Devops - for Managers and Administrators

By Dr. Vishwanath Rao

Objectives of the Course

- To understand the DevOps Concepts and DevOps Tools
- Deploying the main DevOps tools
- To implement automated system update and DevOps lifecycle
- To understand virtualization and performance
- Providing the perfect security for the entire infrastructure

Pre-requisites

- Basic knowledge of object-oriented programming
- This course does not cover Docker and Kubernetes - complete separate course

Who Should do the course

- Software Developers
- Project Managers
- IT Managers
- Development Managers
- Architects

Introduction to DevOps

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

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

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

GIT

Version Control Tool – GIT
Git Repository
* Creating a Git Repository
* Git Workflow
* Tracking File Changes
* Files or directory add to stage * Reset from stage
* Ignoring Files in Git
* Commit to Repository
* Reverting to Earlier Commits * Deleting Files in Git
GitHub – Cloud Repository
* Creating a Repository in GitHub
* Creating a Repository in GitHub Using SSH
* Pulling Commits from GitHub
* Collaborating between Local and Remote Repository * Push local Repository to GitHub or remote Repository * Merging File Changes in Git
* Issue Tracking in GitHub
Branching Merging And Rebasing in Git * Branching in Git
* Merging Branches in Git
* Fast Forward and Recursive Merge
* Recursive MergePreview
* Resolving Merge Conflicts in Git * Stashing in Git
* Rebasing in Git
* Cloning in Git

JENKINS

Installing and Running Jenkins
* Downloading and Installing Jenkins
* Running Jenkins as a Stand-Alone Application * Initial Configuration
Job Types in Jenkins
* Different types of Jenkins Items
* Configuring Source Code Management(SCM) * Working with Subversion
* Working with Git
* Storing Credentials
* Service Accounts
* Schedule Build Jobs
* Polling the SCM

* Polling vs Triggers * Maven Build Steps

Jenkins Plugins
* Jenkins Plugins - SCM
* Jenkins Plugins – Build and Test * Jenkins Plugins – Analyzers
* Jenkins for Teams
* Installing Jenkins Plugins
Distributed Builds with Jenkins * Agent Machines
* Configure Jenkins Master
* Configure Projects
* Conclusion
Continuous Delivery and the Jenkins Pipeline * Continuous Delivery
* Continuous Delivery (cont'd)
* DevOps and Continuous Delivery
* Continuous Delivery Challenges * Continuous Delivery with Jenkins * The Pipeline Plugin
* The Pipeline Plugin (cont'd)
* Defining a Pipeline
* A Pipeline Example
* Pipeline Example (cont'd)
* Parallel Execution
* Creating a Pipeline
* Invoking the Pipeline
* Conclusion

Infrastructure as Code (IaC) – AWS Orchestration using Ansible & Puppet

- • 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

Ansible – A configuration Management (Duration-9hrs)
* Introducing Ansible – A configuration management tool
* Basics / What Will Be Installed
* Understanding Ansible architecture * Control Machine Requirements
* Managed Node Requirements

* Inventory
* Hosts and Groups * Host Variables
* Group Variables
* Learn various ansible Modules * How to use adhoc commands
* Parallelism and Shell Commands * File Transfer
* Managing Packages
* Users and Groups
* Deploying From Source Control
* Managing Services
* Introduction to YAML script

* Playbook
* About Playbooks
* Playbook Language Example – YAML
* How to Write Playbooks
* Tasks in Playbooks
* Understanding about various tasks in playbook
* Introduction to Handlers and variables
* Learn about using handlers, variables in the playbook * Become (Privilege Escalation)
* Roles
* Role Directory Structure
* Using Roles
* Role Duplication and Execution * Role Default Variables
* Role Dependencies
* Role Search Path
* Including and Importing
* Includes vs. Imports
* Importing Playbooks
* Including and Importing Task Files * Including and Importing Roles
* Writing a playbook to install and configure webservers and deplo0y an application
* How to create Ansible Role and use it
* Using an ansible role in playbook

CHEF

Introduction To Chef:
- What is Chef
- Common chef Terminology

- Chef -Server
- Chef- workstation
- Chef Workstation- Looking At Security And Configs
- Chef- Repo
- Chef- Client
- Server And Nodes
- Chef configuration Concept

- Using Chef resources – the building blocks
- Building Chef recipes and cookbooks
- Introduction to testing cookbooks with Test Kitchen
- Collecting details about the system via Ohai
- Attributes – writing dynamic code
- Managing data with templates
- Advanced templating – passing in variables
- Storing your code in a repo - an introduction to Git

PUPPET

Introduction to the course.
Identify system administration functions in Puppet
Identify system administration functions in Puppet code.
Puppet architecture
Describe the Puppet architecture and describe a state model.
Implement a Puppet manifest
Build, validate, and deploy a Puppet manifest.
Troubleshoot Puppet manifests
Find documentation and diagnose errors in Puppet manifests.
Implement Git
Implement Git to manage software.
Find information with Facter
View information about systems using Facter.
Implement Puppet modules
Create Puppet modules and implement classes in a manifest.
Implement relationships in a Puppet module
Implement namespaces, relationships, and dependencies in a Puppet module.
Implement variables and conditionals in a Puppet module
Implement variables and conditionals in a Puppet module.
Identify advanced system administration functions in Puppet
Identify advanced system administration functions in Puppet code.
Implement Puppet
Deploy and configure a Puppet master and a Puppet client.