



LINUX SYSTEMS ADMINISTRATOR, DEVOPS ENGINEER, AND AWS ADMIN SESSION PROGRAM

Contact information: Feel free to reach out to us with any issues, concerns or clarification.

Instructors: Tia Leonard & Eric Kemvou

Phone number: 832-897-8630/832-342-0700

Email: info.devopseasylearning.com

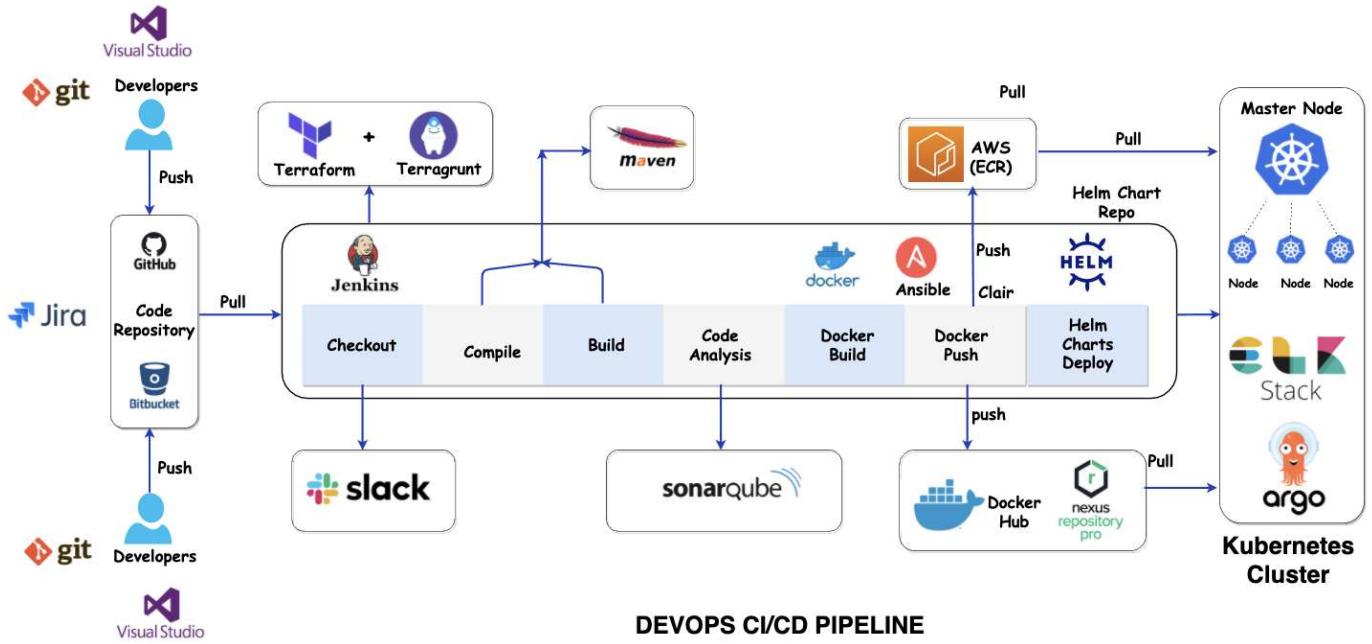
Website: devopseasylearning.com

About the Program:

This program is completely hands-on oriented and designed in a way to help you become a Linux Systems Administrator, DevOps Engineer, and AWS Admin. It will provide you with an in-depth knowledge of the following tools:

Configuration Management	Ansible
Scripting Language	Linux Shell Scripting, Makefile
Containerization and Image Scan	Docker
Continuous Integration (CI)	Jenkins
Build Automation and Code Analysis	Maven, Node, Golang, Sonarqube, Sonar-scanner
Version Control System (VCS)	Git, Github, Bitbucket
Artifact and package manager	AWS S3, Nexus, Helm for kubernetes, Yum, Apt
Monitoring	AWS CloudWatch, ELK, filebeat, Metricbeat, fluentd
Infrastructure As Code (IaC)	Terraform, Terragrunt, Jenkinsfile, docker-compose
Container Orchestration, Deployment strategy and Backup	Kubernetes, AWS EKS, GKE, RKE, Rancher, ArgoCD, Spinnaker, MetalLB
Container Registries	Docker Hub, AWS ECR (Elastic Container Registry), Nexus, Chart-Museum
Web Servers	Apache, Nginx, Tomcat
Bug Tracking and Documentation	Jira, Confluence
SDLC (Software Development Life Cycle)	Agile, Scrum, Waterfall
Linux Distributions	CentOS, RHEL and Ubuntu
Virtualization Platforms	VMware Workstation
Amazon Web Services	All DevOps and Cloud-Engineer Services
Databases	MySQL, Redis, Pgadmin, Postgres
Proxy and LoadBalancer	HA-proxy, Nginx reverse proxy-manager
Communication Tool	Slack
Distributed Systems	Redis-Cluster, Apache Kafka and Zookeeper, ElasticSearch-Cluster

PIPELINE EXAMPLE:



NB: Be advised that this training will be conducted online on Monday and Tuesday between the hours of **7pm - 10pm CST for 7 months via Zoom Meeting**. All enrolled students will be able to login into a dashboard on **devopseasytraining.com** website.

By the end of the course, you will have an in-depth knowledge on how to apply for the following jobs:

- ❖ Linux Systems Administrator
- ❖ Linux Systems Engineer
- ❖ DevOps Engineer
- ❖ DevOps Consultant
- ❖ AWS DevOps Engineer
- ❖ Cloud Support Engineer
- ❖ Cloud Automation Engineer
- ❖ Site Reliability Engineer (SRE)
- ❖ Cloud Support Engineer
- ❖ Platform Engineer

The salaries for these job titles range from **\$80k** to **\$180k** and upwards per a year.

PREREQUISITES :

You do not need to have any IT or information technology background because we will guide you step by step from the beginning through to the end. However, it will be a great idea to go through the IT foundational course using the links below to familiarize yourself with the components of a computer, network, AWS cloud computing, and Infrastructure as Code (**IaC**) using Terraform for AWS cloud. Additionally, your full commitment and passion will be needed to succeed in this course.

IT Courses Foundation:

- CompTIA A+ Certification Training Course: [link](#)
- CompTIA Network+ Training Course: [link](#)
- AWS Certified Cloud Practitioner Training Course: [link](#)
- Automate AWS Infrastructure Using Terraform: [link](#)

Computer Requirements for the Course:

To set up your lab environment for this course, you need to have the following:

- 01 Good laptop or desktop with a core i5 or i7, at least 8G of memory, 250G of HHD (hard disk drive), at least 2.2 GHZ of CPU.
- 01 Monitor at least 24 inches
- 01 External Keyboard
- 01 External mouse

PS: We will provide the whole infrastructure. You just need a computer and internet connection to access our platform.



Basic Software Installation

In this module, we will install some basic IT software to help in being more efficient during the duration of the course. Below software will be installed during the class session:

- Visual Studio Code
- Git Bash
- Google Chrome
- AWS Draw.io
- Putty
- MobaXterm
- Zoom
- WinRAR
- Slack

Introduction To Computer And Networking

What Will I Learn?

In this section, you will be introduced to basic general IT concepts and terminologies such as networking, computer hardware, software and IT terminology. Please exercise patience if you have a general understanding of these concepts as your colleagues familiarize themselves with it.

Learning Objectives:

- Computer software
- Computer hardware
- Networking (IP address, subnet, default gateway, DNS)
- Network devices, protocols, ports, firewalls, etc.
- IT Terminology

Linux System Administrator

What Will I Learn?

The Linux Fundamental part of the course will cover all the necessary concepts required for a Linux System Administrator to be fully functional in their role. We will cover Linux installation, configuration, patching, security, users and groups management, files and directories permission, networking concepts, system hardening and patching, disk management, service, and resources management, along with many others to be covered extensively during the duration of the course.

Learning Objectives:

- Introduction to Linux and Linux distributions
- Linux server installation (Redhat/Centos 6, 7 & 8, Ubuntu)
- Network configuration and linux files system

- Linux basic commands
- Linux files and directories structure
- Files, directories, and permissions management
- Users and groups management
- Memory and process management
- Linux boot up process
- Soft links and hard links
- Linux hardening, patching, and packages management
- Crontab and aliases
- Advance Linux command linux AWK, SET, CUT, PIPE, GREP, SCP
- Vim text editor
- Access Control Lists (ACLs)
- Set SUID, SGID and Sticky Bit
- Password Aging Policies
- Sudoers file management
- Apache or httpd, Nginx and Tomcat web server installation and configuration

Hands-On Exercise:

- Create files, directories, users, groups, etc.
- Copy files and directories from one server to another using scp command
- Host a website in Apache web server
- Manage process and resources utilization with top command, iostat, ps, vmstat, etc.
- Manage files using awk, vim, grep, pipe, cut, etc.
- Monitor log files in real time

Linux Shell Scripting

What Will I Learn?

A shell script is a text file that contains a sequence of commands for a UNIX-based operating system. It is called a shell script because it combines a sequence of commands, that would otherwise have to be typed into the keyboard one at a time, into a single file. In this section, we will cover bash shell scripting to automate processes for system configuration, management, patching, and hardening.

Learning Objectives:

- Introduction to shell scripting
- Type of shell in Linux
- Basic shell scripting concept
- Variables (global, local, environment, and special variables)

- Functions and arguments
- Loops (for loops and while loops)
- Shell decision making (if, elif, else, and case statements)
- Output and input redirection, exit code, read input, echo, shebang
- Shell operators (arithmetic operators, relational operators, boolean operators, string operators, and file test operators)

Hands-On Exercise:

- Create a shell script to configure and manage the iptables file
- Create a script to install applications into Kubernetes cluster
- Create a script to configure and patch new installed servers
- Create a script to monitor log files
- Create a script to install Helm charts into Kubernetes cluster
- Create a script to build and publish a docker image into docker hub and AWS ECR
- Create a script to backup Jenkins jobs into AWS S3 bucket

Amazon Web Services (AWS)

In this section, you will be introduced to AWS and all the other services offered under the AWS platform. Areas to be covered include EC3, IAM, S3, RDS, VPC, Load Balancing, Auto-Scaling, Route 53 and many others.

Overview And Introduction to AWS

What Will I Learn?

Amazon web service or AWS is a platform that offers flexible, reliable, scalable, easy-to-use, and cost-effective cloud computing solutions. In this section, you will learn about the different services provided by AWS. You will be provided with an overview of the important resources required for architecting an application.

Learning Objectives:

- What is Cloud Computing
- Why Cloud Computing?
- Advantages of Cloud Computing
- What is a Private Cloud
- What is a Public Cloud
- What is a Hybrid Cloud
- What is Software as a Service (SaaS)
- What is Platform as a Service(PaaS)
- What is Infrastructure as a Service (IaaS)

- Advantages of AWS
- AWS Global Infrastructure (Regions, availability zones, Edge location)
- AWS Management Console
- Ways to Access AWS Services
- Introduction to AWS Services

Hands-On Exercise:

- Sign-up for AWS free-tier account
- Create an S3 bucket through the console
- Launch an EC2 instance
- Create an IAM user
- Create a keypair

Identity And Access Management (IAM)

What Will I Learn?

AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny access to AWS resources. In this section, you will learn about security management in AWS using Identity Access Management (IAM).

Learning Objectives:

- User management through Identity Access Management (IAM)
- Various access policies across AWS Services
- API keys service access (access key and secret access key)
- Best practices for IAM
- Access billing and create alerts on billing
- Users and groups management
- IAM policies
- Introduction to Multi-Factor Authentication (MFA)

Hands-On Exercise:

- Create new users that can log in to AWS console
- Create policies for new users to have either admin or limited privileges
- Log in to AWS console via MFA
- Create API keys for accessing AWS Services
- Create a budget in the AWS console

AWS Object Storage (S3)

What Will I Learn?

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use, websites, mobile applications, backup and restore, archive, etc. In this section of the course, you will learn about the different Object storage services offered by AWS.

Learning Objectives:

- Introduction to S3 bucket
- Benefits of S3 bucket
- Version Control or versioning
- Replication
- S3 bucket security
- S3 bucket storage classes
- Life cycle policy in S3
- Cost optimization for S3
- Transfer acceleration
- Bucket policy

Hands-On Exercise:

- Create S3 bucket
- Upload files and folder into S3
- Create a bucket policy
- Create a bucket lifecycle policy
- Host a static website on Amazon S3
- Replicate data across regions
- Access a static website through CloudFront

Amazon Elastic Computer (EC2)

What Will I Learn?

EC2 (Elastic Compute Cloud) is the backbone of AWS. In this section, you will learn about the concepts associated with an EC2 instance and their usage. This section covers different Amazon AMIs, a demo on launching an AWS EC2 instance, ways to connect with an instance, and how to host a website on an AWS EC2 instance.

Learning Objectives:

- What is AWS EC2

- Amazon Machine Images (AMI)
- Working with AMIs
- Choosing an AMI
- Deciding/Finding an AMI
- EC2 instances – price model
- On-demand, reserved, scheduled, spot instances, dedicated hosts
- Elastic Cloud Computing Reserved Instance Marketplace
- Bootstrapping with user-data
- Security group
- Security with key pairs
- Working with security group
- Different IPs assigned to an elastic cloud computing instance
- Assigning Elastic IPs
- Start, stop and terminate an EC2 instance
- Login/access to EC2 instance using CLI and putty
- EC2 instance protection
- Instance roles
- Elastic Network Interfaces(ENIs)
- Resources and tags
- Difference between security group and access control list
- What is elastic block storage (EBS)
- Elastic block storage types
- What is EBS encryption

Hands-On Exercise:

- Host your website inside EC2
- Create an AMI
- Create an elastic IP
- Attach an EBS volume externally
- Create a snapshot
- Mount EFS volumes
- Create and delete volumes
- Attach and detach volumes
- Cross-Region snapshot copy & use cases
- Increase the volume size
- Create Volumes and AMIs from snapshots

Virtual Private Cloud (VPC) and Monitoring Services

What Will I Learn?

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. In this section of the course, you will be introduced to Amazon Virtual Private Cloud (VPC). Additionally, you will learn how to implement networking in AWS and how to monitor various instances and services.

Learning Objectives:

- What is AWS VPC?
- What is a default VPC?
- VPC – benefits and components
- What is Network and Access Control List (NaCl) and Security Groups?
- What is a subnet, route table, and association?
- NAT instances and NAT-Gateways
- VPC CIDR notations
- Network Access Control List v/s Security Groups
- NAT – network address translation
- VPC peering
- Overview of VPN and its components
- VPC Flow log
- AWS CloudWatch
- AWS CloudTrail
- Trusted Advisor

Hands-On Exercise:

- Set up the different types of networks in AWS
- Create a non-default VPC and attach it to an EC2 instance
- Access instance inside the private subnet using NAT gateway
- Create a Network and Access Control List (NACL)

Elastic Load Balancing (ELB)

What Will I Learn?

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets such as Amazon EC2 instances, containers, etc. It can handle the various loads of your application traffic in a single availability zone or across multiple availability zones. Elastic Load Balancing offers four types of load balancers that all feature high availability, automatic scaling, and robust security necessary to make your applications fault-tolerant. This section will help you understand how AWS ELB works, its benefits and how to implement it in the real world.

Learning Objectives:

- Introduction to load balancer

- What is ELB?
- How does ELB work?
- Types of ELB
- Difference between the application load balancer and classic load balancer
- What is a listener?
- Elastic load balancer and its types
- What is a sticky session?
- What is a health check?

Hands-On Exercise:

- Create a Classic Load Balancer
- Create security groups for the load balancer
- Add multiple instances to the load balancer
- Configure health check for the load balancer

Auto-Scaling and Route 53

What Will I Learn?

AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. In this section, you will learn about the concepts of Auto-Scaling and Route 53 to manage traffic.

Learning Objectives:

- Introduction to auto-scaling
- Introduction to auto Route 53
- Advantages of auto-scaling
- Launch configuration overview
- Components of Auto-Scaling
- Lifecycle of Auto-Scaling
- Auto-Scaling policy
- Working of Route 53
- Various Routing Policies

Hands-On Exercise:

- Create a launch configuration
- Create Auto Scaling Groups
- Attach & Detach EC2

Relational Database Service (RDS)

What Will I Learn?

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. In this section, you will learn about the different database services offered by AWS.

Learning Objectives:

- AWS RDS overview
- What is AWS RDS?
- Difference between a Relational Database and Non-Relational Database
- Why would you choose a Managed RDS?
- Amazon RDS types and their benefits
- RDS high availability and Multi-AZ deployment
- RDS security and pricing
- RDS backup and restore
- DB instance replication
- RDS snapshot
- Amazon DynamoDB and its benefits

Hands-On Exercise:

- Install Pgadmin and Mysql workbench and connect to the database in AWS
- Create RDS security group
- Create a MySQL DB and PostgreSQL (RDS) in the private subnet
- Create a DynamoDB table

Infrastructure as Code With Terraform

What Will I Learn?

Terraform is a multi-cloud open-source infrastructure as code (IaC) tool for building, managing, and versioning cloud services infrastructure safely and efficiently. In this section, you will use Terraform to create resources in AWS such as EC2, IAM, S3, RDS, VPC, Load Balancing, Auto-Scaling, Route 53, and many others.

Learning Objectives:

- Introduction to Terraform
- Difference between Terraform and AWS CloudFormation
- Why the need for Terraform as opposed to AWS CloudFormation for your organization?
- Terragrunt Installation
- Terraform providers
- Terraform basic commands

- Terraform variables and output
- Terraform modules
- Terraform state management
- Terraform conditions and for loops
- Terraform provisioners
- Terraform user data and resources
- Terraform registry
- Credential management in Terraform
- Terraform count, dynamic block, dependencies, alias and tags

Hands-On Exercise:

- Create a VPC, Security group, EC2, S3, load balancer, RDS Database, auto-scaling group, launch templates, secrets in secret manager and parameter storage for RDS, policies, bastion host
- Launch a Dev environment with Terraform in AWS
- Set up Jenkins in AWS using Terraform
- Integrate Terraform with Jenkins to deploy resources in AWS

Terraform Management With Terragrunt

What Will I Learn?

Terragrunt is a thin wrapper that provides extra tools for keeping Terraform configurations DRY, working with multiple Terraform modules, and managing remote states. In this section, you will learn how to use terragrunt to manage Terraform dependencies, manage Terraform remote state, manage multiple environments at the same time such as Dev, Stage, and Prod.

Learning Objectives:

- Introduction to Terragrunt
- Terragrunt Installation
- Terragrunt basic commands
- Manage a remote state using Terragrunt
- Use Terragrunt to manage Terraform dependencies
- Use terragrunt to manage multiple environments

Hands-On Exercise:

- Launch and manage multiple environments such as Dev and Prod using Terragrunt
- Create DynamoDB to lock the state file using Terragrunt
- Create DynamoDB to store the state file using Terragrunt
- Integrate Terragrunt with Jenkins

DevOps (Development And Operation)

What Will I Learn?

In this section of the training, you will be provided with an in-depth overview of the various DevOps tools listed on page 2

Overview Of DevOps

What Will I Learn?

DevOps is not just about tools but also a set of best practices that enable us to bridge the gap between development and operation teams in the areas of continuous integration and deployment by using an integrated set of tools to automate software delivery. In this section, we will go through DevOps and be introduced to the various DevOps tools.

Learning Objectives:

- Why DevOps?
- What is DevOps?
- DevOps Engineer Skills
- Introduction to DevOps tools
- DevOps terminology
- DevOps delivery pipeline overview

Agile Methodology

What Will I Learn?

Agile is a framework that defines how software development needs to be done. It's not a single or specific method, but a collection of various methodologies and best practices that follow the value statement signed with the customer. In this section, you will be introduced to Agile and Waterfall Methodology.

Learning Objectives:

- Why Agile?
- What is a waterfall?
- What is SDLC?
- Problems with waterfall methodologies
- Agile vs waterfall methodologies
- Sprint life Cycle in Agile
- Good understanding of story point, product backlog, sprint planning, sprint backlog, product owner, scrum team, scrum master, daily standup, story, epic, task, sprint review, sub-tasks

- Overview of Program Increment (PI) planning

Hands-On Exercise:

- Create a backlog
- Create a task, subtasks and story
- Create epics and assign stories to epics
- Calculate a team's PI for 2 weeks sprint increment

Version Control With Git, Bitbucket and Github

What Will I Learn?

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. In this section, we will learn how to use Git to manage the source code locally and to use Github and Bitbucket to store and manage source code as a remote repository.

Learning Objectives:

- What is version control?
- What is Git?
- Why Git for your organization?
- Install Git on Windows and Linux
- Difference between a centralized VCS and distributed VCS
- Install git on Windows and Linux
- Create a Github and Bitbucket accounts
- Create an SSH key for Github and Bitbucket authentication
- Understanding the difference between the feature branch and the release branch
- Set up git global configuration
- Git basic commands
- Working with Gitignore file
- Working with local and Remote Repositories
- Branching and Merging in Git
- Git workflows and pull request
- Understanding the difference between the feature, nonprod, qa, prod and master branch

Hands-On Exercise:

- Manage git merge conflict
- Branching and merging, Stashing, reverting and resetting
- Create and delete local and remote branches
- Create a pull request (PR) to merge code from a feature branch into a release branch
- Consolidate multiple branches into one branch

- Create a script to add, commit and push code from a local repository to a remote repository

Continuous Integration Using Jenkins

What Will I Learn?

Jenkins is an open-source continuous integration, continuous delivery, and deployment (CI/CD) automation software in DevOps written in a Java programming language. Jenkins manages and controls software delivery processes through the entire lifecycle, including build, document, test, package, stage, and deploy. It is mostly used to implement CI/CD workflows in DevOps. In this section, we will learn how to perform continuous integration (CI) and continuous deployment (CD) using Jenkins.

Learning Objectives:

- What is Continuous Integration (CI)?
- Why is Continuous Integration (CI) required?
- Introduction to Jenkins and Architecture
- Jenkins installation in VM, Docker and AWS
- Integrate Jenkins with Github and Bitbucket
- Jenkins Management
- Adding a slave node to Jenkins
- Change Jenkins default home directory
- Learn how to secure Jenkins
- Manage users and permissions using Matrix authentication plugins
- Install and manage plugins
- Building Delivery Pipeline
- Integrate Jenkins with Github to pull the source code automatically from VCS
- Create Multiple-branch jobs, Maven jobs, pipeline jobs and freestyle jobs
- Pipeline as a code using a declarative Pipeline (**Jenkinsfile**)
- Jenkins configuration as code
- Write a declarative pipeline using Groovy to build, test, package, push to docker hub and deploy applications
- Upgrade Jenkins image version and plugins
- Automate Jenkins backup jobs in AWS S3 using script
- Recover Jenkins from backup
- Send email notification if a deployment fails
- Integrate Jenkins with Splunk to monitor jobs

Hands-On Exercise:

- Build a pipeline using Jenkinsfile

- Build jobs on different slaves and automatically install software on Jenkins slaves
- Use a declarative pipeline to deploy an application in docker
- Deploy Jenkins in Kubernetes cluster as a pod
- Upgrade Jenkins Image version and plugins

Build Automation With Maven

What Will I Learn?

Maven is a build automation tool used primarily for Java projects. Maven is based on the concept of a project object model (POM), which helps to automatically manage a project's build, documentation, and reporting. In this section, you will learn how to use Apache Maven to compile, build, and test a Maven project using Jenkins Pipeline (**JenkinsFile**) as code.

Learning Objectives:

- Introduction to Maven
- Maven installation and configuration
- Understanding Maven architecture
- Integrating Jenkins with Maven
- Understanding the war and jar file
- Configuring Jenkins to automatically install Maven on slaves
- Understanding the pom.xml file and plugins
- Maven commands (mvn compile, test, package and so on)
- Build, compile, and package the application through CLI
- Build, compile, and package the application through Jenkins using Freestyle job and Maven Job

Hands-On Exercise:

- Write a Jenkins Declarative pipeline (**JenkinsFile**) to build, compile, test, package and deploy Maven project
- Build and package Maven projects through the command line

Containerization With Docker Docker-Compose

What Will I Learn?

Docker is an open platform for developing, shipping, and running applications. It enables you to separate your applications from your infrastructure software so it can be delivered quickly. With Docker, you can manage your infrastructure the same way you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production. In this

section, you will learn how to use Docker to containerize your application or to package your application and deploy it in any environment where Docker is installed.

Learning Objectives:

- Introduction to Docker
- What are the benefits of Docker
- Difference between Docker and VM
- Installation of docker in AWS and VM
- Create a Docker hub account
- Docker basic commands
- Understanding images and containers
- Understanding docker logs
- Container life cycle
- Sharing and copying
- Managing docker container
- Understanding what is Docker desktop and toolbox
- Understanding port forwarding and mapping in docker
- Understanding docker volume (volume mount and bind mount)
- Introduction to Docker Networking and types of network in Docker
- Publishing image on Docker hub and AWS ECR
- Introduction to DockerFile
- Base image for DockerFile
- Understanding all instructions in DockerFile (COPY, RUN, FROM, CMD, ARG, ENV, LABELS, WORKDIR, ENTRYPOINT, EXPOSE, ADD, etc.)
- Building, publishing images on Docker hub and AWS ECR
- Introduction to Docker-Compose and YMAL
- Pros and cons of Docker-Compose
- Docker-Compose installation
- Understanding all instructions in Docker-Compose such as (version, environment, expose, depends-on, links, build, container_name, image, ports, network, aliases, volumes. etc.)
-

Hands-On Exercise:

- Build docker images from Jenkins using a JenkinsFile and shell script
- Create a docker files to build and deploy applications into Tomcat, Nginx and Apache web server
- Write a script to build docker images and push it in different accounts such as non-prod and prod

- Launch an **Ecommerce** application using Docker-Compose with front-end, API and back-end
- Launch a restaurant application using Docker-Compose
- Launch a **Kaban** application using Docker-Compose
- Launch the whole lap environment user docker compose with tools such as:
 - Jenkins
 - Nexus
 - SonarQube
 - Postgres
 - Pgadmin
 - ELK (ElasticSearch, Logstash, and Kibana)
 - Grafana
 - Splunk
 - Redis cluster with sentinels

Configuration Management with Ansible

What Will I Learn?

Ansible is an open-source software provisioning, configuration management, and application-deployment tool which enables infrastructure as code (IaC). In this section, you will learn how to install and configure Ansible. You will also learn to write Ansible playbooks and roles, execute ad-hoc commands, manage inventory files, use Ansible to deploy in different environments, and manage multiple servers at the same time.

Learning Objectives:

- Introduction to Ansible
- Ansible installation in VM and AWS
- Ansible ad-hoc commands
- Run Ansible Ad-hoc commands via multiple servers
- Ansible inventories (static and dynamic inventories)
- Password authentication setup and explanation
- Introduction to Ansible playbooks
- Basic concepts to writing playbooks (working with different variables)
- Loops and handlers in Ansible
- Tomcat installation and configuration using playbook
- Ansible vault to protect Ansible playbooks with encryption
- Ansible roles

Hands-On Exercise:

- Executing ad-hoc commands

- Download and install roles
- Install Jenkins in AWS using Ansible role
- Integrate Ansible with Jenkins
- Use Ansible to build images, push to docker hub and deploy it to docker
- Use Ansible to install, configure and host a web site in httpd or Apache web server

Container Orchestration with Kubernetes

What Will I Learn?

Kubernetes is an open-source system for automating the deployment, scaling, and management of containerized applications. In this section, you will be introduced to Kubernetes architecture, objects, monitoring, and writing of declarative manifests to deploy applications in the Kubernetes cluster.

Learning Objectives:

- Introduction to Kubernetes
- Understanding container orchestration
- Understanding the difference between Kubernetes and Docker-compose
- Understanding Kubernetes Cluster Architecture
- Understanding Kubernetes objects
- Install kubectl, kubens, and Kubectx
- Setting up Kubernetes Cluster in AWS using Terraform
- Kubernetes basic commands
- Create pods, services, replica set and deployment using imperative approach or CLI
- Create pods, services, replica set and deployment using declarative approach or manifests
- Scale pods up and down using declarative and imperative approach
- Copy file and folder from local into a pod using kubectl
- Understanding namespace, scheduling, labels, selectors, taint, toleration and pods affinity
- Create namespace and deploy resources within a particular namespace
- Good understanding of Grafana, Prometheus, Alertmanager to monitor Kubernetes cluster
- Volume management in Kubernetes
- Set up resource limit for pods
- Troubleshoot a failed deployment
- Deployment strategy in Kubernetes
- Rolling Updates and rollbacks

Hands-On Exercise:

- Manage kubernetes cluster using kubectl, kubectx and kubens
- Create namespace and deploy applications within a namespace

- Create kubernetes manifest files
- Troubleshoot failed deployment
- Monitor real time logs through CLI

Application Deployment In Kubernetes with Helm

What Will I Learn?

Helm is a package manager for Kubernetes that allows developers and operators to easily package, configure, and deploy applications and services into Kubernetes clusters. In this section, you will learn how to package Kubernetes manifest files and deploy it into Kubernetes cluster using helm charts.

Learning Objectives:

- Introduction to Helm
- Helm installation
- Understanding Helm architecture
- Problem installing an application using Kubernetes kubectl command
- Why the need for Helm?
- Understanding the difference between Helm 2 and Helm 3
- Overview of Helm commands
- Overview of Helm repository
- Overview of Helm charts and structure
- Install Helm charts in Kubernetes cluster
- Uninstall Helm charts from Kubernetes cluster

Hands-On Exercise:

- Write Helm charts
- Install metric server in Kubernetes cluster using Helm charts
- Install Jenkins in Kubernetes cluster as a pod using Helm charts

Manage Workflow with Jira

What Will I Learn?

Jira is a proprietary issue tracking product developed by Atlassian that allows bug tracking, issue tracking, and agile project management. In this section, you will learn how to use the Jira story to manage team project evolution.

Learning Objectives:

- Introduction to Jira

- Installing Jira
- Jira workflow
- Create a backlog
- Overview of a story point
- Starting and stopping a sprint
- Define a story point

Hands-On Exercise:

- Create a task, a subtasks, a story, and assign it to a team member
- Create epics and assign stories to epics
- Manage a backlog

Interview Preparation And Job Application

In this section, you will learn how to approach interview questions, how to write your resume based on some IT professional resume samples, how to create your LinkedIn profile and also how to communicate with recruiters.

Learning Objectives:

- Type of recruiters
- Resume build
- LinkedIn profile setup
- Interview questions and answers
- Job posting websites

At the end of each tool mentioned on page 2, and in-class live interview session video will be provided about commonly asked interview questions about that tool and also a file with at least 50 questions and answers will be provided and available on the website.

Mandatory:

Students will be paired in a group of two and will be required to interview each other and upload audios of those interviews on their student accounts to receive instructor feedback. Doing so, will help instructors prepare students in an early stage on how to interview to increase their chances to land an offer letter before the end of the training.

Possible Certifications To Be Taken At The End

At the end of this training, you will be able to take the following certifications if you can check out some extra resources and also put in some additional effort.

- ❖ Red Hat Certified System Administrator (RHCSA) provided by Red Hat
- ❖ AWS Certified Cloud Practitioner provided by Amazon

- ❖ AWS Certified Solutions Architect Associate provided by Amazon
- ❖ Terraform Certified Associate provided by Hashicorp
- ❖ Certified Kubernetes Administrator (**CKA**)