

Exam AZ-203: Developing Solutions for Microsoft Azure – Skills Measured

A NEW VERSION OF THIS EXAM, AZ-204, WILL BE AVAILABLE IN FEBRUARY. You will still be able to take this exam for ~90 days after the release of AZ-204. Specific dates related to the availability of AZ-204 and the retirement of AZ-203 will be posted to the website when they are known. The objective domain for AZ-204 is appended below.

Audience Profile

Candidates for this exam are Azure Developers who design and build cloud solutions such as applications and services. They participate in all phases of development, from solution design, to development and deployment, to testing and maintenance. They partner with cloud solution architects, cloud DBAs, cloud administrators, and clients to implement the solution.

Candidates should be proficient in developing apps and services by using Azure tools and technologies, including storage, security, compute, and communications.

Candidates must have at least one year of experience developing scalable solutions through all phases of software development and be skilled in at least one cloud-supported programming language.

Skills Measured

Develop Azure Infrastructure as a Service Compute Solution (10-15%)

Implement solutions that use virtual machines (VM)

- provision VMs
- create ARM templates
- configure Azure Disk Encryption for VMs

Implement batch jobs by using Azure Batch Services

- manage batch jobs by using Batch Service API
- run a batch job by using Azure CLI, Azure portal, and other tools
- write code to run an Azure Batch Services batch job

Create containerized solutions

- create an Azure Managed Kubernetes Service (AKS) cluster
- create container images for solutions
- publish an image to the Azure Container Registry

- run containers by using Azure Container Instance or AKS

Develop Azure Platform as a Service Compute Solution (20-25%)

Create Azure App Service Web Apps

- create an Azure App Service Web App
- create an Azure App Service background task by using WebJobs
- enable diagnostics logging
- create an Azure Web App for Containers
- monitor service health by using Azure Monitor

Create Azure App Service mobile apps

- add push notifications for mobile apps
- enable offline sync for mobile app
- implement a remote instrumentation strategy for mobile devices

Create Azure App Service API apps

- create an Azure App Service API app
- create documentation for the API by using open source and other tools

Implement Azure functions

- implement input and output bindings for a function
- implement function triggers by using data operations, timers, and webhooks
- implement Azure Durable Functions
- create Azure Function apps by using Visual Studio
- implement Python Azure Functions

Develop for Azure storage (15-20%)

Develop solutions that use storage tables

- design and implement policies for tables
- query table storage by using code
- implement partitioning schemes

Develop solutions that use Cosmos DB storage

- create, read, update, and delete data by using appropriate APIs
- implement partitioning schemes
- set the appropriate consistency level for operations

Develop solutions that use a relational database

- provision and configure relational databases
- configure elastic pools for Azure SQL Database
- create, read, update, and delete data tables by using code
- provision and configure Azure SQL Database serverless instances
- provision and configure Azure SQL and Azure PostgreSQL Hyperscale instances

Develop solutions that use blob storage

- move items in Blob storage between storage accounts or containers
- set and retrieve properties and metadata
- implement blob leasing
- implement data archiving and retention
- implement Geo Zone Redundant Storage

Implement Azure security (10-15%)

Implement authentication

- implement authentication by using certificates, forms-based authentication, or tokens
- implement multi-factor or Windows authentication by using Azure AD
- implement OAuth2 authentication
- implement Managed identities/Service Principal authentication
- implement Microsoft identity platform

Implement access control

- implement CBAC (Claims-Based Access Control) authorization
- implement RBAC (Role-Based Access Control) authorization
- create shared access signatures

Implement secure data solutions

- encrypt and decrypt data at rest and in transit
- create, read, update, and delete keys, secrets, and certificates by using the KeyVault API

Monitor, troubleshoot, and optimize Azure solutions (10-15%)

Develop code to support scalability of apps and services

- implement autoscaling rules and patterns (schedule, operational/system metrics, singleton applications)
- implement code that handles transient faults

- implement AKS scaling strategies

Integrate caching and content delivery within solutions

- store and retrieve data in Azure Redis cache
- develop code to implement CDN's in solutions
- invalidate cache content (CDN or Redis)

Instrument solutions to support monitoring and logging

- configure instrumentation in an app or service by using Application Insights
- analyze and troubleshoot solutions by using Azure Monitor
- implement Application Insights Web Test and Alerts

Connect to and consume Azure services and third-party services (20-25%)

Develop an App Service Logic App

- create a Logic App
- create a custom connector for Logic Apps
- create a custom template for Logic Apps

Integrate Azure Search within solutions

- create an Azure Search index
- import searchable data
- query the Azure Search index
- implement cognitive search

Implement API management

- establish API Gateways
- create an APIM instance
- configure authentication for APIs
- define policies for APIs

Develop event-based solutions

- implement solutions that use Azure Event Grid
- implement solutions that use Azure Notification Hubs
- implement solutions that use Azure Event Hub

Develop message-based solutions

- implement solutions that use Azure Service Bus
- implement solutions that use Azure Queue Storage queues

Exam AZ-204: Developing Solutions for Microsoft Azure

Audience Profile

Candidates for this exam are cloud developers who participate in all phases of development from requirements definition and design, to development and deployment, and maintenance. They partner with cloud DBAs, cloud administrators, and clients to implement solutions.

Candidates should be proficient in Azure SDKs, data storage options, data connections, APIs, app authentication and authorization, compute and container deployment, debugging, performance tuning, and monitoring.

Candidates must have 1-2 years professional development experience and experience with Microsoft Azure. They must be able to program in an Azure Supported Language.

Skills Measured

Develop Azure compute solutions (25-30%)

Implement IaaS solutions

- provision VMs
- create ARM templates
- create container images for solutions
- publish an image to the Azure Container Registry
- run containers by using Azure Container Instance

Create Azure App Service Web Apps

- create an Azure App Service Web App
- enable diagnostics logging
- deploy code to a web app
- configure web app settings
- implement autoscaling rules (schedule, operational/system metrics)

Implement Azure functions

- implement input and output bindings for a function
- implement function triggers by using data operations, timers, and webhooks
- implement Azure Durable Functions

Develop for Azure storage (10-15%)

Develop solutions that use Cosmos DB storage

- select the appropriate API for your solution
- implement partitioning schemes
- interact with data using the appropriate SDK
- set the appropriate consistency level for operations
- create Cosmos DB containers

Develop solutions that use blob storage

- move items in Blob storage between storage accounts or containers
- set and retrieve properties and metadata
- interact with data using the appropriate SDK
- implement data archiving and retention

Implement Azure security (15-20%)

Implement user authentication and authorization

- implement OAuth2 authentication
- create and implement shared access signatures
- register apps and use Azure Active Directory to authenticate users

Implement secure cloud solutions

- secure app configuration data by using the App Configuration and KeyVault API
- manage keys, secrets, and certificates by using the KeyVault API
- implement Managed Identities for Azure resources

Monitor, troubleshoot, and optimize Azure solutions (10-15%)

Integrate caching and content delivery within solutions

- develop code to implement CDN's in solutions
- configure cache and expiration policies
- store and retrieve data in Azure Redis cache

Instrument solutions to support monitoring and logging

- configure instrumentation in an app or service by using Application Insights
- analyze and troubleshoot solutions by using Azure Monitor
- implement Application Insights Web Test and Alerts

- implement code that handles transient faults

Connect to and consume Azure services and third-party services (25-30%)

Develop an App Service Logic App

- create a Logic App
- create a custom connector for Logic Apps
- create a custom template for Logic Apps

Implement API Management

- create an APIM instance
- configure authentication for APIs
- define policies for APIs

Develop event-based solutions

Note: Creating event models is in scope

- implement solutions that use Azure Event Grid
- implement solutions that use Azure Notification Hubs
- implement solutions that use Azure Event Hub

Develop message-based solutions

- implement solutions that use Azure Service Bus
- implement solutions that use Azure Queue Storage queues