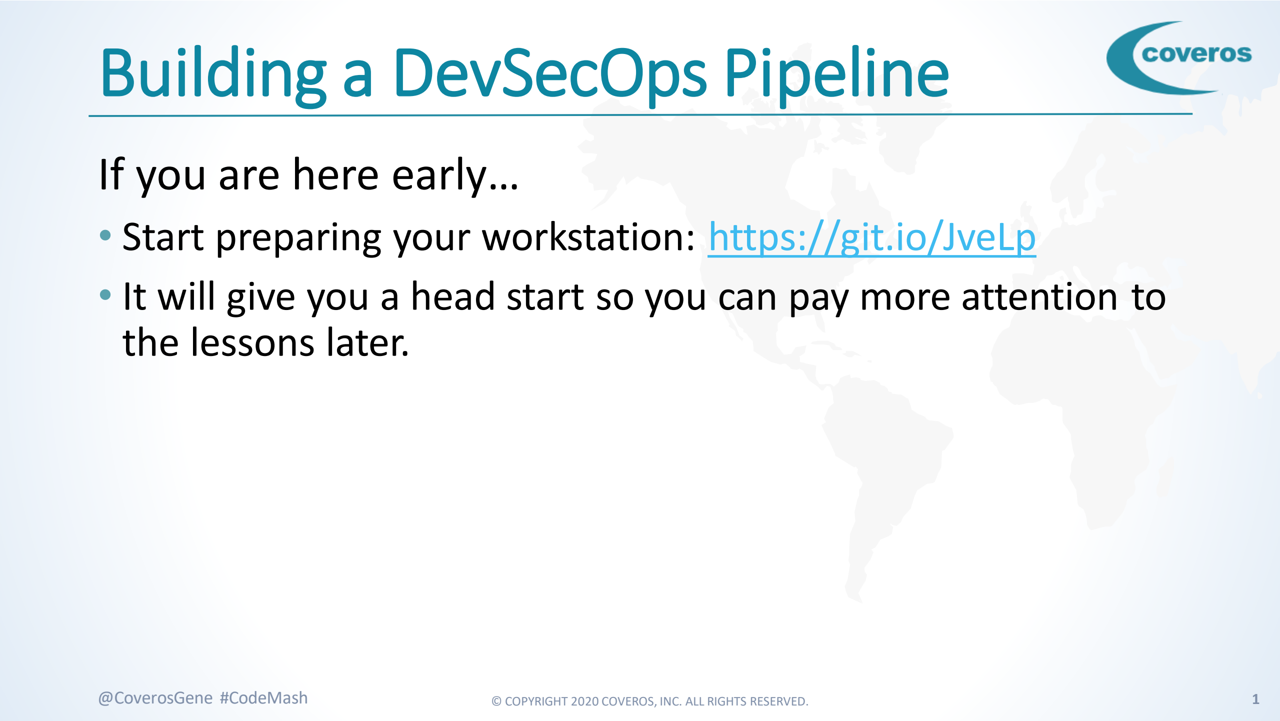


Building a DevSecOps Pipeline

If you are here early…

- Start preparing your workstation: https://git.io/JveLp
- It will give you a head start so you can pay more attention to the lessons later.

@CoverosGene #CodeMash

1

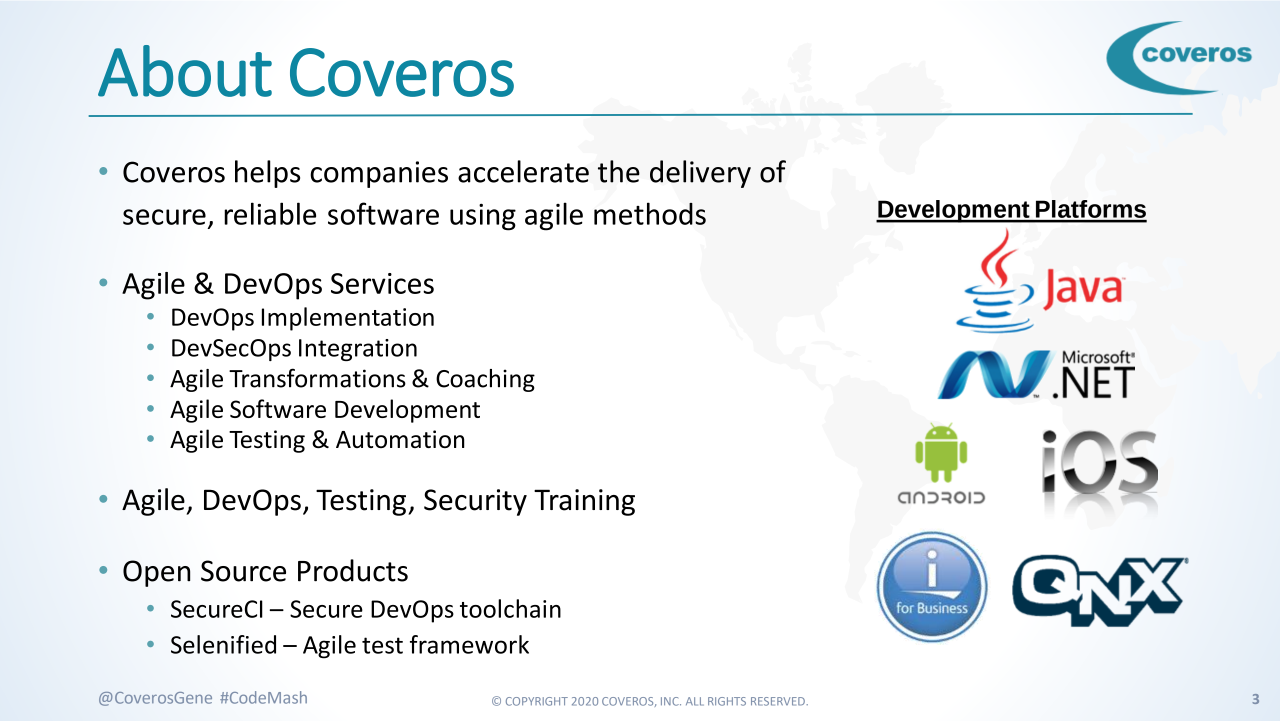
1

Building a DevSecOps Pipeline

Gene Gotimer
Coveros, Inc.

@CoverosGene #CodeMash

2



About Coveros

coveros

- Coveros helps companies accelerate the delivery of secure, reliable software using agile methods
- Agile & DevOps Services
  - DevOps Implementation
  - DevSecOps Integration
  - Agile Transformations & Coaching
  - Agile Software Development
  - Agile Testing & Automation
- Agile, DevOps, Testing, Security Training
- Open Source Products
  - SecureCI – Secure DevOps toolchain
  - Selenified – Agile test framework

Development Platforms

Java

Microsoft .NET

android

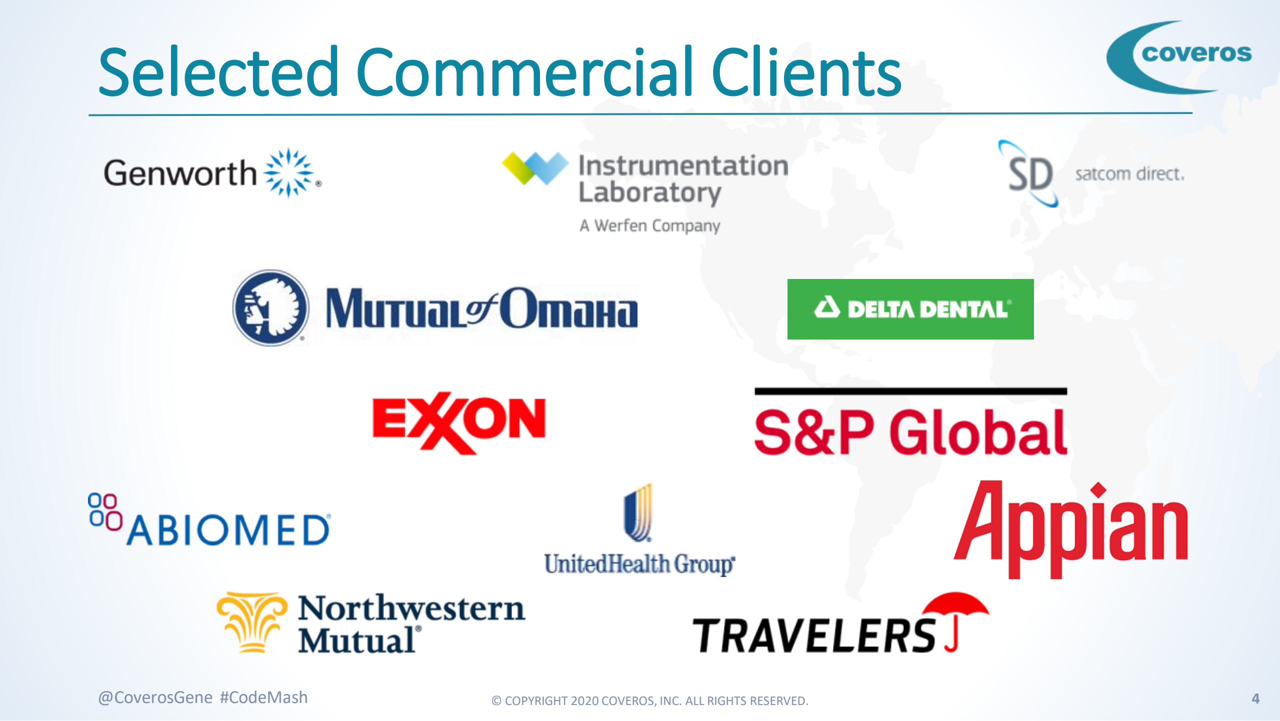
iOS

i for Business

QNX

@CoverosGene #CodeMash

3



Selected Commercial Clients

coveros

Genworth

Instrumentation Laboratory
A Werfen Company

SD satcom direct.

Mutual of Omaha

DELTA DENTAL

EXXON

S&P Global

ABIOMED

UnitedHealth Group

Appian

Northwestern Mutual

TRAVELERS

@CoverosGene #CodeMash

4

Selected Federal Clients

coveros

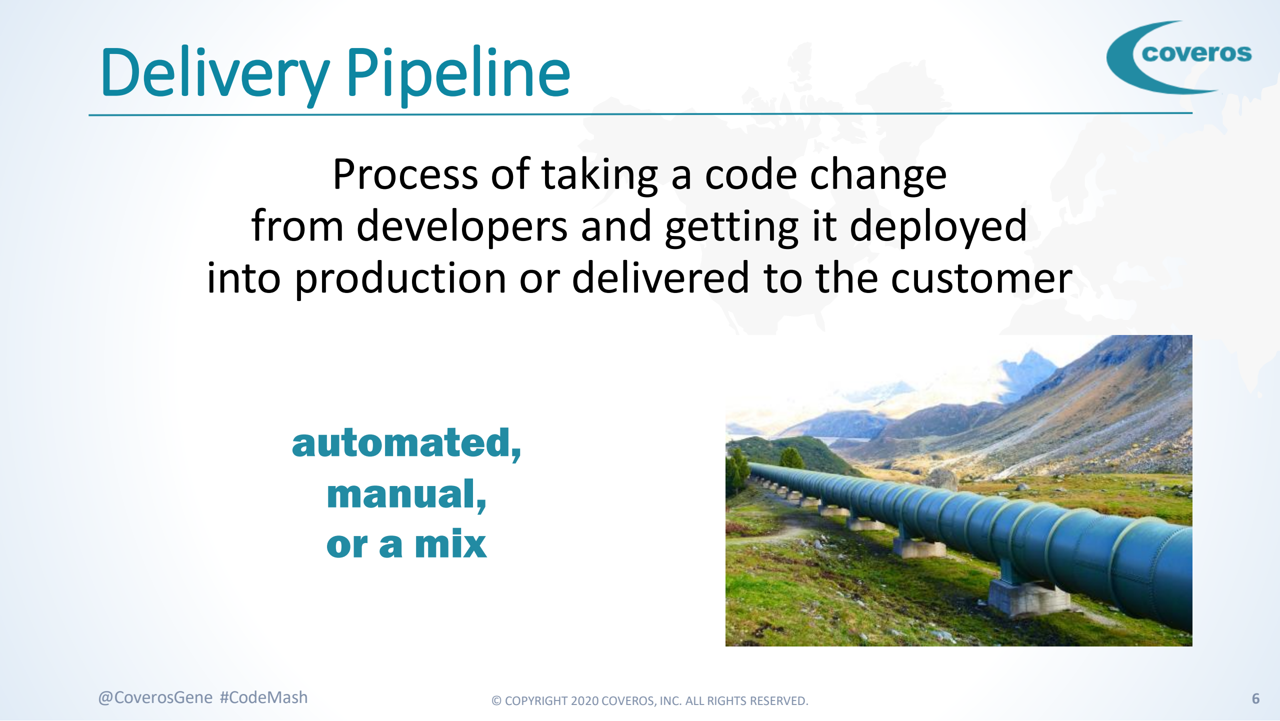
Transportation Security Administration

DISA

U.S. Immigration and Customs Enforcement

CMS CENTERS FOR MEDICARE & MEDICAID SERVICES

U.S. Citizenship and Immigration Services

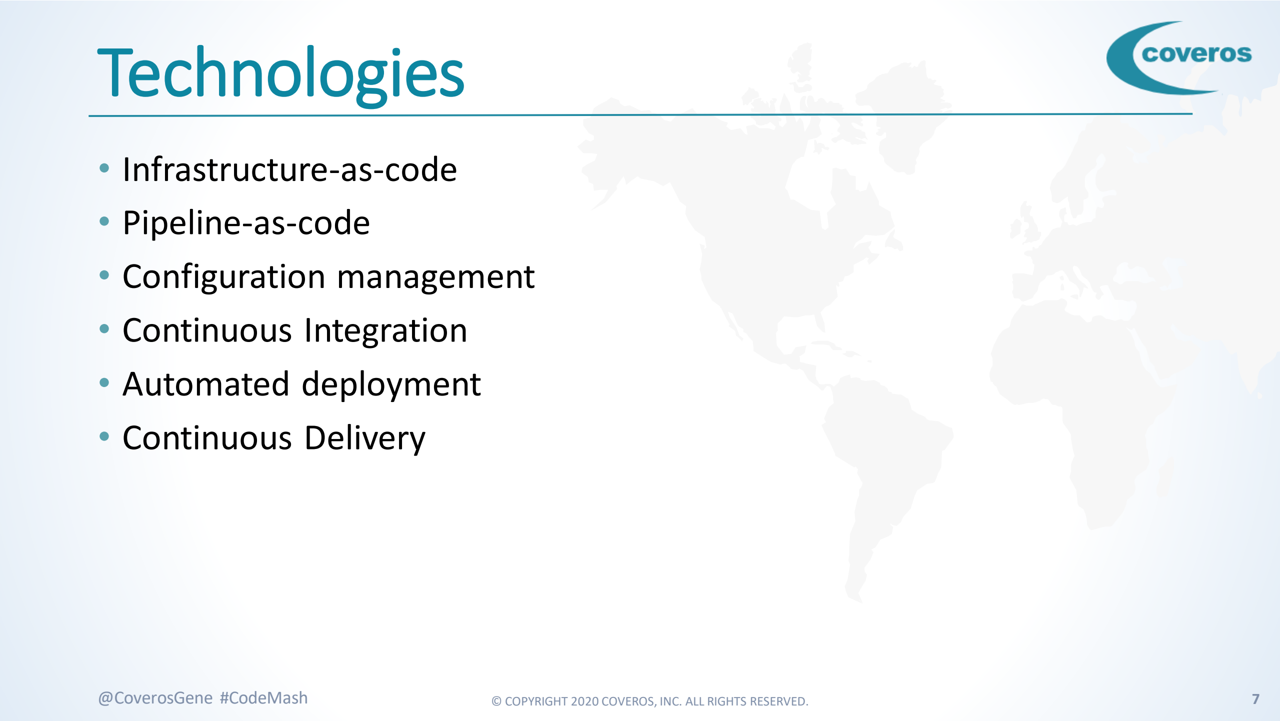


Delivery Pipeline

coveros

Process of taking a code change
from developers and getting it deployed
into production or delivered to the customer

automated,
manual,
or a mix



Technologies

- Infrastructure-as-code
- Pipeline-as-code
- Configuration management
- Continuous Integration
- Automated deployment
- Continuous Delivery

@CoverosGene #CodeMash

7

Tools

coveros
aws
CHEF™
Apache
maven
Sonatype
sonarqube

- AWS
- Chef
- Jenkins
- Maven
- Nexus Repo Manager
- SonarQube

@CoverosGene #CodeMash

8

Lesson 0: Prep Workstation

coveros

- Launch workstation in AWS via the AWS web interface
- Ubuntu Linux with a few apps pre-installed
  - Java, Maven, AWS CLI
- ***Why?***
  - Doing it manually to remind us of the number of steps.
  - Chicken-and-egg problem for automation later.
  - We want to work on AWS's network, not conference Wi-Fi.
  - Linux Ruby is **much** faster than Windows Ruby.
- This is the hardest step of the workshop, because it is manual!

https://git.io/JveLp

Delivery Pipeline

coveros

Process of taking a code change
from developers and getting it deployed
into production or delivered to the customer

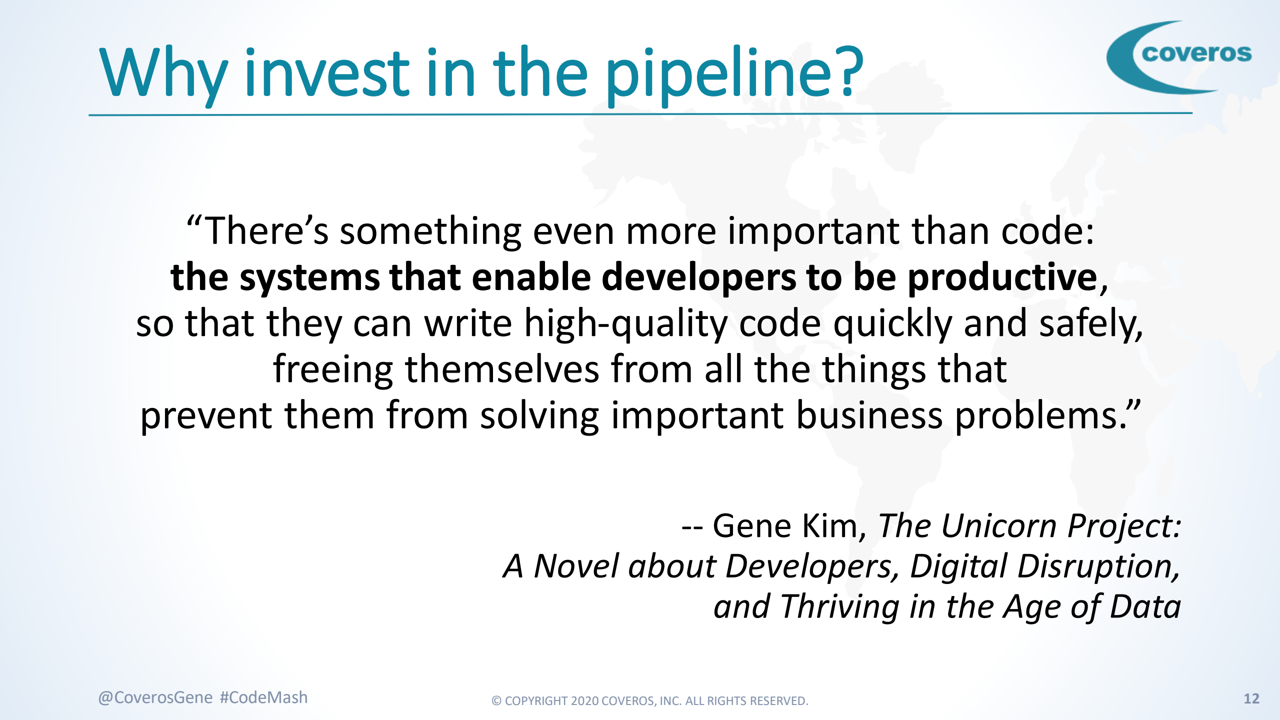
The pipeline is not the goal.

Do we have a viable candidate for production?

coveros
YOU SCHEDULED THE END OF THE TEST PHASE AFTER THE START OF THE PRODUCTION PHASE.
WE'RE FEELING CONFIDENT.
IT'S TOO BAD THAT BEING SMART DOESN'T COME WITH SOME SORT OF GOOD FEELING LIKE THAT.
Dilbert.com DilbertCartoonist@gmail.com
5-17-11 ©2011 Scott Adams, Inc. Dist. by Universal Uclick

@CoverosGene #CodeMash

11



Why invest in the pipeline?

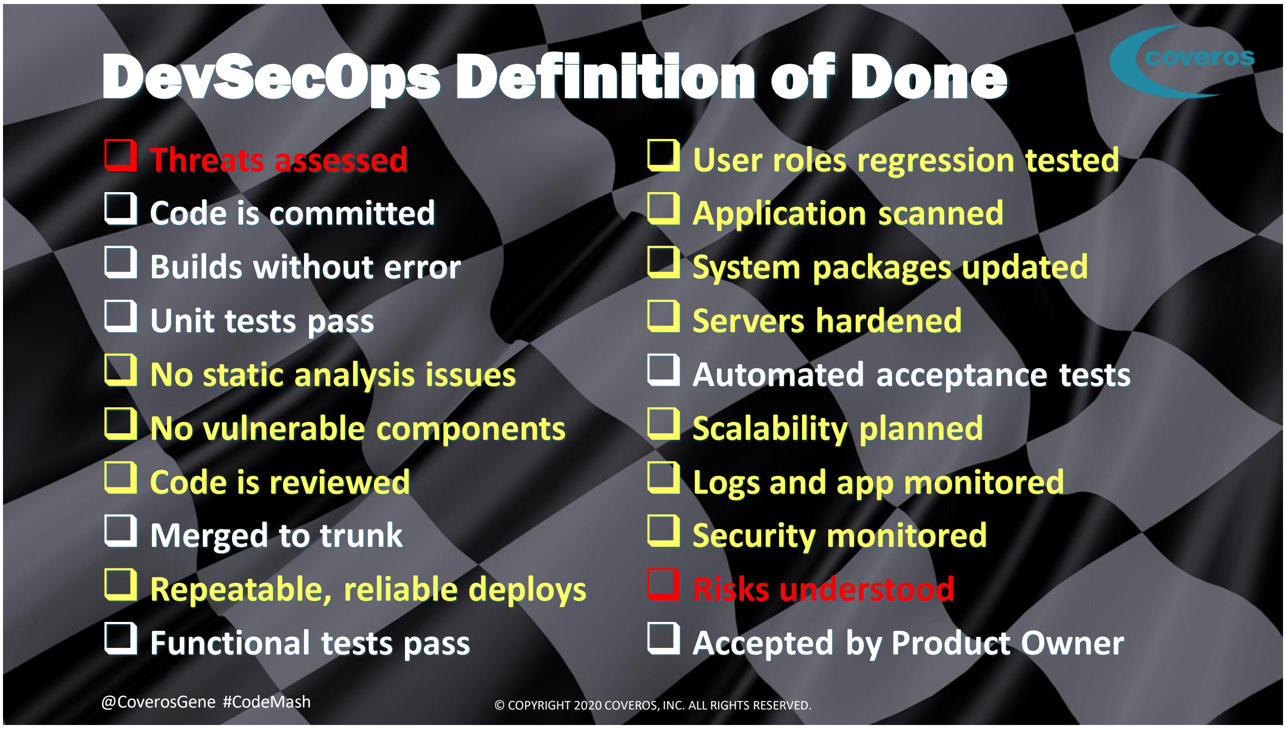
“There’s something even more important than code:
the systems that enable developers to be productive,
so that they can write high-quality code quickly and safely,
freeing themselves from all the things that
prevent them from solving important business problems.”

-- Gene Kim, *The Unicorn Project:*
A Novel about Developers, Digital Disruption,
and Thriving in the Age of Data

@CoverosGene #CodeMash

12

12



DevSecOps Definition of Done

- □ Threats assessed
- □ Code is committed
- □ Builds without error
- □ Unit tests pass
- □ No static analysis issues
- □ No vulnerable components
- □ Code is reviewed
- □ Merged to trunk
- □ Repeatable, reliable deploys
- □ Functional tests pass
- □ User roles regression tested
- □ Application scanned
- □ System packages updated
- □ Servers hardened
- □ Automated acceptance tests
- □ Scalability planned
- □ Logs and app monitored
- □ Security monitored
- □ Risks understood
- □ Accepted by Product Owner

@CoverosGene #CodeMash

13

Assess the threats

STRIDE

Spoofing Identity

Tampering with Data

Repudiation

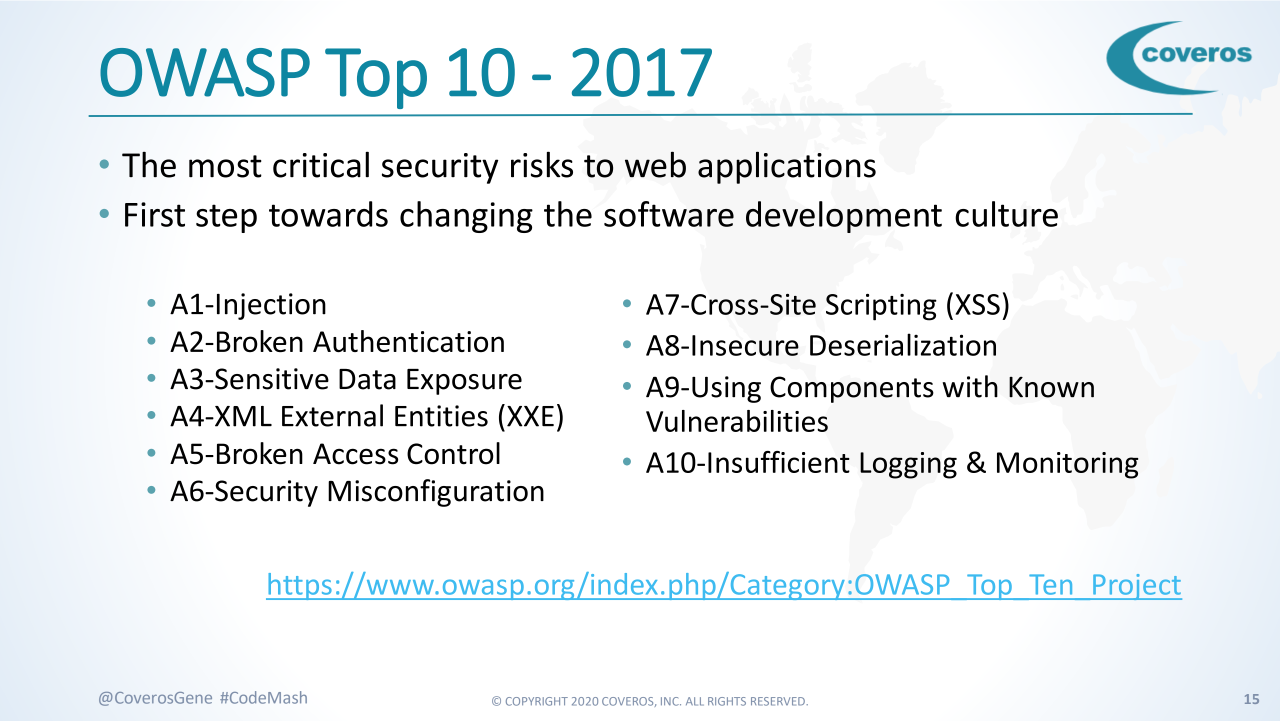
Information Disclosure

Denial of Service

Elevation of Privilege

@CoverosGene #CodeMash

14



OWASP Top 10 - 2017

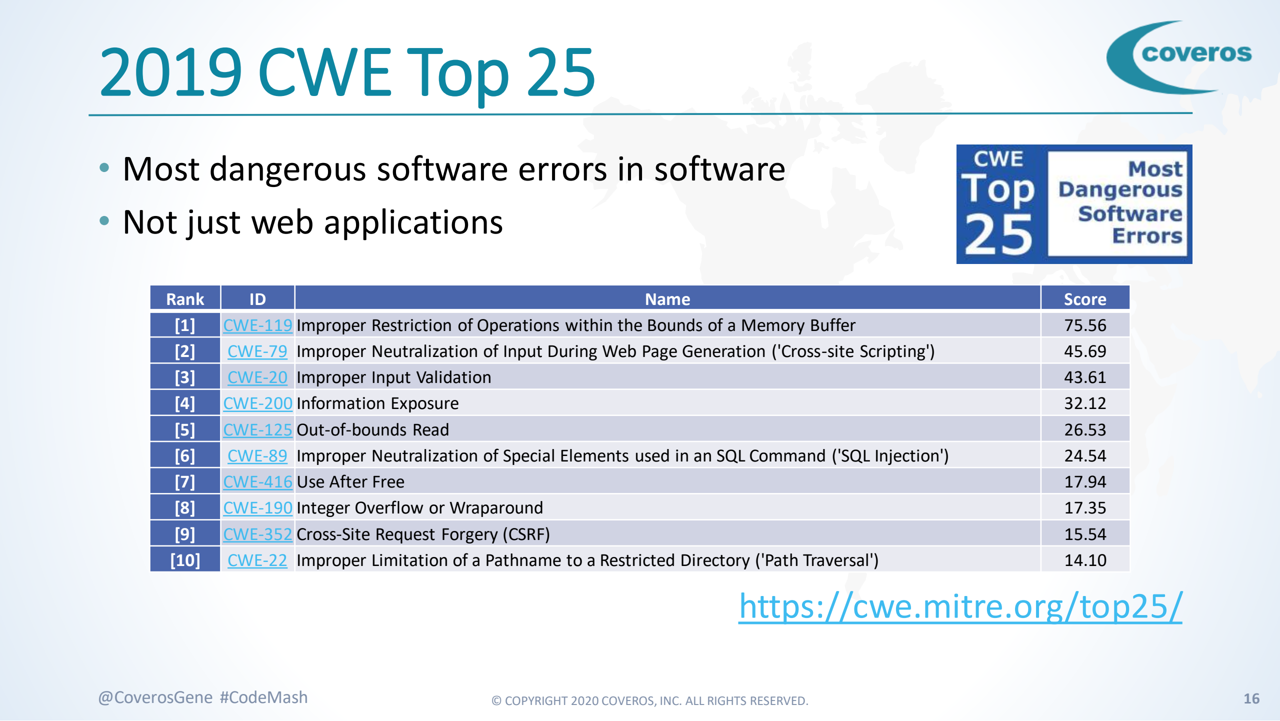
- The most critical security risks to web applications
- First step towards changing the software development culture

  - A1-Injection
  - A2-Broken Authentication
  - A3-Sensitive Data Exposure
  - A4-XML External Entities (XXE)
  - A5-Broken Access Control
  - A6-Security Misconfiguration
  - A7-Cross-Site Scripting (XSS)
  - A8-Insecure Deserialization
  - A9-Using Components with Known Vulnerabilities
  - A10-Insufficient Logging & Monitoring

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

@CoverosGene #CodeMash

15



2019 CWE Top 25

- Most dangerous software errors in software
- Not just web applications

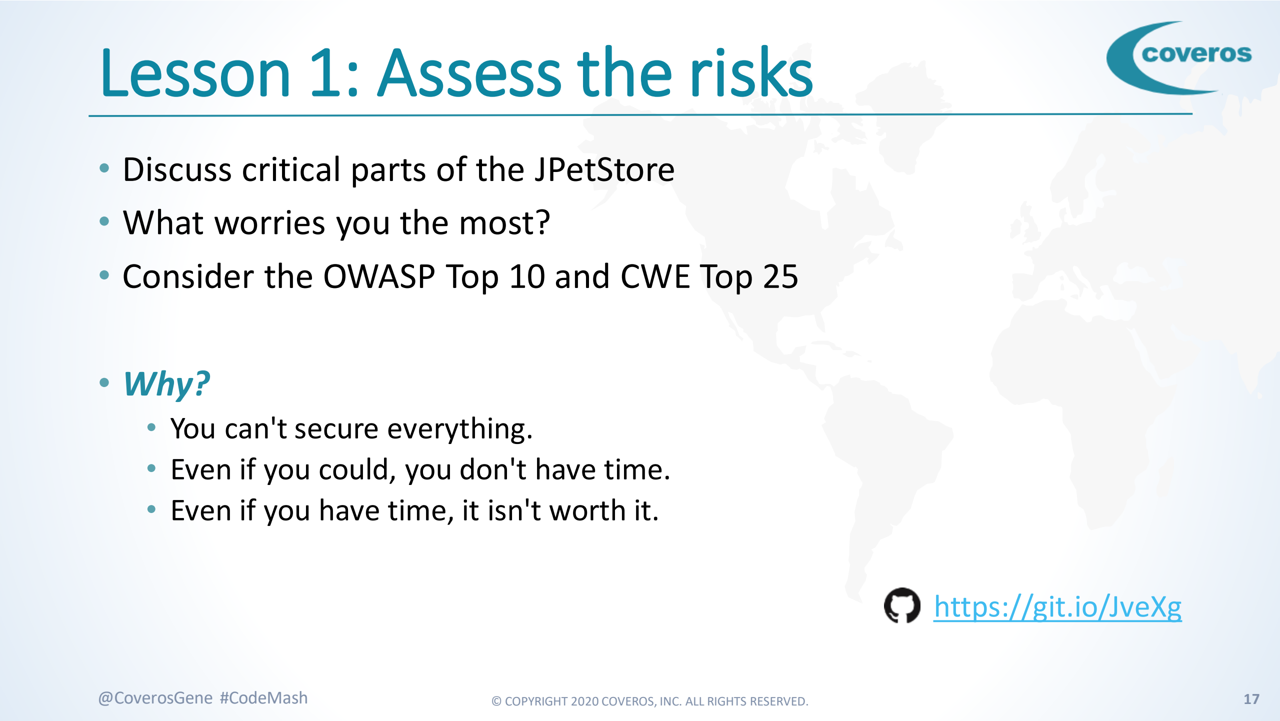
CWE Top 25 Most Dangerous Software Errors

Rank	ID	Name	Score
[1]	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	75.56
[2]	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	45.69
[3]	CWE-20	Improper Input Validation	43.61
[4]	CWE-200	Information Exposure	32.12
[5]	CWE-125	Out-of-bounds Read	26.53
[6]	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	24.54
[7]	CWE-416	Use After Free	17.94
[8]	CWE-190	Integer Overflow or Wraparound	17.35
[9]	CWE-352	Cross-Site Request Forgery (CSRF)	15.54
[10]	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.10

https://cwe.mitre.org/top25/

@CoverosGene #CodeMash

16

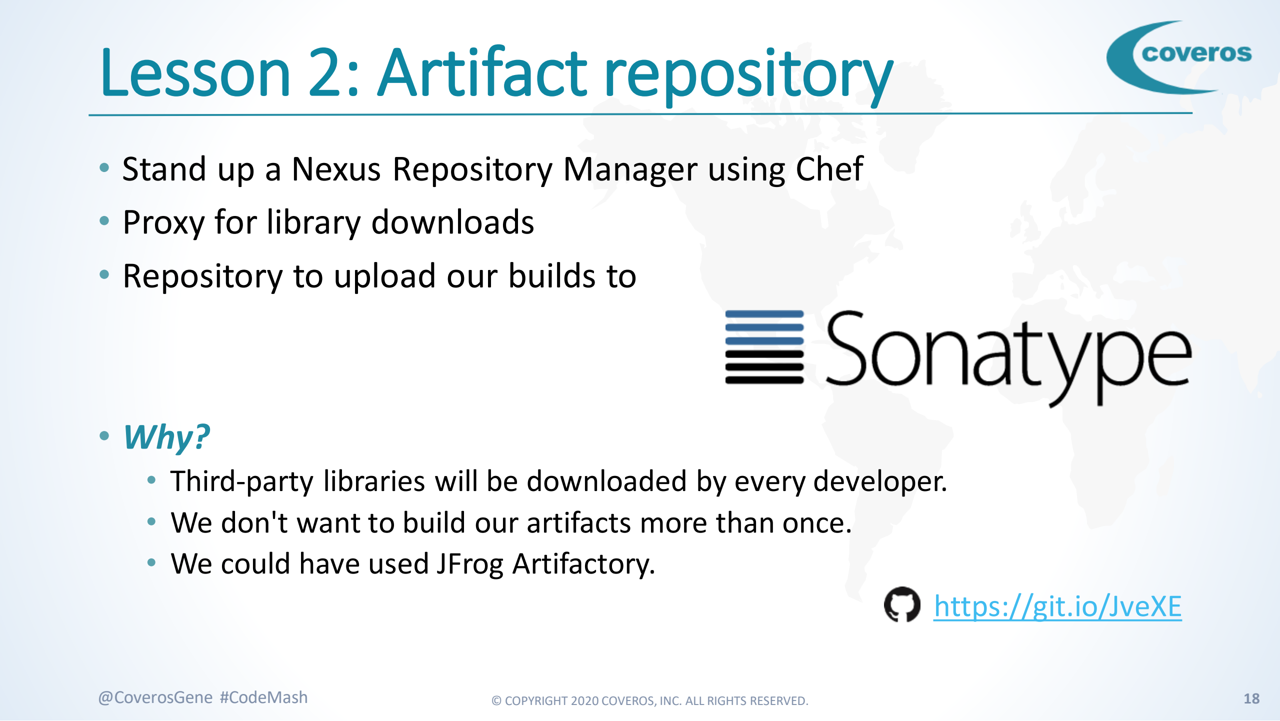


Lesson 1: Assess the risks

- Discuss critical parts of the JPetStore
- What worries you the most?
- Consider the OWASP Top 10 and CWE Top 25

- ***Why?***
  - You can't secure everything.
  - Even if you could, you don't have time.
  - Even if you have time, it isn't worth it.

https://git.io/JveXg

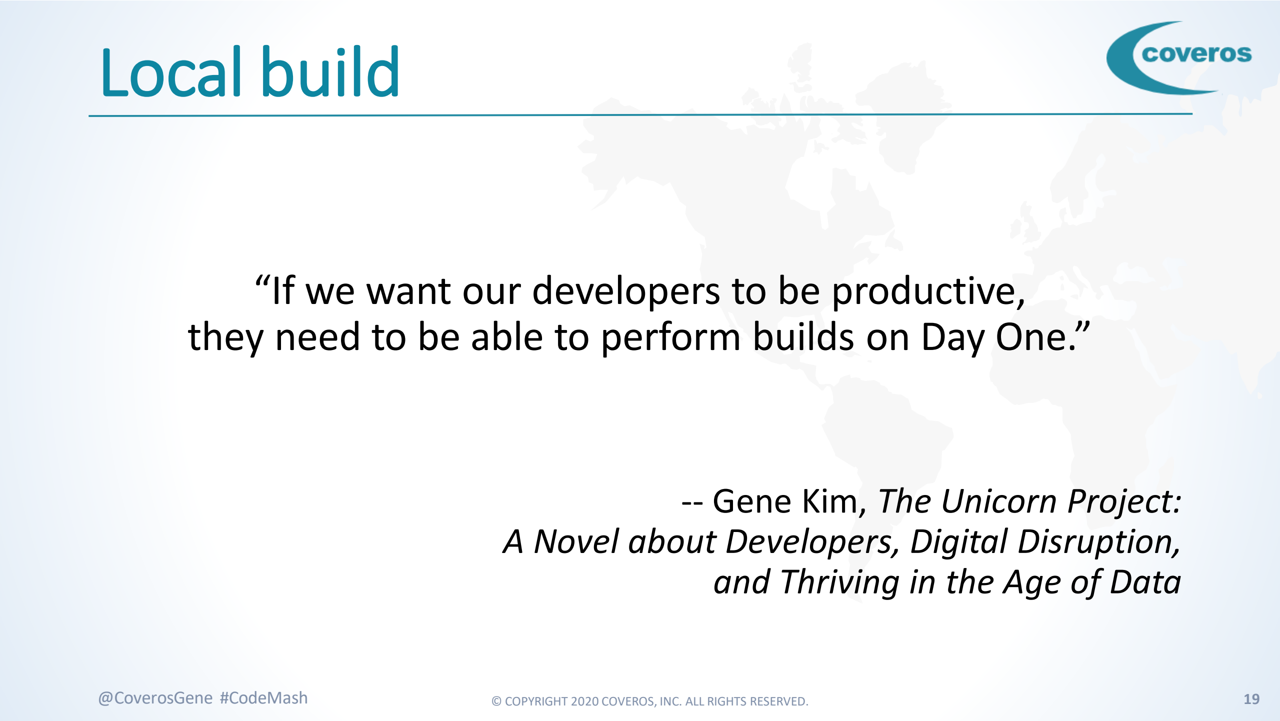


Lesson 2: Artifact repository

- Stand up a Nexus Repository Manager using Chef
- Proxy for library downloads
- Repository to upload our builds to

- ***Why?***
  - Third-party libraries will be downloaded by every developer.
  - We don't want to build our artifacts more than once.
  - We could have used JFrog Artifactory.

https://git.io/JveXE

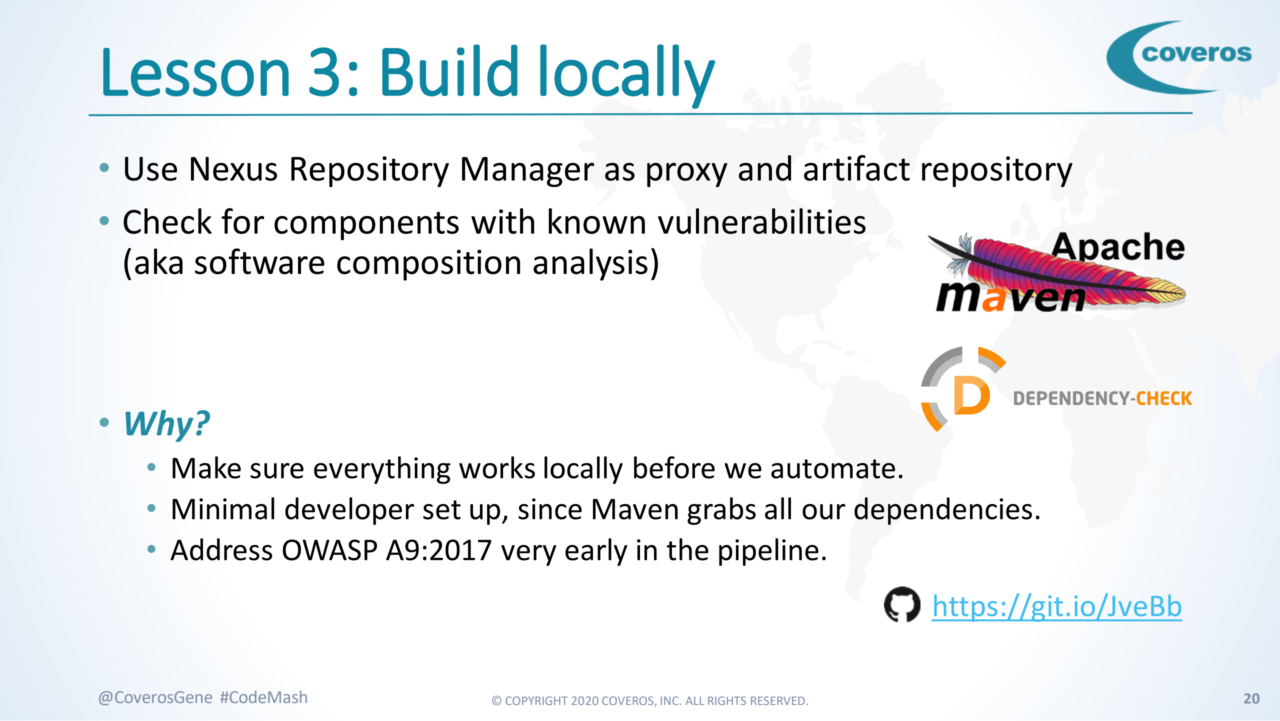


Local build

"If we want our developers to be productive,
they need to be able to perform builds on Day One."

-- Gene Kim, *The Unicorn Project:*
A Novel about Developers, Digital Disruption,
and Thriving in the Age of Data

@CoverosGene #CodeMash



Lesson 3: Build locally

- Use Nexus Repository Manager as proxy and artifact repository
- Check for components with known vulnerabilities (aka software composition analysis)
- ***Why?***
  - Make sure everything works locally before we automate.
  - Minimal developer set up, since Maven grabs all our dependencies.
  - Address OWASP A9:2017 very early in the pipeline.

https://git.io/JveBb

@CoverosGene #CodeMash

coveros

@CoverosGene #CodeMash

Lesson 4: Continuous Integration

coveros

- Build on a neutral machine
- Build automatically whenever code is pushed or when a pull request is created
- ***Why?***
  - Helps avoid the problems with "works on my machine."
  - Pull request reviewers can see that the build is passing or failing.
  - We need something to coordinate progress in our pipeline.

https://git.io/JveEL

@CoverosGene #CodeMash

Encryption

Authentication and authorization

Input validati

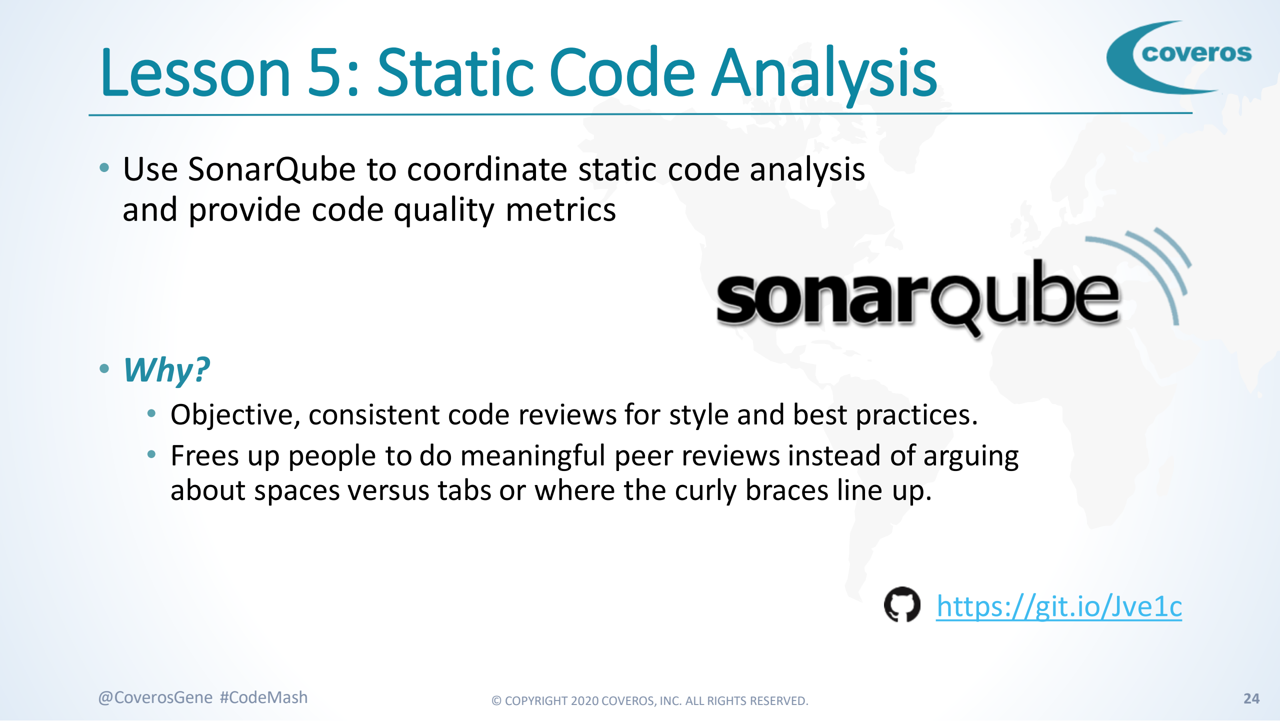
Output encoding

Architecture

Error conditions

Peer review the code

@CoverosGene #CodeMash



Lesson 5: Static Code Analysis

- Use SonarQube to coordinate static code analysis and provide code quality metrics

sonarqube

- ***Why?***
  - Objective, consistent code reviews for style and best practices.
  - Frees up people to do meaningful peer reviews instead of arguing about spaces versus tabs or where the curly braces line up.

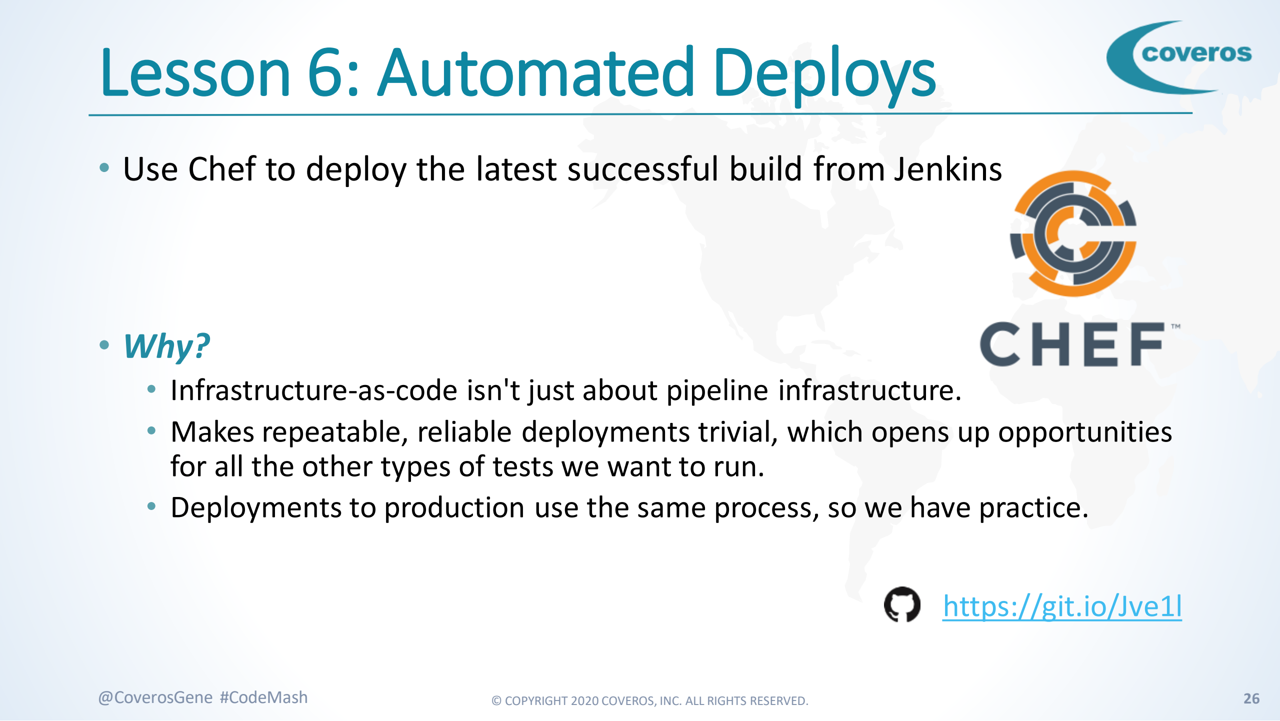
https://git.io/Jve1c

@CoverosGene #CodeMash

Repeatable, reliable deployments

@CoverosGene #CodeMash

25


coveros
CHEF

Lesson 6: Automated Deploys

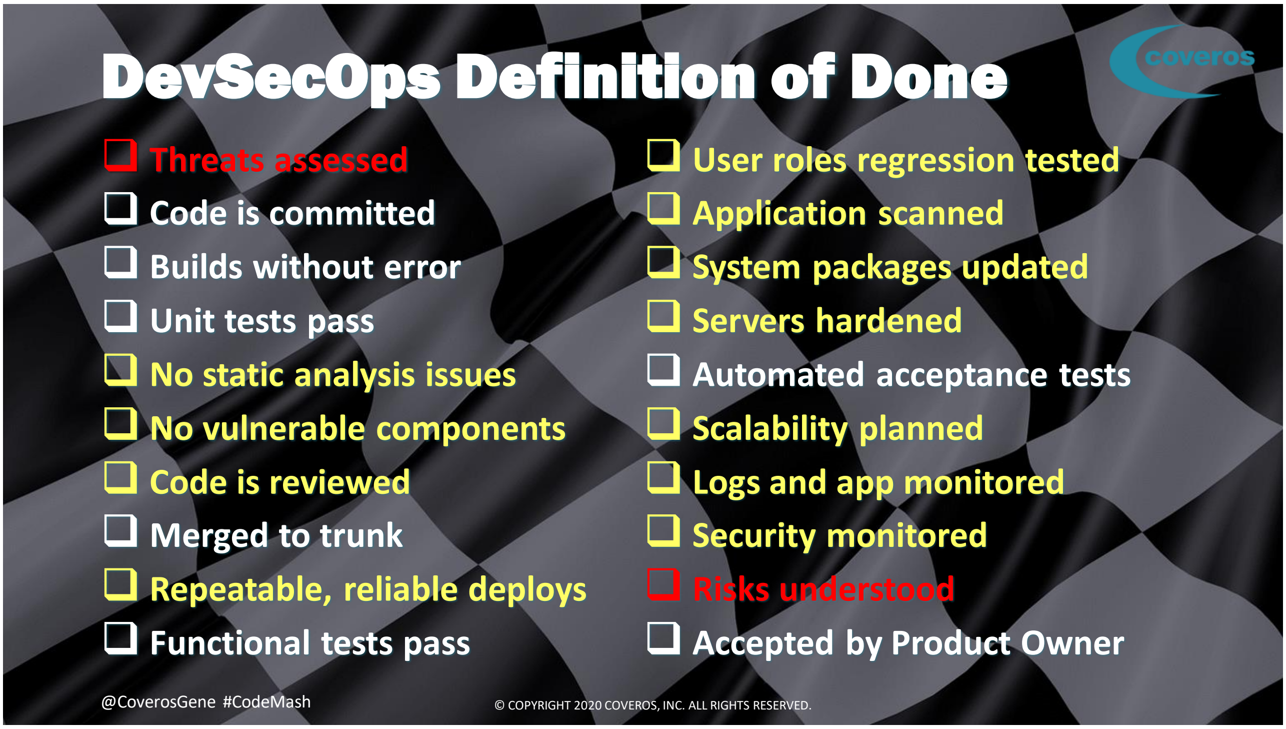
- Use Chef to deploy the latest successful build from Jenkins
- ***Why?***
  - Infrastructure-as-code isn't just about pipeline infrastructure.
  - Makes repeatable, reliable deployments trivial, which opens up opportunities for all the other types of tests we want to run.
  - Deployments to production use the same process, so we have practice.

https://git.io/Jve1l

@CoverosGene #CodeMash

26

26

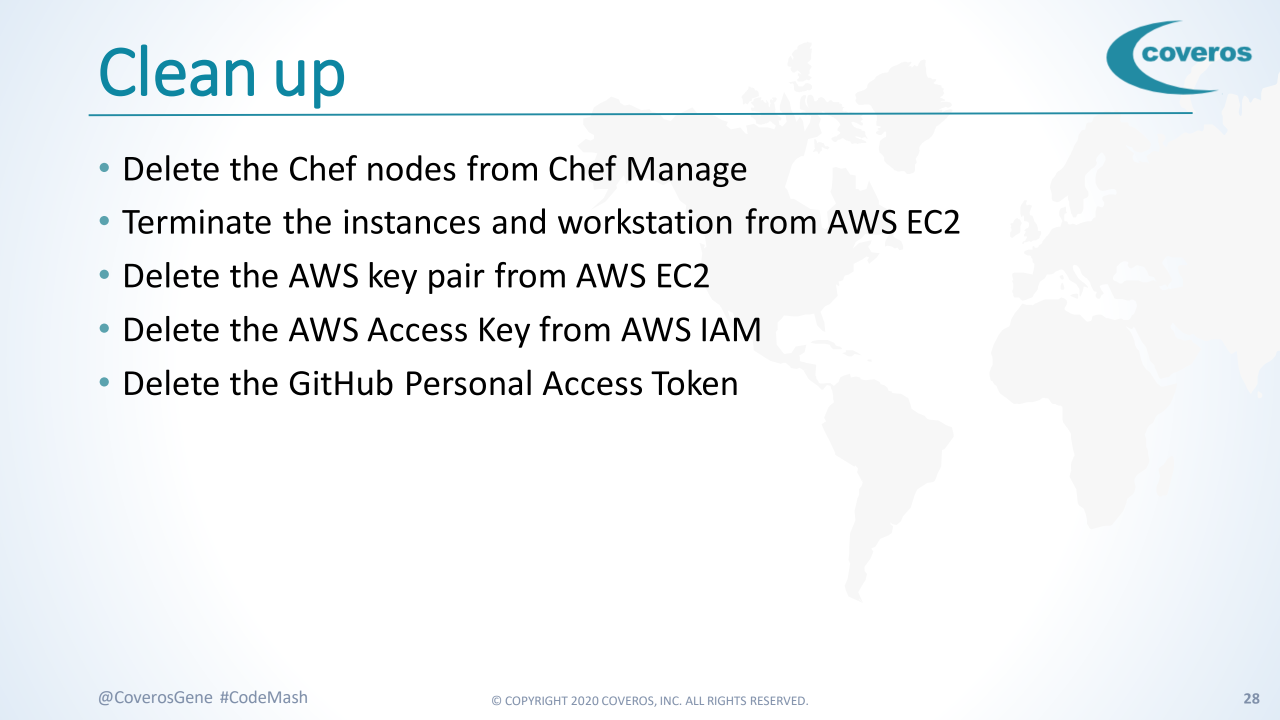


DevSecOps Definition of Done

- ☐ Threats assessed
- ☐ Code is committed
- ☐ Builds without error
- ☐ Unit tests pass
- ☐ No static analysis issues
- ☐ No vulnerable components
- ☐ Code is reviewed
- ☐ Merged to trunk
- ☐ Repeatable, reliable deploys
- ☐ Functional tests pass
- ☐ User roles regression tested
- ☐ Application scanned
- ☐ System packages updated
- ☐ Servers hardened
- ☐ Automated acceptance tests
- ☐ Scalability planned
- ☐ Logs and app monitored
- ☐ Security monitored
- ☐ Risks understood
- ☐ Accepted by Product Owner

@CoverosGene #CodeMash

27



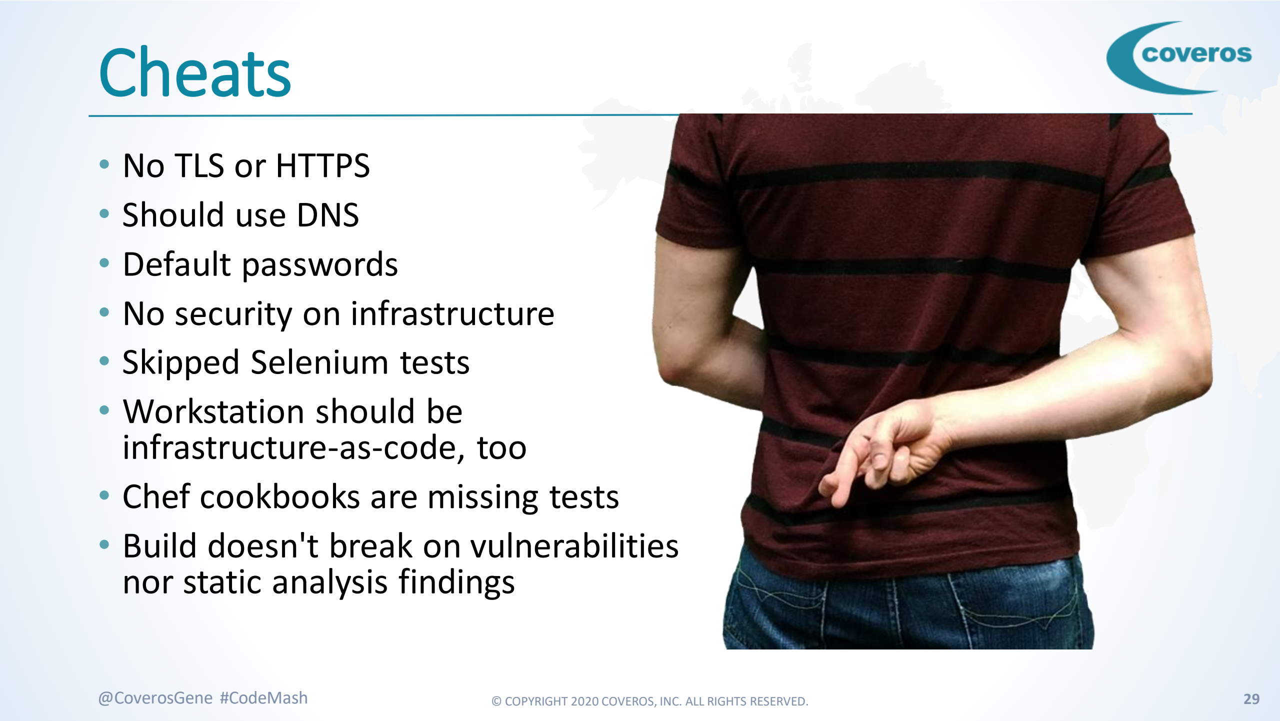
Clean up

- Delete the Chef nodes from Chef Manage
- Terminate the instances and workstation from AWS EC2
- Delete the AWS key pair from AWS EC2
- Delete the AWS Access Key from AWS IAM
- Delete the GitHub Personal Access Token

@CoverosGene #CodeMash

28

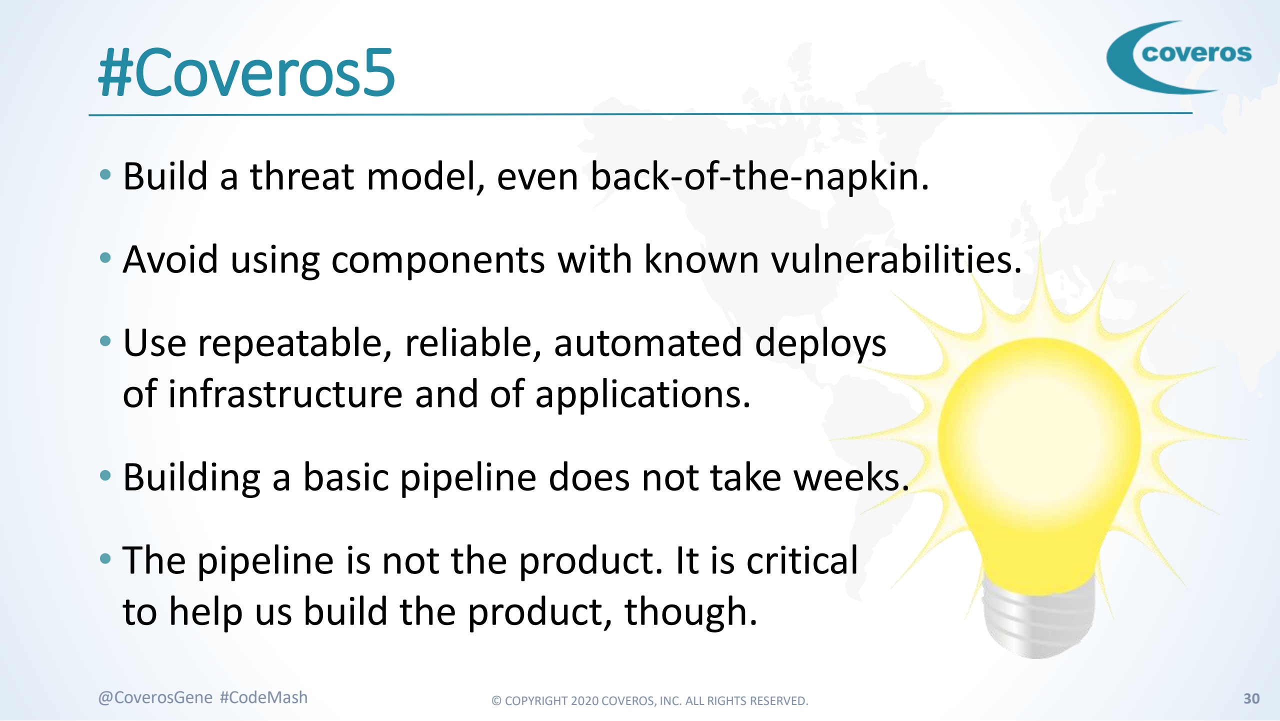
28



Cheats

- No TLS or HTTPS
- Should use DNS
- Default passwords
- No security on infrastructure
- Skipped Selenium tests
- Workstation should be infrastructure-as-code, too
- Chef cookbooks are missing tests
- Build doesn't break on vulnerabilities nor static analysis findings

@CoverosGene #CodeMash



#Coveros5

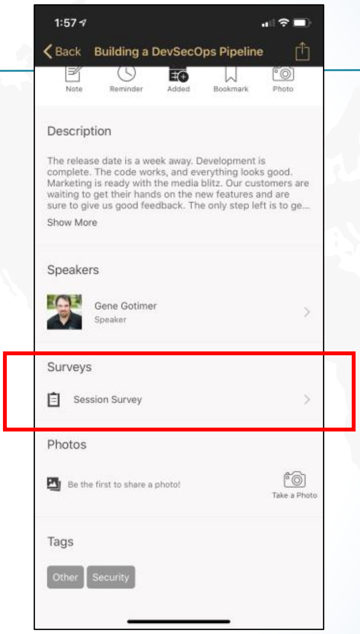
- Build a threat model, even back-of-the-napkin.
- Avoid using components with known vulnerabilities.
- Use repeatable, reliable, automated deploys of infrastructure and of applications.
- Building a basic pipeline does not take weeks.
- The pipeline is not the product. It is critical to help us build the product, though.

@CoverosGene #CodeMash

Session Feedback

- Pull out your phone
- Pull up the AttendeeHub app
- Navigate to this pre-compiler
- Choose Session Survey

- CodeMash and I want feedback
- Please



Building a DevSecOps Pipeline

Description

The release date is a week away. Development is complete. The code works, and everything looks good. Marketing is ready with the media blitz. Our customers are waiting to get their hands on the new features and are sure to give us good feedback. The only step left is to ge...

Show More

Speakers

Gene Gotimer
Speaker

Surveys

Session Survey

Photos

Be the first to share a photo!

Tags

Other Security

Questions?

- I'll be around CodeMash for the rest of the week
- Come see me at “**Tests Your Pipeline Might Be Missing**”
  - Thursday, Jan. 09, 11:45 AM - 12:45 PM in Mangrove

- I am @Gene Gotimer on the TechWell Hub Slack
  https://hub.techwell.com/

- Email: gene.gotimer@coveros.com
- @CoverosGene