

Amer Grgić

Senior Solutions Architect  
Amazon Web Services

# Sneak peek into an engineer's life



Engineer

We need to detect celebrities  
in images. Go build!



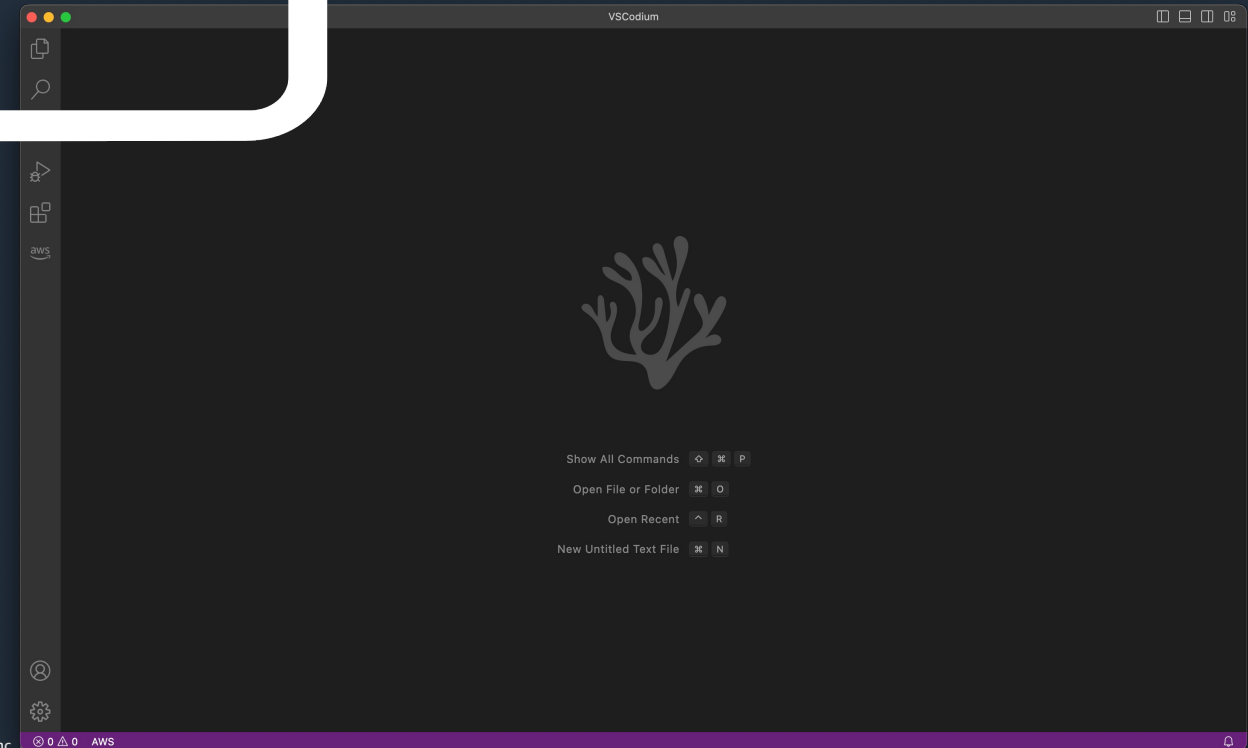
Product Manager

# Sneak peek into an engineer's life



Engineer

Sure, no problem!

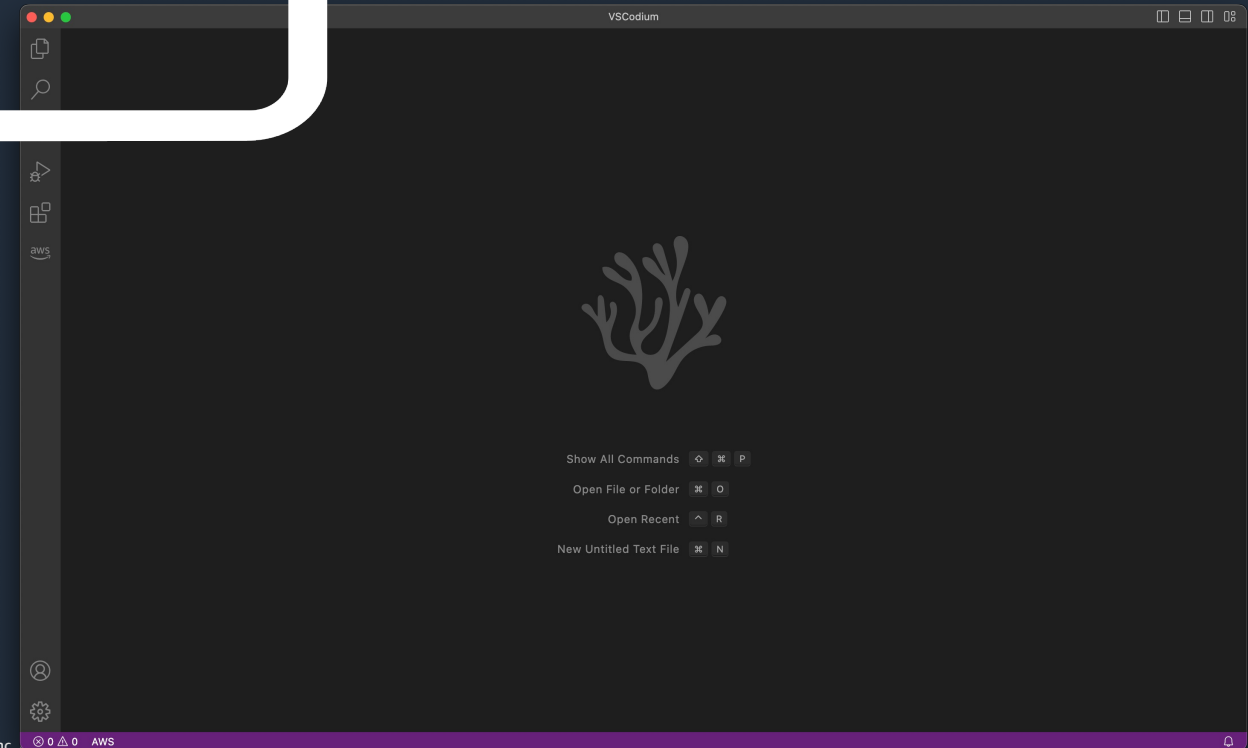


# Sneak peek into an engineer's life



Engineer

Hmm, but where do I start?



The screenshot shows the AWS documentation website for Amazon Rekognition. The main heading is "What is Amazon Rekognition?". Below it, there's a summary paragraph explaining that Amazon Rekognition makes it easy to add image and video analysis to applications by providing an API and service for identifying objects, people, text, scenes, and activities. It also mentions that it can detect inappropriate content and compare faces for various uses like verification, cataloging, and counting.

A sidebar on the left lists navigation options under "Amazon Rekognition Developer Guide":

- What Is Amazon Rekognition? (selected)
- How it works
- Getting started
- Working with images and videos
- Best practices for sensors, input devices, and videos
- Detecting labels
- Detecting and analyzing faces
- Searching faces in a collection
- People pathing
- Detecting personal protective equipment
- Recognizing celebrities
- Moderating content
- Detecting text
- Detecting video segments
- Tutorials
- Code examples
- API Reference
- Security
- Monitoring
- Guidelines and quotas
- Document history
- AWS glossary

On the right side, there's a section "On this page" with links to "Amazon Rekognition and HIPAA eligibility" and "Are you a first-time Amazon Rekognition user?".

Below the main heading, there are three bullet points highlighting key features:

- Searchable image and video libraries** – Amazon Rekognition makes images and stored videos searchable so you can discover objects and changes that appear within them.
- Face-based user verification** – Amazon Rekognition enables your applications to confirm user identities by comparing their live image with a reference image.
- Detection of Personal Protective Equipment** – Amazon Rekognition detects Personal Protective Equipment (PPE) such as face covers, head covers, and hand covers on persons in images. You can use PPE detection where safety is the highest priority. For example, industries such as construction, manufacturing, healthcare, food processing, logistics, and retail. With PPE detection, you can automatically detect if a person is wearing a specific type of PPE. You can use the detection results to send a notification or to identify places where safety warnings or training practices can be improved.

At the bottom of the visible content, another bullet point is partially visible:

- Sentiment and demographic analysis** – Amazon Rekognition interprets emotional expressions such as happy, sad, or surprise, and demographic information such as gender from facial images. Amazon Rekognition can analyze images, and send the emotion and demographic attributes to a prediction of an emotional state, and Rekogn...

aws

Boto3 1.26.96 documentation

Search

Feedback

Do you have a suggestion to improve this website or boto3? Give us feedback.

Quickstart

A Sample Tutorial

Code Examples

ResourcesSign inAsk question

Information as described

Content Moderation mode?

Content Stored video been

Results

Content Moderation result for

on Web Services, Inc. or its affiliates. All rights reserved.

Rekognition

Client

class Rekognition.Client

A low-level client representing Amazon Rekognition

This is the API Reference for Amazon Rekognition Image, Amazon Rekognition Custom Labels, Amazon Rekognition Stored Video, Amazon Rekognition Streaming Video. It provides descriptions of actions, data types, common parameters, and common errors.

Amazon Rekognition Image

- CompareFaces
- CreateCollection
- DeleteCollection
- DeleteFaces
- DescribeCollection
- DetectFaces
- DetectLabels
- DetectModerationLab
- DetectProtectiveEqui
- DetectText
- GetCelebrityInfo
- IndexFaces
- ListCollections
- ListFaces
- RecognizeCelebrities
- SearchFaces
- SearchFacesByImage

Amazon Rekognition

- CopyProjectVersion
- CreateDataset
- CreateProject
- CreateProjectVersion

ON THIS PAGE

Client

Rekognition.Client

Paginators

Waiters

ProductSolutionsOpen SourcePricing

Search

Sign InSign up

aws-samples / amazon-rekognition-code-samplesPublic

NotificationsFork28Star33

<> CodeIssuesPull requestsActionsSecurityInsights

mainamazon-rekognition-code-samples / rekognition-api / 3-celebrity-recognition.ipynbGo to file

Rao Added sample notebooks for Rekognition APIsLatest commit 6 weeks on Jun 8, 2021History

All 0 contributors

524 Lines (524 files)15.1 KBRawBlame

Celebrity Recognition using Amazon Rekognition

This notebook provides a walkthrough of celebrity recognition API in Amazon Rekognition. You can quickly identify well known people in your video and image libraries to catalog footage and photos for marketing, advertising, and media industry use cases.

Initialize stuff

In [ ]:

```
# Initialize Notebook
import boto3
from IPython.display import HTML, display, Image as Image
from PIL import Image, ImageDraw, ImageFont
import time
import os
```

In [ ]:

```
# Current AWS Region. Use this to choose corresponding S3 bucket with sample content
myRegion = boto3.session.Session().
awsRegion = myRegion.region_name
```

In [ ]:

```
# Init clients
rekognition = boto3.client('rekognition')
s3 = boto3.client('s3')
```

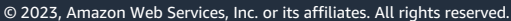
In [ ]:

```
# S3 bucket that contains sample images and videos
# We are providing sample images and videos in this bucket so
# you do not have to manually download/upload test images and videos.
bucketName = "aws-rek-lmsrslmday-" + awsRegion
```

In [ ]:

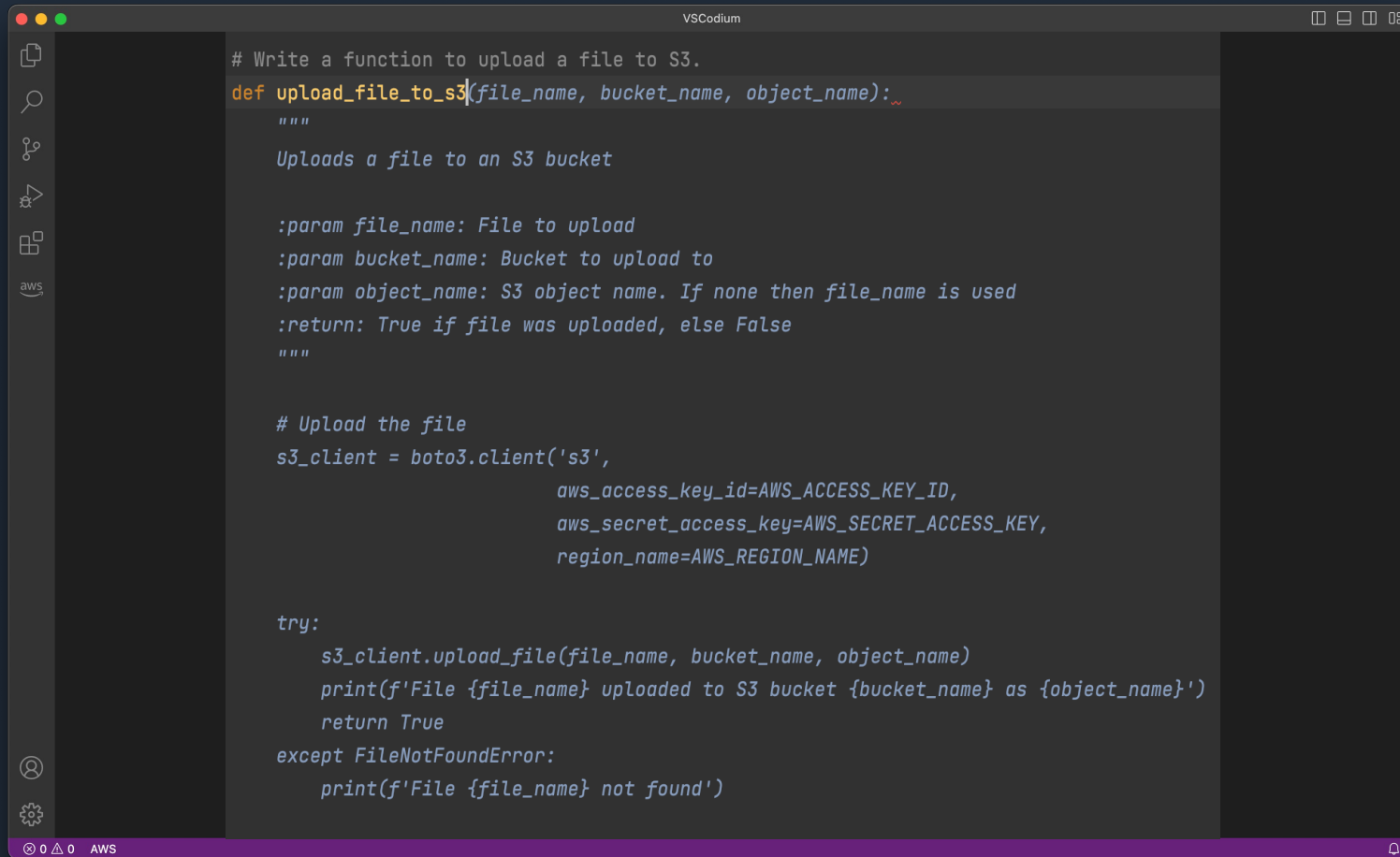
```
# Create temporary directory
# This directory is not needed to call Rekognition APIs.
# We will only use this directory to download images from S3 bucket and draw bounding boxes
# around recognized celebrities to show them here in the notebook.

!mkdir -p temp
tempFolder = "temp/"
```



# Wouldn't it be great if...

IDE++?



```
# Write a function to upload a file to S3.
def upload_file_to_s3(file_name, bucket_name, object_name):
    """
    Uploads a file to an S3 bucket

    :param file_name: File to upload
    :param bucket_name: Bucket to upload to
    :param object_name: S3 object name. If none then file_name is used
    :return: True if file was uploaded, else False
    """

    # Upload the file
    s3_client = boto3.client('s3',
                             aws_access_key_id=AWS_ACCESS_KEY_ID,
                             aws_secret_access_key=AWS_SECRET_ACCESS_KEY,
                             region_name=AWS_REGION_NAME)

    try:
        s3_client.upload_file(file_name, bucket_name, object_name)
        print(f'File {file_name} uploaded to S3 bucket {bucket_name} as {object_name}')
        return True
    except FileNotFoundError:
        print(f'File {file_name} not found')
```

# Quickly building reliable, secure code is challenging



Shortage of developers<sup>1</sup>



Time spent on undifferentiated code



Appropriate use of open source



Data privacy



Time spent learning technologies, APIs, and best practices

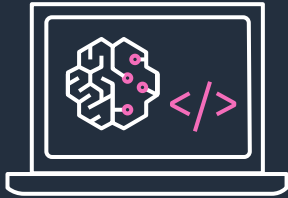


Write secure code

<sup>1</sup>Gartner: [2023 Planning Guide for Application Development](#)

# CodeWhisperer

BUILD APPLICATIONS FASTER AND MORE SECURELY WITH YOUR AI CODING COMPANION



Generate code suggestions in real time



Scan code for hard-to-find vulnerabilities



Flag code that resembles open-source training data or filter by default

Amazon ran a productivity challenge, and participants who used CodeWhisperer were **27% more likely to complete tasks successfully and did so an average of 57% faster** than those who did not use CodeWhisperer

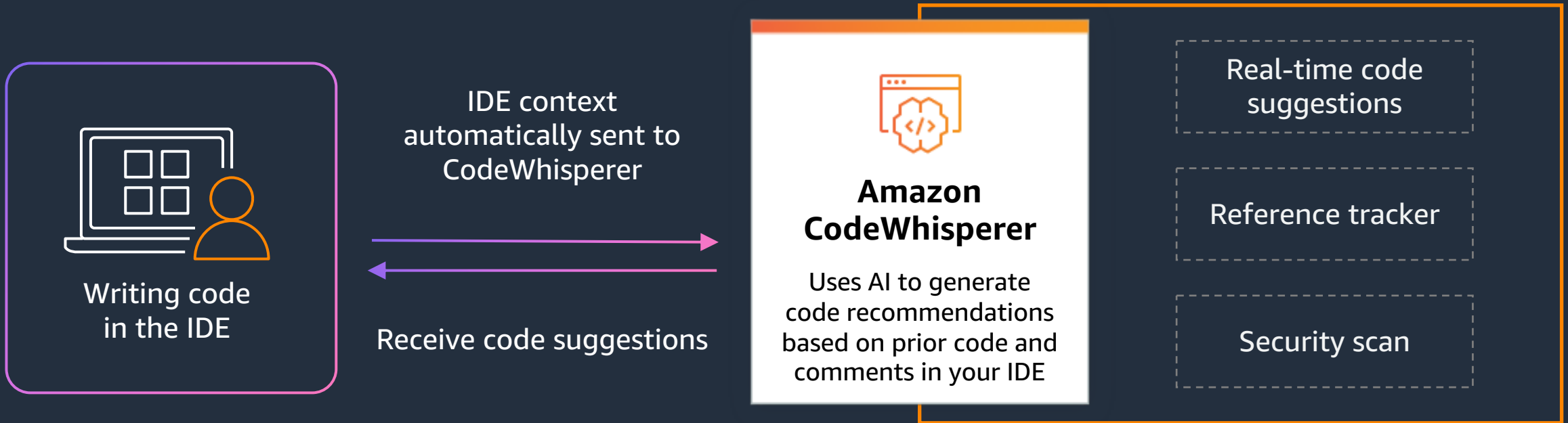


# Code generation

- Get multiple code suggestions in seconds based on natural language (English) description of coding task and surrounding code
- Provides high-quality suggestions for popular AWS services
- Generated code matches developer style and patterns



# How it works



➔ **Content processed by CodeWhisperer Professional is not stored or used for service improvement**

# CodeWhisperer features

## Supported ecosystem programming languages



Go, Rust, PHP,  
Ruby, Kotlin,  
C, C++,  
Shell scripting,  
SQL, and Scala



# CodeWhisperer features

## Supported ecosystem

---

### IDEs



CLion, GoLand,  
WebStorm,  
Rider, PhpStorm,  
RubyMine, and  
DataGrip



# Reference tracking

- Trained on billions of lines of code
- Flags code similar to open-source training data
- Tracks accepted suggestions so that you can provide appropriate attribution
- Enterprise controls to more easily deactivate/filter code suggestions similar to open-source training data

```
ion to iterate over an S3 bucket and send the files to a lambda  
const getFiles = async (bucketName, S3Client) => {
```

reference code under MIT License.

```
params = {  
  bucket: bucketName,
```

```
files = [];  
truncated = true;  
continuationToken;
```

```
(isTruncated) {  
  const response = await s3Client.listObjectsV2(params);  
  truncated = response.IsTruncated;
```

```
contin  
spons
```

```
return
```

## ✔ CodeWhisperer reference log

[3/24/2023, 3:17:34 PM] Accepted recommendation with c  
`response.Contents.forEach((file) => files.push(file.K`  
with reference under MIT from repository `function-temp`  
`/demos/reference-tracker.js (line at 14).`

# Security scanning

- Scan generated and developer-written code to detect security vulnerabilities
- Receive vulnerability remediation suggestions
- Scan for hard-to-find security vulnerabilities
- Supports VS Code and JetBrains IDEs for Python, Java, and JavaScript

🟢 Security scan completed. 3 issues found.



lambda\_function.py

user/projects/



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



The elevated privilege level required to perform operation immediately after the operation is performed. [Line 16]



Recreating AWS clients from scratch in each Lambda function expensive and can lead to availability risks. Clients should be reused in invocations. [Line 190]

# Enterprise controls

- Provision and manage team access through AWS IAM Identity Center and single sign-on
- Set organizational policies, such as automatic filtering of code suggestions similar to open-source code

## Set up CodeWhisperer

Select which users and groups in your organization have access to CodeWhisperer. Authorized users can activate CodeWhisperer through the AWS Toolkit in Visual Studio Code, AWS Cloud9 or JetBrains. CodeWhisperer will then provide suggestions, inside the IDE, for completing the user's code. Before users and groups appear in the lists below, you must add them to your organization in IAM Identity Center. [IAM Identity Center](#)

### Details

This setting applies to all users and groups.

### 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.

### Terms

By setting up CodeWhisperer, you agree to the terms.

[View terms](#)

### Users (3)

### Groups (1)

### Users (3)



Name

Email



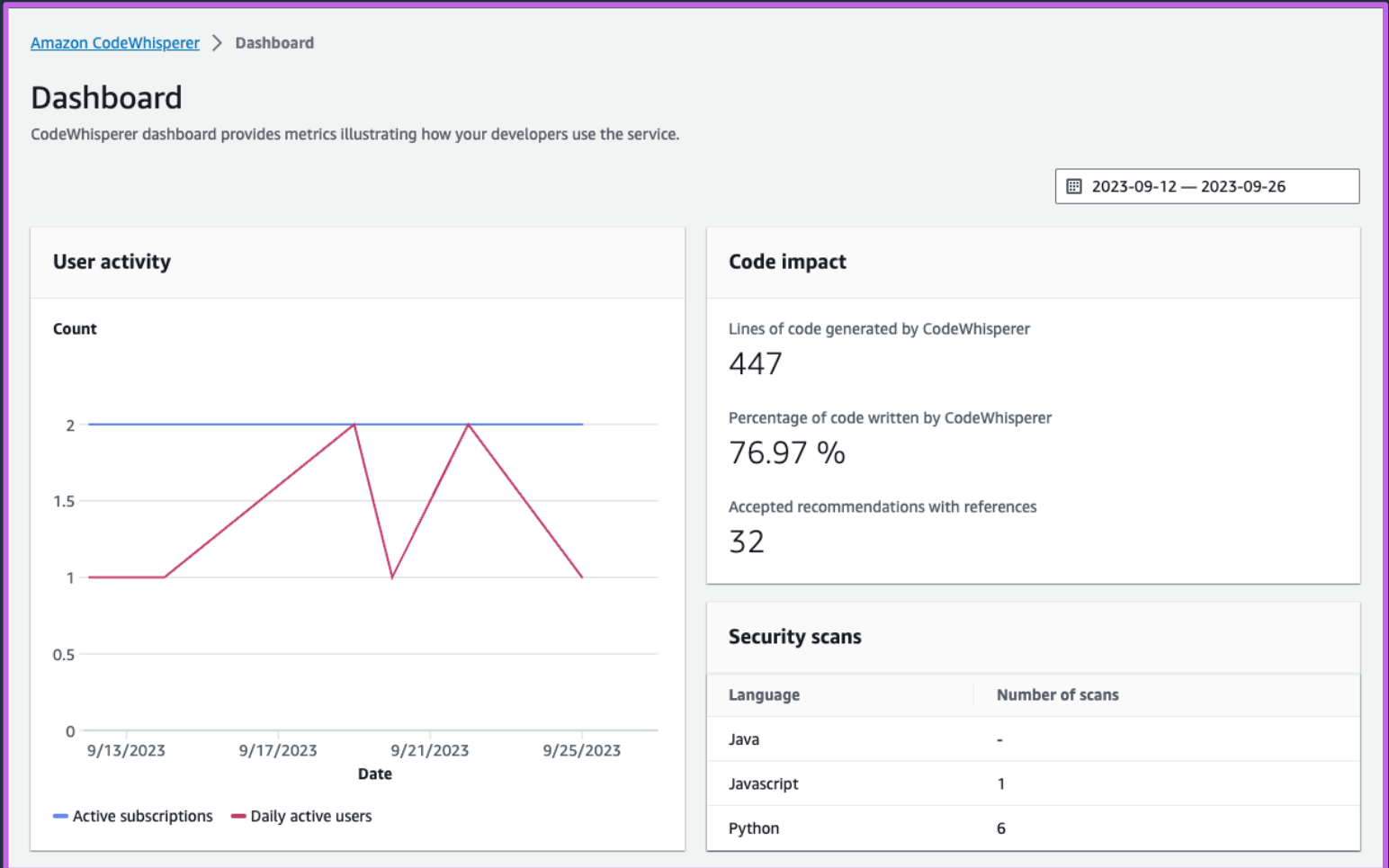
No users or groups will be given access to CodeWhisperer.

[Cancel](#)

[Set up CodeWhisperer](#)

# Productivity Dashboard

- Easily track usage activity trends
- Measure productivity impact using metrics such as lines of code and percentage of total code generated by CodeWhisperer.
- Available with CodeWhisperer Professional Tier





# Customize CodeWhisperer for your org

Preview

CODEWHISPERER CAN GENERATE CODE RECOMMENDATIONS  
BASED ON YOUR INTERNAL CODE BASES



Generate real-time recommendations that include your **internal libraries, APIs, packages, classes, and methods**

**Securely connect your internal code repositories** to create customizations isolated from other customers, and from the LLM powering CodeWhisperer

Fully **control which developers have access** to customizations

# Customize CodeWhisperer for your org

Preview

CODEWHISPERER CAN GENERATE CODE RECOMMENDATIONS  
BASED ON YOUR INTERNAL CODE BASES

```
1 // Process a list of unassigned food deliveries around the driver's current location
2 package anycompany.fooddelivery.fulfillment;
3 import anycompany.delivery.Delivery;
4 import anycompany.delivery.DeliveryService;
5 import anycompany.driver.Driver;
6 import anycompany.driver.DriverLocationService;
7
8 public class FoodDeliveryFulfillment {
9     private DeliveryService deliveryService;
10    private DriverLocationService driverLocationService;
11
12    //Process all the unassigned deliveries
13    public void processUnassignedDeliveries() {
14        List<Delivery> unassignedDeliveries = deliveryService.getUnassignedDeliveries();
15        //Iterate over all the unassigned deliveries and assign them to a driver
16        for (Delivery delivery : unassignedDeliveries) {
17            //Get the nearest drivers for delivery location
18            List<Driver> drivers = driverLocationService.getDriver(delivery.getLocation());
19            for (Driver driver : drivers) {
20                //Assign delivery to driver and send notification
21                boolean isAssigned = deliveryService.assignDeliveryToDriver(delivery, driver);
22                if (isAssigned) {
23                    deliveryService.notifyDelivery(delivery);
24                    driverLocationService.notifyDriver(driver);
25                    break;
26                }
27            }
28        }
29    }
30    ...
31 }
```

Natural language comment prompt

Generated code block that leverages  
internal libraries and APIs

# Demo



# Benefits



Quickly accomplish  
basic coding tasks



More easily use unfamiliar  
APIs and frameworks



Get high-quality  
suggestions for AWS  
services



Use AI responsibly



Improve  
application security



Use your  
favorite tools

# Getting started



**Get started**



**Dive deep with  
a workshop**



**Learn how to build an  
event-driven  
serverless app**



# Thank you!

Amer Grgić

amergr@amazon.nl

