Azure DevOps Syllabus

DevOps Introduction

- Traditional Software Development Life Cycle
- Waterfall Model
- About Agile Methodology.
- What is DevOps?
- DevOps Practices?
- The Challenge
- Benefits of DevOps over Traditional IT
- DevOps Tools
- What is CI and CD
- DevOps as a profession – DevOps Engineer

Azure DevOps

- What is Azure DevOps
- Version History
- Azure DevOps Features
- Azure DevOps Tools and Project Life Cycle
- Create DevOps Account
- Create Organization
- Create Project and Get Started
- Create Users and invite teams members

Azure Boards

- Introduction
- Working with Work Items
- Epic, Feature, User Story, Task, Bug and Test Cases
- Linking Items
- Collaborate with Team members
- Follow a Work Item
- Dashboards

- Capacity Planning with Sprints
- List work items using Queries

Azure Repos, GitHub and GIT

- Version control using Git
- What is Git, Azure Repos and GitHub
- Install Git Locally
- Getting Started with Git Commands
- Updating to DevOps Repository
- Working with Branches
- Merging Branches
- Creating and Committing a Pull Request
- Add a rule to Require a Review
- Squash Merging during Pull Request.
- Working with Merge Conflicts
- Cherry-Picking and Rebase
- Undo Changes using Reset and Revert
- Ignoring files using gitignore
- Managing Git Branches in Azure Repos
- Branch Policies and Branch Permissions
- Branches in Folders
- Working the GitHub Repositories
- Branching Workflow Types
    - Feature Branching
    - Gitflow Branching
    - Forking Workflow
- Summary of Git Commands

Azure Repos TFVC

- About TFVC
- Using TFVC in VS.NET

- Moving from TFVC to Git

Continuous Integration using Azure Build Pipelines

- About Azure DevOps Pipeline
- Understanding the Build Process
- Create a Pipeline using Classic Editor
- Enable Continuous Triggers for Build Pipeline
- Add a status badge to Repository
- Working with Task Groups
- Validate Pull Request based on Build Pipeline result
- Add a Widget to Dashboard

Continuous Integration using YAML Pipelines

- Understanding YAML file format
- Building Azure DevOps Pipeline using YAML
- Publishing results to Artifacts
- Triggering Continuous Integration in YAML
- Filtering Tasks based on branch being built
- Using Templates to Build Multiple Configurations
- Build on Multi-Platform pipeline

Integrating Quality Tests in Azure Pipeline

- Overview of Testing
- Add Unit Tests to your Application
- Integrating Unit Test with CI Pipeline
- Add the Test Widget to Dashboard
- Perform Code Coverage Testing using Cobertura

Scan Code for Vulnerabilities and License Ratings in CI Pipeline

- Sources and Impacts of Technical Dept
- Managing Technical Dept with DevOps and Sonar Cloud

- Scan open-source components using WhiteSource Bolt

Continuous Deployment using Azure Pipelines

- What is Continuous Delivery
- Connecting to Azure Subscription
- Deploying App to App Service using Designer
- Multi-State Pipeline
- Approvals and Gates
- Working with Task Groups
- Deploying App to Virtual Machine
- Deploying App to App Service using YAML
- Add the deployment State to the pipeline
- Deploy Apps to Specific Environment
- Deploy Azure Functions

Deep Dive into CI ad CD Pipeline

- Introduction
- Retention Policies
- About Build Agents
- About Agent Pools
- Create Self Hosted Windows Agent
- About Libraries
- Variables
- Secure Files
- Pipeline Conditions
- Pipeline Demands
- Integrating Pipeline with Microsoft Teams

UI Test using Selenium

- Overview about Functional Tests
- Kind of Functional Tests

- UI Test with Selenium on Local System
- UI Tests in Build and Release Pipeline
- Capture Video

Azure Key Vault

- Introduction to Key Vault
- Secrets vs Keys
- How it Works
- Creating a Key Vault Service and Add Keys and Secrets
- System Assigned and User Assigned Managed Identity
- Reference Key Vault in ARM Template
- Reference Secrets with Dynamic ID

Working with SQL Database

- Creating a Database Project
- Database Deployment using DACPAC
- Deploying Database using SQL Scripts in Pipeline
- Using Multiple Stages and Approvals

IaC using ARM Templates

- About Infrastructure as Code (IaC)
- About ARM Templates
- Sample to Create Storage Account using ARM Template
- Deploy Templates using PowerShell
- Deploy Templates using Azure Portal
- Deploy Templates using Azure Pipeline
- Incremental and Complete Deployment
- Creating VM using ARM Template
- Create linked ARM Templates
- Creating Resource Group and Resources at Subscription Level

IaC using Terraform Templates

- Overview of Terraform
- Terraform Files Structure
- Terraform Commands
- Run a Terraform plan from Azure Cloud Shell
- Provision Terraform Tasks in Azure Pipeline – Classic Editor
- Provision Terraform Task in Azure Pipeline – YAML File

IaC using Ansible

- Overview of Ansible
- Ansible Workflow
- Ansible Components
- Installing Ansible
- Playbook Structure
- Executing a Playbook

Azure Artifacts

- What are Artifacts
- Public and download Build Artifacts
- Publish and download Pipeline Artifacts
- Working with Feed and NuGet Packages
- Share Packages Publicly
- Public NuGet Package from Pipeline to NuGet Feeds
- Upstream Sources and View

Continuous Integration using Jenkins

- Jenkins Management
- Adding a slave node to Jenkins
- Building Delivery Pipeline
- Pipeline as a Code

Working with Containerization using Docker

- Understanding VM and Containers
- What is Docker and its Benefits
- Docker Architecture
- Steps to Create Docker Image
- Build and Publish Docker Image to Docker Hub using Azure Pipeline
- Build and Publish Docker Image to Azure Container Registry using Azure Pipeline
- Deploying to Web App
- Deploying a Docker Container in VM or Local Machine using Docker Compose

Working with Kubernetes

- Deploying Applications to Kubernetes Cluster
- What is Kubernetes
- Kubnetes Server and Client Components
- Creating an AKS Cluster
- Writing Deployment and Service YAML files
- Deploying the Application using Kubectl
- Building a CI and CD Pipeline for Deploying to Kubernetes Cluster.