Devops (7 days)

By Dr. Vishwanath Rao

Day 1

DevOps Fundamentals
* System Development Life Cycle(SDLC)
* SDLC Models
* Agile Methodology (Backlog, Sprint, Scrum Master)
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

Day 2

Build Automation with Maven
* Installing Maven
* Understanding the lifecycle and dependencies of Maven
* Working with the Project Object Model (POM)
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

Day 3

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

Docker Containerization

Introduction
* What can you use Docker for?
* How Docker fits into the development lifecycle
* How Docker ensures consistency from development through UAT and staging, and on to production
* Example use cases of Docker in the real world

Day 4

The components of Docker
* Underlying technology
* Docker client and server
* Filesystem images
* Registries
* Containers
* Networking

Getting set up to start using Docker
* Getting set up on Windows
* Trying out our first container
* Getting set up for production on Linux
* Tweaking your production environment for best performance

Container management
* Container naming
* Starting and stopping containers
* Attaching to a container
* Seeing what is happening in a container
* Running a process inside a container
* Daemonizing a container
* Automatic container restarts
* Deleting containers when we are finished with them

Docker images and repositories
* Docker images explained
* How Docker images work
* Getting a list of images
* Searching for images on a repository
* Pulling an image
* Creating our own image
* Specify an image in a Dockerfile
* Building Dockerfile images
* Using the build cache for templating
* Viewing the image we have created

* Launching a container using our new image

Day 5

Registries
* What is the Docker hub?
* Pushing images to the Docker hub
* Running your own internal Docker registry
* Testing the internal registry

Continuous integration with Docker
* How Docker enables and supports CI
* Getting set up for Jenkins and Docker
* A basic Jenkins job
* Multi configuration jobs
* Drone
* Shippable

A more complex use case: Multi container application stacks
* A container for our NodeJS application
* A base image for our Redis containers
* Creating our Redis back-end cluster
* Capturing logs
* Managing containers

Kubernetes

1. Core Concepts of Kubernetes
2Cluster Orchestration
3Looking at K8S Origination at Google

Open Source
4.Benefits
5.Design Principles

Navigating Kubernetes Architecture
1.Master/Node
2.Kubectl
3.Replication Controller
4.Kubelet

Day 6

5.Kube-Proxy
6.Persistent Volumes
7.Etcd
8.High Availability

Using Kubernetes Features
1.Pods
2.Labels
3.Services
4.Namespaces
5.Resource Quota

Access Control
4.Policies
5.Service Accounts
6.Secrets

Networking and Kubernetes
1.Docker Networking
2.Kubernetes Networking
3.Pod to Pod
4.Exposing Services
5.IP Per Pod
6.Inter Pod Communication
7.Intra Pod Communication

Day 7

Terraform

Understanding Cloud Orchestration code and Terraform[preview]
Why Terraform? [preview]
Terraform Vs. Other Orchestration Tools - Why Terraform is the right choice [preview]
Core Terraform Components
Fundamental Concepts

Providers
Resources
Variables
Data sources
Outputs
Connecting to Cloud using Providers[preview]
Terraform Coding Examples

Provisioning resources with Terraform
Creating and accessing compute instances

Obtaining variables from Terraform into external scripts
Using shell script remote executor from Terraform to configure platform on launched VMs

Deploying Java application on VMs launched through terraform
Use of elastic ip in Terraform (or Azure Static IP if Azure is chosen)
Automation through Hosts entry modification

Using tomcat API to provision application

Understanding RDS (or Azure Database for MySQL if Azure is chosen)
Launching RDS DB (or Azure Database for MySQL if Azure is chosen) Instance using Terraform
Launching Aurora DB (or Azure CosmosDB if Azure is chosen) Instance using Terraform

Day 8

Understanding the use case for modules
Create and use a module
Module Variables

Understanding the use case for Remote State
Managing Remote State using S3 (or Azure Blob Storage if Azure if chosen)

DataSources

Templates
Jenkins Integration

Azure Devops CI CD Pipeline