Performing DevSecOps Automated Security Testing

INITIALIZING THE SETUP FOR AUTOMATED SECURITY TESTING

Peter Mosmans
LEAD PENETRATION TESTER

@onwebsecurity https://www.onwebsecurity.com

Scenario

Maeve

"I'm ready to start using automated security tests"

"How can I perform those automated security tests?"

"What kind of tools can I use?"

Jennifer

Demonstrate a variety of different tools

Demonstrate how tools can be used within existing pipelines

Perform automated security testing

Course Overview

Initializing the Setup for Automated Security Testing

Automating Code Security Testing

Automating Third Party Libraries Security Testing

Automating Container Security Testing

Automating Infrastructure Security Testing

This course demonstrates
different types of tools.
Lots of tools.
Lots of demos.

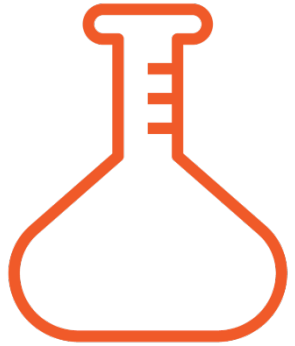
Per Tool We're Going to Look At:

Advantage

Is it genuinely useful?

Compatibility

Is it easy to use?



Trialability

Is it easy to try out?

What's In It for Maeve?

Maeve

Jennifer

Know more about different tools

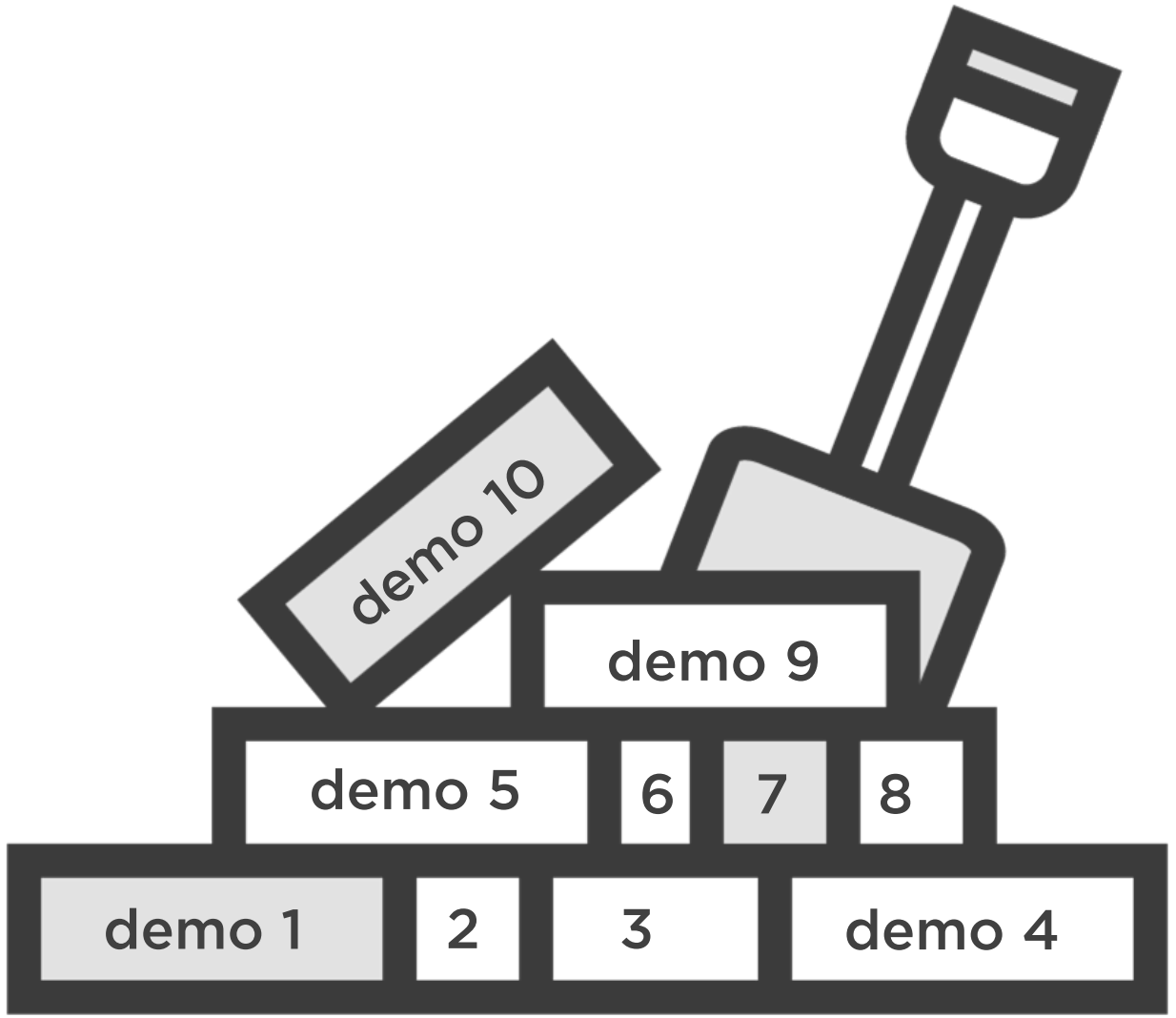
Know more about various types of tools

How to start using tools in the CI/CD pipeline

Who's This Course For?

- DevOps engineers
- Security professionals
- Developers
- Product owners
- Scrum masters
- Anyone interested in automated security testing

Gradual Build Up



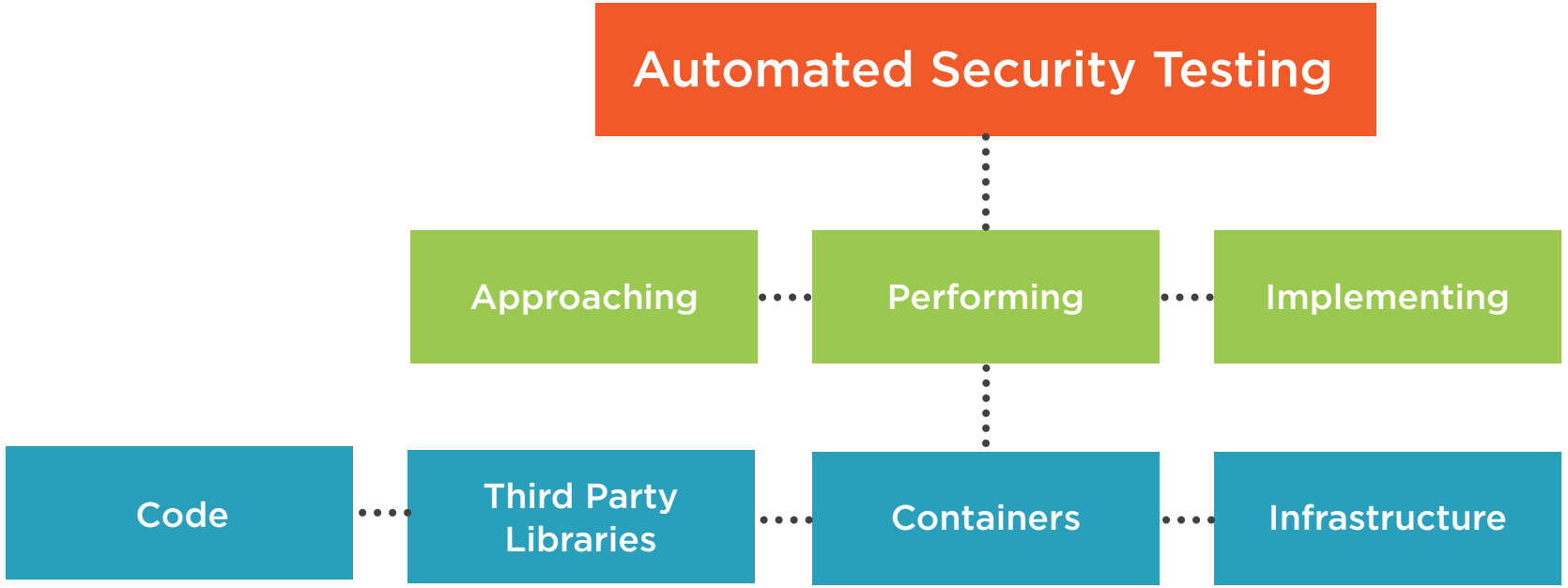
Part of the DevSecOps Path

Approaching
Automated
Security Testing

Performing
DevSecOps
Security Testing

Integrating
DevSecOps
Security Testing

Automated Security Testing Overview



Module Overview

Automating code security testing

Describing the demo lab

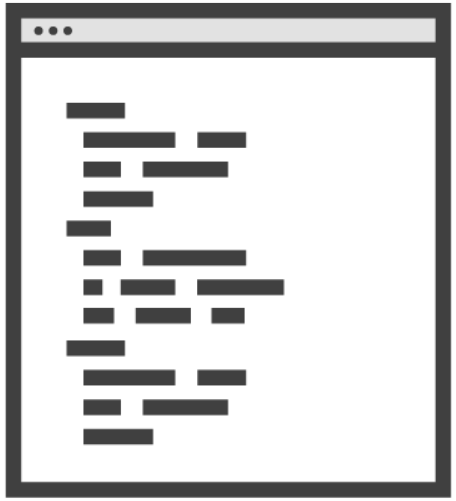
Demos:

- Setting up the demo lab
- Setting up a build pipeline

Summary

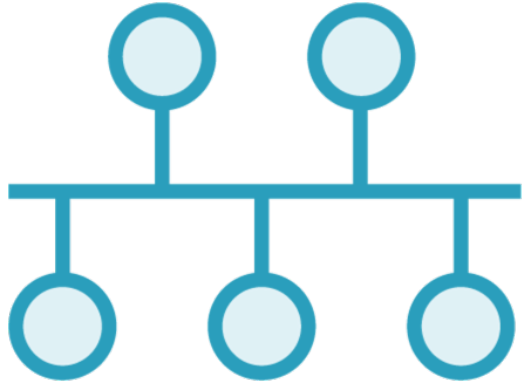
Automating Code Security Testing

What to Test



Code

Containers



Infrastructure

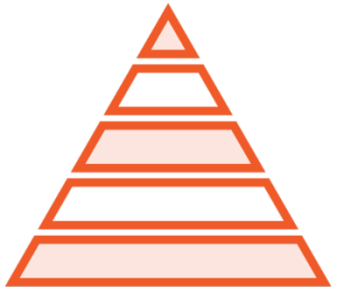
Code

- Readability
- Maintainability and clarity
- Insecure patterns
- (Hardcoded) secrets
- Insecure third-party libraries

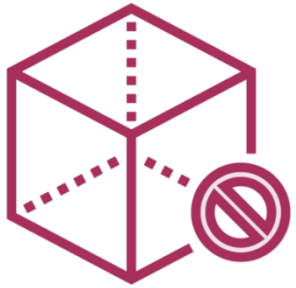
How to Approach Automated Security Testing

Let the team pick its own tools

Start with quick wins first



Invest time in setting a baseline



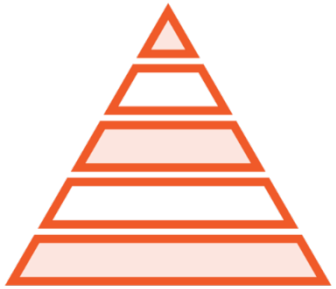
Don’t use hard quality gating

Plan time to configure and weed out false positives

Things to Take into Consideration

Don’t use hard quality gating

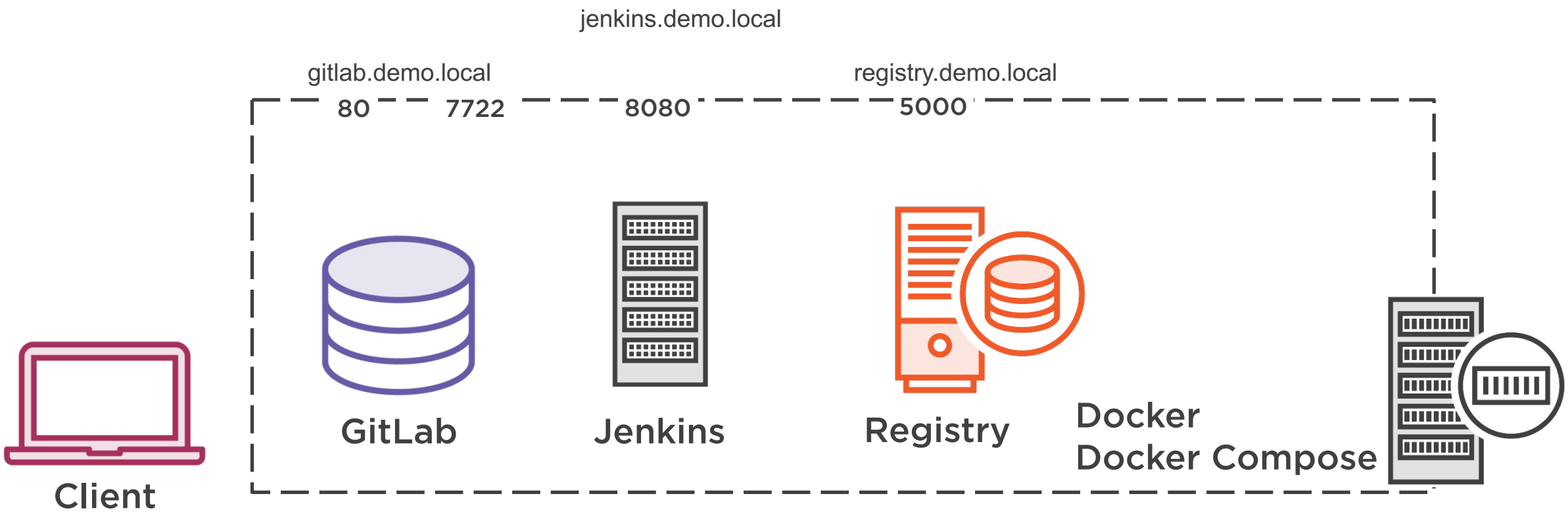
Plan time to configure and weed out false positives



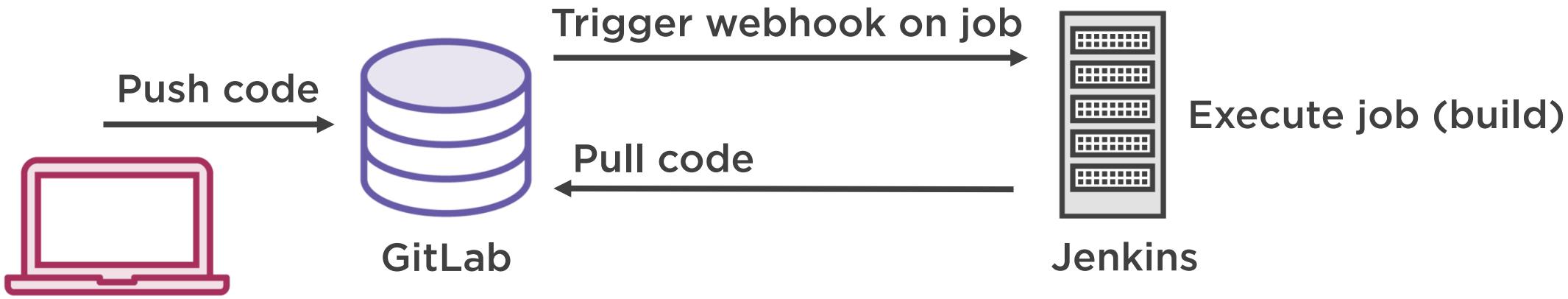
Invest time in setting a baseline

Describing the Demo Lab

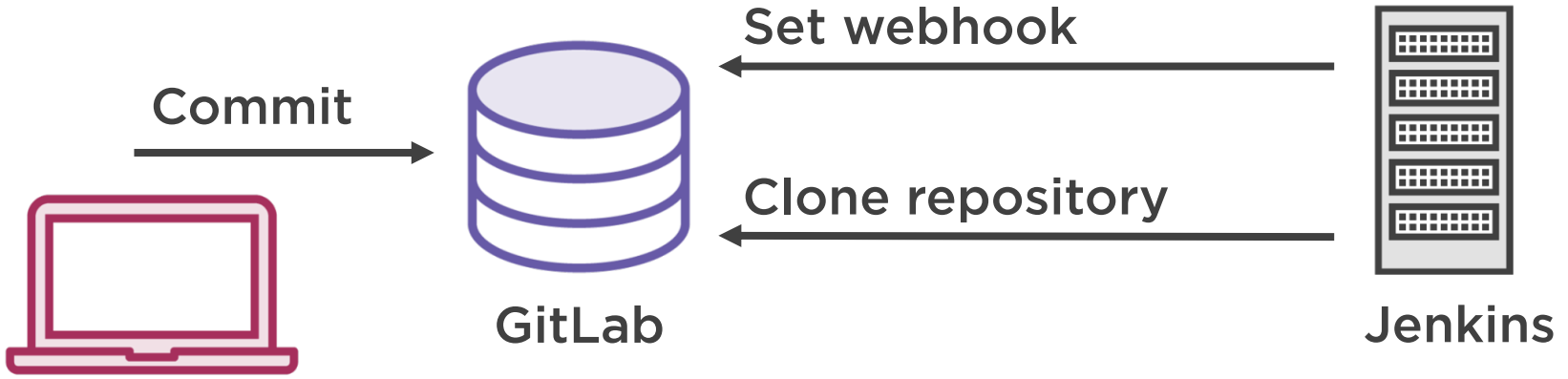
Demo Lab


jenkins.demo.local
gitlab.demo.local
registry.demo.local
80
7722
8080
5000
GitLab
Jenkins
Registry
Docker
Docker Compose
Client

Workflow


Trigger webhook on job
Push code
Pull code
Execute job (build)
GitLab
Jenkins

Access Permissions



Two Demo Projects

Docker base image project

- Builds a Docker image containing security testing tools
- https://github.com/PeterMosmans/tools-image

Node.js web shop project

- A (deliberately insecure) web shop
- https://github.com/bkimminich/juice-shop

More Information

https://gitlab.com/

https://www.jenkins.io/

https://docs.docker.com/registry/

https://github.com/PeterMosmans/devsecops-lab/

https://github.com/PeterMosmans/tools-image/

Demo

Setting up the demo lab

- Run and configure GitLab
- Run and configure Jenkins
- Run Docker registry

Prerequisites:

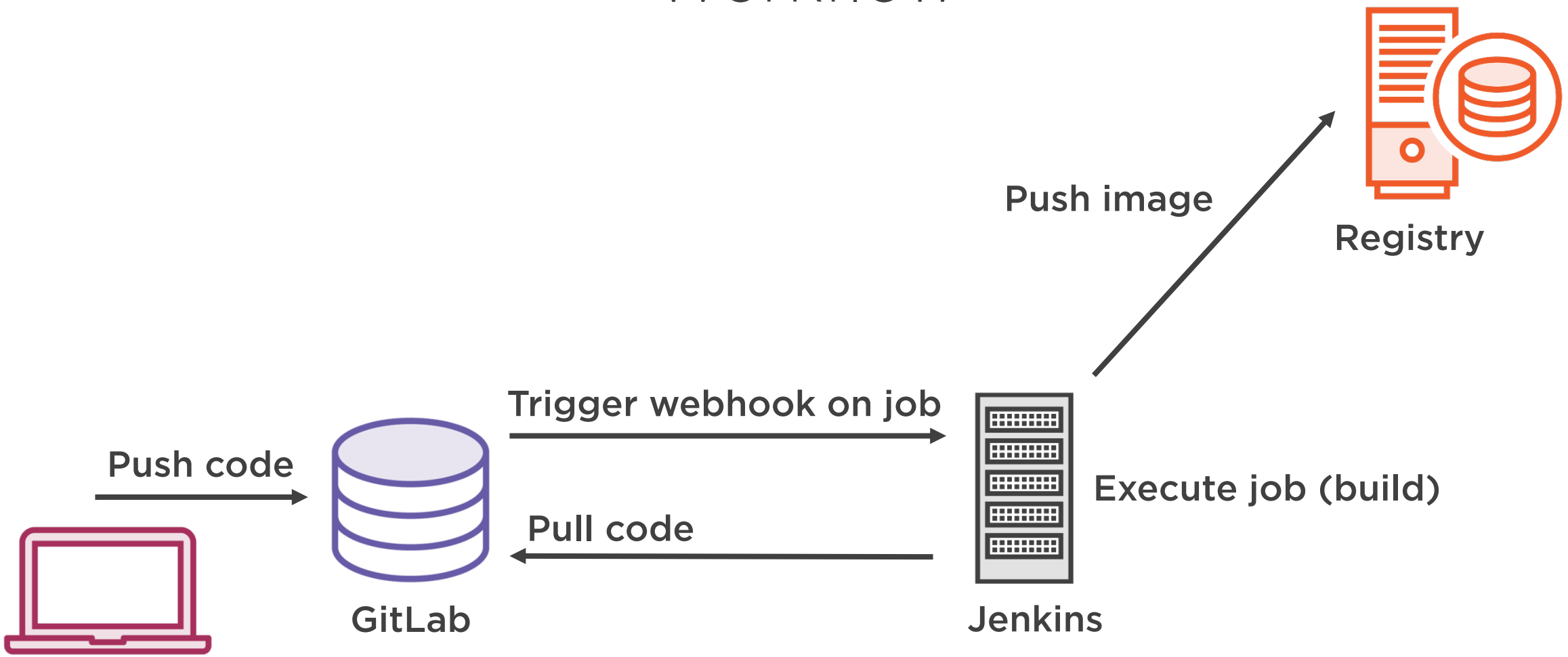
- Docker
- Docker Compose

Demo

Setting up the tools-image project

- Create GitLab project
- Create Jenkins project
- Automatically build tools image

Workflow


Push image
Registry
Trigger webhook on job
Push code
Execute job (build)
Pull code
GitLab
Jenkins

Summary

Setting up a simple automated pipeline

Don’t use hard quality gates up front

Plan time to implement tools

Invest time in setting up a baseline

Next Up

"Now I know how to set up a testing pipeline"

Maeve

"Let's start with security testing code"

Jennifer