Exam AZ-204: Developing Solutions for Microsoft Azure – Skills Measured

This exam was updated on May 18, 2020. Following the current exam guide, we have included a version of the exam guide with Track Changes set to "On," showing the changes that were made to the exam on that date.

Audience Profile

Candidates for this exam should have subject matter expertise designing, building, testing, and maintaining cloud applications and services on Microsoft Azure.

Responsibilities for an Azure Developer include participating in all phases of cloud development from requirements definition and design, to development, deployment, and maintenance. performance tuning, and monitoring.

Azure Developers partner with cloud solution architects, cloud DBAs, cloud administrators, and clients to implement solutions.

A candidate for this exam should have 1-2 years professional development experience and experience with Microsoft Azure. In addition, the role should have ability programming in a language supported by Azure and proficiency in Azure SDKs, Azure PowerShell, Azure CLI, data storage options, data connections, APIs, app authentication and authorization, compute and container deployment, debugging, performance tuning, and monitoring.

Skills Measured

NOTE: The bullets that appear below each of the skills measured are intended to illustrate how we are assessing that skill. This list is not definitive or exhaustive.

NOTE: In most cases, exams do NOT cover preview features, and some features will only be added to an exam when they are GA (General Availability).

Develop Azure compute solutions (25-30%)

Implement IaaS solutions

- provision VMs
- configure VMs for remote access
- create ARM templates
- create container images for solutions by using Docker
- publish an image to the Azure Container Registry

- run containers by using Azure Container Instance
- Azure Kubernetes Service (AKS) is out of scope

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 including SSL, API, and connection strings
- implement autoscaling rules, including scheduled autoscaling, and scaling by operational or 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
- implement scaling (partitions, containers)
- implement server-side programming including stored procedures, triggers, and change feed notifications

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 hot, cool, and archive storage

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
- control access to resources by using role-based access controls (RBAC)

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 CDNs in solutions
- configure cache and expiration policies for FrontDoor, CDNs, or Redis caches 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 log data 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

- 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

The exam guide below shows the changes that were implemented on May 18, 2020.

Audience Profile

Candidates for this exam should have subject matter expertise designing, building, testing, and maintaining cloud applications and services on Microsoft Azure.

Responsibilities for an Azure Developer include participating in all phases of cloud development from requirements definition and design, to development, deployment, and maintenance. performance tuning, and monitoring.

Azure Developers partner with cloud solution architects, cloud DBAs, cloud administrators, and clients to implement solutions.

A candidate for this exam should have 1-2 years professional development experience and experience with Microsoft Azure. In addition, the role should have ability programming in a language supported by Azure and proficiency in Azure SDKs, Azure PowerShell, Azure CLI, data storage options, data connections, APIs, app authentication and authorization, compute and container deployment, debugging, performance tuning, and monitoring.

Skills Measured

NOTE: The bullets that appear below each of the skills measured are intended to illustrate how we are assessing that skill. This list is not definitive or exhaustive.

NOTE: In most cases, exams do NOT cover preview features, and some features will only be added to an exam when they are GA (General Availability).

Develop Azure compute solutions (~~20-25~~25-30%)

Implement IaaS solutions

- provision VMs
- configure VMs for remote access
- create ARM templates
- create container images for solutions by using Docker

- publish an image to the Azure Container Registry
- run containers by using Azure Container Instance
- Azure Kubernetes Service (AKS) is out of scope

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 including SSL, API, and connection strings
- implement autoscaling rules, including scheduled autoscaling, and scaling by operational or 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 (~~15-20~~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
- implement scaling (partitions, containers)
- implement server-side programming including stored procedures, triggers, and change feed notifications

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 hot, cool, and archive storage

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
- control access to resources by using role-based access controls (RBAC)

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 for FrontDoor, CDNs, or Redis caches
- 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 log data 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~~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

Implement API Management

- 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