**Pre-requisite:**

* Experience in any scripting language like Java, PowerShell, and Python etc.
* Knowledge of traditional project deployment process
* Basic knowledge of Azure Cloud, Docker, Git and GitHub

**Note:** For interest of time of everyone, trainer will be doing hands-on demonstrations to showcase how it works. Trainees need to do hands-on after the session is over by following the recorded videos and hands-on instructions provided by trainer.

**Day 1**

**Module 1: Basics of Azure**

* Azure Overview
* What is a Resource Group?
* What is Region?
* How to create a Virtual Machine?

**Module 2: YAML Scripting**

* YAML – BASICS
  + Rules for Creating YAML file
  + Basic Components of YAML File
* YAML– INDENTATION AND SEPARATION
  + Indentation of YAML
  + Separation of Strings
  + YAML – COMMENTS
* YAML – FULL LENGTH EXAMPLE

**Module 3: GIT Concepts**

* Installing Git
* Git Concepts
* Git in Practice
  + Setting Up Your Profile
  + Getting a Git Repository
* Git by Example
  + Creating a Git repository on a shared server (git init)
  + Cloning a remote repository (git clone)
  + Adding files to a local Git repository (git add)
  + Checking for status of files in a local Git repository (git status)
  + Committing files to a local Git repository (git commit)
  + Pushing changes in a local git repository to a remote Git repository (git push)
  + Pulling files from a remote Git repository (git pull)
* Git Integration with Visual Studio

**Module 4: Docker Concepts**

* Start DevOps with Docker
  + DevOps, Docker and Containerization
  + Docker and DevOps - Installation and Introduction
  + First DevOps Use case with Docker
* Important Docker Concepts
  + Docker Images
  + Docker - Detached Mode and Logs
  + Docker Images and Containers
  + Learning Docker Images - Commands
  + Learning Docker Containers - Commands
  + Building Docker Images for Python Application
  + Pushing Python App Docker Image to Docker Hub

**Module 5: Getting Started with Source Control**

**Lessons**

* Introduction to Azure Repos
* Introduction to GitHub

**Lab : Version Controlling with Git in Azure Repos**

After completing this module, students will be able to:

* Describe the benefits of using Source Control
* Describe Azure Repos and GitHub

**Module 6: Managing Technical Debt**

**Lessons**

* Identifying Technical Debt
* Knowledge Sharing within Teams
* Modernizing Development Environments with Codespaces

**Lab : Sharing Team Knowledge using Azure Project Wikis**

After completing this module, students will be able to:

* Manage code quality including technical debt SonarCloud, and other tooling solutions
* Build organizational knowledge on code quality

**Day 2**

**Module 7: Working with Git for Enterprise DevOps**

**Lessons**

* How to Structure Your Git Repo
* Git Branching Workflows
* Collaborating with Pull Requests in Azure Repos
* Managing Git Repositories

**Lab : Version Controlling with Git in Azure Repos**

After completing this module, students will be able to:

* Describe Git branching workflows
* Leverage pull requests for collaboration and code reviews

**Module 8: Configuring Azure Pipelines**

**Lessons**

* The Concept of Pipelines in DevOps
* Azure Pipelines
* Azure DevOps and Open-Source Projects (Public Projects)
* Azure Pipelines YAML versus Visual Designer

**Lab : Understanding Pipeline**

After completing this module, students will be able to:

* Explain the role of Azure Pipelines and its components
* Configure Agents for use in Azure Pipelines

**Module 9: Implementing Continuous Integration using Azure Pipelines**

**Lessons**

* Continuous Integration Overview
* Implementing a Build Strategy
* Integration with Azure Pipelines
* Integrating External Source Control with Azure Pipelines

**Lab : Enabling Continuous Integration with Azure Pipelines**

**Lab : Integrating External Source Control with Azure Pipelines**

After completing this module, students will be able to:

* Explain why continuous integration matters
* Implement continuous integration using Azure Pipelines

**Module 10: Managing Application Configuration and Secrets**

**Lessons**

* Introduction to Security
* Implement a Secure Development Process
* Rethinking Application Configuration Data
* Manage Secrets, Tokens, and Certificates
* Integrating with Identity Management Systems
* Implementing Application Configuration

**Lab : Integrating Azure Key Vault with Azure DevOps**

After completing this module, students will be able to:

* Manage application configuration and secrets
* Integrate Azure Key Vault with a pipeline

**Module 11: Implementing Continuous Integration with GitHub Actions**

**Lessons**

* GitHub Actions
* Continuous Integration with GitHub Actions
* Securing Secrets for GitHub Actions

**Lab : GitHub Actions Continuous Integration**

After completing this module, students will be able to:

* Create and work with GitHub Actions and Workflows
* Implement Continuous Integration with GitHub Actions

**Day 3**

**Module 12: Designing and Implementing a Dependency Management Strategy**

**Lessons**

* Packaging Dependencies
* Package Management
* Migrating and Consolidating Artifacts
* Implementing a Versioning Strategy

**Lab : Package Management with Azure Artifacts**

**Module 13: Designing a Release Strategy**

**Lessons**

* Introduction to Continuous Delivery
* Release Strategy Recommendations
* Building a High-Quality Release pipeline
* Choosing the Right Release Management Tool

**Lab : Controlling Deployments using Release Gates**

**Module 14: Implementing Continuous Deployment using Azure Pipelines**

**Lessons**

* Create a Release Pipeline
* Provision and Configure Environments
* Manage and Modularize Tasks and Templates
* Configure Automated Integration and Functional Test Automation
* Automate Inspection of Health

**Lab : Configuring Pipelines as Code with YAML**

**Lab : Setting up and Running Functional Tests**

**Module 15: Implementing an Appropriate Deployment Pattern**

**Lessons**

* Introduction to Deployment Patterns
* Implement Blue Green Deployment
* Canary Releases

**Module 16: Managing Infrastructure and Configuration using Azure Tools**

**Lessons**

* Infrastructure as Code and Configuration Management
* Create Azure Resources using ARM Templates

**Lab : Azure Deployments using Resource Manager Templates**

**Day 4**

**Module 17: Third Party Infrastructure as Code Tools Available with Azure**

**Lessons**

* Ansible
* Terraform

**Lab : Automating Infrastructure Deployments in the Cloud with Terraform and Azure Pipelines**

**Lab : Ansible with Azure**

**Module 18: Managing Containers using Docker**

**Lessons**

* Implementing a Container Build Strategy
* Implementing Docker Multi-Stage Builds

**Lab : Modernizing Existing ASP.NET Apps with Azure**

After completing this module, students will be able to:

* Implement a container strategy including how containers are different from virtual machines and how microservices use containers
* Implement containers using Docker

**Module 19: Implementing Feedback for Development Teams**

**Lessons**

* Implement Tools to Track System Usage, Feature Usage, and Flow
* Develop Monitoring and Status Dashboards

**Module 20: Implementing System Feedback Mechanisms**

**Lessons**

* Design Processes to Capture and Analyze User Feedback
* Managing Alerts

**Lab : Integration between Azure DevOps and Teams**

**Module 21: Implementing Security in DevOps Projects**

**Lessons**

* Security in the Pipeline

**Lab : Implement Security and Compliance in an Azure DevOps Pipeline**

**Module 22: Validating Code Bases for Compliance**

**Lessons**

* Open-Source Software
* Managing Security and Compliance Policies
* Integrating License and Vulnerability Scans

**Lab : Managing Technical Debt with SonarQube and Azure DevOps**

**Module 23: Azure Test Plan**

* Preparing Test Plans, Executing Test Cases & Progress Reports
  + Azure Test Plans Service introduction
  + Creating Test Plans and Test Cases
  + Tracking test executions
  + Using Progress reports and Runs views