

**CI/CD with GitHub**

**Course Number:** GIT-108
**Duration:** 1 day

**Overview**

This CI/CD with GitHub training course teaches teams how to implement continuous integration and continuous deployment (CI/CD) pipelines leveraging Git and GitHub. Attendees get extensive hands-on exposure to GitHub Actions, building CI/CD pipelines from scratch while learning best practices.

**Prerequisites**

* Experience with Git (students should be comfortable working with Git from the command-line, leveraging branches, merging, pushing, pulling and opening merge/pull requests via a remote GitLab, GitHub, etc.).
* Experience with the command-line or DOS command prompt.
* Experience with Docker and the concepts of containerization is beneficial but not required.

**Materials**

All Git training attendees receive comprehensive course materials.

**Software Needed on Each Student PC**

* Git 2.x or later
* Internet access for all attendees and the instructor

**Objectives**

All students will:

* Describe what CI/CD is and why it is beneficial
* Create custom CI/CD workflows using GitHub Actions
* Understand the benefits of managing infrastructure as code
* Gain a good understanding of how to integrate external systems into GitHub
* Know best practices when implementing CI/CD pipelines
* Get exposure to containerization and creating images

**Outline**

* Introduction to GitHub
	+ Overview of features
	+ How GitHub fits into the SDLC
	+ Comparison to other platforms
* Introduction to CI/CD
	+ What is CI/CD
	+ The typical CI/CD workflow in GitHub
	+ Benefits of CI/CD and best practices
* Defining a typical GitHub Workflow with Actions
	+ Test
	+ Build
	+ Deploy
* Running within your own infrastructure
	+ Intro to Runners
	+ When might you use a runner?
	+ Hosted, Self-hosted and local runners
	+ Advanced Runner Options
* Advanced Workflows
	+ Triggering workflows and actions
	+ Composite Actions
	+ DRY / publishing and/or sharing frequently used actions
	+ Leveraging Docker and JavaScript within Actions
	+ Managing usage restrictions/limits
* Environment Variables and Permissioning
	+ Defining shared variables
	+ Keeping secrets safe
	+ Permissions and Job Policies
	+ Best practices
* Artifacts & Dependencies
	+ Creating and leverage artifacts from your actions
	+ Managing artifacts / storage
	+ Managing artifact dependencies between actions
* Deployments Strategies
	+ Push vs Pull based deployments
	+ Building and leveraging container images
	+ Using a container registry
* Advanced Pipeline Actions
	+ Security Scanning
	+ License Audits
	+ Scheduled Jobs
* Migrating from your existing CI system (Optional - topic is customized based on team needs)
	+ CircleCI
	+ Jenkins
	+ Travic
	+ GitLab / BitBucket
* Conclusion