

The background features a dark navy blue field with abstract, overlapping shapes in vibrant magenta and deep red. Two thin, light blue lines intersect diagonally across the upper right portion of the image. The text is positioned on the left side.

AWS re:Invent

DECEMBER 2 – 6, 2024 | LAS VEGAS, NV

D O P 3 1 3

Get tailored code insights with Amazon Q Developer and private repos

David Ernst

(he/him)

Sr. Specialist SA – DevOps

AWS

Kyle Thomson

(he/him)

Principal Software Dev Eng

AWS



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Generative AI and AWS

Amazon Q overview

Introduction to Amazon Q Developer

Amazon Q Developer customization

Demo

Getting started

< one hour coding time

The median developer spends less than one hour a day writing code¹



About the audience

- Who's heard of Amazon Q before?
- Who's heard of Amazon Q customizations?
- What roles do we have in the room?
 - e.g., Engineer, PM, IT admin
- What languages do folks use?

Amazon Q Developer



Reimagines the experience across the entire software development lifecycle (SDLC)

Helps developers and IT professionals build and manage secure, scalable, and highly available applications

Helps you write, debug, test, optimize, and upgrade your code faster

Converses with you to explore new AWS capabilities, learn unfamiliar technologies, and architect solutions

Amazon Q is built with security and privacy in mind from the start, making it easier for organizations to use generative AI safely

Where are developers spending time in the SDLC?

Plan

- Find accurate and specific technical guidance from relevant documentation, code examples, resources, and so on
- Design changes based on business and technical requirements

Create

- Generate code
- Manage infrastructure

Test and secure

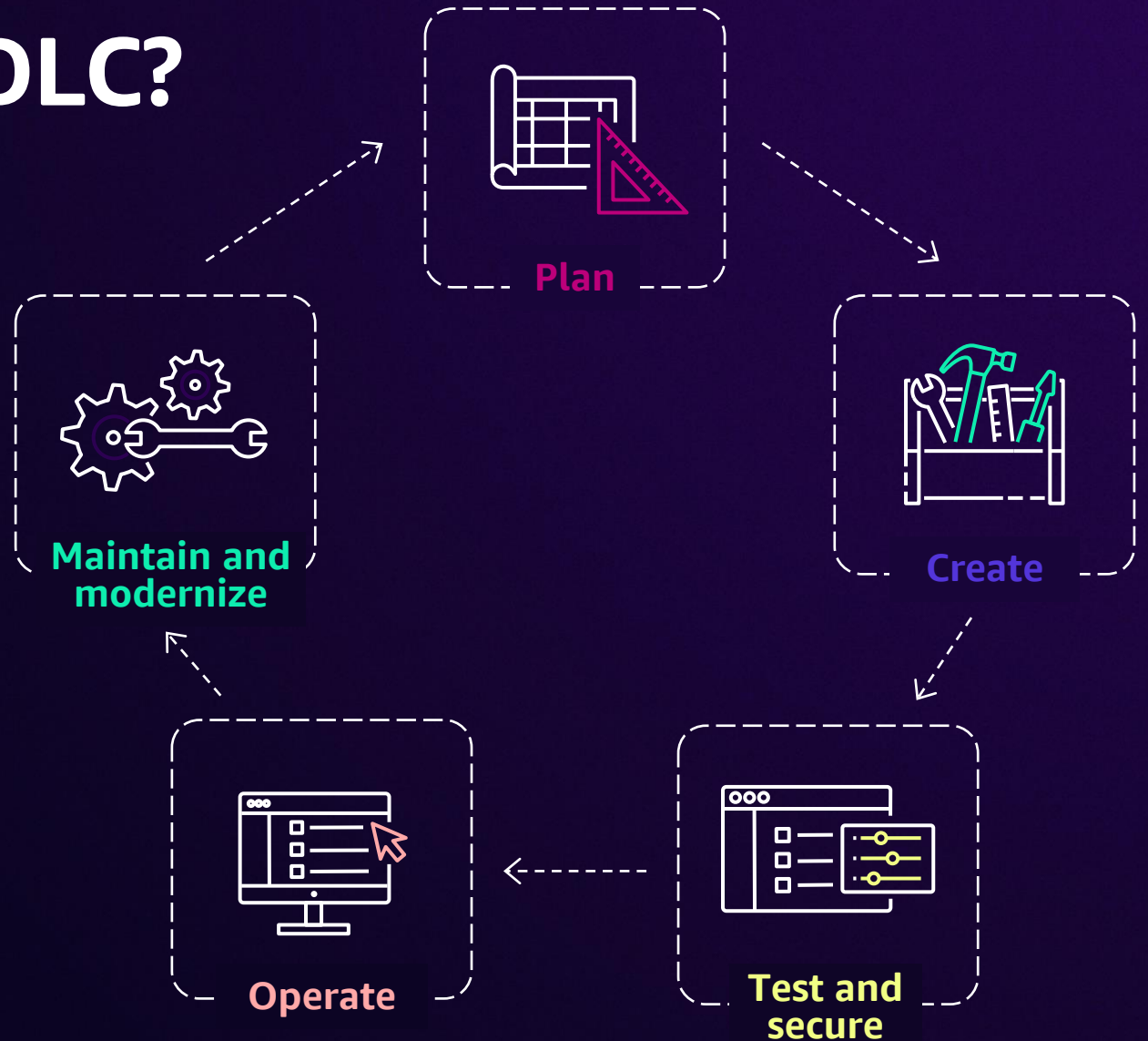
- Apply proper test cases for code changes
- Scan for security vulnerabilities in code

Operate

- Identify and mitigate code issues
- Monitor performance and efficiency

Maintain and modernize

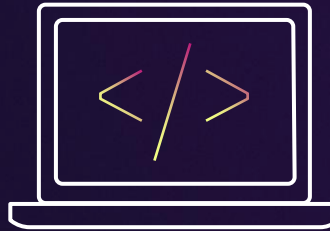
- Modernize and update old code languages and dependencies



Amazon Q is available where you do your work



AWS consoles



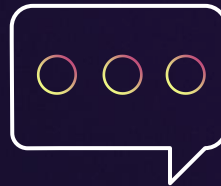
IDEs



AWS documentation



**AWS Console
Mobile Application**

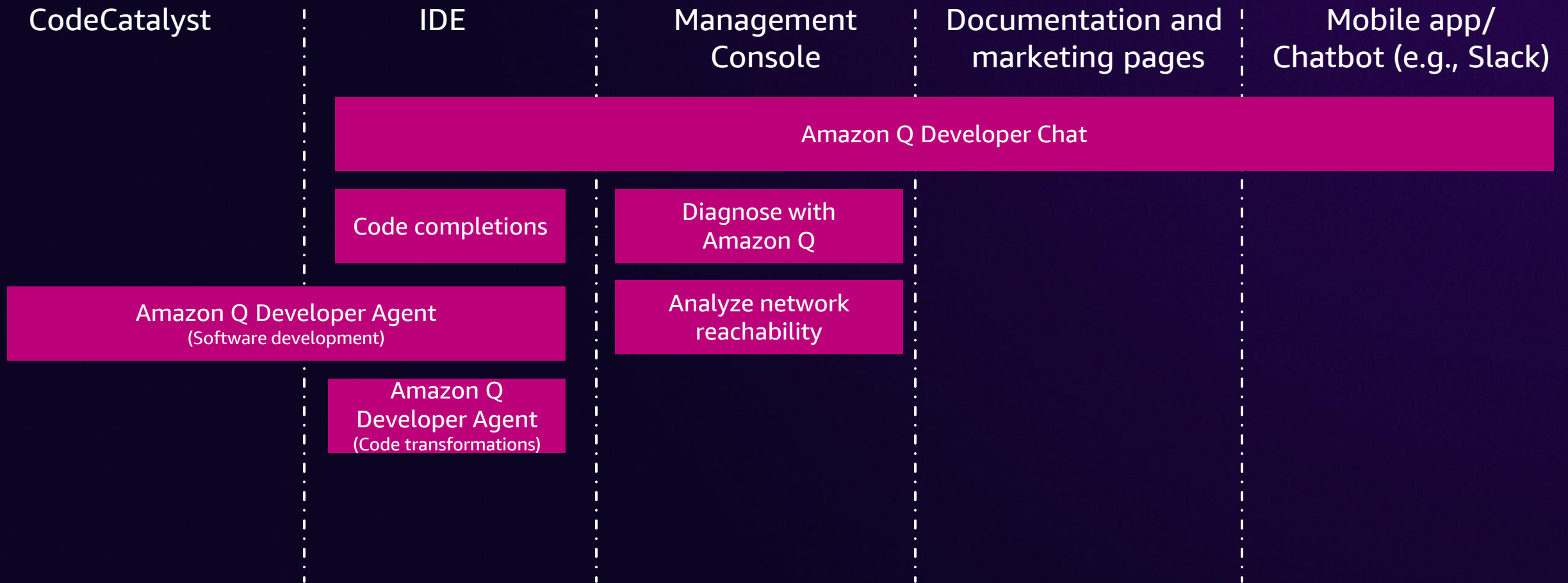


**Slack and Teams
(through AWS Chatbot)**

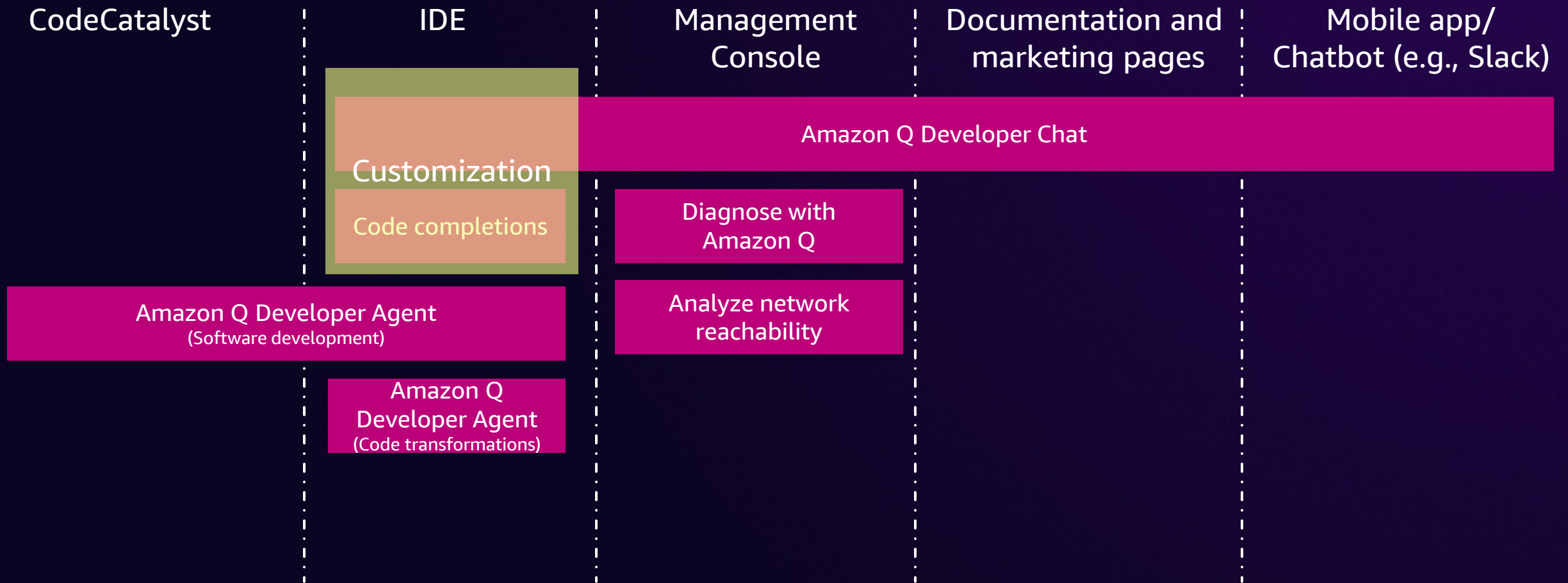


Command Line

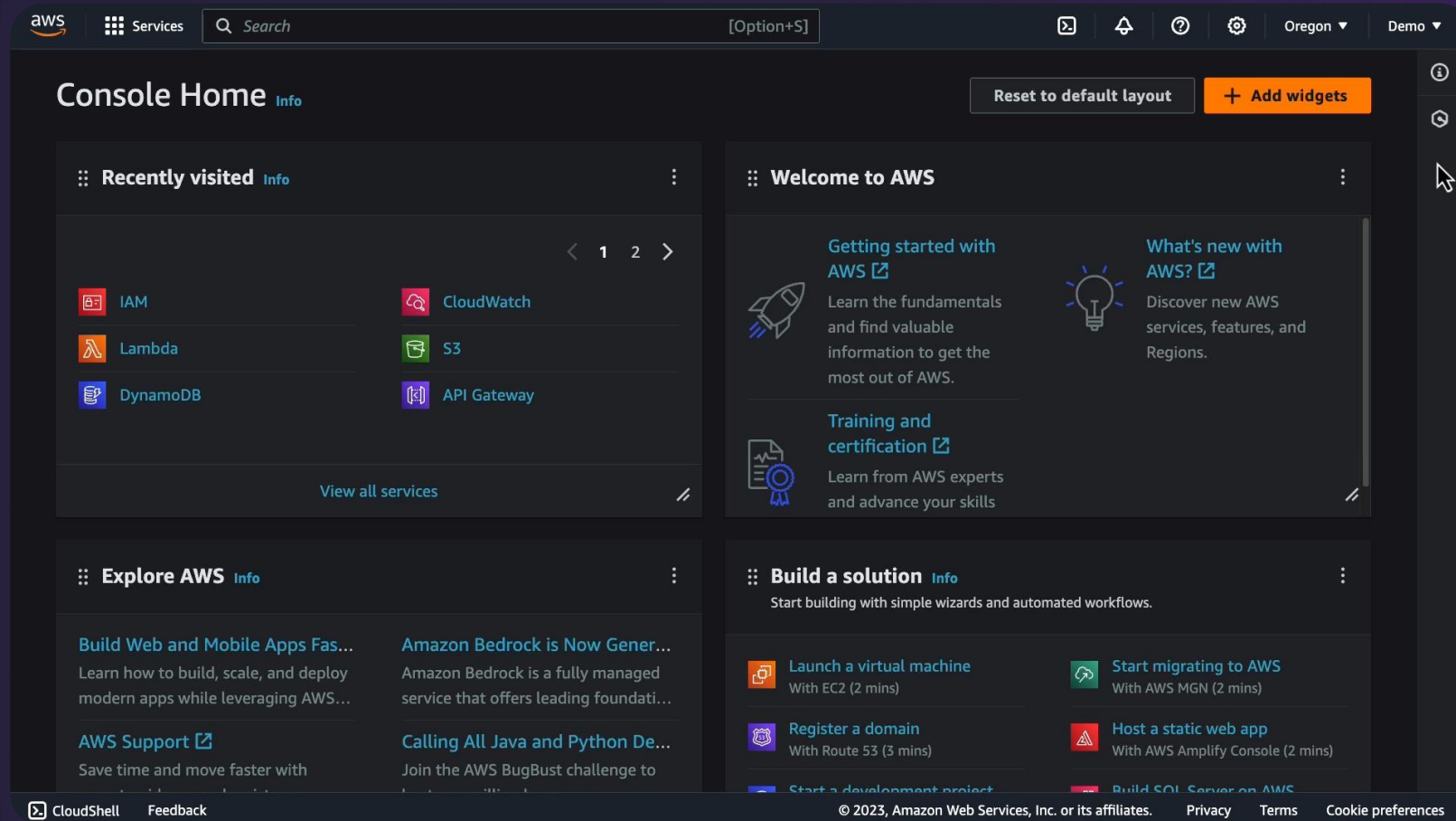
Amazon Q Developer landscape



Amazon Q Developer landscape

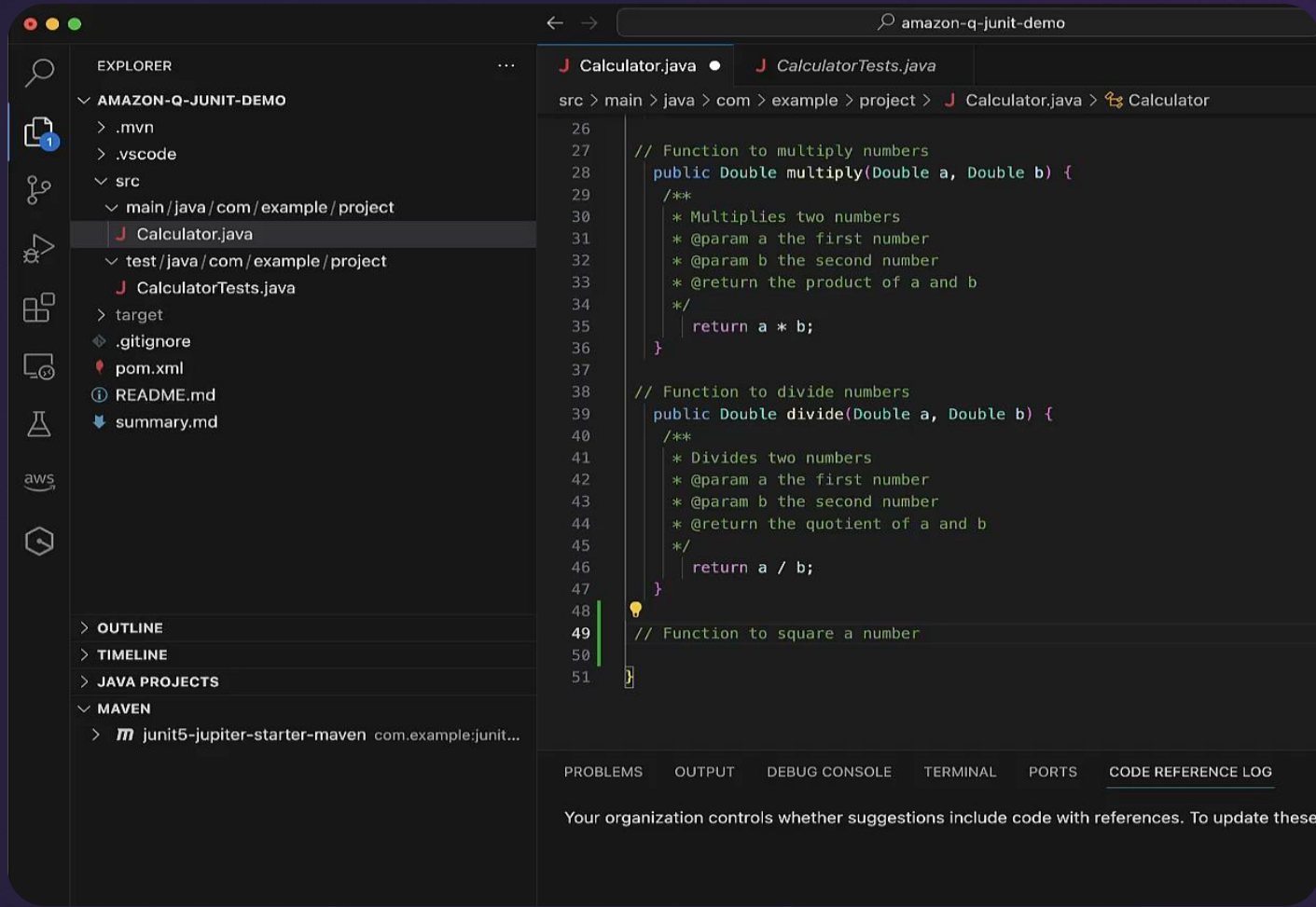


Explore and plan



Ask questions and get guidance from 17 years of AWS knowledge, best practices, and well-architected advice

Create in the IDE



Generate code

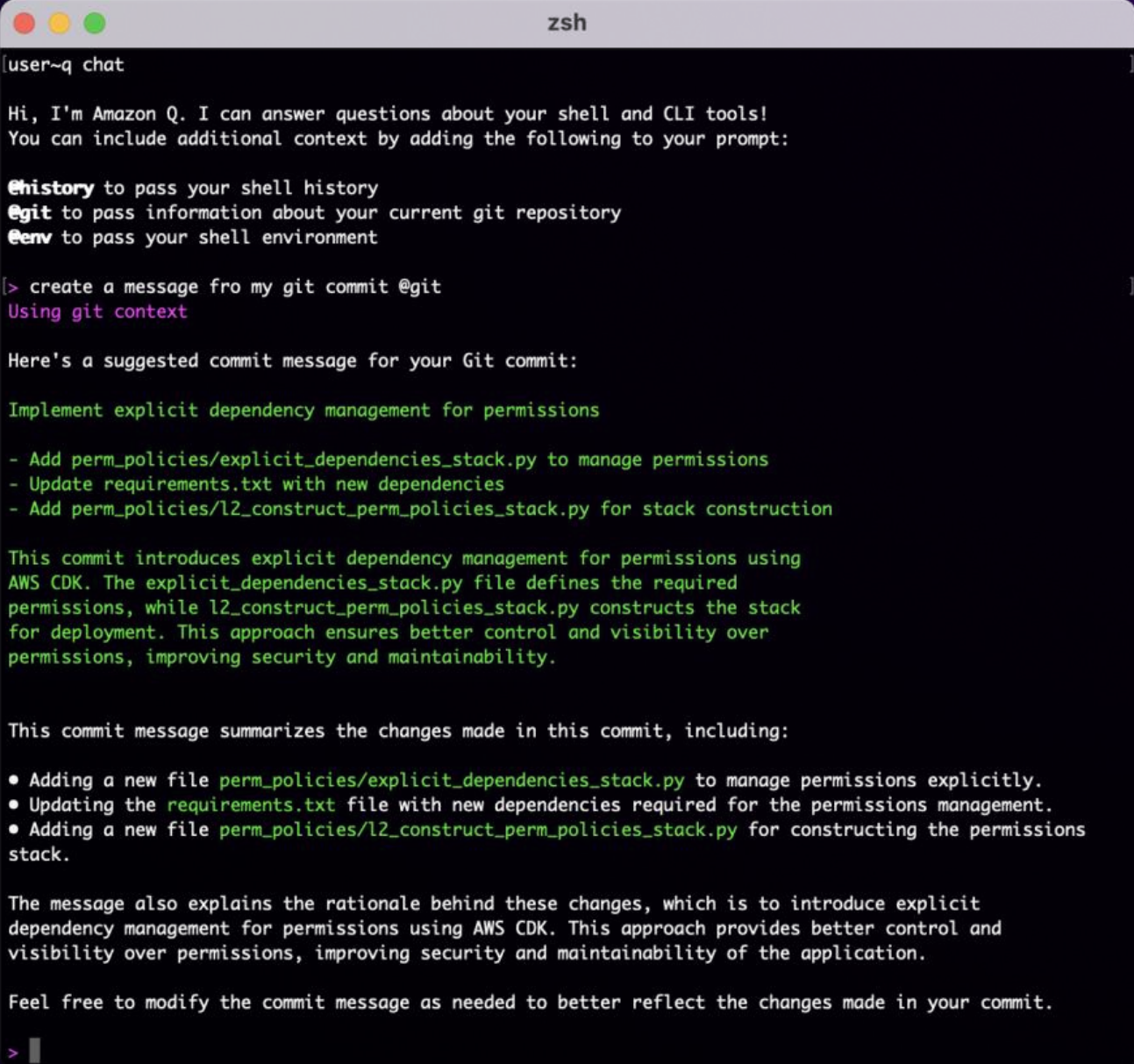
Explain code

Help understand
your code base

Customizable
to your code

Create in the command line

- Contextual CLI completions and AI natural language to bash translation
 - IDE-style completions for 500+ CLIs
 - Natural language to bash translation (Enter 'q!')



```
zsh
[user~q chat]
Hi, I'm Amazon Q. I can answer questions about your shell and CLI tools!
You can include additional context by adding the following to your prompt:

@history to pass your shell history
@git to pass information about your current git repository
@env to pass your shell environment

[> create a message fro my git commit @git]
Using git context

Here's a suggested commit message for your Git commit:

Implement explicit dependency management for permissions

- Add perm_policies/explicit_dependencies_stack.py to manage permissions
- Update requirements.txt with new dependencies
- Add perm_policies/l2_construct_perm_policies_stack.py for stack construction

This commit introduces explicit dependency management for permissions using
AWS CDK. The explicit_dependencies_stack.py file defines the required
permissions, while l2_construct_perm_policies_stack.py constructs the stack
for deployment. This approach ensures better control and visibility over
permissions, improving security and maintainability.

This commit message summarizes the changes made in this commit, including:

• Adding a new file perm_policies/explicit_dependencies_stack.py to manage permissions explicitly.
• Updating the requirements.txt file with new dependencies required for the permissions management.
• Adding a new file perm_policies/l2_construct_perm_policies_stack.py for constructing the permissions
stack.

The message also explains the rationale behind these changes, which is to introduce explicit
dependency management for permissions using AWS CDK. This approach provides better control and
visibility over permissions, improving security and maintainability of the application.

Feel free to modify the commit message as needed to better reflect the changes made in your commit.

> 
```

How often are you writing code against internal libraries?

Create: Customize code recommendations



```
1 // Process a list of unassigned food deliveries around the driver's current location
2
3 package anycompany.fooddelivery.fulfillment;
4 import anycompany.delivery.Delivery;
5 import anycompany.delivery.DeliveryService;
6 import anycompany.driver.Driver;
7 import anycompany.driver.DriverLocationService;
8
9 public class FoodDeliveryFulfillment {
10     private DeliveryService deliveryService;
11     private DriverLocationService driverLocationService;
12
13     //Process all the unassigned deliveries
14     public void processUnassignedDeliveries() {
15         List<Delivery> unassignedDeliveries = deliveryService.getUnassignedDeliveries();
16         //Iterate over all the unassigned deliveries and assign them to a driver
17         for (Delivery delivery : unassignedDeliveries) {
18             //Get the nearest drivers for delivery location
19             List<Driver> drivers = driverLocationService.getDriver(delivery.getLocation());
20             for (Driver driver : drivers) {
21                 //Assign delivery to driver and send notification
22                 boolean isAssigned = deliveryService.assignDeliveryToDriver(delivery, driver);
23                 if (isAssigned) {
24                     deliveryService.notifyDelivery(delivery);
25                     driverLocationService.notifyDriver(driver);
26                     break;
27                 }
28             }
29         }
30     }
31
32     ...
33
34 }
```

Natural language prompt

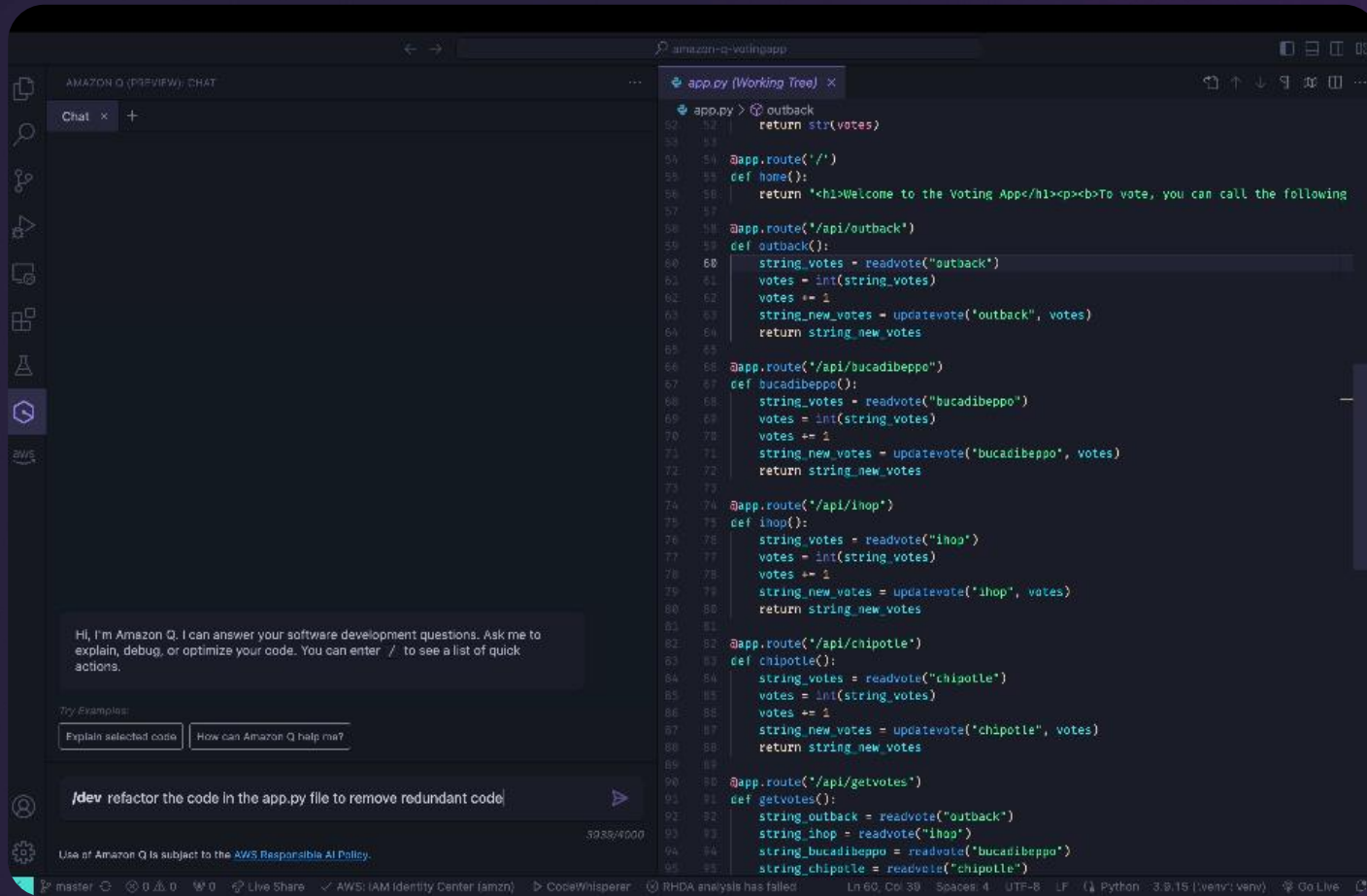
Generated in-line code based on your company's internal code bases

Generate code based on your private repositories

Use Amazon Q Developer chat in the IDE to ask questions about your internal code base

Help understand your internal code base

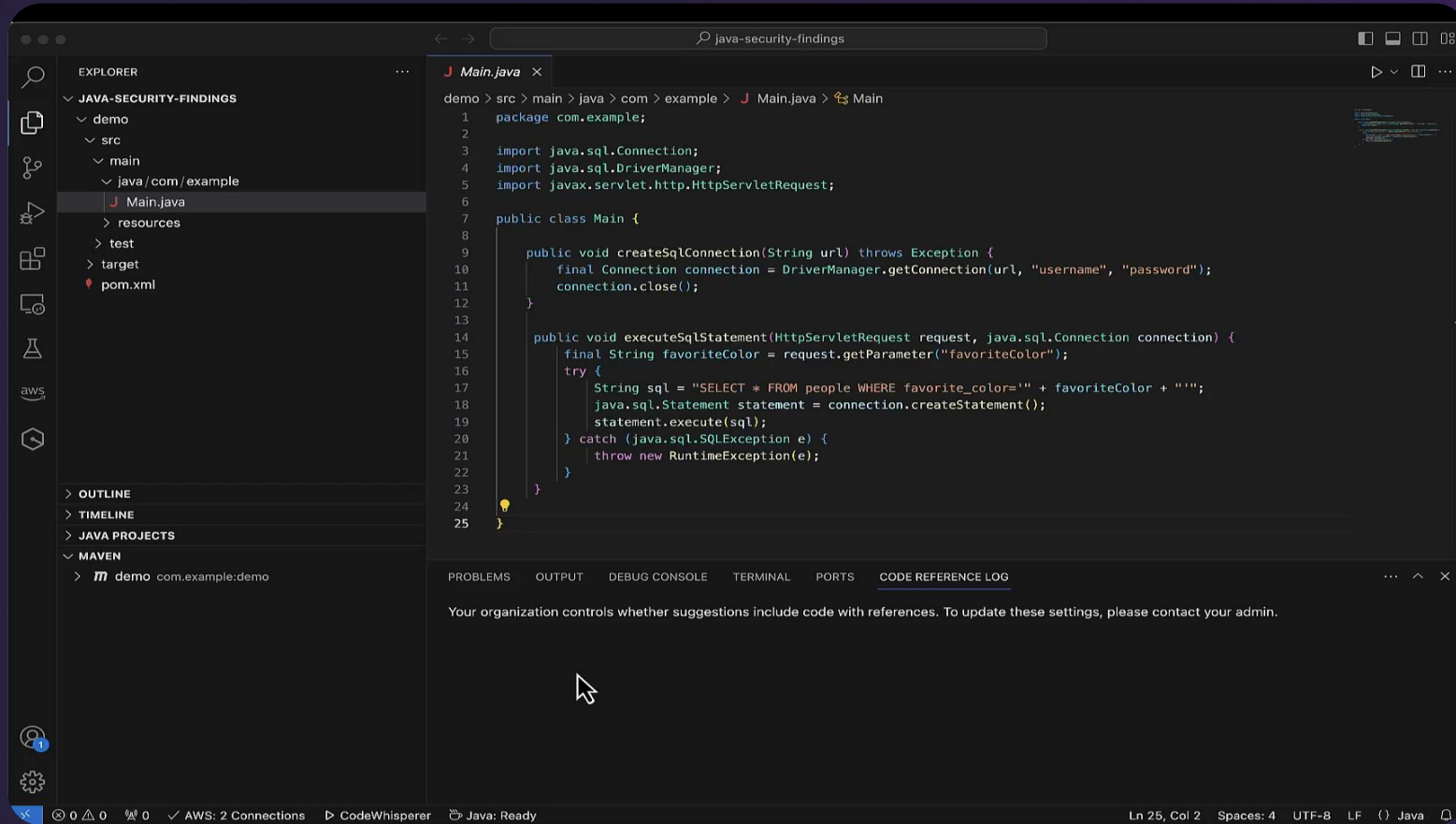
Create: Advanced capability – Amazon Q Developer agent for software development



Write and implement entire application features in minutes, right from a simple natural language prompt in your IDE

Interactive step-by-step instructions, code, tests, and best practices

Test and secure

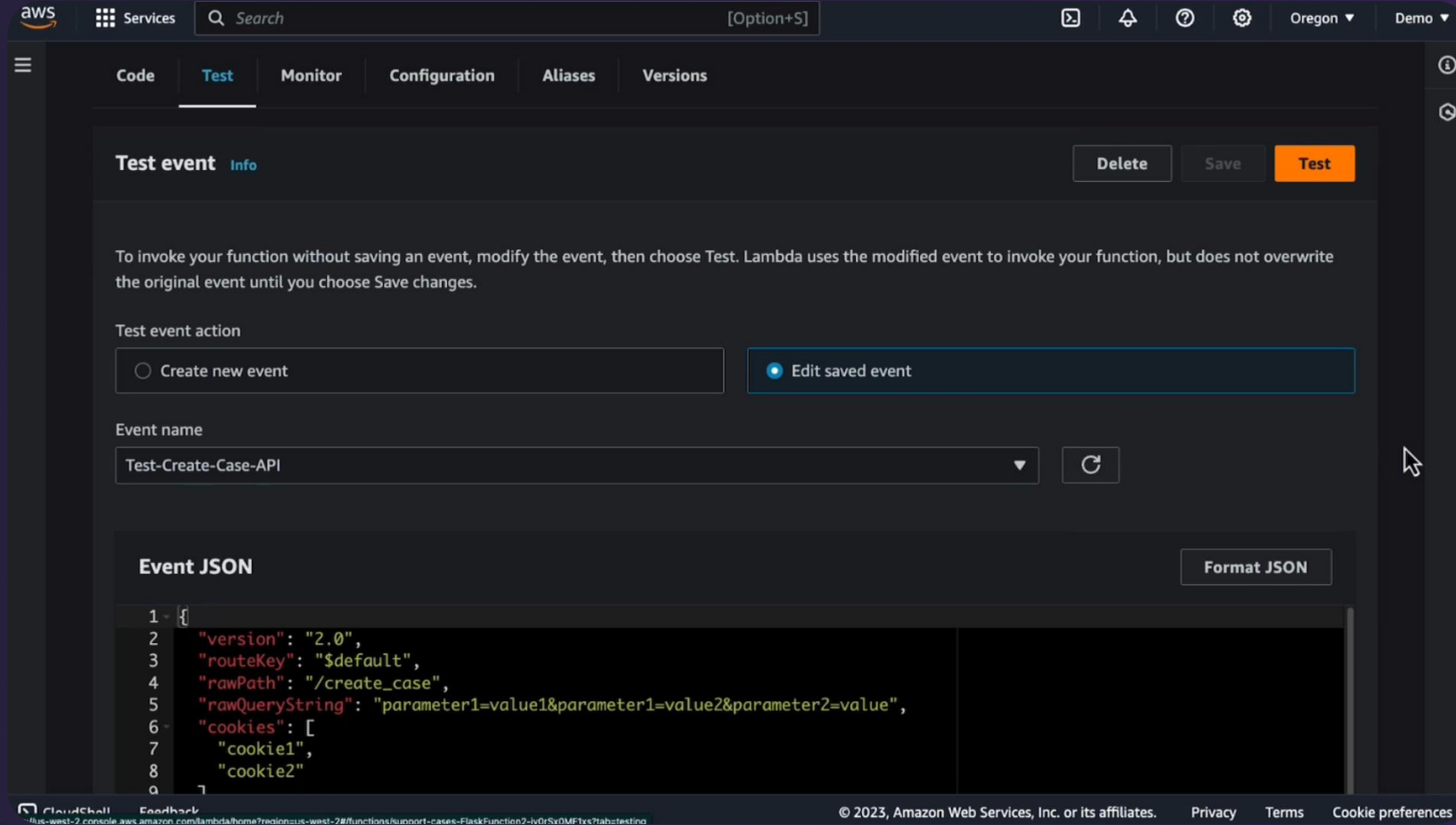


Generate unit tests

Scan entire project
for security
vulnerabilities

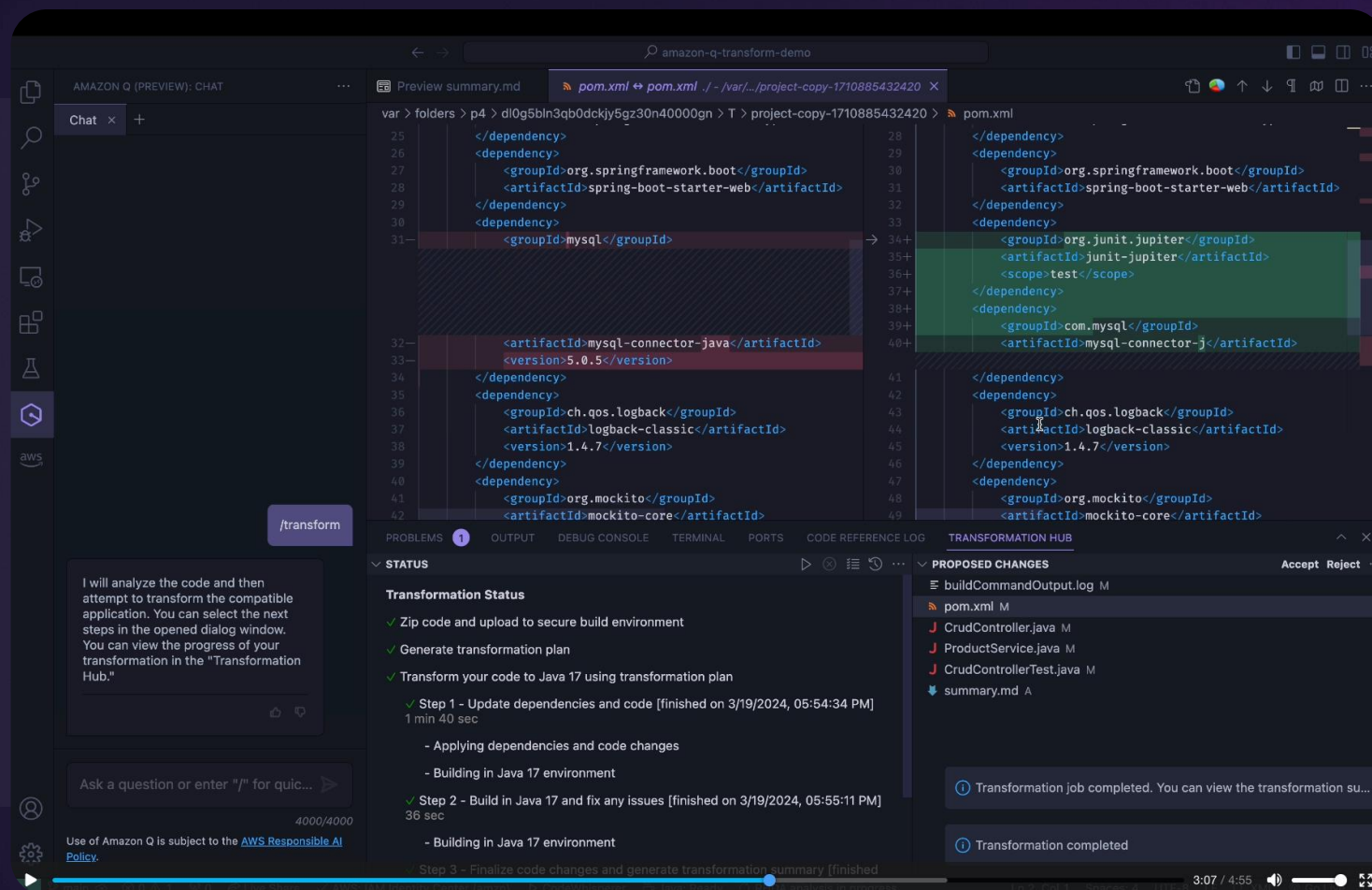
Generate
remediations to
improve security
and code quality

Operate



Diagnose and
troubleshoot
errors in
the console

Analyze network
connectivity and
VPC reachability

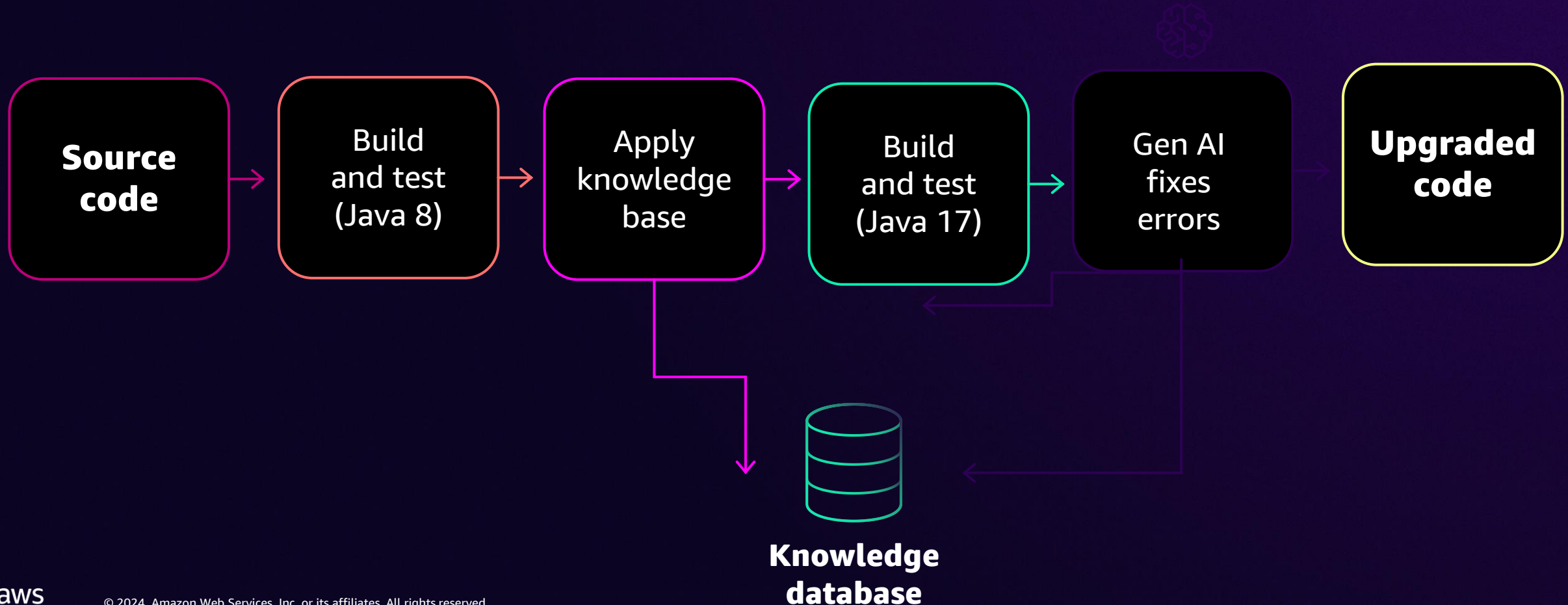


Amazon Q Developer agent for code transformation

Automates the complete process of upgrading and transforming code

Maintain and modernize – Amazon Q Developer agent for code transformation

MODERNIZE LANGUAGE VERSIONS IN A FRACTION OF THE TIME



Amazon Q Customization – personas



Admin or developer

Amazon Q Customization demo – persona – administrators



- Creating customization – AWS Management Console
 - S3
 - Code star connections options
 - Adding users/groups for access
- Retraining and creating new versions – AWS Management Console
 - Deactivating
 - Creating new version of customization
- Persona – Central IT or CCOE

Requirements

-  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 in your data source must be at least 2 MB. Valid file extensions are: .java, .py, .jsx, .tsx, .js, .ts

 Note

Amazon Q Customizations supports the following languages and file extensions:

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

Best practice – Increase volume of examples

While the minimum is 2MB, you will get better results with much larger codebases

Avoid duplication of same code (copy/paste) in training set

Avoid non-unique code (such as `node_modules`, etc.)

Review customization settings

- **Best practice:** enable the chat customization feature
- **Best practice:** configure an encryption key, which is also used for customizations

Advanced settings

These settings apply to all users and groups.

Amazon Q Developer suggestions

☒ Include suggestions with code references

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

☒ Enable chat customization feature

Customization in chat (Preview) configures the selected customization in the IDE (which, by default, is used for in-line code completion) to also be used for chat functionality. This allows developers to use natural language to generate code based on internal libraries. Developers may also ask questions about the codebase, such as how certain functions or libraries are used, or what specific functions, methods, or APIs do.

▼ Encryption key - optional [Info](#)

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)

Connect to your repo

Q Developer supports Bitbucket, GitHub, and GitLab

Q Developer will ingest the default branch in every repository

Best practice: Q Developer also supports Amazon S3. *This is the best option when you need to be selective*

Create a connection [Info](#)

Select a provider

☒ Bitbucket

☐ GitHub

☐ GitHub Enterprise Server

☐ GitLab

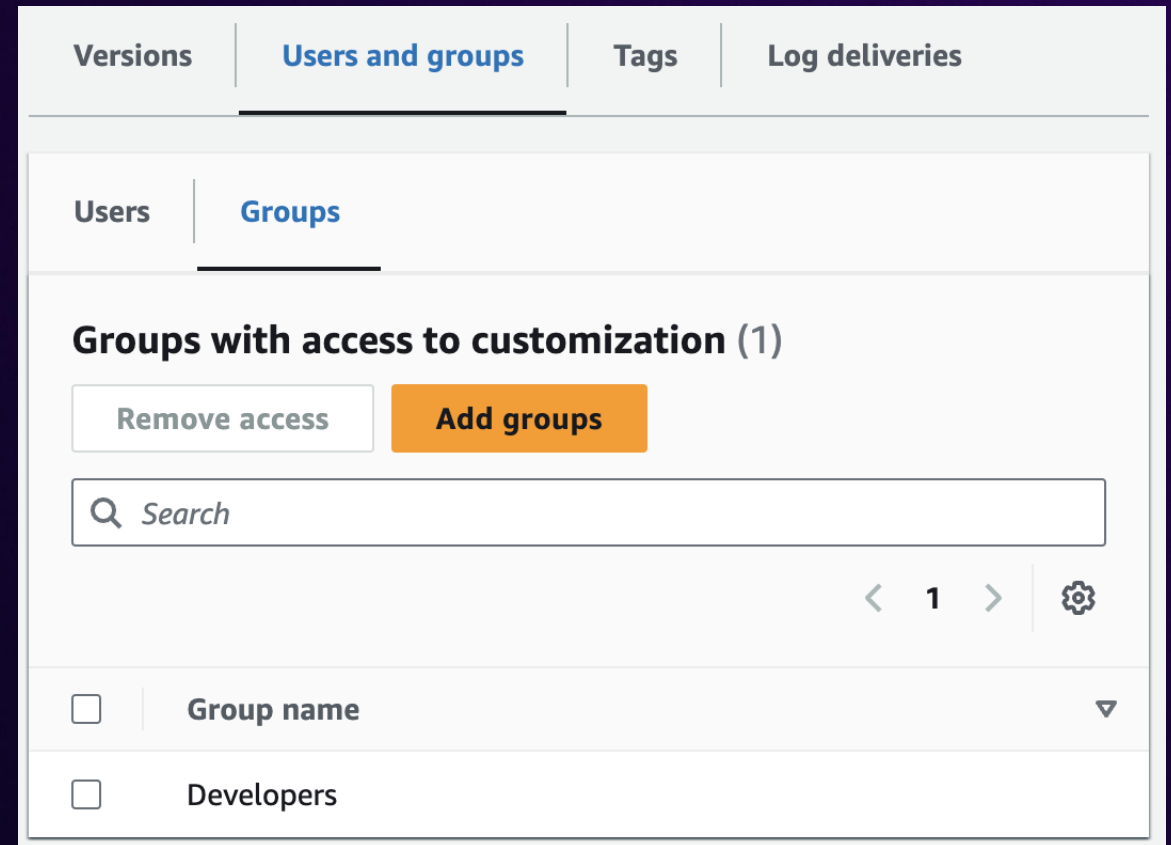
☐ GitLab self-managed

Access control

Add users and groups to control access to various customizations

You can have 8 customizations with 2 active customizations

Best Practice: create a customization for each team or business unit



Amazon Q Developer dashboard

Best practice: use the Amazon Q Developer dashboard to monitor usage of your customization

[Amazon Q Developer](#) > Dashboard

Dashboard [Info](#)

Amazon Q Developer dashboard provides metrics illustrating how your developers use the service.

Active customization: **q-customization-github** ▲ Programming language: **All** ▼ [Clear filters](#) Date range: [Last year](#)

- All
Includes the default model and your active customizations
- Default
General model
- q-customization-github** ✓
q-customization-github
- customization-demo

Code impact

Lines of code generated by Amazon Q Developer
No data available yet

Acceptance rate
No data available yet

Percentage of code written by Amazon Q Developer
No data available yet

Accepted recommendations with references
No data available yet

Amazon Q Customization demo – persona – end user



- Customization in IDE
 - Developer experience
 - In-line examples
 - Chat examples
- Persona – developers

Developers choose customization in IDE

Amazon Q

II Pause Auto-Suggestions Currently RUNNING

Inline Suggestions

</> Open Code Reference Log

Try inline suggestion examples

Resume Auto-Scans PAUSED

Security Scans

Run Project Scan

Select Customization Using Amazon Q foundation (Default)

Other Features

Open Chat Panel

Send Feedback

Connect with us on Github

View Documentation

Select a Customization

You have access to the following customizations

Amazon Q foundation (Default)

Receive suggestions from Amazon Q base model

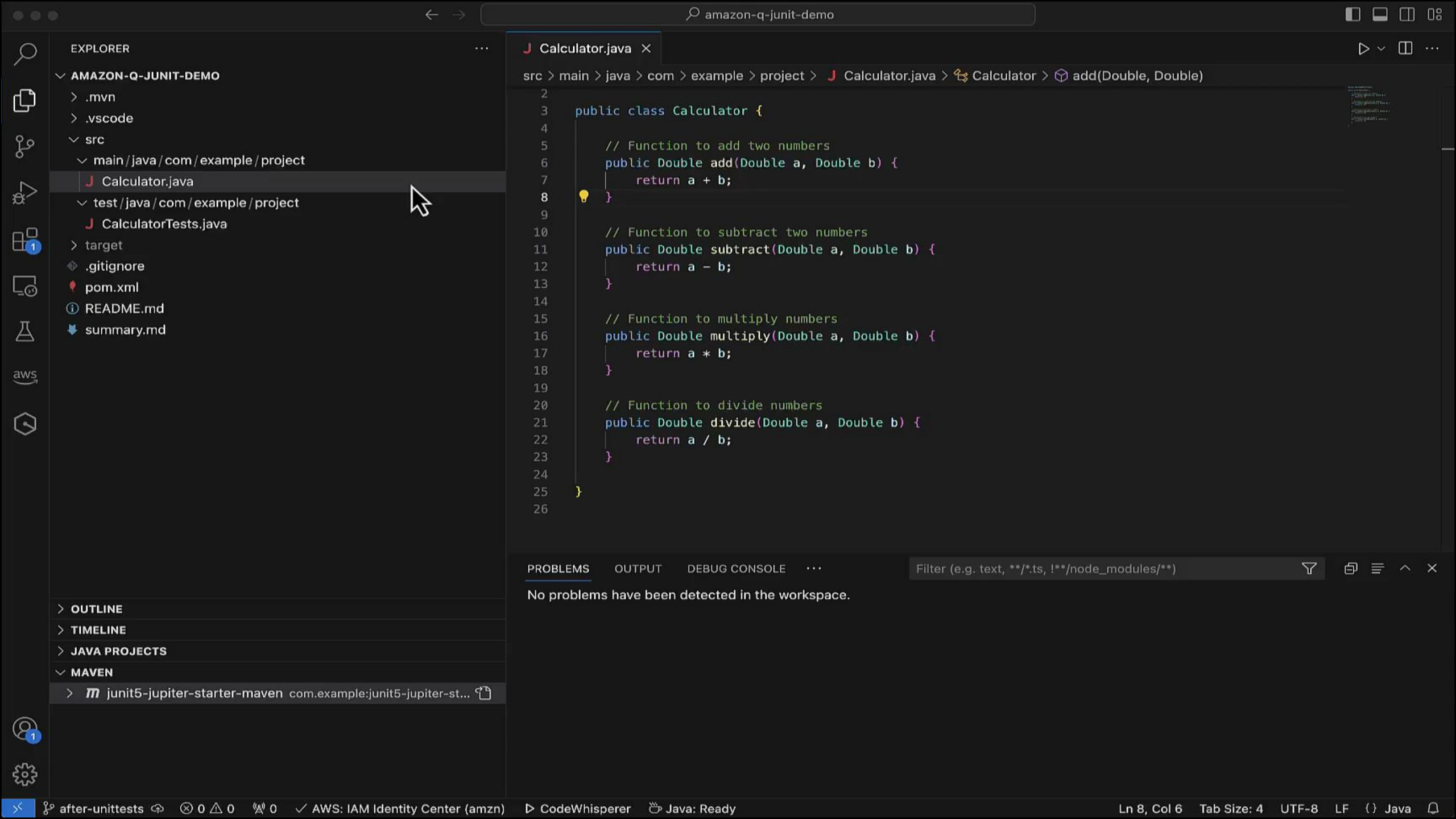
customization-demo Connected

No description provided

q-customization-github New

q-customization-github





Summary of core use cases



Retire tech debt at scale

Modernize projects and upgrade language version and dependencies with Amazon Q Developer Agent for code transformation



Drive developer productivity

Get in-line coding recommendations, summarize and document code, scan for security issues, diagnose and troubleshoot bugs, and write unit tests



Improve security and code quality

Ship more secure code with less effort with code scanning and automated code remediation

“We started using customization with Amazon Q Developer earlier this year, and some of our teams have already seen a **30% increase** in acceptance rate compared to the base version of Amazon Q Developer. The tailored recommendations based on our internal libraries are saving hours each week for our developers through code generation and semantic code search. Some of our developers are now accepting **over 50%** of all recommendations from Amazon Q Developer. Using customization with Amazon Q Developer is like pair-programming with a peer with deep knowledge of our codebase who helps guide our developers on our best practices.”

Girish Bajaj, Vice President, Amazon Prime Video Core Tech

Amazon Prime Video

Get started with Amazon Q Developer: Workshop



**Try Amazon Q
Developer today**



Thank you!

David Ernst

dmernst@amazon.com

[linkedin.com/in/davidmernst](https://www.linkedin.com/in/davidmernst)

Kyle Thomson

kylthoms@amazon.com

[linkedin.com/in/kiiadi/](https://www.linkedin.com/in/kiiadi/)

github.com/kiiadi



Please complete the session survey in the mobile app