



# FULLSTACK DEVOPS & CLOUD ENGINEER

**ACCELERATE YOUR CAREER GEAR TO EXPERIENCE, EXPLORE AND EXCEL THE CUTTING EDGE TECHNOLOGIES OF DEV-OPS, CLOUD AND BIG-DATA IMPLEMENTATION.....**

**Just a Click Ahead to Know More About Us**

**Why You have To Learn from Inceptez**

- <https://rb.gy/5tfvey>

**FAQ about the Course**

- <http://Inceptez.in -> more -> Frequently Asked Questions>
- <https://rb.gy/lkw1lm>

**Our Other Value Added Services**

- <http://Inceptez.in -> more -> Inceptez Interview & Job Support>
- <https://rb.gy/jm7vdw>

**ENQUIRE, ANALYSE, COMPARE AND FINALLY INVEST IN YOUR CAREER ACCEPTING INCEPTEZ AS YOUR CAREER PARTNER TO EXCEPTIONALLY FEEL THE WORLD CLASS TRAINING....**

# ABOUT INCEPTEZ

Incepetz Technologies was founded by a team of Technology Evangelists in 2014 and is one of the leading IT training, Development and staffing company specializing in Big Data, Data Science, Dev-Ops, Cloud Computing and Internet of things (IOT). Incepetz is a non money oriented training center, where we first prioritize Comprehensiveness, Engagement based, Focus based, Competitive model with high Quantity and Quality in all the training as a paramount.

Incepetz Technologies is mastered and administrated by highly skilled industry experts. We are the technology enablers committed to provide comprehensive training to the aspiring professionals in the game changing, high demanding applications such as Hadoop, Spark , Data Analytics, Data Science, DEV-OPS and AWS/Google Cloud Platforms, that are the fastest growing trend setting technologies that provide competitive advantage in the ever changing IT world.



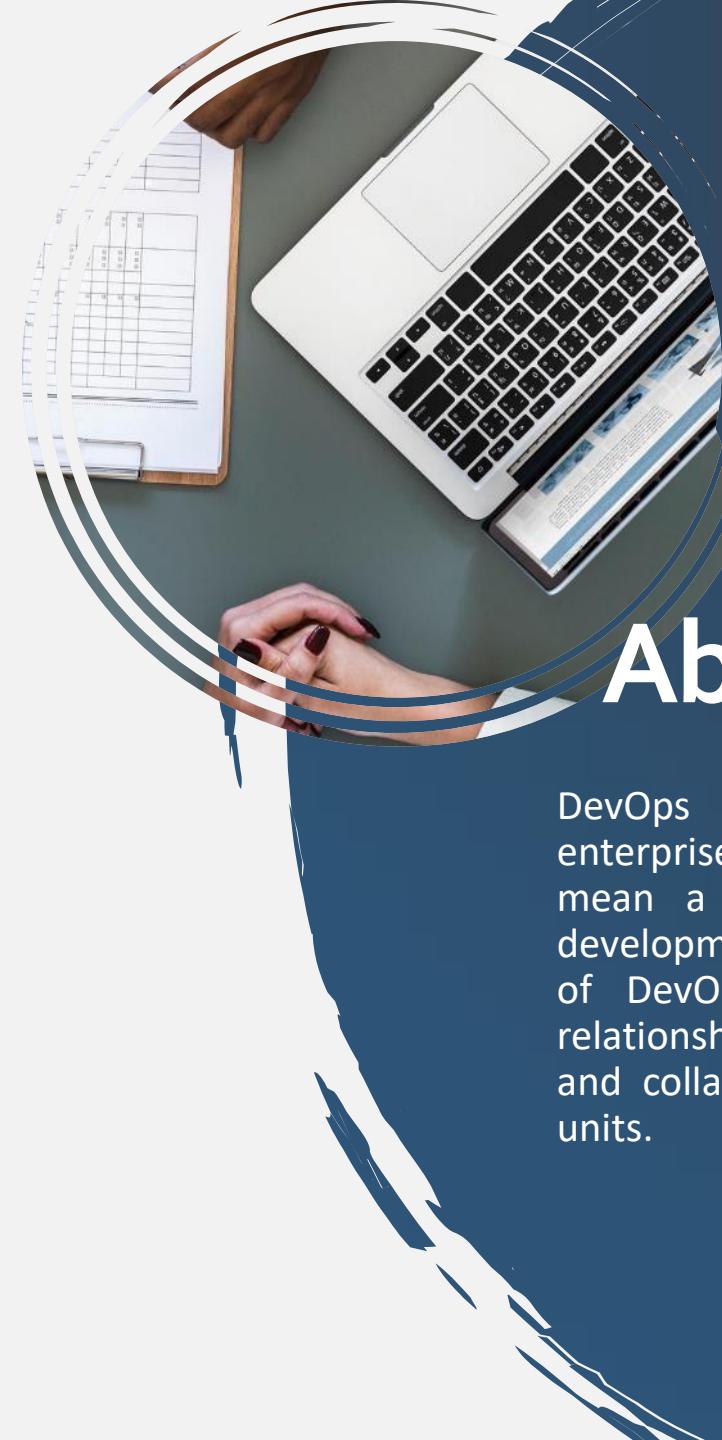
*We Chisel you to make you feel technically Vibrated, Motivated & Overwhelmed*

 TRAINING	 DEVELOPMENT	 SOLUTION	 STAFFING
We are the technology leaders committed to provide comprehensive training to the aspiring professionals in the game	Our software engineering process collects and translates business requirements into imaginative technology solutions that become reality with custom software development.	We are the technical leaders expertise in providing end to end solutions for the cutting edge technologies which industries demands for their business growth and analytics requirements.	Incepetz Technologies provides flexible, innovative recruitment strategies and technologies to maximize recruiting efficiency and reduce cost.

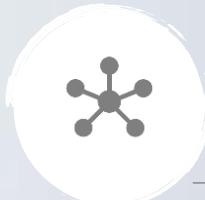
Dev-Ops “**Continuous Software Delivery**”, it’s “**Automation Strategies**” and with its ability to maintain well-balanced collaboration and communication between the “**Development and Operational**” units of its implying organization, Dev-Ops aims at achieving enhanced profits and boosted production outcomes. It is a well-proven fact that effective implementation of Dev-Ops strategies will place the implying organization way ahead of their respective competitors. So the demand for a qualified Dev-Ops engineer is more in the present corporate sector. But in contrast to this demand, there is a shortage of skilled Dev-Ops engineers. And so, many top notching organizations are willing to pay extremely high pay packages for the best-skilled Dev-Ops experts.

## About DEV-OPS

DevOps (development and operations) is an enterprise software development phrase used to mean a type of agile relationship between development and IT operations. The goal of DevOps is to change and improve the relationship by advocating better communication and collaboration between these two business units.



**Prerequisite** *Good to know, anyway don't worry, we will take care of it..*



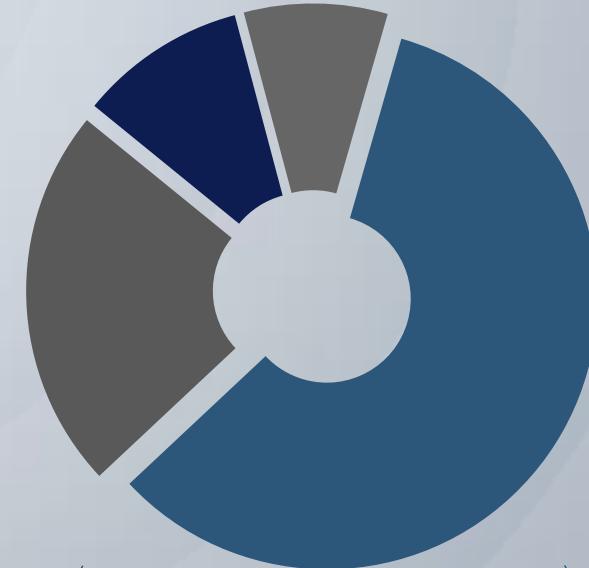
SDLC Knowledge



Good Attitude towards learning



Operating System  
Knowledge



Basic Command/Scripting Knowledge

# Key Component of this Course



DevOPS



Jenkins

Linux



kubernetes



Docker &  
Ansible

Dev-Ops “Continuous Software Delivery”, it’s “Automation Strategies” and with its ability to maintain well-balanced collaboration and communication between the “Development and Operational” units of its implying organization, Dev-Ops aims at achieving enhanced profits and boosted production outcomes.

**Docker** is a platform and tool for building, distributing, and running Docker containers.

**Ansible** is an open source automation platform. ... Ansible can help you with configuration management, application deployment, task automation. It can also do IT orchestration, where you have to run tasks in sequence and create a chain of events which must happen on several different servers or devices.

**Amazon Web Services** is a cloud computing platform that provides customers with a wide array of cloud services. Similarly, AWS is one of the cloud computing providers that provide us computing, storage, networking and lot more services that we can pay as we use.

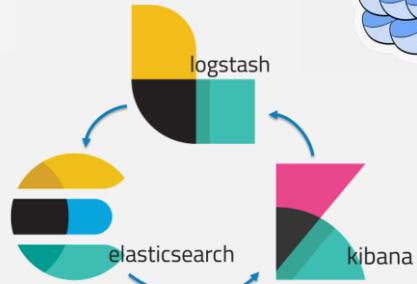


CHEF



Linux Scripting,  
Hadoop & Spark

**Linux Shell Scripting with Hadoop, Spark and DataScience** End to end implementation using DevOps tools. Hadoop & Spark are the open source Apache softwares for BigData management



Monitoring and  
Visualization

The **ELK Stack** is a collection of three open-source products Elasticsearch, Logstash, and Kibana all developed, managed and maintained by Elastic. Elasticsearch is a NoSQL database that is based on the Lucene search engine. Logstash collects and parses logs, and then Elasticsearch indexes and stores the information.

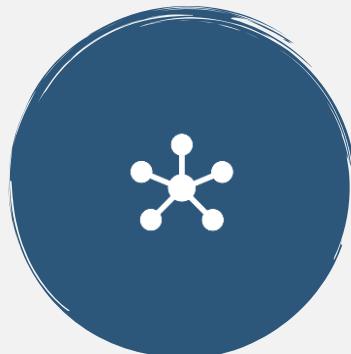
# What's Unique



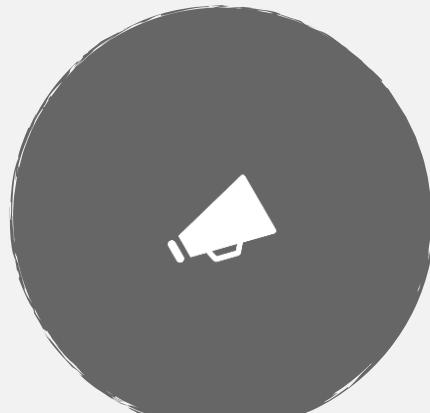
Unique



Best in class



Engagement based



Market Standard

*All under one course DevOps, Cloud, ELK & Cloud Docker, Kubernetes, Ansible, Chef, Git etc with Cloud*

*Job Oriented Training, Professional Environment*

*All Trainings by Industry Experts*

*Extended Training Duration*

*More Focus on Hands-on*

*Use cases & Projects Oriented*

*Interview and Job Support*

*End to end learning materials*

*360 degree Training Model*

*Competitive learning model*

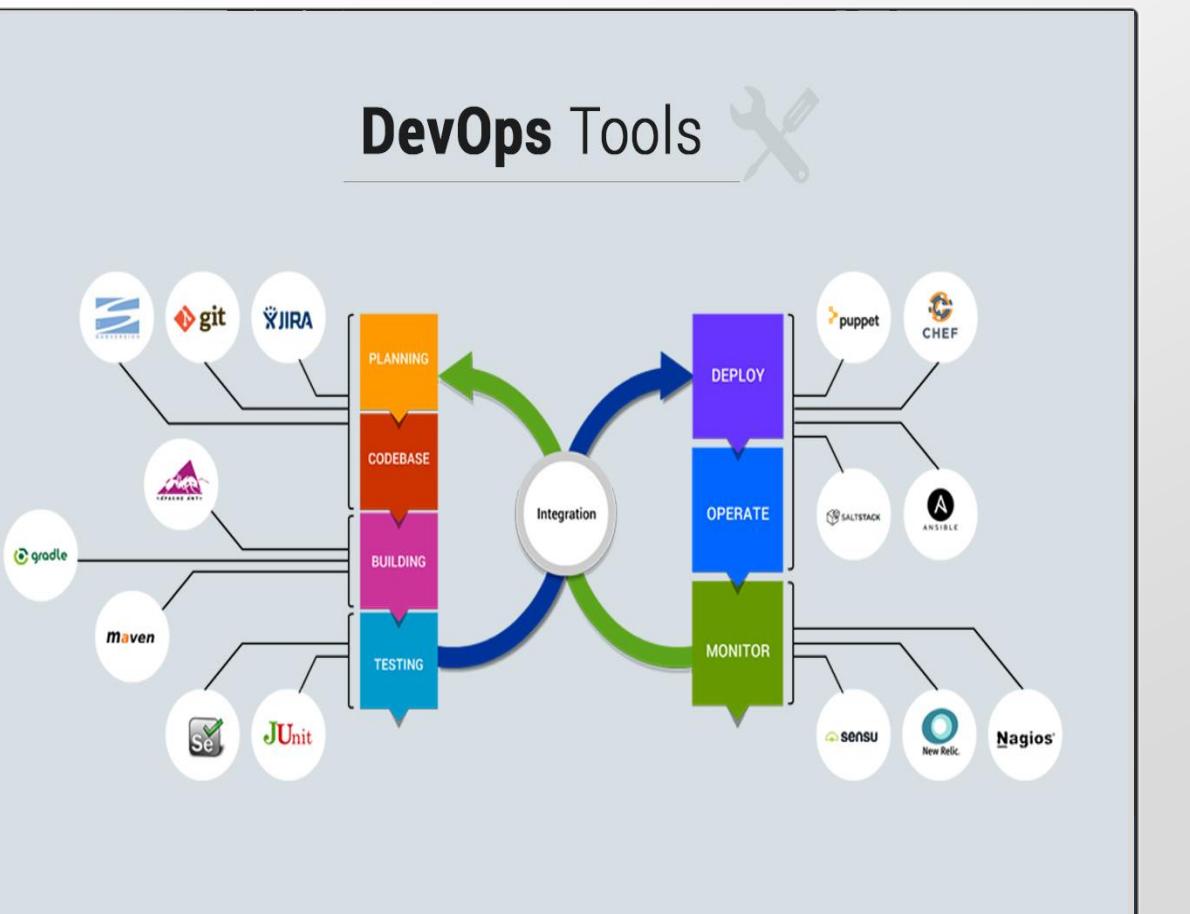
*3 Realtime Projects, Hackathons & Tests*

*Packaging & Deployment*

*Cloud based Hadoop/Spark Clustering using Devops Tools*

*Cloud based Datascience Model Deployment using Devops Tools*

# High Level Course Curriculum



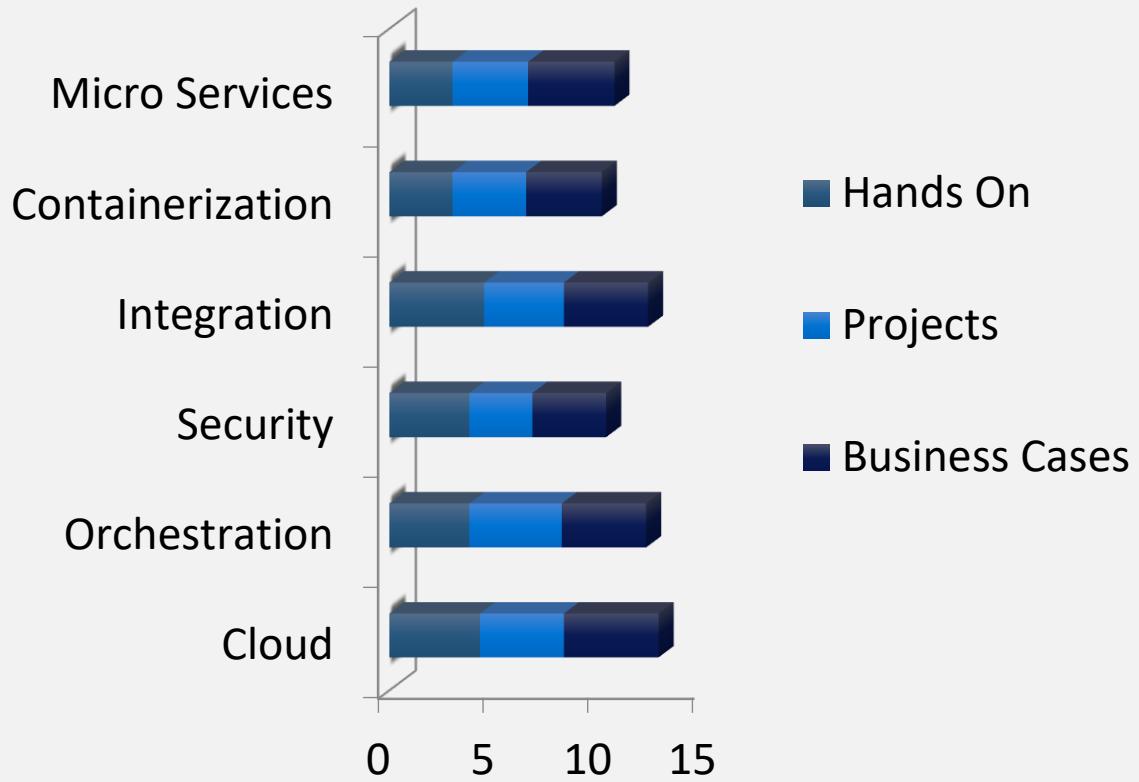
- Overview of DevOps
- Linux Commands, Shell Scripting & Python
- Introduction to VM on Cloud
- Version Control with Git
- Git, Jenkins & Maven Integration
- Continuous Integration using Jenkins
- AWS Components understanding
- Ansible & Chef managed automated Infrastructure with AWS
- Elastic Search, LogStash & Kibana Stack
- Configuration Management with Ansible
- Continuous Deployment & Orchestration: Containerization with Docker & Swarm
- Jenkins Automation pipeline with Blue Ocean
- Kubernetes networking

# Key Content Highlights

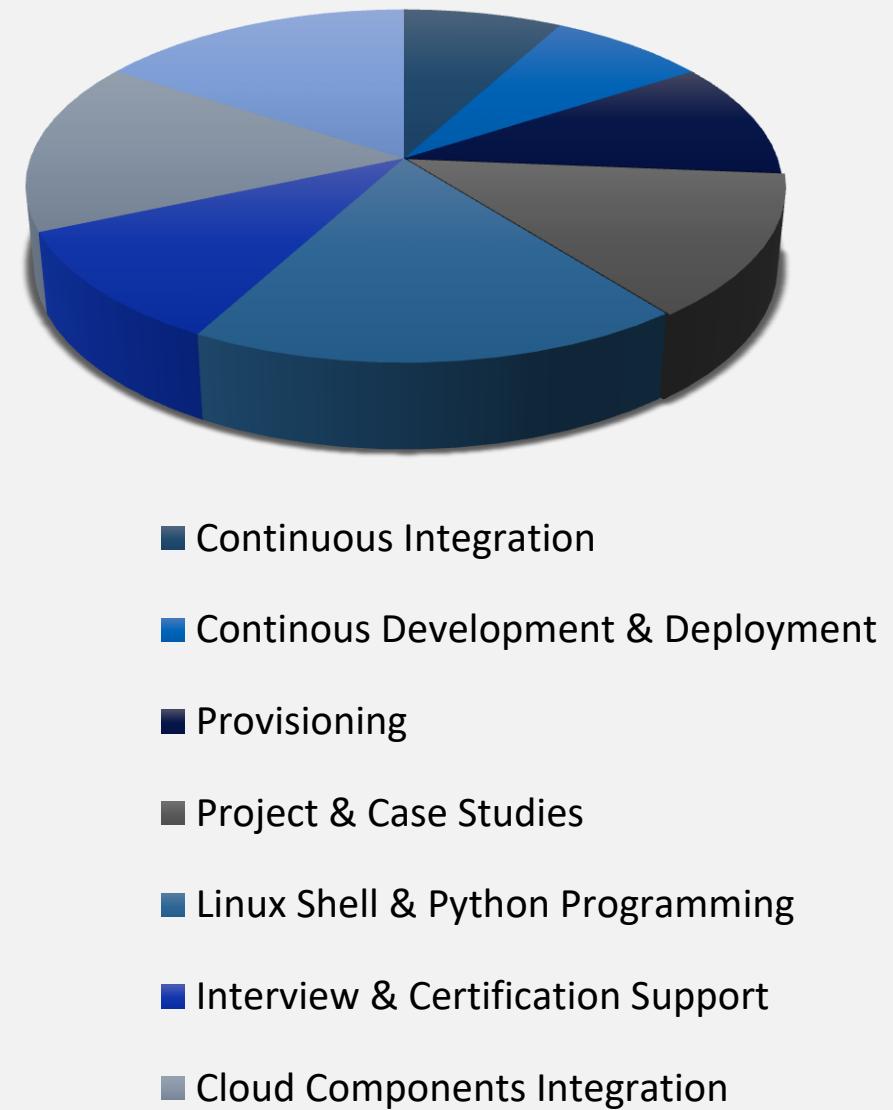
- Realtime Projects & Use Cases
- Best Practices & Daily Roles
- Containerization & Orchestration
- Cloud Environment with End to End deployment and management strategies
- Continuous Development & Testing
- Configuration Management
- Continuous Integration
- Continuous Deployment
- Continuous Monitoring
- Playbooks & Cookbooks
- Virtualization & Cloud
- CI-CD of Cloud, Hadoop & Spark frameworks
- Clustering Automation



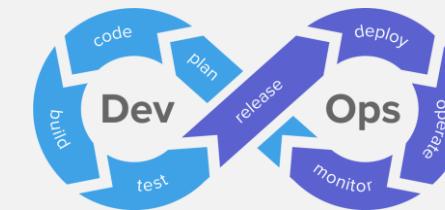
# Proportion of Contents



# Percentage of Feature Coverage



# DevOps Intro & Linux



## Overview of DevOps

In this module you will be introduced to Dev-Ops environment.

- DevOps Introduction
- What is DevOps?
- Why DevOps?
- Where DevOps is useful
- History of DevOps
- DevOps and Software Development Life Cycle
- DevOps main objectives
- DevOps on the Cloud
- Prerequisites for DevOps
- Frameworks for Devops



## Linux Foundation

In this module you will be learning Introduction & Key Components of Linux

- Introduction to Linux
  - What is Operating System?
  - What is Linux?
  - What is Linux Distro?
  - Major Linux Distros
  - History of Linux
- Key Components in Linux
  - Kernel
  - Terminal
  - Shell
  - User
  - Group
  - Partition
  - Filesystem
  - Process



## Linux Intro

In this module you will be linux shell, kernel and OS environment

- Package
- Service
- Network
- GUI
- SSH
- Shell Script
- Overview of Linux
  - Architecture of Linux
  - User Space
  - Kernel Space
- Linux vs Windows
- FAQ on Linux
- Prerequisites for Linux

## Linux Terminal & RunLevels

In this module you will be learning the terminal, run level and user managements

- Kernel
  - What is the Purpose of Kernel?
  - Types of Kernel
  - How Kernel Works?
    - Process Management
    - Memory Management
    - File System
- Manage Kernal
- Terminal
  - What is the Purpose of Terminal?
  - Major Terminal Emulators
  - What is TTY?
  - What is Psuedo-Terminal?
  - Manage Terminals



## Linux Permissions

In this module you will be learning Groups, Disk and Filesystem in Linux

- Shell
  - What is the Purpose of Shell?
  - Major Shells
  - Manage Shells
  - What is Environment Variable?
  - Manage Environment Variables
- User
  - What is the Purpose of User?
  - What is Sudo User?
  - Manage Users
- Group
  - What is the Purpose of Group?
  - Manage Groups
- Partition
  - What is the Purpose of Partition?



## Linux FS & Processes

In this module you will be learning FS and Processes

- Manage Partitions
- Filesystem
  - What is the Purpose of Filesystem?
  - What is INode?
  - Supported Filesystems in Linux
  - Structure of Filesystem
  - Manage Files & Directories in Filesystem
  - Manage Filesystems
  - Manage Ownerships & Permissions in Filesystem
- Process
  - What is the Purpose of Process?
  - What is Daemon?
  - Structure of Process Tree

## Linux Packaging & Services

In this module you will be learning the Packaging methods, services etc

- Process Signaling
- Schedule Process using Crontab
  - Crontab Syntax
  - Manage Crontab
- Manage Processes
- Package
  - What is the Purpose of Package?
  - What is Package Manager?
  - What is the Purpose of Package Manager?
  - Manage Packages
- Service
  - What is the Purpose of Service?
  - Manage Services



## Linux Network & Secure data transfers

In this module you will be learning Networking and Secure shells

- Network
  - What is the Purpose of Network?
  - Types of Network
  - What is IP?
  - What is Port?
  - What is Socket?
    - Raw Socket
    - Stream Socket
    - Datagram Socket
  - Understanding Network
  - Manage Networks
- Useful Commands



## Linux commands & GUI

In this module you will be learning other commands, GUI and Text editors

- GUI
  - What is the Purpose of GUI?
  - Major GUIs
- SSH
  - What is the Purpose of SSH?
  - History of SSH
  - SSH vs Telnet
  - How SSH Works?
  - Install & Configure SSH Server
  - Configure SSH Key-based Authentication
  - SSH to Remote Host
  - Do's and Dont's with SSH
- Interview on Linux

## Linux Scripting

In this module you will be learning Shell Scripting & Automation Services.

- **Shell Script**
  - Purpose of Shell Script?
  - Key Components in Shell Script
  - Writing Shell Script
  - Shebang
  - Comment, Execute , Debug
  - Environment Variable
  - Keyboard Input
  - Arithmetic Operation
  - Pathname Expansion
  - Alias, Startup
  - Conditional Statement
  - Loop, CLI Argument
  - STDIN/STDOUT/STDERR
  - Pipe, Function
- Do's and Dont's with Shell Script



## Version Control with Git

In this module you will be introduced to the Version control using Git Repository.

- **Introduction to Git**
  - Getting Started with Software Development
  - What is Version Control System (VCS)/Source Control Management (SCM)?
  - Need for VCS/SCM
  - Benefits of VCS/SCM
  - What is Git?
  - History of Git
  - Alternatives for Git
- **Key Components in Git**
  - Git Repository
  - Git Branch
  - Git Tag
  - Git Service Provider



## Git Setup & Config

In this module, you will learn about the setup and configuration of Git

- **Overview of Git**
  - Architecture of Git-managed Software Development
  - Centralized Distributed
  - Git vs Subversion
- **FAQ on Git**
- **Setup Git**
  - Install Git
  - Significance of Git Programs
  - Understanding Git System Paths
- **Configure Git**
  - Understanding Git Configurations
  - Apply Global Level Configurations in Git



## Git Handson

In this module, you will do the Hands on and Exploration of the Integration of components

- **Git Repository**
  - What is the Purpose of Git Repository?
  - Tree Structure of Git Repository
  - Initialize Git Repository
  - Status of Git Repository
  - Ignore Unwanted Files and Paths in Git Repository
  - Add Source Tree to Git Repository
  - Commit Changes to Git Repository
  - Manage Git Repositories
  - Deep Dive into Git Repository
    - How Push Mechanism Works in Git Repository?
    - How Pull Mechanism Works in Git Repository?

## Git Service

In this module you will be learning Git as a Service

- Do's and Dont's with Git Repository
- Git Service Provider
  - What is the Purpose of Git Service Provider?
  - Major Git Service Providers
    - GitHub
    - Bitbucket
    - GitLab
  - How to Choose a Git Service Provider?
  - Manage SSH Keys with Git Service Provider
  - Create Git Repository
  - Manage Repositories in Git Service Provider
  - Push Changes from Local Git Repository to Remote Git Repository



## Git Workflow & Case Study

In this module, you will learn about the Git Workflow and case study

- Integrate Git Repository with Jenkins CI Server
- Do's and Dont's with Git Service Provider
- Git Workflow
  - What is Git Workflow?
  - What is the Purpose of Git Workflow?
  - Architecture of Git Workflow
    - Principles of Git Workflow
- Execute Git Workflow
  - Initialize Git Workflow
  - Clone Git Repository
  - Cherry Pick Git Commits from One Branch and Apply to Other Branch
  - Merge Git Branches



## Git Major concepts

In this module, you will do the Hands on and Exploration of the merge, stash, reset and rebase

- Create Release using Git Tag
- Handle Bug After Production Deployment
- Do's and Dont's with Git Workflow
- Case Study
  - How We Implemented Git Workflow for SloopEngine Product Development
- Merge Conflict in Git
  - Why Merge Conflict Occurs in Git Repository?
  - Simulate Merge Conflict in Git Repository
  - Fix Merge Conflict in Git Repository
- Git Stash
  - What is the Purpose of Git Stash?
  - Stash Changes Before Switching to Another Git Branch

# Jenkins – Continuos Integration

## Git Major concepts

In this module, you will do the Hands on and Exploration of the merge, stash, reset and rebase

- **Git Reset**
  - What is the Purpose of Git Reset?
  - Undo Commits on Branch using Git Reset
- **Git Rebase**
  - What is the Purpose of Git Rebase?
  - Rebasing Scenario in Git Repository
  - Rebasing a Branch using Git Rebase
- **Interview on Git**

## Continuous Integration using Jenkins

In this module, you will know how to perform Continuous Integration using Jenkins by building CICD Pipelines.

- **Introduction to Jenkins**
  - What is Automated Testing?
  - What is Automated Build?
  - What is Continuous Integration (CI)?
  - What is Continuous Deployment (CD)?
  - What is Continuous Deployment (CD)?
  - What is CI/CD Pipeline?
  - Understanding CI/CD Pipeline
    - Build Phase in CI/CD Pipeline
    - Test Phase in CI/CD Pipeline
    - Deploy Phase in CI/CD Pipeline
    - Lifecycle of CI/CD Pipeline



# Jenkins

## Jenkins build jobs, pipelining, testing

In this module, you will do the Hands on and Exploration of the Installation, build, test and CD pipelining using Jenkins

- **What is Jenkins?**
- **History of Jenkins**
- **Key Components in Jenkins**
  - Jenkins Plugin
  - Jenkins CLI
  - Jenkins Node
  - Jenkins Project
  - Blue Ocean
- **Overview of Jenkins**
  - Architecture of Jenkins-managed Software Build/Testing
- **FAQ on Jenkins**
- **Prerequisites for Jenkins**
  - Install Ngrok

# Jenkins – Continuous Integration

## Continuous Integration using Jenkins

In this module, you will know how to perform Continuous Integration using Jenkins by setting up the configurations

- Setup Jenkins
  - Install Jenkins
  - Significance of Jenkins Programs
  - Manage Jenkins
- Configure Jenkins
  - Understanding Jenkins System Paths
  - Unlock Jenkins
  - Enable/Disable Jenkins Plugins
  - Create Admin User
  - Create API Token for User
  - Configure Jenkins Location
  - Configure E-mail Notification
  - Configure Extended E-mail Notification
    - Use Gmail SMTP Server

## Jenkins CLI, Nodes & Agents

In this module, you will learn about the configuration of CLI, SSH, Nodes & Freestyle projects

- Configure Agents
- Configure SSH Server
- Jenkins CLI
  - Install Jenkins CLI
  - Configure Jenkins CLI in SSH Mode
- Jenkins Node
  - Using Docker Containers as Jenkins Nodes
  - Configure Jenkins Node
- Jenkins Freestyle Project
  - What is Jenkins Freestyle Project?
  - What is the Purpose of Jenkins Freestyle Project?



# Jenkins

## Jenkins build jobs, pipelining, testing

In this module, you will do the Hands on and Exploration of the Installation, build, action with Freestyle projects using Jenkins

- Setup CI/CD using Jenkins Freestyle Project
  - General
  - Source Code Management
  - Build Triggers
  - Build Environment
  - Build
  - Post-build Actions
- Trigger CI/CD of Jenkins Freestyle Project
  - Configure Webhook on GitHub
- Manage Jenkins Freestyle Projects
- Do's and Dont's with Jenkins Freestyle Project



## Continuous Integration using Jenkins

Learn about the Blue Ocean GUI plugin of Jenkins & the pipeline projects with Ansible plugin integration

- Blue Ocean
  - What is the Purpose of Blue Ocean?
  - Install Blue Ocean
  - Understanding Blue Ocean
- Jenkins Pipeline Project
  - What is Jenkins Pipeline Project?
  - What is the Purpose of Jenkins Pipeline Project?
  - Structure of Declarative Jenkins Pipeline
  - Enable Ansible Plugin in Jenkins
  - Setup CI/CD using Jenkins Pipeline Project
    - General
    - Build Triggers
    - Pipeline



## Jenkins CI/CD pipeline Projects

In this module, you will learn about the integration of git with webhooks with the CI/CD Pipeline integration + Interview discussion.

- Trigger CI/CD of Jenkins Pipeline Project
  - Configure Webhook on GitHub
- Manage Jenkins Pipeline Projects
- Do's and Dont's with Jenkins Pipeline Project
- Interview on Jenkins



## Containerization using Docker

This module introduces Docker to readers, the core concepts and technology behind Docker, container and various operations performed.

- Introduction to Docker
  - Understanding Service Oriented Architecture (SOA)
  - Understanding Microservice Architecture
  - What is Containerization?
  - Evolution of Containers
  - Benefits of Container
  - What is Docker?
  - History of Docker
  - What is Open Container Initiative (OCI)?
  - What is the Purpose of Open Container Initiative (OCI)?

# Docker - Containerization

## Docker Components Overview

Understand about the components of Docker such as image, registry, network, container, compose etc.,

- Key Components in Docker
  - Docker Image
  - Docker Registry
  - Docker Network
  - Docker Container
  - Docker Volume
  - Docker Compose
- Overview of Docker
  - Architecture of Docker-managed Containers
    - Docker Client
    - Docker Daemon
    - Docker Image
    - Docker Container
    - Docker Network
    - Docker Volume

## Containerization using Docker

This module introduces Docker to readers, the core concepts and technology behind Docker, container and various operations performed.

- Docker Registry
- Container vs VM
- FAQ on Docker
- Prerequisites for Docker
- Setup Docker
  - What is Containerd?
  - Install Docker
  - Significance of Docker Programs
  - Manage Docker
  - Deep Dive into Docker
- Configure Docker
  - Understanding Docker Configurations
  - Change Storage & Log Drivers of Docker



## Docker Image management

In this module, you will learn about the end to end management of Docker images & Automation

- Apply Configuration Changes to Docker
- Understanding Docker System Paths
- Docker Image
  - What is the Purpose of Docker Image?
  - Deriving Docker Images
    - Tree Structure of Derived Docker Images
    - Benefits of Deriving Docker Images
  - Build Docker Image Manually
  - Understanding Dockerfile
  - Automated Build of Docker Images using Dockerfile
  - Manage Docker Images

# Docker – Deep Dive



## Docker Network

Handson on the Docker registry, networks, inspection of networks etc

- Deep Dive into Docker Image
- Do's and Dont's with Docker Image
- Docker Network
  - What is Container Network Model (CNM)?
  - What is Container Network Model (CNM)?
  - What is the Purpose of Docker Network?
  - Types of Docker Network
  - How to Choose a Docker Network?
  - Manage Docker Networks
  - Using Host Docker Network
  - Using None Docker Network
  - Using Bridge Docker Network



## Docker Storage & Containers deep dive

Handson of the Docker configuration methodologies with storage and containers deep dive management

- Docker Storage
  - What is the Purpose of Docker Storage?
  - Types of Docker Storage
  - How to Choose a Docker Storage?
  - Manage Docker Storage
    - Using Volume Docker Storage
    - Using Bind Mount Docker Storage
- Docker Container
  - What is the Purpose of Docker Container?
  - Understanding Docker Container
    - How Docker Container is Created?
    - How Docker Container Lifecycle is Controlled?



## Multi Containerization using Compose

Handson of the multi container management using Docker Compose

- Analyzing Docker Container Configurations
- What is Supervisor?
- Managing Microservices within Docker Container using Supervisor
- Design Highly Scalable Docker Container
- Provision Docker Containers Manually
- Manage Docker Containers
- Deep Dive into Docker Container
- Do's and Dont's with Docker Container
- Docker Compose
  - Understanding YAML
  - What is the Purpose of Docker Compose?

# Docker + Kubernetes Introduction

## Docker YAML + Case Study + Interview Support

Handson + of the Docker configuration methodologies, imaging and automation

- Install Docker Compose
- Understanding Docker Compose YAML
- Structure of Docker Compose YAML
- Automated/Orchestrated Provisioning of Docker Containers using Docker Compose
- Manage Docker Containers using Docker Compose
- Case Studies
  - Simulating Sloop Engine Multi-Pod Architecture on Docker
  - How We Scaled Multi-Environment on Docker for Leading Product Startup
- Interview on Docker



## Kubernetes Orchestration

Understanding about Orchestration using Kubernetes + Docker with cloud native computing foundation

- Introduction to Kubernetes
  - What is Docker?
  - What is Cloud Native Computing Foundation (CNCF)?
  - Understanding Container Clustering & Orchestration
    - Benefits of Container Clustering & Orchestration
    - Major Container Clustering & Orchestration Tools
  - What is Kubernetes?
  - Understanding Kubernetes
  - What is Container Network Interface (CNI)?
  - History of Kubernetes

## Kubernetes Components Overview

Key components of Kubernetes Cluster with managed containers

- Key Components in Kubernetes
  - Kubernetes Node
  - Kubernetes Namespace
  - Kubernetes Deployment
  - Kubernetes Pod
  - Kubernetes Service
  - Kubernetes Ingress
  - Kubernetes Role
  - Kubernetes ClusterRole
- Overview of Kubernetes
  - Architecture of Kubernetes-managed Docker Containers
    - Kubernetes API Server
    - Etcd
    - CoreDNS
    - Kubernetes Scheduler



## Kubernetes Setup

Handson on groundwork, installation & Setup of Kubernetes

- Kubernetes Controller Manager
- Kubernetes Agent
- Kubernetes Proxy
- Kubernetes vs Swarm
- FAQ on Kubernetes
- Prerequisites for Kubernetes
  - Install Kubernetes Client
  - Boot Kubernetes Nodes
- Setup Kubernetes
  - Install & Configure Docker
  - Install Kubernetes
  - Significance of Kubernetes Programs
  - Manage Kubernetes



## Kubernetes Config & Node management

Handson on Configuring & manage Kubernetes cluster with nodes with Deep dive of scaling management

- Configure Kubernetes
  - Initialize Kubernetes Cluster
  - Join Nodes to Kubernetes Cluster
  - Prepare Nodes in Kubernetes Cluster
  - Manage Tokens in Kubernetes Cluster
  - Configure Kubernetes User
  - Deep Dive into Kubernetes Cluster
- Kubernetes Node
  - What is the Purpose of Kubernetes Node?
  - Provision Highly Scalable Kubernetes Nodes
  - Scaling Master Kubernetes Nodes



## Kubernities Namespace

Handson on the Kubernetes namespace and deployment

- Scaling Worker Kubernetes Nodes
- Manage Kubernetes Nodes
- Kubernetes Namespace
  - What is the Purpose of Kubernetes Namespace?
  - Understanding Kubernetes Namespace
  - Create Kubernetes Namespace
  - Manage Kubernetes Namespaces
  - Manage Kubernetes Namespaces
- Kubernetes Deployment
  - What is the Purpose of Kubernetes Deployment?
  - Create Kubernetes Deployment
  - Manage Kubernetes Deployments

# Kubernetes Orchestration



kubernetes

## Kubernetes POD + Service

Handson on the understanding and implementation of Kubernetes POD architecture & Expose as service

- Kubernetes Pod
  - What is the Purpose of Kubernetes Pod?
  - Understanding Kubernetes Pod
  - Manage Kubernetes Pods
  - Deep Dive into Kubernetes Pod
- Kubernetes Service
  - What is the Purpose of Kubernetes Service?
  - Ways to Expose Kubernetes Service
  - Create Kubernetes Service
  - Manage Kubernetes Services
  - Deep Dive into Kubernetes Service
  - How Kubernetes Service Discovery Works?

## Kubernetes rules & Security management

Enable the routing rules with the role management for permissions

- Kubernetes Ingress
  - What is the Purpose of Kubernetes Ingress?
  - Understanding Kubernetes Ingress Controller
  - Create Kubernetes Ingress Controller
  - Create Kubernetes Ingress
  - Manage Kubernetes Ingress
  - Manage Kubernetes Ingress Controllers
- Kubernetes Role
  - What is the Purpose of Kubernetes Role?
  - Create Kubernetes Role
  - Create Kubernetes Role Binding
  - Manage Kubernetes Role Bindings

## Kubernetes Cluster role + Real implementation Case study + Interview Discussion

- Manage Kubernetes Roles
- Kubernetes ClusterRole
  - What is the Purpose of Kubernetes ClusterRole?
  - Create Kubernetes ClusterRole
  - Create Kubernetes ClusterRoleBinding
  - Manage Kubernetes ClusterRoleBindings
  - Manage Kubernetes ClusterRoles
- Case Study
  - Running SloopEngine Multi-Pod Architecture on Kubernetes-managed Docker Containers
- Interview on Kubernetes

# Swarm

## Docker Swarm

Intro to the group of virtual machines that are running the Docker application

- Introduction to Swarm
  - What is Docker?
  - Understanding Container Clustering & Orchestration
    - Benefits of Container Clustering & Orchestration
    - Major Container Clustering & Orchestration Tools
  - What is Swarm?
  - Understanding Swarm
  - History of Swarm
- Key Components in Swarm
  - Swarm Node, Stack , Service
- Overview of Swarm
  - Architecture of Swarm-managed Docker Containers
  - Swarm vs Kubernetes

## Configure & DeepDive Swarm Cluster

Handson on configuring nodes and form Swarm cluster and deepdive of nodes

- FAQ on Swarm
- Setup Swarm
  - Boot Swarm Nodes
  - Install & Configure Docker
- Configure Swarm
  - Initialize Swarm Cluster
  - Join Nodes to Swarm Cluster
  - Prepare Nodes in Swarm Cluster
  - Manage Tokens in Swarm Cluster
  - Deep Dive into Swarm Cluster
- Swarm Node
  - Purpose of Swarm Node?
  - Provision Highly Scalable Swarm Nodes
  - Scaling Manager Swarm Nodes
  - Scaling Worker Swarm Nodes



## Swarm Stack & Service

Create and manage Swarm stack and mange the services with multi POD architecture realtime implementation + Interview discussion

- Quorum-based Stability of Manager Swarm Nodes
- Manage Swarm Nodes
- Swarm Stack
  - Purpose of Swarm Stack?
  - Understanding Swarm Stack
  - Create Swarm Stack
  - Manage Swarm Stacks
- Swarm Service
  - Purpose of Swarm Service?
  - Ways to Expose Swarm Service
  - Manage Swarm Services
- Case Study
  - Running SloopEngine Multi-Pod Architecture on Swarm-managed Docker Containers
- Interview on Swarm

# Ansible Playbook

## Config Management with Ansible & Handson

Learn about the Ansible automated deployments, key components, overview

- Introduction to Ansible
  - Infrastructure Management
  - What is Capacity Planning?
    - Major Capacity Planning Tools
  - Understanding Capacity Planning
    - Calculating Percentile
  - What is Provisioning?
    - Major Provisioning Tools
  - What is Deployment?
    - Major Deployment Tools
- Need for Automated Deployments
- Deployment Matrix
  - App Service
  - Data Service
- What is Ansible?



## Ansible Control Machine

Understand the core components of the Ansible deployment tool

- History of Ansible
- Key Components in Ansible
  - Ansible Control Machine
  - Ansible Node
  - Ansible Playbook
  - Ansible Role
  - Ansible Galaxy
  - Ansible Tower
- Overview of Ansible
  - Architecture of Ansible-managed Deployments
  - Ansible Control Machine
  - Ansible Node
  - Ansible Playbook
  - Ansible Role
  - Ansible Inventory

## Ansible Playbook, Role & Nodes

Hands on of the Ansible Control Machine with playbook creation and management

- Ansible vs Chef
- FAQ on Ansible
- Prerequisites for Ansible
  - Understanding YAML
- Ansible Control Machine
  - What is the Purpose of Ansible Control Machine?
  - Install Ansible
  - Significance of Ansible Control Machine Programs
  - Understanding Ansible Control Machine Configurations
  - Configure Ansible Control Machine
  - Configure Highly Scalable Ansible Control Machine
  - Understanding Ansible Control Machine System Paths

# Ansible Playbook



## Ansible Inventory & Playbook

Deep Handson on the inventory management & writing of playbooks

- Deep Dive into Ansible Control Machine
  - How Ansible Control Machine Executes Playbook/Role?
- Do's and Dont's with Ansible Control Machine
- Ansible Inventory
  - What is the Purpose of Ansible Inventory?
  - Manage Ansible Inventories
- Ansible Playbook
  - What is the Purpose of Ansible Playbook?
  - Structure of Ansible Playbook
  - Writing Ansible Playbook
  - Manage Ansible Playbooks
  - Do's and Dont's in Ansible Playbook

## Ansible Role & Vault

This module gives the handson on the Ansible Roles & Vault management

- Ansible Role
  - What is the Purpose of Ansible Role?
  - Structure of Ansible Role
  - Writing Ansible Role
  - Manage Ansible Roles
  - Do's and Dont's with Ansible Role
- Ansible Vault
  - What is the Purpose of Ansible Vault?
  - Create Encrypted File using Ansible Vault
  - Create Encrypted String using Ansible Vault
  - Manage Encryption using Ansible Vault

## Ansible Nodes & Towers

Handson of the Ansible management of nodes and towers with Org structuring

- Ansible Node
  - Understanding Deployment on Ansible Nodes
  - Using Docker Containers as Ansible Nodes
  - Boot Ansible Nodes
  - Trigger Deployment on Ansible Nodes
  - Execute Ad-hoc Commands on Ansible Nodes
- Ansible Tower
  - What is the Purpose of Ansible Tower?
  - Setup & Configure Ansible Tower
  - Using Ansible Tower
  - Create Organization
  - Create Team

# Chef Infra management

## Ansible Nodes Deep dive

Handson on the Ansible boot nodes, credential and group management with Interview discussion

- Create Inventory
- Create Host
- Create Group
- Create Credential
- Create Project
- Create Job Template
- Boot Ansible Nodes
- Trigger Deployment on Ansible Nodes
- Interview on Ansible



ANSIBLE

## Chef End to End

Understand about the Infra as Code management using Chef for provisioning & Deployment.

- Introduction to Chef
  - Infrastructure Management
  - What is Capacity Planning?
    - Major Capacity Planning Tools
  - Understanding Capacity Planning
    - Calculating Percentile
  - What is Provisioning?
    - Major Provisioning Tools
  - What is Deployment?
    - Major Deployment Tools
  - Need for Automated Deployments
    - Need for Automated Deployments
  - Deployment Matrix
    - App Service
    - Data Service



## Chef Infra Server, Manage and Push Job Servers

Chef infrastructure server, chef manage, system paths, push job servers etc.

- What is Chef?
- History of Chef
- Key Components in Chef
  - Chef Infra Server
  - Chef Infra Client
  - Chef Workstation
  - Chef Development Kit
  - Chef Solo
  - Chef Manage
  - Chef Automate
  - Chef Node
  - Chef Push Jobs Server
  - Chef Push Jobs Client
  - Chef Cookbook
  - Chef Role
  - Chef Supermarket

# Chef Workstation

## Chef Managed Deployment

Overview and deep dive on the Chef managed deployment of workstation, nodes, client etc.

- Overview of Chef
  - Architecture of Chef-managed Deployments
  - Chef Infra Server
  - Chef Manage
  - Chef Workstation
  - Chef Node
  - Chef Infra Client
  - Chef Push Jobs Server
  - Chef Push Jobs Client
  - Chef Cookbook
  - Chef vs Ansible
- FAQ on Chef
- Prerequisites for Chef



CHEF

## Chef Infra Server management

Handson with Installation, Configuration and Management of the Chef Infra Server

- Chef Infra Server
  - What is the Purpose of Chef Infra Server?
  - Key Components in Chef Infra Server
  - Install Chef Infra Server
  - Significance of Chef Infra Server Programs
  - Configure Chef Infra Server
  - Understanding Chef Infra Server System Paths
  - Manage Chef Infra Server
- Chef Manage
  - What is the Purpose of Chef Manage?
  - Install Chef Manage

## Chef Infra Server, Manage and Push Job Servers

Chef infrastructure server, chef manage, system paths, push job servers etc.

- Significance of Chef Manage Programs
- Configure Chef Manage
- Understanding Chef Manage System Paths
- Manage Chef Manage
- Chef Push Jobs Server
  - What is the Purpose of Chef Push Jobs Server?
  - Install Chef Push Jobs Server
  - Significance of Chef Push Jobs Server Programs
  - Configure Chef Push Jobs Server
  - Understanding Chef Push Jobs Server System Paths
  - Manage Chef Push Jobs Server

# Chef Workstation

## Chef Workstation

Learn and handson on the chef push jobs server with the workstation using Ruby scripting.

- Chef Workstation
  - What is the Purpose of Chef Workstation?
  - Install Chef Workstation
  - Significance of Chef Workstation Programs
  - Configure Chef Workstation
  - Understanding Chef Workstation System Paths
  - Do's and Dont's with Chef Workstation
- Chef Cookbook
  - What is the Purpose of Chef Cookbook?
  - Structure of Chef Cookbook
  - Writing Chef Cookbook
  - Manage Chef Cookbooks



## CHEF Chef Bootstrap + Case Study + Interview

Case study with end to end deployment in AWS with Interview discussion

## Chef Cookbook, Role & Node creation with Docker

Handon on creating cookbooks for chef, run on solo, creation of chef roles, chef node using docker containers

- Do's and Dont's with Chef Cookbook
- Chef Role
  - What is the Purpose of Chef Role?
  - Create Chef Role
  - Manage Chef Roles
- Chef Node
  - Understanding Deployment on Chef Nodes
  - Using Docker Containers as Chef Nodes
  - Boot Chef Nodes
  - Trigger SSH-based Deployment on Chef Nodes
  - Bootstrap Workflow in Chef Node

- Trigger Agent-based Deployment on Chef Nodes
- Case Study
  - Managing SloopEngine Deployments on AWS OpsWorks using Chef
- Interview on Chef

# Terraform Infrastructure Automation



## Terraform

Define and provision the end to end datacenter infrastructure

- Introduction to Terraform
  - Infrastructure Management
  - What is Capacity Planning?
    - Major Capacity Planning Tools
  - Understanding Capacity Planning
    - Calculating Percentile
  - What is Provisioning?
    - Major Provisioning Tools
- Need for Automated Provisioning
  - What is Deployment?
    - Major Deployment Tools
- What is Terraform?
- History of Terraform
- Key Components in Terraform
  - Terraform Provider

## Terraform on AWS

Learn about the integration of Terraform with AWS platform automation

- Terraform Configuration
- Terraform Registry
- Terraform Cloud
- Overview of Terraform
  - Architecture of Terraform-managed Infrastructure
    - Terraform Configuration
    - Terraform Provider
    - Terraform Provisioner
    - Terraform Registry
  - Terraform vs AWS CloudFormation
- FAQ on Terraform
- Prerequisites for Terraform
  - Create AWS Account
  - Install AWS CLI

## Terraform for realtime

In this module, you will learn applications of Terraform with the end to end config and automation

- Setup Terraform
  - Install Terraform
  - Significance of Terraform Programs
- Terraform Provider
  - What is the Purpose of Terraform Provider?
  - Manage Terraform Providers
- Terraform Configuration
  - What is the Purpose of Terraform Configuration?
  - Structure of Terraform Configuration
  - Writing Terraform Configuration
  - Initialize Terraform Configuration
  - Generate Plan from Terraform Configuration
  - Apply Plan of Terraform Configuration

## Terraform Cloud

Define and provision the end to end  
datacenter infrastructure

- Manage Terraform Configurations
- Do's and Dont's with Terraform Configuration
- Terraform Cloud
  - What is the Purpose of Terraform Cloud?
  - Understanding Terraform Cloud
  - Manage Terraform Cloud
  - Do's and Dont's with Terraform Cloud
- Interview on Terraform

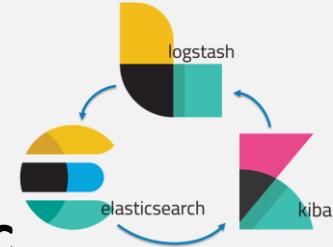


# ELK Stack – Data pipeline

## ELK Stack

Learn about the Elastic Search, LogStash & Kibana

- Introduction to ELK
  - What is Observability?
  - Understanding Observability
    - Log Data for Observability
    - Metric Data for Observability
    - Performance Data for Observability
    - Performance Data for Observability
    - Benefits of Observability
    - How to Choose an Observability Platform?
  - What is Log Management?
    - Major Log Management Tools
  - What is Security Information & Event Management (SIEM)?
    - Major SIEM Tools



## ELK Overview & Elastic Search Deep dive

This module details about the ELK stack and deep dive of Elastic Search components and applications.

- What is Application Performance Management (APM)?
  - Major APM Tools
- What is Infrastructure Monitoring?
  - Major Infrastructure Monitoring Tools
- What is ELK?
  - Managed ELK Services
  - History of ELK
- Key Components in ELK
  - Elasticsearch
  - Logstash
  - Kibana
  - Beats
    - Auditbeat
    - Filebeat
- Functionbeat
- Heartbeat
- Metricbeat
- Packetbeat
- Winlogbeat
- APM
- Overview of ELK
  - Architecture of ELK-managed Observability
  - Beats
  - Elasticsearch
  - Logstash
  - Kibana
  - ELK vs Splunk vs Sumo Logic in Terms of Log Management and SIEM
  - ELK vs NewRelic in Terms of APM

## Components Deep Dive

In this module, you will learn the overview and deep dive comparison of components

# ELK Visualization & Dashboard

## Elastic Search Deep Dive

End to end handson session  
configuring and managing Elastic  
Search Engine

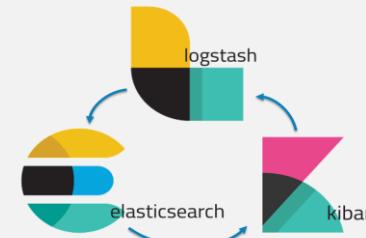
- ELK vs Grafana vs Nagios in Terms of Infrastructure Monitoring
- FAQ on ELK
- Prerequisites for ELK
- Elasticsearch
  - What is the Purpose of Elasticsearch?
  - Key Components in Elasticsearch
  - Using Docker Containers as Elasticsearch Nodes
  - Install Elasticsearch
  - Significance of Elasticsearch Programs
  - Configure Elasticsearch
  - Configure Highly Scalable Elasticsearch



## LogStash

Configure and use Logstash for data enablement using Docker container integration

- Understanding Elasticsearch System Paths
- Manage Elasticsearch
- Logstash
  - What is the Purpose of Logstash?
  - Key Components in Logstash
  - Using Docker Containers as Logstash Nodes
  - Install Logstash
  - Significance of Logstash Programs
  - Configure Logstash
  - Understanding Logstash System Paths
  - Manage Logstash
- Kibana
  - What is the Purpose of Kibana?

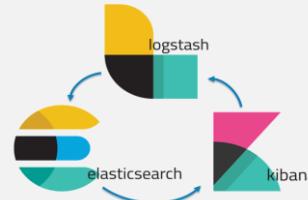


## Kibana Visualization & Dashboard

Overview of Kibana dashboard and visualization creation for analysis

- Key Components in Kibana
- Using Docker Containers as Kibana Nodes
- Install Kibana
- Significance of Kibana Programs
- Configure Kibana
- Understanding Kibana System Paths
- Manage Kibana
- APM
  - What is the Purpose of APM?
  - Key Components in APM
  - Using Docker Containers as APM Nodes
  - Install APM
  - Significance of APM Programs
  - Configure APM

# ELK Beat + Visualization



## APM & FileBeat

End to end handson of APM streams  
& Filebeat components of ELK

- Understanding APM System Paths
- Manage APM
- **Filebeat**
  - What is the Purpose of Filebeat?
  - Key Components in Filebeat
  - Install Filebeat
  - Significance of Filebeat Programs
  - Configure Filebeat
  - Understanding Filebeat System Paths
  - What is Filebeat Module?
  - Key Components in Filebeat Module
  - Structure of Filebeat Module
  - Writing Filebeat Module
  - Manage Filebeat

## MetricBeat

End to end handson session on metricbeat shipper with setup and configuration.

- Metricbeat
  - What is the Purpose of Metricbeat?
  - Install Metricbeat
  - Significance of Metricbeat Programs
  - Configure Metricbeat
  - Understanding Metricbeat System Paths
  - Manage Metricbeat
- Heartbeat
  - What is the Purpose of Heartbeat?
  - Install Heartbeat
  - Significance of Heartbeat Programs
  - Configure Heartbeat

## Kibana Visualization & Dashboard

End to end handson session on Kibana dashboard and visualization creation for analysis of log events.

- Understanding Heartbeat System Paths
- Manage Heartbeat
- Using Kibana
  - Index Pattern
    - What is the Purpose of Index Pattern?
    - Create Index Pattern
  - Discover
    - What is the Purpose of Discover?
    - Execute Lucene Queries on Discover
  - Visualize
    - What is the Purpose of Visualize?
    - Create Timelion Visualization
    - Create Guage Visualization

# Python Programming



## ELK Analytics

Analytics of realtime log using ELK stack and ELK on top of AWS

- Create Pie Visualization
- Dashboard
  - What is the Purpose of Dashboard?
  - Create Dashboard
- Infrastructure
- APM
- Logs
- Uptime
- Dev Tools
- Monitoring
- Watcher
  - What is the Purpose of Watcher?
  - Setup & Configure Slack
  - Create Alert on Watcher



## ELK Case study + Interview

Realtime implementation with the sloopengine log data + Interview discussion

- Simulate Traffic to App
  - Case Studies
  - Analyzing SloopEngine Log Data using ELK
  - Analyzing SloopEngine Metric Data using ELK
- Interview on ELK



## Programming Basics

Start doing programming from basic with Python

- Introduction to Python
  - What is Programming Language?
  - What is Pseudo Code?
  - What is Python?
  - Types of Python
  - What is C-Python?
  - Python Run-Time and Compile-Time
  - Why Python?
  - What is Framework?
  - Types of Framework
  - How to Choose a Framework?
  - What is Micro Framework?
  - What is App/Service?
  - History of Python
  - Pros & Cons of Python

# Python Programming



## Python Components, OOPS

Handson of Python core modules including OOPS, Apps, APIs, Packages and Debugger

- Key Components in Python
  - Python Variable
  - Python Data Structure
  - Python Conditional Statement
  - Python Loop
  - Python Function
  - Python Exception
  - Python Interactive Mode
  - Python OOPS
  - Python Framework
  - Python App
  - Python API
  - Python Package
  - Python Debugger
- Overview of Python
  - Architecture of Python

## Python Deep Dive

End to end handson session on Setup, configure, data structures, exception handling and Functions

- Key Differences Between Python & Other Technologies
- FAQ on Python
- Prerequisites for Python
- Setup Python
- Python Variable
- Python Data Structure
- Python Conditional Statement
- Python Loop
- Python Function
- Python Exception
- Python Interactive
- Python OOPS

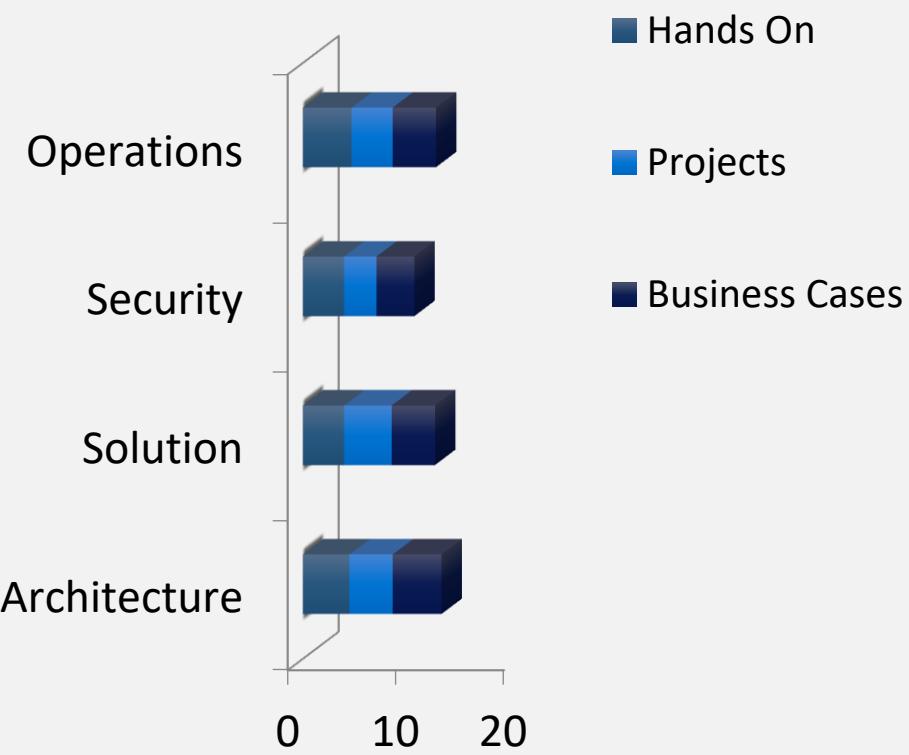
## Python App, API & Packaging

Handson of managing API, App, Package, Debugger + Microservices Case Study + Interview discussion

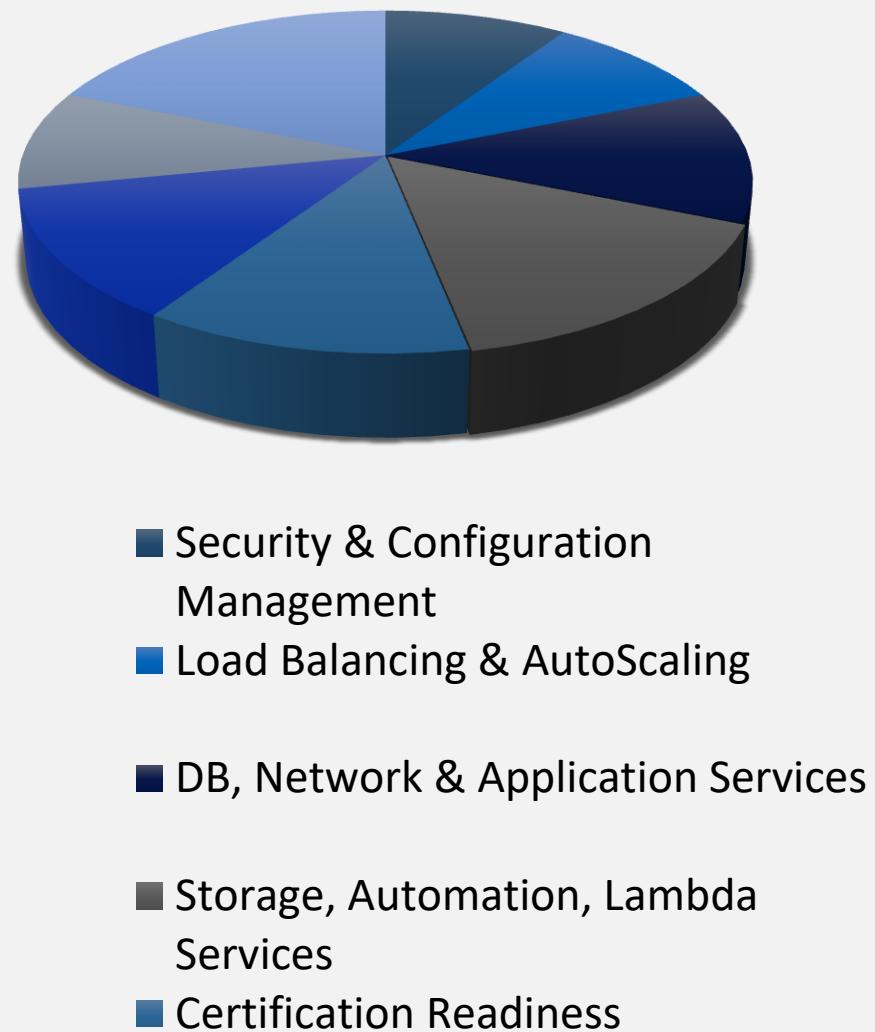
- Python Framework
- Python App
- Python API
- Python Package
- Python Debugger
- Case Studies
  - How We Built SloopEngine Microservices using Python
- Interview on Python



## AWS Proportion of Contents



## Percentage of Feature Coverage



## Introduction & Overview

learning the introduction of AWS Cloud services

- Introduction to AWS
  - What is Cloud?
  - Understanding Cloud
    - IaaS
    - PaaS
    - SaaS
  - Benefits of Cloud
  - What is AWS?
  - Understanding AWS
    - Ways to Operate AWS Services
- Dealing with AWS Certification
  - Three Levels of AWS Certification
  - How we Approach AWS Certification?

## Roadmap

Understand the Services, Devops integration services and Otherservices roadmap of AWS

- Key Facts about AWS Certification
- Our AWS Roadmap
  - Core Services
  - Container Services
  - Notification Services
  - CI/CD Services
  - Email Services
  - Infrastructure Provisioning Services
  - Database Services
  - Serverless Services
  - Deployment Services
  - Other Services
- History of AWS

## AWS Key Services

Understanding of the key services of AWS and the Application of them

- Value Proposition of AWS
  - Agility
  - Cost
  - Elasticity
  - Flexibility
  - Security
- Key Services in AWS
  - AWS IAM
  - Amazon VPC
  - Amazon S3
  - Amazon CloudFront
  - Amazon EC2
  - Amazon Route53
  - Amazon ECR
  - Amazon ECS
  - Amazon EKS

# AWS End to End (IAM & VPC)

## AWS Key Services

Understanding of the key services of AWS and the Application of them

- Amazon SNS
- Amazon SES
- AWS CodeCommit
- AWS CodeDeploy
- AWS CodePipeline
- AWS CloudFormation
- Amazon RDS
- Amazon DynamoDB
- AWS Lambda
- Amazon API Gateway
- AWS OpsWorks
- Overview of AWS
  - Architecture of AWS Cloud Infrastructure
- FAQ on AWS



## IAM Overview

Experience with the AWS core components with handson

- Prerequisites for AWS
  - Create AWS Account
  - Install AWS CLI
  - Install Python AWS SDK
  - Understanding JSON
- AWS IAM
  - What is the Purpose of IAM?
  - Key Components in IAM
  - IAM User
  - What is the Purpose of IAM User?
  - Create IAM User
  - Configure CLI for IAM User
  - Enable MFA for IAM User



## IAM Deep Dive

In this module you will be experiencing Identity Access Management, policy creation, group & role management

- IAM Policy
  - What is the Purpose IAM Policy?
  - Types of IAM Policy
  - Create IAM Policy
- IAM Group
  - What is the Purpose of IAM Group?
  - Create IAM Group
- IAM Role
  - What is the Purpose IAM Role?
  - Create IAM Role
- IAM Access Analyzer
  - What is the Purpose IAM Access Analyzer?
  - Create IAM Access Analyzer
- Do's and Dont's with IAM



# AWS End to End (IAM & VPC)



## AWS VPC

Learn about Virtual Private Cloud - network, internet gateway, route table

- Amazon VPC
  - What is the Purpose of VPC?
  - Key Components in VPC
  - Understanding VPC
  - Architecture of VPC-managed Network
  - VPC Network
    - What is the Purpose of VPC Network?
    - Create VPC Network
- VPC Internet Gateway
  - What is the Purpose of VPC Internet Gateway?
  - Create VPC Internet Gateway



## AWS VPC Handson

Learn about Virtual Private Cloud – Subnet, Access Control Lists, Security Groups etc

- VPC Route Table
  - What is Purpose of VPC Route Table?
  - Create VPC Route Table
- VPC Subnet
  - What is the Purpose of VPC Subnet?
  - Types of VPC Subnet
  - Create VPC Subnet
- VPC Network ACL
  - What is the Purpose of VPC Network ACL?
  - Update VPC Network ACL
- VPC Security Group
  - What is the Purpose of VPC Security Group?



## AWS VPC Handson

Learn about Virtual Private Cloud – Endpoint, DHCP etc.,

- Create VPC Security Group
- VPC Endpoint
  - What is the Purpose of VPC Endpoint?
  - Types of VPC Endpoint
  - Create Gateway VPC Endpoint
- VPC DHCP Options Set
  - What is the Purpose of VPC DHCP Options Set?
- Do's and Dont's with VPC
- Amazon S3
  - What is the Purpose of S3?
  - Key Components in S3
  - Understanding S3

# AWS End to End (S3, Cloudfront, EC2)



## AWS S3

Learn about the Simple Storage Service of AWS – Buckets, Objects, Encryption, Versioning, Security

- **S3 Bucket**
  - What is the Purpose of S3 Bucket?
  - Create S3 Bucket
  - Types of Storage Class in S3 Bucket
  - Upload Objects to S3 Bucket
  - Optimize S3 Bucket for High Scalability
  - How S3 Bucket Encryption Works?
  - Version Objects in S3 Bucket
  - Manage Object Lifecycle in S3 Bucket
- **S3 Access Point**
  - What is the Purpose of S3 Access Point?



## AWS EC2, SNS & Route53

Learn the network and compute components

- Create S3 Access Point for VPC
- Create S3 Access Point for Internet
- Do's and Dont's with S3
- **Amazon CloudFront**
  - What is the Purpose of CloudFront?
  - Key Components in CloudFront
  - Architecture of CloudFront-managed CDN
  - CloudFront Origin Access Identity
    - What is the Purpose of CloudFront Origin Access Identity?
    - Create CloudFront Origin Access Identity



## AWS CLOUDFront

Learn about CloudFront content delivery network access identity, distribution and enablement

- **CloudFront Distribution**
  - What is the Purpose of CloudFront Distribution?
  - Create CloudFront Distribution
  - Clear Cache from CloudFront Distribution
- Do's and Dont's with CloudFront
- **Amazon EC2**
  - What is the Purpose of EC2?
  - Key Components in EC2
  - EC2 Key Pair
    - What is the Purpose of EC2 Key Pair?
  - Create EC2 Key Pair

# AWS End to End (AMI, EBS, ELB, Route 53, ECR)



## Amazon EC2

Learn about the Elastic Cloud Compute feature of Elastic Load Balancer

- EC2 AMI
  - What is the Purpose of EC2 AMI?
  - Create EC2 AMI
- EC2 EBS
  - What is the Purpose of EC2 EBS?
  - Types of EC2 EBS
  - Create EC2 EBS Volume
  - Create Snapshot of EC2 EBS Volume
- EC2 Instance
  - What is the Purpose of EC2 Instance?
  - Types of EC2 Instance
  - Launch EC2 Instances



## Amazon Route 53

Learn about the AWS Route 53 a highly available and scalable cloud Domain Name System (DNS) web service

- Attach EBS Volume to EC2 Instance
- Using EC2 Instances as Ansible Nodes
- Prepare EC2 Instances for Ansible-managed Deployment
- Trigger Ansible-managed Deployment on EC2 Instances
- EC2 ELB
  - What is the Purpose of EC2 ELB?
  - Types of EC2 ELB
  - Create EC2 ELB
- Do's and Dont's with EC2



## Amazon ECR

Amazon Elastic Container Registry Docker container registry to store, manage, and deploy Docker container images

- Amazon Route53
  - What is the Purpose of Route53?
  - Key Components in Route53
  - Route53 Hosted Zone
    - What is the Purpose of Route53 Hosted Zone?
    - Create Route53 Hosted Zone
    - Add Record Sets in Route53 Hosted Zone
  - Do's and Dont's with Route53
- Amazon ECR
  - What is the Purpose of ECR?
  - Key Components in ECR
  - Prerequisites for ECR
    - Update IAM Group to Allow Access to ECR
    - Setup & Configure Docker

# AWS End to End (ECR, ECS, EKS)



## Amazon ECR

Amazon Elastic Container Registry Docker container registry to store, manage, and deploy Docker container images

- Pull Docker Images from Docker Hub
- ECR Repository
  - What is the Purpose of ECR Repository?
  - Create ECR Repository
  - Push Docker Images to ECR Repository
- Do's and Dont's with ECR
- Amazon ECS
  - What is the Purpose of ECS?
  - Key Components in ECS
  - Prerequisites for ECS
    - Update IAM Role to Allow Access to ECS
    - Create VPC Subnet for ECS



## Amazon ECS

Elastic Container Service is a highly scalable, fast, container management service to run, stop, and manage Docker containers on a cluster of EC2

- Create VPC Security Group for ECS
- ECS Cluster
  - What is the Purpose of ECS Cluster?
  - Create ECS Cluster
  - Using EC2 Instances as Container Instances in ECS Cluster
  - Boot Container Instances for ECS Cluster
  - Deploy Container Agent on Container Instances of ECS Cluster
  - Prepare Container Instances in ECS Cluster



## Amazon EKS

Amazon's fully managed Kubernetes Service for most sensitive, critical, secured, reliable and scalable service

- ECS Task Definition
  - What is the Purpose of ECS Task Definition?
  - Create ECS Task Definition
- ECS Service
  - What is the Purpose of ECS Service?
  - Create ECS Service
- Do's and Dont's with ECS
- Amazon EKS
  - What is the Purpose of EKS?
  - Key Components in EKS
  - Prerequisites for EKS
    - Create IAM Role to Allow Access to EKS
    - Update IAM Role to Allow Access to EKS

# AWS End to End (EKS, SNS & SES)



## Amazon EKS

Amazon's fully managed Kubernetes Service for most sensitive, critical, secured, reliable and scalable service

- Create VPC Subnet for EKS
- Create VPC Security Group for EKS
- EKS Cluster
  - What is the Purpose of EKS Cluster?
  - Create EKS Cluster
- EKS Node Group
  - What is the Purpose of EKS Node Group?
  - Using EC2 Instances as Nodes in EKS Node Group
  - Create EKS Node Group
  - Prepare Nodes in EKS Node Group



## Amazon SNS

Amazon Simple Notification Service is the messaging service for both app to app (A2A) and appl to person (A2P) communication

- EKS Control Plane
  - What is the Purpose of EKS Control Plane?
  - Install Kubernetes Client
  - Configure Kubernetes User to Access EKS Control Plane
  - Manage EKS Control Plane
  - Do's and Dont's with EKS
- Amazon SNS
  - What is the Purpose of SNS?
  - Key Components in SNS
  - Prerequisites for SNS
    - Update IAM Group to Allow Access to SNS



## Amazon SES

Simple Email Service (**SES**) is a cost-effective, flexible, and scalable email service that enables developers to send mail from any application

- SNS Topic
  - What is the Purpose of SNS Topic?
  - Create SNS Topic
- SNS Subscription
  - What is the Purpose of SNS Subscription?
  - Create SNS Subscription
  - Send Email to SNS Subscriptions
- Do's and Dont's with SNS
- Amazon SES
  - What is the Purpose of SES?
  - Key Components in SES
  - Prerequisites for SES
    - Update IAM Group to Allow Access to SES

# AWS End to End (CodeCommit & CodeDeploy)

## Amazon CodeCommit

CodeCommit service that hosts secure Git-based repositories, help teams to collaborate on code in a secure and highly scalable ecosystem

- SES Identity
  - What is the Purpose of SES Identity?
  - Verify Email Address as SES Identity
  - Send Email using SES Identity
- Do's and Dont's with SES
- AWS CodeCommit
  - What is the Purpose of CodeCommit?
  - Key Components in CodeCommit
  - Prerequisites for CodeCommit
    - Update IAM User to Allow Access to CodeCommit
    - Update IAM Group to Allow Access to CodeCommit

## Amazon CodeDeploy

AWS CodeDeploy for rapid release of new features, avoid downtime during application deployment & handles the complexity of updating code applications

- Update IAM Role to Allow Access to CodeCommit
- CodeCommit Repository
  - What is the Purpose of CodeCommit Repository?
  - Create CodeCommit Repository
  - Push Source Code to CodeCommit Repository
- Do's and Dont's with CodeCommit
- AWS CodeDeploy
  - What is the Purpose of CodeDeploy?
  - Key Components in CodeDeploy
  - Prerequisites for CodeDeploy



## Amazon CodeDeploy

Deployment Group with EC2 code deployment

- Create IAM Role to Allow Access to CodeDeploy
- CodeDeploy Application
  - What is the Purpose of CodeDeploy Application?
  - Create CodeDeploy Application
- CodeDeploy Deployment Group
  - What is the Purpose of CodeDeploy Deployment Group?
  - Create CodeDeploy Deployment Group
  - Launch EC2 Instances for CodeDeploy Deployment Group
  - Deploy Agent on EC2 Instances of CodeDeploy Deployment Group

# AWS End to End (Code Pipeline, CloudFormation, RDS)

## Amazon CodePipeline

Amazon Simple Notification Service is the messaging service for both app to app (A2A) and app to person (A2P) communication

- Prepare EC2 Instances in CodeDeploy Deployment Group
- Do's and Dont's with CodeDeploy
- AWS CodePipeline
  - What is the Purpose of CodePipeline?
  - Key Components in CodePipeline
  - Prerequisites for CodePipeline
  - Architecture of CodePipeline-managed CI/CD
  - CodePipeline Pipeline
    - What is the Purpose of CodePipeline Pipeline?
    - Create CodePipeline Pipeline



## Amazon CloudFormation

AWS CloudFormation help to collect AWS and third-party resources, provision and manage them throughout their lifecycles, by treating infrastructure as code

- Execute CodePipeline Pipeline
- Do's and Dont's with CodePipeline
- AWS CloudFormation
  - What is the Purpose of CloudFormation?
  - Key Components in CloudFormation
  - CloudFormation vs Terraform
  - CloudFormation Template
    - What is the Purpose of CloudFormation Template?
    - Structure of CloudFormation Template
    - Writing CloudFormation Template



AWS Cloud Formation



Amazon RDS

## Amazon RDS

Amazon Relational Database Service makes it easy to set up, operate, and scale a relational database in the cloud

- CloudFormation Stack
  - What is the Purpose of CloudFormation Stack?
  - Create CloudFormation Stack
- Do's and Dont's with CloudFormation
- Amazon RDS
  - What is the Purpose of RDS?
  - Key Components in RDS
  - Prerequisites for RDS
  - RDS Instance
    - What is the Purpose of RDS Instance?
    - Create RDS Instance
  - Populate Data to RDS Instance
  - Query RDS Instance

# AWS End to End (DynamoDB, Lambda)



## Amazon DynamoDB

Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale

- Optimize RDS Instance for High Scalability
- Backup and Restore RDS Instance
- Do's and Dont's with RDS
- Amazon DynamoDB
  - What is the Purpose of DynamoDB?
  - Key Components in DynamoDB
  - Prerequisites for DynamoDB
    - Update IAM Group to Allow Access to DynamoDB
  - DynamoDB Table
    - What is the Purpose of DynamoDB Table?
    - Create DynamoDB Table



## Amazon Lambda

Lambda lets you run code without provisioning or managing servers. Pay only for the compute time you consume

- Populate Data to DynamoDB Table
- Optimize DynamoDB Table for High Scalability
- Query and Scan DynamoDB Table
- Backup and Restore DynamoDB Table
- Do's and Dont's with DynamoDB
- AWS Lambda
  - What is the Purpose of Lambda?
  - Key Components in Lambda
  - Prerequisites for Lambda
    - Create DynamoDB VPC Endpoint to Connect from Lambda



## Amazon Lambda

Handson on the Lambda function writing using Python with configuration

- Create IAM Role to Allow Access to Lambda
- Lambda Function
  - What is the Purpose of Lambda Function?
  - Create Lambda Function
  - Add Python Code to Lambda Function
  - Add Environment Variables to Lambda Function
  - Configure Test Event for Lambda Function
- Do's and Dont's with Lambda

# AWS End to End (API Gateway & Ops Works)

## Amazon APIGateway

Amazon APIGateway helps developers to create, publish, maintain, monitor, and secure APIs at any scale

- Amazon API Gateway
  - What is the Purpose of API Gateway?
  - Key Components in API Gateway
  - API Gateway API
    - Purpose of API Gateway API?
    - Create API Gateway API
  - API Gateway Resource
    - Purpose of API Gateway Resource?
    - Create API Gateway Resource
    - Create Method in API Gateway Resource
  - API Gateway Stage
    - Purpose of API Gateway Stage?



## Amazon OPSWorks

OpsWorks helps Chef and Puppet to automate the servers configured, deployed, and managed in EC2 instances or on-premises

- Create API Gateway Stage
- Deploy API Gateway Stage
- Do's and Dont's with API Gateway
- AWS OpsWorks
  - What is the Purpose of OpsWorks?
  - Key Components in OpsWorks
  - Prerequisites for OpsWorks
    - Create IAM Role to Allow Access to OpsWorks
  - OpsWorks Stack
    - What is the Purpose of OpsWorks Stack?
    - Create OpsWorks Stack
  - OpsWorks Layer
    - Purpose of OpsWorks Layer?



## OpsWorks

## Amazon OPSWorks

Create & Deploy opsworks app with Casestudy and Interview Discussion

- Create OpsWorks Layer
- Launch EC2 Instances in OpsWorks Layer
- OpsWorks App
  - Purpose of OpsWorks App?
  - Create OpsWorks App
- OpsWorks Deployment
  - Purpose of OpsWorks Deployment?
  - Create OpsWorks Deployment
- Do's and Dont's with OpsWorks
- Case Study
  - Design & Implementation of SloopEngine Multi-Pod Architecture on AWS
  - Interview on AWS

<https://sloopengine.io/>



- ✓ *Managing Git Workflow on SloopEngine Git repository*
- ✓ *Simulating SloopEngine Multi-Pod Architecture on Docker*
- ✓ *Scaling Story of Leading Product Startup in Docker and AWS*
- ✓ *End to End project execution in line with SLOOP STASH project*
- ✓ *Project Example for Packaging*
- ✓ *Project Example Docker image creation manually*
- ✓ *Project Example Docker image creation using Dockerfile*
- ✓ *Project Example deployment using CICD(Jenkins) Docker and Kubernetes*
- ✓ *Containerizing various frameworks and Docker on AWS*
- ✓ *Using Git for version control and tracking of software*
- ✓ *Automating the IT infrastructure of a company*
- ✓ *Automating applications using different pipelines*
- ✓ *DevOps Continuous Integration*
- ✓ *Provision EC2 Virtual Machine with Ansible*
- ✓ *How to speed up the setting up and configure a software tool of an organization*
- ✓ *How to integrate the software projects deployed in diverse environments using Docker*
- ✓ *End to End integration of DEVOPS components in AWS Cloud*
- ✓ *Cloud based Hadoop/Spark Clustering using Devops Tools*



# Projects

Key Stuffs behind the success that provides real experience...



## THANK YOU

Instructor - *Mr. John Sundarraj*

Founder/CTO - *SloopStash*

<https://sloopstash.com>

<https://sloopengine.io>



*Inceptez Technologies Pvt. Ltd.* 

+91 70107 90330 

[info@inceptez.com](mailto:info@inceptez.com) 

[www.inceptez.com](http://www.inceptez.com) 

[www.inceptez.in](http://www.inceptez.in)

