
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

SEC307

Ship securely: Automated security testing for developers

Alvin Delagon (he/him)

Developer Specialist Senior Solutions Architect, DevAx

Amazon Web Services

Anitha Deenadayalan (she/her)

Developer Specialist Senior Solutions Architect, DevAx

Amazon Web Services

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Shift left – security overview

Automated security testing

Additional security tools and processes

Cultural aspects

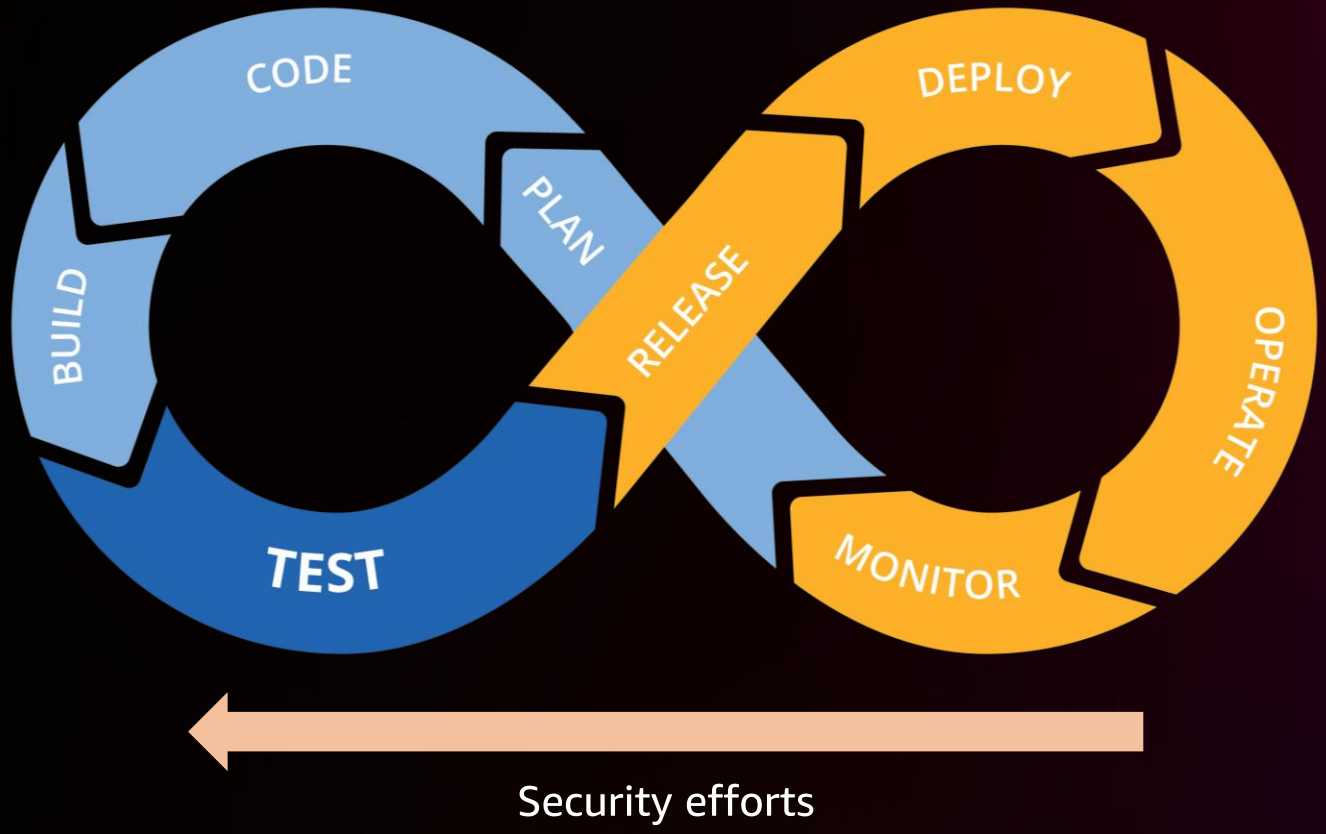
Workshop

Takeaways

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

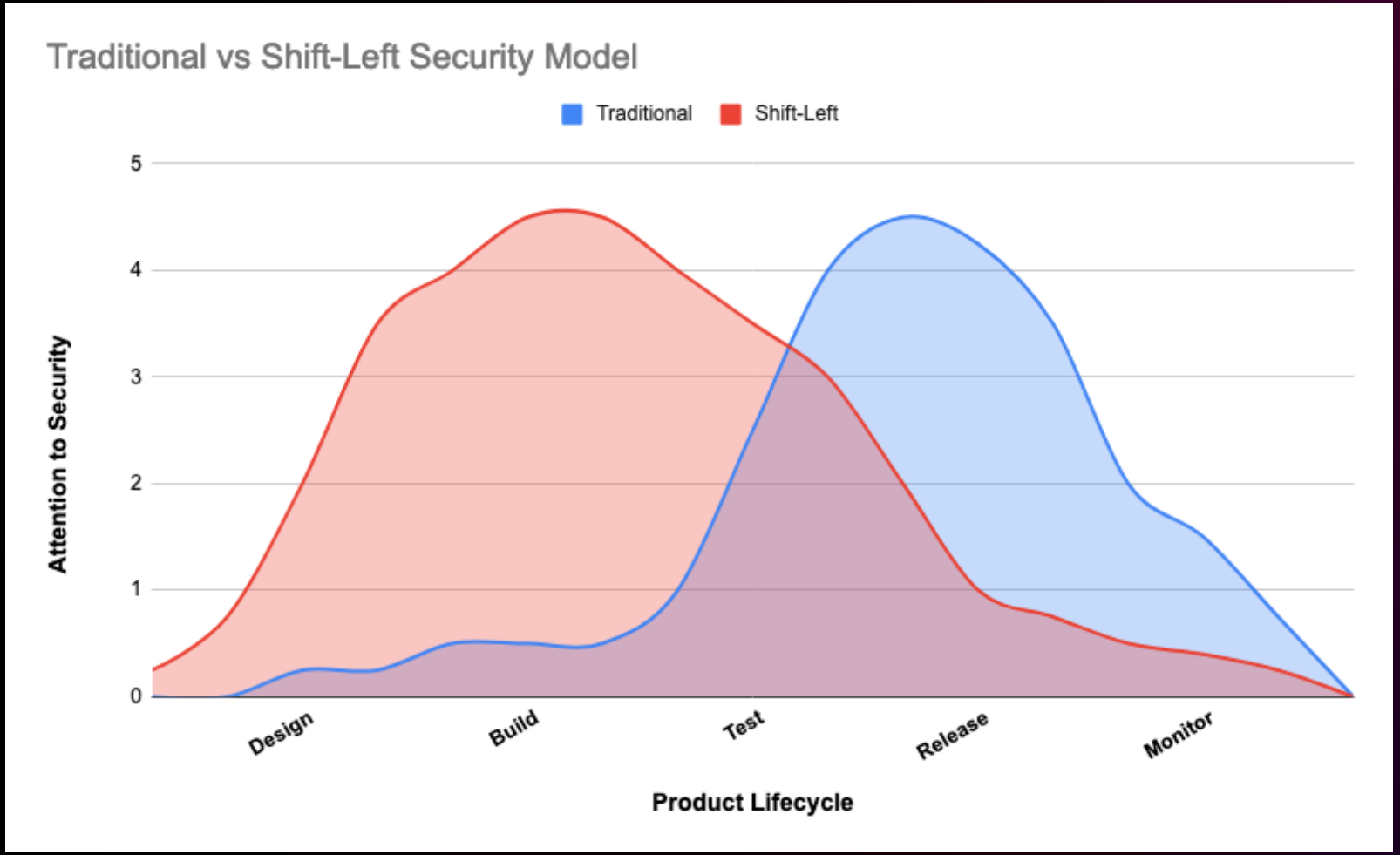
Shift left: Security

Move attention to security to the **earliest** possible point in the software development lifecycle



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Attention to security



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Does this mean more work for developers?

Investment in order to sleep better

Cheaper to fix security issues earlier[1]

Faster time to market, less turnaround time

Most of it can be automated

[1] Error Cost Escalation through the Project Lifecycle – https://ntrs.nasa.gov/citations/20100036670

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Automated security testing

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Are we writing code securely?

STATIC APPLICATION SECURITY TESTING (SAST)

SAST tools catch bugs and vulnerabilities in your application, with thousands of automated static code analysis rules

Amazon CodeGuru

sonarqube

Checkmarx

snyk

aws

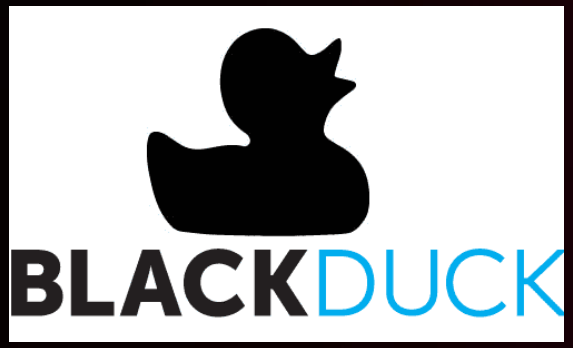
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How about code written by others?

SOFTWARE COMPOSITION ANALYSIS (SCA)

Most modern applications are composed of code written by others[2]

SCA tools check your dependencies for any vulnerabilities using a known database



[2] Veracode State of Security Report – https://www.veracode.com/state-of-software-security-report

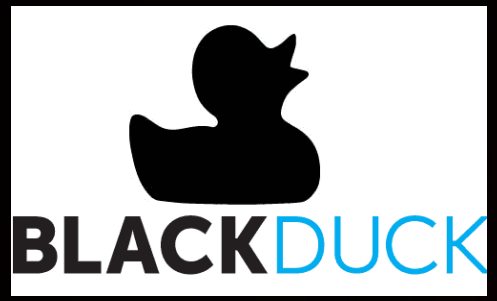
aws

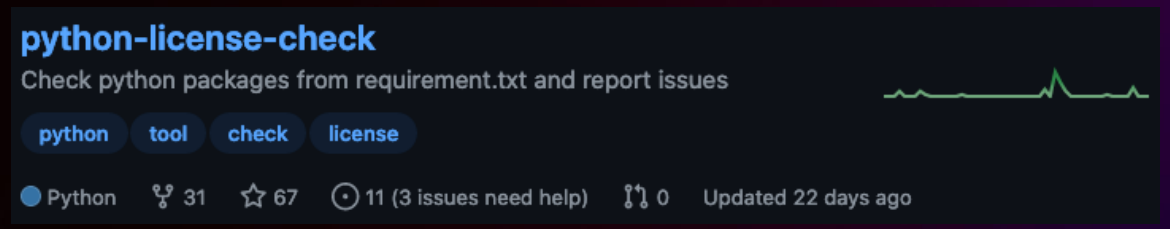
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Are we protecting our intellectual property?

LICENSE CHECKING

Copyleft licenses such as **GNU General Public License** require you to distribute your application under the same license





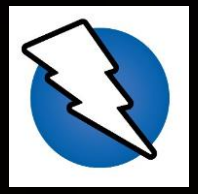
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Are our security controls working?

DYNAMIC APPLICATION SECURITY TESTING (DAST)

DAST tests the security posture of your application during **runtime**

This is used in tandem with other AppSec tools in order to validate your security controls

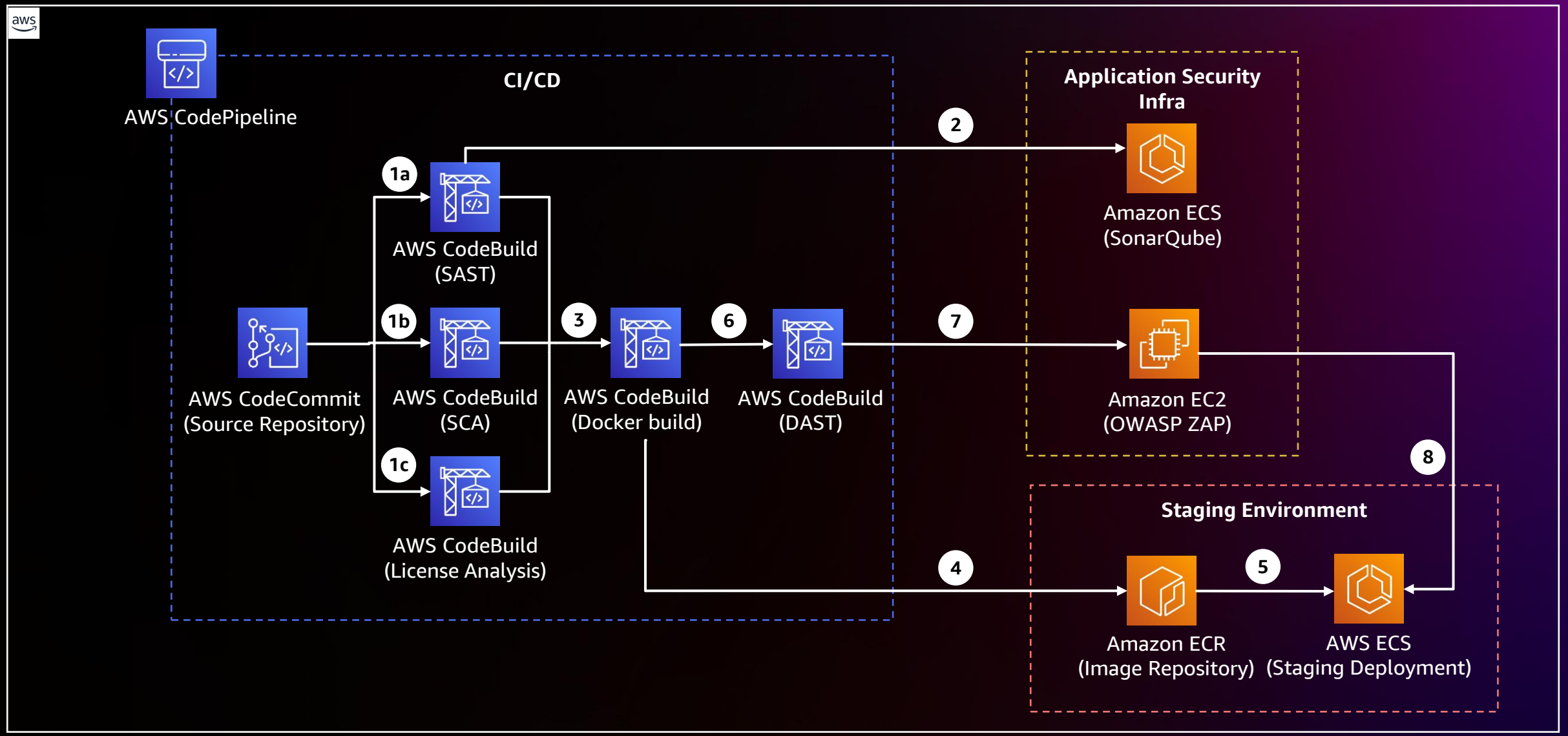


OWASP ZAP

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Automated security testing



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Additional security tools & processes

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Security before writing code

THREAT MODELING

Helps to identify **security requirements** prior to development[3,4]

Threat ID	Asset	Entry point	Attack summary	Technique
TR-01	User account	Login page	Being able to hijack someone's account and conduct malicious activities	Brute-force attacks, password dumps, session hijacking, phishing

Threat ID	Preventive/Corrective controls
TR-01	• Multi-factor authentication • Password lifecycle and rules • Secure generation of session IDs • Session lifecycle • Secure storage of session cookies • Regular security reminders to customers

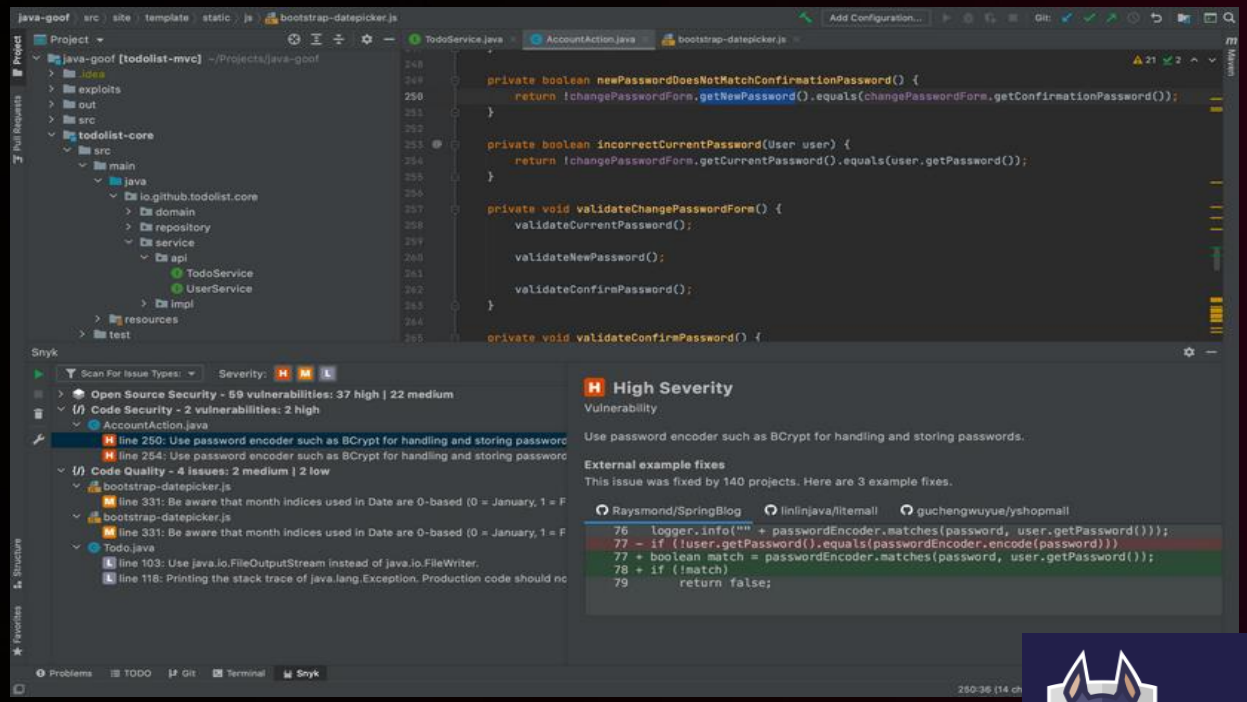
[3] How to approach threat modeling – https://aws.amazon.com/blogs/security/how-to-approach-threat-modeling/
[4] Threat modeling the right way for builders – https://catalog.workshops.aws/threatmodel

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Security while writing code

IDE SECURITY EXTENSIONS

IDE extensions can provide quick and actionable security findings as you write your code



snyk

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Security in source control

GIT HOOKS

Using Git hooks (pre-commit) allows you to trigger security scans automatically on your codebase to prevent accidental leakage of sensitive info such as AWS credentials and SSH keys

Talisman

GitHound

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Security in configuration

SECRETS MANAGEMENT

Sensitive configurations such as database credentials are better stored somewhere else and accessed securely

AWS Secrets Manager

AWS Systems Manager

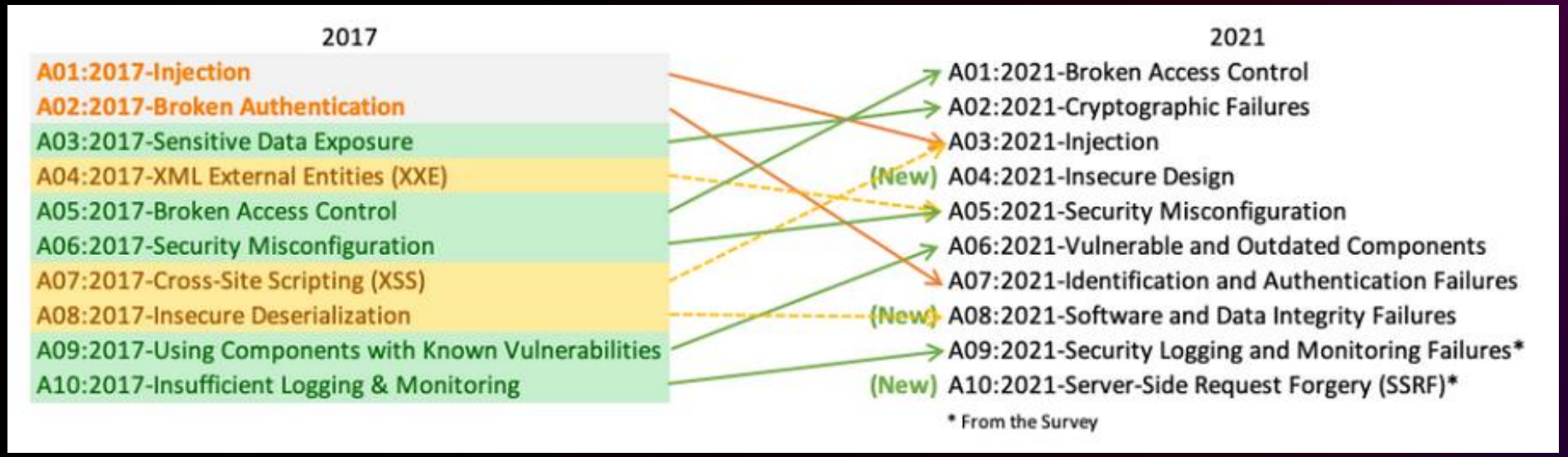
Parameter store

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Be a better code reviewer

https://owasp.org/www-project-top-ten/



The OWASP Word Mark and OWASP & Design Logo are registered or unregistered service marks of OWASP Foundation, Inc. in the United States and other countries. All rights reserved. Unauthorized use strictly prohibited.

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Cultural aspects

- Automation alone will not solve the problems
- Share security knowledge across different teams
  - **Build allies** – identify security champions and setup a community of practice
  - Conduct regular security-centric lunch-and-learn sessions
- Focus on building an inclusive culture and avoid the “blame game”

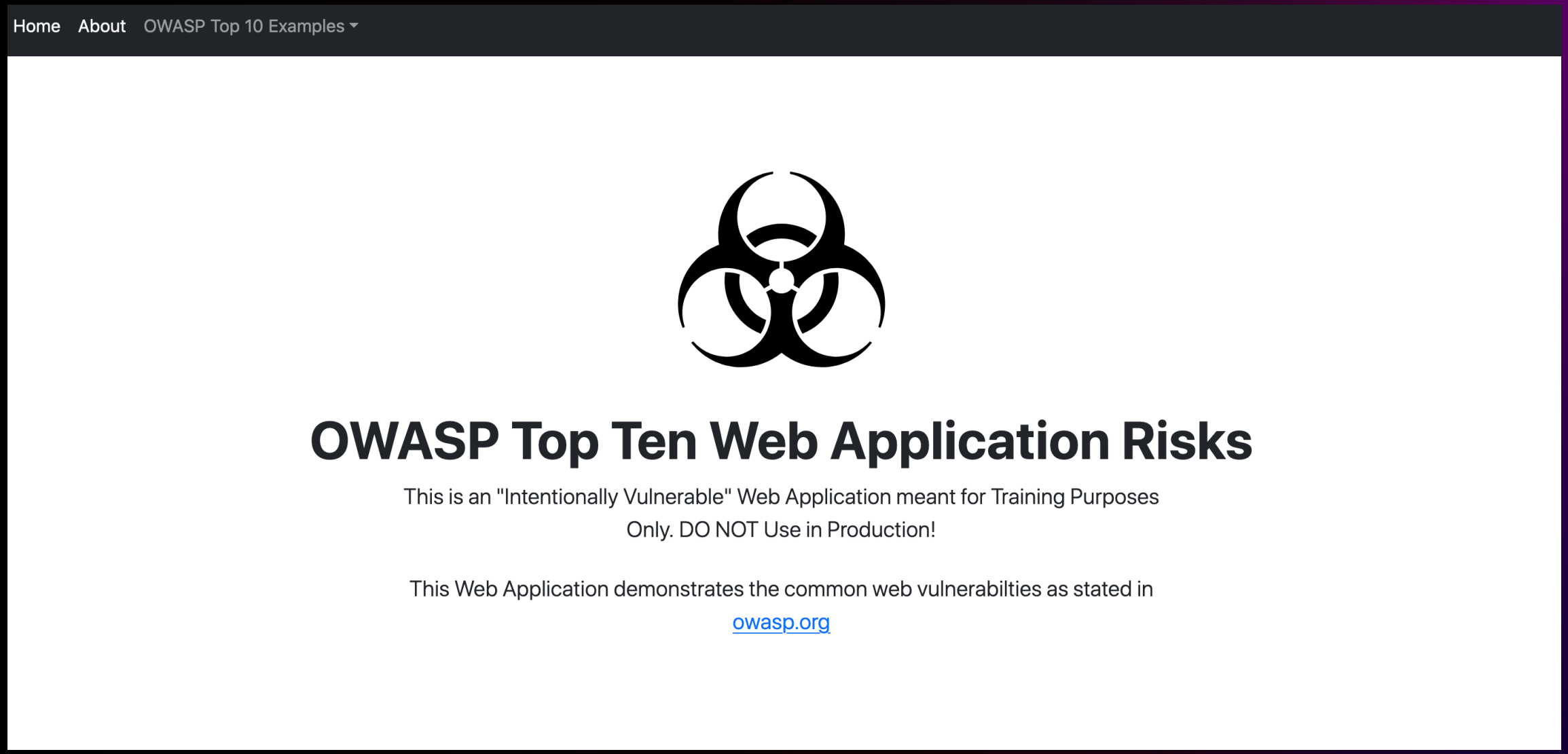
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Let’s start building!

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Exploit and remediate



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

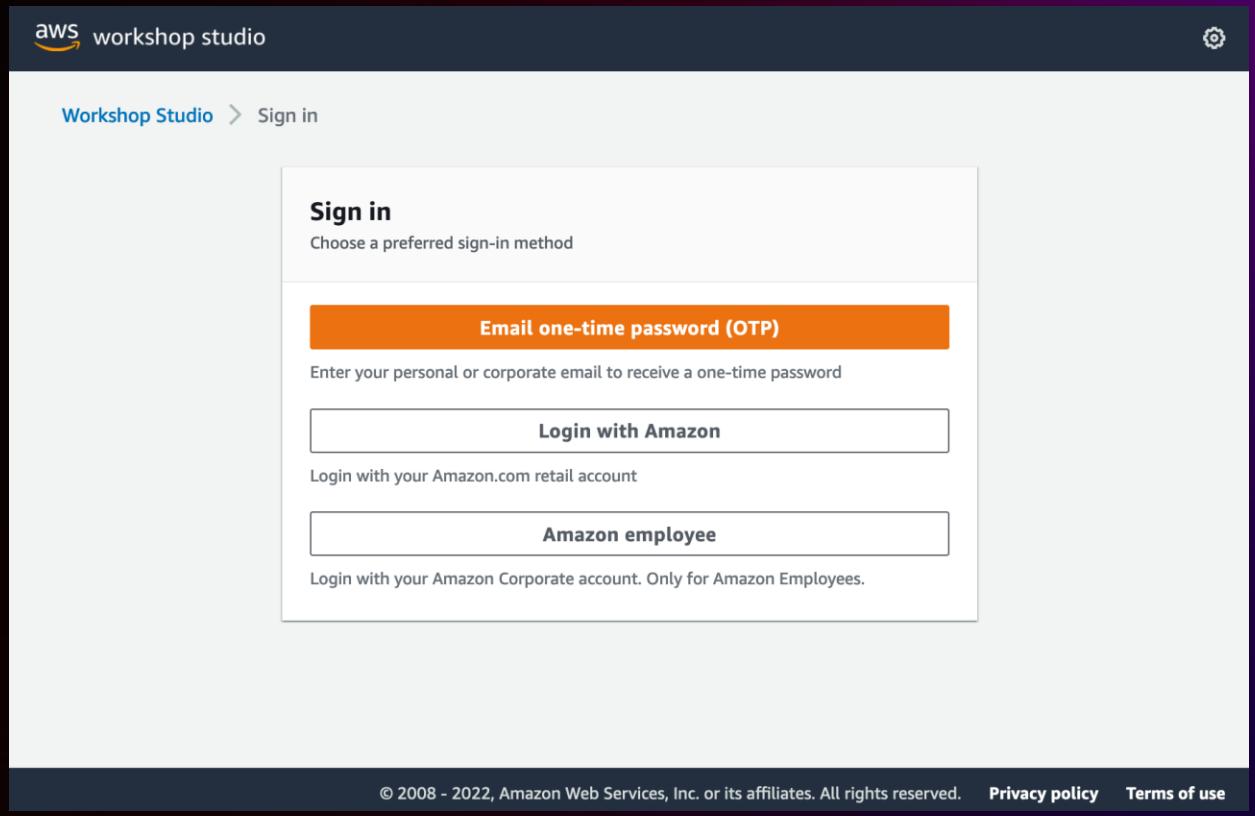
Getting started with this workshop

- As a participant, you will have access to an AWS account with any optional preprovisioned infrastructure and IAM policies needed to complete this workshop
- The AWS account will only be available for the duration of this workshop – you will lose access to the account thereafter
- The optional preprovisioned infrastructure will be deployed to a specific AWS Region. Check your workshop content to determine whether other Regions will be used.
- Be sure to review the terms and conditions of the event. Do not upload any personal or confidential information in the account.

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 1: Sign in via your preferred method

One-click join link: https://s12d.com/sec4devs



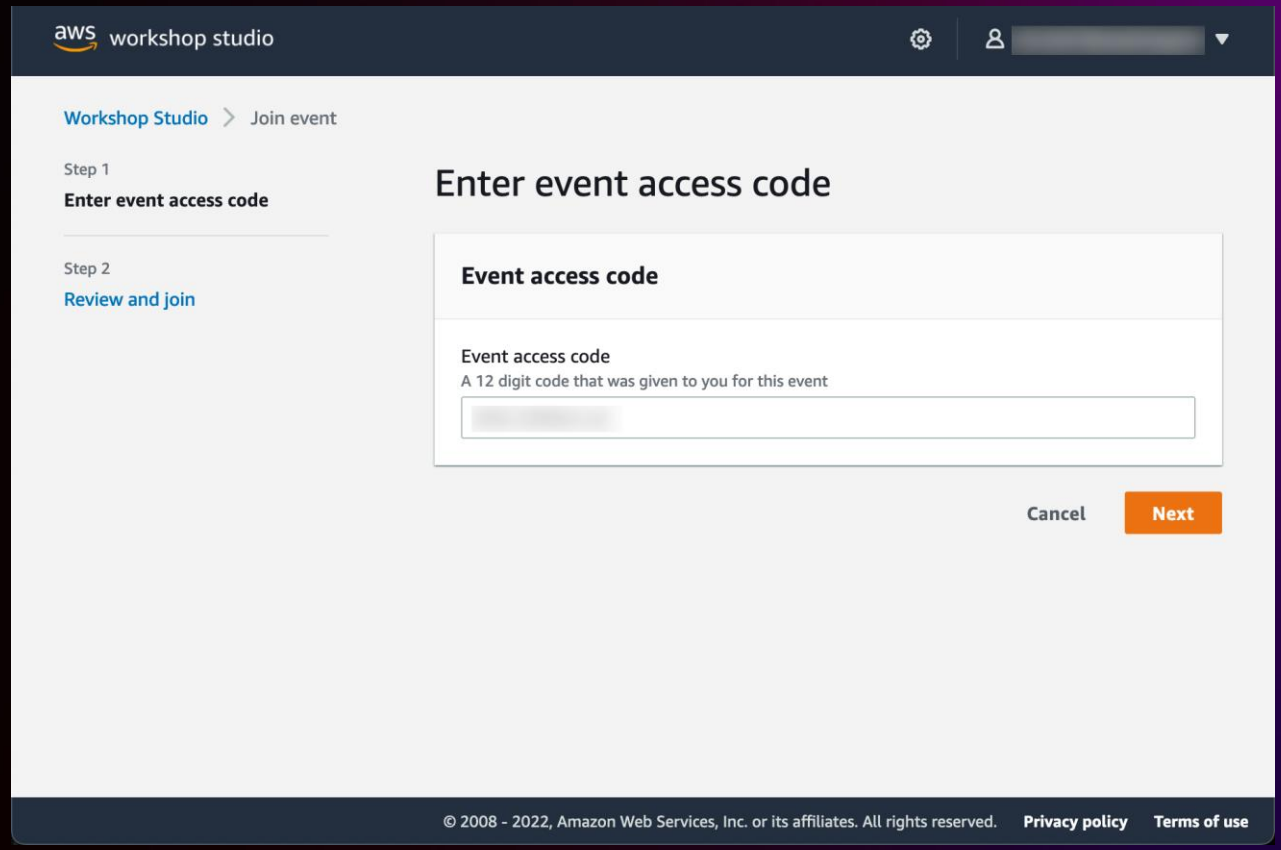
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 2 (optional): Enter event access code

Enter 12-digit event access code:

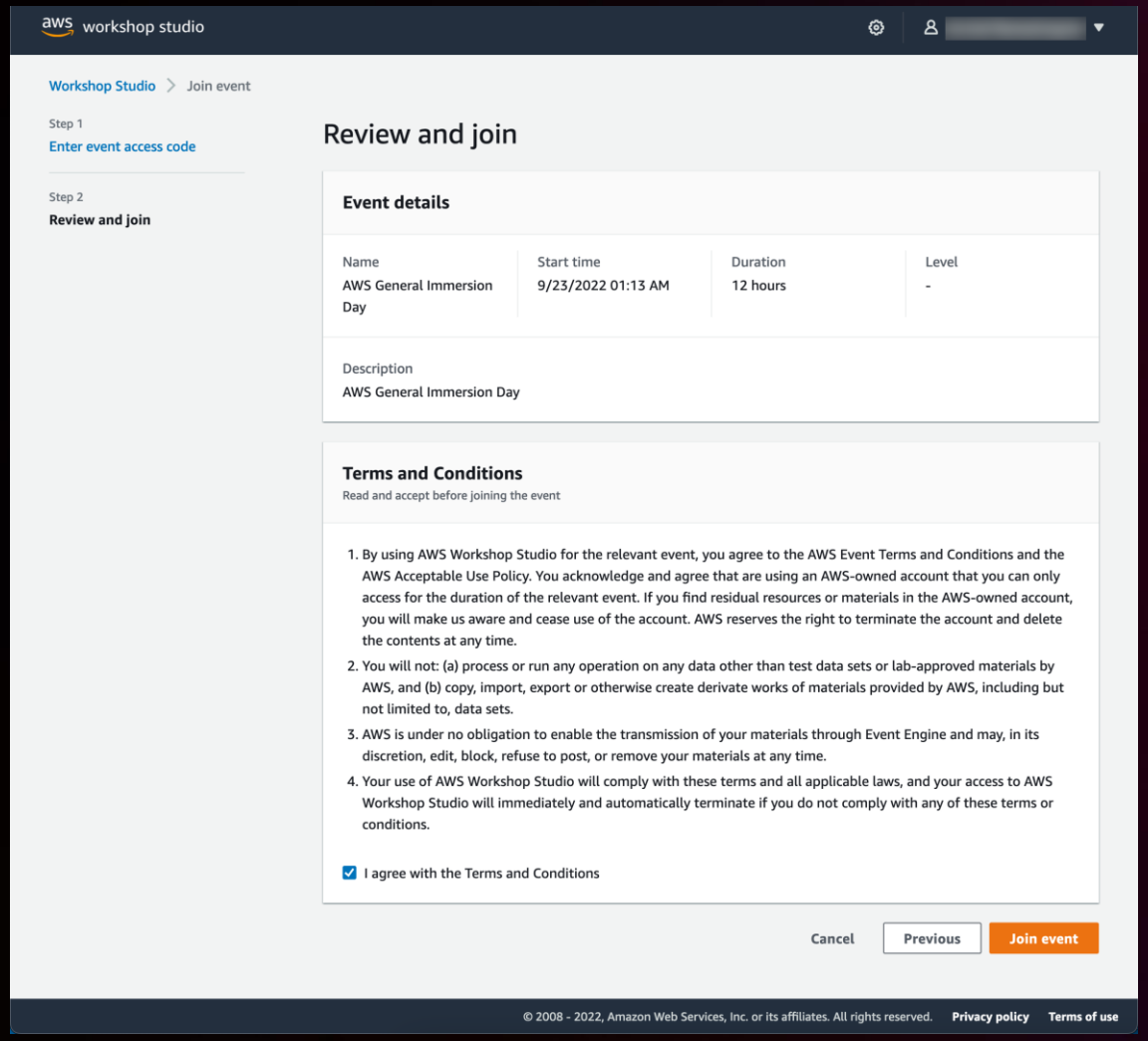
6586-0e0893-16

If you were given a one-click join link, you can skip this step



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

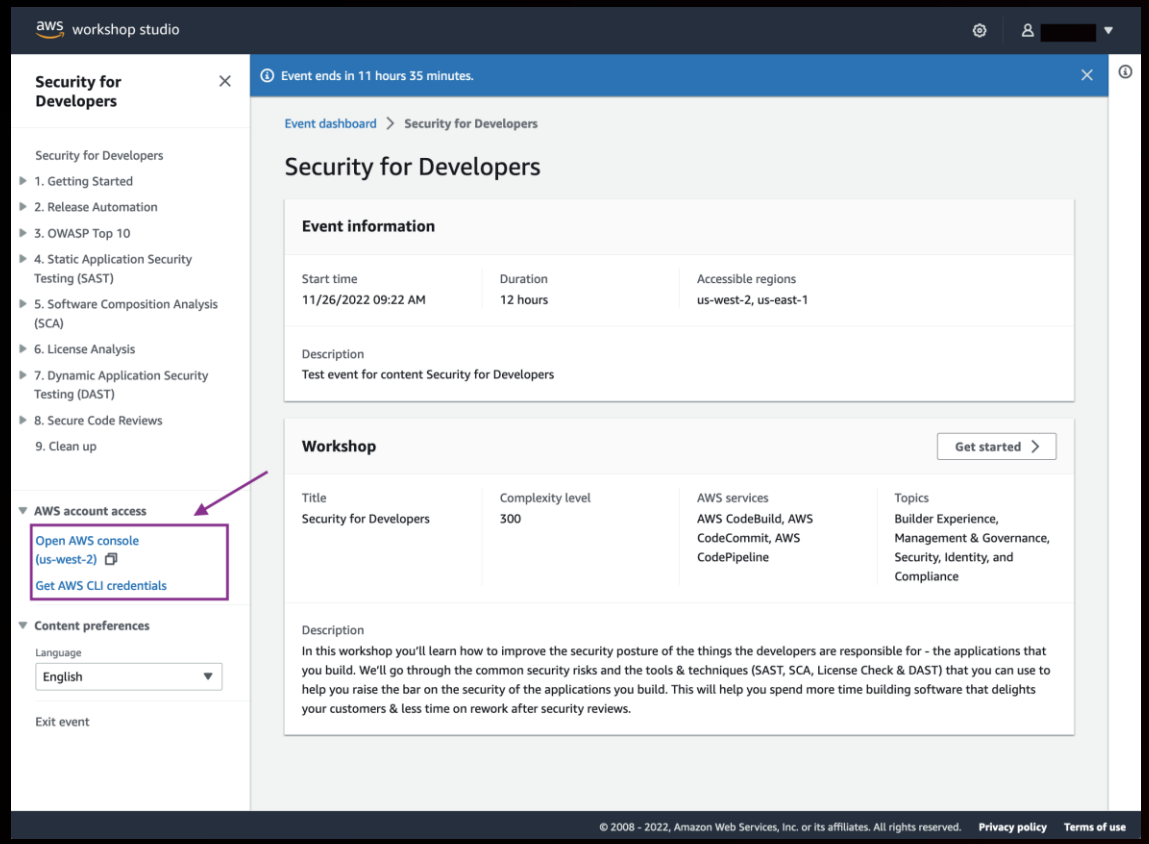
Step 3: Review terms and join event

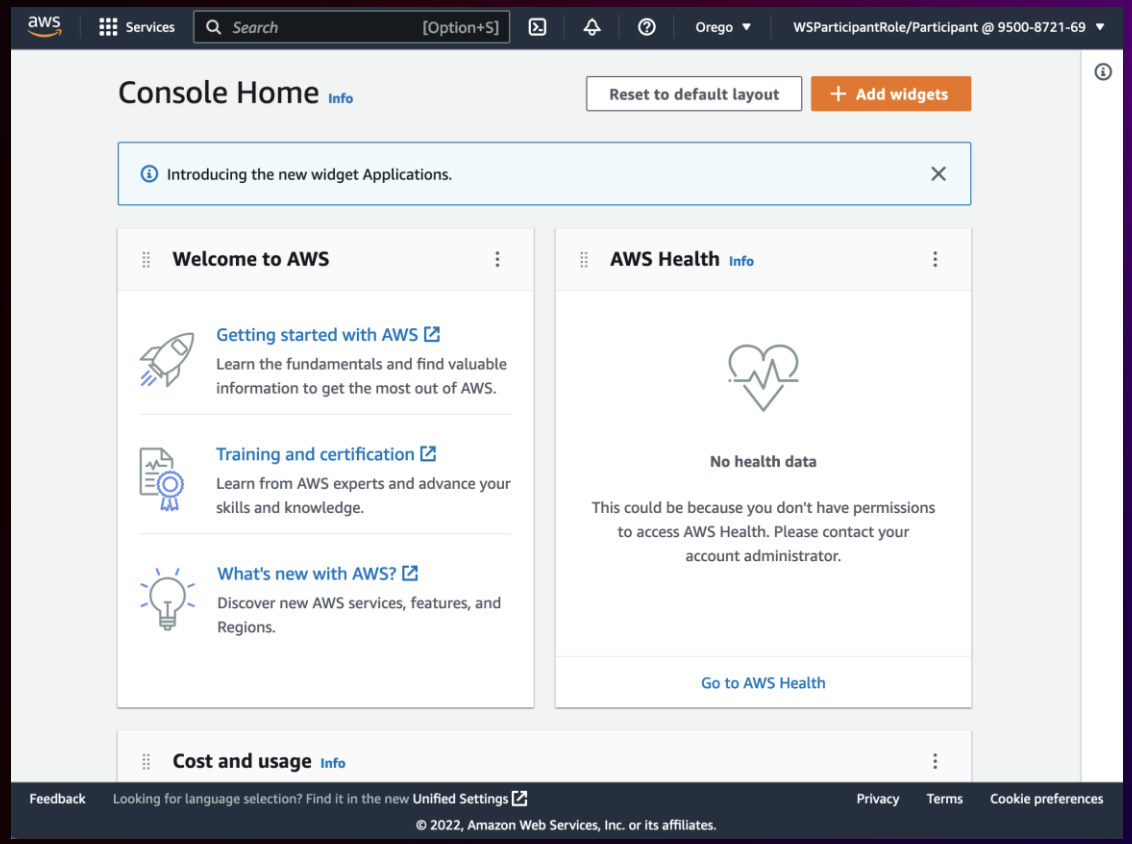


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 4: Access AWS account

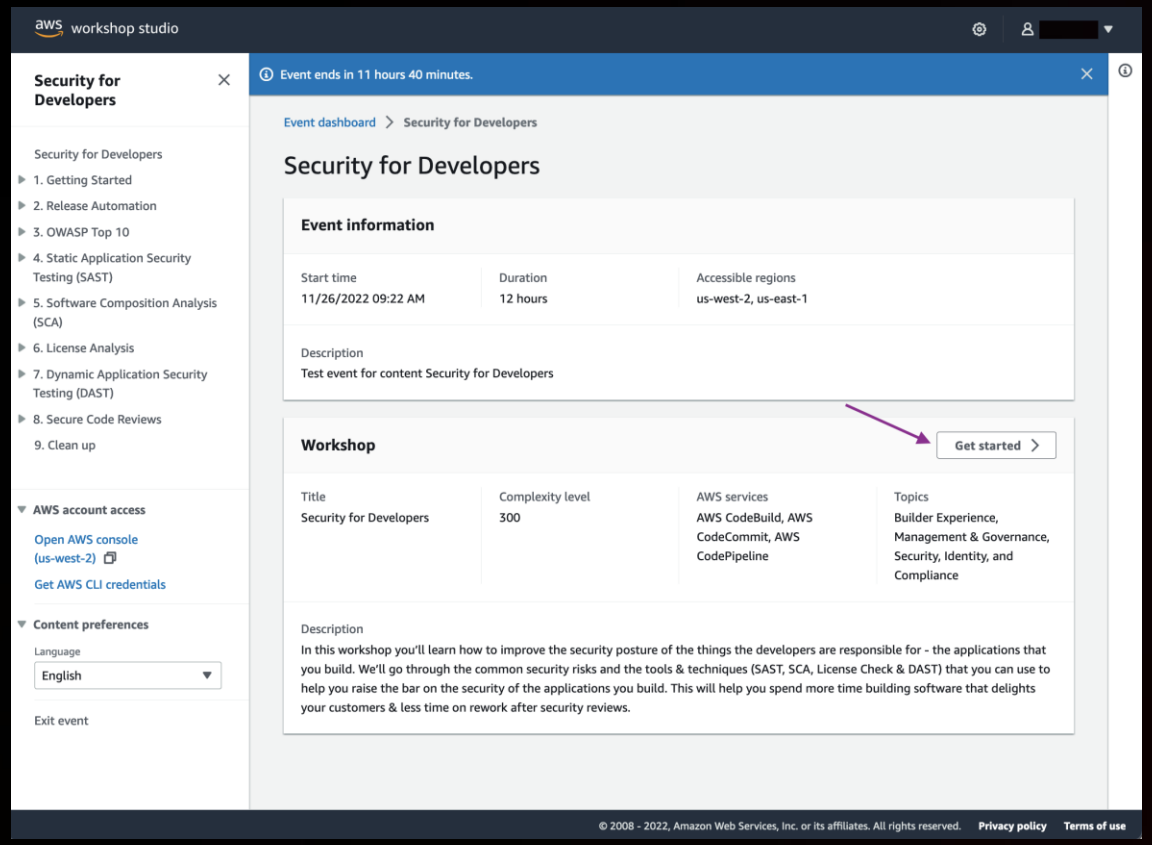
Access the AWS Management Console or generate AWS CLI credentials as needed

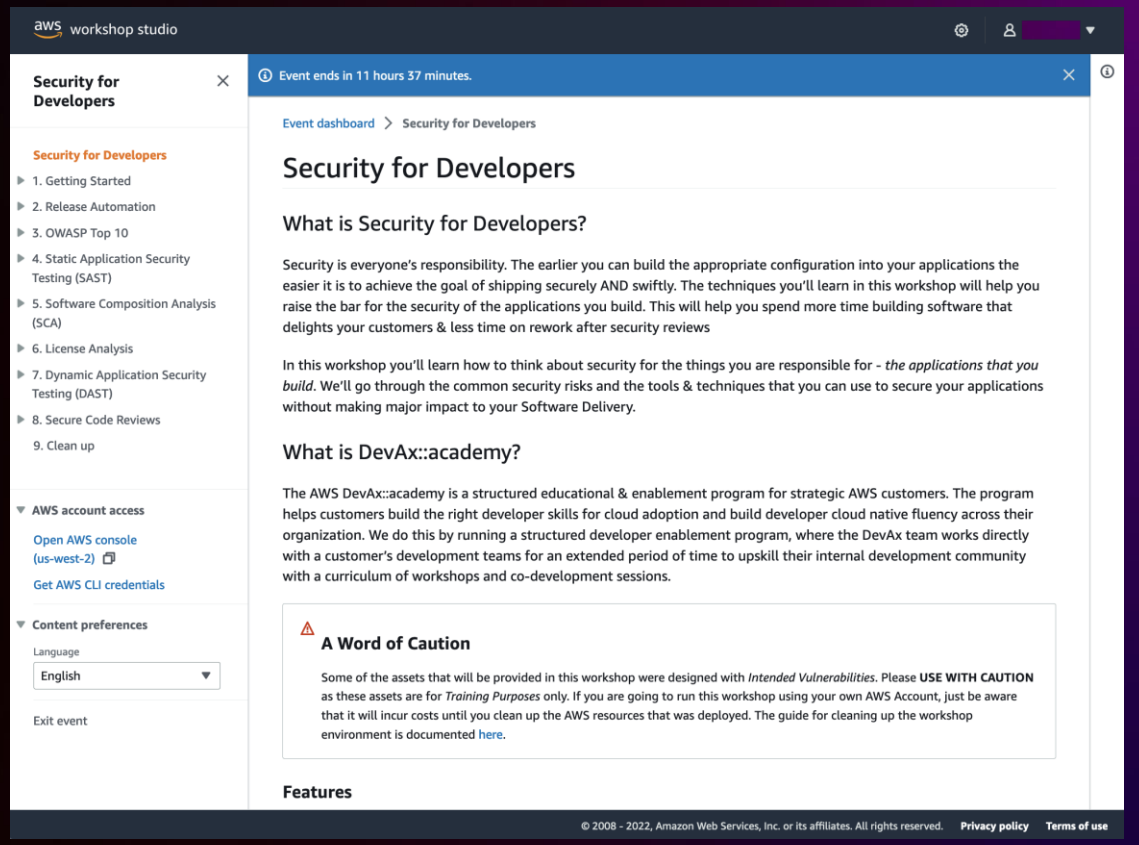




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 5: Get started with the workshop





© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Key takeaways

- Security **is everyone's responsibility**
- Embrace security as an **integral part of the process**, use **feedback** to refine the process
- Security must be **empowering**, not blocking
- DevSecOps **is not** one size fits all

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Alvin Delagon
adelagon@amazon.com

adelagon (GitHub)
adelagon

Anitha Deenadayalan
anithade@amazon.com

awsanitha (GitHub)
anitha-deenadayalan

Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.