



# Microsoft Azure DevOps

## About SNIT

SNIT is an Institute dedicated to helping students get their careers to succeed through Computer Networking, Microsoft Azure Cloud, AWS Cloud, Web Development, Digital Marketing Courses, and Software Training programs

We provide course certification training programs in Hyderabad for Cisco, Microsoft Azure, AWS, and many more.

We have partnered with various companies and agencies to help place our students in the best matching position with Companies.

## SNIT Services

SNIT Solutions is a Private Limited organization. We provide managed IT Services

Digital Marketing Agency that focuses on growing your business successfully.

Corporate Training services in various domains including Information Technology.

## Azure DevOps Curriculum

### Description

DevOps has become increasingly critical to a team's success these days. To help developers ship software faster and with higher quality, Microsoft provides its tool, Azure DevOps. It's one of the most complete offerings in the public cloud, which works great for any type of application regardless of the framework or platform. The objective of this course is to learn how we can implement Azure infrastructure deployment using Azure DevOps.

### Learning Objective:

Introduction - Course Introduction

Overview of Cloud Computing and types of clouds

Setting Up Development Environment



# Microsoft Azure DevOps

Introduction GitLab

Setup GitLab Pipeline using GIT Runner Continuous integration and continuous Deployment

Setup Azure and Azure DevOps - Understand the fundamentals of Azure DevOps and its basic setup

Azure Repos - Understand about version control capabilities of Azure DevOps

Azure Pipelines - Understand how to use Azure DevOps for Azure Resource provisioning.

Pipeline customization - Understand how to bring in management and control within Azure Pipelines

Production deployments - Understand how production deployment scenarios can be addressed

## **Module 1 Introduction to DevOps**

What is Software Development Software Development Life Cycle

Traditional Models for SDLC

What is the Agile Model

Why DevOps?

DevOps Tools

## **Module 2 Software Version Control**

What is Version Control?

Types of Version Control System

Introduction to Git

Git three-stage architecture



Common Git Commands

Working with Branches

Merge/Clone

## **Hands-On**

Common Git Commands

Creating a local repository

Cloning a remote repository

Git Push/

Merging Branches

Branch Creating the Branches, switching the branches, merging the branches.

## **Module 3 GIT HUB**

Creating GitHub accounts and repositories

Push code from local to remote repos

Pull code from remote repo to local repos

User collaboration and permissions on GitHub Repo

Generate public key and add to GitHub

Check-in the code without credentials from local to remote

## **Module 5 Azure DevOps and Portal**

Azure DevOps architecture key features

Azure DevOps organizations

Overview of Organization

Overview of projects

Types of project

## **Hands-on**



Create Azure DevOps Portal

Create an Organization

Create a new Project

Teams in Azure DevOps

Add Users and Manage Users

## **Module 6 Azure Boards**

What is Azure Boards

Basic Work Process

Overview of Agile Planning Tools

Connecting boards

Kanban boards

Work items

Backlogs

Sprints

Scrum and plans

## **Hands-on**

create a few work items

Create EPIC

Create few features

create a few user stories

Create a backlog for your project.

Set up iterations

## **Module 7 Azure Repo**

Azure Repos



Key concepts in azure repos

Azure Artifacts

## **Hands-on**

Deploy a web app code to Repo

Built Service Connection

## **Module 8 Azure Pipelines**

Azure Pipelines What is Azure Pipelines

Why use Azure Pipelines

Key Concepts in Pipelines

## **Hands-on**

Deploying to Azure

CI Triggers in pipelines

YAML Basics

Setting up CI build

Adding Tests to the Pipeline

Implement CI/CD in Azure DevOps

Create Azure resources like web apps and deploy the application using Pipelines.

## **Module 4 Jenkins – Continuous Integration**

Introduction to Continuous Integration

Introduction about Jenkins

Jenkins Architecture

Understanding CI/CD Pipelines



## Hands-on

Installing Jenkins and configuring Jenkins

Java installation and configuration

Maven Installation

Working Jenkins DashBoard

Create a Job Built a Job

Setting up the Global Environments for Jobs

Adding and updating Plugins

Tomcat installation and configuration

## Module 9 Continuous Deployment: Containerization with Docker

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

## Module 10 Kubernetes: Container Orchestration tool

- Introduction to ContainerOrchestration
- Kubernetes Core Concepts
- Understanding Pods
- ReplicaSet and Replication Controller
- Deployments
- DaemonSets

## Hands-On:

- Kubectl Common Commands



- Deployments
- DaemonSets
- Scaling in Kubernetes

## Module 11 Terraform with Azure

- Introduction to Terraform
- What is Infrastructure as a code?
- IaC vs Configuration Management

### Hands-On:

- Installing Terraform on AWS
- Basic Operations in Terraform
- Deploy Resources through Code

## Module 12 Sonarqube

- What Is Software Testing?
- Features Of Sonarqube Sonarqube

### Hands-On

Installation Sonarqube

Test a Web application

## Module 13 Maven

Why Maven?

What Is Maven?

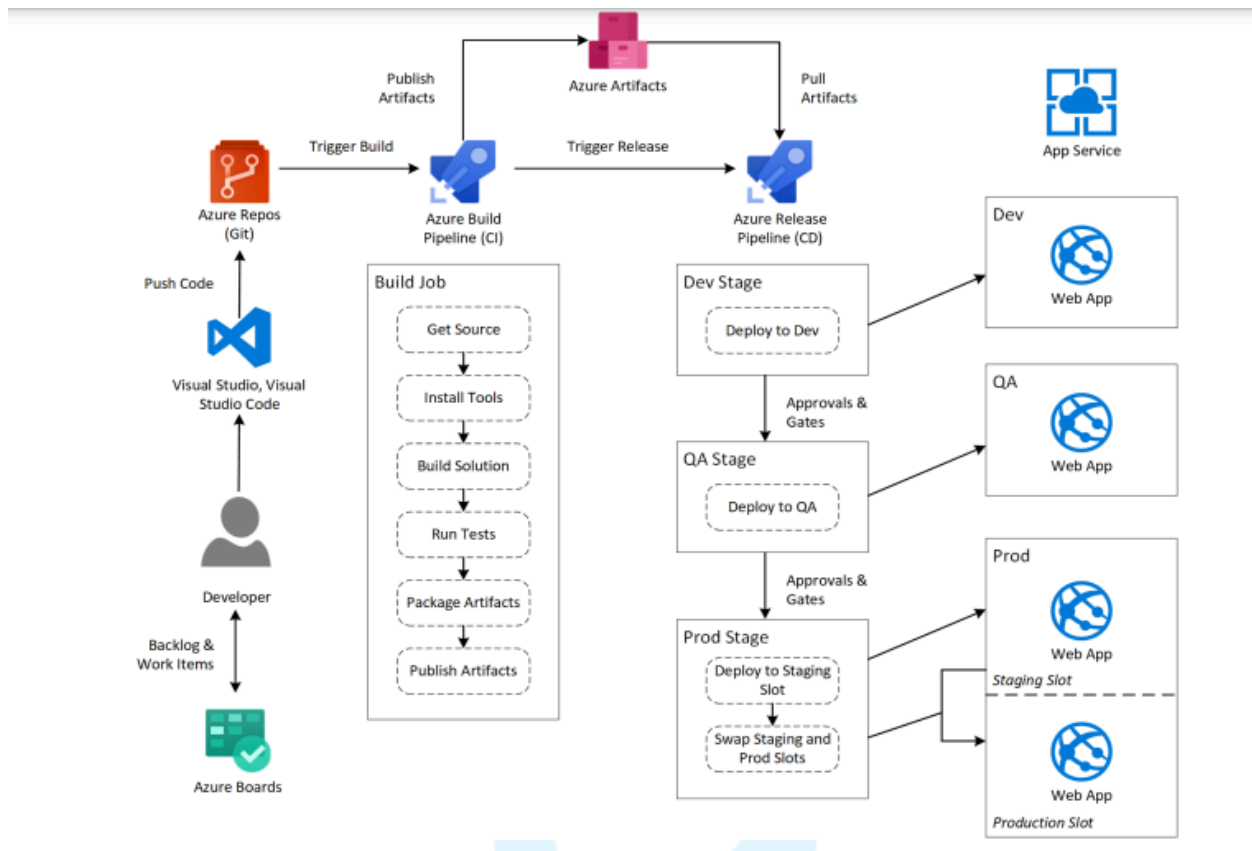
What Does Maven Do?

### Hands-on

- Maven Installation
- Maven Repositories
- Building Pom Files

## Azure DevOps Project Architecture

The primary purpose of application architecture is to support the development life cycle of the system. Good architecture makes the system easy to understand, easy to develop, easy to maintain, and easy to deploy. The ultimate benefit of the architecture is to minimize the lifetime cost of the system and to maximize programmer productivity. The goal of the given architecture is to learn and implement important principles and design patterns to build various domain applications like eCommerce, edTech, Food Delivery, or other Service-based industries applications.



## What will you learn through the project?

After the project, attendees will be able to;





# Microsoft Azure DevOps

- Create Sprint Plans and Tracking.
- Manage backlogs, Bugs, and Capacity planning.
- Manage source code using Git or TFS.
- Create Build pipelines for the dev environment.
- Create release pipelines for code deployment to production.
- Publish own packages using Artifacts.
- Monitoring Applications.

## **Hardware Requirements:**

Memory – Minimum 8 GB RAM

Processor – Intel Core i3 CPU @2.00 GHz or later

Storage – 250 GB HDD/SDD or later

## **Software Requirements:**

Operating System – Windows 8.1 or above, Ubuntu 14 or later

Windows PowerShell 4.0 or later (Install Azure Module)

Microsoft Azure SDK for .NET v2.9 (prefer latest)

How will I execute the practicals?

You will be executing all the practicals on a free tier Azure account, which we will be creating during our class.

Teaching Contact Hours

Total duration- 30 days, 6 days/week, and 1.5 hours/day. Theory and Particles

Entry Requirements

This course can be taken by Technical and NON-Technical Graduates.

How will this course enhance my career Growth?



# Microsoft Azure DevOps

This course will help you become either one of the following,

Azure Administrator

Azure DevOps Engineer Expert

Azure Developer Associate

Azure Solutions Architect Expert

Azure Security Engineer Associate