

## DevOps Using GitHub

---

### Course Summary

#### Description

GitHub boasts being the “largest and most advanced development platform in the world” ever evolving to meet the demands of its expansive developer community. The centerpiece of GitHub, as its name implies, is in the provisioning of Git remote repositories. However, providing the network storage of code has never been its full story. GitHub has always sought to empower development teams to deliver new and improved ways to support the entire software development process. In this training, we will focus on GitHub’s support of DevOps starting with the use of WebHooks to easily integrate with external systems then moving on to Actions for automation of the coding lifecycle. Explore the GitHub DevOps platform, examine its typical workflows, and experience GitHub tooling through practical demonstrations and exercises. Learn about DevOps solution architecture and how GitHub is supporting creative scripted processes, delivering effective time saving automation for continuous integration and continuous delivery.

#### Objectives

At the end of this course, students will be able to:

- Describe DevOps
- Understand the GitHub’s DevOps platform
- Create and work with Actions for Continuous Integration
- Define environments in support of Continuous Delivery
- Manage DevOps execution from development through to deployment

#### Topics

- Introduction to DevOps
- Introduction to DevOps on GitHub
- GitHub WebHooks
- GitHub Actions
- Deployment Environments
- Introduction to Docker

#### Audience

This course is designed for development and operations teams who are new to DevOps and the GitHub DevOps platform

#### Prerequisites

Foundational Git experience and knowledge are required including:

- Working with the local repository
- Pushing and pulling to and from the remote repository

Experienced using GitHub for Git projects is assumed.

Comfortable working in bash (unix shell) is an asset.

#### Duration

Three days

## DevOps Using GitHub

---

### Course Outline

#### I. *DevOps Fundamentals*

- A. Definition
  1. What is DevOps?
  2. Key Terms
- B. Overview of DevOps
  1. The DevOps Story
  2. More Than CI/CD
  3. Business Drivers
  4. Where Technology Fits In
- C. The Big Picture
  1. DevOps Vision
  2. DevOps Scope
  3. DevOps Goals
  4. DevOps Context
  5. DevOps and Agile
  6. DevOps and Cloud
- D. Moving to DevOps
  1. DevOps Readiness
  2. DevOps Roadmap
  3. Activity: Setup Repository
  4. Activity: Setup Repository – Results
- E. DevOps Fundamentals – Summary

#### II. *DevOps Lifecycle*

- A. DevOps Lifecycle
  1. DevOps Lifecycle – Visual
  2. Lifecycle Phases
  3. Hope and Reality
- B. Continuous Creativity
  1. Plan
  2. Code
  3. Create
  4. Build
  5. Test
  6. Supporting Continuous Integration
- C. Continuous Experience
  1. Release
  2. Deploy
  3. Operate
  4. Monitor
  5. Supporting Continuous Deployment
  6. Activity: Getting Started with DevOps
  7. Activity: Getting Started with DevOps – Results
- D. DevOps Lifecycle – Summary

#### III. *Introduction to DevOps on GitHub*

- A. GitHub DevOps
  1. GitHub DevOps Lifecycle
  2. GitHub DevOps Components
  3. Runners and Servers
  4. Cloud Definition
  5. GitHub Hosted DevOps
  6. Self Hosted GitHub DevOps
  7. Self Hosted GitHub DevOps Requirements
- B. Overview of Integrations
  1. Integration
  2. GitHub APIs
  3. GitHub Apps
  4. GitHub App Configuration
  5. GitHub App Management
  6. Using URL parameters
  7. Activity: Install GitHub App
  8. Activity: Install GitHub App – Results
- C. Overview of WebHooks
  1. What Are WebHooks?
  2. Why Use WebHooks?
  3. WebHooks Architecture
  4. Adding WebHooks
  5. WebHook Payloads
  6. Activity: Use a WebHook
  7. Activity: Use a WebHook – Results
- D. Overview of Actions
  1. What Are Actions?
  2. Workflow Scopes
  3. Workflow Architectures
  4. CI/CD Actions
  5. Activity: First Actions Workflow
  6. Activity: First Actions Workflow – Results
- E. Introduction to DevOps on GitHub ~ Summary

#### IV. *Understanding WebHooks*

- A. WebHooks and Actions
  1. WebHooks Use Cases
  2. WebHooks – Pros and Cons
  3. Actions Use Cases
  4. Actions – Pros and Cons
  5. WebHooks vs Actions

#### Designing DevOps on GitHub

## DevOps Using GitHub

---

### Course Outline (cont'd)

- B. Working with WebHooks
    1. WebHook Interface
    2. WebHook API
    3. WebHook Scopes
    4. Activity: Explore WebHook Controls
    5. Activity: Explore WebHook Controls – Results
  - C. WebHook Events and Payloads
    1. WebHook Events
    2. WebHook Payloads
    3. WebHook Events By Use Case
    4. WebHook Payloads By Events
    5. Activity: Review Payloads for WebHook Events
    6. Activity: Review Payloads for WebHook Events – Results
  - D. WebHook Management
    1. Security
    2. Testing
    3. Status
    4. Activity: Manage WebHooks
    5. Activity: Manage WebHooks – Results
  - E. Understanding WebHooks – Summary
- V. *Coding GitHub Actions*
- A. Action Workflow Definition
    1. The GitHub Workflow YAML File
    2. File Structure
  - B. GitHub Action Workflow File Syntax
    1. Workflows
    2. Jobs
    3. Steps
    4. Implementation Options
    5. Workflow Commands
    6. Environment Variables
    7. Workflow Rules
    8. Activity: Sample Workflows
    9. Activity: Sample Workflows – Results
  - C. Executing Workflows
    1. Runners
    2. Workflow Triggers
    3. Workflow Events
    4. Configuring Workflows
    5. GitHub Actions Interface
    6. Activity: Trying Trigger Types
    7. Activity: Trying Trigger Types – Results
  - 8. Activity: Update Actions Workflow
  - 9. Activity: Update Actions Workflow – Results
  - D. Coding GitHub Actions – Summary
- VI. *GitHub CI/CD Templates*
- A. GitHub CI/CD Templates Overview
    1. GitHub CI/CD Templates Use Cases
    2. GitHub CI/CD Template Categories
    3. Working with the CI/CD Templates
    4. Activity: Examine the CI/CD Templates
    5. Activity: Examine the CI/CD Templates – Results
  - B. Customize CI/CD Templates
    1. Custom CI/CD Templates
    2. Why Customize?
    3. Reuse with Workflows
    4. Composite Workflows
    5. Creating Custom Workflow Templates
    6. Activity: Custom Workflow Template
    7. Activity: Custom Workflow Template – Results
  - C. GitLab CI/CD Template Library – Summary
- VII. *Introduction to Docker*
- A. Overview of Docker
    1. What is Docker?
    2. Docker Concepts
    3. Use Cases
    4. Containerization vs Virtualization
    5. Docker Images
    6. Activity: Getting Started with Docker
    7. Activity: Getting Started with Docker – Results
  - B. Working with Docker
    1. Docker Architecture
    2. Docker Components
    3. Define a Docker Image
    4. Configure Docker Build the Image
    5. Run the Image
    6. Test Docker
    7. Activity: Running Docker
    8. Activity: Running Docker– Results
  - C. Introduction to Docker – Summary

## DevOps Using GitHub

---

### Course Outline (cont'd)

#### VIII. *GitHub Packages*

- A. Introduction to GitHub Packages
  - 1. What Are Packages?
  - 2. GitHub Package Components
  - 3. Package Types
- B. Working with GitHub Packages
  - 1. Create a Package
  - 2. Publish a Package
  - 3. View a Package
  - 4. Install a Packages
  - 5. Delete a Package
  - 6. Continuous Delivery with GitHub Packages
  - 7. Activity: Create a GitHub Package
  - 8. Activity: Create a GitHub Package – Results
  - 9. Activity: Add a Package Workflow
  - 10. Activity: Add a Package Workflow – Results
- C. Docker Packages
  - 1. Docker for DevOps
  - 2. Using Docker Images
  - 3. Docker Image Configuration
  - 4. Activity: Work with Docker on GitHub
  - 5. Activity: Work with Docker on GitHub – Results
- D. GitHub Packages – Summary

#### IX. *GitHub Deployments*

- A. Working With Deployments
  - 1. What are Deployments?
  - 2. Deployment vs Delivery
  - 3. Controlling Deployments
  - 4. Interactive Deployments
  - 5. Configuring Environments
  - 6. Managing Environments
  - 7. Infrastructure as Code
  - 8. Viewing Environments and Deployments
  - 9. Deployment Status
  - 10. Deployment History
  - 11. Activity: Deploy Application
  - 12. Activity: Deploy Application – Results
- B. Continuous Deployments Using Docker
  - 1. Continuous Deployment Use Cases
  - 2. Why Docker for Continuous Deployment?
  - 3. Continuous Deployment Components
  - 4. Continuous Deployment Workflows
  - 5. Activity: CI/CD Using Docker
  - 6. Activity: CI/CD Using Docker – Results
- C. GitHub Deployments – Summary