AZ-400: Designing and Implementing Microsoft DevOps solutions

Microsoft - Cloud

Live Training (também disponível em presencial)

- **Localidade:** Aveiro
- **Data:** 09 Nov 2020
- **Preço:** 1890 € (Os valores apresentados não incluem IVA. Oferta de IVA a particulares e estudantes.)
- **Horário:** Laboral das 9h30 - 17h30
- **Nível:** Avançado
- **Duração:** 45h

Sobre o curso

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.

After completing this course, students will be able to:

- 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
- Migrate from TFVC to Git
- Scale Git for Enterprise DevOps
- Recommend artifact management tools and practices
- Abstract common packages to enable sharing and reuse
- Migrate and consolidate artifacts
- Migrate and integrate source control measures
- Manage application config and secrets
- Develop a project quality strategy
- Plan for secure development practices and compliance rules
- Implement and manage build infrastructure

- Explain why continuous integration matters
- Implement continuous integration using Azure DevOps
- Manage code quality including: technical debt, SonarCloud, and other tooling solutions
- Manage security policies with open source, OWASP, and WhiteSource Bolt
- Implement a container strategy including how containers are different from virtual machines and how microservices use containers
- Implement containers using Docker
- Inspect open source software packages for security and license compliance to align with corporate standards
- Configure build pipeline to access package security and license rating
- Configure secure access to package feeds
- Inspect codebase to identify code dependencies that can be converted to packages
- Identify and recommend standardized package types and versions across the solution
- Refactor existing build pipelines to implement version strategy that publishes packages
- Manage security and compliance
- 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
- Explain deployment patterns, both in the traditional sense and in the modern sense
- Choose a release management tool
- 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
- Classify an Agent, Agent Queue, and Agent Pool
- 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
- Embed testing in the pipeline
- List the different ways to inspect the health of your pipeline and release by using alerts, service hooks, and reports
- Create a release gate
- Describe deployment patterns
- Implement Blue Green Deployment
- Implement Canary Release
- Implement Progressive Exposure Deployment
- 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
- Implement a mobile DevOps strategy
- Apply infrastructure and configuration as code principles.
- 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
- Deploy and configure infrastructure using 3rd party tools and services with Azure, such as Chef, Puppet, Ansible, SaltStack, and Terraform
- Define an infrastructure and configuration strategy and appropriate toolset for a release pipeline and application infrastructure
- Implement compliance and security in your application infrastructure
- Design practices to measure end-user satisfaction
- Design processes to capture and analyze user feedback from external sources
- Design routing for client application crash report data
- Recommend monitoring tools and technologies
- Recommend system and feature usage tracking tools
- 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

---

Destinatários

- Students interested in planning and implementing DevOps processes and projects or in passing the Microsoft Azure DevOps Solutions certification exam.

Pré-requisitos

- Fundamental knowledge about Azure, version control, Agile software development, and core software development principles. It would be helpful to have experience in an organization that delivers software.

Programa

- Planning for DevOps
- Getting started with Source Control
- Scaling git for enterprise DevOps
- Consolidating Artifacts & Designing a Dependency Management Strategy
- Managing application config & secrets
- Planning for Quality and Security
- Implement & Manage Build Infrastructure
- Implementing Continuous Integration in an Azure DevOps Pipeline
- Managing Code Quality and Security Policies
- Implementing a Container Build Strategy
- Manage Artifact versioning, security & compliance
- Design a Release Strategy
- Set up a Release Management Workflow
- Implement an appropriate deployment pattern
- Implement process for routing system feedback to development teams
- Infrastructure and Configuration Azure Tools
- Azure Deployment Models and Services
- Create and Manage Kubernetes Service Infrastructure
- Third Party and Open Source Tools available with Azure
- Implement Compliance and Security in your Infrastructure
- Recommend and design system feedback mechanisms
- Optimize feedback mechanisms

Module 1: Planning for DevOps

Lessons

- Transformation Planning
- Project Selection
- Team Structures

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 KPIs
- Create a team and agile organizational structure

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 GitLab : Integrating Azure Repos and Azure Pipelines with Eclipse

After completing this module, students will be able to:

- Describe the benefits of using Source Control
- Migrate from TFVC to 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 Internal Source

Lab : Code Review with Pull Requests

After completing this module, students will be able to:

- Scale Git for Enterprise DevOps

Module 4: Consolidating Artifacts & Designing a Dependency Management Strategy

Lessons

- Packaging Dependencies
- Package Management

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

Module 5: Managing application config & 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

After completing this module, students will be able to:

- Manage application config and secrets

Module 6: Planning for Quality and Security

Lessons

- Planning a Quality Strategy
- Planning Secure Deveopment

Lab : Feature Flag Management with LaunchDarkly and Azure DevOps

After completing this module, students will be able to:

- Develop a project quality strategy
- Plan for secure development practices and compliance rules

Module 7: Implement & Manage Build Infrastructure

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 Soruce Projects (Public Projects)
- Azure Pipelines YAML vs Visual Designer
- Set Up Private Agents
- Integrate Jenkins with Azure Pipelines

- Integrate External Source Contorl with Azure Pipelines
- Analyze and Integrate Docker Multi-Stage Builds

After completing this module, students will be able to:

- Implement and manage build infrastructure

Module 8: Implementing Continuous Integration in an Azure DevOps Pipeline

Lessons

- Continuous Integration Overview
- Implementing a Build Strategy

Lab : Enabling Continuous Integration with Azure PipelinesLab : Creating a Jenkins Build Job and Triggering CI

After completing this module, students will be able to:

- Explain why continuous integration matters
- Implement continuous integration using Azure DevOps

Module 9: Managing Code Quality and Security Policies

Lessons

- Managing Code Quality
- Managing Security Policies

Lab : OWASPLab : Managing Technical Debt with Azure DevOps and SonarCloud

After completing this module, students will be able to:

- Manage code quality including: technical debt SonarCloud, and other tooling solutions.
- Manage security policies with open source, OWASP, and WhiteSource Bolt.

Module 10: Implementing a Container Build Strategy

Lessons

- Implementing a Container Build Strategy

Lab : Existing .NET Applications with Azure and Docker Images

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 11: Manage Artifact versioning, security & compliance

Lessons

- Package Security
- Open Source Software
- Integrating License and Vulnerabiltiy Scans
- Implement a Versioning Strategy (Git Version)

Lab : Updating Packages

After completing this module, students will be able to:

- Inspect open source software packages for security and license compliance to align with corporate standards
- Configure build pipeline to access package security and license rating
- Configure secure access to package feeds
- Inspect codebase to identify code dependencies that can be converted to packages
- Identify and recommend standardized package types and versions across the solution
- Refactor existing build pipelines to implement version strategy that publishes packages
- Manage security and compliance

Module 12: Design a Release Strategy

Lessons

- Introduction to Continuous Delivery
- Release strategy recommendations
- Building a High Quality Release pipeline
- Choosing a deployment pattern
- Choosing the Rigth Release Management Tool

Lab : Building a Release Strategy

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

- Explain deployment patterns, both in the traditional sense and in the modern sense
- Choose a release management tool

Module 13: 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 Automation
- Automate Inspection of Health

Lab : Automating your infrastructure deployments in the Cloud with Terraform and Azure PipelinesLab : Setting Up Secrets in the Pipeline with Azure Key VaultLab : Setting Up Secrets in the Pipeline with Azure Key VaultLab : Setting up and Running Functional TestsLab : Using Azure Monitor as a Release GateLab : Creating a Release Dashboard

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
- Classify an Agent, Agent Queue, and Agent Pool
- 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
- Embed testing in the pipeline
- List the different ways to inspect the health of your pipeline and release by using alerts, service hooks, and reports
- Create a release gate

Module 14: Implement 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 : Blue-Green Deployments

After completing this module, students will be able to:

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

Module 15: 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

After completing this module, students will be able to:

- 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

Module 16: Implement a mobile DevOps strategyLessons

- Introduction to Mobile DevOps
- Introduction to Visual Studio App Center
- Manage mobile target device sets and distribution groups
- Manage target UI test device sets
- Provision tester devices for deployment
- Create public and private distribution groups

After completing this module, students will be able to:

- Implement a mobile DevOps strategy

Module 17: 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
- Additional Automation Tools

Lab : Deploy to Azure using ARM Templates

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 ARM templates, PowerShell, and Azure CLI

Module 18: Azure Deployment Models and Services

Lessons

- Deployment Modules and Options
- Azure Infrastructure-as-a-Service (IaaS) Services
- Azure Automation with DevOps
- Desired State Configuration (DSC)
- Azure Platform-as-a-Service (PaaS) services
- Azure Service Fabric

Lab : Azure Automation – IaaS or PaaS deployment

After completing this module, students will be able to:

- Describe deployment models and services that are available with Azure

Module 19: Create and Manage Kubernetes Service Infrastructure

Lessons

- Azure Kubernetes Service

Lab : Deploy and Scale AKS Cluster

After completing this module, students will be able to:

- Deploy and configure a Managed Kubernetes cluster

Module 20: Third Party and Open Source Tools available with Azure

Lessons

- Chef
- Puppet
- Ansible
- Cloud-init
- Terraform

Lab : Provision and configure an App in Azure Using X

After completing this module, students will be able to:

- Deploy and configure infrastructure using 3rd party tools and services with Azure, such as Chef, Puppet, Ansible, SaltStack, and Terraform

Module 21: Implement Compliance and Security in your Infrastructure

Lessons

- Security and Compliance Principles with DevOps
- Azure security Center

Lab : Integrate a scanning extension or tool in an Azure DevOps pipeline/security center

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 22: 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

After completing this module, students will be able to:

- Design practices to measure end-user satisfaction
- Design processes to capture and analyze user feedback from external sources
- Design routing for client application crash report data

- Recommend monitoring tools and technologies
- Recommend system and feature usage tracking tools

Module 23: 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 PostMortems and a Just Culture

After completing this module, students will be able to:

- 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