Microsoft Azure Fundamentals - AZ 900

Duration: 20 Hours

Overview

As a candidate for this certification, you're a technology professional who wants to demonstrate foundational knowledge of cloud concepts in general and Microsoft Azure in particular. This certification is a common starting point in a journey towards a career in Azure.
You can describe Azure architectural components and Azure services, such as:

- Compute
- Networking
- Storage

You can also describe features and tools to secure, govern, and administer Azure.
You should have skills and experience working with an area of IT, such as:

- Infrastructure management
- Database management
- Software development

1) Describe cloud concepts

Describe cloud computing

- Define cloud computing
- Describe the shared responsibility model
- Define cloud models, including public, private, and hybrid
- Identify appropriate use cases for each cloud model
- Describe the consumption-based model
- Compare cloud pricing models

Describe the benefits of using cloud services

- Describe the benefits of high availability and scalability in the cloud
- Describe the benefits of reliability and predictability in the cloud
- Describe the benefits of security and governance in the cloud
- Describe the benefits of manageability in the cloud

Describe cloud service types

- Describe infrastructure as a service (IaaS)
- Describe platform as a service (PaaS)
- Describe software as a service (SaaS)
- Identify appropriate use cases for each cloud service (IaaS, PaaS, SaaS)

2) Describe Azure architecture and services

Describe the core architectural components of Azure

- Describe Azure regions, region pairs, and sovereign regions
- Describe availability zones
- Describe Azure datacenters
- Describe Azure resources and resource groups
- Describe subscriptions
- Describe management groups
- Describe the hierarchy of resource groups, subscriptions, and management groups

Describe Azure compute and networking services

- Compare compute types, including container instances, virtual machines (VMs), and functions
- Describe VM options, including Azure Virtual Machines, Azure Virtual Machine Scale Sets, availability sets, and Azure Virtual Desktop
- Describe resources required for virtual machines
- Describe application hosting options, including the Web Apps feature of Azure App Service, containers, and virtual machines
- Describe virtual networking, including the purpose of Azure Virtual Networks, Azure virtual subnets, peering, Azure DNS, Azure VPN Gateway, and Azure ExpressRoute
- Define public and private endpoints

Describe Azure storage services

- Compare Azure storage services
- Describe storage tiers
- Describe redundancy options
- Describe storage account options and storage types
- Identify options for moving files, including AzCopy, Azure Storage Explorer, and Azure File Sync
- Describe migration options, including Azure Migrate and Azure Data Box

Describe Azure identity, access, and security

- Describe directory services in Azure, including Microsoft Azure Active Directory (Azure AD), part of Microsoft Entra and Azure Active Directory Domain Services (Azure AD DS)
- Describe authentication methods in Azure, including single sign-on (SSO), multifactor authentication, and passwordless
- Describe external identities and guest access in Azure
- Describe Conditional Access in Microsoft Azure Active Directory (Azure AD), part of Microsoft Entra
- Describe Azure role-based access control (RBAC)
- Describe the concept of Zero Trust
- Describe the purpose of the defense in depth model
- Describe the purpose of Microsoft Defender for Cloud

3) Describe Azure management and governance

Describe cost management in Azure

- Describe factors that can affect costs in Azure
- Compare the Pricing calculator and the Total Cost of Ownership (TCO) calculator
- Describe the Azure Cost Management and Billing tool
- Describe the purpose of tags

Describe features and tools in Azure for governance and compliance

- Describe the purpose of Azure Blueprints
- Describe the purpose of Azure Policy
- Describe the purpose of resource locks
- Describe the purpose of the Service Trust Portal

Describe features and tools for managing and deploying Azure resources

- Describe the Azure portal
- Describe Azure Cloud Shell, including Azure CLI and Azure PowerShell
- Describe the purpose of Azure Arc
- Describe Azure Resource Manager and Azure Resource Manager templates (ARM templates)

Describe monitoring tools in Azure

- Describe the purpose of Azure Advisor
- Describe Azure Service Health
- Describe Azure Monitor, including Log Analytics, Azure Monitor alerts, and Application Insights

AZ-104T00: Microsoft Azure Administrator

Duration: 40 Hours

Audience profile

As a candidate for this exam, you should have subject matter expertise in implementing, managing, and monitoring an organization's Microsoft Azure environment, including virtual networks, storage, compute, identity, security, and governance.

As an Azure administrator, you often serve as part of a larger team dedicated to implementing an organization's cloud infrastructure. You also coordinate with other roles to deliver Azure networking, security, database, application development, and DevOps solutions.

You should be familiar with:

- Operating systems
- Networking
- Servers
- Virtualization

In addition, you should have experience with:

- PowerShell
- Azure CLI
- The Azure portal
- Azure Resource Manager templates
- Microsoft Entra ID

Skills at a glance

- Manage Azure identities and governance (20–25%)
- Implement and manage storage (15–20%)
- Deploy and manage Azure compute resources (20–25%)
- Implement and manage virtual networking (15–20%)
- Monitor and maintain Azure resources (10–15%)

Manage Azure identities and governance (20–25%)

Manage Microsoft Entra users and groups

- Create users and groups
- Manage user and group properties

- Manage licenses in Microsoft Entra ID
- Manage external users
- Configure self-service password reset (SSPR)

Manage access to Azure resources

- Manage built-in Azure roles
- Assign roles at different scopes
- Interpret access assignments

Manage Azure subscriptions and governance

- Implement and manage Azure Policy
- Configure resource locks
- Apply and manage tags on resources
- Manage resource groups
- Manage subscriptions
- Manage costs by using alerts, budgets, and Azure Advisor recommendations
- Configure management groups

Implement and manage storage (15–20%)

Configure access to storage

- Configure Azure Storage firewalls and virtual networks
- Create and use shared access signature (SAS) tokens
- Configure stored access policies
- Manage access keys
- Configure identity-based access for Azure Files

Configure and manage storage accounts

- Create and configure storage accounts
- Configure Azure Storage redundancy
- Configure object replication
- Configure storage account encryption.
- Manage data by using Azure Storage Explorer and AzCopy

Configure Azure Files and Azure Blob Storage

- Create and configure a file share in Azure Storage
- Create and configure a container in Blob Storage
- Configure storage tiers
- Configure snapshots and soft delete for Azure Files
- Configure blob lifecycle management
- Configure blob versioning

Deploy and manage Azure compute resources (20–25%)

Automate deployment of resources by using Azure Resource Manager (ARM) templates or Bicep files

- Interpret an Azure Resource Manager template or a Bicep file
- Modify an existing Azure Resource Manager template
- Modify an existing Bicep file
- Deploy resources by using an Azure Resource Manager template or a Bicep file
- Export a deployment as an Azure Resource Manager template or convert an Azure Resource Manager template to a Bicep file

Create and configure virtual machines

- Create a virtual machine
- Configure Azure Disk Encryption
- Move a virtual machine to another resource group, subscription, or region
- Manage virtual machine sizes
- Manage virtual machine disks
- Deploy virtual machines to availability zones and availability sets
- Deploy and configure an Azure Virtual Machine Scale Sets

Provision and manage containers in the Azure portal

- Create and manage an Azure container registry
- Provision a container by using Azure Container Instances
- Provision a container by using Azure Container Apps

- Manage sizing and scaling for containers, including Azure Container Instances and Azure Container Apps

Create and configure Azure App Service

- Provision an App Service plan
- Configure scaling for an App Service plan
- Create an App Service
- Configure certificates and Transport Layer Security (TLS) for an App Service
- Map an existing custom DNS name to an App Service
- Configure backup for an App Service
- Configure networking settings for an App Service
- Configure deployment slots for an App Service

Implement and manage virtual networking (15–20%)

Configure and manage virtual networks in Azure

- Create and configure virtual networks and subnets
- Create and configure virtual network peering
- Configure public IP addresses
- Configure user-defined network routes
- Troubleshoot network connectivity

Configure secure access to virtual networks

- Create and configure network security groups (NSGs) and application security groups
- Evaluate effective security rules in NSGs
- Implement Azure Bastion
- Configure service endpoints for Azure platform as a service (PaaS)
- Configure private endpoints for Azure PaaS

Configure name resolution and load balancing

- Configure Azure DNS
- Configure an internal or public load balancer
- Troubleshoot load balancing

Monitor and maintain Azure resources (10–15%)

Monitor resources in Azure

- Interpret metrics in Azure Monitor

- Configure log settings in Azure Monitor
- Query and analyze logs in Azure Monitor
- Set up alert rules, action groups, and alert processing rules in Azure Monitor
- Configure and interpret monitoring of virtual machines, storage accounts, and networks by using Azure Monitor Insights
- Use Azure Network Watcher and Connection Monitor

Implement backup and recovery

- Create a Recovery Services vault
- Create an Azure Backup vault
- Create and configure a backup policy
- Perform backup and restore operations by using Azure Backup
- Configure Azure Site Recovery for Azure resources
- Perform a failover to a secondary region by using Site Recovery
- Configure and interpret reports and alerts for backups

AZ-400T00: Designing and Implementing Microsoft DevOps solutions

Duration: 32 Hours (4 Days)

Overview

The AZ-400T00-A: Designing and Implementing Microsoft DevOps solutions course is a comprehensive learning path for IT professionals who aim to enhance their expertise in DevOps practices using Microsoft technologies. This course focuses on teaching participants how to combine people, processes, and technologies to continuously deliver valuable products and services that meet end-user needs and business objectives. Learners will explore various aspects of DevOps such as Source control, continuous integration (CI), continuous delivery (CD), Dependency management, Application infrastructure, and Continuous feedback. The course includes practical hands-on labs that allow participants to apply the learned concepts in real-world scenarios. By the end of the course, participants will be well-equipped to take the Microsoft Certified: DevOps Engineer Expert exam. They will gain skills in DevOps strategies, Azure Repos, Azure Pipelines, Build infrastructure management, and Implementing a secure CD pipeline. The course also covers advanced topics like managing Infrastructure as code using Azure and Desired State Configuration (DSC), as well as designing and implementing a Dependency management strategy. This training is crucial for IT professionals looking to advance their careers in the DevOps domain and for organizations aiming to implement DevOps practices to improve

their deployment frequency and product quality while maintaining a secure and compliant environment.

Audience Profile

The AZ-400T00-A course is designed for professionals seeking expertise in Microsoft DevOps solutions to streamline development and operations.

- DevOps Engineers
- Software Developers
- IT Professionals with a focus on CI/CD and automation
- System Administrators transitioning to DevOps roles
- Release Managers
- Cloud Solutions Architects
- Technical Project Managers
- IT Managers looking to implement DevOps practices
- Quality Assurance Engineers
- Security Professionals involved in development and operations
- Operations Support Staff
- Professionals working with Git source control systems
- Infrastructure Engineers
- Professionals interested in containerization and orchestration with Kubernetes
- Technical Leads overseeing cross-functional DevOps teams
- Application Developers building cloud-native applications on Azure
- Professionals interested in implementing Agile planning with Azure Boards
- Engineers focusing on building secure and compliant development processes

Skills at a glance

- Design and implement processes and communications (10–15%)
- Design and implement a source control strategy (10–15%)
- Design and implement build and release pipelines (50–55%)
- Develop a security and compliance plan (10–15%)
- Implement an instrumentation strategy (5–10%)
- Design and implement processes and communications (10–15%)

Design and implement traceability and flow of work

- Design and implement a structure for the flow of work, including GitHub

- Flow
- Design and implement a strategy for feedback cycles, including
- notifications and GitHub issues
- Design and implement integration for tracking work, including GitHub
- projects, Azure Boards, and repositories
- Design and implement source, bug, and quality traceability

Design and implement appropriate metrics and queries for DevOps

- Design and implement a dashboard, including flow of work, such as cycle
- times, time to recovery, and lead time
- Design and implement appropriate metrics and queries for project
- planning
- Design and implement appropriate metrics and queries for development
- Design and implement appropriate metrics and queries for testing
- Design and implement appropriate metrics and queries for security
- Design and implement appropriate metrics and queries for delivery
- Design and implement appropriate metrics and queries for operations

Configure collaboration and communication

- Document a project by configuring wikis and process diagrams,
- including Markdown and Mermaid syntax
- Configure release documentation, including release notes and API
- documentation
- Automate creation of documentation from Git history
- Configure integration by using webhooks
- Configure integration between Azure Boards and GitHub repositories
- Configure integration between GitHub or Azure DevOps and Microsoft
- Teams
- Design and implement a source control strategy (10–15%)

Design and implement branching strategies for the source code

- Design a branch strategy, including trunk-based, feature branch, and
- release branch
- Design and implement a pull request workflow by using branch policies
- and branch protections
- Implement branch merging restrictions by using branch policies and
- branch protections

Configure and manage repositories

- Design and implement a strategy for managing large files, including Git
- Large File Storage (LFS) and git-fat
- Design a strategy for scaling and optimizing a Git repository, including
- Scalar and cross-repository sharing
- Configure permissions in the source control repository
- Configure tags to organize the source control repository
- Recover specific data by using Git commands
- Remove specific data from source control
- Design and implement build and release pipelines (50–55%)

Design and implement a package management strategy

- Recommend package management tools including GitHub Packages
- registry and Azure Artifacts
- Design and implement package feeds and views for local and upstream
- packages
- Design and implement a dependency versioning strategy for code assets
- and packages, including semantic versioning (SemVer) and date-based
- (CalVer)
- Design and implement a versioning strategy for pipeline artifacts

Design and implement a testing strategy for pipelines

- Design and implement quality and release gates, including security and
- governance
- Design a comprehensive testing strategy, including local tests, unit tests,
- integration tests, and load tests
- Implement tests in a pipeline, including configuring test tasks,
- configuring test agents, and integration of test results
- Implement code coverage analysis

Design and implement pipelines

- Select a deployment automation solution, including GitHub Actions and
- Azure Pipelines
- Design and implement a GitHub runner or Azure DevOps agent
- infrastructure, including cost, tool selection, licenses, connectivity, and
- maintainability

- Design and implement integration between GitHub repositories and
- Azure Pipelines
- Develop and implement pipeline trigger rules
- Develop pipelines by using YAML
- Design and implement a strategy for job execution order, including
- parallelism and multi-stage pipelines
- Develop and implement complex pipeline scenarios, such as hybrid
- pipelines, VM templates, and self-hosted runners or agents
- Create reusable pipeline elements, including YAML templates, task
- groups, variables, and variable groups
- Design and implement checks and approvals by using YAML-based Environments

Design and implement deployments

- Design a deployment strategy, including blue-green, canary, ring,
- progressive exposure, feature flags, and A/B testing
- Design a pipeline to ensure that dependency deployments are reliably
- ordered
- Plan for minimizing downtime during deployments by using virtual IP
- address (VIP) swap, load balancing, rolling deployments, and
- deployment slot usage and swap
- Design a hotfix path plan for responding to high-priority code fixes
- Design and implement a resiliency strategy for deployment
- Implement feature flags by using Azure App Configuration Feature
- Manager
- Implement application deployment by using containers, binaries, and
- scripts
- Implement a deployment that includes database tasks

Design and implement infrastructure as code (IaC)

- Recommend a configuration management technology for application
- infrastructure
- Implement a configuration management strategy for application
- infrastructure
- Define an IaC strategy, including source control and automation of
- testing and deployment
- Design and implement desired state configuration for environments,
- including Azure Automation State Configuration, Azure Resource
- Manager, Bicep, and Azure Automanage Machine Configuration
- Design and implement Azure Deployment Environments for on-demand
- self-deployment

Maintain pipelines

- Monitor pipeline health, including failure rate, duration, and flaky tests
- Optimize a pipeline for cost, time, performance, and reliability
- Optimize pipeline concurrency for performance and cost
- Design and implement a retention strategy for pipeline artifacts and
- dependencies
- Migrate a pipeline from classic to YAML in Azure Pipelines
- Develop a security and compliance plan (10–15%)

Design and implement authentication and authorization methods

- Choose between Service Principals and Managed Identity (including system-assigned and user-assigned)
- Implement and manage GitHub authentication, including GitHub Apps,
- GITHUB_TOKEN, and personal access tokens
- Implement and manage Azure DevOps service connections and personal
- access tokens
- Design and implement permissions and roles in GitHub
- Design and implement permissions and security groups in Azure DevOps
- Recommend appropriate access levels, including stakeholder access in
- Azure DevOps and outside collaborator access in GitHub
- Configure projects and teams in Azure DevOps

Design and implement a strategy for managing sensitive information in automation

- Implement and manage secrets, keys, and certificates by using Azure Key
- Vault
- Implement and manage secrets in GitHub Actions and Azure Pipelines
- Design and implement a strategy for managing sensitive files during
- deployment, including Azure Pipelines secure files
- Design pipelines to prevent leakage of sensitive information

Automate security and compliance scanning

- Design a strategy for security and compliance scanning, including
- dependency, code, secret, and licensing scanning
- Configure Microsoft Defender for Cloud DevOps Security
- Configure GitHub Advanced Security for both GitHub and Azure DevOps
- Integrate GitHub Advanced Security with Microsoft Defender for Cloud
- Automate container scanning, including scanning container images and
- configuring an action to run CodeQL analysis in a container

- Automate analysis of licensing, vulnerabilities, and versioning of opensource components by using Dependabot alerts
- Implement an instrumentation strategy (5–10%)

Configure monitoring for a DevOps environment

- Configure Azure Monitor and Log Analytics to integrate with DevOps
- tools
- Configure collection of telemetry by using Application Insights, VM
- Insights, Container Insights, Storage Insights, and Network Insights
- Configure monitoring in GitHub, including enabling insights and creating
- and configuring charts
- Configure alerts for events in GitHub Actions and Azure Pipelines.

Analyze metrics from instrumentation

- Inspect infrastructure performance indicators, including CPU, memory,
- disk, and network
- Analyze metrics by using collected telemetry, including usage and
- application performance
- Inspect distributed tracing by using Application Insights
- Interrogate logs using basic Kusto Query Language (KQL) queries