

Microsoft Azure DevOps (Customized)

Pre-requisite

- Cloud computing concepts, including an understanding of PaaS, SaaS, and IaaS implementations.
- Both Azure administration and Azure development with proven expertise in at least one of these areas.
- Version control, Agile software development, and core software development principles. It would be helpful to have experience in an organization that delivers software.

Course Outline

Day 1.1 (Half Day)

Module 1: Planning for DevOps

Lessons

- Transformation Planning
- Project Selection
- Team Structures
- Migrating to Azure DevOps

Lab : Agile Planning and Portfolio Management with Azure Boards

After completing this module, students will be able to:

- Plan for the transformation with shared goals and timelines
- Select a project and identify project metrics and Key Performance Indicators (KPI's)
- Create a team and agile organizational structure
- Design a tool integration strategy
- Design a license management strategy (e.g. Azure DevOps and GitHub users)
- Design a strategy for end-to-end traceability from work items to working software
- Design an authentication and access strategy

Module 2: Getting Started with Source Control & Working with Git

Lessons

- Types of Source Control Systems
- Introduction to Azure Repos
- How to Structure Your Git Repo
- Git Branching Workflows
- Collaborating with Pull Requests in Azure Repos

Lab : Version Controlling with Git in Azure Repos

Module 3: Configuring Azure Pipelines

Lessons

- The Concept of Pipelines in DevOps
- Azure Pipelines
- Evaluate use of Hosted versus Self-Hosted Agents
- Agent Pools
- Pipelines and Concurrency
- Azure DevOps and Open-Source Projects (Public Projects)
- Azure Pipelines YAML versus Visual Designer

Day 1.2 (Half Day)

Lab : Configuring Agent Pools and Understanding Pipeline Styles

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 4: 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
- Set Up Self-Hosted Agents

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 5: 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

Lab : Creating a Release Dashboard

After completing this module, students will be able to:

- Differentiate between a release and a deployment
- Define the components of a release pipeline



- Explain things to consider when designing your release strategy
- Classify a release versus a release process, and outline how to control the quality of both
- Describe the principle of release gates and how to deal with release notes and documentation
- Choose a release management tool

Day 2

Module 6: 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

After completing this module, students will be able to:

- Explain the terminology used in Azure DevOps and other Release Management Tooling
- Describe what a Build and Release task is, what it can do, and some available deployment tasks
- Explain why you sometimes need multiple release jobs in one release pipeline
- Differentiate between multi-agent and multi-configuration release job
- Use release variables and stage variables in your release pipeline
- Deploy to an environment securely using a service connection
- List the different ways to inspect the health of your pipeline and release by using alerts, service hooks, and reports

Module 7: 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 8: Designing and Implementing a Dependency Management Strategy

Lessons

- Packaging Dependencies
- Package Management

- Migrating and Consolidating Artifacts
- Package Security
- Implementing a Versioning Strategy

Lab : Package Management with Azure Artifacts

After completing this module, students will be able to:

- Recommend artifact management tools and practices
- Abstract common packages to enable sharing and reuse
- Migrate and consolidate artifacts
- Migrate and integrate source control measures

Day 3

Module 9: Implementing an Appropriate Deployment Pattern

Lessons

- Introduction to Deployment Patterns
- Implement Blue Green Deployment
- Feature Toggles
- Canary Releases
- Dark Launching
- AB Testing
- Progressive Exposure Deployment

Lab : Feature Flag Management with LaunchDarkly and Azure DevOps

After completing this module, students will be able to:

- Describe deployment patterns
- Implement Blue Green Deployment
- Implement Canary Release
- Implement Progressive Exposure Deployment

Module 10: Managing Infrastructure and Configuration using Azure Tools

Lessons

- Infrastructure as Code and Configuration Management
- Create Azure Resources using Azure CLI
- Azure Automation with DevOps
- Desired State Configuration (DSC)

Lab : Azure Deployments using Azure Command Line Interface

After completing this module, students will be able to:

- Apply infrastructure and configuration as code principles.
- Deploy and manage infrastructure using Microsoft automation technologies such as PowerShell and Azure CLI

Module 11: Third Party Infrastructure as Code Tools Available with Azure

Lessons

- Terraform

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

Module 12: Managing Containers using Docker

Lessons

- Azure Docker
 - Docker Architecture
 - Docker Images
- Implementing a Container Build Strategy
- Implementing Docker Multi-Stage Builds

Lab : Modernizing Existing 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
- Implement Docker multi-stage builds

Day 4

Module 13: Creating and Managing Kubernetes Service Infrastructure

Lessons

- Azure Docker and Kubernetes Fundamentals
 - Vagrant Introduction
 - Building and Provisioning VM with Vagrant
- Azure Kubernetes Service
 - Containerization and Orchestration
 - Clusters
 - Storage
 - Container Registry
 - Kubernetes Secrets
- Kubernetes Tooling
- Integrating AKS with Pipelines
- Monitoring AKS

Lab : Deploying a Multi-Container Application to Azure Kubernetes Service

After completing this module, students will be able to:

- Deploy and configure a Managed Kubernetes cluster

Module 14: Implementing Feedback for Development Teams

Lessons

- Monitoring

- Types of Monitoring
- Third Party Tools for Monitoring – Ex. Datadog
- Investigating Application Crashes
- Implement Tools to Track System Usage, Feature Usage, and Flow
- Develop Monitoring and Status Dashboards
- Integrate and Configure Ticketing Systems

Lab : Monitoring Application Performance with Application Insights

After completing this module, students will be able to:

- Implement tools to track system usage, feature usage, and flow
- Configure crash report integration for client applications
- Implement routing for client application crash report data
- Develop monitoring and status dashboards
- Integrate and configure ticketing systems with development team's work management

Module 15: Implementing Security in DevOps Projects

Lessons

- Security in the Pipeline
- Azure Security Center

Lab : Implement Security and Compliance in an Azure DevOps Pipeline

After completing this module, students will be able to:

- Define an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructure
- Implement compliance and security in your application infrastructure

Module 16: 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

After completing this module, students will be able to:

- Describe the potential challenges with integrating open-source software
- Inspect open-source software packages for security and license compliance
- Manage organizational security and compliance policies
- Integrate license and vulnerability scans into build and deployment pipelines

Configure build pipelines to access package security and license ratings

Module 17: Implementing System Feedback Mechanisms

Lessons

- Site Reliability Engineering
- Design Practices to Measure End-User Satisfaction
- Design Processes to Capture and Analyze User Feedback
- Design Processes to Automate Application Analytics



- Managing Alerts
- Blameless Retrospectives and a Just Culture

Lab : Integration between Azure DevOps and Teams

After completing this module, students will be able to:

- Define Site Reliability Engineering
- Design processes to measure end-user satisfaction and analyze user feedback
- Design processes to automate application analytics
- Manage alerts and reduce meaningless and non-actionable alerts
- Carry out blameless retrospectives and create a just culture