Azure DevOps

Course Overview:

The Microsoft Azure DevOps certification training will help learners master the concepts of both DevOps and Azure and develop formidable skills in cloud architecture, Resource Manager & Virtual Network Connectivity. Azure DevOps is the platform from Microsoft that provides a complete end-to-end set of DevOps tools for developing and deploying software. With efficiency being a constant goal in the process of software development, Azure DevOps is a powerful tool for implementing and managing a solid DevOps toolchain. Azure DevOps offers several services from Azure Boards for planning and tracking, to Azure Pipelines for building, testing, and deploying, to Azure Test Plans for testing and shipping, and much more.

Course Outline:

1. Introduction to Microsoft Azure

- Introduction to cloud computing
- What is Microsoft Azure?
- Microsoft Azure Services
- Creating a Microsoft Azure Account
- Azure CLI, Azure PowerShell
- Microsoft Azure Architecture

2. Introduction to Azure Storage

- Azure Resources & Subscriptions
- Azure Resource Manager
- Managing Azure Resources
- Azure Tags
- Azure Storage Account & its types
- Azure Blob Storage
- Azure Files Storage
- Azure File Sync

Hands-on Exercise:

- Manage Resource Groups in Azure

- Move resources from one resource group to another
- Apply tags
- Create storage account
- Access storage account
- Create blob storage
- Upload in blob storage

3. Azure Virtual Machines

- Azure Virtual Machines
- Data Disks in Azure
- Azure VMs & Interfaces
- Custom Images of Azure VM
- Virtual Machine Scale Sets
- Virtual Machine Availability Sets

Hands-on Exercise:

- Creating and Configuring an Azure VM
- Deploying a custom image of Azure VM
- Virtual Machine Scale Sets.

4. Azure App and Container services

- App Service Web App for Containers
- App Service plan
- Networking for an App Service
- Deployment slots
- Container image
- Azure Kubernetes Service
- Azure Container Registry

Hands-on Exercise:

- Create an App Service Web App for Containers
- Create a container image
- Configure Azure Kubernetes Service
- Publish and automate image deployment to the Azure Container Registry

5. Azure Networking - I

- Azure Virtual Networks
- Azure Vnet Components
- IP Address – Public & Private IPs
- Azure Vnet Subnets
- Azure Network Interface Cards (NIC)
- Network Security Group (NSG)
- Azure DNS
- Private DNS

Hands-on Exercise:

- Vnet creation
- Create and configure vnet-vnet peering
- Verify virtual network connectivity
- Assign static IP to VM
- Create route tables
- Add routes
- Create NIC
- Attach NIC to VM
- Create DNS
- Add RecordSet
- Create NSG

6. Azure Networking - II

- Application Gateway
- Azure Traffic Manager
- Application Security Groups
- Azure Load Balancers
- Azure Firewall
- Azure Bastion
- Network Watcher

Hands-on Exercise:

- Create an internal load balancer
- Create a Public load balancer
- Application Gateway

- Implement Azure Traffic Manager
- Deploy and configure Azure Bastion Service

7. Microsoft Azure Active Directory

- Azure Active Directory (Azure AD)
- Windows AD Vs Azure AD
- Azure AD Users
- Azure AD Groups
- Azure AD Tenants
- Authentication Options
- Azure AD Connect
- Self Service Password Reset (SSPR)
- Multi-Factor Authentication (MFA)

Hands-on Exercise:

- Add or delete users using Azure Active Directory
- Add or delete tenants using Azure Active Directory
- Create a basic group and add members

8. Azure Monitoring

- Azure Monitor
- Azure Metrics
- Log Analytics
- Application Insights
- Backup reports
- Backing Up Azure Virtual Machines
- VM Backup Policies
- Restoring Azure Virtual machines,

9. Infrastructure Setup - AKS

Installation of DevOps Tools on cloud

- GIT/GITHUB
- Azure CLI
- Terraform
- AKS

10. Introduction to DevOps

- What is Software Development
- Software Development Life Cycle
- Traditional Models for SDLC
- Why DevOps?
- What is DevOps?
- DevOps Lifecycle
- DevOps Tools

11. Project Management

- What PM?
- What is the importance of PM in the Org?
- What is Agile/Scrum?
- What is a backlog?
- What are Sprint/Iterations?

12. Test Management

- What is Test Management?
- What is Test Plan?
- What is Test Suite?
- What are Test Cases?
- What is the Difference Between
- Manual and Automation Testing?
- What are Azure artifacts
- Connecting to azure pipelines
- What are Azure test plans
- Testing web apps?

13. Software Version Control

- What is Version Control?
- Types of Version Control System
- Introduction to SVN
- Introduction to Azure Repo
- Introduction to Git
- Git Lifecycle
- Common Git Commands

- Working with Branches in Git
- Merging Branches
- Resolving Merge Conflicts
- Git Workflow

14. Docker

- Introduction to Docker
- Understanding Docker Lifecycle
- Components of Docker Ecosystem
- Common Docker Operations
- Creating a DockerHub Account
- Committing changes in a Container
- Pushing a Container Image to DockerHub
- Creating Custom Docker Images using Dockerfile

Hands-on Exercise:

- Common Docker Operations
- Creating a DockerHub Account
- Committing Changes to a Container
- Pushing container to DockerHub
- Creating Local Image Repository
- Building an Image using Dockerfile

15. Configuration Management using Puppet

- Need of Configuration Management
- Configuration Management Tools
- What is Puppet
- Puppet Architecture
- Setting up Master Slave using Puppet
- Puppet Manifests
- Puppet Modules
- Applying configuration using Puppet
- Puppet File Server

Hands-on Exercise –

- Setting up Master Slave on AWS

- Testing Connection of nodes with Puppet
- Creating a Manifest
- Deploying Manifest on Node
- Creating a Module
- Deploying sample software on nodes using Puppet Modules and Manifests
- Implementing a File Server Module on Puppet

16. Azure DevOps and Azure Repos

- Azure DevOps architecture
- Key features
- Azure DevOps tools
- Azure DevOps organizations and projects
- Introduction to Azure Repos
- Compare TFVC and Git
- Key concepts in azure repos
- Search your code in Repos
- What is TFVC
- Azure Repos Integrations

17. Azure Pipelines

- What is Azure Pipelines
- Why use Azure Pipelines
- Deploying to Azure
- Key concepts in Pipelines
- CI Triggers in pipelines
- YAML Basics
- Ecosystems and Integration
- Setting up CI build
- Adding Tests to the Pipeline
- Agents and Tasks

18. Azure Boards

- What is Azure Boards
- Why use azure boards
- Agile project management best practices
- Basic concepts of Azure Boards

- Connecting boards to Github
- Work items
- Kanban boards
- Sprints
- Azure Boards integrations

19. AKS

- Create-AKS-Cluster
- Ingress-Basic
- Ingress - Context Path Based Routing
- Kubernetes-Namespaces
- Azure-VirtualNodes-for-AKS
- Azure-VirtualNodes-MixedMode-Deployments
- Azure-Container-Registry-ACR

20. Terraform with Azure IAC

- What is Infrastructure as a code
- IaC vs Configuration Management
- Introduction to Terraform
- Installing Terraform on Local and Vms
- Basic Operations in Terraform
  - init
  - validate
  - plan
  - apply
  - destroy
- Terraform Code Basics
- Deploying an end-to-end architecture on Azure using Terraform

21. Azure Artifacts and Azure Test Plans

- What are Azure artifacts
- Key concepts in artifacts
- working with packages
- Feeds

- views and upstream sources
- Connecting to azure pipelines
- What are Azure test plans
- Exploratory and manual testing
- Test from kanban board
- Creating Test Plans
- Testing web apps

22. MAVEN

- AGENDA
- WHY MAVEN?
- WHAT IS MAVEN?
- WHAT DOES MAVEN DO?
- BUILDING POM FILES
- MAVEN REPOSITORIES
- MAVEN INSTALLATION

23. SONARQUBE

- WHAT IS SOFTWARE TESTING?
- DYNAMIC TESTING
- STATIC TESTING
- WHAT IS SONARQUBE?
- FEATURES OF SONARQUBE
- SONARQUBE INSTALLATION

Prerequisites:

- Fundamental knowledge about Azure, version control, Agile software development, and core software development principles. It would be helpful to have experience in an organization that delivers software.

- To become a Microsoft Certified: Azure DevOps Engineer Expert, you must either earn the Azure Administrator Associate or Azure Developer Associate certification.
- Great pre-requisite courses for those certifications are Microsoft Azure Administrator (AZ-104) or Developing Solutions for Microsoft Azure (AZ-204)

Who can Attend:

- Candidates for this exam should have subject matter expertise working with people, processes, and technologies to continuously deliver business value.
- Responsibilities for this role include designing and implementing strategies for collaboration, code, infrastructure, source control, security, compliance, continuous integration, testing, delivery, monitoring, and feedback.
- A candidate for this exam must be familiar with both Azure administration and development and must be expert in at least one of these areas.

Number of Hours: 40hrs

Certification: AZ-400

Key Features:

- One to One Training
- Online Training
- Fastrack & Normal Track
- Resume Modification
- Mock Interviews
- Video Tutorials
- Materials
- Real Time Projects
- Virtual Live Experience
- Preparing for Certification
- 24/7 Support Team

DevOps:

1. What is DevOps?

- DevOps is a set of practices that combines software development (Dev) and IT operations (Ops) to shorten the development lifecycle and deliver high-quality software continuously.

2. Key DevOps Practices:

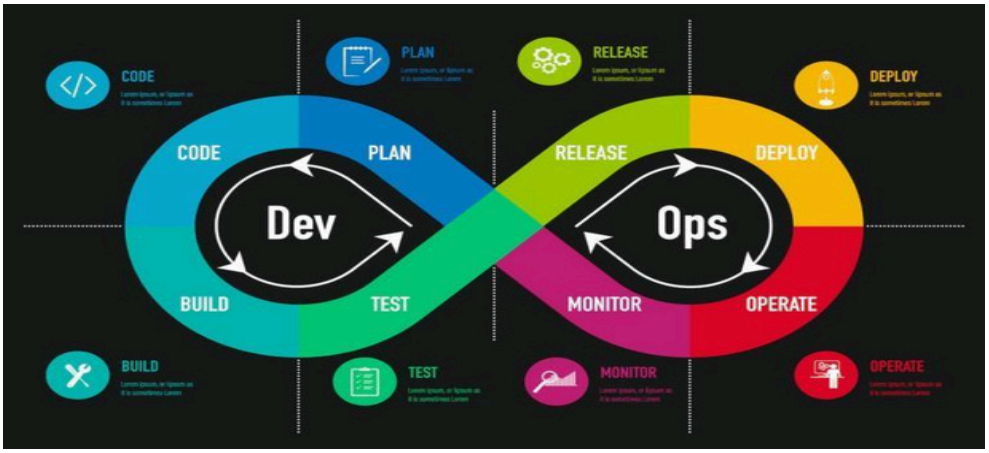
- **Continuous Integration (CI):** Automating the building and testing of code changes frequently.
- **Continuous Deployment (CD):** Automating the deployment of code changes to production environments.
- **Infrastructure as Code (IaC):** Managing and provisioning infrastructure using cod e.
- **Collaboration and Communication:** Fostering collaboration between development and operations teams.

3. Tools in DevOps:

- Terraform, Jenkins, Docker, Kubernetes, Ansible, Git, Shell scitpting and more.

4. Why Learn DevOps?

- DevOps practices improve collaboration, efficiency, and delivery speed.
- It helps in automating repetitive tasks, reducing errors, and improving the overall software development lifecycle.



Course Outline for Devops:

DevOps Essentials

- Why DevOps?
- What is DevOps?
- DevOps Market Trends
- DevOps Engineer Skills
- DevOps Delivery Pipeline
- DevOps Ecosystem
- Use Case
- Real time scenarios

Linux:

- Understanding Operating system structure. And File System Hierarchy.
- Basic and Advance commands
- Linux Standard Build.
- Device Management and Disk devices.
- File Systems and Mounting.
- Logical volume Management.
- RAID concepts.
- User Management and User profiles.
- Managing user quota and sudo.
- File permissions and access control list.
- Process Management
- Booting Process and Service Management.
- Package management using rpm and source packages
- Linux Networking and bonding.
- Remote Access tool SSH
- CRON and AT
- Setting up YUM server and Client
- Deep dive about Web server (APACHE)
- File sharing using NFS and AutoFS
- Firewall Management (IP tables)
- Understanding Kernel and Kernel modules.
- Centralized log management using RSYSLOG.
- NTP

GIT: Version Control

- What is Git
- About Version Control System and Types
- GIT Basics
- GIT Command Line
- Installing Git
- Installing on Linux
- Installing on Windows
- Creating repository
- Cloning, check-in and committing
- Fetch pull and remote
- Branching
- Creating the Branches, switching the branches, merging the branches.

Github

- Fork a project on github.
- Clone your github fork to your computer
- Create a topic branch for your own work in your local clone
- Commit changes to your local repository
- Push the changes to your github fork
- Send a pull request back to the original project

Infra as a code (Terraform)

- Introduction to Terraform
- Terraform variables and types
- Installation of Terraform and configure AWS account
- Writing Terraform script from scratch to build AWS infra
- Creating basic script for reusable
- Terraform loops and if statements
- Creating all the AWS important services using terraform
- Terraform Modules
- Remote state file
- Terraform import and workspace
- Sensitives in terraform

Containerization (Docker)

- Basic about Container Technologies
- Comparison with containers with virtualization
- Docker Architecture and work flow
- Installation and configuration of Docker Engine
- Docker Images and creating custom images
- Managing images and Docker hub
- Container managing commands
- Container Networking and storage devices
- Creating two tier application using containers
- Container Volumes
- Creating docker services
- Docker compose and Docker commit
- Docker registry

Kubernetes:

- Kubernetes key concepts
- Kubernetes Architecture
- kube API server
- Etcd
- Kube scheduler
- Kube controller
- Kubelet
- Kube proxy
- Deploying EKS cluster using AWS management console
- Deploying EKS cluster using Terraform automation
- Node groups and cluster auto scaling
- Setting up metrics server
- Creating ECR registry and repository
- Kubernetes services: cluster IP, node port and Load balancer
- Ingress controller using Nginx
- Setting up Lens for kubernetes IDE
- Pods
- Multi container pods and Init containers
- Deployment, Daemon set, Stateful set
- Jobs and cron jobs
- Configmap and Kubernetes secrets
- Horizontal pod autoscaler
- Rolling out and Rollback

- Setting up EBS storage-class and PVC using EBS CSI driver
- User management with service accounts, roles, role binding
- Releasing application across DEV, ACCEPTANCE, PRODUCTION kubernetes cluster
- Deploying multi-tier application on EKS cluster through CICD
- **Helm charts** basics and configuration
- **Argo CD** for Git ops approach
- **Blue green deployment** for zero down time application release

Continuous Integration Tools (JENKINS)

- Continuous integration fundamentals
- Installing a Jenkins environment
- Master and slave setup
- Managing plugins
- Jenkins jobs, builds and projects
- Deep dive into pipeline
- Pipe line creation from scratch
- Scripted pipeline vs Declarative pipeline
- Maven integration and builds
- Integrating with source repository
- Code quality and code coverage using **Sonar qube**
- Automated deployments

Continuous Integration suing (GIT LAB)

- Configuring Git for Gitlab CI
- Creating a new project
- Building the project using Gitlab CI
- Running jobs in parallel and background
- Deploying the project using Gitlab CI
- Pipeline triggers / Retrying failed jobs / Pipeline schedules
- Creating job templates
- Quick introduction to unit testing in CI pipelines
- How to structure a CI/CD pipeline in GitLab CI.

ArgoCD

- What is ArgoCD?
- Overview of Argo CD Features and Architecture
- Argo CD as a Kubernetes (K8S) controller
- Components (API server, repository server, controller)
- Kubernetes with GitOps

- Getting Started with Argo CD
- Configuring Argo CD
- Deploying an Application with Argo CD
- Deployment history and rollback
- Automated syncing
- Integrating Argo CD Into a CI/CD Pipeline
- CI/CD workflow completion

Repository manager (Nexus)

- Repository management
- Installing Nexus and integrate with maven build
- Configuring repository for build with Nexus
- Managing Nexus settings
- Nexus plugins
- Software release with Nexus staging.

SonarQube

- Installing SonarQube
- Analyzing SourceCode
- Plugin management
- Local and branch Analysis
- Sonar API Basics

Configuration Management (Ansible)

- Installing and configure Ansibe setup
- Ansible overview and docs
- Inventory management
- Running adhoc commands
- Different variables in ansible
- Writing playbooks
- Deep dive into ansible roles
- Ansible facts and vaults

Shellscripting (Shell)

- Shell Variables
- Shell operators and positional parameters
- Shell debugging, running and scheduling
- Background jobs

- Here statements
- Arrays
- Case statement
- Select statement
- Arithmetic operators
- Functions
- Select and break
- Many real-time examples

Project: Demo Pipeline creation

- Ci/cd with aws instance integrating Git, Terraform, Jenkins/GIT lab, Docker, kubernetes, argoCD, Sonar qube, nexus
- Various application deployments, work flow and trouble shooting

Interview Preparation:

- With real-time scenarios, tasks and projects
- Agile methodology, sprint planning and change execution methods
- Trouble shooting and Day to Day activities
- Weekly meet with students who got job and who currently looking for jobs to collaborate and helping existing students to talk in interview with good communication skills and technical knowledge