

DevOps Course Content

Welcome Note:

DevOps and Cloud are all about the infrastructure and majority of infra will be related to Linux. DevOps without Linux and Cloud concepts is very minimal and narrow. So, keeping that in mind, our training will start with Linux basis, AWS Cloud, Operations, infrastructure, and then we will jump in to the DevOps and that's how it makes you comfortable throughout our course. And the course is completely designed on a real time use cases and with a project oriented approach.

Getting Started:

- ➔ AWS Account Creation
- ➔ Installing basic requirements on your workstation
- ➔ Understanding the Cloud Environment
- ➔ Why Cloud?
- ➔ What's IAAS, PAAS, and SAAS?
- ➔ What all we do and the services we consume.
- ➔ Explaining why Linux.
- ➔ Flavors and the Modules in Linux.
- ➔ Linux EC2 Instance creation and demonstration the password less authentication
- ➔ Exploring the linux file system.
- ➔ Linux CLI and with lot of commands
- ➔ Permissions, Access in Linux.
- ➔ Security and Vulnerability concepts on Linux
- ➔ User and Network Management
- ➔ Shared File System
- ➔ Various Shells
- ➔ Troubleshooting the issues in Linux
- ➔ Module installations and defining your own linux set up.
- ➔ Package Installation using the Yum and RPM
- ➔ Source Code Installation.
- ➔ SE Linux Concept.

Application Workflow:

- ➔ Demonstrating the applications and modules involved in it.
- ➔ Latest application vs Obsolete application and the various components
- ➔ How to define and integrate the WebServer, AppServer and projecting your infra using proxy (NGINX / Apache Firewall)
- ➔ Implementing the Proxy ,Application and WebServer on the top of Linux EC2
- ➔ Defining the security on VPC using the Security Group and the NACL and the security layers
- ➔ Implementing the application on EC2 and EBS.
- ➔ Discussing the basics in Linux Shell Scripting
 - Editing, Printing and the redirections
 - Variables and the use cases
 - Functions
 - Input and Output Redirection
 - Loops
 - Conditions
 - Security Concepts
 - Password Less Authentication

GIT: (SCM)

- ➔ What is Git and why you need Git?
- ➔ DSVN vs SCM
- ➔ What is SVN and what is versioning?
- ➔ Application code versioning.
- ➔ Branching
- ➔ Branch and Deployment Strategies.
- ➔ Architecture of GIT
- ➔ Fetch, check in, check out, and revert, cherry-pick mechanism.
- ➔ How to revert back in case of corrupted code using Git and the issues over that.
- ➔ User Account Creation on GitLab / GitHub /bit Bucket.
- ➔ Branching Strategies in Git and the approval mechanism.
- ➔ Git Access Provision and Approval Mechanism.
- ➔ GIT Security.

[ANSIBLE: \(Configuration Management Tool\)](#)

- Understanding the ansible and various ways to use the Ansible
- Ansible Alternatives.
- Ansible Direct Modules
- Automating Manual tasks using the Ansible
- Writing various playbooks to do the configuration management.
- Deploying the applications using playbooks
- Playbook Security over Vault
- Application deployment using PUSH
- Ansible Pull Mechanism.
- Multi-Stage and Multi-Project Strategies
- Galaxy Modules in Ansible.
- Security and Vulnerability Setup.

[TERRAFORM: \(IAAC: Infrastructure As Code Tool\)](#)

- What and Why Terraform
- Various Cloud Providers
- Terraform Installation
- EC2 Creations
- Understanding and provisioning the AWS Cloud Infra using the Terraform Code
- Network and Security deployment over code.
- Variables , Loops in Terraform
- State file Approach.
- Remote State file approach Over S3.
- Interpolations , Conditions
- Data Sources and the prominence
- Deploying the code along with the infra creation using the Terraform
- Various Strategies on Local-Exec, Remote-Exec,
- Use cases on File-Provisioners, Local-files
- Terraform Console and Graphs in Terraform
- Methodology to update obsolete machines.
- Output's and Modules
- Best practices in modules
- Various Deployment Strategies by terraform.
- Integrating Ansible + Terraform

JIRA (Project Tracking & Agile)

- What is JIRA
- Sprint Cycle
- Work Load Assignment
- Git Integration with GIRA
- SPRINT and Retro Discussion
- Project Collaboration with JIRA.
- Tracking Points
- Bit Bucket
- JIRA + Bit Bucket flow
- KANBAN Usage.
- Approvals and Branching over JIRA
- Bug and release strategy.

JENKINS (CICD Framework Integration Utility)

- What is Jenkins?
- Projects and Tools supported by Jenkins.
- Modules and the executors in Jenkins
- Exploring all the major options in Jenkins.
- Security In Jenkins
- Achieving the Multi Stage Environment
- Deploying Master Slave Jobs
- Parallel Job Processing
- Jenkins Failover.
- Securing Jenkins Jobs in case of failure
- Deploying Multiple Environments using the Ansible, Terraform over the Jenkins
- What is Agile Workflow
- Waterfall model vs Sprint Approach.
- Understanding the CICD process and use cases of CICD.
- Introduction to PIPELINE
- Job creation using multiple available approaches.
- Creating various Pipeline Jobs to fulfil the complete CICD Pipeline using the Declarative approach and Groovy Approach

The below are CICD automations efforts that we will achieve with Jenkins Declarative Pipeline:

- Code Commit Using the GitHub/GitLab
- Access Control and Authentication Modules.
- Incorporating branches and approvals in pipeline.
- Maven Phases & goals.
- Incorporating Maven Wrapper jobs.
- Code Compile using the Maven / Graddle
- Deploying and configuring the Pylint/SoanrQube.
- Code Quality, Analisys and vulnerability check using the SoanrQube / Pylint.
- Code Quality Check and decision making.
- Deploying the TestCafe / Selenium for UI Testing.
- Deploying Artifacts AWS S3.
- Mitigating the authentication modules.
Authentication Modules over the vulnerabilities.
- Deploying and configuring the Nexus.
- Vaulting the artifacts in NEXUS for the prod module
- Employing the HARNESS Delegate the automate the CICD
- Incorporating the centralized the monitoring and roll back of CICD using the Harness with security and Machine Learning.
- With Harness we can build the pipeline in minutes.

Docker: (Container Runtime)

- ➔ What is docker?
- ➔ Various Container Technologies
- ➔ Docker Hub Registry Process.
- ➔ Virtualization vs Containerization
- ➔ Docker Installation
- ➔ Deploying Docker Containers
- ➔ Persistent Volume
- ➔ Port forwarding and security
- ➔ Making Docker Images using Dockerfile
- ➔ Customizing the Dockerfile.
- ➔ Securing our Dockerfile in registry.
- ➔ Image Size Reduction Techniques.
- ➔ Multi Stage Docker Imaging
- ➔ Port Masking

Kubernetes: (Container Orchestrator)

- Why Kubernetes and Architecture
- Installing and configuring the cluster.
- Cluster using the KubeAdm.
- Defining the cluster using GKE /KOPS.
- Kubernetes Cluster on Google Cloud
- Virtual Network FlannelD
- Adding nodes to the cluster
- Deploying and deleting the cluster
- Listing and filtering the nodes
- Pods & Affinity and Non Affinity Policies.
- Replica sets / Daemons Sets
- Taints and Tolerate
- Multi Container Pod
- Container Log Analysis
- Service Deployment in Pod
- Listing and queries on pods
- Network Limitations and Best Practices

AWS Cloud Concepts:

- On Prem Infra vs AWS Cloud VPC
- Network Security Layer
- NAT, IGW and NACL Integration.
- EC2 Deployment both Linux & Linux
- Connectivity over Keys
- RDS and DB Deployment
- S3 and RR Secure Storage Solution
- EBS Volumes
- Storage Lifecycle Methodologies
- EFS (NFS Solution)
- Backup Solution GLACIER
- IAM Roles and access provision
- AWS Programmatic Setup
- Website Deployment using the S3
- Elastic Bean Stack
- AWS Route53
- AWS CloudWatch
- SNS Alerts and ALARM Deployment

