Exam AZ-301: Microsoft Azure Architect Design – Skills Measured

A NEW VERSION OF THIS EXAM, AZ-304, WILL BE AVAILABLE ON OR AROUND JUNE 29, 2020. You will be able to take this exam until it retires on or around September 30, 2020. The exam guide for AZ-304 is appended below.

Audience Profile

Candidates for this exam are Azure Solution Architects who advise stakeholders and translates business requirements into secure, scalable, and reliable solutions.

Candidates 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 requires managing how decisions in each area affects an overall solution.

Candidates must be proficient in Azure administration, Azure development, and DevOps, and have expert-level skills in at least one of those domains.

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).

Determine workload requirements (10-15%)

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 (tags) for resources to be monitored
- determine levels and storage locations for logs
- plan for integration with monitoring tools
- recommend appropriate monitoring tool(s) for a solution
- specify mechanism for event routing and escalation
- design auditing for compliance requirements
- design auditing policies and traceability requirements

Design for identity and security (20-25%)

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 (15-20%)

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 (10-15%)

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 (Azure, on-prem, hybrid, Recovery Time Objective (RTO), Recovery Level Objective (RLO), Recovery Point Objective (RPO))
- 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 (10-15%)

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 (15-20%)

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

AZ-301/AZ304 Comparison

Microsoft Azure Architect Design Exam

Current Skills Measured as of January 15, 2020	Updated list of Skills Measured (Ignore the numbering)
Audience Profile Candidates for this exam are Azure Solution Architects who advise stakeholders and translates business requirements into secure, scalable, and reliable solutions. Candidates 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 requires managing how decisions in each area affects an overall solution. Candidates must be proficient in Azure administration, Azure development, and DevOps, and have expert-level skills in at least one of those domains.	**Audience Profile** Candidates for this exam are Azure Solution Architects who advise stakeholders and translate business requirements into secure, scalable, and reliable solutions. Candidates should have advanced experience and knowledge of IT operations, including networking, virtualization, identity, security, business continuity, disaster recovery, data platform, budgeting, and governance. This role requires managing how decisions in each area affects an overall solution. Candidates must have expert-level skills in Azure administration and have experience with Azure development processes and DevOps processes.
1. Determine Workload Requirements **1.1. Gather Information and requirements** • identify compliance requirements • identify identity and access management infrastructure • identify service-oriented architectures (integration patterns, service design, service discoverability) • identify accessibility requirements (Web Content Accessibility Guidelines (WCAG)) • identify availability requirements	**7. Design Monitoring (10-15%)** **7.1. Design for cost optimization** • recommend a solution for cost management and cost reporting • recommend solutions to minimize costs **7.2. Design a solution for logging and monitoring** • determine levels and storage locations for logs

(Service Level Agreement) • identify capacity planning and scalability requirements • identify deployability requirements (repositories, failback, slot-based deployment) • identify configurability • identify governance requirements • identify maintainability requirements (logging, debugging, troubleshooting, recovery, training) • identify security requirements (authentication, authorization, attacks) • identify sizing requirements (i.e., support costs, optimization) • recommend changes during project execution (ongoing) • evaluate products and services to align with solution • create testing scenarios **1.2. Optimize Consumption Strategy** • optimize app service costs • optimize compute costs • optimize identity costs • optimize network costs • optimize storage costs **1.3. Design an Auditing and Monitoring Strategy** • define logical groupings (tags) for resources to be monitored • determine levels and storage locations for logs • plan for integration with monitoring tools • recommend appropriate monitoring tool(s) for a solution • specify mechanism for event routing and escalation • design auditing for compliance requirements	• plan for integration with monitoring tools including Azure Monitor and Azure Sentinel • recommend appropriate monitoring tool(s) for a solution • choose a mechanism for event routing and escalation • recommend a logging solution for compliance requirements • NOT: resource-specific monitoring. This objective should ONLY cover the all-up holistic monitoring strategy

• design auditing policies and traceability requirements	
2. Design for Identity and Security **2.1. Design Identity Management** • choose an identity management approach • design an identity delegation strategy • design an identity repository (directory, application, systems, etc.) • design self-service identity management • design user and persona provisioning • define personas • define roles • recommend appropriate access control strategy (Attribute-based, Discretionary Access, History-based, Identity-based, Mandatory, Organization-based, Role-based, Rule-based, Responsibility-based) **2.2. 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 **2.3. Design Authorization** • choose an authorization approach • define access permissions and privileges • design secure delegated access (oAuth, OpenID, claims etc.) • recommend when and how to use API	**8. Design Identity and Security (25-30%)** **8.1. Design authentication** • recommend a solution for single-sign on • recommend a solution for authentication • recommend a solution for Conditional Access, including multi-factor authentication • recommend a solution for network access authentication • recommend a solution for a hybrid identity including Azure AD Connect and Azure AD Connect Health • recommend a solution for user self-service • recommend and implement a solution for B2B integration • NOT: federation with ADFS **8.2. Design authorization** • choose an authorization approach • recommend a hierarchical structure that includes management groups, subscriptions and resource groups • recommend an access management solution including RBAC policies, access reviews, role assignments, physical access, Privileged Identity Management (PIM), Azure AD Identity Protection, Just In Time (JIT) access **8.3. Design governance** • recommend a strategy for tagging • recommend a solution for using Azure Policy • recommend a solution for using Azure

Keys
 	
2.4. Design for Risk Prevention for Identity
 	
• design a risk assessment strategy (e.g., access reviews, RBAC policies, physical access)
• 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
 	
2.5. Design a Monitoring Strategy for Identity and Security
 	
• design for alert notifications
• design an alert and metrics strategy
• recommend authentication monitors
	Blueprint
	
	8.4. Design security for applications
	
	• recommend a solution that includes KeyVault
	  ○ what can be stored in KeyVault
	  ○ KeyVault operations
	  ○ KeyVault regions
	• recommend a solution that includes Azure AD Managed Identities
	• recommend a solution for integrating applications into Azure AD

3. Design a Data Platform Solution **3.1. 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 (relevancy, structure, frequency, size, durability, etc.) • 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	**9. Design Data Storage (15-20%)** **9.1. Design a solution for databases** • select an appropriate data platform based on requirements • recommend database service tier sizing • recommend a solution for database scalability • recommend a solution for encrypting data at rest, data in transmission, and data in use • NOT: data caching • NOT: MariaDB, PostGreSQL, MySQL **9.2. Design data integration** • recommend a data flow to meet business requirements • recommend a solution for data integration, including Azure Data

3.2. 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 **3.3. 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 **3.4. Design a Monitoring Strategy for the Data Platform** • design for alert notifications • design an alert and metrics strategy • monitor Azure Data Factory pipelines	Factory, Azure Data Bricks, Azure Data Lake, Azure Synapse Analytics **9.3. Select an appropriate storage account** • choose between storage tiers • recommend a storage access solution • recommend storage management tools
4. Design a Business Continuity Strategy **4.1. 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 (planned/unplanned) • design the site recovery network • recommend recovery objectives (Azure,	**10. Design Business Continuity (10-15%)** **10.1. Design a solution for backup and recovery** • recommend a recovery solution for Azure hybrid and on-premises workloads that meets recovery objectives (RTO, RLO, RPO) • design and Azure Site Recovery solution ○ recommend a site recovery

on-prem, hybrid, Recovery Time Objective (RTO), Recovery Level Objective (RLO), Recovery Point Objective (RPO)) • identify resources that require site recovery • identify supported and unsupported workloads • recommend a geographical distribution strategy **4.2. 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 scenario(s) • document recovery requirements • identify resources that require backup • recommend a geographic availability strategy **4.3. [there is no objective 4.3]** **4.4. 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	replication policy o recommend a solution for site recovery capacity o recommend a solution for site failover and failback (planned/unplanned) o recommend a solution for the site recovery network • recommend a solution for recovery in different regions • recommend a solution for Azure Backup management • design a solution for data archiving and retention o recommend storage types and methodology for data archiving o identify business compliance requirements for data archiving o identify requirements for data archiving o identify SLA(s) for data archiving o recommend a data retention policy **10.2. Design for high availability** • recommend a solution for application and workload redundancy, including compute, database, and storage • recommend a solution for autoscaling • identify resources that require high availability • identify storage types for high availability • recommend a solution for geo-redundancy of workloads
5. Design for Deployment, Migration,	**11. Design Infrastructure (25-30%)**

and Integration

5.1. 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

5.2. 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

5.3. 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

6. Design an Infrastructure Strategy

11.1. Design a compute solution

- recommend a solution for compute provisioning
- determine appropriate compute technologies, including virtual machines, App Services, Service Fabric, Azure Functions, Windows Virtual Desktop, and containers
- recommend a solution for containers
  - AKS versus ACI and the configuration of each one
- recommend a solution for automating compute management
- NOT: monitoring, backups, recovery, availability, security, storage; VMWare

11.2. Design a network solution

- recommend a solution for network addressing and name resolution
- recommend a solution for network provisioning
- recommend a solution for network security
  - private endpoints
  - firewalls
  - gateways
  - etc.
- recommend a solution for network connectivity to the Internet, on-premises networks, and other Azure virtual networks
- recommend a solution for automating network management
- recommend a solution for load balancing and traffic routing

11.3. Design an application architecture

- recommend a microservices architecture including Event Grid, Event

6.1. Design a Storage Strategy

- design a storage provisioning strategy
- design storage access strategy
- identify storage requirements
- recommend a storage solution
- recommend storage management tools

6.2. Design a Compute Strategy

- design a compute provisioning strategy
- design a secure compute strategy
- determine appropriate compute technologies (eg. Virtual machines, functions, service fabric, container instances, etc.)
- design an Azure HPC environment
- identify compute requirements
- recommend management tools for compute

6.3. 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

6.4. Design a Monitoring Strategy for Infrastructure

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

Hubs, Service Bus, Storage Queues, Logic Apps, Azure Functions, and webhooks

- recommend an orchestration solution for deployment of applications including ARM templates, Logic Apps, or Azure Functions
    - select an automation method
    - choose which resources or lifecycle steps will be automated
    - design integration with other sources such as an ITSM solution
    - recommend a solution for monitoring automation
- recommend a solution for API integration
    - design an API gateway strategy
    - determine policies for internal and external consumption of APIs
    - recommend a hosting structure for API management
    - recommend when and how to use API Keys

11.4. Design migrations

- assess and interpret on-premises servers, data, and applications for migration
- recommend a solution for migrating applications and VMs
- recommend a solution for migration of databases
    - determine migration scope, including redundant, related, trivial, and outdated data