



slalom

ADC-C3

Amazon CodeWhisperer

Build applications faster and more securely with
your AI coding companion

Damian Niezgoda (he/him)

Solutions Architect
AWS

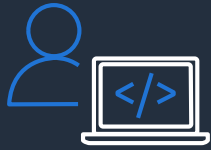
Eimear Loughnane (she/her)

Director of Software Engineering
Slalom

Agenda

- Customer challenges
- Service overview, benefits, and features
- Demos
- Getting started

Quickly building reliable, secure code is challenging



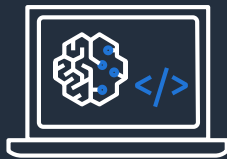
Shortage of developers¹



Time spent on undifferentiated code



Appropriate use of open source



Data privacy



Time spent learning technologies, APIs, and best practices

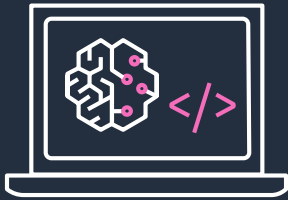


Write secure code

¹Gartner: [2023 Planning Guide for Application Development](#)

CodeWhisperer

BUILD APPLICATIONS FASTER AND MORE SECURELY WITH YOUR AI CODING COMPANION



Generate code suggestions in real time



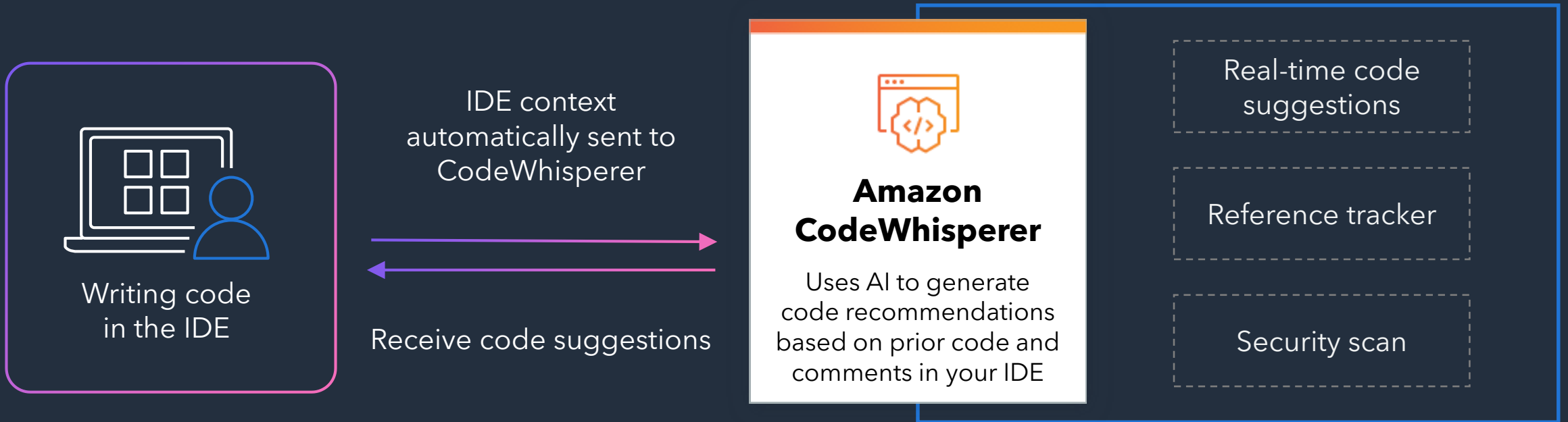
Scan code for hard-to-find vulnerabilities



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

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

How it works



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

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  
respons
```

```
return
```

✔ CodeWhisperer reference log

[3/24/2023, 3:17:34 PM] Accepted recommendation with code `response.Contents.forEach((file) => files.push(file.Key))` with reference under MIT from repository `function-template/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 condition lead to accidentally writing production data into test a

⚠ The elevated privilege level required to perform operation immediately after the operation is performed. [Line 1

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

Use with your tools

BUILD APPLICATIONS FASTER AND MORE SECURELY WITH YOUR AI CODING COMPANION



Code generation



AND

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



AND

Amazon SageMaker Studio, CLion,
GoLand, WebStorm, Rider, PhpStorm,
RubyMine, and DataGrip


Demos

slalom



Eimear Loughnane

Director of Software Engineering, Slalom

python > 01_basics >  example1.py > ...

```
1 fakeUsers = [  
2     { "name": "George Washington", "id": "user1" },  
3 ]
```


example3a.sql

```
1  -- table of students
2  CREATE TABLE students(
3      student_id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
4      first_name VARCHAR(64),
5      last_name VARCHAR(64),
6      email_address VARCHAR(64)
7  );
8
9  -- table of courses
10 CREATE TABLE courses(
11     course_id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
12     course_number
13     course_name VARCHAR(64),
14     course_description VARCHAR(64),
15 );
16
17 -- table of teachers
18 CREATE TABLE teachers(
19     teacher_id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
20     first_name VARCHAR(64),
21     last_name VARCHAR(64),
22     email_address VARCHAR(64)
23 );
24
25 -- add fake students for testing
26 insert into students(first_name, last_name, email_address)
27 VALUES
28 ('George', 'Washington', 'XXXXXXXXXXXXXXXXX'),
```

```
1  -- table of students
2  CREATE TABLE students(
3      student_id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
4      first_name VARCHAR(64),
5      last_name VARCHAR(64),
6      email_address VARCHAR(64)
7  );
8
9  -- table of courses
10 CREATE TABLE courses(
11     course_id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
12     course_number
13     course_name VARCHAR(64),
14     course_description VARCHAR(64),
15 );
16
17 -- table of teachers
18 CREATE TABLE teachers(
19     teacher_id INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
20     first_name VARCHAR(64),
21     last_name VARCHAR(64),
22     email_address VARCHAR(64)
23 );
24
25 -- add fake students for testing
26 insert into students(first_name, last_name, email_address)
27 VALUES
28 ('George', 'Washington', 'XXXXXXXXXXXXXXXXX'),
```


example6a.sql

1 -- list student_id, and average grade

2 SELECT s.student_id, AVG(g.grade)

3 FROM students s

4 INNER JOIN grades g

5 ON s.studentid = g.studentid

6 GROUP BY s.student_id

7


8 -- list student_id, and average grade using a correlated subquery

1 -- list student_id, and average grade
2 SELECT s.student_id, AVG(g.grade)
3 FROM students s
4 INNER JOIN grades g
5 ON s.studentid = g.studentid
6 GROUP BY s.student_id

example5a.sql

```
16
17 -- table of grades for each student enrolled in a course
18 CREATE TABLE grades(
19     student_id INT,
20     course_id INT,
21     grade float
22 );
23
24 -- list student_id, full_name, and average grade for each student.
25 SELECT s.student_id, s.first_name + ' ' + s.last_name AS full_name, AVG(grade) AS average_grade
26 FROM students s
27 INNER JOIN grades
28 ON students.student_id = grades.student_id
29 GROUP BY students.student_id, students.first_name, students.last_name;
30
31 -- Add an index to the grades and students table to optimize the above query
```

```
16
17 -- table of grades for each student enrolled in a course
18 CREATE TABLE grades(
19     student_id INT,
20     course_id INT,
21     grade float
22 );
23
24 -- list student_id, full_name, and average grade for each student.
25 SELECT s.student_id, s.first_name + ' ' + s.last_name AS full_name, AVG(grade) AS average_grade
26 FROM students s
27 INNER JOIN grades
28 ON students.student_id = grades.student_id
29 GROUP BY students.student_id, students.first_name, students.last_name;
30
31 -- Add an index to the grades and students table to optimize the above query
```

```
python > 01_basics >  example2.py
```

```
1 # Function to upload a file to an S3 bucket
```



```
[ ]: import pyodbc
server = 'tcp:myserver.database.windows.net'
database = 'mydb'
username = 'myusername'
password = 'mypassword'
```

```
[ ]: # Connect to the database
cnxn = pyodbc.connect('DRIVER={ODBC Driver 17 for SQL Server};SERVER='+server+';DATABASE='+database+';UID='
cursor = cnxn.cursor()
```

```
[ ]: # Get the number records in the students table
```





Code



Glue PySpark



```
spark = glueContext.spark_session
job = Job(glueContext)
```

Example: Connect to SQL Server and get list of students

```
[ ]: from pyspark.sql import SparkSession
server = 'jdbc:sqlserver://localhost;database=demo;user=myusername;password=mypassword'

# Create Spark Session
spark = SparkSession.builder.appName("Spark SQL").getOrCreate()

# Create DataFrame from SQL Server
```

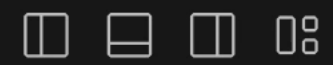
Example: Visualize data with matplotlib

```
[ ]: import matplotlib.pyplot as plt

# Set X-axis and Y-axis values
x = [5, 2, 8, 4, 9]
y = [10, 4, 8, 5, 2]

# Create a bar chart
```

Security Scanning



example1.py ×



08_security > example1.py > ...

```
1 # Running a command without validating input
2 import subprocess
3 cmd = input('Enter a command to run. ');
4 subprocess.Popen(cmd, shell=True, stdin=subprocess.PIPE)
```

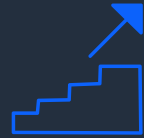


Impact on engineering



What we did

- 4 pods, 2 in London 2 in Denver
- Half used Code Whisperer, half did not
- Identical code base, identical backlog
- No training or previous experience on code base or on Code Whisperer
- 1 sprint to complete



Deliver productivity

Participants who used Code Whisperer completed their tasks 20-50% faster than those who did not use Code Whisperer

Inexperienced engineers saw a bigger boost than more experienced engineers



Developer happiness

Participants who used Code Whisperer reported increases in work satisfaction and engagement

Developers spent less time doing mundane tasks and more time on satisfying work

Developers spent more time 'in the flow'

Getting Started

UPDATE THIS PRESENTATION HEADER IN SLIDE MASTER

Code whisperer implementation

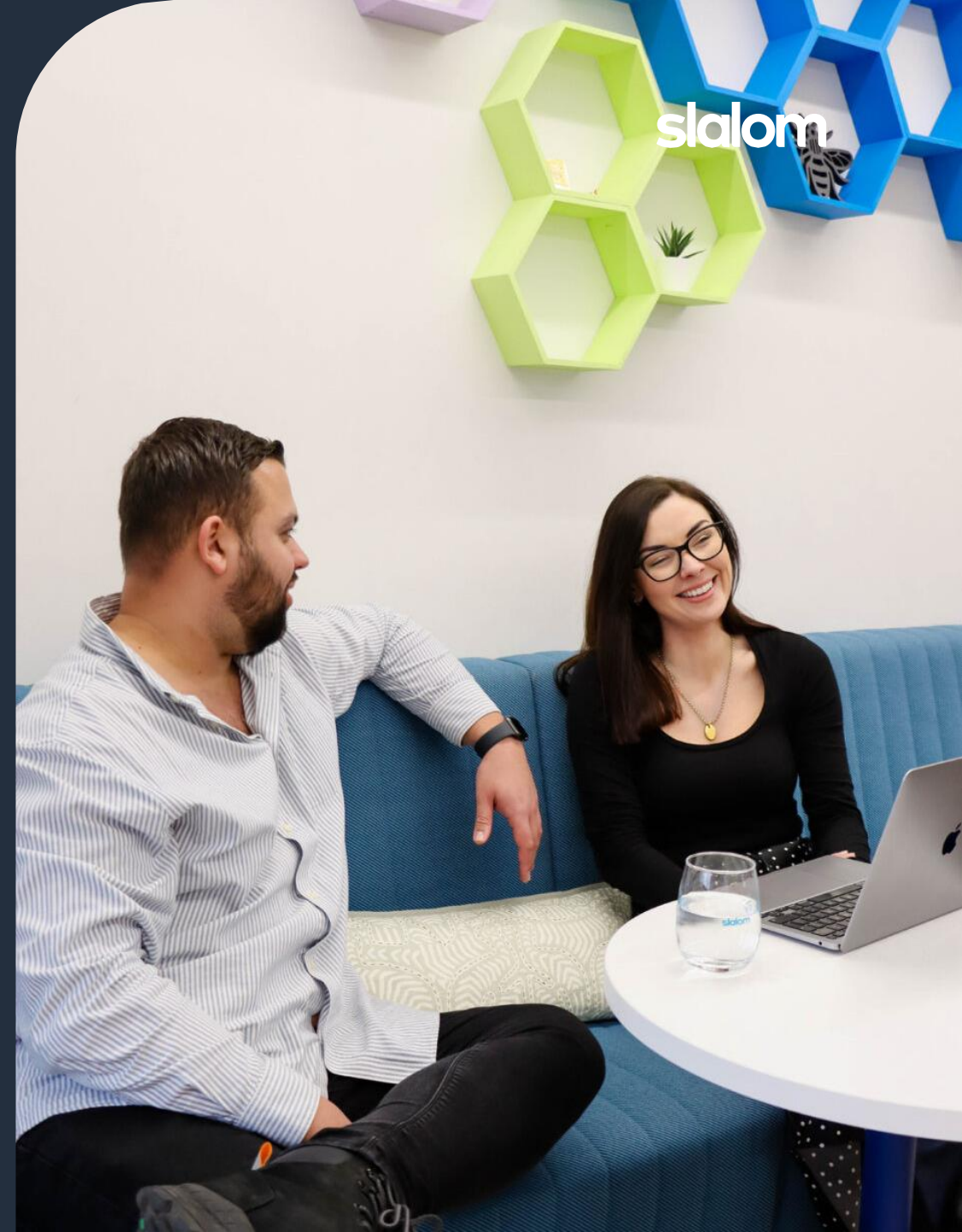
Key considerations for your adoption plan:

- **Leverage AWS support**
AWS funding options available for qualifying POCs
- **Manage like a project**
Define your champions and team to realise the value of the tool
- **Plan your phasing**
Be deliberate in your team choice for usage
- **Deliver an outcome**
Use this as an opportunity to address a pain point / new feature
- **It's all about engineering culture**
Promote it, critique it, show the value to the team!



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

© SLALOM. ALL RIGHTS RESERVED. PROPRIETARY AND CONFIDENTIAL.



Three stages for Gen AI success



Educate & Enable

- Participate in workshop
- Explore implementations
- Pick a challenge
- Establish success criteria



Experiment & Learn

- Develop a Proof of Concept
- Set up an innovation lab
- Host a hackathon



Activate

- Evaluate metrics
- Define a path to production

Generative AI workshops

1

Code Whisperer

- A deep dive into Amazon Code Whisperer
- How to adapt to your software engineering ecosystem
- Pilot planning and impact measurement
- Code Whisperer Product Roadmap

2

Generative AI foundations

- AI landscape: Model options and tradeoffs
- Options for technical infrastructure, data quality, and engineering
- Responsible AI: Risk, privacy, and use policy
- High-level roadmap for AI project implementation

3

Generative AI proof of concepts

- Interactive chatbot using Amazon Bedrock for an automotive customer
- Evaluating animation character generation for the gaming industry
- Field engineering troubleshooter using Amazon Bedrock



Thank you!

Damian Niezgoda

 @dniegoda

Eimear Loughnane

 @eimearloughnane



Please complete
the session survey