Devops Engineer Program with Project Implementation

By the end of this course, you will have developed a strong foundation in DevOps principles and gained hands-on experience by working on real-time projects.

This will be completely Project based learning. By the projects we are implementing, we will be having an experience of (3-4 years to claim minimum).

You will be confident enough to apply the skills learned and clear any DevOps Interview and sustain as a Devops Engineer.

The Main Aim of the Course is to have a real time experience while learning each tool and integrating with various Devops Tools with CICD implementation.

Duration: 4 Months (Total 160+ hours)
"Timings":
Monday to Friday: 7.10 AM to 8.10 AM IST (Devops Tools)
Saturday and Sunday: 8.15-10.15 AM IST (Docker and Kubernetes with CKA and CKAD)

9381062032

i27
Academy

1. Devops Overview:

- What is Devops ?
- How do we start our project and what teams are involved?
- What problems do Devops Solve?

2. Cloud Foundation:

- What is Cloud, Cloud computing
- Various Clouds
- What is Infrastructure?
- Account Setup in AWS and GCP

3. Linux

- Overview of UNIX/Linux Operating
- Linux File System
- ls,date and cal commands
- Working with Directories
- Working with Files
- Comparing Files
- Creation of Link Files
- Word count command (wc command)
- Sorting content of the file
- Find unique content in the file by using uniq command
- Input and Output of Commands and Redirection
- Piping
- How to use multiple commands in a single line
- Regular Expressions and Wildcard Characters
- Command Aliasing
- Locate and Find Commands
- Compression and Uncompression of files(tar,gzip,gunzip,bzip2,bunzip2)
- grep command
- Cut, Paste and tr Commands
- File Permissions
- SSH keys configuration
- Working with editors

4. Version Control with Git

- What is Version Control ?
- What is Git and GitHub
- Basic concepts of Git
- Setup Git repository Remote and Local
- Initialising a Git Project locally and connecting
- Git commands handson
- Working with branches and merge branches
- Rebase
- Undoing commits
- What are merge conflicts and how to resolve merge conflicts in real time .
- Going back in history
- What are tags , how and where we use in real world
- Git Reset
- Git Merge
- Git Rebase
- Git checkouts
- Git Administration tasks

5. Maven:

- What is a Build tool ?
- Various Build tools ?
- What is MAVEN ?
- Why do Devops engineers need to use Maven
- Installing maven and configuring Java on Linux machine
- Maven Repository.
- Project Structure of Maven
- Creating a sample project from scratch using Maven.
- Maven Lifecycles and Goals
- Maven Dependency Management
- Configuring how to use private/company dependencies.
- Integration with Jenkins

6. Nexus

- What is an artifactory ?
- What is Nexus?
- Installing nexus on a Linux VM
- Nexus Overview
- Nexus Installation and Configuration
- Repository creation.
- Snapshot management

- Release management
- Nexus repository configuration
- Artifacts Management.
- Uploading the artifacts to Nexus
- Downloading the artifacts in the CD process
- Nexus repo management
- Managing of Versions

7. Sonarqube:

- Use case of Sonarqube
- What is Sonarqube ??
- Understanding the Sonarqube Architecture
- Installing sonarqube on linux machine
- Configuring projects to run on Sonarqube
- Creating QualityProfiles, quality gates for the applications
- Performing scans based on the code quality
- Admin activities.
- Users Creation in sonar
- Scanning a microservice with Maven build tool.
- Scanning a microservice with Gradle build tool
- Integrating SonarQUbe with Jenkins
- Integration with Jenkins

8. Jfrog

- Jfrog Artifactory OVerview .
- Installing Jfrog Artifactory in Linux.
- Integrating the Jfrog in Maven.
- Managing/Creating of Users and groups in Jfrog Artifactory
- Integrating Jfrog with Jenkins

9. Jenkins:

- Introduction:
  - What is Continuous Integration
  - What is Continuous Delivery
  - What is Continuous Deployment ?
- What is Jenkins ?
- Installation:
  - Installing in Redhat/Centos
  - Installing in ubuntu

- Jenkins Directory Structure
- Jenkins Graphical User Interface

- Exploring the GUI
- System Configuration
- Job Configuration
- Create a CICD for JAVA App using Maven
- Jenkins Plugins:
  - Managing Plugins
  - Updating Plugins
  - Adding Plugins
- Freestyle Jobs:
  - Various Job Structures
  - Parameters
  - Notifications
- Agent and Distributed Builds:
  - Why do we need Agents ?
  - Setting up a Build Agent .
  - Distributing a build on various agents
  - Healthcheck of Build Agents.
- UpStream, Downstream and Triggers
  - What are Upstream and Downstream jobs, and how to trigger
  - Artefacts and fingerprints
- Security:
  - Understanding on Authentication and Authorization part .
  - OVerview of Jenkins Security
  - Matrix Based Security
  - Managing Folders
  - Creating Users, Roles and Groups.
  - Deep dive into Credentials
- Update and Maintenance:
  - Various ways to upgrade jenkins
  - Backing up Jenkins
- Notifications and Alerts:
  - How to Setup Email Notifications.
  - Various Notifications Plugins
- What is a Pipeline :
  - Various Pipeline Jobs
  - The Blue Ocean Editor
- Scripted vs Declarative Pipeline
  - Creating Scripted Pipelines
  - Creating Declarative Pipelines
  - Understanding Groovy and DSL
- **Deep diving Jenkins Declarative Pipeline**
- **Implementing Production Grade Pipelines used in Large Product based Companies**
- JenkinsShared Libraries
- Exploring options under Mange Jenkins
- Pipeline Triggers
- Multi Branch Pipelines Implementation.

- ○ Implementing Master Slave architecture.
- ○ Shared Libraries vs Global Libraries

10. Docker

- ● Introduction to Docker:
  - ○ Monolithic Application Overview
  - ○ Microservices Application Overview.
  - ○ Introduction to Containers
  - ○ Understanding Docker Architecture , Docker Engine
  - ○ Various types of installations available for Docker.
- ● Docker Containers:
  - ○ Learning Docker Basic Commands:
  - ○ Container LifeCycle.
  - ○ Docker Port Binding.
  - ○ Docker commands
  - ○ VM vs Container
- ● Docker Networking:
  - ○ Overview of Docker Networking.
  - ○ Bridge networking
  - ○ Implementing user-defined bridge networking.
  - ○ Host networking.
  - ○ Networking Commands and implementing with containers.
- ● Docker Storage:
  - ○ What is Persistency ? how to achieve it in containers ??
  - ○ Overview of Storage Drivers.
  - ○ Understanding Bind Mount and TMPFS Mount.
  - ○ Overview of Docker Volumes.
  - ○ Using Volumes for Persistent Storage.
- ● Docker Image Management and Registry:
  - ○ What is an Image ?
  - ○ Inspecting and Docker image
  - ○ How to Create and Pull/Push an Image.
  - ○ What are Image tags ? How do we implement it in the real world?
  - ○ Image Layering.
  - ○ Committing Changes.
  - ○ Deploying and Configuring Private Registry.
  - ○ Using Multi Stage Builds.
- ● Building Image with Dockerfile
  - ○ Dockerfile Introduction
  - ○ Working with dockerfile instructions.
- ● Integration with Jenkins

11. Ansible

- ○ What is Configuration Management?
- ○ What is Ansible , how does it work ?

- Installing Ansible and Configuring it.
- Push vs Pull architecture
- Adding nodes to the controller after configuration .
- Creating Playbooks.
- Working with Ansible Components
- Inventory
- Managing host key checking and ssh keys
- Ansible Modules
- Ansible Collections
- Ansible Variables
- **Interview Questions Discussion**

12. Kubernetes Advanced(CKA and CKAD)

- ### Introduction to Kubernetes
  - History of Kubernetes.
  - Need for Orchestration and Management tool
  - Introduction to Kubernetes.

- ### Kubernetes Architecture:
  - Kubernetes Components
  - Kubernetes Master and Worker Node Architecture.
  - Working with Kubernetes:
    - Kube API Server
    - ETCD
    - Kube Controller Manager
    - Kube Scheduler
    - Kubelet
    - Kube Proxy

- ### BootStrap a Kubernetes Cluster on Cloud
  - Creating Infra ready on the cloud.
  - Bootstrapping Master and Worker nodes using KUBEADM and managed cluster

- ### Working with Kubernetes POD's and Deployments :
  - Overview on Pod , Container Vs Pod ???
  - What are Namespaces, and how would they be in real time projects ?
  - Understanding YAML.
  - Creation of Pod's through Imperative and Declarative.
  - What are Labels, Selectors , Annotations and their use in K8S.
  - Disadvantage of Pods and introduction to Deployments.
  - Difference between ReplicaSets and Replication Controllers.
  - Deep Dive into Deployments.
  - Scaling Application with RC, RS , and Deployments.
  - Deployment Strategies in Kubernetes, which are used in Production Deployments.
  - Deploying with Blue/Green and Canary strategies
  - Rolling updates and Rollbacks.
  - What are Jobs and Cronjobs
  - DaemonSets

- **Working with Services :**
  - What are Services in Kubernetes, and what's the importance of using it ???
  - Deepdive into K8S Services.
  - Creating ClusterIP, NodePort, LoadBalancer and Headless Service.
  - What's an Ingress Controller and how to create them.

- **Observability:**
  - Implementing Liveness and Readiness Probes.
  - Understanding the real use case of the probes.
  - Installing Metrics Server in Kubernetes.
  - Horizontal Pod Autoscaler.
  - Automated Scaling of Application with HPA and Metric Server.
  - Logging, Monitoring Applications and debugging.

- **Advanced Kubernetes Scheduling:**
  - How scheduling works in Kubernetes
  - Manually scheduling pods on different nodes.
  - Advanced Pod Scheduling with Node Affinity and Anti Affinity
  - Understanding Node Taints and Pod Tolerations.
  - Understanding INIT-Containers.
  - Understanding DaemonSets.
  - Static Pods
  - Working with Resource Limits in Kubernetes.

- **Kubernetes Resources, QoS, and Namespace Quota**
  - Resource Requests and Limits for CPU and Memory.
  - Quality of Service (QoS) Classes
  - Namespace Resource Quota:

- **Networking In Kubernetes:**
  - Kubernetes Networking Overview.
  - CNI in Kubernetes
  - Understanding DNS in Kubernetes
  - Ingress Networking.
  - Validating and Mutating Admission Controllers
  - Custom Resource Definitions
  - Api Deprecation

- **Storage In Kubernetes:**
  - Why Learn Kubernetes Storage
  - Introduction to Storage in Kubernetes
  - Difference between ephemeral and persistent storage.
  - Volumes: Concepts and Types
  - Persistent Volumes (PV) and Persistent Volume Claims (PVC)
  - Dynamic Volume Provisioning
  - Storage Classes and Provisioners
  - Volume Access Modes
  - Managing Storage Resources

- **ConfigMaps , Secrets and Environment Variables in Kubernetes:**
  - ConfigMaps in Kubernetes and UseCases in realtime
  - Creating and Using ConfigMaps

- Mounting ConfigMaps as Volumes
- What is a Secret?
- Difference Between ConfigMaps and Secrets
- Creating and Using Secrets
- Environment Variables in Kubernetes
- What are Environment Variables?
- Setting Environment Variables in Pods
- Environment Variables from ConfigMaps and Secrets

- **Role-Based Access Control (RBAC) in Kubernetes Network Policies in Kubernetes**
  - What is RBAC? Use case of RBAC ?
  - Authenticating Mechanisms ?
  - Understanding Various TLS certificates for Cluster Components.
  - What is Kubeconfig and its Structure ?
  - Managing Multiple Kubernetes Clusters and switching between clusters
  - Roles and Role Bindings
  - ClusterRoles and ClusterRole Bindings
  - Service Accounts in Kubernetes

- **Network Policies in Kubernetes Cluster Management:**
  - What is a Network Policy?
  - Components of a Network Policy
  - Creating and Applying Network Policies
  - Real Time use cases

- **Cluster Management:Managed Kubernetes Service:**
  - Introduction to High Availability in Kubernetes.
  - Working with OS upgrades.
  - Draining a node safely during maintenance.
  - Upgrading Kubernetes Cluster.
  - Backing Up and Restoring ETCD.

- **Managed Kubernetes Service:**
  - Creating a High available cluster in GKE
  - Creating various types of clusters based on requirement.
  - Dynamic volume allocation using SC
  - Zero downtime upgrades of cluster
  - Private container registry(GCR)

- **Multi Container Pods**
  - What are Multi Container pods and use cases
  - Init containers
  - Static Pod

- **Ingress Controller:**
  - What is an Ingress Controller?
  - What is an Ingress Controller?

- **Troubleshooting in Kubernetes**
  - Checking Cluster and Node logs
  - Troubleshooting a broken cluster.
  - Troubleshooting broken applications.
  - Various issue we get in deploying an application

- **Helm Fundamentals:**
  - Helm Introduction
  - Install helm
  - Create Helm Charts

- **LENS Ide**
- **CKA Preperation BootCamp**

13. **Python:**
   - Module 1: Python Basics
     - Introduction to Python and its relevance in DevOps
     - Installing Python and setting up the environment
     - Running Python: Interactive shell vs script files
     - Variables, Data Types (`int`, `str`, `bool`, `float`)
     - Basic I/O: `print()`, `input()`
     - Type conversion and string formatting

   - Module 2: Control Structures
     - Conditional statements: `if`, `else`, `elif`
     - Loops:
     - For , while loop
     - break, continue, pass

   - Module 3: Functions & Command-line Inputs
     - Defining and calling functions
     - Parameters and return values
     - Built-in vs user-defined functions
     - Using `sys.argv` for CLI argument parsing
     - Real-time use cases:
     - Reusable health-check functions
     - Accept arguments for backup/restore operations

   - Module 4: File Handling
     - Opening and closing files
     - Reading (`read()`, `readlines()`) and writing (`write()`, `writelines()`)
     - With statement for safe file handling

   - Module 5: OS & Subprocess Automation
     - os module:
       - File/folder operations (`mkdir`, `listdir`, `remove`)
       - Environment variables
     - subprocess module:
       - Running shell commands
       - Capturing command output

- Module 6: Working with Data Formats
  - JSON:
    - Reading and writing JSON data (`json.load()`, `json.dump()`)
  - YAML:
    - Using `pyyaml` to load and dump YAML

- Module 7: HTTP & REST APIs
  - Introduction to `requests` module
  - HTTP methods: GET, POST, PUT, DELETE
  - Authentication and headers
  - Handling JSON response

- Module 8: Error Handling & Logging
  - Exceptions: `try`, `except`, `finally`, `raise`
  - Logging with `logging` module
  - Debug-level, Info-level logs

- Module 9: Modules & Packages
  - Importing built-in modules
  - Creating custom modules
  - `pip` for installing external packages
  - Virtual environments

Real Time Devops Projects with Microservices Architecture.

- ○ Super Excited!!!!!!!!!!!!!!!!!!
- ○ This will be one of its kind implementation, where we will be working on Projects from the Scratch
- ○ By Implementing this Complete Project, one can confidently clear any Interview and work in any company too, as we are implementing in a Production grade with Best Practices.
- ○ This will be End to End Workflow , where we will be creating a company Scenario.
- ○ Right from Gathering the requirements of the business.
- ○ Creating infrastructure for the projects we deploy.
- ○ Implement a CICD pipeline with all the tools we have learned.
- ○ Deploying the application to the production environment.
- ○ Creating jenkins pipelines from scratch.
- ○ This project consists of a complete CICD which we generally do with production grade applications.

Here are the Modules/Project we will be Implementing

- ● Prerequisite:
    - ○ Understanding the Company Scenario
    - ○ Getting Familiar with Product Architecture.
    - ○ Microservices Functional architecture along with Infrastructure Architecture.
    - ○ Jenkins Pipeline Architecture
    - ○ Deploying Planning for our Product in a Production Grade Setup

- ● Module 1:
    - ○ Setting up the infrastructure based on the requirement

- ● Module 2:
    - ○ Once Infra is ready we will be deploying all our configurations using Ansible, on the above Created Servers , we shall be writing many playbooks for this implementation with best practises.

- ● Module 3:
    - ○ Implementing Jenkins Master and Jenkins Slave setup for our CICD Implementations.
    - ○ We shall be using both Static and Dynamic as well for these implementations to achieve advanced scenarios.

- ● Module 4
    - ○ All hands on deck !!!!!!!!
    - ○ Creating a Jenkins Pipelines for all our Microservices
    - ○ This will be a Complete Production Grade Pipeline with many reusable methods written in Groovy
    - ○ Module 4a:
        1. Github Organisation Setup
        2. Implementing Multi Branch Pipelines
        3. Jenkins with Build Integration , and creating methods
        4. Jenkins with Sonar Integration with Quality gates implementation
        5. Jenkins with Nexus integration for all artifacts

6. Jenkins with Docker Integration for all our Microservices image creations using Dockerfile
7. Jenkins with Docker Deployments on different environments with prod approval steps and production grade deployment pipeline

- Module 5:
  - Create Dockerfiles and Jenkins Pipelines for our frontend based application.
  - Modifying the Source code and deploying the application in different environments
  - Create a web server with reverse proxy configurations to access our application

- Module 6:
  - Implementing Kubernetes in our Pipeline.
  - Creating Kubernetes Clusters.
  - Configuring kubernetes prerequisite on the Jenkins Slave using Ansible
  - Creating Manifest files for all our Microservices and Frontend applications to deploy in Kubernetes
  - Modifying the existing pipelines to add the Kubernetes Deployment stage

- Module 7
  - Creating Shared Libraries in jenkins for all our applications.
  - We will recreate all the above functionalities into shared libraries
  - This will be helping us to reuse the same methods and pipelines across the Organization (Super Cool and advanced!!!!!)

- Module 8
  - Implementing Helm Charts in our Pipeline
  - Creating Charts with environment based values and integrating with Shared Libraries for Our Deployment

- Module 9
  - Advanced Kubernetes Admin operations with Jenkins pipelines
  - We will Create multiple jobs to make sure we are giving developers an easy deployment for all their activities without us being involved in all stages.

14. **Devops Engineer Roles and Responsibilities**
15. **Day to Day Activities of a Devops Engineer**
16. **Devops Engineer Resume building**
17. **Tips to Crack Devops Interview**

Contact Details

Siva

Phone Number: +91 9381062032

Mail : devopswithcloud@gmail.com

Youtube Channel: https://www.youtube.com/@i27academy