



Azure DevOps Curriculum

Module-1

Introduction to DevOps

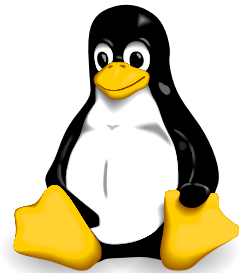
- What is Software Application Development?
- SDLC (Software Development Life Cycle)
- What is DevOps? Why DevOps?
- Who can learn DevOps?
- What are the prerequisites to learn DevOps?
- DevOps Job Market Trends
- Why are DevOps skills in high demand across industries?
- History and Evaluation of DevOps
- Traditional IT vs Agile vs DevOps
- DevOps Life Cycle
- Benefits and Business values of DevOps

➤ Introduction to Cloud and Azure

- What is a Data centre? What is IT infrastructure?
- Physical servers and Virtual servers
- Difference between Physical and virtual servers
- Virtualisation concept
- What is Cloud computing?
- Benefits of Cloud
- Cloud service providers in IT (Azure, AWS, GCP, Oracle Cloud, IBM Cloud, etc)
- Cloud Regions and Availability zones
- Why Azure Cloud?
- Azure free tier account set up
- Azure virtual Machine (VM) creation for practice

➤ Linux Basics

- Introduction to OS
- Types of OS
- Linux OS architecture
- Linux commands
- User and permissions Management
- Package Management
- Shell scripting basics



Module-2

Azure

➤ Azure Administration Services

- Microsoft Entra ID (Azure Active Directory)
 - Domains
 - Tenants
 - Subscriptions
 - Licenses
 - Users
 - Groups
 - Azure resource access management workflow
 - Resource groups
 - Resource manager
 - Resource providers
 - AD Roles



➤ Virtual Machines

- Azure Virtual Machines
- Images (OS) and sizes (resources)
- Password and key-based authentication
- Disk types (OS disk and Data disk)
- Disaster recovery
- Monitoring
- Custom data and cloud-init (post actions)

➤ Azure Virtual Network

- Introduction to networking concepts
- IP address calculations
- Classes, CIDR, Subnets, IPv4, broadcast, gateway, Hub, Switches,
- Routers
- Virtual network
- Subnet
- Public IP address
- NAT gateways
- Security (Firewall, DDoS, Bastion Host)
- DNS servers
- Address spaces
- Security
- Network manager
- Peering
- Service endpoints
- Private endpoints
- Monitoring



➤ Load Balancing

- Introduction to traffic and load balancing
- OSI layers
- Load Balancer
- Application Gateway
- Traffic manager
- Front Door and CDN profiles

➤ Storage

- Introduction to ephemeral and persistent storage
- Different storage types
- Azure Disks
- Blob storage
- Azure Files
- Data protection (Storage recovery options)
- Encryption

Module-3

DevOps

- Introduction to DevOps
- DevOps stages

➤ Version Control System

- What is Version Control?
- Why Git is important in DevOps
- Centralised vs Distributed VCS
- Git workflow in real projects
- Basic Git terminology
- Git Installation & Repository Basics
- Installing Git (Windows/Linux)
- Git Bash & CLI overview
- Git configuration (username, email)
- Creating a Git repository
- Working directory, staging & repository
- Core Git Commands
- Understanding commit history
- Best practices for clean commits
- Branching, Merging & Conflicts
- What are branches?
- Creating & switching branches
- Merging branches
- Fast-forward vs merge commit
- Merge conflict resolution
- Remote Repositories & GitHub/Azure Intro
- What is a remote repository?



➤ Intro to Azure DevOps & Azure Repos

- Azure Repos Fundamentals
- Azure DevOps organisation & projects
- Creating an Azure Repos Git repository

- Repo permissions & access control
- Cloning Azure Repos
- Work with Azure Repos
- Branch Management in Azure Repos
- Branch creation & deletion
- Default branch settings
- Branch naming conventions
- Branch policies (intro)
- Apply branch rules
- Pull Requests & Code Review
- What is a Pull Request (PR)?
- Creating PRs in Azure Repos
- Code reviews & approvals
- Handling PR conflicts
- Linking work items to PRs
- PR-based workflow
- Azure Repos with CI/CD
- Azure Repos + Azure Pipelines integration
- Triggering CI builds on commits
- YAML pipeline basics
- Build validation policies
- Commit-triggered CI pipeline



➤ Azure Pipelines

- Introduction to Pipelines
- Pipeline YAML
- Azure Pipeline agents (Microsoft-hosted and Self-hosted agents)
- CI with Azure Repos
- CD with Kubernetes, Self-hosted agents, Serverless platforms, etc.
- RBAC (Pipeline access control)
- pool
- steps
- tasks
- script
- Triggers (Manual and Auto)
- Pipeline variables
- template

➤ Azure Boards

- Introduction to agile model and Scrum
- Introduction to boards
 - New
 - In-progress
 - Staging
 - Done
- Backlogs
- Work items
- Sprints
- Plans
- Queries



Module-4

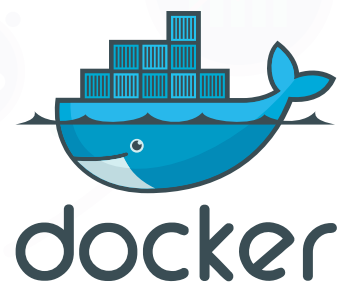
Containerization- Docker

➤ Introduction to Containers & Docker

- What is containerization?
- Why Docker in DevOps
- Virtual Machines vs Containers
- Docker architecture (Client, Engine, Registry)
- Docker in real DevOps projects

➤ Docker Installation & Setup

- Docker installation
- Docker Desktop overview
- Docker CLI basics
- Docker Hub account setup



➤ Docker Images & Containers

- Images vs Containers
- Image lifecycle
- Pulling images from Docker Hub
- Running & managing containers
- Container logs & inspection

➤ Core Docker Commands

➤ Dockerfile & Image Creation

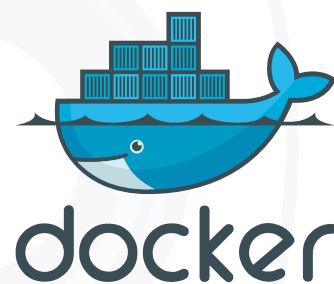
- What is a Dockerfile?
- Dockerfile instructions:
- Build, tag & version images
- Dockerfile best practices

➤ Docker Networking & Storage

- Docker networking basics
- Bridge network & port mapping
- Docker volumes & bind mounts
- Container data persistence

➤ Docker Compose

- What is Docker Compose?
- docker-compose.yml structure
- Multi-container applications
- Use cases in development & testing



➤ Docker Registry & Azure Integration

- Docker Hub (public & private repos)
- Azure Container Registry (ACR)
- Push & pull images
- Image security basics

➤ Docker with Azure DevOps CI/CD

- Docker in CI/CD pipelines
- Build Docker images using Azure Pipelines
- Push images to ACR / Docker Hub
- Azure DevOps Docker tasks
- Secrets & environment variables

Container Orchestration Tool - Kubernetes

➤ **Introduction to Kubernetes**

- Why Kubernetes?
- Kubernetes architecture
- Master & worker nodes
- Control plane components
- Kubernetes vs Docker Swarm (overview)

➤ **Kubernetes Core Objects**

- Pods
- ReplicaSets
- Deployments
- Services (ClusterIP, NodePort, LoadBalancer)
- Namespaces

➤ **Kubernetes Configuration**

- YAML basics for Kubernetes
- Creating & managing resources
- Labels & selectors
- Resource requests & limits

➤ **Kubernetes Networking & Storage**

- Cluster networking basics
- Services & ingress (overview)
- Volumes & persistent volumes (PV, PVC)
- ConfigMaps & Secrets

➤ **Kubernetes Operations**

- Scaling applications
- Rolling updates & rollbacks
- Health checks (liveness & readiness probes)
- Logs & monitoring basics



kubernetes

➤ Azure Kubernetes Service (AKS)

- What is AKS?
- AKS architecture overview
- Deploying applications to AKS
- AKS vs self-managed Kubernetes
- Helm concepts

➤ Kubernetes with Azure DevOps CI/CD

- Build Docker image push to ACR
- Deploy to AKS using Azure Pipelines
- Kubernetes service connections
- YAML pipelines for AKS deployment

➤ Monitoring

- Prometheus
- Grafana
- EFK
- Kubernetes Security
 - Kube-bench (aquasecurity)
 - KubeLinter



kubernetes

➤ Azure Monitor

- Differences between metrics and logs
- Activity logs
- Alerts
- Metrics
- Logs
- Autoscale
 - Horizontal scaling (scale-in and scale-out VMs)
 - Vertical scaling (scale-in and scale-out size)
- Insights
- Service Health
- Managed services
 - Prometheus
 - Grafana

- Azure Content Delivery Network
- Azure Multifactor Authentication
- Microsoft Defender for Cloud
 - Security alerts
 - Inventory
 - Cloud Security Explorer
 - Workbooks
 - Cloud security
- Web Application Firewall
 - Introduction vulnerability
 - Basic understanding of web application exploits
 - WAF configuration
 - WAF policy
 - WAF for Azure Application Gateway
 - WAF for Front door
 - WAF for CDN
- Introduction to DevSecOps
- DevSecOps working principles

Module-5

Terraform- an Infrastructure management tool

- **What is IT infrastructure?**
- **Introduction to Infrastructure as Code (IaC) and Terraform**



Terraform

- What is Infrastructure as Code?
- Why Terraform? Features and Use Cases
- Terraform vs Other IaC Tools (CloudFormation, Pulumi, Ansible)
- Terraform Architecture and Workflow
- Supported Providers Overview (AWS, Azure, GCP, Kubernetes)

➤ Getting Started with Terraform

- Installing Terraform (Windows/Linux/macOS)
- Terraform CLI Basics
- First Terraform Configuration – Provisioning EC2 on AWS
- Terraform Workflow: Init --> Plan --> Apply --> Destroy
- Hands-on: Your First Terraform Project

➤ Terraform Configuration Language (HCL)

- Overview of HCL Syntax
- Resources, Providers, and Variables
- Data Sources and Outputs
- Local Values and Expressions
- Input Validation and Descriptions

➤ Variables and Outputs

- Defining Input Variables
- Variable Types and Validation Rules
- Using Environment Variables and terraform.tfvars
- Output Values and Dependencies
- Sensitive Data Handling

➤ Terraform State Management

- What is Terraform State?
- Local vs Remote State Files
- State Locking and State File Structure
- Terraform Workspaces (Managing Environments)

➤ Providers and Resources

- What is a Provider?
- Provider Block Configuration
- Using Multiple Providers (e.g., AWS + Kubernetes)
- Resources – Basics and Arguments
- Data Sources – Referencing Existing Infrastructure



Terraform

➤ Modules in Terraform

- What are Modules?
- Creating and Reusing Modules
- Organising Code with Root and Child Modules
- Calling Modules with Parameters
- Using Public Modules from the Terraform Registry

➤ Terraform with AWS (or Azure/GCP)

- Setting up Terraform with AWS Credentials
- Creating VPC, Subnets, Security Groups, and EC2 Instances
- Using IAM Roles and Policies
- Deploying S3 Buckets, RDS, and Load Balancers
- Hands-on: Full Infrastructure Deployment on AWS

➤ Remote State and Backend Configuration

- What is a Backend?
- Configuring Remote Backends (S3)
- State Locking with DynamoDB (AWS)
- Secure State Storage Best Practices
- Hands-on: Set up S3 Backend with Locking

➤ Provisioners and Templating

- Using local-exec and remote-exec Provisioners
- Connection Block for Remote Execution
- File Provisioner to Upload Files
- Template Files and Interpolation
- When (and When NOT) to Use Provisioners

➤ Terraform Functions and Expressions

- Built-in Functions (lookup, join, length, etc.)
- Conditional Expressions and Loops (for, count, for_each)
- Dynamic Blocks and Complex Structures
- Hands-on: Create a Scalable, Parameterised Deployment



Terraform

➤ Terraform Cloud and Enterprise

- Introduction to Terraform Cloud
- Remote Execution and Workspaces in Terraform Cloud
- Variable and Secret Management
- Terraform Enterprise Features (RBAC, Policy as Code)
- Hands-on: Deploy via Terraform Cloud

Module-6

Ansible

➤ Introduction to Configuration Management

- What is Configuration Management?
- Introduction to Ansible
- Ansible vs Other Tools (Puppet, Chef, Salt)
- Use Cases and Benefits of Ansible
- Architecture Overview – Control Node and Managed Nodes

➤ Setting Up Ansible

- Installing Ansible (Linux, Mac, Windows via WSL)
- Understanding Ansible Inventory (Static and Dynamic)
- Setting up SSH Key Authentication
- Running Your First Ad-Hoc Command
- Basic Troubleshooting and Common Errors

➤ Ansible Core Concepts

- Modules and Ad-Hoc Commands
- Playbooks – Structure and YAML Syntax
- Tasks, Handlers, and Tags
- Variables – Types and Precedence
- Facts and setup Module

➤ Writing and Running Playbooks

- Anatomy of a Playbook
- Executing Playbooks with ansible-playbook
- Conditional Execution (when)
- Looping in Playbooks (with_items, loop)
- Using Tags for Task Selection



➤ Variables and Templates

- Defining Variables (Inventory, Playbook, Host Vars, Group Vars)
- Registering Variables from Commands
- Using Jinja2 Templating
- Template Module and .j2 Files
- Variable Precedence and Best Practices

➤ Roles and Reusability

- Introduction to Roles in Ansible
- Creating and Using Roles
- Role Directory Structure
- Importing Roles from Ansible Galaxy
- Best Practices for Role-Based Projects

➤ Ansible Inventory Management

- Static Inventory – INI and YAML
- Dynamic Inventory (AWS, Azure, GCP, etc.)
- Inventory Grouping and Variables
- Host and Group Variable Files
- Inventory Plugins and Scripts



ANSIBLE

➤ Error Handling and Debugging

- Using `ignore_errors`, `failed_when`, and `block`
- Error Messaging with `debug` and `msg`
- Rescue Blocks and Error Recovery
- Validating Playbooks (`ansible-lint`, `--syntax-check`)
- Debugging Playbooks and Verbose Modes

➤ Ansible Vault – Managing Secrets

- What is Ansible Vault?
- Encrypting and Decrypting Files
- Using Vault in Playbooks
- Editing Encrypted Files Securely
- Vault IDs and Best Practices

➤ **Advanced Playbook Techniques**

- Import vs Include
- Handlers and Notifications
- Delegation and Local Actions
- Asynchronous Tasks and Polling
- Working with Collections

Module - 7

➤ **Real-life Projects**

➤ **Interview-oriented and advanced topics**