ExamLabs

Microsoft Azure DevOps Solutions
Study Guide
Exam AZ-400

Table of Contents

1. Getting Started
    - Cloud computing
        - Advantages of cloud computing
        - Deployment Patterns in Azure
        - Infrastructure as a Service
        - Platform as a Service
        - Software as a Service
    - Understanding Azure
    - Azure as an intelligent cloud
    - Azure Resource Manager
        - The ARM architecture
        - Limitations of Azure Service Manager (ASM)
        - ARM advantages
        - ARM concepts
            - Resource providers
            - Resource types
            - Resource groups
            - Resource and resource instances

ARM features

Virtualization

Containers

Docker

Interacting with the intelligent cloud

Azure Portal

PowerShell

Azure Command-Line Interface (CLI)

Azure REST API

ARM templates

Deployments

Summary

2. Azure Solution Availability and Scalability

High availability

SLA

Factors affecting high availability

Planned maintenance

Unplanned maintenance

Application deployment architecture

High availability versus scalability

High availability versus disaster recovery

Azure high availability

Concepts

Availability sets

Fault domain

Update domain

Availability zones

Load balancing

VM high availability

Compute high availability

Storage high availability

PaaS high availability

High-availability platforms

Data high availability

Azure CosmosDB

Azure SQL replication

Azure table storage

Application high availability

Load balancing

Azure load balancers

- Public load balancing
- Internal load balancing
- Port forwarding
- Azure application gateway
- Azure Traffic Manager

Architectural considerations for high availability

- High availability within Azure regions
- High availability across Azure regions
- Best practices
  - Application high availability
  - Deployment
  - Data management
  - Monitoring

Scalability

- Scalability versus performance
- Azure scalability
  - Concepts
    - Scaling
    - Scaling up
    - Scaling down
    - Scaling out
    - Scaling in
    - Auto scaling
  - PaaS scalability
    - PaaS – Scaling up and down
    - PaaS – Scaling out and in
  - IaaS scalability
  - VM scale sets
    - VMSS architecture
    - VMSS scaling
      - Horizontal versus vertical scaling
      - Capacity
      - Auto scaling

Upgrades and maintenance

- Application updates
- Guest updates
- Image updates
- Best practices of scaling provided by VMSS
  - The preference for scaling out
  - Bare-metal versus dormant instances
  - Configuring the maximum and minimum number of instances appropri

- ately
- Concurrency
- Stateless
- Caching and CDN
- N+1 design

Summary

3. Security and Monitoring
   - Security
     - Security life cycle
     - Azure security
     - IaaS security
       - Network Security Groups
       - NSG design
       - Firewalls
         - Firewall design
       - Reducing the attack surface area
       - Implementing jump servers
     - PaaS security
       - Operations Management Suite (OMS)
       - Storage
     - Azure SQL
     - Azure Key Vaults
     - Security monitoring and auditing
       - Azure Monitor
       - Azure Security Center
   - Monitoring
   - Azure monitoring
     - Azure activity logs
     - Azure diagnostic logs
     - Azure application logs
     - Guest and host operating system logs
     - Azure Monitor
     - Azure Application Insights
     - Azure Log Analytics
   - Application Insights
     - Provisioning
   - Log Analytics
     - Provisioning
     - OMS agents
     - Search

Solutions
Alerts
Executing runbooks on Alerts
Integrating PowerBI
Summary
4. Cross-Subscription Deployments Using ARM Templates
ARM templates
Deploying resource groups with ARM templates
Deploying resources across subscriptions and resource groups
Another example of cross-subscription and resource-group deployments
Deploying cross-subscription and resource-group deployments using linked te
mplates
Summary
5. ARM Templates - Modular Design and Implementation
Problems with the single template
Reduces flexibility in changing templates
Troubleshooting large templates
Dependency abuse
Reduced agility
No reusability
Understanding the Single Responsibility Principle
Faster troubleshooting and debugging
Modular templates
Deployments resources
Linked templates
Nested templates
Free-flow configurations
Known configurations
Summary
6. Designing and Implementing Serverless Solutions
Serverless
The evolution of serverless
Principles of serverless technology
Azure Functions advantages
FaaS
Azure Functions runtime
Azure Functions bindings and triggers
Monitoring
Authentication and authorization

- Azure Functions configuration
  - Platform configuration
  - App Service Function settings
- Azure Functions cost plans
- Azure Functions use cases
- Types of Azure Functions

Creating your first Azure Functions

Creating an event-driven Function

Function proxies

Understanding workflows
Durable Functions
- Steps for creating a Durable Functions

Creating a connected architecture with Functions

Summary

7. Azure Integration Solutions
   Azure Event Grid
   - Event Grid architecture
   - Resource events
   - Custom events

   Azure Logic Apps
   - Activity
   - Connectors
   - Working on a logic app

   Creating an end-to-end solution using Serverless technologies
   - Problem statement
   - Vision
   - Solution
   - Architecture
     - Azure Automation
     - Custom Azure Event Grid topic
     - Azure Logic Apps
     - Azure Functions
   - Prerequisites
   - Implementation
     - Step 1
     - Step 2
     - Step 3
     - Step 4
     - Step 5
     - Step 6

Step 7
Step 8
Step 9
Step 10
Step 11
Testing
Summary
8. Cost Management
Understanding billing
Invoicing
Enterprise agreement customers
Usage and quotas
Resource providers
The usage and billing APIs
Azure pricing models
Azure Hybrid Benefit
Azure reserved virtual machine instances
Pay-as-you-go accounts
Enterprise Agreements
The cloud solution provider model
The Azure pricing calculator
Best practices
Compute best practices
Storage best practices
Platform-as-a-Service (PaaS) best practices
General best practices
Summary
9. Designing Policies, Locks, and Tags
Azure tags
Tags with PowerShell
Tags with Azure Resource Manager templates
Resource groups versus resources
Azure policies
Built-in policies
Policy language
Allowed fields
Azure locks
Azure RBAC
Custom Roles

How are locks different from RBAC?
An example of implementing Azure governance features
Background
RBAC for Company Inc
Azure policies
Deployments to certain location
Tags of resources and Resource Groups
Diagnostic logs and Application Insights for all resources
Azure Locks
Summary
10. Azure Solutions Using Azure Container Services
Azure Container Registry
Azure Container Instances
Azure Kubernetes Service
Kubernetes architecture
Master nodes
Pods
API server
Kubelets
Kube-Proxy
Replication controller/controller manager
Azure Kubernetes architecture
Provisioning Azure Kubernetes Service
App Service containers
Comparing all container options
Containers on virtual machines
Containers on virtual machines with Kubernetes as the orchestrator
Azure Kubernetes Service
Containers on Azure App Service
Containers in Azure Container Instances
Containers in Azure Functions
Containers in Service Fabric
Summary
11. Azure DevOps
DevOps
DevOps practices
Configuration management
Desired State Configuration
Chef, Puppet, and Ansible
ARM Templates

- Continuous integration
  - Build automation
  - Test automation
  - Packaging
- Continuous deployment
  - Test environment deployment
  - Test automation
  - Staging environment deployment
  - Acceptance tests
  - Deployment to production
- Continuous delivery
- Continuous learning

Azure DevOps

- Team Foundation Version Control
- Git

Preparing for DevOps

- Provisioning Azure DevOps organization
- Provisioning the Azure Key Vault
- Provisioning a configuration-management server/service
- Provisioning log analytics
- Azure Storage account
- Source images
- Monitoring tools
- Management tools

DevOps for PaaS solutions

- Azure App Services
- Deployment slots
- Azure SQL
- The build-and-release pipeline

DevOps for virtual machine (IaaS)-based solutions

- Azure Virtual Machines (VM)
- Azure public load balancers
- The build pipeline
- The release pipeline

DevOps for container-based (IaaS) solutions

- Containers
  - Docker
  - Dockerfile
- The build pipeline
- The release pipeline

Azure DevOps and Jenkins
Azure Automation
    Provisioning the Azure Automation account
    Creating DSC configuration
    Importing the DSC configuration
    Compiling the DSC configuration
    Assigning configurations to nodes
    Browsing the server
Azure for DevOps
Summary

12. Azure OLTP Solutions Using Azure SQL Sharding, Pools, and Hybrid
Azure cloud services
OLTP applications
Relational databases
Deployment models
    Databases on Azure virtual machines
    Databases hosted as managed services
Azure SQL Database
    Application features
Single instance
    High availability
    Backups
    Geo-replication
    Scalability
    Security
        Firewall
        Azure SQL Server on dedicated networks
        Encrypted databases at rest
        Dynamic Data Masking
        Azure Active Directory integration
Elastic pools
Managed Instance
SQL database pricing
    DTU-based pricing
    vCPU based pricing
    How to choose the appropriate pricing model
Summary

13. Azure Big Data Solutions Using Azure Data Lake Storage and Data Factory
Data integration

ETL

A primer on Data Factory
A primer on Data Lake Storage
Understanding big data processing
Ingestion
Processing
Storage for consumption
Presentation of data
Migrating data from Azure Storage to Data Lake Gen2 Storage
Preparing the source storage account
Provisioning a new resource group
Provisioning a storage account
Creating a new Data Lake Gen2 service
Creating a new Data Factory pipeline
Repository settings
Creating the first dataset
Creating the second dataset
Creating a third dataset
Creating a pipeline
Add one more copy data activity
Publishing
Final result
Summary

14. Azure Stream Analytics and Event Hubs
A primer on Event Hubs
Events
Event streaming
Event Hubs
Architecture of Event Hubs
Consumer groups
Throughput
A primer on Stream Analytics
Hosting environment
Streaming units
A sample application using Event Hubs and Stream Analytics
Provisioning a new resource group
Creating an Event Hubs namespace
Creating an event hub
Provisioning logic apps
Provisioning the Storage account

Creating a storage container

Creating Stream Analytics jobs

Running the application

Summary

15. Designing IoT Solutions

IoT

IoT architecture

Connectivity

Identity

Capture

Ingestion

Storage

Transform

Analytics

Presentation

Azure IoT

Identity

Capture

Ingestion

Storage

Transform and analytics

Presentation

IoT Hubs

Protocols

Device registration

Message management

Device-to-cloud messaging

Cloud-to-device messaging

Security

Security in IoT

Scalability

SKU edition

Units

High availability

Summary

Preface

Over the years, Azure cloud services have grown quickly, and the number of organizations adopting Azure for their cloud services has also been on the increase. Leading industry giants are discovering that Azure fulfills their extensive cloud requirements.

This book starts with an extensive introduction to all the categories of designs available with Azure. These design patterns focus on different aspects of the cloud, including high availability and data management. Gradually, we move on to various other aspects, such as building your cloud deployment and architecture. Every architect should have a good grasp of some of the important architectural concerns related to any application. These relate to high availability, security, scalability, and monitoring. They become all the more important because the entire premise of the cloud is dependent on these important concerns. This book will provide architects with all the important options related to scalability, availability, security, and the monitoring of **Infrastructure of a Service (IaaS)** as well as **Platform as a Service** (**PaaS**) deployments. Data has become one of the most important aspects of cloud applications. This book covers the architecture and design considerations for deploying **Online Transaction Processing** (**OLTP**) applications on Azure. Big data and related data activities, including data cleaning, filtering, formatting, and using **Extract-Transform-Load** (**ETL**) services are provided by the Azure Data Factory service. Finally, serverless technologies are gaining a lot of traction with their orchestration using Azure Logic Apps. This will also be covered comprehensively in this book.

By the end of this book, you will be able to develop a fully-fledged Azure cloud instance.

What this book covers

`Chapter 1`, *Getting Started*, introduces the Azure cloud platform. It provides details regarding IaaS and PaaS and provides an introduction to some of the important features that help in designing solutions.

`Chapter 2`, *Azure Solution Availability and Scalability*, takes you through an architect's perspective for deploying highly available and scalable applications on Azure.

`Chapter 3`, *Security and Monitoring*, helps you to understand how security is undoubtedly the most important non-functional requirement for architects to implement.

`Chapter 4`, *Cross-Subscription Deployments Using ARM Templates*, explains how ARM templates are the preferred mechanism for provisioning resources.

`Chapter 5`, *ARM Templates – Modular Design and Implementation*, focuses on writing modular, maintainable, and extensible Azure Resource Manager (ARM) templates.

`Chapter 6`, *Designing and Implementing Serverless Solutions*, focuses on providing an explanation of the serverless paradigm, Azure Functions, and their capabilities.

`Chapter 7`, *Azure Integration Solutions*, is a continuation of the previous chapter, continuing the discussion on Serverless technologies, covering Azure Event Grid as part of serverless events, and Azure Logic Apps as part of Serverless workflows.

`Chapter 8`, *Cost Management*, focuses on calculating the cost of deployment on Azure using the Azure cost calculator. It also demonstrates how changing the location, size, and type of resources affects the cost of solutions and provides best practices for reducing the overall cost of Azure deployments.

`Chapter 9`, *Designing Policies, Locks, and Tags*, helps you to understand the best practices for implementing policies and locks, and how both can work together to provide complete control over Azure resources.

`Chapter 10`, *Azure Solutions Using Azure Container Services*, sheds some light on numerous services, including Azure Container Services, Azure Container Registry, and Azure Container Instances for hosting containers, as well managing them using orchestration services such as Kubernetes.

`Chapter 11`, *Azure DevOps*, is about adopting and implementing practices that reduce risk considerably and ensure that high-quality software can be delivered to the customer.

`Chapter 12`, *Azure OLTP Solutions Using Azure SQL Sharding, Pools, and Hybrid*, focuses on various aspects of using the transaction data store, such as Azure SQL, and other open source databases typically used in OLTP applications.

`Chapter 13`, *Azure Big Data Solutions Using Azure Data Lake Storage and Data Factory*, focuses on big data solutions on Azure. We will study Data Lake Storage, Data Lake Analytics, and Data Factory.

`Chapter 14`, *Azure Stream Analytics and Event Hubs*, concerns the creation of solutions for these events. It focuses on reading these events, storing and processing them, and then making sense of them.

`Chapter 15`, *Designing IoT Solutions*, covers topics related to IoT Hub, Stream Analytics, Event Hubs, registering devices, device-to-platform conversion, and logging and routing data to appropriate destinations.

Conventions used

There are a number of text conventions used throughout this book.

`CodeInText`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "Browse to the extracted `*.ova` file for Kali Linux and click **Open**."

A block of code is set as follows:

```
html, body, #map {
 height: 100%;
 margin: 0;
 padding: 0
}
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
[default]
exten => s,1,Dial(Zap/1|30)
exten => s,2,Voicemail(u100)
exten => s,102,Voicemail(b100)
exten => i,1,Voicemail(s0)
```

Any command-line input or output is written as follows:

```
$ mkdir css
$ cd css
```

Bold: Indicates a new term, an important word, or words that you see on screen. For example, words in menus or dialog boxes appear in the text like this. Here is an example: "Select System info from the Administration panel."

Warnings or important notes appear like this.

Tips and tricks appear like this.

Getting Started

Every few years, there are technological innovations that change the entire landscape and ecosystem around them. If we go back in time, the 70s and 80s were the time of mainframes. They were huge, occupying large rooms, and almost all computing work was carried out by them. It was difficult to procure one and it was also time-consuming. Enterprises used to order months in advance, before they could have an operational mainframe set up.

The first part of the 90s was the era of personal computing and the internet. Computers became much smaller in size and were comparatively easier to procure. Continuous innovation on the personal computing and internet fronts changed the entire computer industry. People had a desktop through which they could run multiple programs and could connect to the internet. The rise of the internet also propagated the rise of client-server deployments. Now, there could be centralized servers hosting applications and services that could be reached by anyone who had a connection to the internet anywhere on the globe. This was also when server technology gained a lot of prominence. Windows NT was released during this time and was followed by Windows 2000 and Windows 2003 at the turn of the century.

The most remarkable innovation of the 2000s was the rise and adoption of portable devices, especially smartphones, and with them came a plethora of apps. Apps could connect to centralized servers on the internet and could carry out business as normal. Users were no longer dependent on browsers to make this work. All servers were typically either self-hosted or hosted with a service provider, such as an **Internet Service Provider** (**ISP**).

Users did not have much control over their servers. Multiple customers and their deployments were part of the same server, even without customers knowing about it.

However, there was something else happening toward the middle and later parts of the first decade of the 2000s. This was the rise of cloud computing,

and it again rewrote the entire landscape of the IT industry. Initially, adoption was slow and people approached it with caution, either because the cloud was in its infancy and yet had to mature, or because people had various negative notions about what it was.

We will cover the following topics in the chapter:

- Cloud computing
- IaaS, PaaS, and SaaS
- Understanding Azure
- Azure Resource Manager
- Virtualization, Containers, and Docker
- Interacting with the intelligent cloud