Microsoft Azure DevOps

AZ - 400 Microsoft Azure DevOps Training Course Content

Module1: Session 1: Design a DevOps strategy

Migration and consolidation strategy - DevOps tools

➢ Analyse existing artifact - deployment packages, NuGet, Maven, npm

➢ Container repositories

➢ Test management tools

➢ Recommend migration and integration strategies

- o Artifact repositories
- o Source control
- o Test management
- o Work management

➢ Work management tools

Module2: Session 2: Understanding Agile work management approach

➢ Identify and recommend project metrics, KPIs, and DevOps measurements

➢ Agile work management

➢ Mentor team members on Agile techniques and practices

➢ Scaling Agile practices

➢ Understanding in-team and cross-team collaboration mechanisms

Module3: Session 3: Design a quality strategy

➢ Analyze existing quality environment

➢ Working quality metrics

➢ Feature flag lifecycle

Microsoft Azure DevOps

➢ Measuring and managing technical debt

➢ Changes to team structure to optimize quality

➢ Recommend performance testing strategy

Module4: Session 4: Design a secure development process

➢ Inspect and validate code base for compliance

➢ Inspect and validate infrastructure for compliance

➢ Secure development strategy

➢ Integrate code security validation - static code analysis

➢ Integrate infrastructure security validation

Module5: Session 5: Design a tool integration strategy

➢ To design a license management strategy

- VSTS users
- concurrent pipelines
- test environments,
- open source software licensing
- third-party DevOps tools and services
- package management licensing

➢ Design a strategy for end-to-end traceability from work items to working software

➢ Integrating monitoring and feedback to development teams

➢ Authentication and access strategy

➢ Integrating on-premises and cloud resources

Module6: Session 6: Implement DevOps development processes

Design a version control strategy

➢ Working with Branching models

➢ Introduction to Version control systems

➢ Understaning Code flow strategy

Microsoft Azure DevOps

Module7: Session 7: Implement and integrate source control

➢ External source control

➢ Integrate source control into third-party continuous integration and continuous deployment (CI/CD) systems

Module8: Session 8: Implement and manage build infrastructure

➢ Private and hosted agents

➢ Working with third party build systems

➢ Concurrent pipelines

➢ Manage Azure pipeline configuration

- Agent queues
- Service endpoints
- Pools
- Webhooks

Module9: Session 9: Implement code flow

➢ Pull request strategies

➢ Branch and fork strategies

➢ configure branch policies

Module10: Session 10: Implement a mobile DevOps strategy

➢ Manage mobile target device sets and distribution groups

➢ Target UI test device sets

➢ Provision tester devices for deployment

➢ Create public and private distribution groups

Module11: Session 11: Managing application configuration and secrets

➢ Secure and compliant development process

➢ General (non-secret) configuration data

➢ secrets, tokens, and certificates

➢ applications configurations

- Web App
- Azure Kubernetes Service
- containers

➢ Secrets management

- Web App
- Azure Kubernetes Service
- containers
- Azure Key Vault

➢ Managing security and compliance in the pipeline

Module12: Session 12: Implement continuous integration

Manage code quality and security policies

➢ Monitor code quality

➢ Configure build to report on code coverage

➢ Automated test quality

➢ Test suites and categories

➢ Monitor quality of tests

➢ Security analysis tools

- SonarQube,
- White Source Bolt
- Open Web Application Security Project

Module13: Session 13: Implement a container build strategy

➢ create deployable images

- Docker
- Hub
- Azure Container Registry

➢ Docker multi-stage builds

Module14: Session14: Implement a build strategy

Microsoft Azure DevOps

➢ Design build triggers, tools, integrations, and workflow

➢ Hybrid build process

➢ Multi-agent builds

➢ Build tools and configuration (e.g. Azure Pipelines, Jenkins)

➢ set up an automated build workflow

Module15: Session 15: Implement continuous delivery

Design a release strategy

➢ Release tools

➢ Identify and recommend release approvals and gates

➢ Measuring quality of release and release process

➢ Recommend strategy for release notes and documentation

➢ select appropriate deployment pattern

Module16: Session 16: Set up a release management workflow

➢ Automate inspection of health signals for release approvals by using release gates

➢ Configure automated integration and functional test execution

➢ Create a release pipeline

- Azure Kubernetes Service
- Service Fabric
- WebApp

➢ Create multi-phase release pipelines

➢ Integrate secrets with release pipeline

➢ Provision and configure environments

➢ Manage and modularize tasks and templates - task and variable groups

Module17: Session17: Implement an appropriate deployment pattern

➢ Implement blue-green deployments

➢ Implement canary deployments

➢ Implement progressive exposure deployments

➢ Scale a release pipeline to deploy to multiple endpoints

- Deployment groups
- Azure Kubernetes Service
- Service Fabric

Module18: Session 18: Implement dependency management

Design a dependency management strategy

➢ Artifact management tools and practices (Azure Artifacts, npm, Maven, Nuget)

➢ Abstract common packages to enable sharing and reuse

➢ Inspect codebase to identify code dependencies that can be converted to packages

➢ Standardized package types and versions across the solution

➢ Refactor existing build pipelines to implement version strategy that publishes packages

Module19: Session 19: Manage security and compliance

➢ Inspect open source software packages for security and license compliance to align with corporate standards (e.g., GPLv3)

➢ Configure build pipeline to access package security and license rating (e.g., Black Duck, White Source)

➢ Configure secure access to package feeds

Module20: Session 20: Implement application infrastructure

Design an infrastructure and configuration management strategy

➢ Existing and future hosting infrastructure

➢ Infrastructure as Code (IaC) technologies

➢ managing technical debt on templates

➢ Transient infrastructure for parts of a delivery lifecycle

➢ Mitigate infrastructure state drift

Module21: Session 21: Implement Infrastructure as Code (IaC)

➢ Create nested resource templates

➢ Manage secrets in resource templates

➢ Provision Azure resources

➢ Recommend an Infrastructure as Code (IaC) strategy

➢ Recommend appropriate technologies for configuration management

- ARM Templates
- Terraform
- Chef
- Puppet
- Ansible

Module22: Session 22: Manage Azure Kubernetes Service infrastructure

➢ Provision Azure Kubernetes Service - ARM templates, CLI

➢ Create deployment file for publishing to Azure Kubernetes Service - kubectl, Helm

➢ Develop a scaling plan

Module23: Session 23: Implement infrastructure compliance and security

➢ Compliance and security scanning

➢ Prevent drift by using configuration management tools

➢ Automate configuration management by using PowerShell Desired State Configuration (DSC)

➢ Automate configuration management by using a VM Agent with custom script extensions

➢ Set up an automated pipeline to inspect security and compliance

Module24: Session 24: Implement continuous feedback

Recommend and design system feedback mechanisms

➢ Design practices to measure end-user satisfaction - Send a Smile, app analytics

➢ Design processes to capture and analyze user feedback from external sources - Twitter, Reddit, Help Desk

➢ Design routing for client application crash report data

➢ Monitoring tools and technologies

➢ Feature usage tracking tools

Module25: Session 25: Implement process for routing system feedback to development teams

➢ Configure crash report integration for client applications

➢ Develop monitoring and status dashboards

➢ Implement routing for client application crash report data

➢ Implement tools to track system usage, feature usage, and flow

➢ Integrate and configure ticketing systems with development team's work management system

- IT Service Management connector
- ServiceNow Cloud Management
- App Insights work items

Module26: Session 26: Optimize feedback mechanisms

➢ Analyze alerts to establish a baseline

➢ Analyze telemetry to establish a baseline

➢ Perform live site reviews and capture feedback for system outages

➢ Perform ongoing tuning to reduce meaningless or non-actionable alerts

Module27: Real time example and interview question and answer.