

# Azure Training

## Day 1

### **Azure Services:**

Objective would be to get an idea of the Core Azure Services. It should also cover in detail about the Setup, Functionality and Usage and how each component integrated with each other.

- Azure compute services
  - Azure Virtual Machines
  - Azure App Service
  - Azure Container Instances
  - Azure Kubernetes Service
  - Azure Functions
  - Azure Virtual Desktop
  - Azure Virtual Machine Scale Sets
- Azure architectural components
  - Overview of Azure subscriptions, management groups, and resources
  - Azure regions, availability zones, and region
  - Azure resources and Azure Resource Manager
  - Azure subscriptions and management groups
  - Resource Groups
  - Service Principals
- Azure Networking and Security
  - VPN Gateway
  - Virtual Network peering
  - ExpressRoute
  - Network Security Group
  - Subnets
  - Vnets
  - Public/Private Endpoints
  - Azure Firewall
  - Azure DDoS protection
  - Azure Key Vault
  - Microsoft Sentinel
  - Azure Key Vault
  - Microsoft Defender for Cloud,

- Policy compliance
  - Security alerts
  - Secure score
- Application Gateway
- Azure Storage
  - Azure Files
  - Azure Blob Storage
  - Storage Accounts
- Azure DNS
- Containers
  - Azure Container Registry
- Azure Management tools
  - Azure CLI
  - Azure Portal,
  - Azure PowerShell
  - Cloud Shell
  - Azure Advisor
  - Azure Resource Manager (ARM) templates
  - Azure Monitor
  - Azure Service Health
  - Azure Log Analytics

## Day 2

- Azure identity, governance, privacy, and compliance
  - Authentication and Authorization
  - Azure Active Directory and how it can be used
  - Conditional Access, Multi-Factor Authentication (MFA), and Single Sign-On (SSO)
  - Role-Based Access Control (RBAC) for Azure resources and how they are defined.
  - Resource locks
- Azure cost management
  - Identify factors that can affect costs (resource types, services, locations, ingress, and egress traffic)
  - Functionality and usage of Azure Cost Management
  - Pricing calculator and the Total Cost of Ownership (TCO) calculator

### **Infrastructure as Code:**

#### **Terraform**

This module should discuss about the Advance concepts of Terraform and the **lab sessions should help us in deploying resources to Azure via Terraform**. The integration with a CI/CD pipeline should be covered

- Why Terraform
- How Terraform works -> Providers, Plugins, Workspaces
- How Terraform compares to other infrastructure as code tools
- Terraform Syntax
- Manage Terraform state
  - Terraform state
  - Shared storage for state files
  - Locking state files
  - Isolating state files
  - File Layout -> How to have a Folder structure for deploying on Multiple Environments
- Reading Data Objects
- Reusable infrastructure with Terraform modules
- loops, if-statements, deployments

## Day 3

#### **Ansible**

This module should Cover how Configuration Management can be used for Provisioning resources in the Azure Cloud.

- How to Write Playbooks
- Ansible Modules
- Jinja Templates
- Inventory and Host Management
- Dynamic Inventory

## **Packer**

Module should cover the basics of Packer and how images are Built to be used in the VM,VMSS and how they can be stored in the Azure Image Registry

- Template
- Builders
- Provisioners

## **Container Orchestration:**

### **Kubernetes:**

The module should cover Kubernetes in an Intermediate level with the Lab Topics covering deployment of applications in the Kubernetes.

- Kubernetes Architecture
- Kubectl CLI
- Kubernetes Resources and objects.
- Jobs and CronJobs
- Multi-container Pod design
- Persistent and ephemeral volumes
- Deployments
- API deprecations
- Probes and health checks
- Debugging in Kubernetes
- Understand authentication, authorization, and admission control
- Understanding and defining resource requirements, limits, and quotas
- ConfigMaps
- Secrets
- ServiceAccounts
- SecurityContexts
- Ingress rules to expose applications
- Network Policies

# Day 4

## **Using Helm Charts in Kubernetes:**

The module should help participants develop helm charts and deploy them in a K8S environment .The integration with a CI/CD pipeline should be covered .

- Helm Architecture – HelmV3
- Why Should You Use Helm
- Helm Charts, Releases, and Repositories
- Developing Helm Charts
- Helm Chart Tree Structure
  - chart.yaml
  - values.yaml
  - templates (dir)
  - charts
- Storing Helm Charts
- Deploying Helm Charts
- Helm CLI commands

## **CI/CD with Azure Pipelines and Azure Devops**

The module should help participants develop an completed CI/CD solution using azure devops, azure pipelines and Azure Services.

- Version controlling with Git in Azure Repos
  - Managing GIT repositories,Hooks, Pull Requests, Git branches and workflows
- Explore Azure Pipelines and Features
  - Pipeline Basics
  - Build History
  - Triggers
  - Task&Templates
  - Jobs&Stages
  - Libraries
  - Variables
  - Securing files -> Secrets
  - Approvals and checks and gates
  - Configure Security and Retention policies
- Executing parallel Builds using Azure Devops
- Integration to Azure Services with Service Connections from Azure Devops using Azure Pipelines
- Auto scaling with Azure Agent Pools.
- Implement a continuous CI/CD process using Azure Agent Pools
- Deployment patterns using Azure Pipelines.



# **Day 5**

- **Azure Devops**

- User Hierarchy/Management in Azure Devops
- Managing permission to resources using PAT tokens
- Agent Pool Configuration (Self Hosted, Virtual Machine Scale Set)
- Project Hierarchy and Structure in Azure Devops
- Hooks
- Test Plans
- Artifacts

## **CI/CD with GitHub Actions and GitHub**

The module should help participants develop an completed CI/CD solution using Github, Github Actions and Azure Services.

- Version controlling with Git in GitHub
  - Managing GIT repositories, Hooks, Pull Requests, Git branches and workflows
- Understanding GitHub Actions
  - Components of GitHub Actions
  - Workflows
  - Managing Workflows
  - Events
  - Jobs
  - Actions
  - Packaging with GitHub Actions
  - Runners
  - Using Secrets in Workflows
  - Automatic token authentication
  - Deployments to Services in Azure
  - Hardening Deployments
- Managing Workflow Runs
- Project/User management in GitHub
- Project Hierarchy and Structure IN GitHub
- Continuous deployment using GitHub Actions
- GitHub-hosted runners
- Self-hosted runners
- Autoscaling with self-hosted runners

## **Implement security and validate code bases for compliance**

- Introduction to Secure DevOps
- Implement open-source software

- Software Composition Analysis
- Static analyzers
- Lab: Implement White Source
- Lab : Managing technical debt with SonarQube and Azure DevOps
- Lab : Implement security and compliance in Azure DevOps Pipelines