Course Outline

Day 1 & 2

Module 1: DevOps Big Picture
1.1: What and why of DevOps
1.2: DevOps Tools – Overview and Use cases
1.3: Source Control Management (SCM Tools)
1.4: Continuous Integration
1.5: Static ode Analysis Tool
1.6: Storage Artifacts
1.7: Continuous Deployment
1.8: Containerization
1.9: Configuration Management

Module 2: DevOps in Action
2.1: SCM Live demo
2.1.1: GitHub – Create an account and fork your application code
2.1.2: Git clone the GitHub code, uses maven to compile and package Java source code
2.1.3: Deploy .jar file manually

2.2: Continuous Integration Tool
2.2.1: Jenkins – Deploy Jenkins on Ubuntu 14.04 server
2.2.2: Continuous Integration setup – Jenkins and GitHub

2.3: Code Packaging automation
2.3.1: Automation Maven test, Compile and Package (.jar) using Jenkins

2.4: Static Code Analysis
2.4.1: Understand SonarQube
2.4.2: Deploy and configure SonarQube
2.4.3: Students to Integrate Jenkins (CI) server with SonarQube
2.4.4: SonarQube – static code analysis and set quality gates

2.5: Storage Artifact
2.5.1: Understand Nexus
2.5.2: Deploy and configure nexus
2.5.3: Nexus storage artifact integration
2.5.4: Store your end software products in Nexus

2.6: Continuous Deployment
2.6.1: Add slave nodes to Jenkins
2.6.2: Automate deployment of your .jar file to the server
2.6.3: Building Pipeline scripts and stages in SDLC
2.6.4: Add Cucumber test- automation in pipeline
2.6.5: Add OWSAP web Application vulnerability check in the pipeline

Day 3

Module 3: Course Overview
3.1: Introduction to the course

Module 4: Understanding Docker
4.1: Module Introduction
4.2: What is Virtualization
4.3: what are Containers
4.4: Containerization and virtualization differences
4.5: Case study: 100 developer environments
4.6: Difference between a Windows and Linux containers
4.7: Docker ecosystem and components

Module 5: Installing Docker
5.1: Module Introduction
5.2: Install Docker on Centos 7

Module 4: Containers on Centos 7 Docker host
6.1: Module Introduction
6.2: Deploy, Login, exit container
6.3: List, Start, Stop, Restart containers
6.4: where containers are stored
6.5: working with container hostnames
6.6: working on multiple containers
6.7: Container stats and inspect
6.8: Container networking
6.9: Deleting containers

Module 5: Docker Images
5.1: Module Introduction
5.2: Introduction to Docker Images
5.3: Docker hub - create your account
5.4: Explore and pull images from the Docker hub
5.5: Docker commit Build and Push Your own image
5.6: Launch the container using your own image
5.7: Build Your own image using DockerFIle

Module 6: Jenkins with docker pipeline
6.1: Create a pipeline which will Dockerise the application and deploy an application on a Docker container

Day 4
Module 7: Ansible Big Picture
7.1: What and why of Ansible

7.2: Ansible use cases and terminologies
7.3: Controller server
7.4: Nodes
7.5: Playbook
7.6: Ansible tower

Module 8: Ansible management server deployment
8.1: Deploy a Centos7 server
8.2: Ssh to Centos7 server
8.3: Install and configure Ansible
8.4: Create password-less authentication keys
8.5: Define nodes to be managed by the Ansible control server

Module 9: Ansible Node server deployment
9.1: Deploy a RHEL server + 1 Ubuntu 16.04 server
9.2: Ssh to Centos7 server
9.3: Create passwordless authentication keys

Module 10: Ansible HTTPD Playbook for RHEL node
10.1: Create an HTTPd playbook directory
10.2: Write .yml file
10.3: Write HTTPd package install code for RHEL server
10.4: Write HTTPd service restart code
10.5: Write template resource type to push index.html and log.png to node
10.6: Write user and group creation code
10.7: Apply the playbook on Cento's node and validate if website is up

Module 11: Ansible Apache2 Playbook for ubuntu node
11.1: Create an Apache2 playbook directory
11.2: Write .yml file
11.3: Write Apache2 package install code for the Ubuntu server
11.4: Write Apache2 service restart code
11.5: Write template resource type to push index.html and log.png to node
11:6: Write user and group creation code
11.7: Apply the playbook on the Ubuntu node and validate if the website is up

Module 12: Windows 2016 server node deployment
12.1: Deploy a Windows 2016 server
12.2: Configure it as a Windows node under Ansible management

Module 13: Playbook for windows 2016 node
13.1: Create an IIS playbook directory
13.2: Write resource code to automate IIS role on the Windows server
13.3: Apply the playbook on the Windows node and validate if the website is up

Day 5

Module 14: Kubernetes
14.1: Kubernetes architecture overview
14.2: Deploy Kubernetes master
14.3: Deploy Kubernetes minion node 1

Module 15: Jenkins with Kubernetes Integration
15.1: Create a new pipeline script that deploys applications on Kubernetes
15.2: Configure a Docker slave node with Jenkins
15.3: Fork source code
15.4: Run a job that uses Docker, SonarQube, and nexus by Jenkins CI/CD pipeline

Module 16: Jenkins with Ansible integration
16.1: Create Jenkins Job to compile package a Java web app file
16.2: Define Ansible nodes
16.3: Integrate Ansible in the Jenkins pipeline to deploy a web app to Ubuntu servers