Microsoft Azure Solution Architect

Description

Candidates for the Azure Solutions Architect Expert certification should have subject matter expertise in designing and implementing solutions that run on Microsoft Azure, including aspects like compute, network, storage, and security.

Responsibilities for this role include advising stakeholders and translating business requirements into secure, scalable, and reliable cloud solutions.

An Azure Solutions Architect partners with cloud administrators, cloud DBAs, and clients to implement solutions.

A candidate for this certification should have advanced experience and knowledge across various aspects of IT operations, including networking, virtualization, identity, security, business continuity, disaster recovery, data management, budgeting, and governance–this role should manage how decisions in each area affect an overall solution. In addition, this role should be proficient in at least one of these Azure knowledge domains: administration, development, or DevOps.

Skills measured

- Deploy and configure infrastructure
- Implement workloads and security
- Create and deploy apps
- Implement authentication and secure data
- Develop for the cloud and for Azure storage
- Determine workload requirements
- Design for identity and security
- Design a data platform solution
- Design a business continuity strategy
- Design for deployment, migration, and integration
- Design an infrastructure strategy

PREREQUISITES

- Microsoft Certified Azure Administrator Associate
- Microsoft Certified Azure Developer Associate

Course Content

Deploy and configure infrastructure

Analyze resource utilization and consumption

- configure diagnosticsettings on resources
- create baseline for resources
- create and test alerts
- analyze alerts across subscription
- analyze metrics across subscription
- create action groups
- monitor for unused resources
- monitor spend
- report on spend
- utilize Log Search queryfunctions
- view alerts in Azure Monitorlogs
- visualize diagnostics data using Azure Monitor Workbooks

Create and configure storage accounts

- configure network access to the storage account
- create and configure storageaccount
- generate shared access signature
- implement Azure AD authentication for storage
- install and use Azure Storage Explorer
- manage access keys
- monitor activity log by using Azure Monitor logs
- implement Azure storage replication
- implement Azure storage accountfailover

Create and configure a VM for Windows and Linux

- configure high availability
- configure monitoring
- configure networking
- configure storage
- configure virtual machine size
- implement dedicated hosts
- deploy and configure scale sets

Automate deployment of VMs

- modify Azure Resource Manager template
- configure location of new VMs
- configure VHD template
- deploy from template
- save a deployment as an Azure Resource Manager template
- deploy Windows and Linux VMs

Create connectivity between virtual networks

- create and configure Vnet peering
- create and configure Vnet to Vnet connections
- verify virtual network connectivity
- create virtual network gateway

Implement and manage virtual networking

- configure private IP addressing
- configure public IP addresses
- create and configure network routes
- create and configure network interface
- create and configure subnets
- create and configure virtual network
- create and configure Network Security Groups and Application Security Groups

Manage Azure Active Directory

- add custom domains
- configure Azure AD Identity Protection
- configure Azure AD Join
- configure self-service password reset
- implement conditional access policies
- manage multiple directories
- perform an access review

Implement and manage hybrid identities

- install and configure Azure AD Connect
- configure federation
- configure single sign-on
- manage and troubleshoot Azure AD Connect
- troubleshoot password sync and writeback

Implement solutions that use virtual machines (VM)

- provision VMs

- create Azure Resource Manager templates
- configure Azure Disk Encryption for VMs
- implement Azure Backup for VMs

Implement workloads and security

Migrate servers to Azure

- migrate servers using Azure Migrate

Configure serverless computing

- create and manage objects
- manage a Logic App resource
- manage Azure Function app settings
- manage Event Grid
- manage Service Bus

Implement application load balancing

- configure application gateway
- configure Azure Front Door service
- configure Azure Traffic Manager

Integrate on premises network with Azure virtual network

- create and configure Azure VPN Gateway
- create and configure site to site VPN
- configure ExpressRoute
- configure Virtual WAN
- verify on premises connectivity
- troubleshoot on premises connectivity with Azure

Implement multi factor authentication

- configure user accounts for MFA
- configure fraud alerts
- configure bypass options
- configure trusted IPs
- configure verification methods

Manage role-based access control

- create a custom role
- configure access to Azure resources by assigning roles

- configure management access to Azure
- troubleshoot RBAC
- implement Azure Policies
- assign RBAC Roles

Create and deploy apps

Create web apps by using PaaS

- create an Azure app service Web App
- create documentation for the API
- create an App Service Web App for Containers
- create an App Service background task by using WebJobs
- enable diagnostics logging

Design and develop apps that run in containers

- configure diagnosticsettings on resources
- create a container image by using a Dockerfile
- create an Azure Kubernetes Service
- publish an image to the Azure Container Registry
- implement an application that runs on an Azure Container Instance
- manage container settings by using code

Implement authentication and secure data

Implement authentication

- implement authentication by using certificates, forms-based authentication, tokens, or Windows-integrated authentication
- implement multi-factor authentication by using Azure AD
- implement OAuth2 authentication
- implement Managed Identities for Azure resources Service Principal authentication

Implement secure data solutions

- encrypt and decrypt data at rest and in transit
- encrypt data with Always Encrypted
- implement Azure Confidential Compute
- implement SSL/TLS communications
- create, read, update, and delete keys, secrets, and certificates by using the KeyVault API

Develop for the cloud and for Azure storage

Configure a message-based integration architecture

- configure an app or service to send emails
- configure Event Grid
- configure the Azure Relay service
- create and configure a Notification Hub
- create and configure an Event Hub
- create and configure a Service Bus
- configure queries across multiple products

Develop for autoscaling

- implement autoscaling rules and patterns (schedule, operational/system metrics)
- implement code that addresses singleton application instances
- implement code that addresses transient state

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
- implement Azure SQL Database managed instances
- create, read, update, and delete data tables by using code

Determine workload requirements

Gather information and requirements

- identify compliance requirements
- identify identity and access management infrastructure
- identify service-oriented architectures
- identify accessibility requirements
- identify availability requirements
- identify capacity planning and scalability requirements
- identify deployability requirements
- identify configurability
- identify governance requirements
- identify maintainability requirements

- identify security requirements
- identify sizing requirements
- recommend changes during project execution
- evaluate products and services to align with solution
- create testing scenarios

Optimize consumption strategy

- optimize app service costs
- optimize compute costs
- optimize identity costs
- optimize network costs
- optimize storage costs

Design an auditing and monitoring strategy

- define logical groupings for resources to be monitored
- determine levels and storage locations for logs
- plan for integration with monitoring tools
- recommend appropriate monitoring tools for a solution
- specify mechanism for event routing and escalation
- design auditing for compliancerequirements
- design auditing policies and traceability requirements

Design for identity and security

Design identity management

- choose an identity management approach
- design an identity delegation strategy
- design an identity repository
- design self-service identity management
- design user and persona provisioning
- define personas
- define roles
- recommend appropriate access control strategy

Design authentication

- choose an authentication approach
- design a single sign on approach
- design for IPSec authentication
- design for logon authentication

- design for multi-factor authentication
- design for network access authentication
- design for remote authentication

Design authorization

- choose an authorization approach
- define access permissions and privileges
- design secure delegated access
- recommend when and how to use API Keys

Design for risk prevention for identity

- design a risk assessment strategy
- evaluate agreements involving services or products from vendors and contractors
- update solution design to address and mitigate changes to existing security policies, standards, guidelines and procedures

Design a monitoring strategy for identity and security

- design for alert notifications
- design an alert and metrics strategy
- recommend authentication monitors

Design a data platform solution

Design a data management strategy

- choose between managed and unmanaged data store
- choose between relational and non-relational databases
- design a data auditing strategy
- design a data caching strategy
- identify data attributes
- recommend database service tier sizing
- design a data retention policy
- design for data availability
- design for data consistency
- design for data durability
- design a data warehouse strategy

Design a data protection strategy

- recommend geographic data storage
- design an encryption strategy for data at rest
- design an encryption strategy for data in transmission

- design an encryption strategy for data in use
- design a scalability strategy for data
- design secure access to data
- design a data loss prevention (DLP) policy

Design and document data flows

- identify data flow requirements
- create a data flow diagram
- design a data flow to meet business requirements
- design data flow solutions
- design a data import and export strategy

Design a monitoring strategy for the data platform

- design for alert notifications
- design an alert and metrics strategy
- monitor Azure Data Factory pipelines

Design a business continuity strategy

Design a site recovery strategy

- design a recovery solution
- design a site recovery replication policy
- design for site recovery capacity
- design for storage replication
- design site failover and failback
- design the site recovery network
- recommend recovery objectives
- identify resources that require site recovery
- identify supported and unsupported workloads
- recommend a geographical distribution strategy

Design for high availability

- design for application redundancy
- design for autoscaling
- design for data center and fault domain redundancy
- design for network redundancy
- identify resources that require high availability
- identify storage types for high availability
- design a disaster recovery strategy for individual workloads
- design failover/failback scenarios
- document recovery requirements

- identify resources that require backup
- recommend a geographic availability strategy

Design a data archiving strategy

- recommend storage types and methodology for data archiving
- identify business compliance requirements for data archiving
- identify requirements for data archiving
- identify SLA(s) for data archiving

Design for deployment, migration, and integration

Design deployments

- design a compute deployment strategy
- design a container deployment strategy
- design a data platform deployment strategy
- design a messaging solution deployment strategy
- design a storage deployment strategy
- design a web app and service deployment strategy

Design migrations

- recommend a migration strategy
- design data import/export strategies during migration
- determine the appropriate application migration method
- determine the appropriate data transfer method
- determine the appropriate network connectivity method
- determine migration scope, including redundant, related, trivial, and outdated data
- determine application and data compatibility

Design an API integration strategy

- design an API gateway strategy
- determine policies for internal and external consumption of APIs
- recommend a hosting structure for API management

Design an infrastructure strategy

Design a storage strategy

- design a storage provisioning strategy
- design storage access strategy

- identify storage requirements
- recommend a storage solution
- recommend storage management tools

Design a compute strategy

- design a compute provisioning strategy
- design a secure compute strategy
- determine appropriate compute technologies
- design an Azure HPC environment
- identify compute requirements
- recommend management tools for compute

Design a networking strategy

- design a network provisioning strategy
- design a network security strategy
- determine appropriate network connectivity technologies
- identify networking requirements
- recommend network management tools
- recommend network security solutions

Design a monitoring strategy for infrastructure

- design for alert notifications
- design an alert and metrics strategy