

Topic & Skill	Content Coverage
DevOps Linux Essentials	<ul style="list-style-type: none">o OS Introductiono Linux File structureo Basic Operational Commands<ul style="list-style-type: none">1. Files and directories2. documentation - man pages3. VI/nano Editor4. Shell Scripting - Variables Loops and conditions5. Controlling Processes6. Utility commands like find, grep, sort, head, tail, tn, ftp, tar.7. Conditions and Loops8. Package management - Ubuntu, RHEL & Amazon Linux9. linux based OS - RHEL, Ubuntu, Debian etc <p><i>#All essential linux commands required for the 30 day training program</i></p>
Introduction to DevOps	<ul style="list-style-type: none">o Basic Programming + Compiling + Packaging + Releaseo Define DevOpso What is DevOpso SDLC models, Lean, ITIL, Agileo Why DevOps?o DevOps Goals & Benefitso DevOps CI/CD Processo DevOps Toolso Source Code Managemento Purpose of Build & Release processo Configuration managemento Monitoring – Significance and how too What is DevSecOpso What is SREo Difference between SRE & DevOpso SRE - Key terminologies

GIT : Version Control
Maven Build Tool

- o Introduction
- o What is Git
- o About Version Control System and Types
- o Difference between CVCS and DVCS
- o GIT Basics
- o GIT Command Line
- o Branching & Merging Strategies
- o Different type of GIT based tools

Git Hands On:

- o Creating repository
- o Cloning, check-in and committing
- o Fetch pull and remote
- o Branching
- o Creating the Branches, switching the branches, merging the branches.

Introduction to Maven

POM file structure

Dependency management

Build jar/war files using Maven build

Jenkins CI/CD Orchestration

- o Understanding Continuous Integration Vs Continuous Delivery Vs Continuous Deployment
- o Jenkins installation - Windows, Linux, CCloud Platforms
- o Introduction about Jenkins
- o Jenkins Architecture
- o Build Cycle - Java
- o Jenkins – Plugins
- o Pipeline as Code – Declarative(default) Vs Scripted

Jenkins CI/CD Orchestration

Jenkins Hands On:

- o Jenkins hands on
- o master slave configuration
- o Creating Jobs - Declarative pipeline
- o Parameterized Jobs
- o Running the Jobs
- o Setting up the global environments for Jobs
- o Adding and updating Plugins
- o Disabling and deleting jobs

Introduction to AWS and DevOps Principles	<ul style="list-style-type: none">o Introduction to AWS Services - Overview of Key AWS services used in DevOps, including compute, storage, databases, networking and more.o DevOps Culture and Principles - Understanding the cultural shift, Collaboration, automation, measurement and sharing (CAMS) principles of DevOps.o Version Control and Collaboration - Using Git for version control and collaboration on code <p>Lab exercises to cover basic usecases of EC2, App Hosting, S3, IAM, SNS, SQS, Route 53, VPC, Cloud Formation, CloudFront, AMI Snapshots, Static IP, Elastic Block storage etc.</p>
AWS - Continuous Integration and Continuous Deployment	<ul style="list-style-type: none">o Building CI/CD Pipelines - Setting up end to end CI/CD pipelines using AWS CodePipeline, AWS CodeBuild and AWS CodeDeploy.o Automated Testing - Integrating testing frameworks into the CI/CD pipeline for ensuring code quality - Sonar and JUNITo Blue-Green Deployments - Implementing Blue Green deployment strategies for minimising downtime during releases. <p>Lab exercises to cover basic use cases for AWS CodePipeline, AWS CodeBuild and AWS CodeDeploy.</p>
AWS - Scalability, Security and Monitoring	<ul style="list-style-type: none">o Scalable strategies - Designing scalable architectures using AWS autoscaling, Elastic Load Balancing and serverless services.o Security Best Practices - Managing security through AWS IAM, security groups, encryption.o Monitoring and Observability - Implementing monitoring using Amazon CloudWatch<ul style="list-style-type: none">Optional - other observability tools like prometheus and grafana. <p>Lab exercises to cover basic usecases on ELB, ASG, IAM, CloudWatch</p>
AWS - Infrastructure as Code and Automation	<ul style="list-style-type: none">o Infrastructure as Code (IaC) Fundamentals - In-depth look at IaC concepts using AWS CloudFormationo IaC Best Practices - Designing reusable and modular infrastructure components using templateso Automation with AWS Services - Implementing automated workflows using AWS Lambda, Step functions and other automation services. <p>Lab exercises to cover basic usecases on CloudFormation, AWS Lambda, Step Functions</p>

<p>AWS -Advanced topics and Best Practices</p>	<ul style="list-style-type: none"> o Advanced containerization - In-depth exploration of Docker, Kubernetes, Amazon ECS and Amazon EKS. (Moved to week three) o Serverless Architecture - Detailed look at serverless concepts using AWS Lambda, API Gateway and event-driven architectures - SNS SQS o Cost Optimization Strategies - Understanding Cost allocation, budgeting and Optimizing resource usage using AWS Services. o DevOps Tools and Ecosystem - Exploring integration of DevOps Tools - Jenkins, Git, maven in AWS. <p>Lab exercises to cover basic usecases on Docker, Kubernetes, ECS, EKS, AWS Lambda, API Gateway and event driven architectures.</p>
<p>Docker– Containers</p>	<ul style="list-style-type: none"> o Introduction o What is a Docker & Containerisation o Understanding the Docker components o Platforms for Docker o installation o Creating containers o Docker compose <p>hands-on</p> <ul style="list-style-type: none"> o Create different environments with Docker o Build Containerized pipeline using Docker for Sample Web application in Jenkins
<p>Kubernetes - Container Orchestration Platform Fundamentals</p>	<ul style="list-style-type: none"> o Introduction to Kubernetes o Container Orchestration concepts o Kubernetes architecture o Setting up kubernetes cluster o Kubernetes Objects o Deploying Applications o Scaling and Autoscaling
<p>Kubernetes - Container Orchestration Platform Best Practices and Operations</p>	<ul style="list-style-type: none"> o Persistent Storage o ConfigMaps and secrets o Ingress Controllers o Network Policies o Deployments o Statefulsets o Custom resource Definition
<p>Kubernetes - Container Orchestration Platform Best Practices and Operations</p>	<ul style="list-style-type: none"> o Kubernetes Best Practices o Monitoring and Logging o Kubernetes Security o Cluster Backup and Disaster Recovery - Conceptual Knowledge o CI/CD Integration o Kubernetes Troubleshooting o Statefulsets o Helm o Amazon ECS and Amazon EKS. (moved from week 2)

Kubernetes - Container Orchestration Platform Advanced Kubernetes concepts	<ul style="list-style-type: none">o Kubernetes Advance Topicso Deployements using ArgoCD
Ansible - Config Management	<ul style="list-style-type: none">o What is Ansibleo Why Ansibleo Basic Ansible Terminologyo Infra as Codeo Introduction to Playbookso Playbook Structureo Introduction to Moduleso Variables and Factso Ansible Configuration Hierarchyo Ansible in the Cloud <p>Hands-on:</p> <ul style="list-style-type: none">o Creating an Ansible Home Baseo Setting up Test Environmento Creating Playbookso Jenkins integrationo Trigger playbooks form Jenkins
Project Milestone 2 - Create an end to end CI/CD pipeline in AWS platform using Jenkins as the orchestration tool, Github as the SCM, Maven as the Build tool, Deploy in a docker instance and create a Docker image, Store the docker image in ECR, Achieve Kubernetes deployment using the ECR image. Build a sample java web app using maven.	
Python Fundamentals	<ul style="list-style-type: none">o Introduction to Pythono Python Basics - Variables, Basic IO, Operatorso Control Structureso Functionso List and Tupleso Dictionarieso File Handlingo Exception Handlingo Modules and Packageso Object Oriented Programmingo Introduction to Libraries (Optional)o Practice Examples - Scripting exercises and examples throughout the day to reinforce concepts.
Azure Fundamentals	<ul style="list-style-type: none">o Introduction to Microsoft Azureo Azure Services and Solutionso Azure Portal and Azure Resource Managero Azure Virtual Machineso Azure Storageo VPC + Azure Networking (Optional)o Azure Identity and access Managemento Azure App Serviceso Monitoring and Management (Optional)o Security and compliance (Optional)o Cost Management and Optimisation (Optional)

Azure DevOps

- o Introduction to Azure DevOps
- o Version control with Azure Repos
- o Azure Pipelines CI/CD Fundamentals
- o Building CI Pipelines
- o Deploying applications with CD Pipelines

Lab:

- o Building CI Pipelines
- o Deploying using CD Pipelines

Terraform Basics

Understanding Infrastructure as Code (IaC)

- o Introduction to Infrastructure as Code (IaC)
- o Why Terraform?
- o Terraform vs. other IaC tools
- o Key concepts: Declarative vs. imperative, Desired State Configuration (DSC), Idempotency

Getting Started with Terraform

- o Installing Terraform
- o Initializing a Terraform project
- o Terraform configuration language (HCL)
- o Terraform workflow: plan, apply, destroy
- o Managing state files

Terraform Fundamentals

- o Providers and resources
- o Variables and data types
- o Outputs
- o Modules: organization and reusability
- o Terraform state management: remote state, backends

HANDS On-Lab

- o Setting up a simple infrastructure with Terraform
- o Deploying and managing resources on a cloud provider (e.g., AWS, Azure, Google Cloud)

Terraform Advanced	<p>Advanced Terraform Configuration</p> <ul style="list-style-type: none">o Dependency managemento Workspaceso Remote execution with Terraform Cloud/Enterpriseo Using provisioners and local-execo Understanding count, for_each, and dynamic blocks <p>Terraform in Production</p> <ul style="list-style-type: none">o Best practices for Terraform code organizationo Security considerationso Continuous Integration/Continuous Deployment (CI/CD) pipelines with Terraformo Managing Terraform with version control (e.g., Git) <p>Terraform Modules and Collaboration</p> <ul style="list-style-type: none">o Building and publishing moduleso Using community moduleso Collaborative Terraform development with Git and version controlo Terraform Enterprise features for team collaboration <p>Troubleshooting and Optimization</p> <ul style="list-style-type: none">o Debugging Terraform configurationso Terraform graph and plan inspectiono Performance optimization techniqueso Handling Terraform errors and common pitfalls
GitHub actions	<ul style="list-style-type: none">o Introduction to Github actionso Getting started with Github actionso Basic workflowso Building and testing with Github actionso Continuous Deployment with GitHub Actions
Introduction to Site Reliability Engineering	<ul style="list-style-type: none">o Introduction to Site Reliability Engineeringo SRE VS DevOpso SLIs, SLOs and Error Budgetso Monitoring and alertingo Incident management and Post Mortemso Load Balancing and traffic Managemento Capacity Planning and Auto Scalingo Disaster recovery and Auto-Scaling
SRE Advanced	<ul style="list-style-type: none">o Chaos Engineeringo Automation and Infrastructure as Codeo Distributed Monitoring and Observabilityo Performance Optimization and Efficiencyo Managing Microservices and Containerso Continuous improvement and SRE culture

Splunk Essentials	<ul style="list-style-type: none">o Introduction to Splunko Benefits of Splunk in DevOpso Splunk Architectureo Search Processing Language (SPL)o Splunk Search Interfaceo Data Visualizationo Hands-on Labs
Lab Use Cases / POC	<ul style="list-style-type: none">o Revision and practice of concepts through hands-on exercises.