



# Welcome! GitLab



PLURALSIGHT

Hello

**HELLO**  
my name is

**Allen Sanders**  
Senior Technology Instructor  
Pluralsight ILT

**About me...**



- 30 years in the industry
- 25 years in teaching
- Certified Cloud architect
- Passionate about learning
- Also, passionate about Reese's Cups!



# Agenda

- Learning objectives
- What is GitLab?
- Building your application in GitLab
- Securing your application in GitLab



## How We're Going to Work Together

- Slides and words to highlight key concepts
- Demos to bring those concepts “to life”
- Discussion groups and lab work (which will take place in sandboxes provided via AWS WorkSpaces) for hands-on reinforcement
- NOTE: I welcome being interrupted – if you need more info, or clarification, or anything else, just break in and ask. I am here to help you.



# Learning Objectives



## Learning Objectives

- Understand key concepts and considerations when leveraging DevSecOps in an Agile environment to help “shift security left” during software development
- Construct and prioritize a threat model for an application being developed and use that threat model as a planning tool to guide each phase of the SDLC in a security-minded manner
- Speak to the value of DevSecOps and its consistent application as a set of standards and best practices



## Learning Objectives

- Monitor, patch and scan for vulnerabilities in the Operating System (Windows and Linux) and underlying Infrastructure configuration
- Effectively utilize GitLab for Source Code Management and CI/CD, including:
  - Understanding and navigating practical activities related to source code repositories in GitLab
  - Effectively use Software Composition Analysis (SCA), Static Application Security Testing (SAST), and Dynamic Application Security Testing (DAST)



# What is GitLab?



## GitLab Features

Complete SDLC

Full Support for CI/CD

Robust Version  
Control (Git)

Full Support for  
DevSecOps

Plan & Track Work

Infrastructure Support

Full Support for  
Package & Release

# Terms



- Project – Where work is organized, managed, tracked, and delivered as part of software creation (like a repository)
- Group – Collection of projects or other groups (like a project)
- Issue – Unit of work in GitLab project (akin to a User Story)
- Epic – Collection of related issues
- Merge Request – Request for merge of one developer's changes with the rest of the team (like a Pull Request)
- Milestone – Deliverable(s) out of a GitLab project for completion (like a release)

# GitLab Flow



<https://about.gitlab.com/topics/version-control/what-is-gitlab-flow/>



# Building Your Application in GitLab

## LAB:

Complex Pipeline

[https://docs.gitlab.com/ee/ci/quick\\_start/tutorial.html](https://docs.gitlab.com/ee/ci/quick_start/tutorial.html)

## LAB:

Publish Packages with CI/CD

[https://docs.gitlab.com/ee/user/packages/pypi\\_repository/auto\\_publish\\_tutorial.html](https://docs.gitlab.com/ee/user/packages/pypi_repository/auto_publish_tutorial.html). For the Python app, you can use the app & tests at <https://github.com/KernelGamut32/python-jenkins>.

## LAB:

Setup CI/CD Steps

[https://docs.gitlab.com/ee/tutorials/setup\\_steps/index.html](https://docs.gitlab.com/ee/tutorials/setup_steps/index.html)

# LAB:

## Project Runners

[https://docs.gitlab.com/ee/tutorials/create\\_register\\_first\\_runner/index.html](https://docs.gitlab.com/ee/tutorials/create_register_first_runner/index.html)





# Securing Your Application in GitLab

## LAB:

### Dependency Scanning

[https://docs.gitlab.com/ee/tutorials/dependency\\_scanning.html](https://docs.gitlab.com/ee/tutorials/dependency_scanning.html)

## LAB:

### Docker Container Scanning

[https://docs.gitlab.com/ee/tutorials/container\\_scanning/index.html](https://docs.gitlab.com/ee/tutorials/container_scanning/index.html)

## LAB:

Security End-to-End

<https://university.gitlab.com/learn/course/hands-on-lab-security-essentials/main/hands-on-challenge-gitlab-security-essentials>

## LAB:

### Security End-to-End

Update your Docusaurus solution to include the various concepts we've been learning – add dependency scanning, SAST, DAST, additional stages, etc. to build out a more full-featured pipeline. Address any High or Critical vulnerabilities uncovered.



# Thank you!

If you have additional questions,  
please reach out to me at:  
[asanders@gamuttechnologysvcs.com](mailto:asanders@gamuttechnologysvcs.com)



PLURALSIGHT