Linux

Linux Course Outline

Module 1: Introduction to Server & Linux

- **Client-Server Architecture**
  - Concepts and examples
  - Roles of client and server in a network
- **GUI vs CLI**
  - Differences and use cases
  - Advantages of CLI for server management
- **Navigating through CLI**
  - Basic commands: `pwd`, `cd`, `ls`
  - Understanding the file system hierarchy
- **Getting Help**
  - Using `man` pages
  - Help commands like `--help` and `info`

Module 2: File Hierarchy System (FHS)

- **Using Relative Path and Absolute Path**
  - Understanding and using paths
- **Common File Types**
  - Regular files, directories, links,

Module 3: Basic File Management

- **Create Files and Directories**
  - `touch` and `mkdir`
- **File and Directory Operations**
  - Copy (`cp`), paste, remove (`rm`), move (`mv`), rename
- **File Editing**
  - Introduction to text editors (vim)

Module 4: Basic User Management

- **User Login Activity and Information**
  - Viewing login records (`last`, `who`, `w`)
- **Local User Authentication Files**
  - Understanding `/etc/passwd`, `/etc/shadow`
- **User Account Management**

- ○ `useradd`, `usermod`, `userdel`
- ○ Custom configurations and user profiles

Module 5: Advanced User Management

- **Setting Passwords**
  - ○ Using `passwd` command
- **Linux Groups and Their Management**
  - ○ Group management commands: `groupadd`, `groupmod`, `groupdel`
- **Substituting Users and Super User (SUDO)**
  - ○ Configuring and using `sudo`
  - ○ Security best practices for sudoers\

Module 6: Advanced File Management

- **File and Directory Access Permissions**
  - ○ Understanding and setting permissions with `chmod`
  - ○ Symbolic and numeric modes

Module 7: Variables

- ○ Environment variables and shell variables
- ○ Variable substitution
- **Command Substitution**
  - ○ Using backticks and `$(command)`

Module 8: Linux Automation and Task Scheduling

- **Job Scheduling**
  - ○ One-time jobs with `at`
  - ○ Recurring jobs with `cron`
  - ○ Managing `cron` jobs (`crontab`)

Module 9: Package Management and Systemctl

- **Installing Packages**
  - ○ Using package managers (e.g., `apt`, `yum`)
- **Uninstalling Packages**
  - ○ Removing packages with package managers
- **Managing Services with `systemctl`**
  - ○ Starting, stopping, enabling, and disabling services
  - ○ Checking service status

AWS

Module 1: Introduction to Cloud Computing

- **What is Cloud Computing**
    - Definition and key concepts
    - On-demand availability and scalability
- **Cloud Implementation Models**
    - Public, Private, and Hybrid Clouds
- **Advantages of Cloud Computing**
    - Cost efficiency, flexibility, disaster recovery, and more

Module 2: Getting Started with AWS

- **Creating an AWS Free Tier Account**
    - Account setup and initial configuration
- **AWS Global Infrastructure**
    - Overview of Regions, Availability Zones & Edge locations
- **Tour of the AWS Console & Services in AWS**
    - Navigation and key services overview

Module 3: Elastic Compute Cloud (EC2)

- **Creating EC2 Instances**
    - Instance types, AMIs, and instance launch
- **Security Groups & Classic Ports**
    - Configuring security groups and common port settings
- **SSH Overview**
    - Key pair creation and SSH connection setup
- **Private vs Public vs Elastic IP**
    - IP addressing and usage

Module 4: Identity and Access Management (IAM)

- **IAM Introduction**
    - Core IAM concepts and architecture
- **Managing Users and Groups**
    - Creating and managing IAM users and groups
- **IAM Group Policies and Inline Policies**
    - Difference and use cases
-
- **AWS CloudShell**
    - Introduction and use cases
- **IAM Best Practices**
    - Security best practices and guidelines
- **AWS Policy Simulator**
    - Testing and validating policies

Module 5: EC2 Instance Storage

- **EBS Overview**
    - Types and uses of Elastic Block Store
- **Managing EBS Volumes**
    - Creating, attaching, and resizing volumes
- **EBS Snapshots**
    - Creating and managing snapshots
- **Cross-AZ and Regional Replication for EBS Volumes: Best Practices**
    - Configuring replication for high availability
- **EBS Encryption**
    - Setting up encryption for data security
- **Amazon Machine Images (AMI)**
    - Creating and managing AMIs
- **Amazon EFS**
    - Setting up and using Elastic File System
- **EFS Lifecycle Management**
    - Managing the lifecycle of files in EFS
- **EBS vs EFS**
    - Comparison and use cases

Module 6: Simple Storage Service (S3)

- **Introduction to S3**
    - Overview and use cases
- **S3 Buckets and Objects**
    - Creating and managing buckets and objects
- **S3 Versioning**
    - Enabling and managing versioning

- **S3 Security and Bucket Policies**
  - Setting up bucket policies and access control
- **Static Website Hosting**
  - Hosting a static website on S3
- **Cross-Region and Same-Region Replication**
  - Configuring replication for data redundancy
- **S3 Storage Classes**
  - Overview of storage classes and cost optimization
- **Configuring S3 Lifecycle Rules**
  - Automating transitions and expiration

Module 7: High Availability and Scalability

- **High Availability and Scalability**
  - Core concepts and benefits
- **Introduction to Load Balancing**
  - Basic principles and use cases
- **Different Types of Load Balancers in AWS**
  - Classic Load Balancers , Application Load Balancers
- **Elastic Load Balancer**
  - Configuration and management
  - Sticky Sessions (Cookies)
  - Cross-Zone Load Balancing
  - Connection Draining
- **Auto Scaling Groups**
  - Setting up and configuring ASGs
- **Scaling Policies**
  - Dynamic scaling policies
- **Scaling Triggers with CloudWatch Alarms**
  - Configuring alarms for auto-scaling

Module 8: DNS and Route 53

- **What is DNS**
  - Basic DNS concepts
- **Route 53 Overview**
  - Features and benefits
- **Registering a Domain**
  - Domain registration and management
- **Creating Records**
- **Integrating 3rd Party Domains with Route 53**
  - Setting up external domains

Module 9: Cloud Networking and VPC

- **Introduction to Networking**
  - Basic networking concepts in the cloud
- **CIDR**
  - Understanding and using CIDR notation
- **Creating Public and Private Subnets**
  - Subnet creation and management
- **Creating Public and Private VPCs**
  - VPC setup and configuration
- **Creating a Custom VPC**
  - Advanced VPC configurations
- **Creating and Managing Internet Gateways**
  - Setting up internet connectivity
- **VPC Peering**
  - Configuring peering connections between VPCs

Module 10: CloudFront and Global Accelerator

- **Introduction to CloudFront**
  - Content delivery and caching
- **Setting Up CloudFront Distributions**
  - Configuring and managing distributions
- **Global Accelerator Overview**
  - Improving performance and availability

Module 11: Budgets and Cost Management

- **AWS Budgets Overview**
  - Setting up and managing budgets
- **Cost Management Tools**
  - Using AWS Cost Explorer and reports
- **Cost Optimization Best Practices**
  - Strategies for reducing AWS costs

Git and GitHub

Module 1: Introduction to Version Control

- **What is VCS and Its Advantages?**
  - Introduction to Version Control Systems (VCS)
  - Benefits of using VCS (e.g., collaboration, version tracking, backup)
- **Installing and Configuring Git**
  - Downloading and installing Git on various operating systems
  - Initial configuration (`git config` for username, email, etc.)
    - `git config --global user.name "Your Name"`
    - `git config --global user.email "your.email@example.com"`
- **Creating a GitHub Account**
  - Setting up a GitHub account
  - Overview of GitHub features and interface
- **Familiarizing with Git Bash**
  - Basic command-line interface commands
  - Navigating directories, and basic file operations in Git Bash
    - Commands: `pwd`, `ls`, `cd`, `touch`, `mkdir`, `rm`

Module 2: Demystifying the Git Workflow

- **Working Area**
  - Understanding the working directory
  - Tracking changes with `git status`
  - Viewing differences with `git diff`
- **Staging Area**
  - Adding changes to the staging area with `git add`
  - Viewing staged changes with `git diff --staged`
- **Local Repository**
  - Committing changes with `git commit`
  - Viewing commit history with `git log`
    - Commands: `git commit -m "commit message"`, `git log`

Module 3: Remote Repositories

- **Managing Remote Repositories**
  - Connecting to remote repositories with `git remote`
  - Viewing and managing remote connections
    - Commands: `git remote add origin <URL>`, `git remote -v`
- **Pushing Code to Remote Repositories**
  - Pushing changes with `git push`

- ○ Understanding the push workflow and upstream branches
    - ■ Command: `git push origin main`
- **Cloning Repositories**
    - ○ Cloning repositories with `git clone`
    - ○ Working with cloned repositories
        - ■ Command: `git clone <URL>`

Module 4: Branching & Merging Repositories

- **Creating & Managing Branches**
    - ○ Creating branches with `git branch` and `git checkout`
    - ○ Switching branches with `git switch`
    - ○ Viewing branches with `git branch --list`
        - ■ Commands: `git branch <branch-name>`, `git checkout <branch-name>`, `git switch <branch-name>`
- **Merging Branches**
    - ○ Merging branches with `git merge`
    - ○ Understanding fast-forward and recursive merges
        - ■ Command: `git merge <branch-name>`
- **Resolving Merge Conflicts**
    - ○ Identifying merge conflicts
    - ○ Manually resolving conflicts and committing the resolution

Jenkins Course Outline

Module 1: Introduction to Jenkins & Installation

- **Introduction to Jenkins**
    - ○ What is Jenkins and its benefits in CI/CD
    - ○ Overview of Jenkins architecture
- **Installation**
    - ○ Installing Jenkins on various operating systems
    - ○ Initial setup and configuration
    - ○ Securing Jenkins with basic security settings

Module 2: Building Your First Job on Jenkins

- **Introduction to Jenkins Dashboard**

- ○ Overview of the Jenkins interface
- ○ Key components of the dashboard
- **Creating Your First Job**
  - ○ Job types and their uses
  - ○ Step-by-step guide to creating a freestyle project
  - ○ Configuring job settings
- **Redirecting Your Job Output to a File**
  - ○ Configuring build steps to redirect output
  - ○ Managing build logs and artifacts

Module 3: Creating Users in Jenkins & Role-Based Access

- **Creating Various Users**
  - ○ Adding and managing users in Jenkins
- **Installing Required Plugins**
  - ○ Identifying and installing essential plugins for user management
- **Assigning Roles**
  - ○ Setting up role-based access control (RBAC)
  - ○ Assigning roles and permissions to users

Module 4: Jenkins with Email

- **Configuring Jenkins to Send Mail**
  - ○ Setting up email notifications
  - ○ Configuring SMTP server settings
- **Sending Mail to Multiple Accounts**
  - ○ Creating email notifications for job results
  - ○ Configuring email notifications for multiple recipients
  - ○

Module 5: Upstream & Downstream Jobs, Periodic Jobs

- **Configuring Upstream & Downstream Jobs in Jenkins**
  - ○ Setting up job dependencies
  - ○ Managing job execution order
- **Configuring Scheduled Jobs in Jenkins**
  - ○ Setting up cron jobs for periodic builds
  - ○ Managing build schedules

Module 6: Jenkins & Git

- **Introduction to Git Plugin**
  - ○ Installing and configuring the Git plugin
- **Git with Poll SCM**
  - ○ Setting up SCM polling to trigger builds
  - ○ Configuring Git repository integration

- **Remote Triggers with Git**
  - Setting up remote build triggers using Git

Module 7: Jenkins Pipeline as Code & Jenkinsfile

- **Introduction to Jenkinsfile**
  - Understanding Pipeline as Code concepts
  - Jenkinsfile syntax and structure
- **Writing Your First Pipeline as Code**
  - Creating a simple pipeline script
  - Using declarative and scripted pipelines

Module 8: Jenkins & Terraform Pipeline (Project)

- **Creating an EC2 Instance with Jenkins Using Terraform**
  - Introduction to Infrastructure as Code (IaC) with Terraform
  - Setting up Terraform on Jenkins
  - Writing a Jenkins pipeline to provision an EC2 instance using Terraform

Terraform Course Outline

Module 1: Introduction to Infrastructure as Code (IaC)

- **Introduction to IaC**
  - Definition and benefits of IaC
  - Comparison of IaC tools
- **Understanding IaC Concepts**
  - Declarative vs. imperative configurations
  - Overview of Terraform architecture
- **Installing Terraform on Servers**
  - Step-by-step installation on various operating systems
  - Verifying installation
- **Setting up Visual Studio Code**
  - Installing and configuring Visual Studio Code
  - Installing Terraform extensions for VS Code

Module 2: Deploying Infrastructure with Terraform

- **Authentication and Authorization**
  - Setting up credentials for cloud providers
  - Managing access with IAM roles and policies
- **Launching First VM through Terraform**
  - Writing the first Terraform configuration file
  - Initializing and applying configurations to launch a VM
- **Provider Tiers**

- ○ Understanding different providers and their configurations
- ○ Configuring provider-specific settings
- **Creating a GitHub Repository with Terraform**
  - ○ Automating repository creation with Terraform
  - ○ Managing repository settings and permissions
- **Terraform Destroy**
  - ○ Safely destroying infrastructure
  - ○ Managing resource lifecycle with `terraform destroy`
- **Understanding Terraform State Files**
  - ○ The role of state files in Terraform
  - ○ Managing and securing state files
- **Desired and Current States**
  - ○ Understanding the concept of desired and current states
  - ○ How Terraform reconciles these states

Module 3: Read, Generate, Modify Configurations

- **Cross Resource Attributes**
  - ○ Referencing attributes between resources
  - ○ Using resource outputs as inputs to other resources
- **Output Values**
  - ○ Defining and using output values
  - ○ Exporting data from your configurations
- **Terraform Variables**
  - ○ Defining and using variables
  - ○ Variable types and validation

Module 4: Modules and Remote State Management

- **Understanding DRY Principle**
  - ○ Importance of DRY (Don't Repeat Yourself) in Terraform
  - ○ Structuring configurations to avoid redundancy
- **Implementing EC2 Module with Terraform**
  - ○ Creating reusable EC2 module
  - ○ Using modules in your configurations
- **Variables and Terraform Modules**
  - ○ Passing variables to modules
  - ○ Module outputs and inter-module communication
- **Implementing Remote Backend with Terraform S3**
  - ○ Setting up remote state storage in S3
  - ○ Configuring Terraform to use remote backends

Ansible Course Outline

Module 1: Introduction

- **How Ansible Works**
    - Introduction to Ansible and its architecture
    - Understanding the control node and managed nodes
    - Push-based configuration management
- **Setting Up Ansible**
    - Installing Ansible on various operating systems
    - Configuring Ansible control node
    - Verifying the installation

Module 2: YAML and INI Files

- **YAML 101**
    - Basics of YAML syntax
    - Writing and reading YAML files
    - Common pitfalls in YAML
- **YAML Challenge**
    - Practical exercises to reinforce YAML skills
- **Inventory Files 101**
    - Introduction to Ansible inventory
    - Structure and format of inventory files
    - Using inventory files to manage nodes
- **INI Challenge**
    - Writing inventory files in INI format
    - Practical exercises on creating and managing INI files
- **Writing AWS Inventory Files**
    - Inventory with AWS
    - Configuring Ansible to use AWS inventory
    - Practical example of AWS inventory setup

Module 3: Playbooks

- **Understanding the Documentation**
    - Navigating the Ansible documentation
    - Finding module-specific documentation
    - Understanding Ansible's documentation structure
- **Creating Your First Playbook**
    - Structure of a playbook
    - Writing a simple playbook
    - Playbook syntax and best practices
- **Running Playbooks**
    - Executing playbooks with `ansible-playbook`
    - Common options and flags
    - Debugging playbook runs

Module 4: Services, Handlers & Shell

- **Using the Service Module**
  - Managing services with Ansible
  - Starting, stopping, and restarting services
  - Example: Managing a web server service
- **Understanding Handlers**
  - Concept of handlers in Ansible
  - When to use handlers
  - Creating and triggering handlers
- **Creating Handlers**
  - Writing handlers in playbooks
  - Using handlers to manage service state
- **Using Shell & Debug Modules**
  - Running shell commands with Ansible
  - Using the `shell` and `command` modules
  - Debugging playbooks with the `debug` module

Docker Course

Module 1: Introduction to Containers

- **Server vs Virtual Machines vs Containers**
  - Comparison of different virtualization technologies
  - Advantages of containers over VMs
- **Problems Docker Solves**
  - Challenges in software deployment and scalability
  - How Docker addresses these challenges
- **Installing and Configuring Docker**
  - Step-by-step installation on various operating systems
  - Basic configuration settings

Module 2: Docker Basics

- **Docker CLI**
  - Overview of Docker command-line interface
  - Commonly used Docker commands

- **Images vs Docker Containers**
  - Understanding the difference between images and containers
  - Managing images and containers with Docker CLI
- **Attached and Detached Mode**
  - Running containers in attached and detached mode
  - Use cases for each mode

Module 3: Image Creation

- **Working with Docker Images**
  - Pulling images from Docker Hub
  - Listing and inspecting images with Docker CLI
- **Overview of Dockerfile**
  - Introduction to Dockerfile syntax and directives
  - Building custom images using Dockerfile
- **Writing Custom Dockerfiles**
  - Writing Dockerfiles for custom applications
  - Best practices for Dockerfile design
- **Building Custom Images from Dockerfiles**
  - Using `docker build` command to build images
  - Building images with build-time arguments and environment variables
- **Managing Images with CLI**
  - Managing image tags and versions
  - Removing unused images

Module 4: Docker Hub

- **Creating Docker Hub Account**
  - Setting up a Docker Hub account
  - Overview of Docker Hub features
- **Tagging Docker Images**
  - Tagging images with version numbers and labels
  - Best practices for image tagging
- **Docker Commit**
  - Committing changes to containers as new images
  - Creating images from running containers
- **Pushing Images to Central Repository**
  - Pushing images to Docker Hub
  - Configuring authentication for Docker Hub
- **AWS Elastic Container Registry (ECR)**
  - Overview of AWS ECR
  - Pushing images to AWS ECR

Module 5: Docker Networking

- **Overview of Docker Networking**
  - Understanding Docker networking concepts
  - Different network modes in Docker
- **Port Binding**
  - Exposing container ports to the host machine
  - Mapping container ports to specific host ports
- **Bridge and Host Networks**
  - Understanding bridge and host network modes
  - Use cases for each network mode

Kubernetes Course Outline

Module 1: Understanding Container Orchestration

- **Introduction to Kubernetes Architecture**
  - Overview of Kubernetes components (Master Node, Worker Nodes)
  - Understanding Kubernetes control plane (API Server, Scheduler, Controller Manager, etc.)
  - Kubernetes cluster architecture and its benefits
- **Installing and Configuring Kubectl**
  - Installing `kubectl` on various operating systems
  - Configuring `kubectl` to interact with a Kubernetes cluster
  - Basic `kubectl` commands and their usage

Module 2: Understanding Managed Kubernetes Architecture (Elastic Kubernetes Service)

- **Elastic Kubernetes Service (EKS) Overview**
  - Introduction to Amazon EKS and its components
  - Differences between self-managed and managed Kubernetes services
- **Eksctl and Kubeconfig File**
  - Installing and configuring `eksctl`
  - Using `eksctl` to create and manage EKS clusters
  - Configuring `kubeconfig` to access the EKS cluster
- **Creating Clusters Using AWS CLI**
  - Step-by-step guide to creating a Kubernetes cluster using AWS CLI
  - Managing cluster nodes and configurations
- **Understanding Eksctl Tool**
  - Detailed overview of `eksctl` commands and functionalities
  - Best practices for managing EKS clusters with `eksctl`

Module 3: Understanding Pods

- **Managing Pods Using CLI and YAML Formats**
  - Introduction to Pods and their lifecycle
  - Creating, managing, and deleting Pods using `kubectl` commands
  - Writing YAML files to define Pod specifications
  - Practical examples of Pod configurations

Module 4: Replicasets and Deployments

- **Replicasets**
  - Introduction to Replicasets and their purpose
  - Creating and managing Replicasets using `kubectl` and YAML
  - Scaling applications with Replicasets
- **Deployments**
  - Understanding Deployments and their role in application management
  - Creating and managing Deployments using `kubectl` and YAML
  - Using Deployments to ensure high availability and scalability
- **Rolling Updates and Rollbacks in Deployments**
  - Implementing rolling updates to deploy new versions of applications
  - Monitoring and managing rolling updates
  - Performing rollbacks to previous versions in case of issues