Cyber Injection Points in the DevOps CI/CD Pipeline

Keith Conway

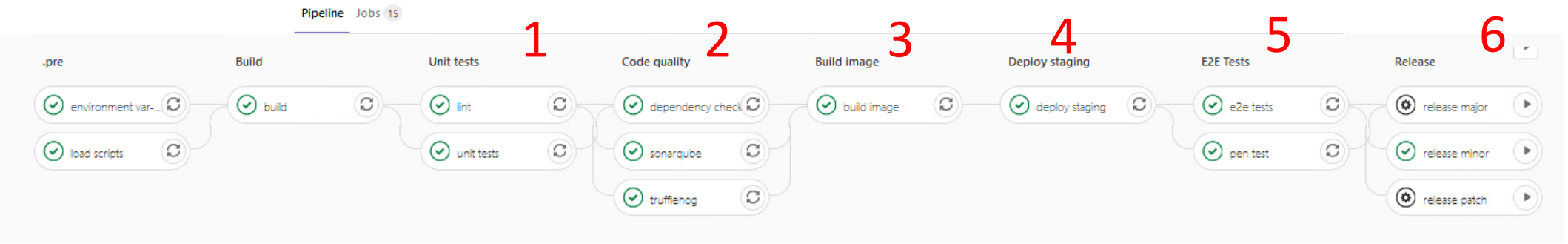
28July2021

Cyber Injection Points in the DevOps CI/CD Pipeline

- Continuous Integration (CI)
- Continuous Deployment (CD)
- Development to Operations (DevOps)
- Cyber – very generic term that will not be answered today

- My role is to identify and document cyber related best practices that could be used as a starting point for new programs or incorporated into existing programs. (slide 22)

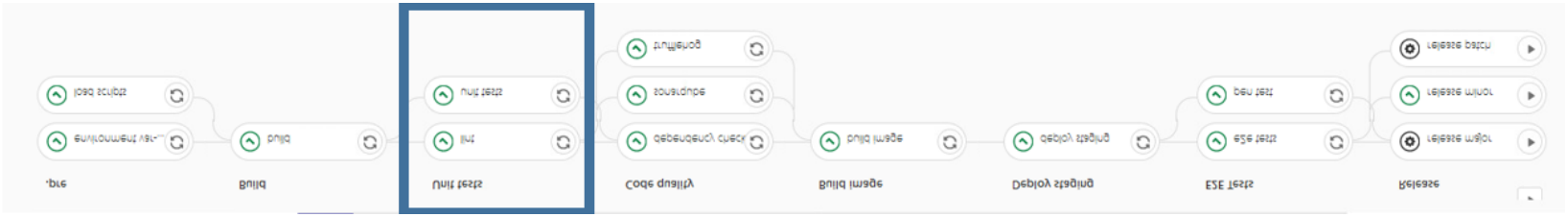
DevOps overview picture



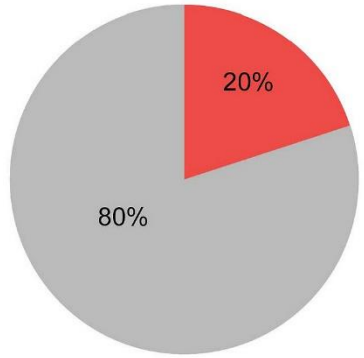
1. Unit Test
2. Code Quality
3. Build Image
4. Deploy Staging
5. E2E Tests
6. Release

Automated Unit Tests

- Automated Unit Tests - In computer science, test coverage is a measure used to describe the degree to which the source code of a program is executed when a particular test suite runs. A program with high test coverage, measured as a percentage, has had more of its source code executed during testing, which suggests it has a lower chance of containing undetected software bugs compared to a program with low test coverage.
- Atlassian is quoted – ‘it is generally accepted that **80%** coverage is a good goal to aim for.’



Automated Unit Test

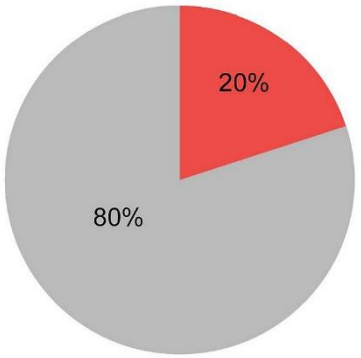


Unit tests are performed on individual classes and methods to ensure that they properly satisfy their API contracts with other classes. At this level, unit tests must be tested as isolated units without any interaction or dependency on other classes or methods. Unit tests are typically written by the developers themselves to verify the behavior of their code.

https://owasp.org/www-pdf-archive/AutomatedSecurityTestingofWebApplications-StephendeVries.pdf

Automated Unit Test - Cyber

Input Validation:
When testing security functionality it is important that both valid input is accepted (a functional requirement), and also that invalid and potentially dangerous data is rejected. Testing boundary and unexpected conditions is essential for security tests.



Automated Unit Test

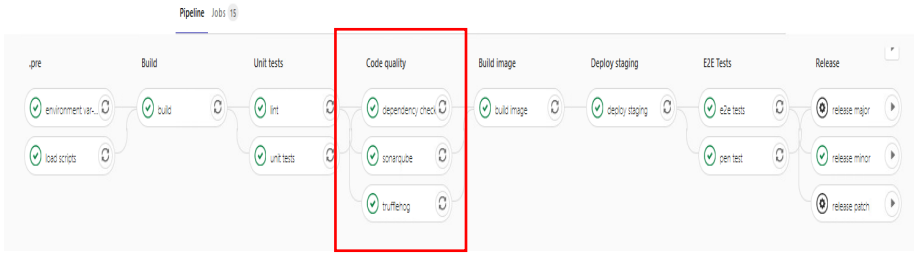
- OWASP top 10 that are tied to input validation:
- #1 – Injection
- #4 – XML external entities (XXE)
- #7 – Cross Site Scripting (CSS)
- #8 – Insecure Deserialization

Automated Unit Tests

- Questions?

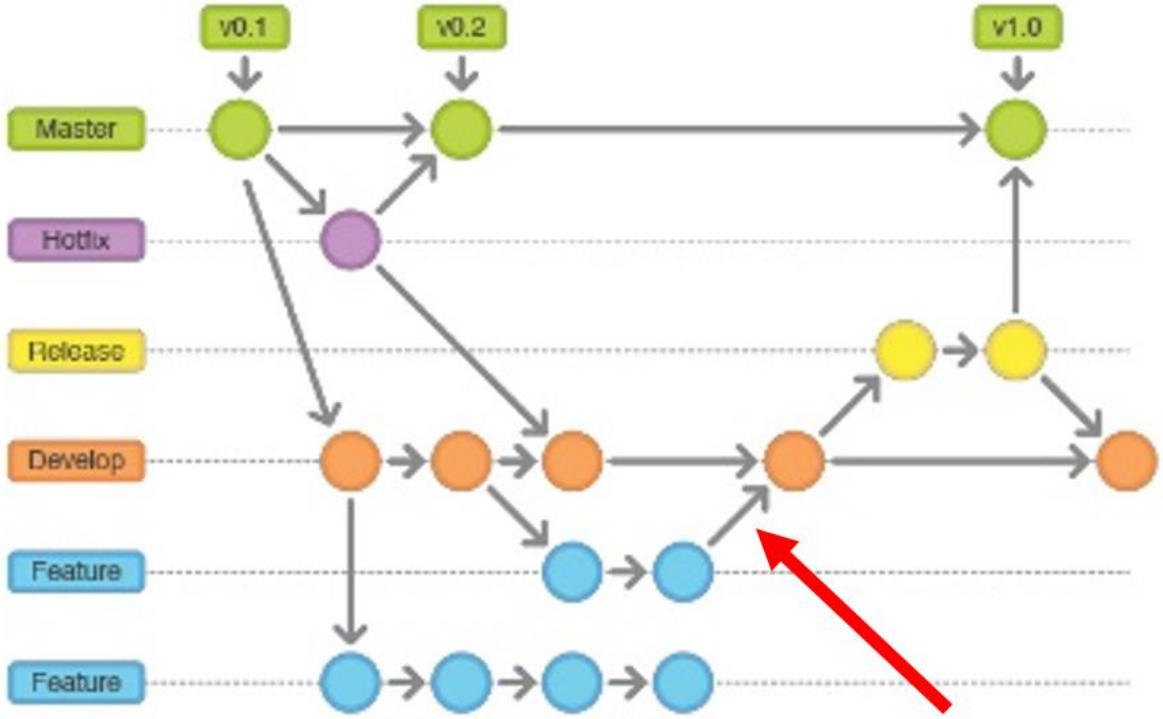
Code Security / Static Code Analysis

- Stop the leak
  - [Water Leak Changes the Game for Technical Debt Management (sonarsource.com)]
  - High Level Overview – stop rule violating code from being checked into the repository.
  - Addressing rule violating code that is already checked in is a different task.
- The main point is to discuss a security gate or a analysis to be required for a peer review.



Code Security / Static Code Analysis

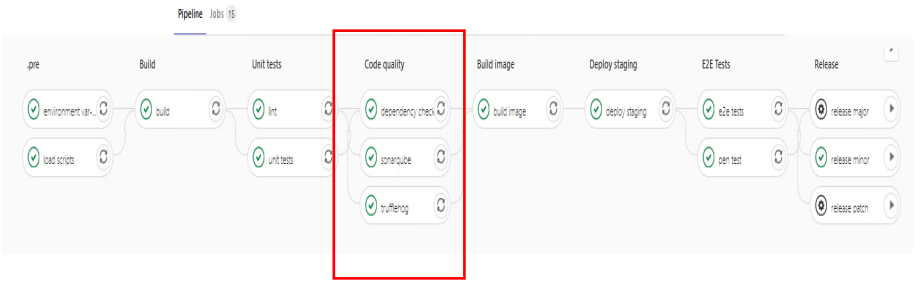
- Master
  - The holy grail aka Production
- Hotfix
  - maintenance outside of dev
- Release
  - ready for UAT
- Develop
  - main development stream
- Feature
  - individual feature / story



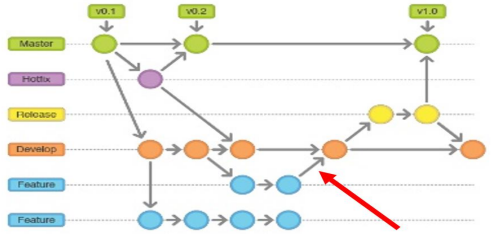
Code Security / Static Code Analysis

- Ways to stop rule violating code from being checked into the repo:

1. Quality Gate
   - Automated way to stop rule violating code from
   - Being checked into the repo.
2. Peer Review
   - Could have Code Security / Static Code Analysis report requirement



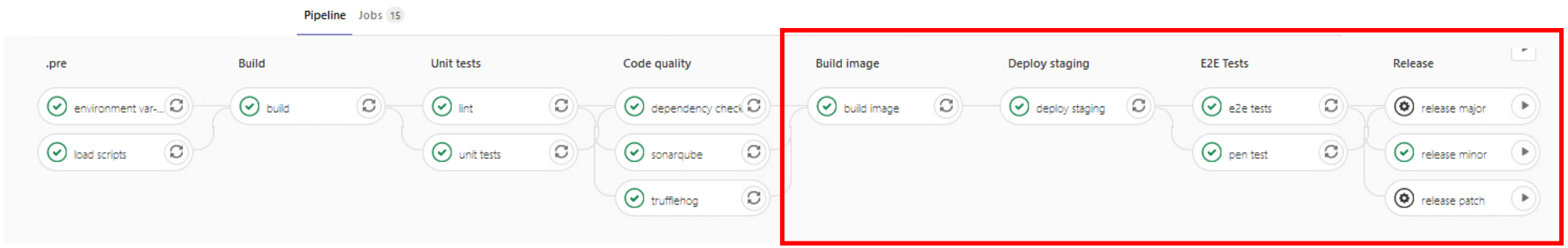
- Master
  - The holy grail aka Production
- Hotfix
  - maintenance outside of dev
- Release
  - ready for UAT
- Develop
  - main development stream
- Feature
  - individual feature / story



Code Security / Static Code Analysis

- Any questions on the static code analysis process?

Build Image / Deploy Staging / Release



Build Image: (Development)
Deploy Staging: (Integration)
Release: (Operations)

Infrastructure as Code

- What is Infrastructure as Code? - Azure DevOps | Microsoft Docs
- Environment drift – maintain the settings of individual deployment environments – becoming a snowflake – a unique configuration.
  - With snowflakes, administration and maintenance of infrastructure involves manual processes which are hard to track and contributed to errors.

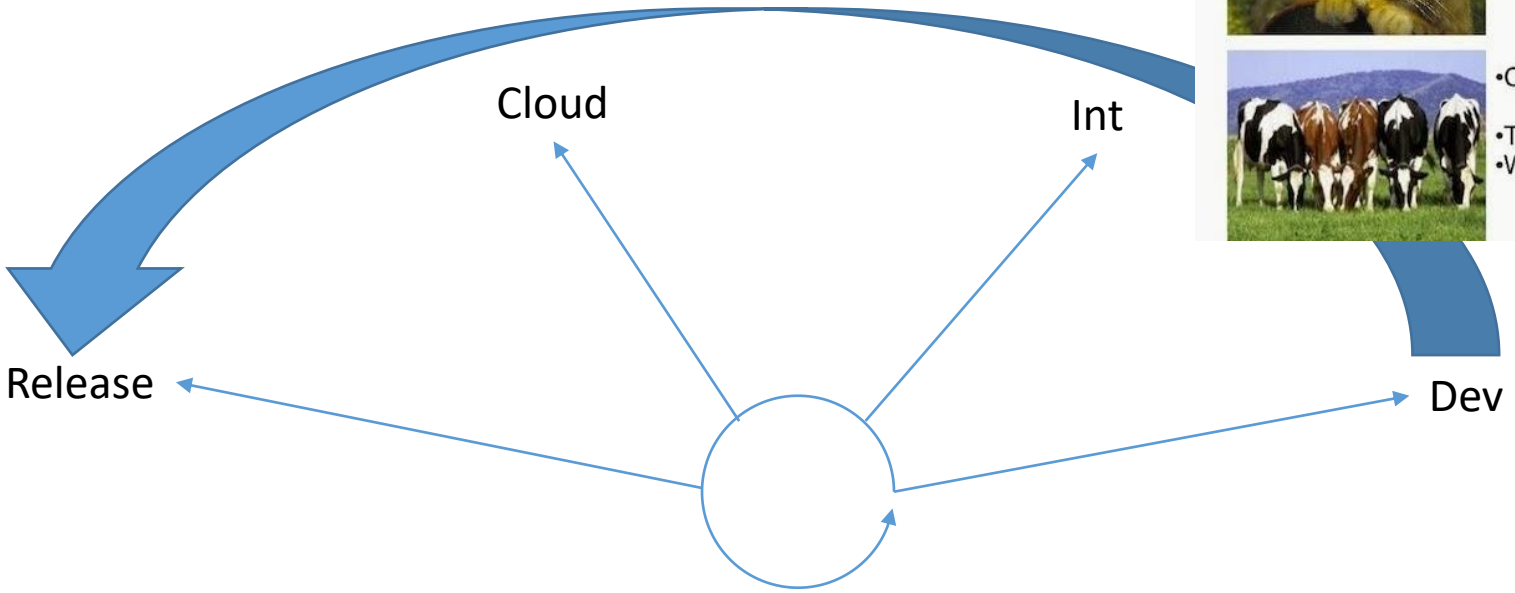
Infrastructure as Code

- Idempotence is a principle of Infrastructure as Code. Idempotence is the property that a deployment command always sets the target environment into the same configuration, regardless of the environment's starting state.
  - *Idempotency is achieved by either automatically configuring an existing target or by discarding the existing target and recreating a fresh environment.*

Hardened Operating System

- Hardened Operating System can be used by Development, different levels of Integration, and Operations.
  - Hardened VM can be copied to different areas: Dev, Int, Ops
  - Hardened Operating System will be versioned and can be used for either VM/container or bare metal : Dev, Int, Ops
  - Hardening via STIGs: will need to confirm that each environment is consistent: Dev, Int, Ops

DevOps Hardened VM Deployment



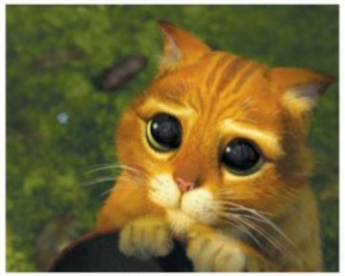
- Pets are given names like pussinboots.cern.ch
- They are unique, lovingly hand raised and cared for
- When they get ill, you nurse them back to health

- Cattle are given numbers like vm0042.cern.ch
- They are almost identical to other cattle
- When they get ill, you get another one

During a schedule a new hardened VM can be pushed to: Dev, Int, Cloud, Ops. Schedule could be 7, 15, 30, 60, 90 day.
Automated script used to install/configure service on new hardened VM.
Automated installation script developed / configured / tested on Dev/Int network – alleviates manual one time configurations at Release site (Snowflake vs Idempotent).

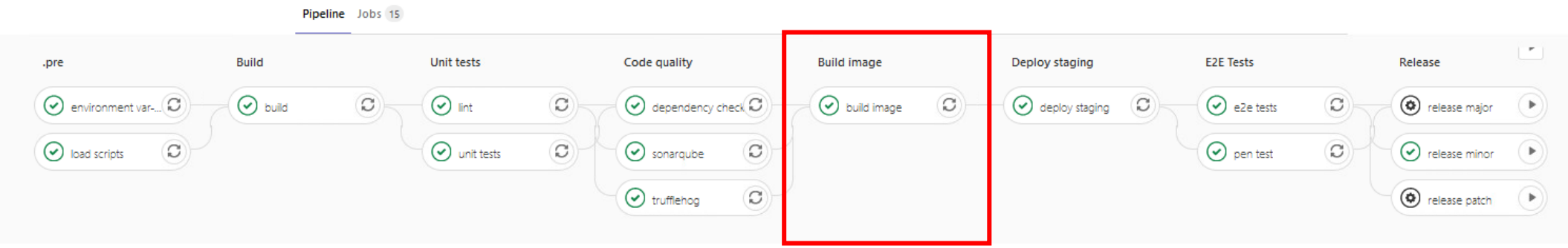
Hardened Operating System CI/CD DevOps Pipeline

Pets vs Cattle



- Pros:
  - No upstream configuration issues as each phase of dev / test is using same operating system
  - Can be used for third party integrators as well
  - * No ownership of machine – which can be changed out when a new configuration managed operating system is generated.
- Cons:
  - Generating configuration managed operating system takes time to develop
  - Development is slower because operating system changes go through change control board
  - * No ownership of machine – which can be changed out when a new configuration managed operating system is generated.

Build Image

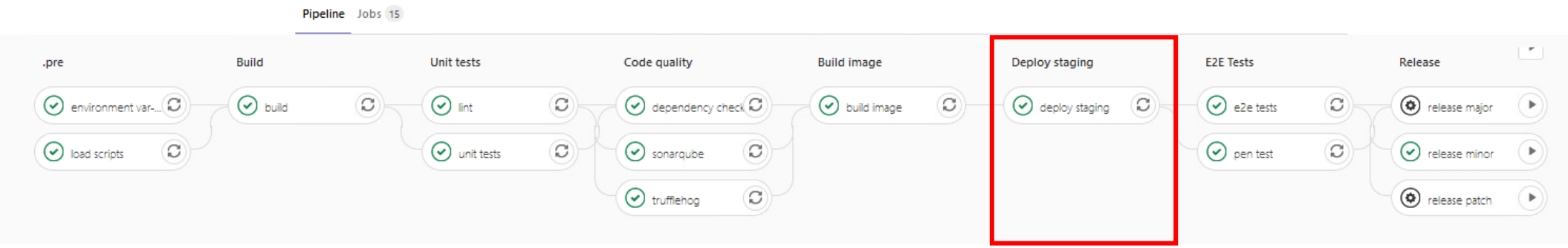


Build Image: (Development) This is where the software build is installed on an approved hardened operating system image via an automated script.

- Hardened operating system could be a VM, a container, or a operating system installation disk (gold image)

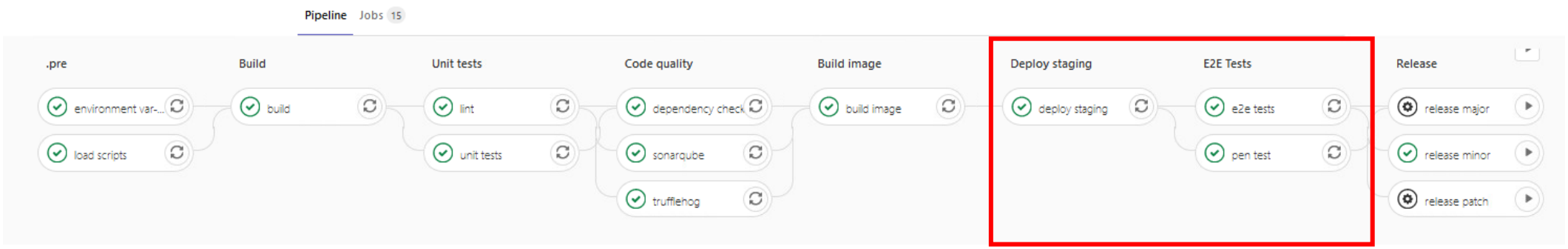
We will need to have the approved hardened operating system image available via a repository.
We will need to identify what type of script language to use to install the software service.

Deploy Staging



Deploy Staging: (Integration) This is where the software build (application or service) is installed on a approved hardened VM image via an automated script.
The software build and automated script are supplied – the hardened VM needs to be part of an integration server (not owned by the software developer).

Deploy Staging and E2E Tests



E2E Tests: End to End

Could be an automated test to pass a security requirement.

OWASP ASVS

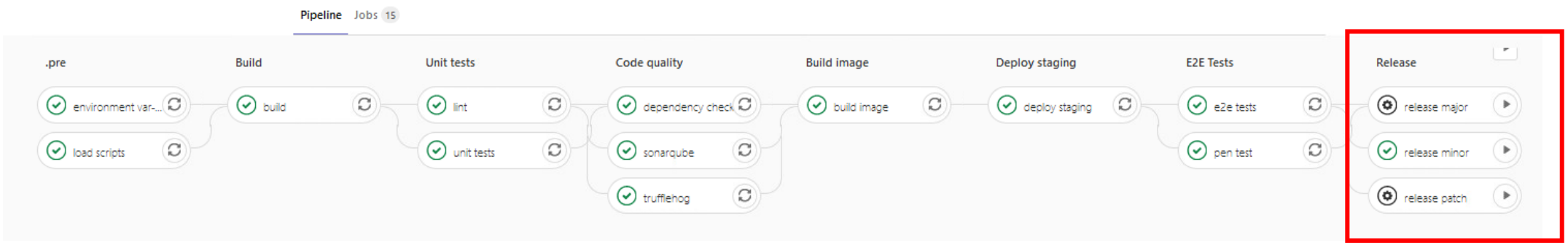
https://owasp.org/www-project-application-security-verification-standard/

Security architecture has almost become a lost art in many organizations. The days of the enterprise architect have passed in the age of DevSecOps. The application security field must catch up and adopt agile security principles while re-introducing leading security architecture principles to software practitioners. Architecture is not an implementation, but a way of thinking about a problem that has potentially many different answers, and no one single "correct" answer. All too often, security is seen as inflexible and demanding that developers fix code in a particular way, when the developers may know a much better way to solve the problem. There is no single, simple solution for architecture, and to pretend otherwise is a disservice to the software engineering field.

OWASP ASVS

- The primary aspects of any sound security architecture:
  - Availability
  - Confidentiality
  - processing integrity
  - non-repudiation
  - privacy.

Release



Release: (Operations) This is when the VM/Container created from the Deploy Staging phase is transitioned to operations.

Thank you – Keith Conway

- Questions?

Keith Conway

Keith is a lifelong learner and has always been interested in understanding the bigger picture. He has a BS in Computer Science from SDSU and an MBA from a local university in San Diego, Alliant International Univ.

- Cyber certifications include Certified Information Systems Security Professional (CISSP), Certified Ethical Hacker (CEH), Security+, and a certified pentester (penetration tester) through Offensive Security (OSWP).
- Programming certifications include Oracle Java Developer and Amazon Web Services (AWS) Cloud Solutions Architect.
- INCOSE Certified Systems Engineering Professional (CSEP) and has served two years on the INCOSE San Diego board of directors.