Designing and Implementing Microsoft DevOps solutions

Course ID : AZ-400T00-A

Overview :

This course provides the knowledge and skills to design and implement DevOps processes and practices. Students will learn how to plan for DevOps, use source control, scale Git for an enterprise, consolidate artifacts, design a dependency management strategy, manage secrets, implement continuous integration, implement a container build strategy, design a release strategy, set up a release management workflow, implement a deployment pattern, and optimize feedback mechanisms.

Objectives :

- Plan for the transformation with shared goals and timelines
- Select a project and identify project metrics and KPIs
- Create a team and agile organization structure
- Describe the benefits of using Source Control
- Deploy and manage infrastructure using Microsoft automation technologies such as ARM templates, PowerShell, and Azure CLI
- Describe deployment models and services that are available with Azure
- Deploy and configure a Managed Kubernetes cluster

Audience(s) :

- Everyone who is interested in designing and implementing DevOps processes or in passing the Microsoft Azure DevOps Solutions certification exam.

Level :

- Advanced

Course Duration :

- 5 Days

Delivery Mode:

- Virtual & Classroom Training

Location:

- Tunisia / Bahrain

Language:

- English / French

Exam:

- AZ-400

Certification

- N/A

Course Outline :

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

Module 2: Getting started with Source Control Lessons

- What is Source Control
- Benefits of Source Control
- Types of Source Control Systems
- Introduction to Azure Repos
- Introduction to GitHub
- Migrating from Team Foundation Version Control (TFVC) to Git in Azure Repos
- Authenticating to Git in Azure Repos

- Lab: Version Controlling with Git

Module 3: Scaling Git for enterprise DevOps Lessons

- How to Structure your Git Repo
- Git Branching Workflows
- Collaborating with Pull Requests in Azure Repos
- Why care about GitHooks
- Fostering Inner Source

- Lab: Code Review with Pull Requests

Module 4: Consolidating Artifacts & Designing a Dependency Management Strategy Lessons

- Packaging Dependencies
- Package Management
- Migrating and Consolidating Artifacts

- Lab: Updating Packages

Module 5: Implementing Continuous Integration with Azure Pipelines Lessons

- The concept of pipelines in DevOps
- Azure Pipelines
- Evaluate use of Hosted vs Private Agents
- Agent Pools
- Pipelines and Concurrency
- Azure DevOps and Open Source Projects (Public Projects)

- Lab: Enabling Continuous Integration with Azure Pipelines
- Lab: Integrating External Source Control with Azure Pipelines

Module 6: Managing Application Config and Secrets Lessons

- Introduction to Security
- Implement secure and compliant development process
- Rethinking application config data
- Manage secrets, tokens, and certificates
- Implement tools for managing security and compliance in a pipeline

- Lab: Integrating Azure Key Vault with Azure DevOps

Module 7: Managing Code Quality and Security Policies Lessons

- Managing Code Quality
- Managing Security Policies

- Lab: Managing Technical Debt with Azure DevOps and SonarCloud

Module 8: Implementing a Container Build Strategy Lessons

- Implementing a Container Build Strategy

- Lab: Modernizing Existing ASP.NET Apps with Azure

Module 9: Manage Artifact versioning, security & compliance Lessons

- Package security
- Open-source software
- Integrating license and vulnerability scans
- Implement a versioning strategy

- Lab: Manage Open-Source Security and License with White Source

Module 10: Design a Release Strategy Lessons

- Introduction to Continuous Delivery
- Release strategy recommendations
- Building a High-Quality Release pipeline
- Choosing a deployment pattern
- Choosing the right release management tool

Module 11: Set up a Release Management Workflow Lessons

- Create a Release Pipeline
- Provision and Configure Environments
- Manage and Modularize Tasks and Templates
- Integrate Secrets with the release pipeline
- Configure Automated Integration and Functional Test

Module 12: Implement an appropriate deployment pattern Lessons

- Introduction to Deployment Patterns
- Implement Blue Green Deployment
- Feature Toggles
- Canary Releases
- Dark Launching

- Lab: Feature Flag Management with Launch Darkly and Azure DevOps

Module 13: Implement process for routing system feedback to development teams Lessons

- Implement Tools to Track System Usage, Feature Usage, and Flow
- Implement Routing for Mobile Application Crash Report Data
- Develop Monitoring and Status Dashboards
- Integrate and Configure Ticketing Systems

- Lab: Monitoring Application Performance

Module 14: Infrastructure and Configuration Azure Tools Lessons

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

- Lab: Azure Deployments using Resource Manager Templates

Module 15: Azure Deployment Models and Services Lessons

- Deployment Modules and Options
- Azure Infrastructure-as-a-Service (IaaS) Services
- Azure Platform-as-a-Service (PaaS) services
- Serverless and HPC Computer Services
- Azure Service Fabric

- Lab: Deploying a Dockerized Java app to Azure Web App for Containers

Module 16: Create and Manage Kubernetes Service Infrastructure Lessons

- Azure Kubernetes Service

Module 17: Third Party Infrastructure as Code Tools available with Azure Lessons

- Chef
- Puppet
- Ansible
- Terraform

- Lab: Infrastructure as Code
- Lab: Automating Your Infrastructure Deployments in the Cloud with Terraform and Azure Pipelines

Module 18: Implement Compliance and Security in your Infrastructure Lessons

- Security and Compliance Principles with DevOps
- Azure security Center
-
- Lab: Implement Security and Compliance in an Azure DevOps Pipeline

Module 19: Recommend and design system feedback mechanisms Lessons

- The inner loop
- Continuous Experimentation mindset
- Design practices to measure end-user satisfaction
- Design processes to capture and analyze user feedback
- Design process to automate application analytics

- Lab: Integration between Azure DevOps and Teams

Module 20: Optimize feedback mechanisms Lessons

- Site Reliability Engineering
- Analyze telemetry to establish a baseline
- Perform ongoing tuning to reduce meaningless or non-actionable alerts
- Analyze alerts to establish a baseline
- Blameless Retrospectives and a Just Culture