

# Advanced DevOps Workshop

By Dr. Vishwanath Rao

## Day 1

### Introduction to DevOps

- Comparison -> Waterfall, Agile & DevOps methodologies
- Understanding the DevOps movement & culture
- DevOps Lifecycle – All About 'Continuous'

- Continuous Development
- Continuous Testing
- Continuous Integration
- Continuous Deployment
- Continuous Monitoring
- Continuous Feedback

- DevOps Strategy & Milestone planning – Process, people skills & tools
- DevOps Tools overview

### Source Code Version Control using Distributed Version Control Tool – Git

- Various Git Flavors : GitHub, BitBucket, GitLab etc
- Git 101 using GitHub -> Server & Client set up – Registration on GitHub, installing Git
- **Starting Out With Git**

1. Install Git on windows
2. Create a git repository
3. Perform basic git functions on the repository

### Branching, Merging and Working with Remotes

1. Clone an existing repository
2. Create a branch
3. Merge a branch onto another branch.

### Experimenting with Workflows

1. Use a centralized repository in the same way as you would with non-distributed

version control system

2. Use a "pull-request" style of development, as is common with git.

### **Using the GitFlow Workflow**

In this lab you will explore the GitFlow workflow to create features, releases, hotfixes, and various other types of branches.

## **Day 2**

- Introduction to Jenkins and Jenkins Installation (Generic war file) on every participants machine and configuring Sample free style Jobs
  - Parametrized Build, SMTP notifications, Back-up plugin Advanced Jenkins - Maven, Eclipse, JUnit & Tomcat Deployment, Master-Slave, Pipelines
  - Continuous Integration: GitHub + Jenkins Integration using web-hooks to trigger automated build based on code push demonstrated with a python application
- 
- **Continuous Integration Server – Jenkins/Hudson**
    - Jenkins pipeline with groovy
    - Integration with Sonar and other plugins
    - Jenkins Master-Slave Concepts, Set-up a Jenkins master-slave on participant's machine
    - Jenkins 2.0 features, Jenkinsfile, Pipeline Plugins for connecting Upstream/downstream

## **Day 3**

### **Docker & Containers**

- Docker Introduction, container analogy, difference between containers & VMs
- Docker Workflow
- Docker Architecture – Engine, Container, Docker host, Image, Registry

- Docker Installation
- Docker 101 commands, attaching & detaching from containers, daemon containers, starting, stopping & removing containers
- Sample exercises for running various application containers – PHP, Java, Python etc

### **Continuous on Docker**

- Dockerfile detailed syntax – FROM, COPY, ADD, ENV, USER, WORKDIR, VOLUME, EXPOSE, RUN, CMD, ENTRYPOINT, .dockerignore file
- Building & storing images, pushing them to public repositories, Docker hub registration
- Understanding Storage, Layers & inspecting docker images with exercise
- Linking Containers – Storing data, volumes, mounting volumes, read-only mounts, Docker networking, mapping & exposing ports, exercise to launch apache webserver
- Docker compose – example with installing & running a sample web-app -> Content Management Software + Database

## **Day 4**

### **Deployment Strategies in Kubernetes**

- Implement a gitops deployment pipeline using Flux
- Implement blue / green deployments using native Kubernetes
- Implement canary deployments using Istio

### **Infrastructure as Code (IaC) – AWS Orchestration using Ansible**

- Need for writing Infrastructure as Code
- Brief overview & comparison of various IaC Tools : Chef/Puppet/Ansible
- Infrastructure on Cloud & Introduction to Terraform
- Deep Dive into Ansible
  - Ansible Play & Resources, Organizing recipes with PlayBooks, Revision Control with GitHub

- Sample playbooks for setting up Apache webserver.

## **Comprehensive Exercise**

- Ansible playbook for Continuous delivery (deploy python application from GIT tag, copy to destination server, deploy with correct configurations, run tests and validations, rollback if necessary)

## **Introduction to Apache Maven**

- Build Tools for Java
- Build Tools for Java (cont'd)
- History of Build Tools
- Traditional Scripting
- 'make'
- Problems with Make
- Manual Build with JavaC
- ANT
- Pros and Cons of Ant
- Apache Maven
- Goals of Maven

Day 5

## **ELK Stack**

- Introduction to ELK
- Kibana Dashboard
- ELK Demo
- Log Shipping

## **Continuous Code Quality**

- Continuous Code Quality
- What is SonarQube
- SonarQube - Benefits
- SonarQube (Multilingual)
- Seven Axes of Quality

- Potential Bugs
- Tests
- Comments and Duplications
- Architecture and Design
- Complexity
- SonarQube Installation
- SonarQube Components
- Code Quality (LOC, Code Smells)
- Code Quality (Project Files)
- Code Quality (Code)
- Summary

## **Best Practices**

- Who are the folks using the various solutions
- DevOps Implementation Basics
- DevOps Implementation Checklist
- Lean Patterns
- Process Theory
- Tool Pattern
- Culture Alignment
- Culture Antipatterns
- Process Antipatterns
- Technology Antipatterns