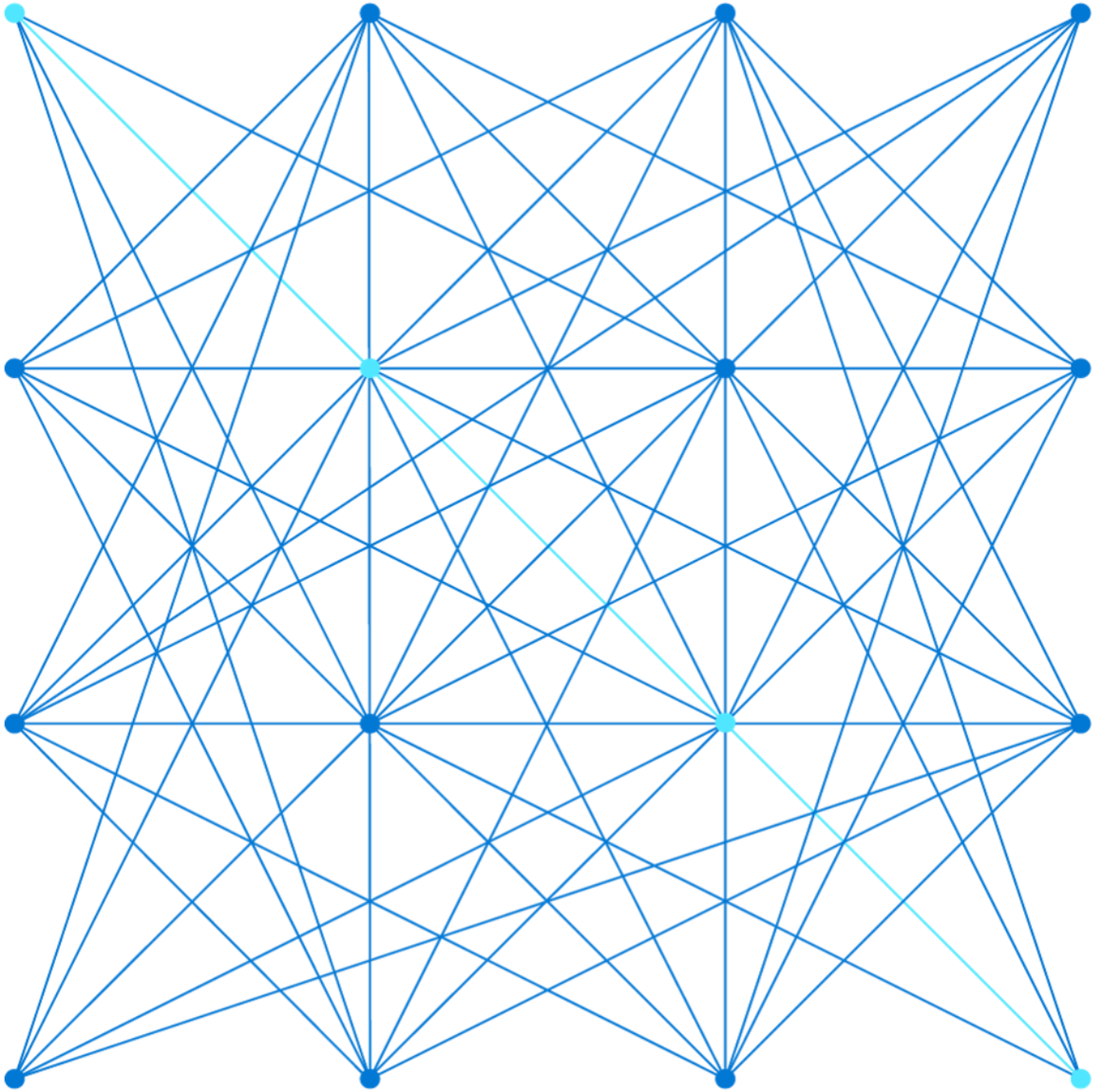
Microsoft Azure

AZ-104T00A Administer Azure Storage



Administer Azure Storage Introduction

Configure Storage Accounts

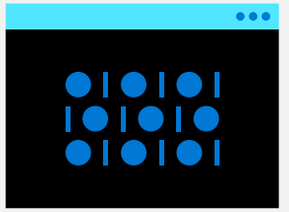
Configure Blob Storage

Configure Storage Security

Configure Azure Files and File Sync

Lab 07 – Manage Azure Storage

Configure Storage Accounts



Configure Storage Accounts Introduction

Implement Azure Storage

Explore Azure Storage Services

Determine Storage Account Kinds

Determine Replication Strategies

Access Storage

Secure Storage Endpoints

Demonstration – Secure a Storage Endpoint

Summary and Resources

Implement Azure Storage

A service that you can use to store files, messages, tables, and other types of information

Durable, secure, scalable, managed, accessible

Storage for virtual machines, unstructured data and structured data

Two tiers: Premium and Standard

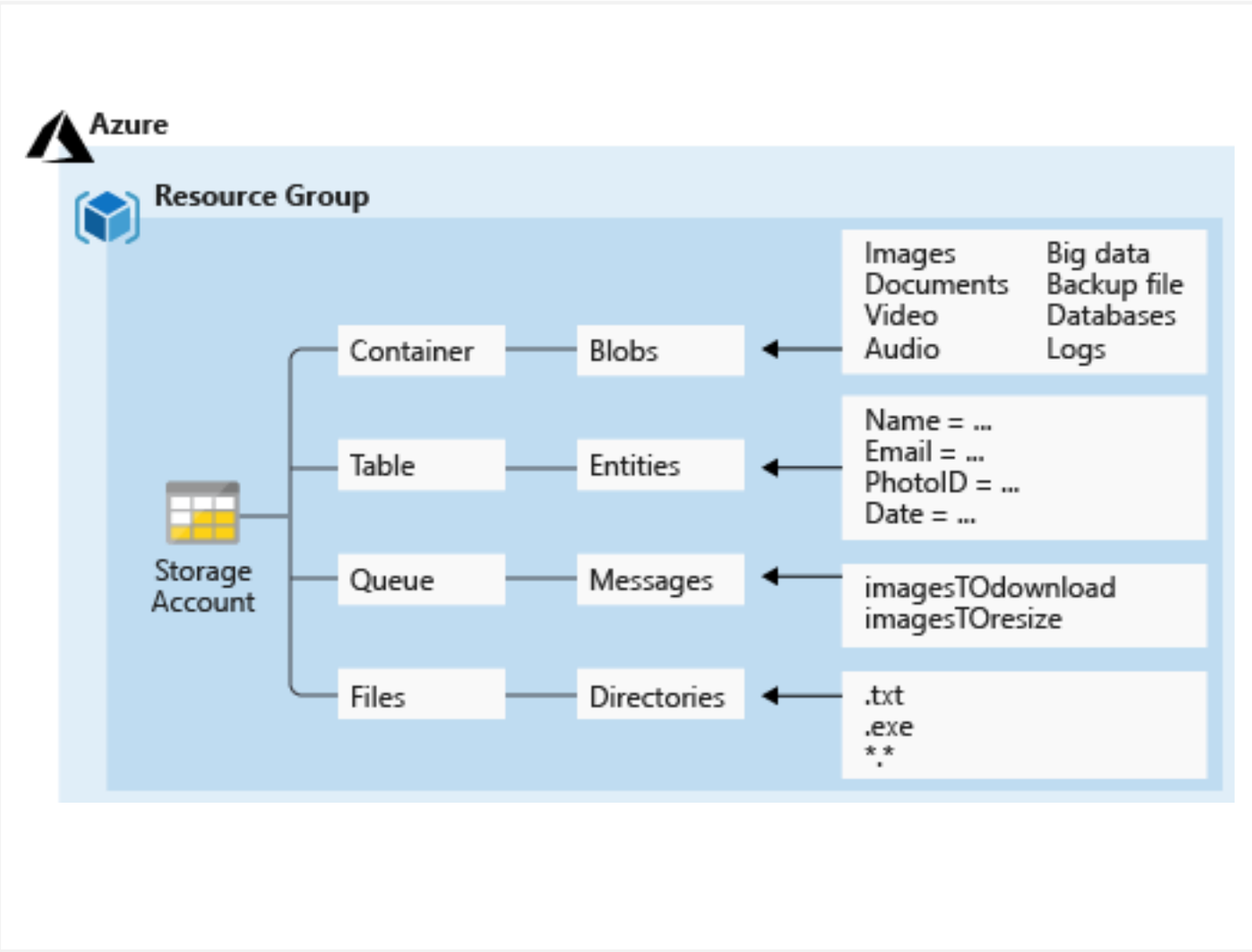
Explore Azure Storage Services

Azure Containers: A massively scalable object store for text and binary data

Azure Tables: Ideal for storing structured, non-relational data

Azure Queues: A messaging store for reliable messaging between application components

Azure Files: Managed file shares for cloud or on-premises deployments

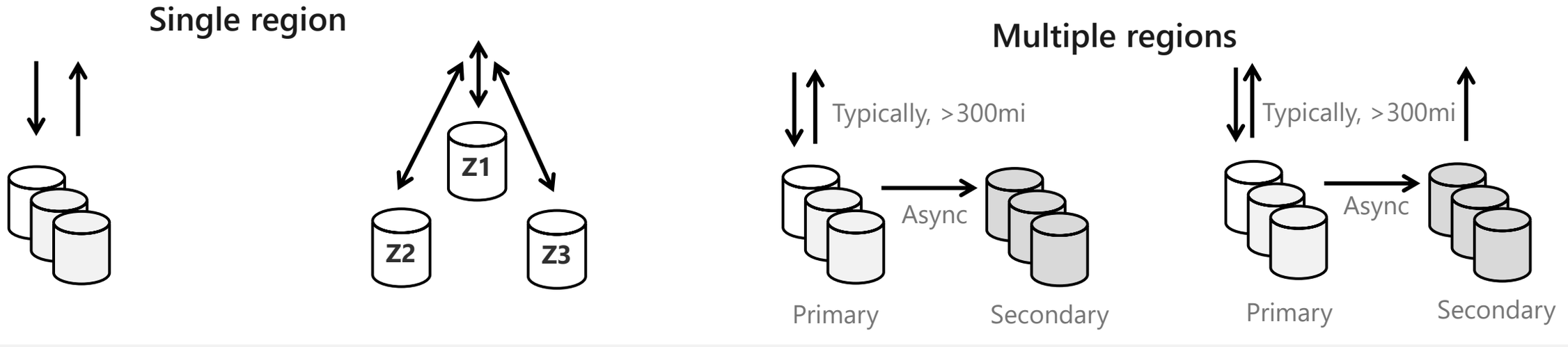


Determine Storage Account Kinds

Storage Account	Recommended usage
Standard general-purpose v2	Most scenarios including Blob, File, Queue, Table, and Data Lake Storage.
Premium block blobs	Block blob scenarios with high transactions rates, or scenarios that use smaller objects or require consistently low storage latency.
Premium file shares	Enterprise or high-performance file share applications.
Premium page blobs	Premium high-performance page blob scenarios.

All storage accounts are encrypted using Storage Service Encryption (SSE) for data at rest

Determine Replication Strategies (1 of 2)



LRS

- Three replicas, one region
- Protects against disk, node, rack failures
- Write is acknowledged when all replicas are committed
- Superior to dual-parity RAID

ZRS

- Three replicas, three zones, one region
- Protects against disk, node, rack, and zone failures
- Synchronous writes to all three zones

GRS

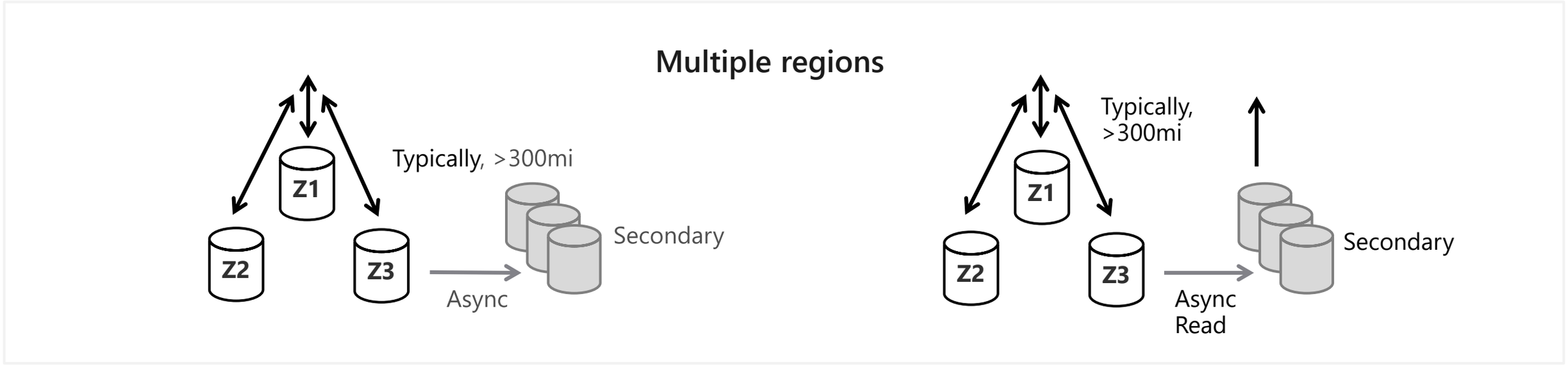
- Six replicas, two regions (three per region)
- Protects against major regional disasters
- Asynchronous copy to secondary

RA-GRS

- GRS + read access to secondary
- Separate secondary endpoint
- Recovery point objective (RPO) delay to secondary can be queried

Continued next slide

Determine Replication Strategies (2 of 2)



GZRS

- Six replicas, 3+1 zones, two regions
- Protects against disk, node, rack, zone, and region failures
- Synchronous writes to all three zones and asynchronous copy to secondary

RA-GZRS

- GZRS + read access to secondary
- Separate secondary endpoint
- RPO delay to secondary can be queried

Access Storage

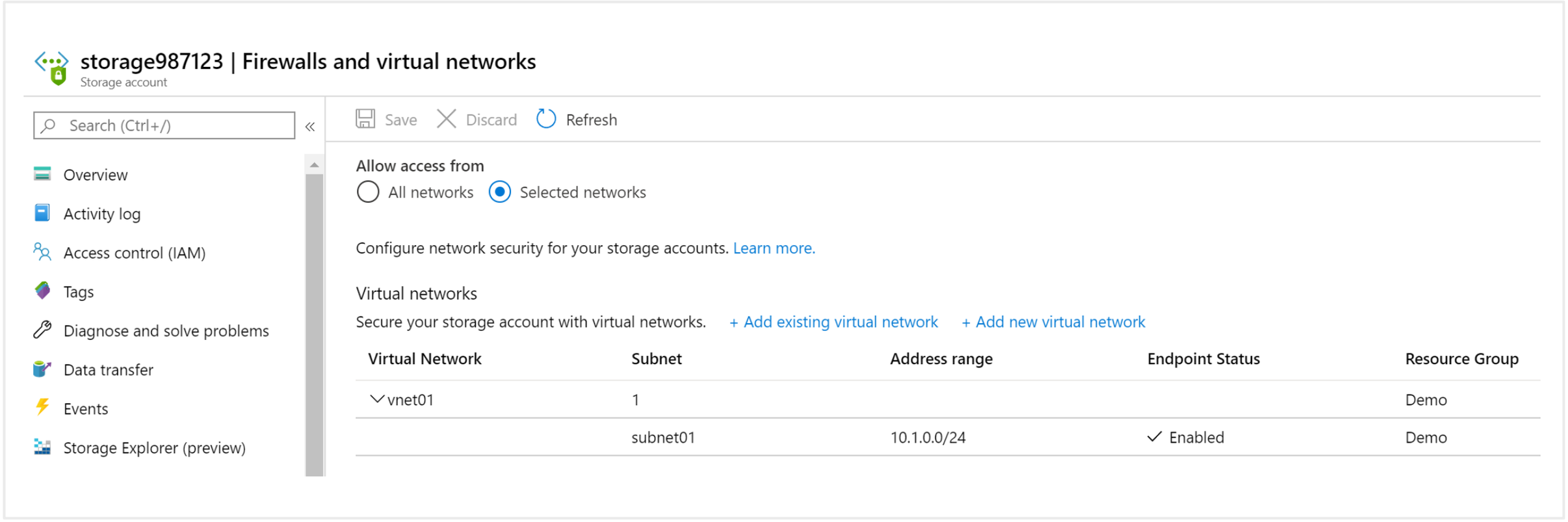
Every object has a unique URL address – based on account name and storage type

Container service: https://*mystorageaccount*.blob.core.windows.net
Table service: https://*mystorageaccount*.table.core.windows.net
Queue service: https://*mystorageaccount*.queue.core.windows.net
File service: https://*mystorageaccount*.file.core.windows.net

If you prefer you can configure a custom domain name

CNAME record	Target
blobs.contoso.com	contosoblobs.blob.core.windows.net

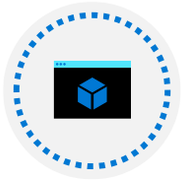
Secure Storage Endpoints



Firewalls and Virtual Networks restrict access to the Storage Account from specific Subnets on Virtual Networks or public IP's

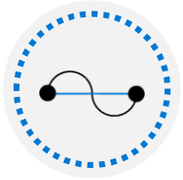
Subnets and Virtual Networks must exist in the same Azure Region or Region Pair as the Storage Account

Demonstration – Secure a Storage Endpoint



Create a storage account

Upload a file to the storage account



Create a subnet service endpoint

Secure the storage to the service endpoint

Test the storage endpoint

Summary and Resources – Configure Storage Accounts

Knowledge Check Questions

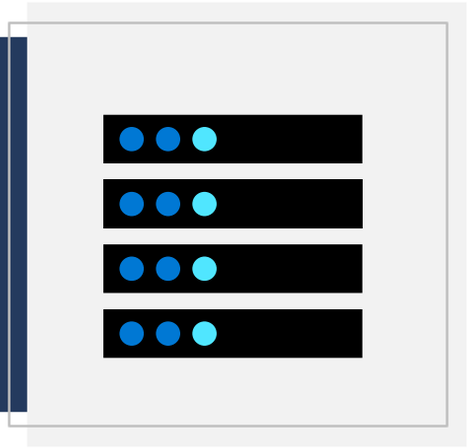
Microsoft Learn Modules (docs.microsoft.com/Learn)

Create an Azure Storage account (Sandbox)

Provide disaster recovery by replicating storage data across regions and failing over to a secondary location

A *sandbox* indicates a hands-on exercise.

Configure Blob Storage



Configure Blob Storage Introduction

Implement Blob Storage

Create Blob Containers

Create Blob Access Tiers

Add Blob Lifecycle Management Rules

Determine Blob Object Replication

Demonstration – Blob Storage

Summary and Resources

* Upload Blobs and Determine Storage Pricing are not covered.

Implement Blob Storage

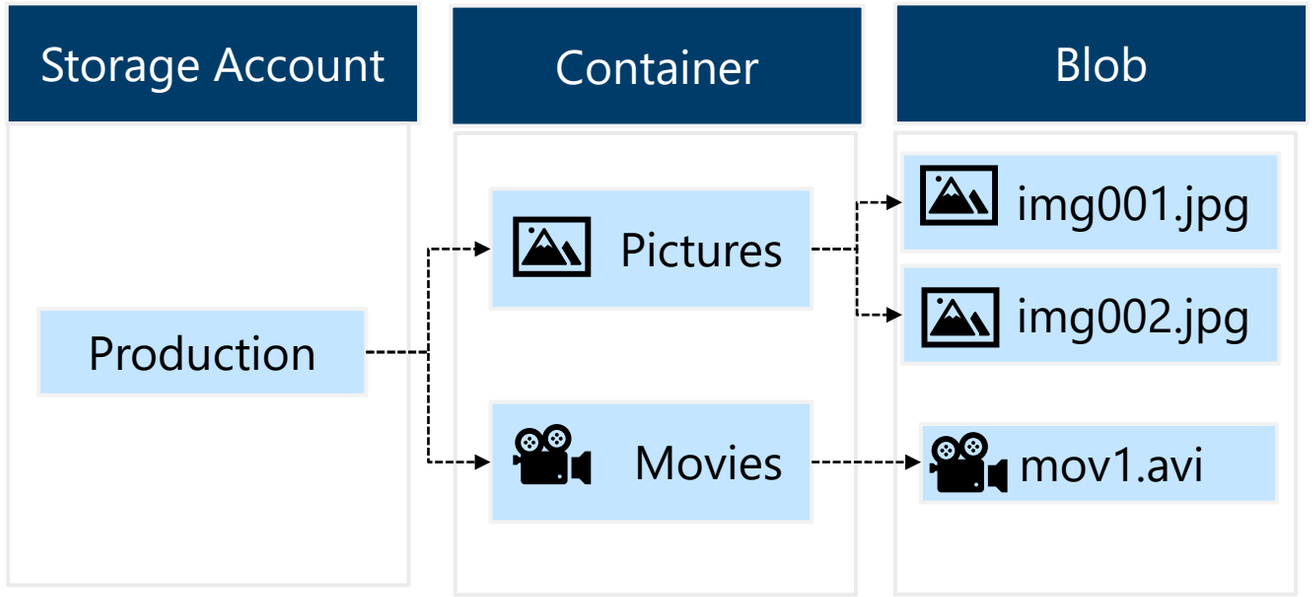
Stores unstructured data in the cloud

Can store any type of text or binary data

Also referred to as *object storage*

Common uses:

- Serving images or documents directly to a browser
- Storing files for distributed access
- Streaming video and audio
- Storing data for backup and restore, disaster recovery, archiving
- Storing data for analysis by an on-premises or Azure-hosted service



Create Blob Containers

All blobs must be in a container

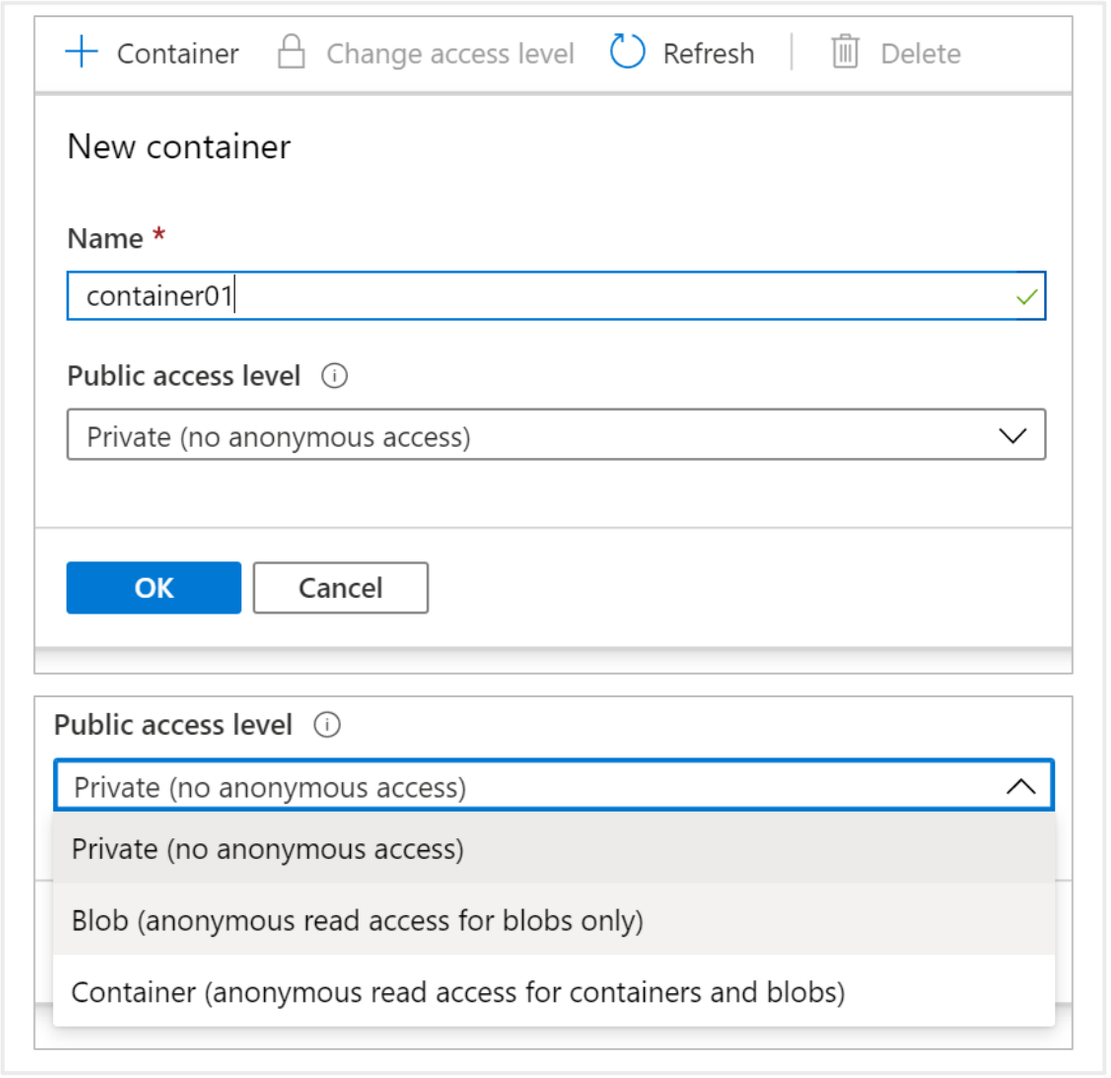
Accounts have unlimited containers

Containers can have unlimited blobs

Private blobs – no anonymous access

Blob access – anonymous public read access for blobs only

Container access – anonymous public read and list access to the entire container, including the blobs

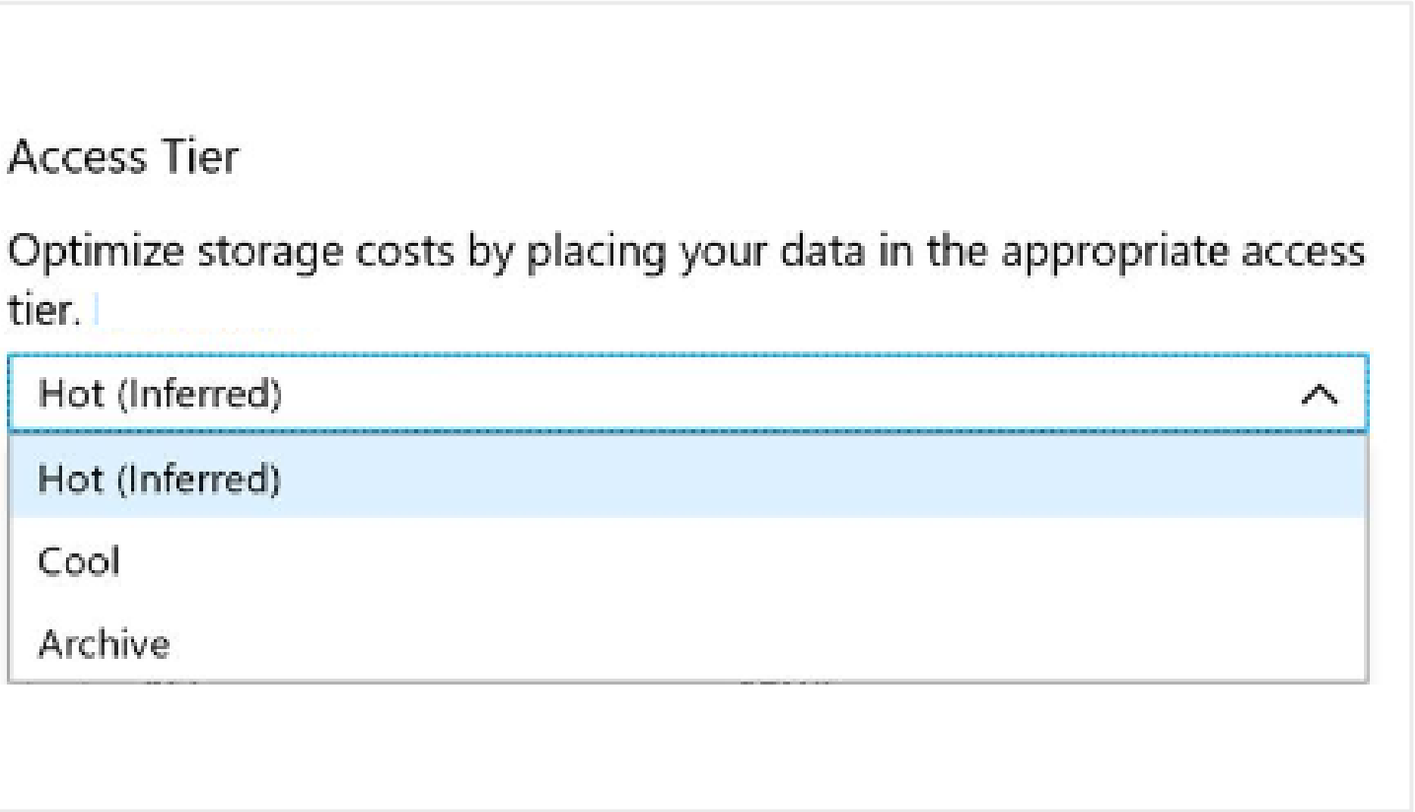


Create Blob Access Tiers

Hot tier – Optimized for frequent access of objects in the storage account

Cool tier – Optimized for storing large amounts of data that is infrequently accessed and stored for at least 30 days

Archive – Optimized for data that can tolerate several hours of retrieval latency and will remain in the Archive tier for at least 180 days



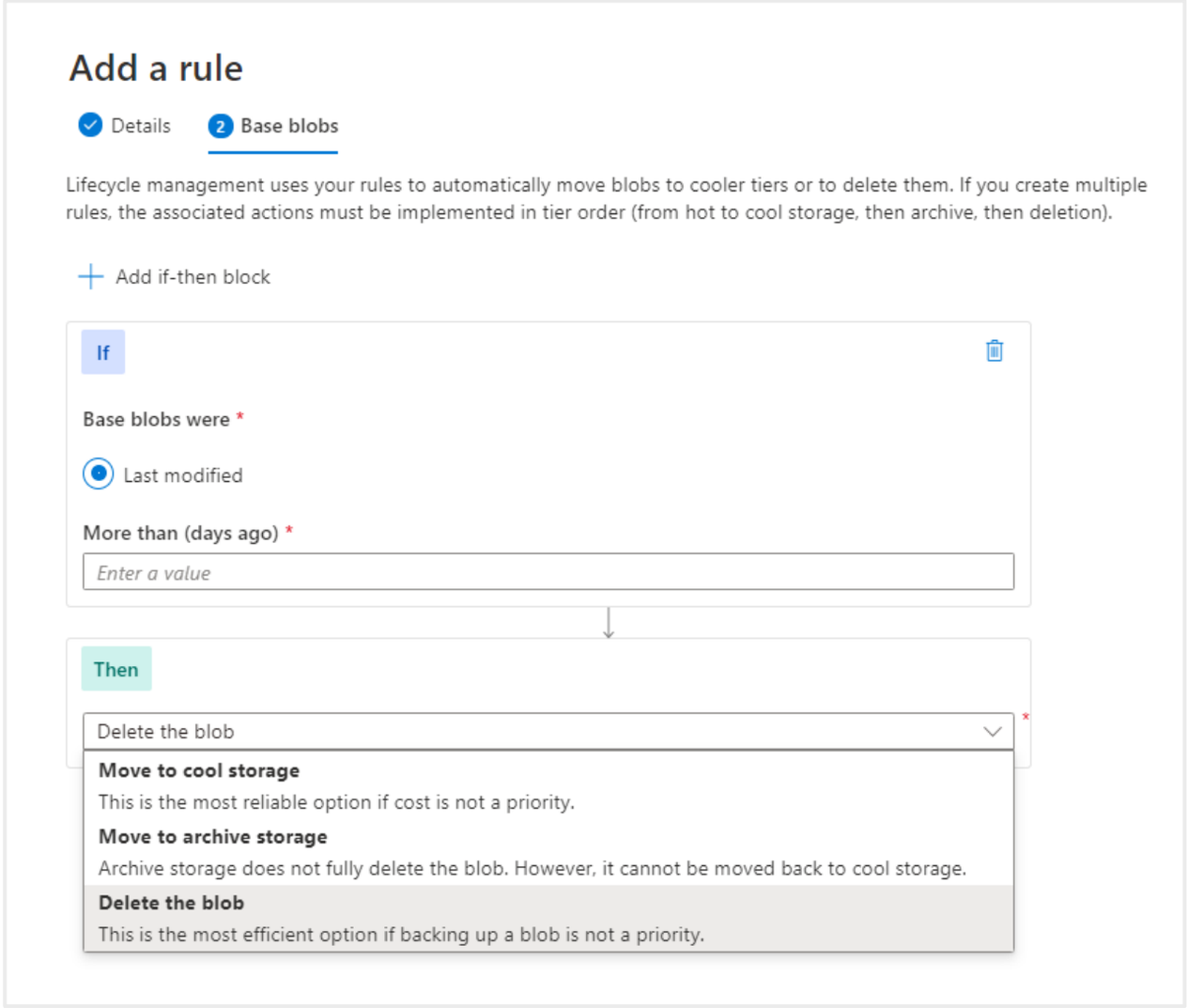
You can switch between these access tiers at any time

Add Blob Lifecycle Management Rules

Transitioning of blobs to a cooler storage tier to optimize for performance and cost

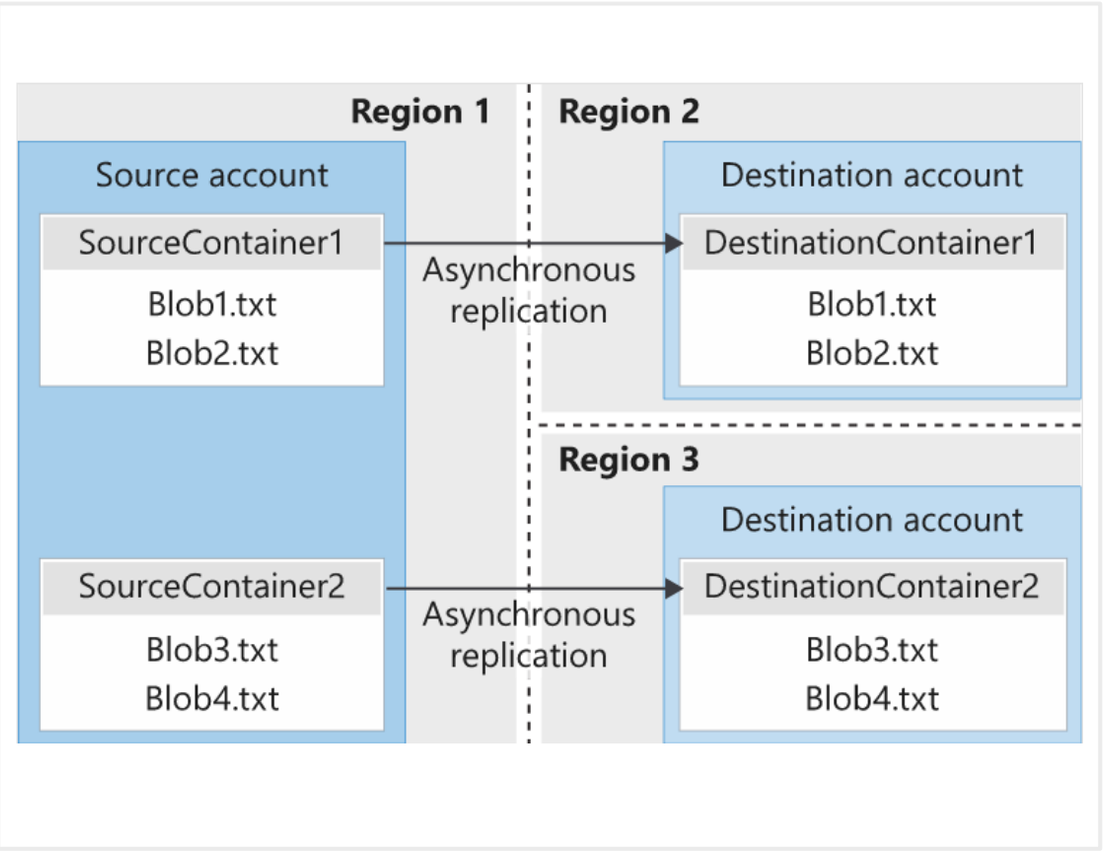
Delete blobs at the end of their lifecycle

Apply rules to filtered paths in the Storage Account



Determine Blob Object Replication

- Asynchronous to any other Region
- Minimizes latency for read requests
- Increases efficiency for compute workloads
- Optimizes data distribution
- Optimizes costs



Demonstration – Blob Storage

Create a container

Upload a block blob

Download a block blob

Summary and Resources - Configure Blob Storage

Knowledge Check Questions

Microsoft Learn Modules (docs.microsoft.com/Learn)

Optimize storage performance and costs using Azure Blob storage tiers (Sandbox)

Gather metrics from your Azure Blob Storage containers (Sandbox)

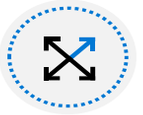
A *sandbox* indicates a hands-on exercise.

Configure Storage Security

Configure Storage Security Introduction

Review Storage Security Strategies

Create Shared Access Signatures



Identify URI and SAS Parameters

Demonstration – SAS (Portal)

Determine Storage Service Encryption

Create Customer Managed Keys

Apply Storage Security Best Practices

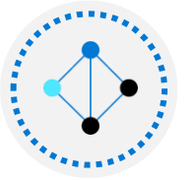
Summary and Resources

Review Storage Security Strategies

Storage Service Encryption

Shared Access Signatures – delegated access

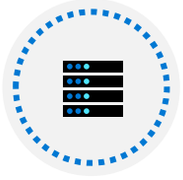
Authentication with Azure AD and RBAC



Shared Key – encrypted signature string

Client-side encryption, HTTPS, and SMB 3.0 for data in transit

Anonymous access to containers and blobs



Azure disk encryption

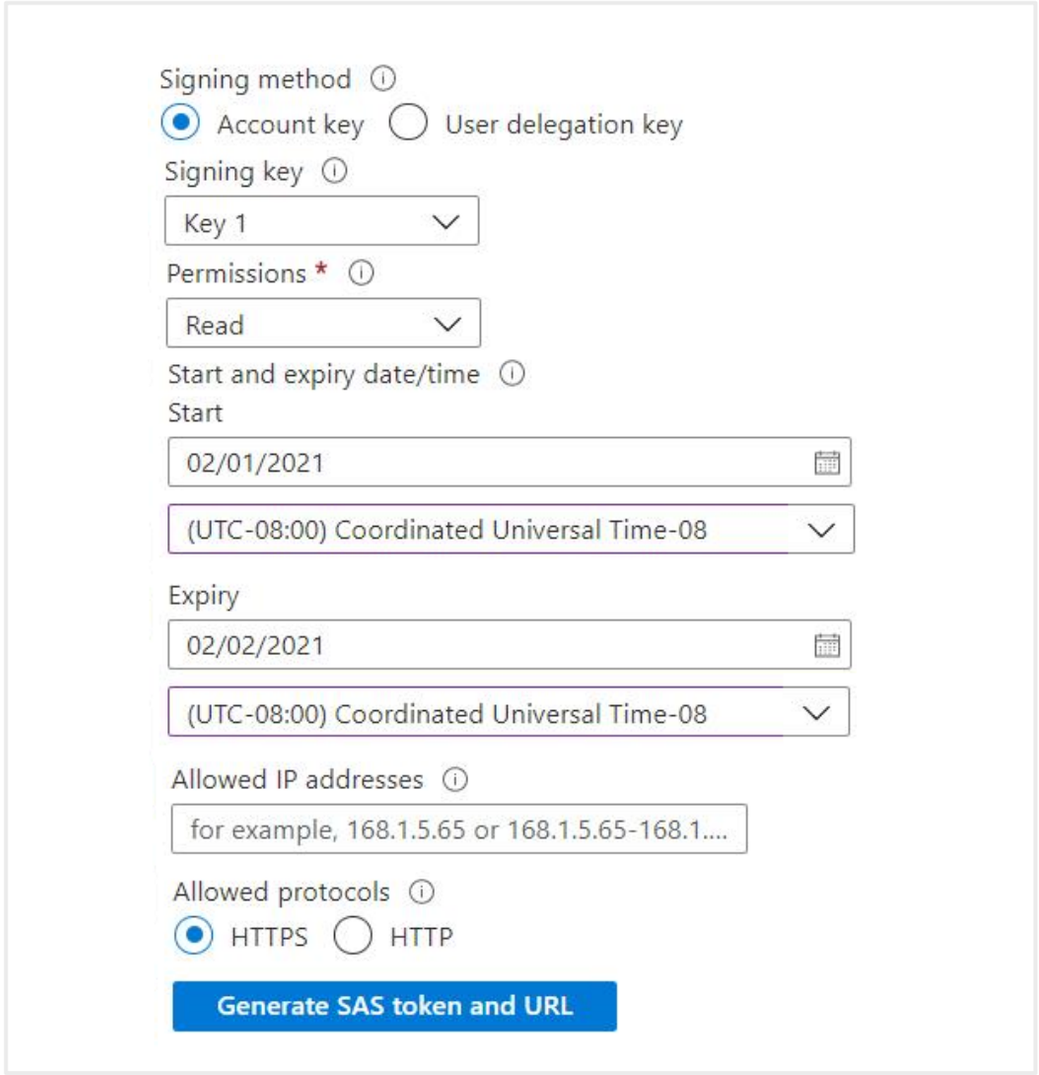
Create Shared Access Signatures

Provides delegated access to resources

Grants access to clients without sharing your storage account keys

The account SAS delegates access to resources in one or more of the storage services

The service SAS delegates access to a resource in just one of the storage services



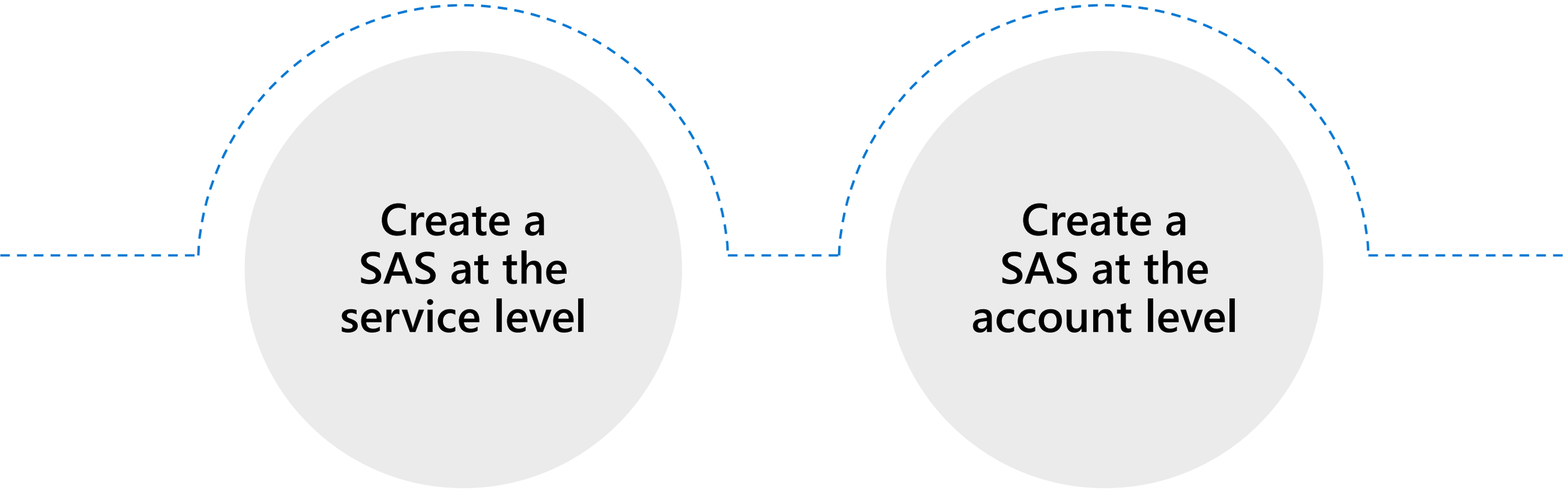
Identify URI and SAS Parameters

- A SAS is a signed URI that points to one or more storage resources
- Consists of a storage resource URI and the SAS token


```
https://myaccount.blob.core.windows.net/?sp=r&st=2020-05-
11T18:31:43Z&se=2020-05-12T02:31:43Z&spr=https&sv=2019-10-
10&sr=b&sig=jOqABJZHfUVeBQ3yVn7kWiCKlO0sxCiK1rzEchfAz8U%3D
```

Includes parameters for resource URI, storage services version, services, resource types, start time, expiry time, resource, permissions, IP range, protocol, signature

Demonstration – SAS (Portal)



Determine Storage Service Encryption

Protects your data for security and compliance

Automatically encrypts and decrypts your data

Encrypted through 256-bit AES encryption

Is enabled for all new and existing storage accounts and cannot be disabled

Is transparent to users

You can use your own key (next topic)

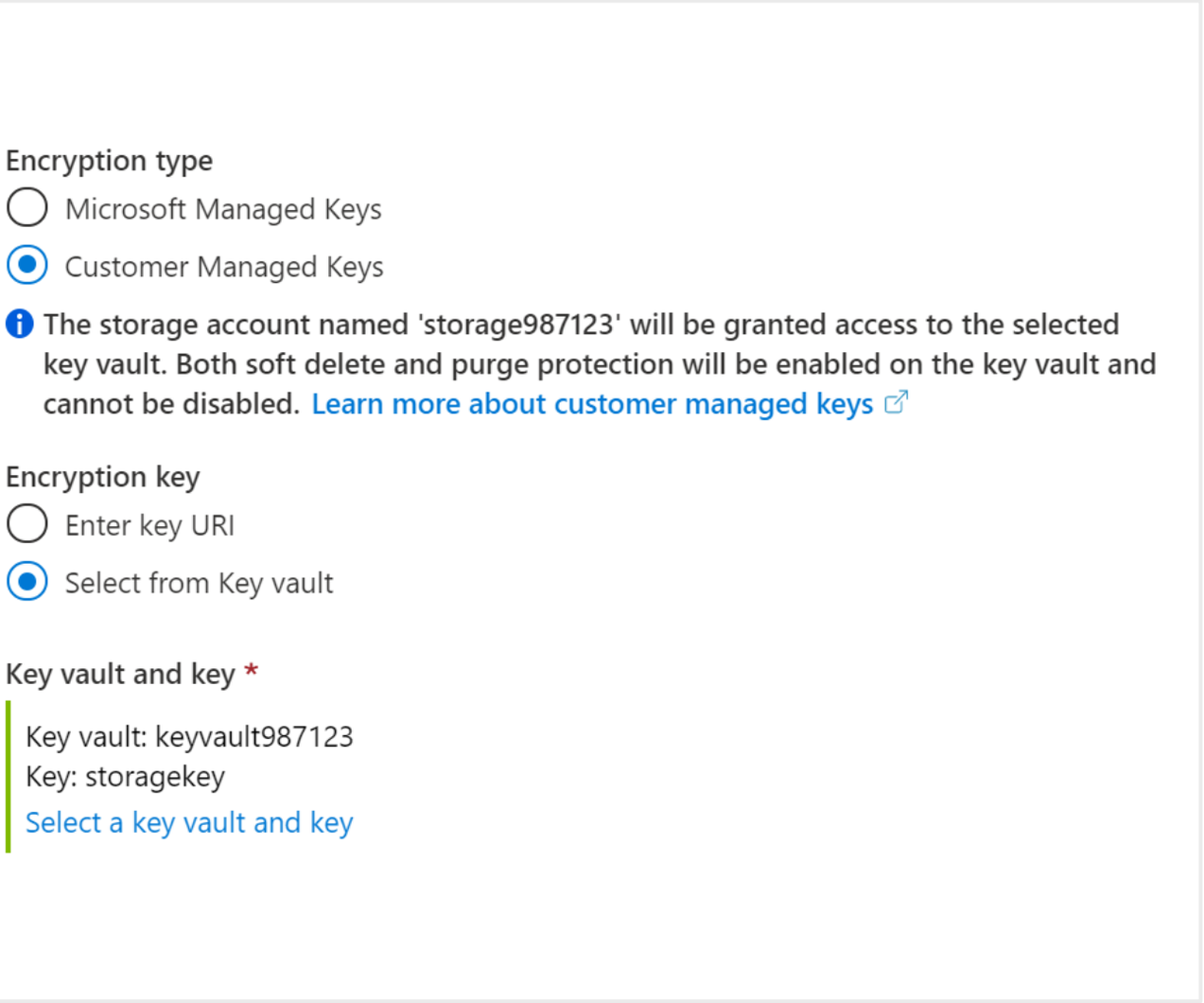
Create Customer Managed Keys

Use the Azure Key Vault to manage your encryption keys

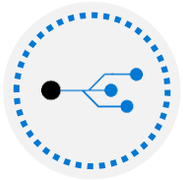
Create your own encryption keys and store them in a key vault

Use Azure Key Vault's APIs to generate encryption keys

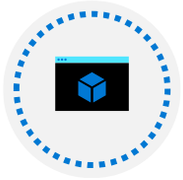
Custom keys give you more flexibility and control



Apply Storage Security Best Practices



Always use HTTPS to create or distribute an SAS

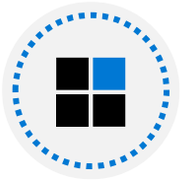


Reference stored access policies where possible

Use near-term expiration times on an ad hoc SAS

Use Storage Analytics to monitor your application

Be careful with SAS start time



Be specific with the resource to be accessed

Understand that your account will be billed for any usage

Validate data written using SAS

Don't assume SAS is always the correct choice

Summary and Resources - Configure Storage Security

Knowledge Check Questions

Microsoft Learn Modules (docs.microsoft.com/Learn)

- Secure your Azure Storage account
- Control access to Azure Storage with shared access signatures (Sandbox)
- Implement storage security

A *sandbox* indicates a hands-on exercise.

Configure Azure Files and File Sync

Configure Azure Files and File Sync Introduction

Compare Files to Blobs

Manage File Shares

Create File Share Snapshots

Demonstration – File Shares

Implement Azure File Sync

Identify Azure File Sync Components

Deploy File Sync

Configure Storage with Tools (summary only)

Summary and Resources

Compare Files to Blobs

Feature	Description	When to use
Azure Files	SMB interface, client libraries, and a REST interface that allows access from anywhere to stored files	• Lift and shift an application to the cloud • Store shared data across multiple virtual machines • Store development and debugging tools that need to be accessed from many virtual machines
Azure Blobs	Client libraries and a REST interface that allows unstructured data (flat namespace) to be stored and accessed at a massive scale in block blobs	• Support streaming and random-access scenarios • Access application data from anywhere

Manage File Shares

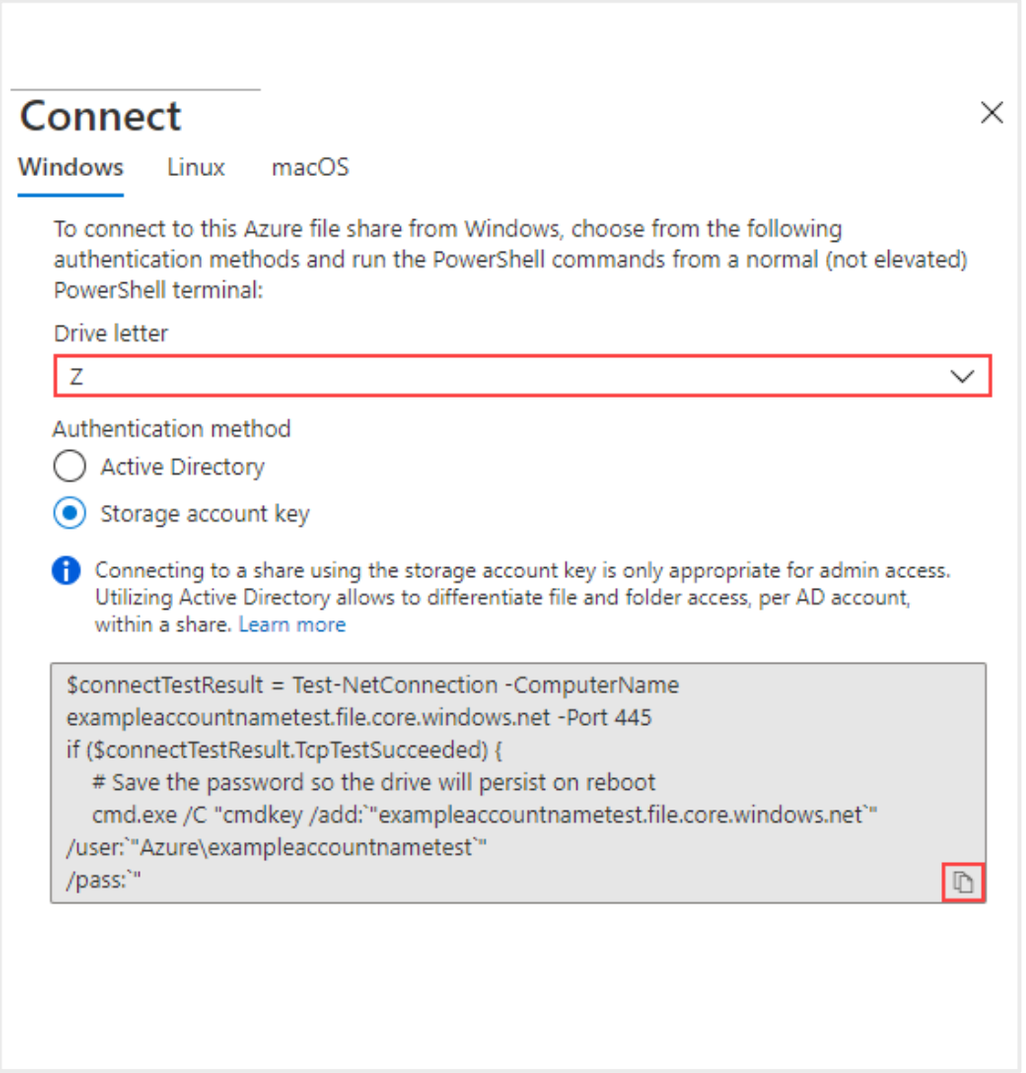
File share quotas

Windows – ensure port 445 is open

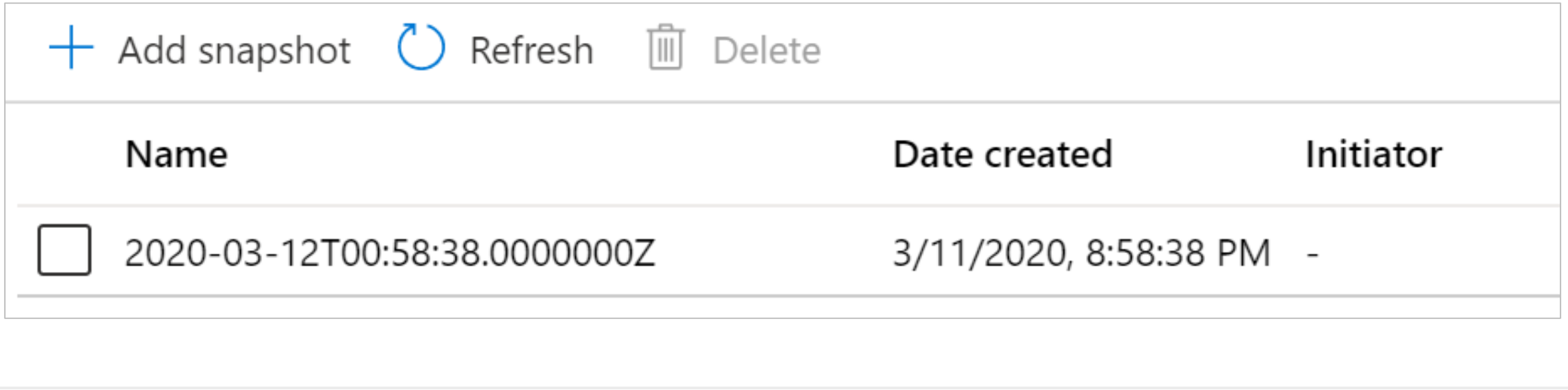
Linux – mount the drive

MacOS – mount the drive

Secure transfer required – SMB 3.0 encryption



Create File Share Snapshots



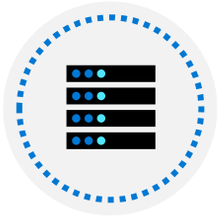
Incremental snapshot that captures the share state at a point in time

Is read-only copy of your data

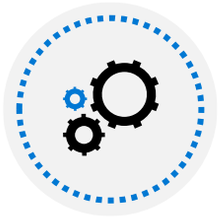
Snapshot at the file share level, and restore at the file level

- Protection against application error and data corruption
- Protection against accidental deletions or unintended changes
- General backup purposes

Demonstration – File Shares



Create a file share and upload a file



Manage snapshots

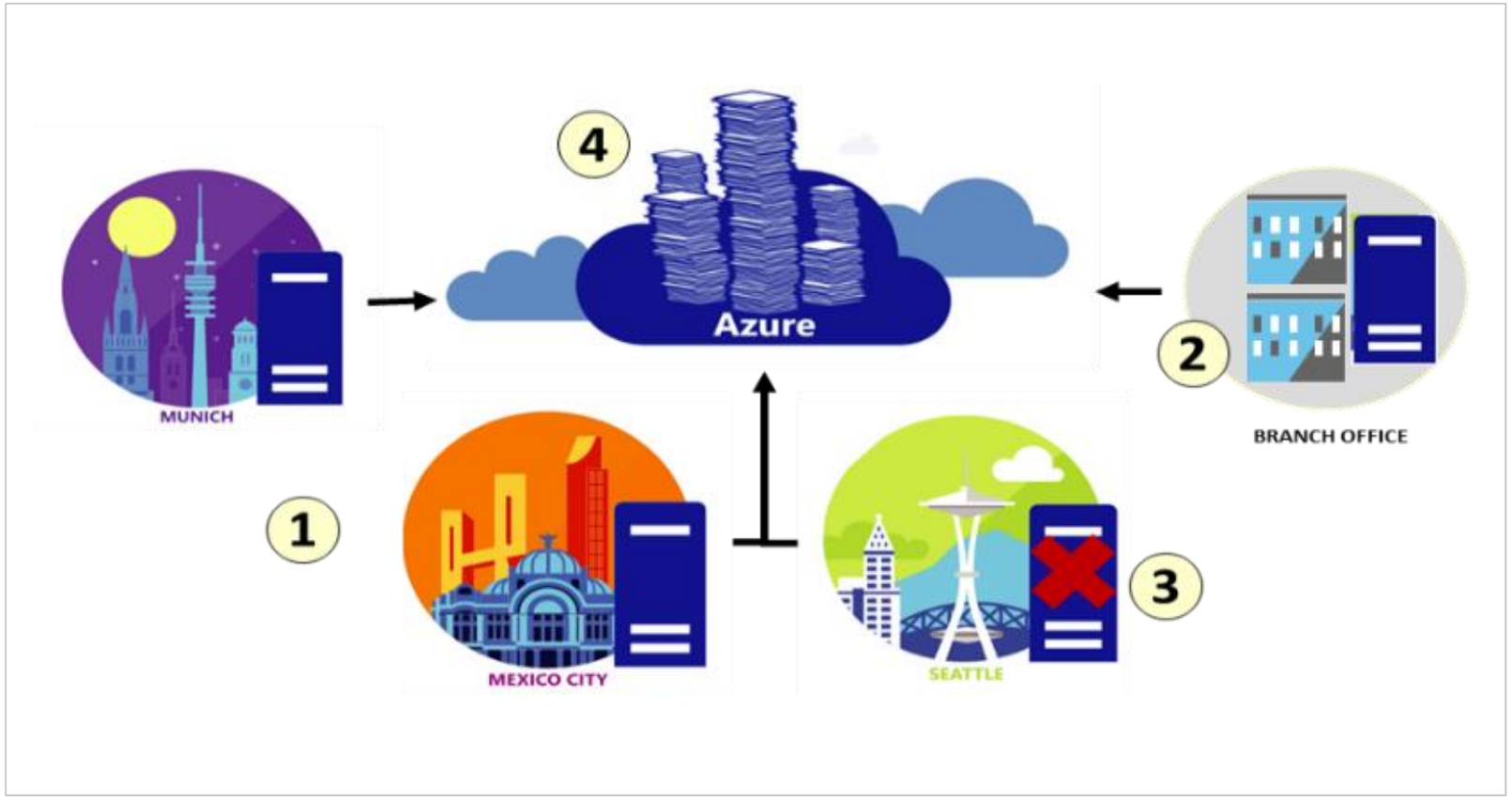
Create a file share (PowerShell - optional)

Mount a file share (PowerShell - optional)

Implement Azure File Sync

Centralize your organization's file shares in Azure Files, while keeping the flexibility, performance, and compatibility of an on-premises file server

1. Lift and shift
2. Branch Office backups
3. Backup and Disaster Recovery
4. File Archiving



Identify File Sync Components

The **Storage Sync Service** is the top-level resource

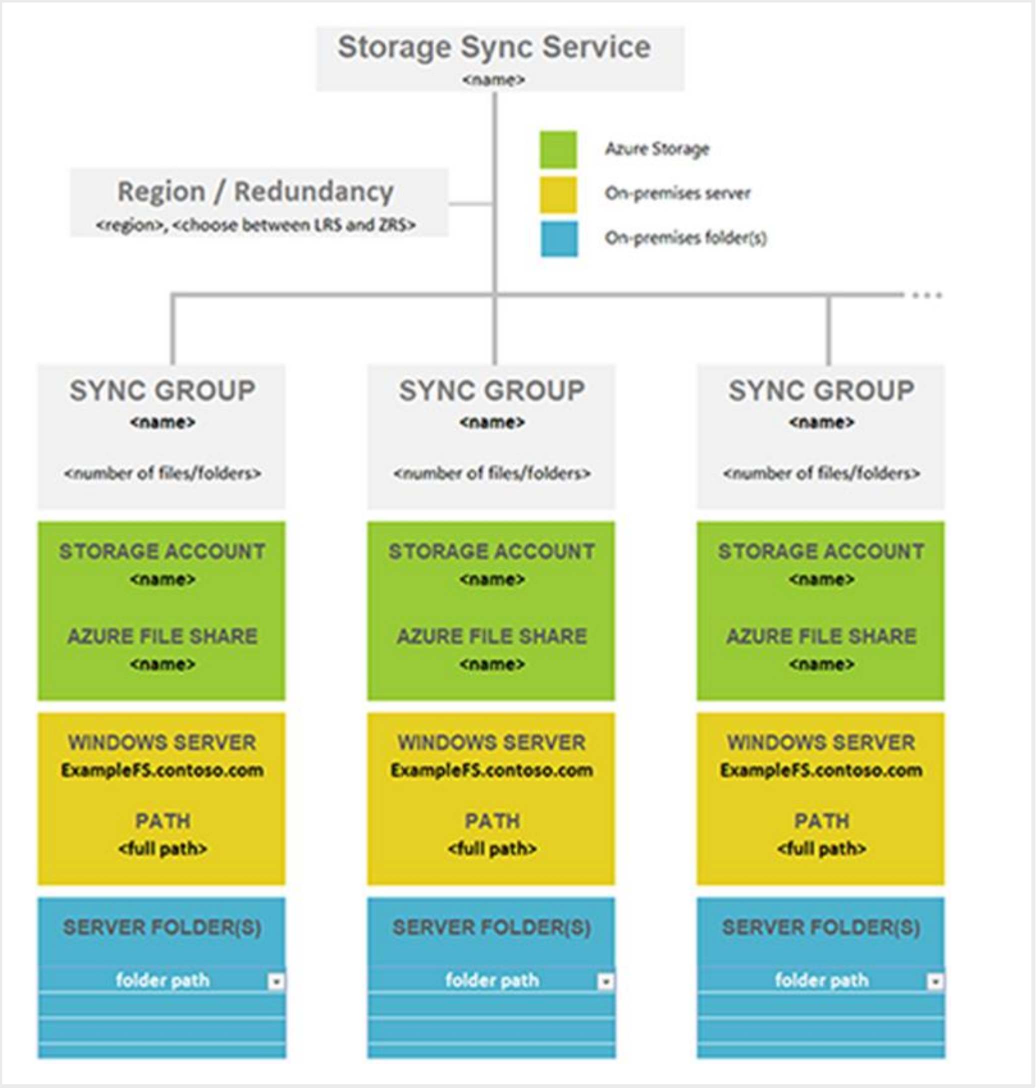
The **registered server** object represents a trust relationship between your server (or cluster) and the Storage Sync Service

The **Azure File Sync agent** is a downloadable package that enables Windows Server to be synced with an Azure file share

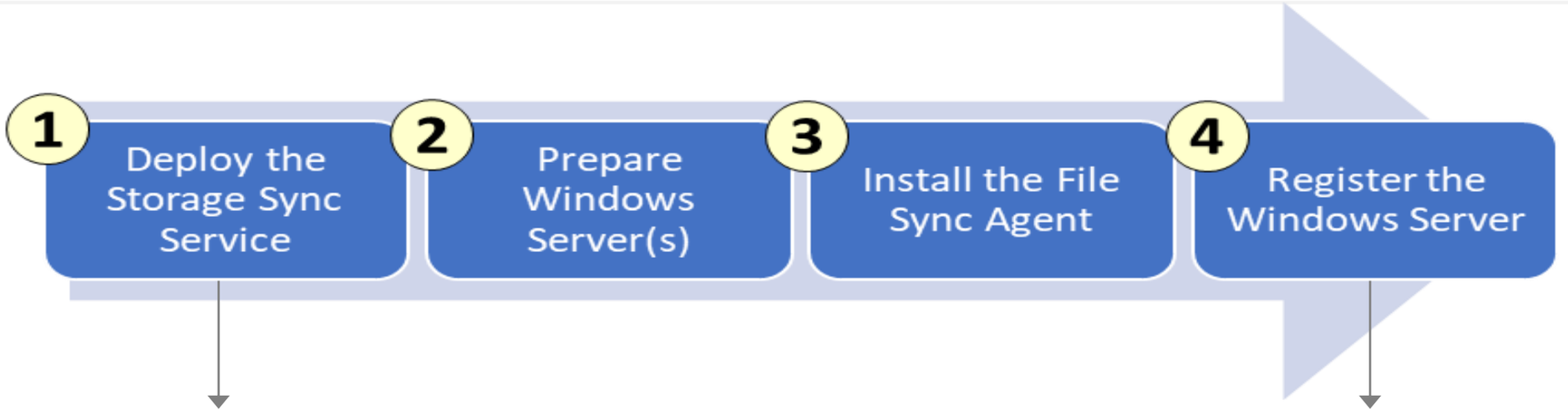
A **server endpoint** represents a specific location on a registered server, such as a folder

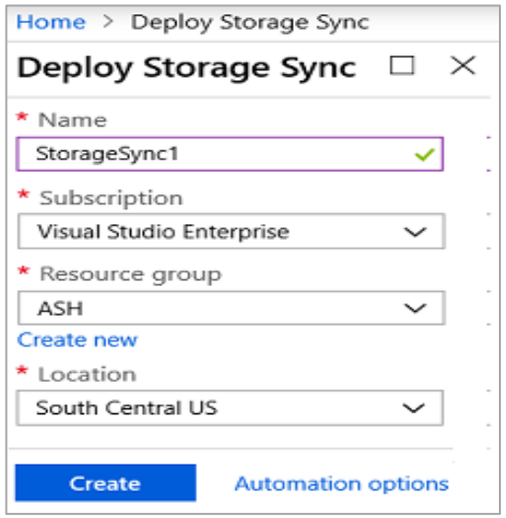
A **cloud endpoint** is an Azure file share

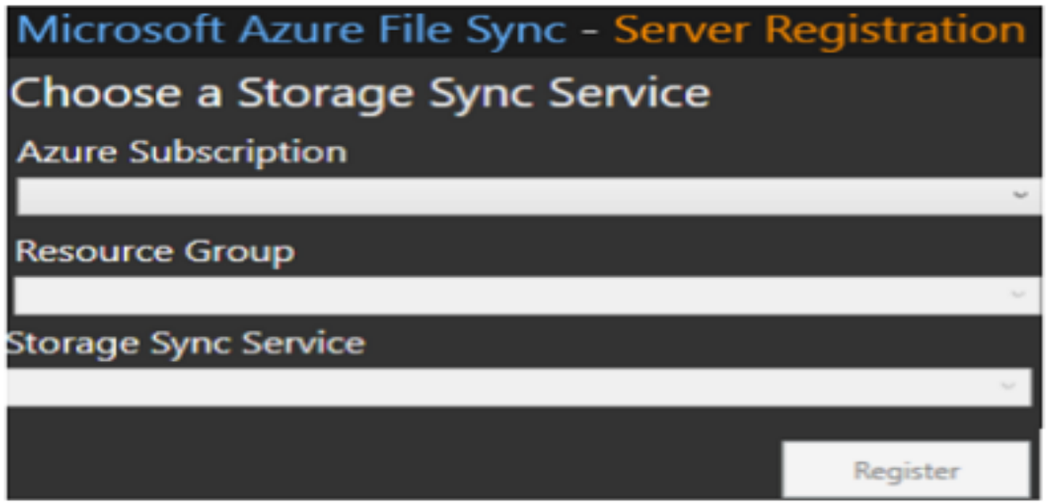
A **sync group** defines which files are kept in sync



Setup File Sync

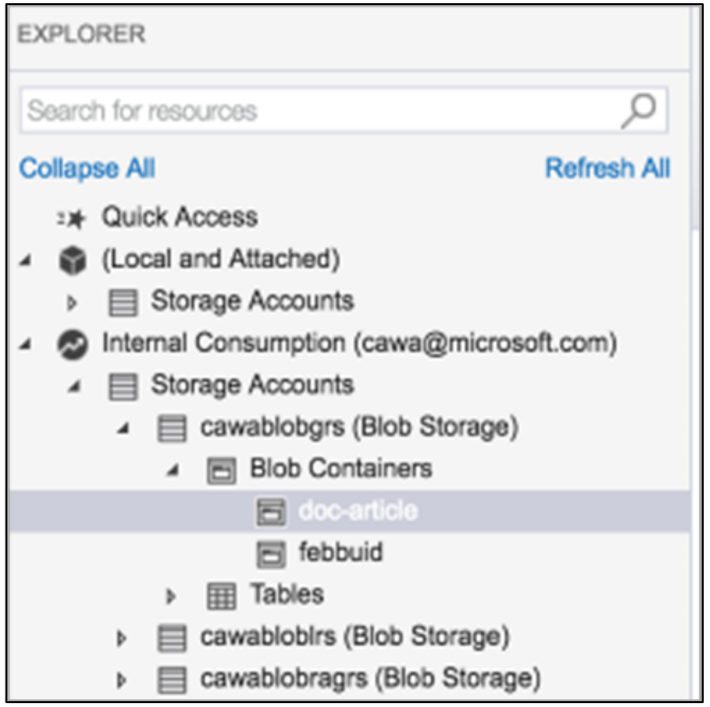




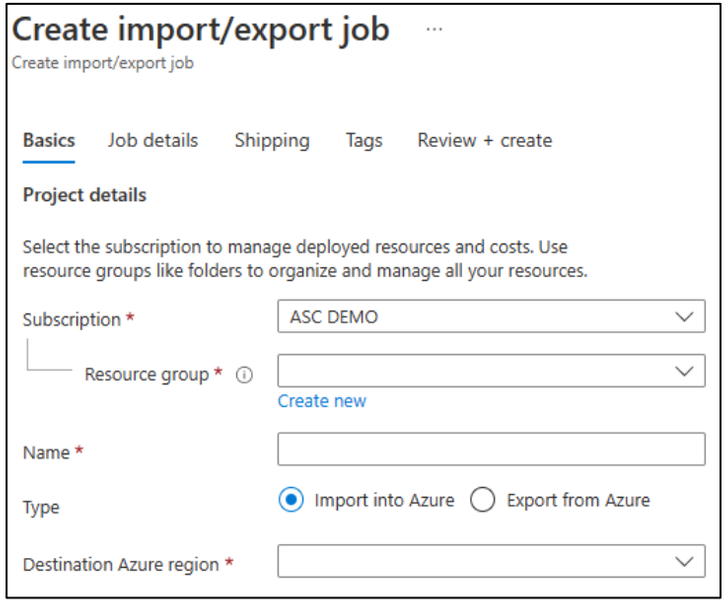


Configure Storage with Tools

Azure Storage Explorer



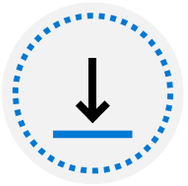
The Import and Export service



AzCopy

```
azcopy copy [source] [destination] [flags]
```

Demonstration – Storage Explorer (optional)

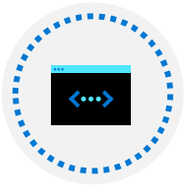


Download and install Storage Explorer

Connect to an Azure subscription

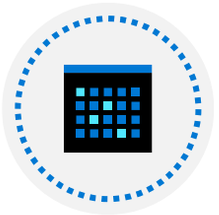
Attach an Azure Storage account

Generate a SAS connection string for the account you want to share



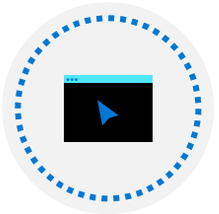
Attach to a storage account by using a SAS Connection string

Demonstration – AzCopy (optional)

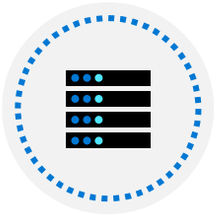


Install the AzCopy tool

Explore the help



Download a blob from Blob storage to the file system



Upload files to Azure blob storage

Summary and Resources - Configure Azure Files and File Sync

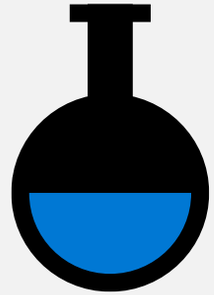
Knowledge Check Questions

Microsoft Learn Modules (docs.microsoft.com/Learn)

- Extend your on-premises file share capacity using Azure File Sync
- Implement a hybrid file server infrastructure
- Upload, download, and manage data with Azure Storage Explorer (Sandbox)
- Export large amounts of data from Azure by using Azure Import/Export
- Copy and move blobs from one container or storage account to another from the command line and in code (Sandbox)

A *sandbox* indicates a hands-on exercise.

Lab 07 – Manage Azure Storage



Lab 07 – Manage Azure Storage

Lab scenario

You need to evaluate the use of Azure Storage for storing files residing currently in on-premises data stores. While many of these files are not accessed frequently, there are some exceptions. You would like to minimize cost of storage by placing less frequently accessed files in lower-priced storage tiers. You also plan to explore different protection mechanisms that Azure Storage offers, including network access, authentication, authorization, and replication. Finally, you want to determine to what extent Azure Files service might be suitable for hosting your on-premises file shares

Objectives

Task 1:
Provision the lab environment

Task 2:
Create and configure Azure storage accounts

Task 3:
Manage blob storage

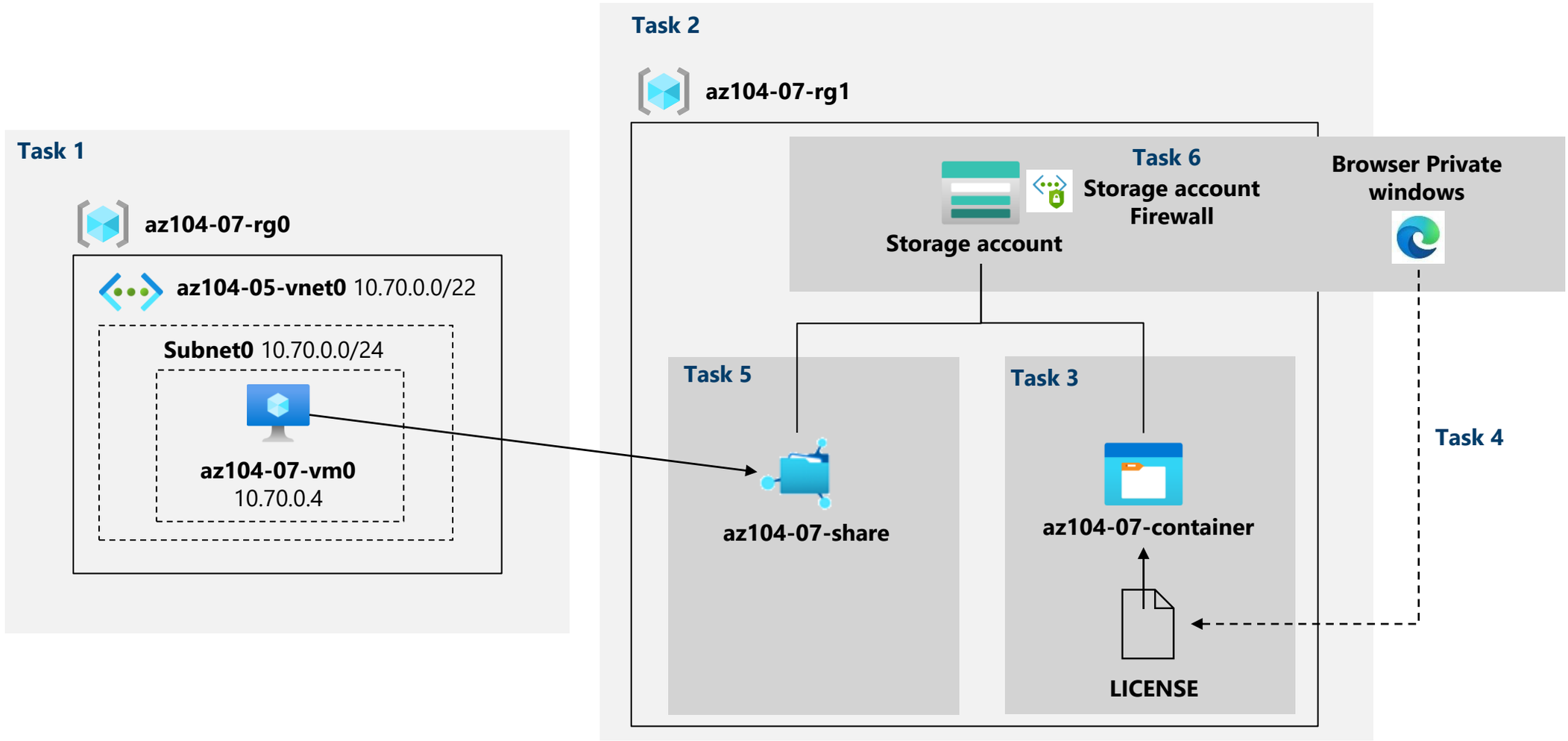
Task 4:
Manage authentication and authorization for Azure Storage

Task 5:
Create and configure an Azure Files shares

Task 6:
Manage network access for Azure Storage

Next slide for an architecture diagram

Lab 07 – Architecture diagram



End of presentation