

Course:

AZUREAPES : ARCHITECTING AZURE SOLUTIONS BOOTCAMP

AUDIENCE PROFILE

Candidates for this course define the appropriate cloud native, cloud migration, and hybrid cloud solutions to meet the required functional, operational, and deployment requirements through the solution lifecycle.

Course participants will learn and practice about the features and capabilities of Azure services to be able to identify trade-offs and make decisions for designing public and hybrid cloud solutions. About 40% of the course is used for Hands-On-Labs, and another 25% of the time will be used to have architectural design sessions in teams, based on several real-time inspired business case studies. The attendee will absorb knowledge around understanding and executing DevOps concepts, learn how to provision Azure resources using ARM templates, and designing highly resilient workloads running on Azure, taking Business Continuity and Disaster Recovery into consideration.

Although not specifically built as such, this course is a more than helpful training resource towards taking the Microsoft Official Exam 70-535 Architecting Azure Solutions.

OBJECTIVE DOMAINS THROUGHOUT THE COURSE

1. DESIGN COMPUTE INFRASTRUCTURE

1.1. DESIGN SOLUTIONS USING VIRTUAL MACHINES

Design VM deployments by leveraging availability sets, fault domains, and update domains in Azure; use web app for containers; design VM Scale Sets; design for compute-intensive tasks using Azure Batch; define a migration strategy from cloud services; recommend use of Azure Backup and Azure Site Recovery.

1.2. DESIGN SOLUTIONS FOR SERVERLESS COMPUTING

Use Azure Functions to implement event-driven actions; design for serverless computing using Azure Container Instances; design application solutions by using Azure Logic Apps, Azure Functions, or both; determine when to use API management service.

1.3. DESIGN MICROSERVICES-BASED SOLUTIONS

Determine when a container-based solution is appropriate; determine when container-orchestration is appropriate; determine when Azure Service Fabric (ASF) is appropriate; determine when Azure Functions is appropriate; determine when to use API management service; determine when Web API is appropriate; determine which platform is appropriate for container orchestration; consider migrating existing assets versus cloud native deployment; design lifecycle management strategies.

1.4. DESIGN WEB APPLICATIONS

Design Azure App Service Web Apps; design custom web API; secure Web API; design Web Apps for scalability and performance; design for high availability using Azure Web Apps in multiple regions; determine which App service plan to use; design Web Apps for business continuity; determine when to use Azure App Service Environment (ASE); design for API apps; determine when to use API management service; determine when to use Web Apps on Linux; determine when to use a CDN; determine when to use a cache, including Azure Redis cache.

1.5. CREATE COMPUTE-INTENSIVE APPLICATIONS

Design high-performance computing (HPC) and other compute-intensive applications using Azure Services; determine when to use Azure Batch; design stateless components to accommodate scale; design lifecycle strategy for Azure Batch.

2. DESIGN DATA IMPLEMENTATION

2.1. ARCHITECT FOR AZURE STORAGE SOLUTIONS

Determine when to use Azure Blob Storage, blob tiers, Azure Files, disks, and StorSimple

2.2. ARCHITECT FOR AZURE DATA SERVICES

Determine when to use Data Catalog, Azure Data Factory, SQL Data Warehouse, Azure Data Lake Analytics, Azure Analysis Services, and Azure HDInsight

2.3. ARCHITECT FOR RELATIONAL DATABASE STORAGE

Determine when to use Azure SQL Database and SQL Server Stretch Database; Architect for scalability and features; determine when to use Azure Database for MySQL and Azure Database for PostgreSQL; Architect for HA/DR, geo-replication; Architect a backup and recovery strategy

2.4. ARCHITECT FOR NOSQL STORAGE

Determine when to use Azure Redis Cache, Azure Table Storage, Azure Data Lake, Azure Search, Time Series Insights

2.5. ARCHITECT FOR COSMOSDB STORAGE

Determine when to use MongoDB API, DocumentDB API, Graph API, Azure Tables API; Architect for cost, performance, data consistency, availability, and business continuity

3. DESIGN NETWORKING IMPLEMENTATION

3.1. 3.1 DESIGN AZURE VIRTUAL NETWORKS

Design solutions that use Azure networking services: design for load balancing using Azure Load Balancer and Azure Traffic Manager; define DNS, DHCP, and IP strategies; determine when to use Azure Application Gateway; determine when to use multi-node application gateways, Traffic Manager and load balancers.

3.2. 3.2 DESIGN EXTERNAL CONNECTIVITY FOR AZURE VIRTUAL NETWORKS

Determine when to use Azure VPN, ExpressRoute and Virtual Network Peering architecture and design; determine when to use User Defined Routes (UDRs); determine when to use VPN gateway site-to-site failover for ExpressRoute.

3.3. 3.3 DESIGN SECURITY STRATEGIES

Determine when to use network virtual appliances; design a perimeter network (DMZ); determine when to use a Web Application Firewall (WAF), Network Security Group (NSG), and virtual network service tunneling.

3.4. 3.4 DESIGN CONNECTIVITY FOR HYBRID APPLICATIONS

Design connectivity to on-premises data from Azure applications using Azure Relay Service, Azure Data Management Gateway for Data Factory, Azure On-Premises Data Gateway, Hybrid Connections, or Azure Web App's virtual private network (VPN) capability; identify constraints for connectivity with VPN; identify options for joining VMs to domains.

4. DESIGN SECURITY AND IDENTITY SOLUTIONS

4.1. DESIGN AN IDENTITY SOLUTION

Design AD Connect synchronization; design federated identities using Active Directory Federation Services (AD FS); design solutions for Multi-Factor Authentication (MFA); design an architecture using Active Directory on-premises and Azure Active Directory (AAD); determine when to use Azure AD Domain Services; design security for Mobile Apps using AAD.

4.2. SECURE RESOURCES BY USING IDENTITY PROVIDERS

Design solutions that use external or consumer identity providers such as Microsoft account, Facebook, Google, and Yahoo; determine when to use Azure AD B2C and Azure AD B2B; design mobile apps using AAD B2C or AAD B2B.

4.3. DESIGN A DATA SECURITY SOLUTION

Design data security solutions for Azure services; determine when to use Azure Storage encryption, Azure Disk Encryption, Azure SQL Database security capabilities, and Azure Key Vault; design for protecting secrets in ARM templates using Azure Key Vault; design for protecting application secrets using Azure Key Vault; design a solution for managing certificates using Azure Key Vault; design solutions that use Azure AD Managed Service Identity.

4.4. DESIGN A MECHANISM OF GOVERNANCE AND POLICIES FOR ADMINISTERING AZURE RESOURCES

Determine when to use Azure RBAC standard roles and custom roles; define an Azure RBAC strategy; determine when to use Azure resource policies; determine when to use Azure AD Privileged Identity Management; design solutions that use Azure AD Managed Service Identity; determine when to use HSM-backed keys.

4.5. MANAGE SECURITY RISKS BY USING AN APPROPRIATE SECURITY SOLUTION

Identify, assess, and mitigate security risks by using Azure Security Center, Operations Management Suite Security and Audit solutions, and other services; determine when to use Azure AD Identity Protection; determine when to use Advanced Threat Detection; determine an appropriate endpoint protection strategy.

5. DESIGN SOLUTIONS BY USING PLATFORM SERVICES

5.1. DESIGN FOR ARTIFICIAL INTELLIGENCE SERVICES

Determine when to use the appropriate Cognitive Services, Azure Bot Service, Azure Machine Learning, and other categories that fall under cognitive AI

5.2. DESIGN FOR IOT

Determine when to use Stream Analytics, IoT Hubs, Event Hubs, real-time analytics, Time Series Insights, IoT Edge, Notification Hubs, Event Grid, and other categories that fall under IoT

5.3. DESIGN MESSAGING SOLUTION ARCHITECTURES

Design a messaging architecture; determine when to use Azure Storage Queues, Azure Service Bus, Azure Event Hubs, Event Grid, Azure Relay, Azure Functions, and Azure Logic Apps; design a push notification strategy for Mobile Apps; design for performance and scale

5.4. DESIGN FOR MEDIA SERVICE SOLUTIONS

Define solutions using Azure Media Services, video indexer, video API, computer vision API, preview, and other media related services

6. DESIGN FOR OPERATIONS

6.1. DESIGN AN APPLICATION MONITORING AND ALERTING STRATEGY

Determine the appropriate Microsoft products and services for monitoring applications on Azure; define solutions for analyzing logs and enabling alerts using Azure Log Analytics; define solutions for analyzing performance metrics and enabling alerts using Azure Monitor; define a solution for monitoring applications and enabling alerts using Application Insights

6.2. DESIGN A PLATFORM MONITORING AND ALERTING STRATEGY

Determine the appropriate Microsoft products and services for monitoring Azure platform solutions; define a monitoring solution using Azure Health, Azure Advisor, and Activity Log; define a monitoring solution for Azure Networks using Log Analytics and Network Watcher service; monitor security with Azure Security Center

6.3. DESIGN AN OPERATIONS AUTOMATION STRATEGY

Determine when to use Azure Automation, Chef, Puppet, PowerShell, Desired State Configuration (DSC), Event Grid, and Azure Logic Apps; define a strategy for auto-scaling; define a strategy for enabling periodic processes and tasks

Why training from Azure-APEs

- +20 years IT Industry background
- Long-standing Microsoft Certified Trainers (+10 years experience)
- Most trainers are also MVPs (Azure or other specific domains)
- We provide training all over the globe
- We already trained more than 25.000 students worldwide, within larger customers, Microsoft Partners as well as Microsoft FTEs
- Tight contacts with Azure and Azure Stack Engineering, Marketing and Support Teams
- Quarterly visits to the Microsoft HQ in Redmond to discuss business planning, engineering roadmap and technical insights
- All training content (presentations, hands-on-labs, case studies,...) we provide is self-authored, up-to-date
- Room for discussions, time for question-answering
- Recognized technical writers, book authors, courseware creators
- We love sharing knowledge, with a passion for providing training