DevOps Fundamentals

Day-1

- ### DevOps Introduction
  - Agile – DevOps relationship
  - DevOps case Study
  - DevOps business Drivers
  - Direct / Indirect Benefits
  - End-To-End DevOps Delivery Pipeline
  - DevOps tools Landscape
  - Linux Basics
    - Linux File system basics
    - Linux commands
    - Linux Admin Commands
  - Network fundamental concepts
  - Load Balancing
    - Methods and Strategies
    - Hardware Load Balancers, Virtual LB, Load Balancer as a Service in Cloud
  - Virtualization
    - Operating System virtualization
    - Desktop virtualization
    - Network virtualization
    - Server virtualization
    - Introduction to Vagrant and Packer.
- ### Version Control System
  - Distributed VCS
    - Introduction
    - Version control systems
    - Local, Centralized and distributed
    - Installing Git on Linux, Windows
    - Local repository Configuration.
    - Git Essentials
    - Cloning, check-in and committing
    - Fetch pull and remote
    - Branching and Merging. Merging Strategies.
    - Git Commands
    - Git remote repository application – GitHub, GitLab
- ### Jenkins – CI / CD Tool
  - CI and CD Fundamentals
    - Understand need of a CI / CD in software development lifecycle.

    - Benefits of CI / CD to business.
- Jenkins Overview
    - Understand Jenkins as a tool for automation / orchestration.
    - Jenkins installation
    - Jenkins security
    - Jenkins use and implementation in production environment.

Day-2

- Jenkins Jobs/Projects/Builds
    - Understand how to, manage and execute job / builds, and how to monitor their state. Two standard job types will be covered:
    - Freestyle Job
    - Maven build Job
    - Jobs to achieve continuous testing against your code, demonstrating the builds reporting capabilities within Jenkins,
    - Unit Testing
    - Integration Testing
- Plugin Management
    - Understand how Plugin are managed in Jenkins. Installing Plugins from command line
- Working with SCM
    - Integrate a repository browser with Jenkins and use Subversion as an example in the lab environment.
    - We will also integrate GIT for code pull and push.
- Code Quality and Code Coverage Metrics
    - Continuous quality metrics monitoring.
    - Understand how, SonarQube is integrated with Jenkins using Maven POM.xml file
- Parameterized Builds
    - The parameterized build feature in Jenkins is a key building block in creating more complex workflows in Jenkins.
    - Look at how this feature works and experiment with it in the lab.
- Trigger build / job remotely
    - Understand how to trigger a Jenkins build / job remotely, benefits of this feature.
- Automated Deployments
    - Implement continuous deployment, how to make Jenkins deploy to Java application servers, using Tomcat as an example.
- Integrate issues tracking tool, like Jira
    - Integrate Jira with Jenkins and learn how an issue can be tracked / created / updated using ‘Jira plugin’ and similar.
- Integrate Centralized Repository tool
    - How to use Artifactory as a centralized repository and integrate it with Jenkins.

    - Use Jenkins jobs to push Build artifacts to Artifactory server and pull from repository to deploy it to application server.
- Continuous Deployment
    - Integrate Ansible as Configuration management tool to perform a working Continuous Deployment in production environment.
    - Integrate Docker and use Docker Compose to automate the entire process of creating a production level container environment and deploy source code updates to containers.
- Pipeline
    - Learn how a script can be used to define a build / job in Jenkins. Define a pipeline job using Groovy scripting.
    - See how to define new Pipeline-based jobs and how to use them to achieve durability, versatility and extensibility.
    - Learn how to manage job configuration as code as well as Jenkins DSL concepts in order to help you implement development supply chains.
    - Understand how Pipeline job makes the Jenkins job definition more manageable, maintainable, and portable.

Day-3

• Ansible – Configuration Management Tool

- Ansible Introduction:
    - Describe the terminology and architecture of Ansible.
    - Understanding requirement.
- Ansible installation:
    - Install Ansible, Select platform, Check option, Understand differences.
    - Run ad hoc commands.
- What are playbooks?
    - Implement playbooks.
    - Write Ansible plays and execute a playbook.
    - Manage variables and inclusions
    - Describe variable scope and precedence; manage variables and facts in a play.
- Implement task control
    - Manage task control, handlers, and tags in Ansible playbooks.
- Using Jinja2 templates
- What are Ansible Roles?
    - Create and manage roles.
- What is Ansible vault?
    - Implement Ansible Vault
    - Manage encryption with Ansible Vault.
- Talking about Ansible in practical environment. Implement Ansible in a DevOps environment.
    - Integrate with CI / CD tools etc.

- **Docker – Containers.**
  - Introduction
    - What is a Docker
    - Use case of Docker
    - Platforms for Docker
    - Dockers vs Virtualization
    - What are Micro-serviced applications.
  - Architecture
    - Docker Architecture.
    - Important Docker components
    - Understanding the Docker components
  - Installation
    - Installing Docker on Linux.
    - Understanding Installation of Docker on Windows.
    - Some Docker commands.

Day- 4

  - Provisioning
    - Docker Hub.
    - Downloading Docker images.
    - Running Docker images
    - Running commands in container.
    - Running multiple containers.
  - Custom images
    - Creating a custom image.
    - Running a container from the custom image.
    - Publishing the custom image.
  - Docker Networking
    - Accessing containers
    - Linking containers
    - Exposing container ports
    - Container Routing
  - Docker Compose
    - Introduction to Compose-file
    - Using compose to run micro services application in container environment.
- **Kubernetes - Orchestration**
  - Containers in Production Environment
  - Kubernetes Architecture
  - Kubernetes Cluster components
    - Nodes
    - Control Plane
    - Kubelet
  - Kubernetes API objects

  - PODs
  - Replication controller.
  - Service
  - Deployment
  - Volumes
  - Namespace
  - Secrets
  - ConfigMaps
- Application deployment to Kubernetes cluster
- Cluster Management
- Introduction to RedHat Openshift

- **Introduction to Continuous Monitoring**

  - NagiosXI