



SUMMIT  
ONLINE

B U I 1 0

# A builder's guide to testing

**Bikram Bora**

Solutions Architect  
Amazon Web Services

# The journey of a line of code



# The journey of a line of code

---

## Dev

Iteratively write, test, debug

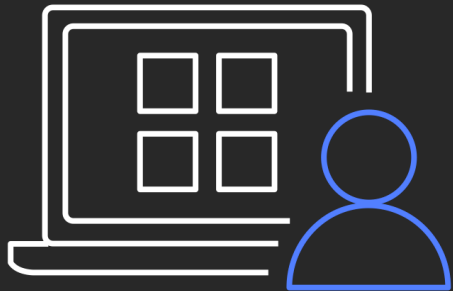


# The journey of a line of code

---

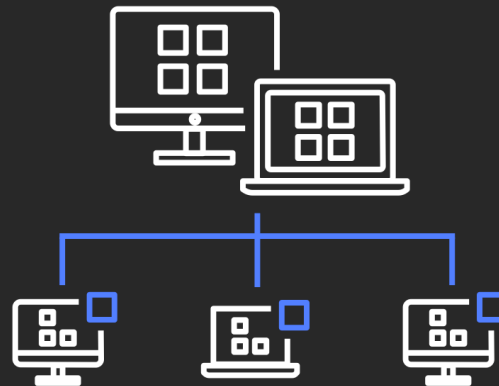
## Dev

Iteratively write, test, debug



## Integration

Integration, performance, acceptance criteria



# The journey of a line of code

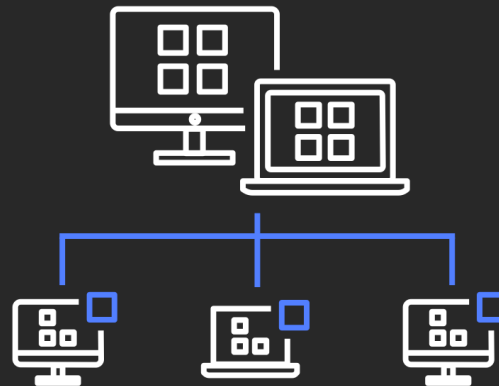
## Dev

Iteratively write, test, debug



## Integration

Integration, performance, acceptance criteria



## Deploy

Performance, load, monitoring

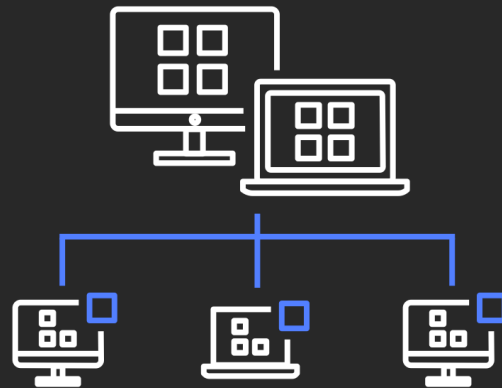


# The journey of a line of code

Dev  
and test



Integration  
and test



Deploy  
and test

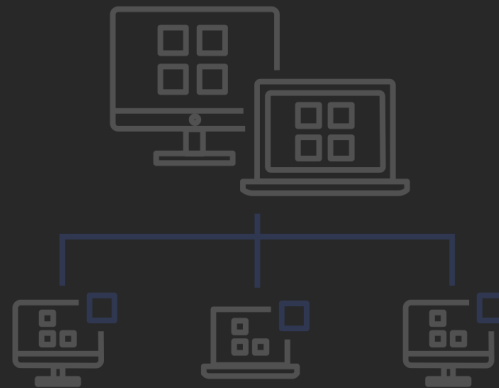


# The journey of a line of code

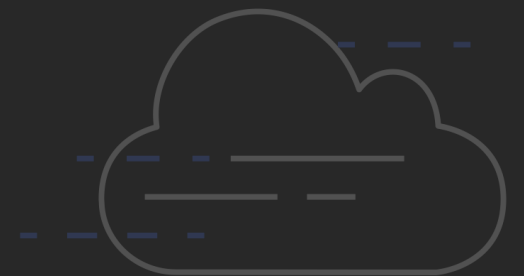
Dev  
and test



Integration  
and test



Deploy  
and test





# The dev phase

## Dev and test

Iteratively write, test, debug



Native AWS integration for IDEs & dev platforms

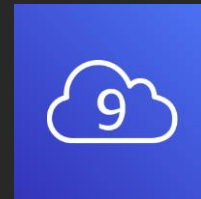
# The dev phase

## Dev & test

Iteratively write, test, debug



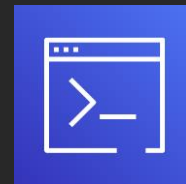
## Native AWS integration for IDEs & dev platforms



AWS Cloud9



Eclipse



AWS Command Line Interface  
Version 2.0

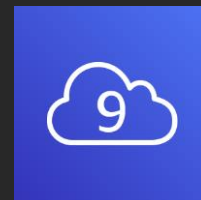
# The dev phase

## Dev and test

Iteratively write, test, debug



## Native AWS integration for IDEs & dev platforms



AWS Cloud9



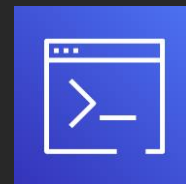
Visual Studio



VS Code



Eclipse



AWS Command Line Interface  
Version 2.0

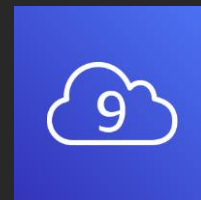
# The dev phase

## Dev and test

Iteratively write, test, debug



## Native AWS integration for IDEs & dev platforms



AWS Cloud9



Visual Studio



VS Code



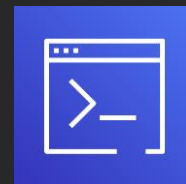
Eclipse



IntelliJ IDEA



PyCharm



AWS Command Line Interface  
Version 2.0

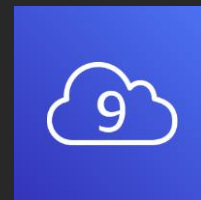
# The dev phase

## Dev and test

Iteratively write, test, debug



## Native AWS integration for IDEs & dev platforms



AWS Cloud9



Visual Studio



VS Code



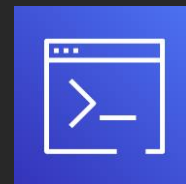
Eclipse



IntelliJ IDEA



PyCharm



AWS Command Line Interface  
Version 2.0



AWS Amplify for  
iOS



AWS Amplify for  
Android

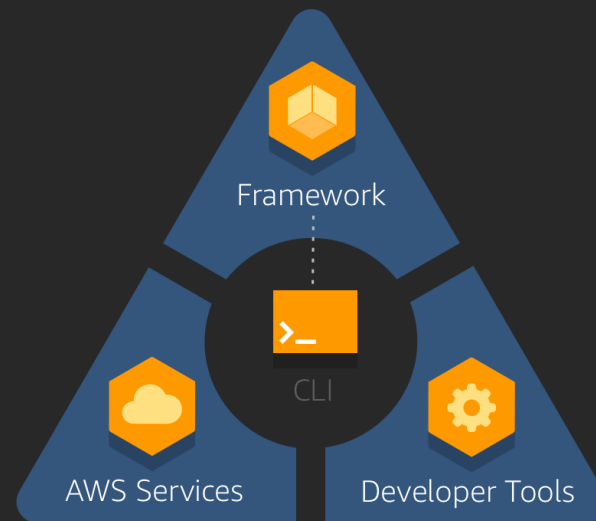
# The dev phase

## Dev and test

Iteratively write, test, debug



### AWS Amplify



### AWS Serverless Application Model (SAM)



# The dev phase

Locally run and test cloud services

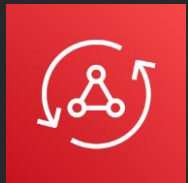
---

# The dev phase

## Locally run and test cloud services



Amazon API Gateway  
Local



AWS AppSync  
Local



# The dev phase

## Locally run and test cloud services



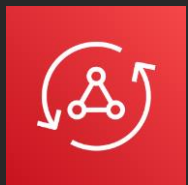
Amazon API Gateway  
Local



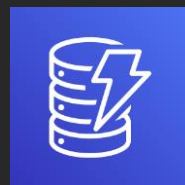
AWS Lambda  
Local



AWS Step Functions  
Local



AWS AppSync  
Local



Amazon DynamoDB  
Local

# The dev phase

## Locally run and test cloud services



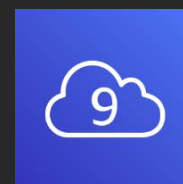
Amazon API Gateway  
Local



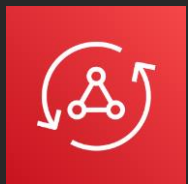
AWS Lambda  
Local



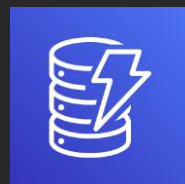
AWS Step Functions  
Local



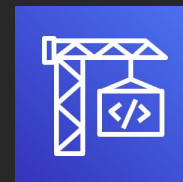
AWS Cloud9  
Local



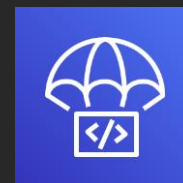
AWS AppSync  
Local



Amazon DynamoDB  
Local



AWS CodeBuild  
Local



AWS CodeDeploy  
Local

# The dev phase

## Locally run and test cloud services



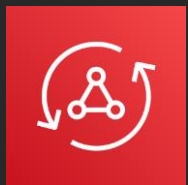
Amazon API Gateway  
Local



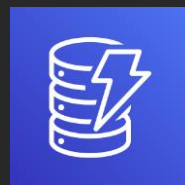
AWS Lambda  
Local



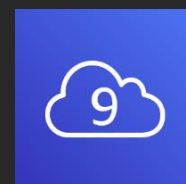
AWS Step Functions  
Local



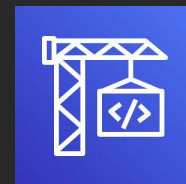
AWS AppSync  
Local



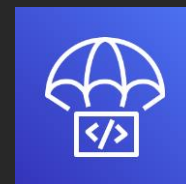
Amazon DynamoDB  
Local



AWS Cloud9  
Local



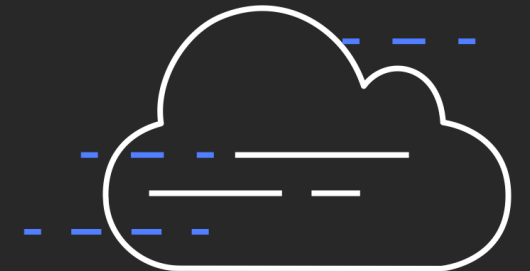
AWS CodeBuild  
Local



AWS CodeDeploy  
Local

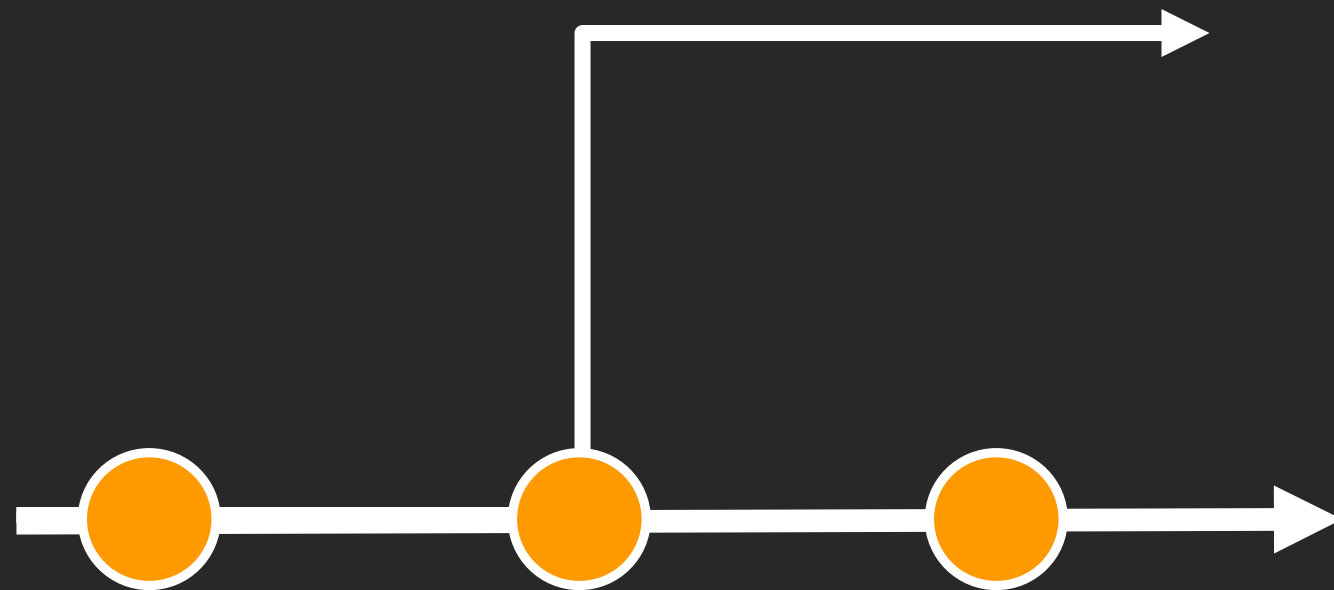


## Deploy in AWS



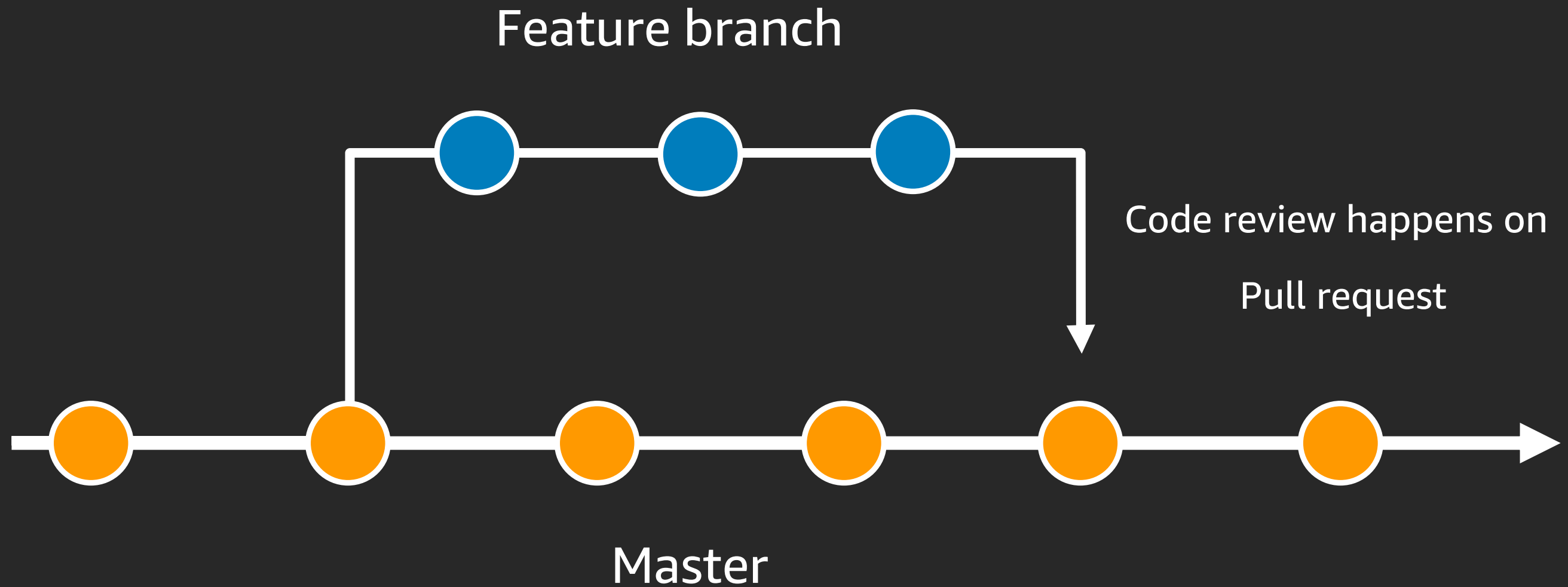
New branch

Feature branch



Master

# Branch to be merged



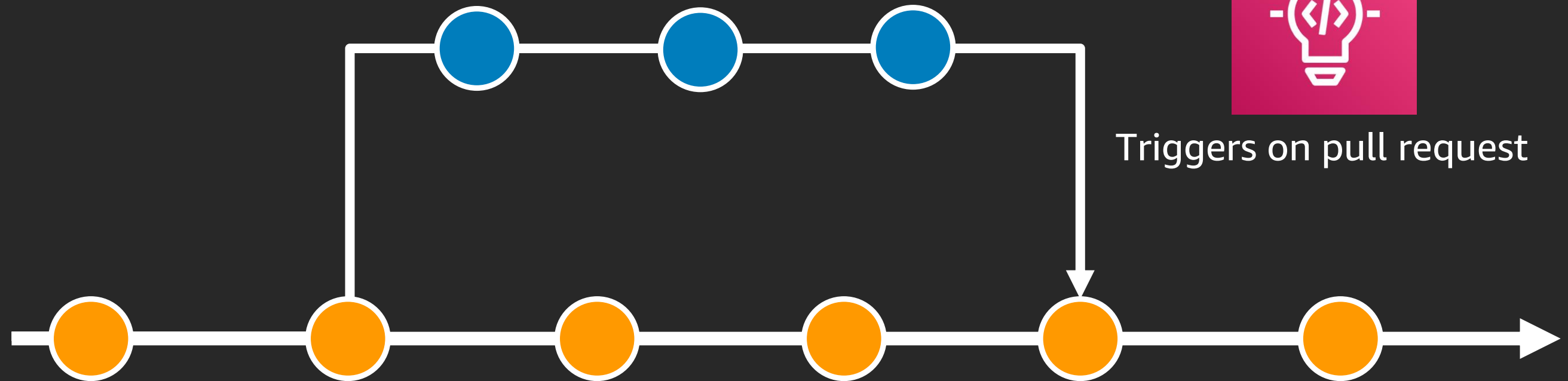
# Amazon CodeGuru: Reviewer

CodeGuru Reviewer

Feature branch



Triggers on pull request



Master

# Amazon CodeGuru: Reviewer

## CodeGuru Reviewer

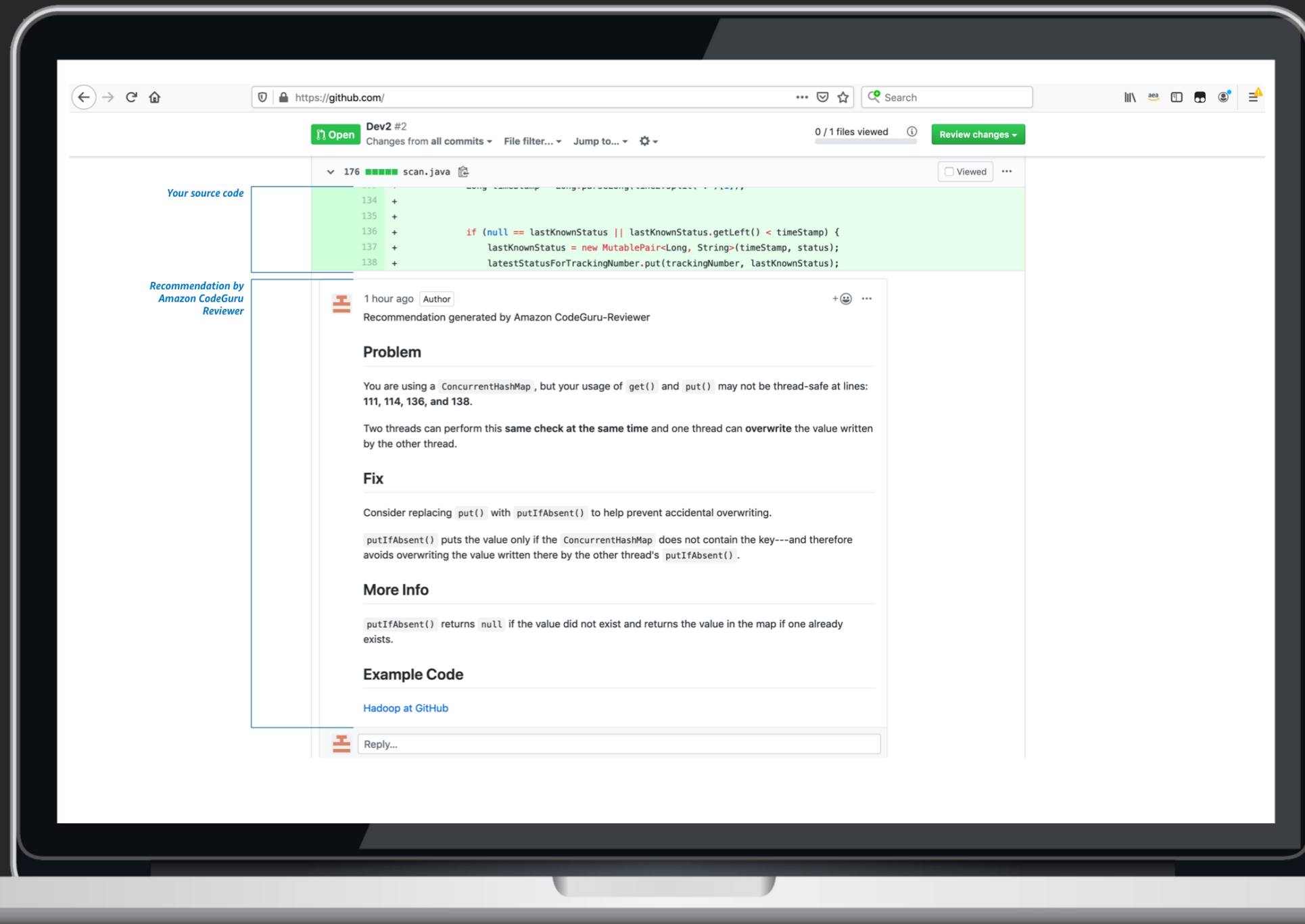


- Finds issues in your code and recommends how to remediate them
- You can associate existing code repositories on GitHub or AWS CodeCommit



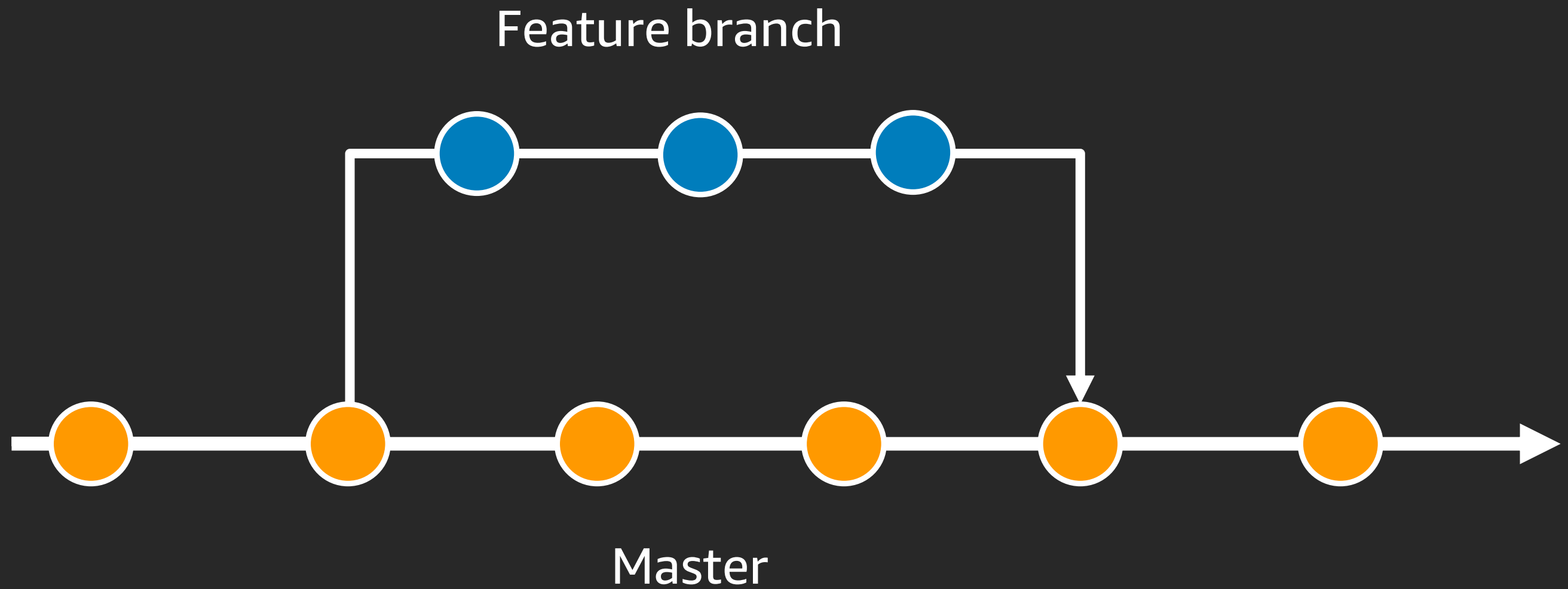
AWS CodeCommit

# Amazon CodeGuru: Reviewer demo





Pull request accepted

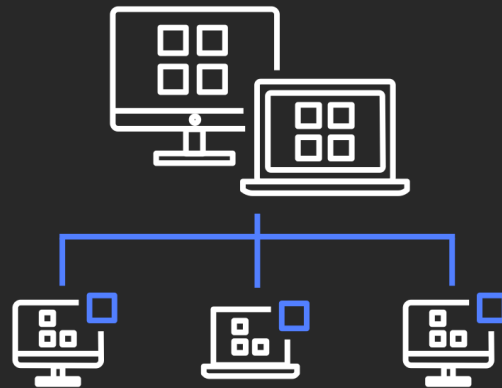


# The journey of a line of code

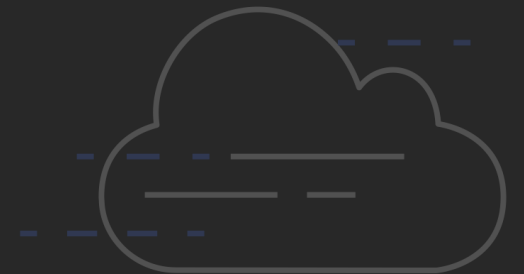
Dev &  
Test



Integration &  
test



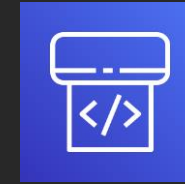
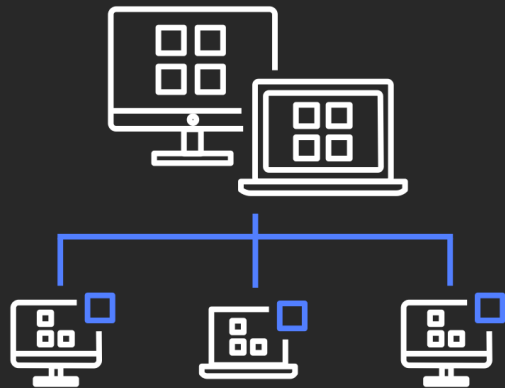
Deploy &  
Test



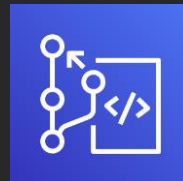
# The build environment

## Integration and test

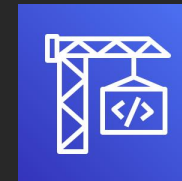
Integration, performance,  
acceptance criteria



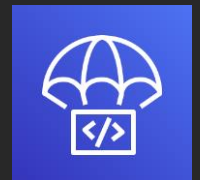
AWS CodePipeline



AWS CodeCommit



AWS CodeBuild



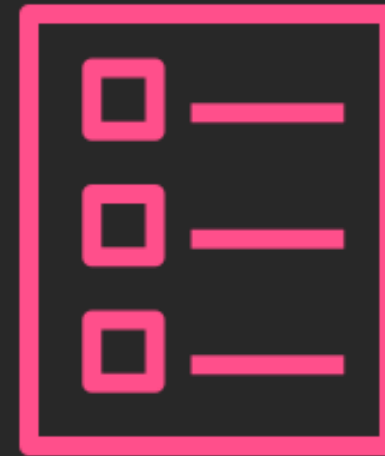
AWS CodeDeploy

# AWS CodeBuild

---



AWS CodeBuild



Buildspec.yml

# 1. Setup environment



```
version: 0.2
```

```
run-as: Linux-user-name
```

```
env:
```

```
  variables:  
    key: "value"  
  parameter-store:  
    key: "value"  
  exported-variables:  
    - variable  
    - variable  
  secrets-manager:  
    key: secret-id:json-key:version-stage:version-id  
  git-credential-helper: yes
```

1. Setup environment
2. Configure phases



phases:

install:

```
run-as: Linux-user-name
runtime-versions:
runtime: version
commands:
- command
finally:
- command
```

pre\_build:

```
run-as: Linux-user-name
commands:
- command
finally:
- command
```

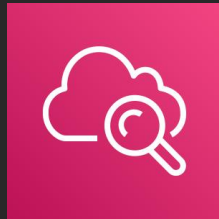
build:

```
run-as: Linux-user-name
commands:
- command
finally:
- command
```

post\_build:

```
run-as: Linux-user-name
commands:
- command
finally:
- command
```

1. Setup environment
2. Configure phases
3. Setup output reports



## reports:

```
report-name-or-arn:  
files:  
- location  
base-directory: location  
discard-paths: yes  
file-format: JunitXml | CucumberJson | VisualStudioTrx | TestNGXml
```

1. Setup environment
2. Configure phases
3. Setup output reports
4. Save build artifacts



## artifacts:

```
files:
- location
name: artifact-name
discard-paths: yes
base-directory: location
secondary-artifacts:
artifactIdentifier:
files:
- location
name: secondary-artifact-name
discard-paths: yes
base-directory: location
artifactIdentifier:
files:
- location
discard-paths: yes
base-directory: location
cache:
paths:
- path
```



# AWS CodeBuild



AWS CodeBuild



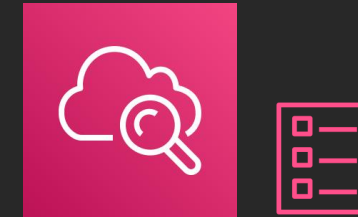
1. Download source code



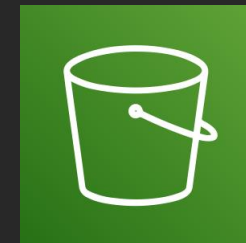
2. Build the code in a managed fresh environment



3. Run testing suite



4. Stream Logs to Amazon CloudWatch



5. Get built artifacts in Amazon S3

# UI testing at scale on AWS



Blackboard LMS

# UI testing at scale on AWS



Blackboard LMS



Selenium UI Tests

# UI testing at scale on AWS



Blackboard LMS



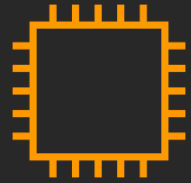
Selenium UI Tests



Reduce Test Time

# UI testing at scale on AWS

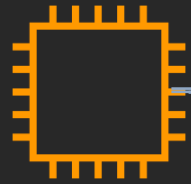
Test Runner initiates tests



# UI testing at scale on AWS

Selenium + Chromium  
running in Lambda

Test Runner initiates tests

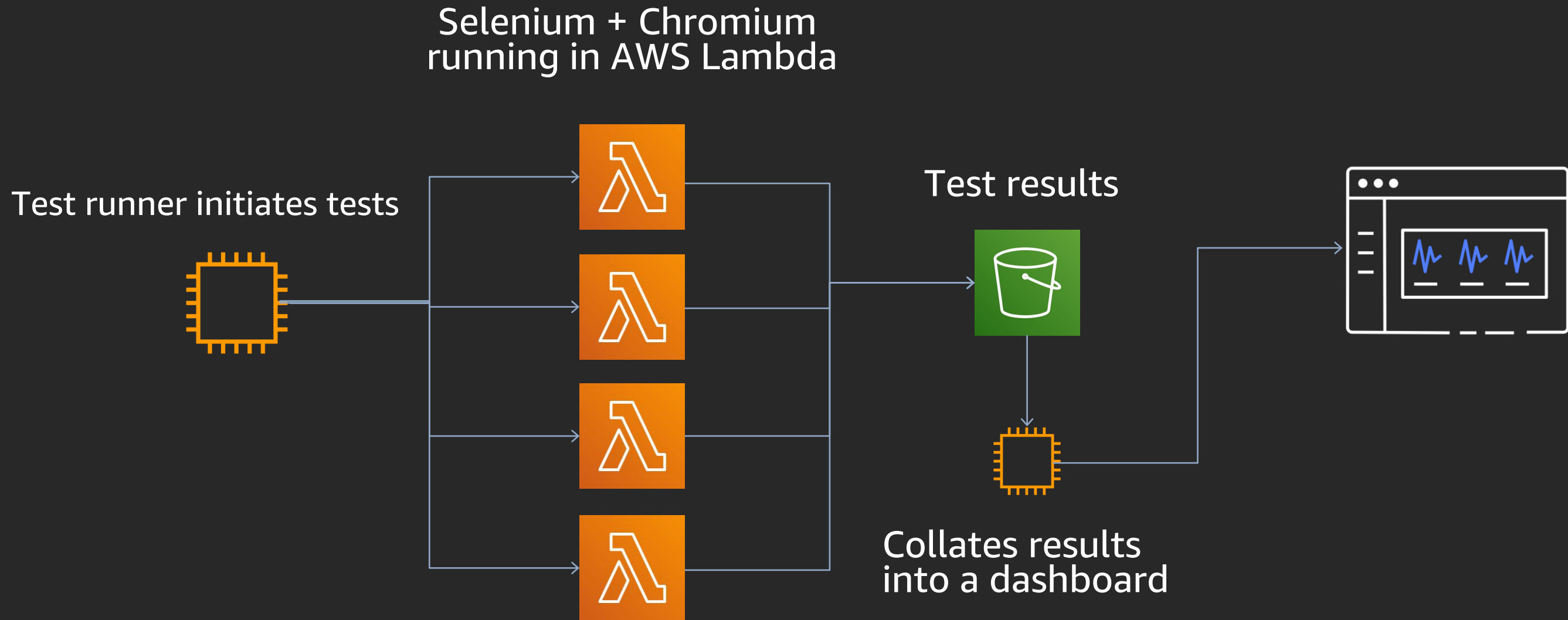


# UI testing at scale on AWS

Selenium + Chromium  
running in AWS Lambda



# UI testing at scale on AWS



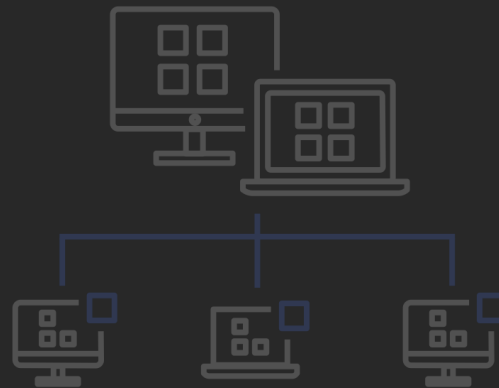


# The journey of a line of code

Dev  
and test



Integration  
and test



Deploy  
and test



# The staging environment

---

## Deploy and test

Performance, load, monitoring



Performance and load testing

Canary deployments

A/B testing

Monitoring and optimisation

# Load testing tooling

# Load testing tooling



Locust.io

# Load testing tooling



Locust.io



Apache JMeter

# Load testing tooling



Locust.io



Apache JMeter



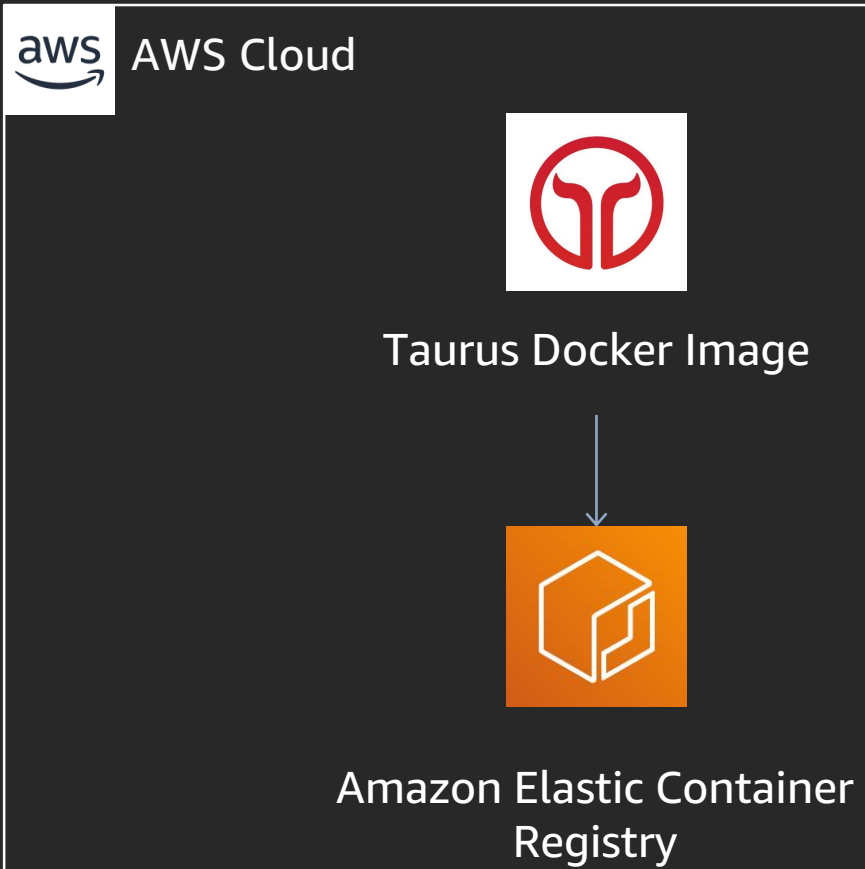
Taurus

# Distributed load testing on AWS



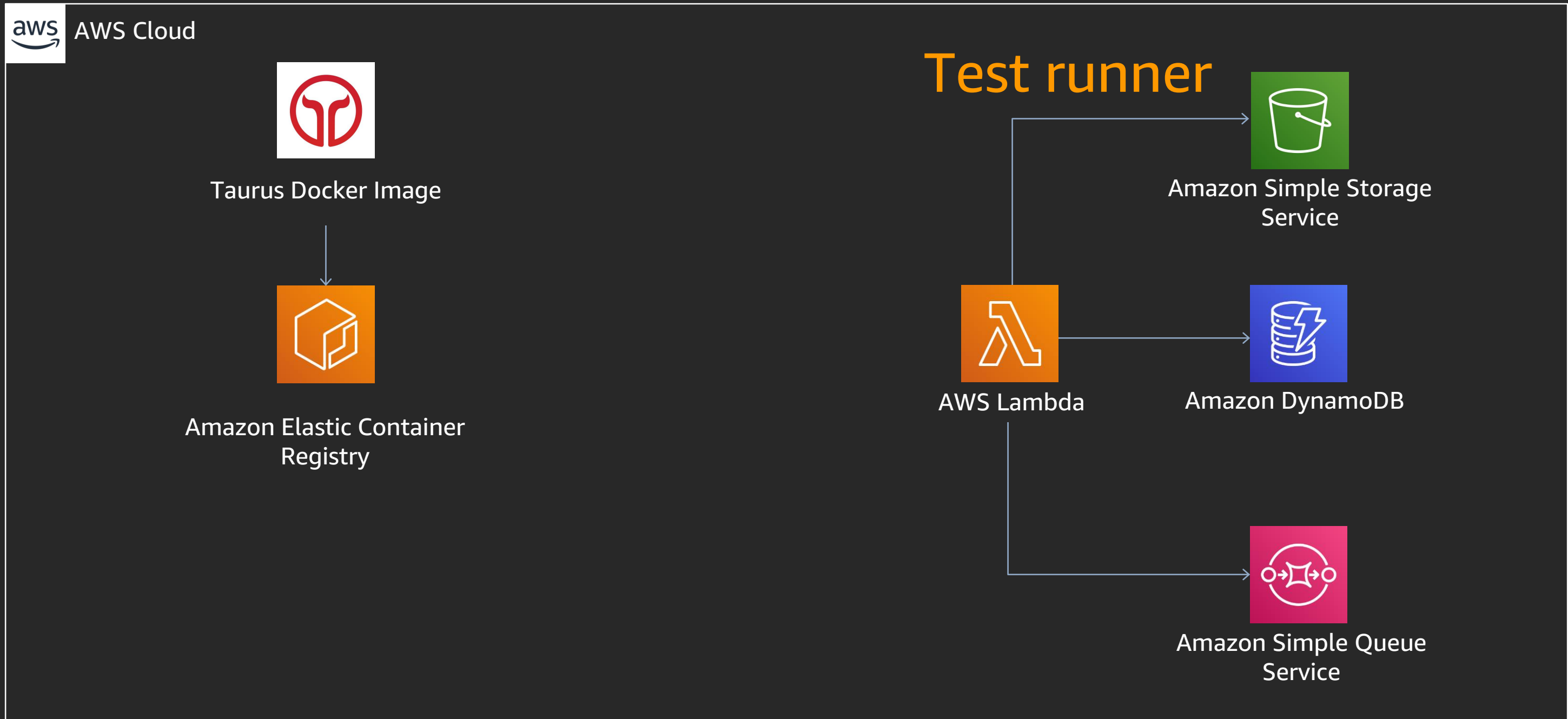
AWS Cloud

# Distributed load testing on AWS

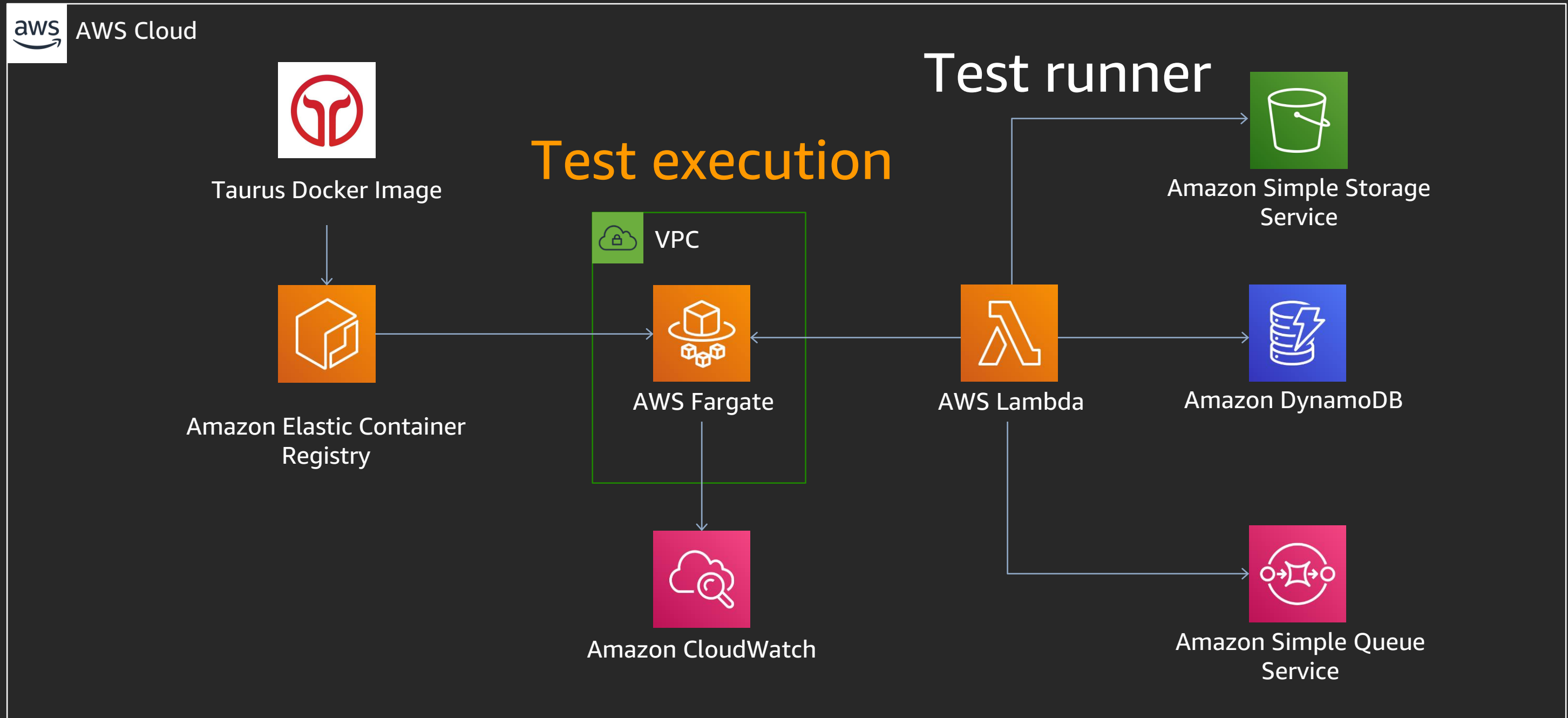




# Distributed load testing on AWS

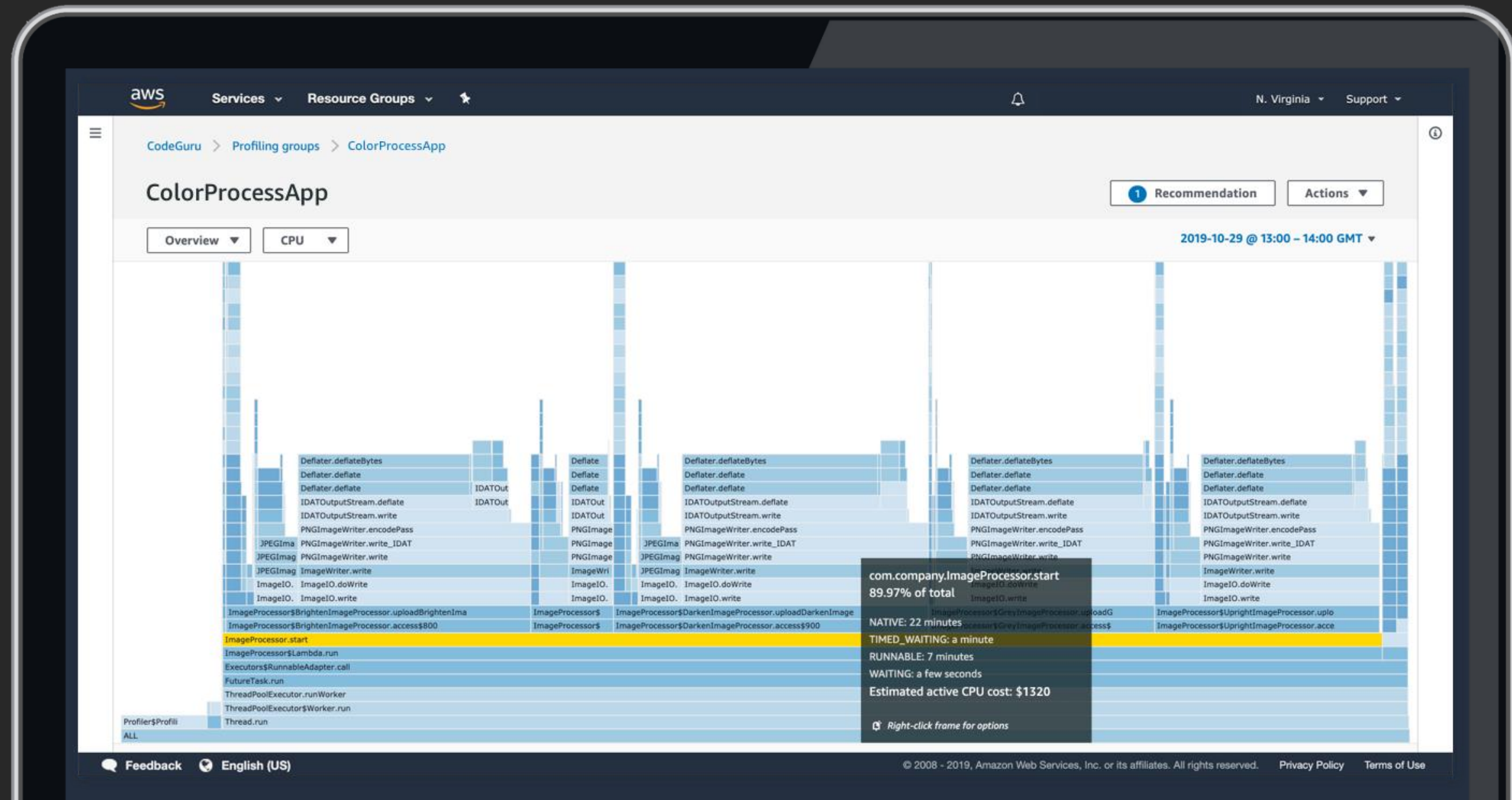


# Distributed load testing on AWS



# Amazon CodeGuru: Profiler demo

Find your most expensive lines of code



# The journey of a line of code



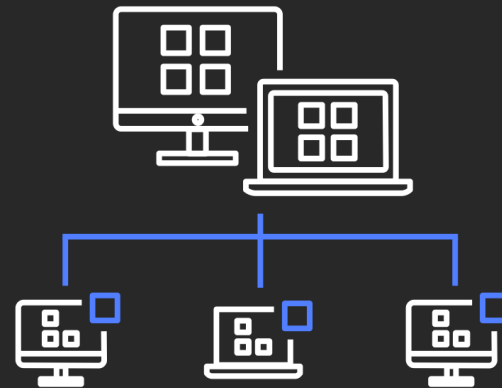
## Dev and test

Iteratively write, test, debug



## Integration and test

Integration, performance, acceptance criteria



## Deploy and test

Performance, load, monitoring

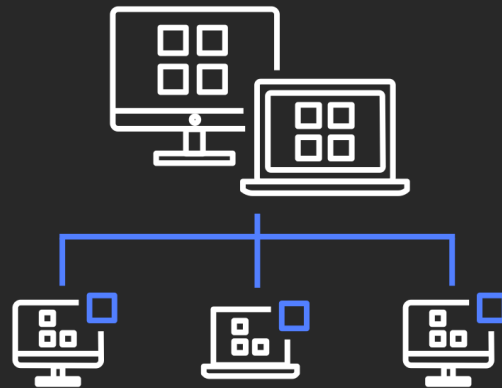


# The journey of a line of code

Dev  
and test



Integration  
and test



Deploy  
and test



# Thank you!