

**DevOps CI/CD Pipeline**

**Course Number:** DVOP-150  
**Duration:** 3 days

**Overview**

Developers don’t use isolated software tools, so why learn tools in isolation? This DevOps CI/CD Pipeline training teaches attendees Git from the ground up, allowing them to securely push code to a remote Git repository. Next, attendees learn how to configure servers using a Configuration Management tool from code stored in Git. Finally, they integrate Git and Configuration Management with a CI/CD tool to build, test, and deploy their code into Test, Staging, and Production environments, creating an automated end-to-end DevOps pipeline.

The outline below can use your specific combination of technologies to create a custom class with the all tools you need and nothing you don't. For example, build a class covering Bitbucket-Ansible-CircleCI.

**Pick One Git Topic:**

* Bitbucket
* GitHub
* GitLab
* AzureRepos
* AWS CodeCommit

**Pick One Configuration Management System:**

* Ansible
* Chef

**Pick One CI/CD Pipeline Tool:**

* CircleCI
* Jenkins
* AzurePipelines
* AWS CodePipeline
* Atlassian Bamboo
* GitHub Actions

**Prerequisites**

All students must have:

* Understanding the purpose of infrastructure servers, such as web servers, load balancers and database servers
* Familiarity with scripting of any kind
* Experience with working at the command line of either Unix, Linux or Windows PowerShell
* Experience with a text editor of any kind

**Materials**

All DevOps CI/CD Pipeline training students receive comprehensive courseware.

**Software Needed on Each Student PC**

* A modern web browser and an Internet connection free of restrictive firewalls, so that the student can connect by SSH and Remote Desktop (RDP) into remote virtual machines.

**Objectives**

* Leverage the fundamentals of Git so they can effectively share, collaborate, backup, and version any code.
* Use Configuration Management to spin up dozens, hundreds, or even thousands of servers and get exactly what is desired from each one, every time.
* Apply Continuous Integration/Continuous Deployment to automatically download code and test it, and then make code approval requests from the larger dev team to be delivered to the end location.

**Outline**

* Introduction
* Technology Overview
  + Git - Source Control Management
  + Configuration Management
  + Continuous Integration / Continuous Deployment
* Git: Source Control Management
  + Purpose overview and use cases
  + Git workflow
  + Configuring git
  + Getting help with git
  + Local vs. Global vs System configurations
  + Basic git commands
  + Creating local git repositories
  + Branching and merging
  + Using remote repositories (Github, Bitbucket, GitLab, AzureRepos or AWS CodeCommit)
  + Pushing code to remote repositories using public and private SSH keys
* Configuration Management:  Ansible or Chef
  + Architecture and call flow
  + Installation, configuration and validation
  + Managing hosts and nodes
  + Managing servers through scripting
  + Shipping policies: writing repeatable code
  + Shipping quality code:
    - Linting
    - Unit testing
    - Integration testing
  + Dynamic configuration and discovering server details
  + Using variables across multiple servers
  + Creating groups of managed servers
  + Using community configurations
  + Automating Configuration Management updates
* CI/CD
  + CI/CD overview, use cases and history
  + CI/CD architecture
  + Project configurations
  + CI/CD as code
  + Environment variables and parameters
  + Distributed builds
  + Customizing project views
  + Managing credentials and secrets
  + Integrating CI/CD with git
  + Triggers: webhooks and polling
  + Notifications: instant messenger integration and SMTP email
  + Approving builds
  + Testing code through CI/CD
  + Running builds from multiple git branches
  + Applying conditional logic within CI/CD scripts
  + Deploying Configuration Management code with CI/CD:
* Conclusion