

# Prerequisites (Free Training for Beginners. 2 weeks):

## Linux Basics:

- Basic Linux Commands
- User management in Linux
- Permissions
- Troubleshooting and Log management

## Linux Administration

- Package managers in Linux
- Introduction to Web Applications
- Apache and Apache modules
- PHP Web Application frameworks
- Application Servers and Tomcat
- Java Web application frameworks
- Database Servers - MySQL
- A sample Web application that run in Linux environment.

## Networking Basics

- Networking basics
- Internet and ISPs
- NAT gateways, Routers and Switches
- DNS
- Hosting your first website and running your webapplication

## Shell Scripting

- Linux Shell
- Introduction to Bash
- Introduction to Linux process
- Positional Parameters
- Conditional statements

- Loops in Bash
- Function
- GetOps
- Case statement
- Linux Administration using shell scripting
- Real time scripts for different system administration activities

## Python Programming

- Why Python?
- Introduction to Object oriented programming
- Interactive programming concepts
- Data Types
- Dictionary, List and Tuples
- Loop, Function, conditional statements
- Python modules and real time use
- Creating module in python
- Practical scripts and real time examples

# Build and Release Process

## Git / Github

- SCM - introduction
- Workflow of Developers in companies
- Git introduction ✓
- Local and remote repository
- Working copy and Local repository
- SSH access to Remote repository
- Git stash
- Tags
- Branches
- Best Branching practices
- How does it matters to your company
- How the whole workflow look like in a real project
- Introduction to Release process and different environments in companies
- The Continuous Integration process
- Github and advantages
- Github and collaborative development

## Ant

- Build Fundamentals
- XML scripting
- Properties and Variables
- ClassPath and Project orchestration
- Dependencies in Ant
- Building A java project using Ant.
- Introduction to artifacts and artifact repository.
- Introduction to Unit testing.

## Maven

- Introduction to Build Management
- executable code and their significance in DevOps

- Maven - Introduction
- Classes and unit test in Java
- First Project - demo
- Dependency management in Maven.
- Goals In Maven
- Repositories - Local and Remote
- Maven plugins
- Dependency and Plugin repositories
- SNAPSHOT in Maven
- Unit testing with Maven
- Report generation using Maven
- Integration with Jenkins
- How Maven helps in CI / CD
- Maven Release plugin
- The Project

## Jenkins

- Introduction to Continuous Integration and Continuous Delivery
- Introduction to Jenkins
- Plugins in Jenkins
- Your first Job in Jenkins
- Jenkins to build your actual projects
- Integration with Github, Maven, Nexus, Email, Dockers, Ant etc
- Triggers in Jenkins
- Jenkins master slave configuration
- Jenkins reporter plugin
- Upstream and Downstream projects
- Unit testing
- Fingerprint plugin
- Jenkins for release
- Build jobs
- Building the CI / CD pipeline and best practices
- Merging in Git and integration with Jenkins
- Artifact repositories and integration with Jenkins.
- Final Project

# Configuration Management and Monitoring

## Nagios

- Monitoring - Why its important?
- Nagios and Your real time infrastructure
- Hosts and Hostgroups
- Service and Service groups
- Host and Service templates
- Contacts and Contact groups
- Time periods and definitions
- Commands and definitions
- Nagios Plugins
- Writing a Nagios plugin
- Sample Nagios plugins to monitor your infrastructure.
- Nagios in real time
- Grouping your machines for monitoring. Best practices
- Automating your Nagios management.
- Integrating with other tools such as Jenkins, Ansible etc.

## Ansible

- Introduction to Configuration Management
- Ansible and SSH
- Ansible architecture
- Advantages over Other tools
- Ansible command line
- Playbooks in Ansible - demo
- Security
- Introduction to YAML
- Ansible inventory
- Ansible Modules

- Advanced Playbook usage
- Ansible Roles
- Roles and your production infrastructure
- Ansible Galaxy, Demo
- Config Release using Ansible, Git etc
- Autoscaling using Ansible
- Integration with Cloud/AWS
- Final Project

## Chef

- Introduction to Chef
- Comparison with Ansible
- Configuration management project - How it can be done on Chef

## Puppet

- Introduction
- Comparison with Ansible
- Configuration management project - How it can be done on Puppet.

# Containers and Orchestration

## Dockers

- Introduction to Virtualization
- Introduction to Containerization
- Containerization Vs Virtualization
- Docker Container
- Docker Images
- PHP Application Deployment using Docker.
- Life cycles of a container
- Introduction to Microservices
- Docker cluster - Swarm
- Services and Stack
- Docker-compose
- Build your container infrastructure
- Containerizing your existing Application
- Docker Networking
- Best Security Practices
- Docker images as your artifacts
- Release process using Dockers
- Integration with Jenkins

## Kubernetes

- Containerization and Clusters
- Kubernetes Architecture
- Setting up Clusters
- Deploying containerized applications.
- Modules
- Scaling up Applications
- Change Management
- Kubernetes API Server
- Master and Node components in Kubernetes

- apiserver, etcd, scheduler, controller managers and cloud controller managers
- Kubelet and Container runtime
- Addons
- Final Project

# Amazon Web Services

## EC2 - Virtual Machine management in AWS

- Introduction Cloud Computing and Virtualization
- AWS Global Architecture
- Instance types
- Elastic Block Storage (EBS)
- Key Pairs and Automation
- Elastic IP and Dynamic IP
- AMI, Snapshot
- Security Groups

## ELB - Elastic Loadbalancers

- Target Groups
- LoadBalancers
- Roundrobin and application performance.
- Running your web applications behind LoadBalancers
- Health Checks and High availability.
- Cross Zone Load Balancing.
- Grouping your machines for LoadBalancing, Best Practices.

## Autoscaling

- Scaling your applications based on Demand
- Significance of Autoscaling in modern infrastructures.
- Sample scenarios
- Autoscaling groups
- Launch Configurations
- Integration with Ansible.

## VPC

- Networking Concepts in AWS

- Traditional Networking and Cloud Networking.
- Building your first Network in the Cloud
- Internet Gateways in Cloud
- Private and Public Networks
- Subnets and Routing Tables
- ACLS, Security Groups
- NAT Gateways and their functions
- Security practices in VPC.
- Designing a tier 3 application infrastructure in cloud.

## Route 53

- Hosted Zone
- Hosting your first website
- A, MX and CNAME records
- High Availability for your webapplications across multiple regions.
- Failover and Health Checks

## S3

- Object storages and significance.
- Storing your Objects in cloud
- Integrating your php / java applications with S3
- Static Website Hosting at cheap price
- Versioning
- S3 for Backup

## IAM

- Access management - Introduction
- Restricting your aws resources access.
- IAM Users
- IAM Groups
- IAM Roles

- Access Key and Secret Keys
- Integrating your running applications with AWS cloud.

## **RDS**

- Relational Databases - Introduction
- Introduction to PaaS
- Running your relational databases in Cloud
- Migrating your existing Databases to Cloud
- Security Practices to secure your data.

## **SES:**

- SMTP - Introduction
- SMTP as a service
- Sending out emails from your running applications

## **CloudWatch**

- Monitoring your AWS resources
- Significance of monitoring
- Monitoring and scaling your applications
- Metrics
- Alarms
- Events
- Cloudwatch Vs Nagios

## **CloudFront**

- Introduction to Content Delivery Networks.
- CDN as a service
- Running your website in Cloudfront for fast access by Internet
- DNS Load Balancing and CDNs.

## **AWS CLI Tools:**

- Automating your infrastructure.
- Importance of Automation in modern infrastructures.
- Build an infrastructure from your scripts.
- Best Automation practices followed by companies.

## **Python Integration**

- Manage your AWS cloud using a python program
- Python boto module for AWS automation.
- Importance of Python in cloud computing.
- Best practices.

## PROJECTS (Already Included in Training)

- 1) **DevOps CI/CD Pipeline project:**  
Build a pipeline for a java project, build and automate the series of steps a changed code need to undergo until its ready for production use
- 2) **DevOps Configuration Management Project:** Design a configuration management system to manage your production infrastructure activities and configurations
- 3) **Containerizing your Java Application:** Run your full stack Java application in docker containers.
- 4) **AWS Project:** Building a Highly available, Fault tolerant and Scalable infrastructure in the cloud to run your software services (java/php applications)