Implementing Pipeline as Code Using Jenkins

- Learn practical techniques for building and working with Jenkinsfile
- Understand declarative and scripted pipelines
- Utilize core pipeline-as-code concepts like nodes, stages and steps
- Develop multi-branch pipelines with Jenkins Pipeline and Jenkinsfiles
- Understand pipeline visualization

As organizations look to improve the speed with which they deliver software, they increasingly turn to Continuous Integration/Continuous Delivery (CI/CD) pipelines and infrastructure-as-code software architecture and delivery techniques to help leverage value from their DevOps adoptions. While many of the steps in a pipeline are automated, management of the pipeline itself remains a largely manual process. Pipeline as code gives teams the ability to define and manage an entire DevOps CI/CD pipeline in code, allowing them to store pipeline configurations in source control, version them, and independently test them.

The Jenkins CI server supports pipeline as code through a concept known as a Jenkinsfile. This is a configuration file that allows teams to define each step in their pipeline. This means that by using a Jenkinsfile, developers no longer have to manually create Jenkins jobs or actively manage the pipeline and can focus on developing and testing their applications.

This course is an extension to ourFoundations of DevOps—ICAgile Certification[1] course and will teach you practical techniques for building and working with Jenkinsfile. Upon completion of this course, students will understand and have hands-on experience with Jenkinsfile, including:

- Declarative and scripted pipelines
- Core pipeline-as-code concepts like nodes, stages, and steps
- Developing Multi-branch pipelines with Jenkins Pipeline and Jenkinsfiles
- Pipeline visualization
- CI/CD Best Practices

Hands-on Exercises

In this 1-day hands-on workshop, students will build a multibranch pipeline using Jenkins and Git.

Who Should Attend

This course is especially appropriate for both Developers and Operations Engineers. Both will learn ways to collaborate more in the orchestration of builds, artifact management, and automated deployments. Basic familiarity with the Linux command line interface is assumed.

Laptop Required

With their laptops, participants will construct, experiment with, and orchestrate their own pipelines gaining valuable experience on the hows and whys of pipeline as code as well as potential implementation pitfalls. Participants will be provided a virtual machine image to work within, minimizing prep time outside of class. Participants will need to obtain free GitLab and Oracle accounts, install Putty [2] on their laptop, and download the (large) image file prior to the start of class. Complete instructions will be emailed to all registrants.

Course Outline

Introduction to Pipeline As Code
What is a pipeline?
Infrastructure as Code
Pipeline as Code

Overview of Jenkins
Freestyle vs pipeline jobs
Plugins

Building and Maintaining Jenkinsfiles
Scripted vs Declarative style
Defining pipeline stages and steps
Connecting to SCM, artifact repositories, and other CI/CD infrastructure
Environment variables and credentials
Introduction to the Groovy language
Restrictions imposed by the Groovy sandbox
Using Global Libraries to share pipeline code between projects
Maintenance and refactoring strategies
Versioning

Pipeline visualization
Traditional pipeline visualization
Pipeline visualization using Blue Ocean

Managing Resources
Sharing resources between branches and jobs
Ensuring resource cleanup

CI/CD Best Practices for Multi-branch Pipelines
Testing strategies
Deployment strategies
Notification strategies