[Basic + Advanced - Complete set of DevOps tools - 100% Job Guaranteed]

- By Karthik M(8 years Realtime EXP on Devops)

Devops Introduction:

1. What are the software development models available ?
2. Why Devops ?
3. When to use and when not to use?
4. What are the delivery challenges faced?
5. Types of Source code management?
6. Tools Covered under this course
7. Bonus – Introduction on Kubernetes.

Tools Covered: **[Basic + Advanced - Complete set of DevOps tools - 100% Job Guaranteed]**

1. SCM - **Git**
2. Build Tool - **Maven**
3. Continuous Integration - **Jenkins**
4. Containerization - **Docker**
5. Configuration management System - **Chef**
6. Cloud - **AWS - EC2, S3, VPC, IAM, Route53, Queue Services, Auto scaling** - Advanced explanation on each options on these services.
7. Static code analysis - **Sonarqube**
8. Code Coverage - **Jacoco**
9. Artifactory - **Jfrog**
10. Introduction and architecture of **Container Orchestration - Kubernetes**
11. **Linux basic and Scripting** Introduction.
12. Introduction to **Google Cloud Platform.**
13. Introduction to **Ruby and Groovy Scripts.**

Note: All the above listed software's are open sources. And Practice is done on **Google Cloud Platform.**

GIT:

- **Day 1**:

  1. What is VCS?
  2. Detailed Git Architecture
  3. Git Installation
  4. Github Account creation.
  5. Setting up remote repository

- **Day 2**:

  6. Git initial Configurations.
  7. Create Central/remote Repository locally
  8. Deep dive – History of verifications of commits
  9. Git Basic Commands.

- **Day 3**:

  10. Branching model/Strategy
  11. Git advanced Commands
  12. Git diff
  13. Git show
  14. Git push
  15. Git checkout
  16. Git reset -And its types
  17. Git rebase
  18. Git revert
  19. Git stash
  20. Ignoring files
  21. Git clean

22. Git log

23. Git tag

- **Day 4** :

24. Merging

25. Resolving merge Conflict

26. Hooks

27. Git Pull

28. Git Cherrypick

29. Git Fetch

30. Differences between GIT and SVN

Maven: Build Tool

- **Day 1:**

1. What is Build Management ?

2. What is Maven?

3. Why build tool is required ?

4. Maven Architecture?

5. Maven Repositories?

6. Maven Installations

- **Day 2:**

1. Project Name(GAV) and why they are important?

2. Generate sample project structure

3. Maven Build Life cycle

4. Examples on Maven goals

5. Verify Built artifacts

- **Day 3:**

  1. What is POM?
  2. Maven Plugin management
  3. Different ways of invoking plugins
  4. Different command line options
  5. What is the importance of Settings.xml?

- **Day 4 :**

  1. Deployment Automation, Dependency declaration,
  2. Multi Module Projects
  3. Maven update version – for release
  4. Documentation – Building own site
  5. Software quality
  6. Build Types
  7. Real-time project deployment
  8. Code Coverage
  9. Sonarqube – Code quality tool

Jenkins: CICD Tool

- **Day 1:**

  - What is CI and its benefits ?
  - Why Jenkins?
  - Crontab Syntax
  - Jenkins Architecture
  - Jenkins Installation

- **Day 2:**
    - **Configure Systems**
    - Configure Global Security
    - Global Tool Configuration
    - Reload Configurations from disk
    - Manage Plugins
    - System Information
    - System log
    - Load Statistics
    - Manage Nodes
    - Manage Users
    - Prepare to Shutdown
    - How to change port of Jenkins
    - How to change home directory of Jenkins
    - How to migrate Jenkins form one server to another
    - Real-time scenarios
    - Jenkins folder structure
- **Day 3:**
    - Why Job Configuration is required?
    - Job Creation in Jenkins
    - CI setup – Exercise
    - Downstream dependency
    - CI and CD
    - Alter/create View – Project specific
    - Real-time scenarios

    - Build Pipeline

- **Day 4:**
    - Jfrog –Artifactory
    - Sonarqube – Static code quality analyzer
    - Jacoco – Code coverage
    - Real time project - Integrate Jfrog, Sonarqube, JaCoCo and Docker in Jenkins.
    - Maintenance of jenkins
    - Jenkins Best Practices
    - Popular Plugins
    - Code Review – An Agile process
    - Defect Tracking

Docker: Containerization

- **Day 1**
    - Introduction to Docker
    - Difference between Physical and Virtual server
    - Docker supported platforms
    - Installations and verify the Docker.
- **Day 2**
    - Managing Docker Containers.
    - Docker run command and understanding the entire command to create a container.
    - Inspecting Containers and Various Commands.
    - List running containers only
    - Show the last container which you have created(stopped/running)
    - List all containers(stopped and running)
    - Naming the container

- Rename a container
- Deleting a container
- Delete all containers at once.
- Starting a stopped container
- Shortcut Keys - to work with Container
- Attaching to a running container
- Inspecting the container's processes
- Stoping a container from 'host machine'
- Show last 4 containers (stopped/running)
- Find More About The Container
- Create demonized container
- Remove all running containers
- Remove all running/stopped containers

- **Day 3**
  - Deep Dive into Docker Images.
  - Listing docker images
  - SETTING-UP NGINX SERVER ON UBUNTU MANUALLY: Project
  - Images types
  - Creating docker image using "docker build" command
  - Writing Dockerfile.
  - Building docker image
  - Listing docker image
  - Testing Image
  - Data Volumes - Advanced Topic

- **Day 4**
  - Exec - command
  - Docker useful commands
  - Build image without using existing image/image layers
  - To copy a file to docker container from host machine
  - Docker logs
  - Docker Hub and real-time Project.
  - Working with docker-hub images
  - Deleting all Images
  - Deleting an Image
  - Pushing custom images to docker repository
  - Searching docker images in docker hub
  - Pulling the images
  - Docker Benifits
  - Real time project

Chef: Configuration management tool

- **Day 1**
  - Why Chef is required?
  - What are Chef features ?
  - Chef Architecture
  - Chef environment – terminologies
  - Hosted Chef server setup
  - AWS Instances – To setup node and WS – Points to Note

- **Day 2**
  - Workstation Setup - ChefDK Installation
  - Connect WS to Server
  - Setup Node And connect to Hosted chef
  - Chef Recipe Syntax
  - Simplest form of Chef recipe code
  - What are Resources
  - How to do roll back in Chef
- **Day 3**
  - Commonly used Chef resources
  - Commands to generate cookbook and Recipe
  - Test your recipe locally before applying to PROD
  - Real-time Exercise – Write a recipe and upload to server to configure node
  - Chef Recipe – File resource
  - Chef Recipe – Array
  - Setting up Firewall -Multiple ports
  - Chef Recipe – User resource
  - Chef Recipe – Directory resource
  - Chef Recipe – Remote file resource
  - Chef Recipe – Execute resource
  - Chef Recipe – Template resource
  - Real-time –requirement to install apache service
  - Ruby has hash
  - Login to chef server – to check the hash of node
  - Adding the recipe to run_list to execute on nodes

- **Day 4**
  - Optimize chef recipe – using attributes
  - Real-time scenario – tomcat install
  - Optimize chef recipe – using attributes
  - Attributes precedence - Chart
  - How does chef-client works
  - How to stop executing the recipe in other platforms
  - Chef Sample project – In Github
  - Chef – Best practices
  - Community cookbooks - Supermarket
  - ROLES
  - Creating Roles
  - Exercise – on roles
  - CLEANUP
  - Chef - Environment
  - Databag - concept