AWS re:Inforce

JUNE 13 - 14, 2023 | ANAHEIM, CA

APS371

Integrating open-source security tools with the AWS code services

Daniel Begimher

Senior Security Architect
Global Services Security
AWS

Pat Gaw

Principal Security Consultant
Global Services Security
AWS

Agenda

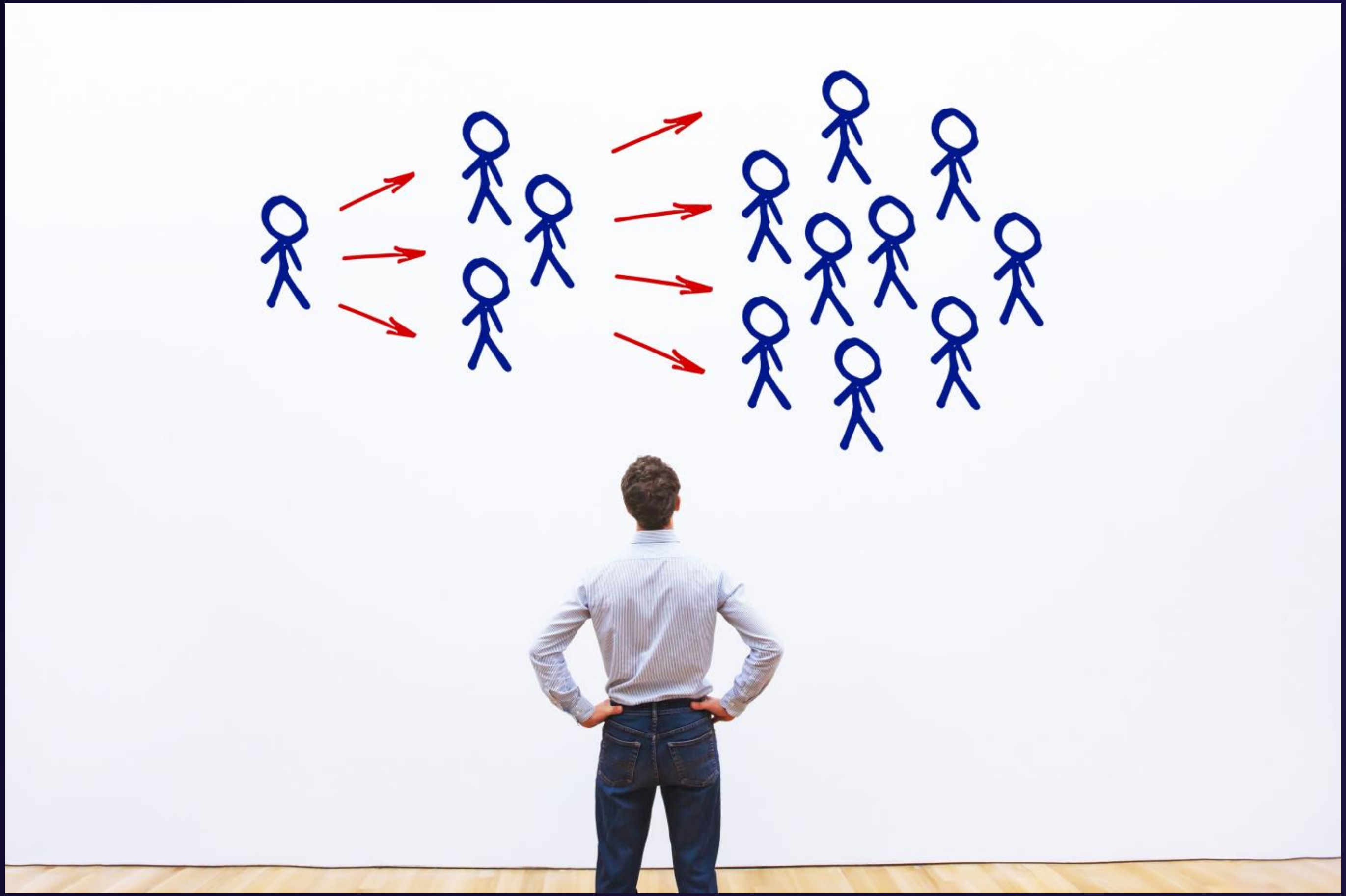
Application security challenges

Application security principles at scale

Automated security helper (ASH)

Hands-on labs

Scaling application security is challenging for any organization



How AWS scales application security

> “Service teams own the security of their service; AWS Security owns the security of AWS.”

CJ Moses

Chief Information Security Officer, AWS

Core principles of scaling application security

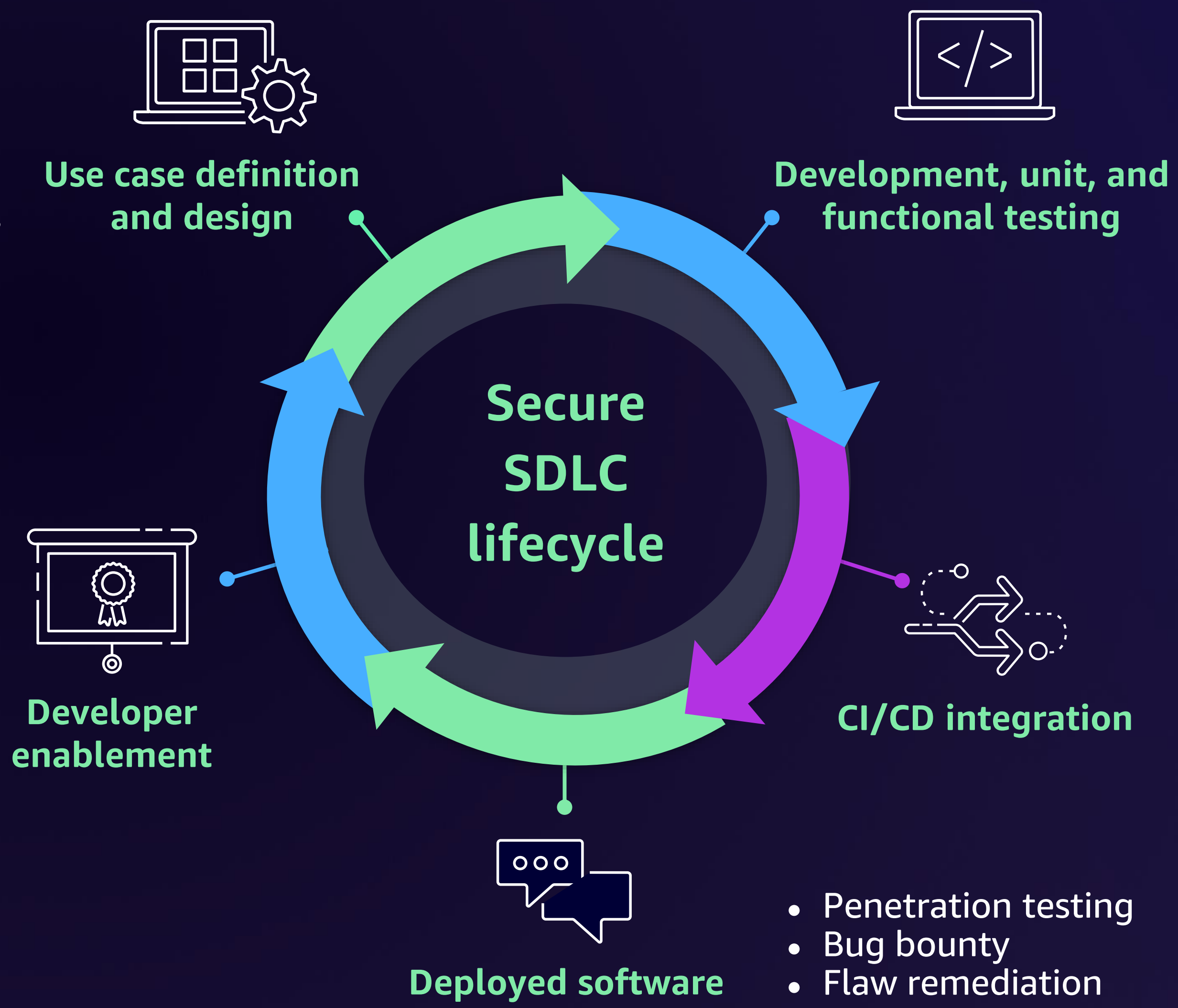
Organization

Clear expectations

Metrics

Automation

What does a secure SDLC look like?



Use case definition and design

- Threat modeling
- Security design reviews

Development, unit, and functional testing

- Secure code review
- Dynamic application security testing
- Security unit testing

CI/CD integration

- Static analysis security testing (SAST)
- Software composition analysis (SCA)
- Infrastructure as code (IaC) misconfigurations
- Software bill of materials (SBOM)
- Dynamic analysis security testing (DAST)

Deployed software

- Penetration testing
- Bug bounty
- Flaw remediation

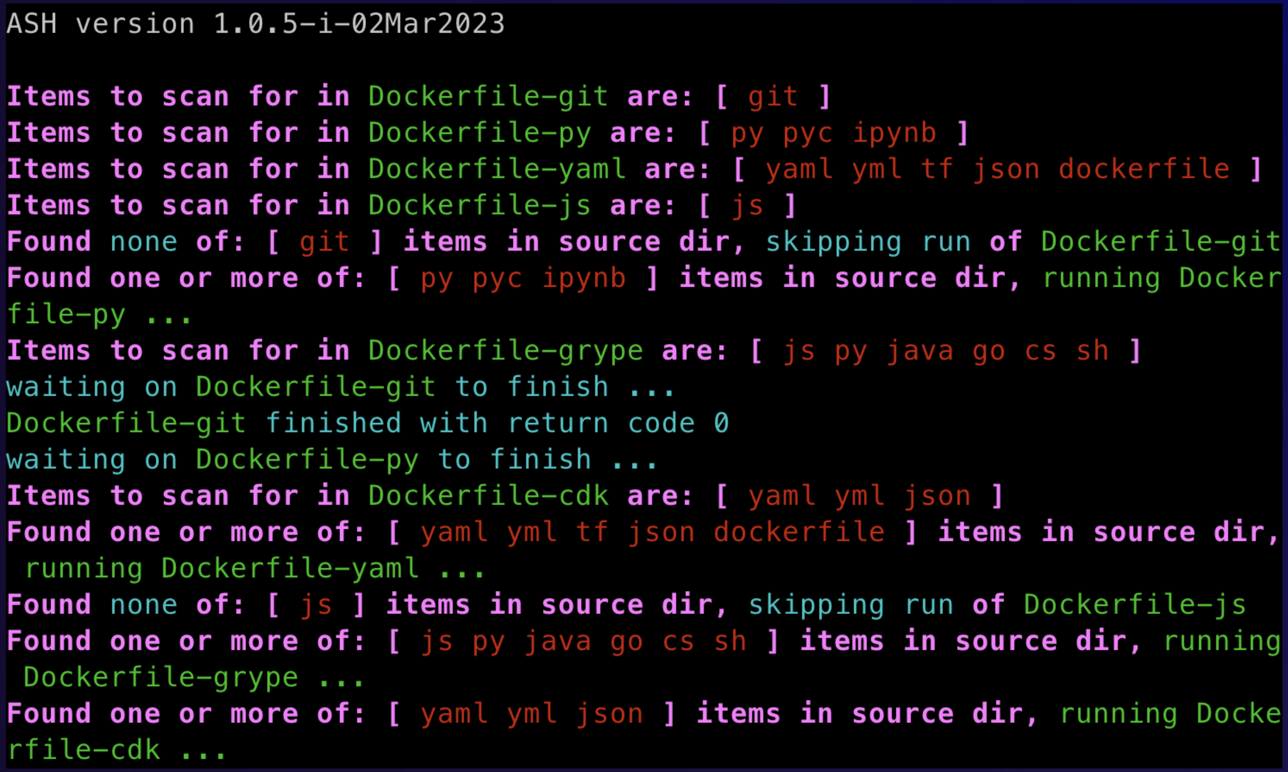
Developer enablement

- Security guardians
- Secure local IDE

How do we make it easier for developers to find security issues early?

What is automated security helper (ASH)?

- Simplifies deployment of multiple open source SAST, SCA, and IaC security scanning solutions
- Creates SBOM
- Detects security vulnerabilities and IaC misconfigurations
- Supports 12 different software languages and frameworks
- Simple to run and integrate at multiple points of the developer workflow

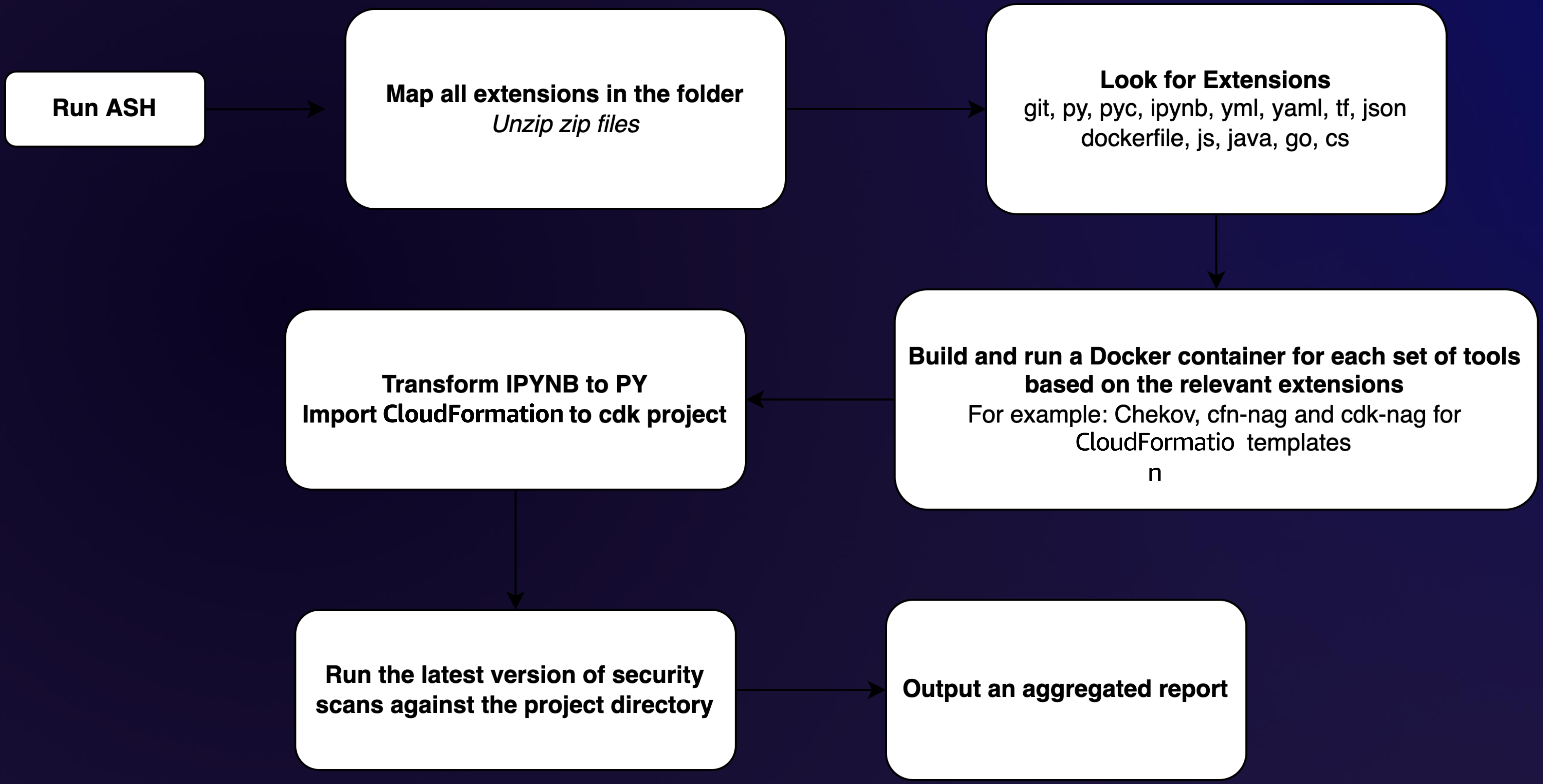


```
ASH version 1.0.5-i-02Mar2023

Items to scan for in Dockerfile-git are: [ git ]
Items to scan for in Dockerfile-py are: [ py pyc ipynb ]
Items to scan for in Dockerfile-yaml are: [ yaml yml tf json dockerfile ]
Items to scan for in Dockerfile-js are: [ js ]
Found none of: [ git ] items in source dir, skipping run of Dockerfile-git
Found one or more of: [ py pyc ipynb ] items in source dir, running Docker
file-py ...
Items to scan for in Dockerfile-grype are: [ js py java go cs sh ]
waiting on Dockerfile-git to finish ...
Dockerfile-git finished with return code 0
waiting on Dockerfile-py to finish ...
Items to scan for in Dockerfile-cdk are: [ yaml yml json ]
Found one or more of: [ yaml yml tf json dockerfile ] items in source dir,
 running Dockerfile-yaml ...
Found none of: [ js ] items in source dir, skipping run of Dockerfile-js
Found one or more of: [ js py java go cs sh ] items in source dir, running
 Dockerfile-grype ...
Found one or more of: [ yaml yml json ] items in source dir, running Docke
rfile-cdk ...
```

https://github.com/aws-samples/automated-security-helper

ASH architecture



Run ASH

Map all extensions in the folder
Unzip zip files

Look for Extensions
git, py, pyc, ipynb, yml, yaml, tf, json
dockerfile, js, java, go, cs

Build and run a Docker container for each set of tools based on the relevant extensions
For example: Chekov, cfn-nag and cdk-nag for CloudFormatio templates
n

Transform IPYNB to PY
Import CloudFormation to cdk project

Run the latest version of security scans against the project directory

Output an aggregated report

Lab time!

Module 1: Run ASH in your IDE

- Learning objective – understand how to use ASH in the local environment to identify security issues early in the development cycle
- High-level steps

❑ Log in to AWS Cloud9

❑ Scan your code with different open source tools

❑ Clone ASH

❑ Scan your code with ASH

❑ Review the results

❑ Fix the code

Module 2: Integrate ASH into the build pipeline

- Learning objectives – learn how to integrate ASH scanning into the software build pipeline
- High-level steps

❑ Provision CI pipeline leveraging AWS CodePipeline, AWS CodeCommit, and AWS CodeBuild

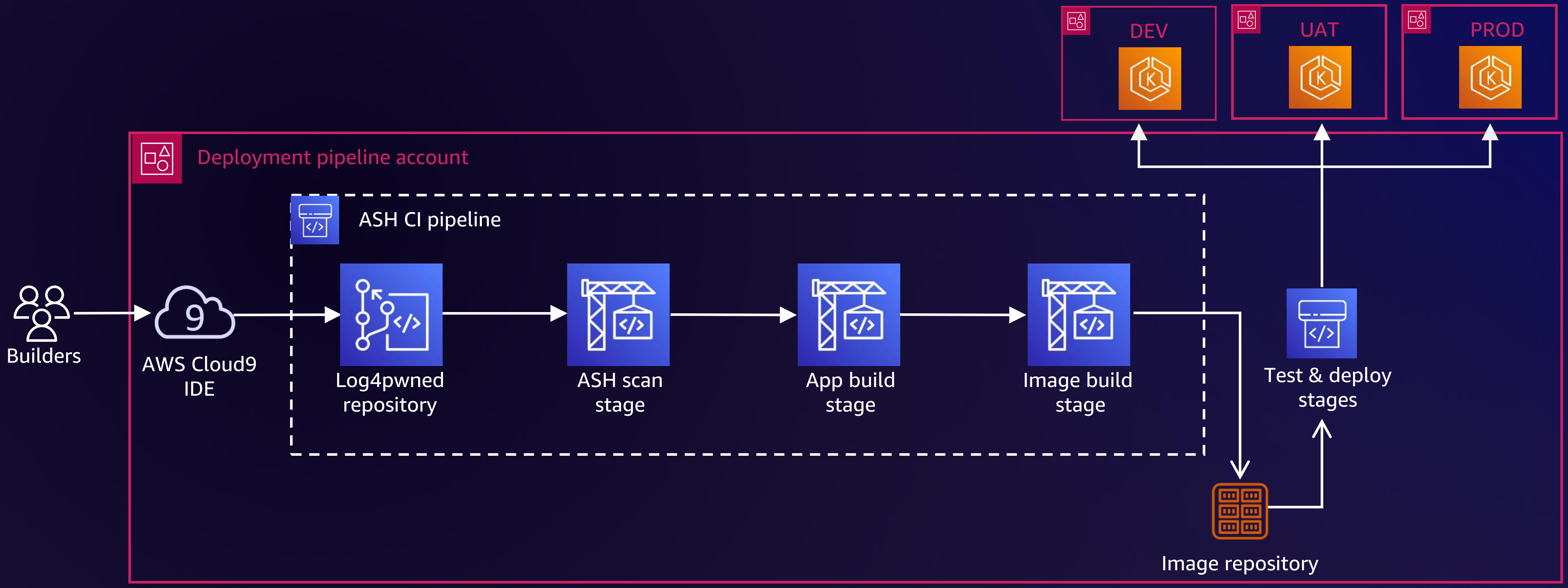
❑ Push a vulnerable Java application

❑ Integrate ASH in the pipeline

❑ Review results

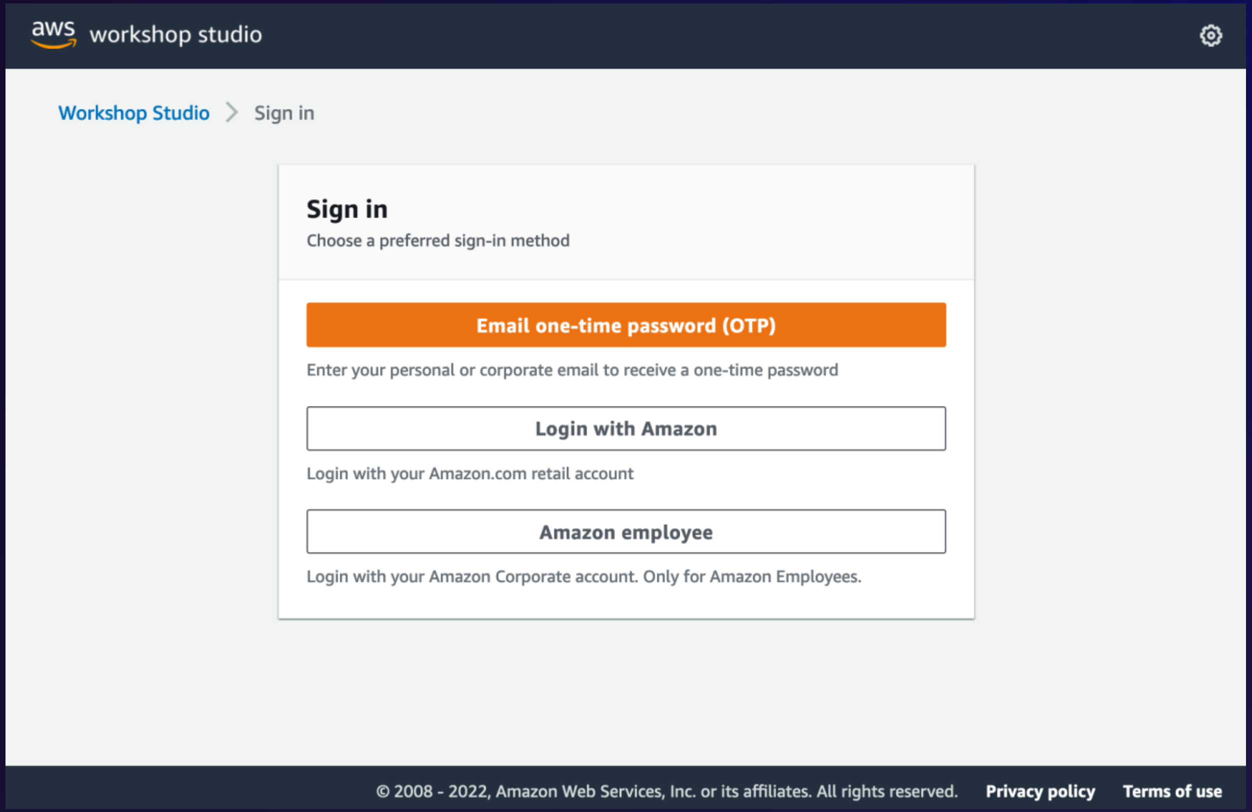
❑ Fix the application

Pipeline architecture with ASH security scanning



Step 1: Sign in via your preferred method

One-click join link: https://s12d.com/ash-workshop



Workshop Studio > Sign in

Sign in

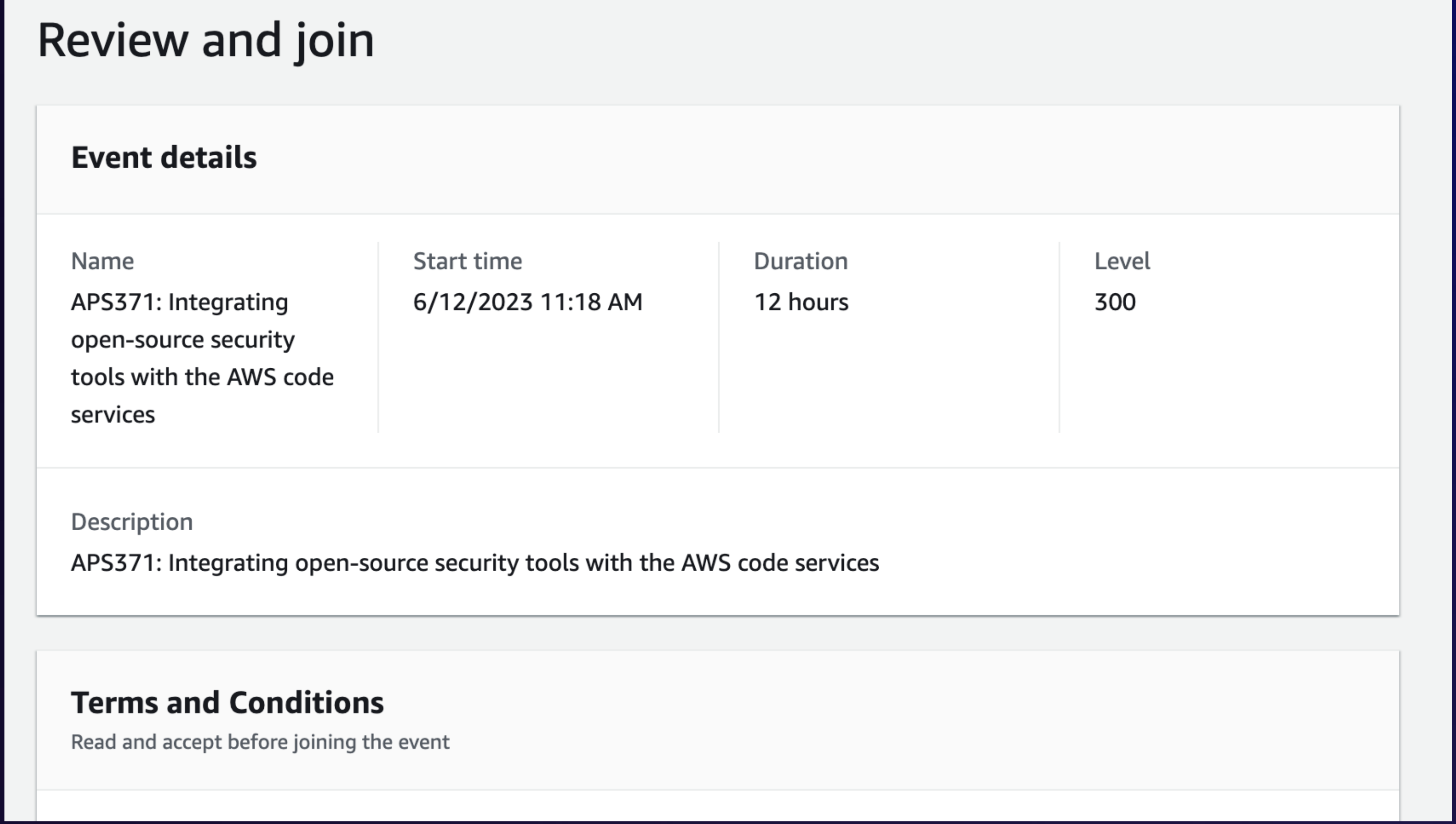
Choose a preferred sign-in method

Enter your personal or corporate email to receive a one-time password

Login with your Amazon.com retail account

Login with your Amazon Corporate account. Only for Amazon Employees.

Step 2: Review terms and join event



Review and join

Event details

Name	Start time	Duration	Level
APS371: Integrating open-source security tools with the AWS code services	6/12/2023 11:18 AM	12 hours	300

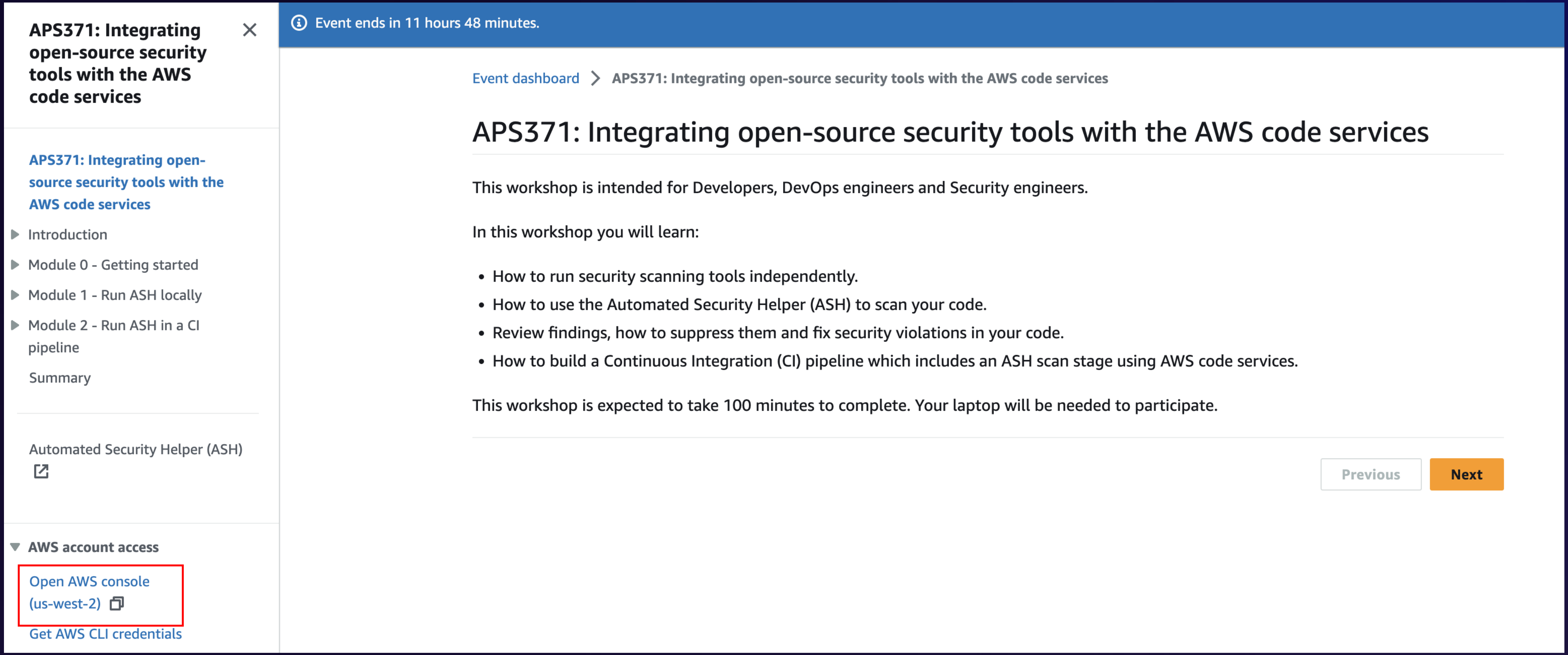
Description

APS371: Integrating open-source security tools with the AWS code services

Terms and Conditions

Read and accept before joining the event

Step 3: Access AWS account



APS371: Integrating open-source security tools with the AWS code services

APS371: Integrating open-source security tools with the AWS code services

- Introduction
- Module 0 - Getting started
- Module 1 - Run ASH locally
- Module 2 - Run ASH in a CI pipeline
- Summary

Automated Security Helper (ASH)

AWS account access

Open AWS console (us-west-2)

Get AWS CLI credentials

Event ends in 11 hours 48 minutes.

Event dashboard > APS371: Integrating open-source security tools with the AWS code services

APS371: Integrating open-source security tools with the AWS code services

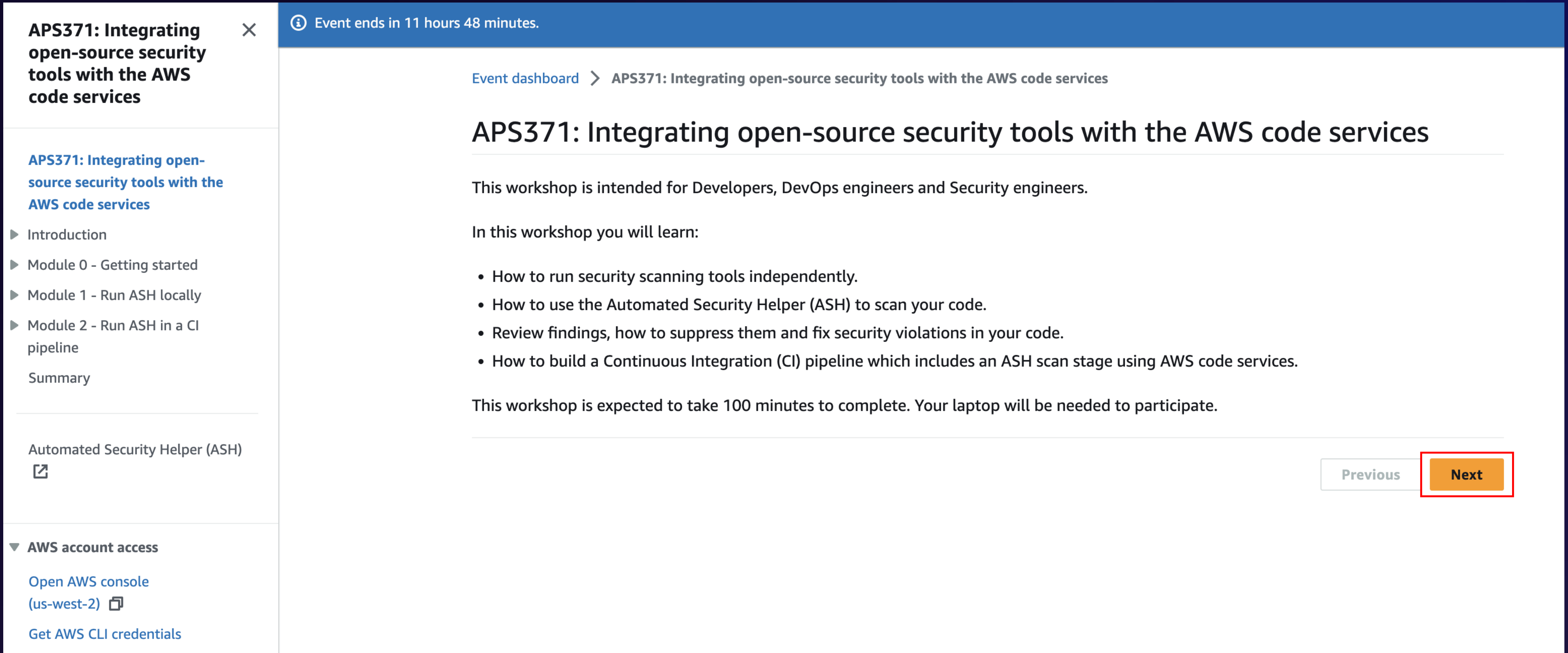
This workshop is intended for Developers, DevOps engineers and Security engineers.

In this workshop you will learn:

- How to run security scanning tools independently.
- How to use the Automated Security Helper (ASH) to scan your code.
- Review findings, how to suppress them and fix security violations in your code.
- How to build a Continuous Integration (CI) pipeline which includes an ASH scan stage using AWS code services.

This workshop is expected to take 100 minutes to complete. Your laptop will be needed to participate.

Step 4: Get started with the workshop



APS371: Integrating open-source security tools with the AWS code services
APS371: Integrating open-source security tools with the AWS code services
Introduction
Module 0 - Getting started
Module 1 - Run ASH locally
Module 2 - Run ASH in a CI pipeline
Summary
Automated Security Helper (ASH)
AWS account access
Open AWS console (us-west-2)
Get AWS CLI credentials
Event ends in 11 hours 48 minutes.
Event dashboard
APS371: Integrating open-source security tools with the AWS code services
Previous
Next

APS371: Integrating open-source security tools with the AWS code services

This workshop is intended for Developers, DevOps engineers and Security engineers.

In this workshop you will learn:

- How to run security scanning tools independently.
- How to use the Automated Security Helper (ASH) to scan your code.
- Review findings, how to suppress them and fix security violations in your code.
- How to build a Continuous Integration (CI) pipeline which includes an ASH scan stage using AWS code services.

This workshop is expected to take 100 minutes to complete. Your laptop will be needed to participate.

Resources to get you started

ASH binary
GitHub repository

Workshop Studio link

Thank you!

Please complete the session survey in the mobile app

Daniel Begimher

Senior Security Architect
Global Services Security
AWS

Pat Gaw

Principal Security Consultant
Global Services Security
AWS