Introduction to Groovy Scripting in Jenkins Environment

Groovy is an object-oriented scripting language based on Java platform. It is a dynamic language and is dynamically compiled to Java Virtual Machine (JVM) byte-code and interoperates with other Java code and libraries. Groovy increases the productivity of java developers and most of testing tools SoapUI,Spock etc. and development frameworks like Grails and build automation frameworks Gradle are based on groovy.
Groovy is integrated in **Jenkins** Continuous Integration server for scripting and administration of Jenkins server.
This course enables the software developer teams to understand the basics of the Groovy scripting language and its usage in Jenkins environment.

Prerequisite Knowledge

The participants must have sound Java object oriented programming skills and having exposure to scripting language is required.

Objectives

To introduce the usage of Groovy scripts in Jenkins environment.

Target population

Maximum 20 participants with every one working on separate System having the preconfigured set up as specified.

Training Methodology

The theoretical topics are discussed interactively and technical details are demonstrated with practical examples. The participants work on the hands on case studies which strengthen the concepts learned**.**
Each topic is supplemented with practical demonstrations and hands on exercises.

Class room setup

The Intel core **c**ompatible CPU with Ubuntu 18 64 bit System having minimum 8GB RAM and 500GB HDD. Other required software includes JDK1.8 64 bit latest, Google Chrome and Firefox Mozilla latest browser, Jenkins war ver. 2.249.1 distribution, zip distributions for Apache-Groovy-binary ver. 3.0.7, Apache Tomcat ver.8.0, GIT client,Eclipse-2020, Docker latest stable version, Adobe Acrobat Reader to be installed.

Live internet connection with reasonable speed and download permissions is required to download the dependencies and plug-ins.

The participants must have admin rights on their machines.

Training Duration: Three Days

Training Instructor: Prakash Badhe.

Course outline

This course plan is based on the discussions with the client team.

Day1

Groovy Introduction

- The need of Groovy
- Groovy architecture
- Groovy programming features
- The Groovy Interpreter
- Groovy Compiler
- Groovy keywords
- Groovy Console and Groovy Shell
- Groovy keywords
- Groovy Class and Script execution
- Decompose the groovy .class files
- Groovy Scaffolding on top of JVM

Groovy Usage

- Groovy shorthand and defaults
    - Assumed imports
    - Default visibility, optional semicolon
    - Optional parentheses and types
    - Optional return keyword
- Groovy variables and access control
- Dynamic typing in Groovy
- Strings in groovy
- Interpolation in groovy
- Groovy Object Oriented Programming.
- Groovy encapsulation and abstraction
- Manage Strings with groovy
- Groovy control structures
    - Groovy truth
    - The for, while loop
    - Switch statement
    - Loop control in groovy

Groovy Java interoperability

- Call groovy scripts and classes in java code
- Invoke java in groovy scripts

More with Groovy

- Groovy Functions and Closures
- Groovy collections and ranges
- Working with Lists and Maps
- Assertions with groovy
- Overview on Groovy Builder API

Overloaded Operators in Groovy

- Groovy overloaded operators and usage
- Spread and spread-dot
- Null safe dereference(safe navigation)
- Ternary and Elvis operators

Day 2

The Jenkins CI Server Introduction

- Role of Jenkins
- Jenkins Features
- Jenkins installation and configuration
- Install the required plug-ins
- Jenkins Pipeline ,Groovy, Git and Email plug-in usage
- Eclipse GIT SCM integration(gitHub)
- Jenkins GIT SCM integration(gitHub)
- Jenkins Jobs overview
- Jenkins jobs with Pipeline
- Overview on automation with apache ant

Jenkins Pipeline Jobs

- Java web application project build automation with Ant
- Create New Jenkins Pipeline job
- Declarative and Scripted Pipeline
- Use of Blue Ocean plug-in for pipeline editing
- Pipeline agents and stages in the job
- Configure groovy code in Pipeline phases with reusable blocks
- Configure to trigger the build process
- Execute the job build with Apache ant
- Configure pre-build activities
- Execute the Unit test cases/suites with JUnit
- Configure the build activities
- Execute shell scripts
- Execute the build script with Ant
- Execute the code quality measurement process.
- Configure post-build activities

- Deliver the application at deployment stage
- Execute acceptance tests
- Configure email notifications
- Generate test reports
- Execute and monitor the job in Jenkins
- Manage the logs of build Job.

More on Jenkins jobs

- Parameterized jobs
- Job dependencies on other jobs
- Analyze the job failures

Day3

Jenkins Groovy Scripting

- Infrastructure as Code for Jenkins pipeline jobs
- Execute the pipeline jobs fetched from Github
- Manage Environment Variables with groovy script
- The structure of Jenkinsfile
- Pipeline parameters and syntax
- Jenkins Groovy Console usage
- Jenkins Groovy API overview
- Execute the groovy scripts in Jenkins Groovy Console
- Execute the groovy script from shell
- Execute the groovy script remotely with Http Post method.
- Multibranch Pipeline to create different branches of the same project
- Pipeline Replay execution
- Pipeline Restart from a stage
- Restart from the Classic UI
- Restart from the Blue Ocean UI
- Preserve the stash artifacts in stages
- Define and use Shared Libraries
    - Global Shared Libraries
    - Folder-level Shared Libraries
    - Automatic Shared Libraries
- Dynamic loading of shared libraries
- Define global variables and custom steps
- Manage library versions
- Define DSL
- Use third party libraries
- Declarative Pipelines in shared libraries

Jenkins with Docker

- Build the image from Dockerfile in pipeline stage
- Manage containers in pipeline job
- The Docker-Pipeline plug-in
- Run sidecar containers in parallel
- Build the image from repository
- Publish the Docker image

****

****