

70-532: Developing Microsoft Azure Solutions

Exam Design

Target Audience

Candidates of this exam are experienced in designing, programming, implementing, automating, and monitoring Microsoft Azure solutions. Candidates should be proficient with development tools, techniques, and approaches used to build scalable and resilient solutions.

Objective Domain

Note: This document shows tracked changes that are effective as of March 22, 2018.

Create and Manage Azure Resource Manager Virtual Machines (20-25%)

Deploy workloads on Azure Resource Manager (ARM) virtual machines (VMs)

Identify workloads that can and cannot be deployed; run workloads including Microsoft and Linux; create and provision VMs including custom VM images; deploy workloads using [Ansible and Terraform](#); ~~reserve VMs by using~~[leverage](#) Azure Reserved Instances (RIs)

Perform configuration management

Automate configuration management by using PowerShell Desired State Configuration (DSC) or VM Agent (custom script extensions); enable remote debugging; implement VM template variables to configure VMs

Scale ARM VMs

Scale up and scale down VM sizes; [implement Accelerated Networking](#); deploy ARM VM Scale Sets (VMSS); configure ARM VMSS auto-scale

Design and implement ARM VM storage

Configure disk caching; plan for storage capacity; configure shared storage; configure geo-replication; implement ARM VMs with Standard and Premium Storage; implement Azure Disk Encryption for Windows and Linux ARM VMs; implement Azure Disk Storage; implement StorSimple

Monitor ARM VMs

Configure ARM VM monitoring; configure alerts; configure diagnostic and monitoring storage location; enable Application Insights at runtime; Monitor VM workloads by using Azure Application Insights; monitor VMs using Azure [OMS](#) Log Analytics ~~and OMS~~; monitor Linux and Windows VMs by using the Azure Diagnostics Extension; monitor VMs by using Azure Monitor

Manage ARM VM availability

Configure multiple ARM VMs in an availability set for redundancy; configure each application tier into separate availability sets; combine the Load Balancer with availability sets; perform automated VM maintenance; [use availability zones to protect from datacenter failures](#)

Design and Implement DevTest Labs

Create and manage custom images and formulas; configure a lab to include policies and procedures; configure cost management; secure access to labs; use environments in a lab; [claim and un-claim VMs; use artifacts to deploy and set up applications](#)

Design and Implement a Storage and Data Strategy (25-30%)

Implement Azure Storage blobs and Azure Files

Read data; change data; set metadata on a storage container; store data using block and page blobs; stream data using blobs; access blobs securely; implement async blob copy; configure Content Delivery Network (CDN); design blob hierarchies; configure custom domains; scale blob storage and implement blob tiering; create connections to files from on-premises or cloud-based Windows or Linux machines; shard large datasets; implement blob leasing; implement Storage Events; implement Azure File Sync; [implement virtual network service endpoints; implement Azure Files backup](#)

Implement Azure storage tables, queues, and Azure Cosmos DB Table API

Implement CRUD with and without transactions; design and manage partitions; query using OData; scale tables and partitions; add and process queue messages; retrieve a

batch of messages; scale queues; choose between Azure Storage Tables and Azure Cosmos DB Table API

Manage access and monitor storage

Generate shared access signatures, including client renewal and data validation; create stored access policies; regenerate storage account keys; configure and use Cross-Origin Resource Sharing (CORS); set retention policies and logging levels; analyze logs; monitor Cosmos DB storage

Implement Azure SQL Database

Choose the appropriate database tier and performance level; configure and perform point in time recovery; enable geo-replication; import and export data and schema; scale Azure SQL [databases](#)[Database instances](#); manage elastic pools, including DTUs and eDTUs; manage limits and resource governor; implement Azure SQL Data Sync; implement graph database functionality in Azure SQL [Database](#); design multi-tenant applications; secure and encrypt data; manage data integrity; enable metrics and diagnostics logs for monitoring; use adaptive query processing to improve query performance; implement sharding and elastic tools; implement SQL Server Stretch Database

Implement Azure Cosmos DB

Choose a Cosmos DB API surface; create Cosmo DB API databases and collections; query documents; run Cosmos DB queries; create Graph API databases; execute GraphDB queries; implement MongoDB database; manage scaling of Cosmos DB, including managing partitioning, consistency, and RU/m; manage multiple regions; implement stored procedures; implement JavaScript within Cosmos DB; access Cosmos DB from REST interface; manage Cosmos DB security

Implement Redis caching

Choose a cache tier; implement data persistence; implement security and network isolation; tune cluster performance; integrate Redis caching with ASP.NET session and cache providers; implement Redis data types and operations; [implement geo-replication for availability and disaster recovery scenarios](#)

Implement Azure Search

Create a service index; add data; search an index; handle search results; [implement synonyms](#)

Manage Identity, Application, and Network Services (10-15%)

Integrate an app with Azure Active Directory (AAD)

Develop apps that use WS-federation, OAuth, and SAML-P endpoints; query the directory by using Microsoft Graph API, MFA and MFA API; [implement Azure Managed Service Identity](#)

Design and implement a messaging strategy

Develop and scale messaging solutions using service bus queues, topics, relays, event hubs, Event Grid, and notification hubs; monitor service bus queues, topics, relays, event hubs and notification hubs; determine when to use Event Hubs, Service Bus, IoT Hub, Stream Analytics, and Notification Hubs; implement Azure Event Grid

Develop apps that use AAD B2C and AAD B2B

Design and implement .NET MVC, Web API, and Windows Desktop apps that leverage social identity provider authentication, including Microsoft account, Facebook, Google+, Amazon, and LinkedIn; leverage Azure AD B2B to design and implement applications that support partner-managed identities, enforce multi-factor authentication

Manage secrets using Azure Key Vault

Configure Azure Key Vault; manage access, including tenants; implement HSM protected keys; manage service limits; implement logging; implement key rotation; store and retrieve app secrets including connection strings, passwords, and cryptographic keys; [implement Azure Managed Service Identity](#)

Design and Implement Azure Compute, Web, and Mobile Services (35-40%)

Design Azure App Service Web Apps

Define and manage App Service plans; configure Web Apps settings, certificates, and custom domains; manage Web Apps by using the API, Azure PowerShell, Azure Cloud Shell, and Xplat-CLI; implement diagnostics, monitoring, and analytics; design and configure Web Apps for scale and resilience; use Azure Managed Service Identity to access other Azure AD-protected resources including Azure Key Vault; [identify usage scenarios for App Service Isolated](#)

Implement Azure Functions and WebJobs

Create Azure Functions; implement a webhook Function; create an event processing Function; implement an Azure-connected Function; design and implement a custom binding; debug a Function; integrate a Function with storage; implement and configure proxies; integrate with App Service plan; [implement build serverless applications, including Azure Event Grid-based serverless applications](#)

Implement API Management

Create managed APIs; configure API Management policies; protect APIs with rate limits; add caching to improve performance; monitor APIs; customize the Developer portal; add authentication and authorization to applications by using API Management; configure API versions by using API Management; implement git-based configuration using API Management

Design Azure App Service API Apps

Create and deploy API Apps; automate API discovery by using Swagger and Swashbuckle; use Swagger API metadata to generate client code for an API app; monitor API Apps

Develop Azure Logic Apps

Create a Logic App connecting SaaS services; create a Logic App with B2B capabilities; create a Logic App with XML capabilities; trigger a Logic App from another app; create custom and long-running actions; monitor Logic Apps; integrate a logic app with a function; access on-premises data; implement Logic Apps with Event Grid

Develop Azure App Service Mobile Apps

Create a Mobile App; add offline sync to a Mobile App; add authentication to a Mobile App; add push notifications to a Mobile App; [distribute a Mobile App; add analytics to a Mobile App; collect crash data from devices; implement location-aware applications by using Azure location-based services](#)

Design and implement Azure Service Fabric Applications

Create a Service Fabric application; build an Actors-based service; add a web front-end to a Service Fabric application; monitor and diagnose services; migrate apps from cloud services; create, secure, upgrade, and scale Service Fabric Cluster in Azure; scale a Service Fabric app; deploy an application to a Container

Design and implement Third Party Platform as a Service (PaaS)

Design and implement Third Party Platform as a Service (PaaS)": Implement Cloud Foundry; implement OpenShift; provision applications by using Azure Quickstart Templates; build applications that leverage Azure Marketplace solutions and services; implement solutions that use Azure Bot Service; create Azure Managed Applications; implement Docker Swarm applications; implement Kubernetes applications

Design and Implement DevOps

Instrument an application with telemetry; discover application performance issues by using Application Insights; deploy Visual Studio Team Services with Continuous integration (CI) and Continuous development (CD); deploy CI/CD with third party platform tools (Jenkins, GitHub, Chef, Puppet; TeamCity); implement mobile DevOps by using HockeyApp; perform root cause analysis using Azure Time Series Insights

[Design and implement Kubernetes applications on Azure Container Service \(AKS\)](#)

[Create container images; manage container images using Azure Container Registry or Docker Hub; build YAML application definition; scale applications; update running applications; monitor using Container Monitoring solution in Log Analytics; build development cluster; manage configuration values](#)