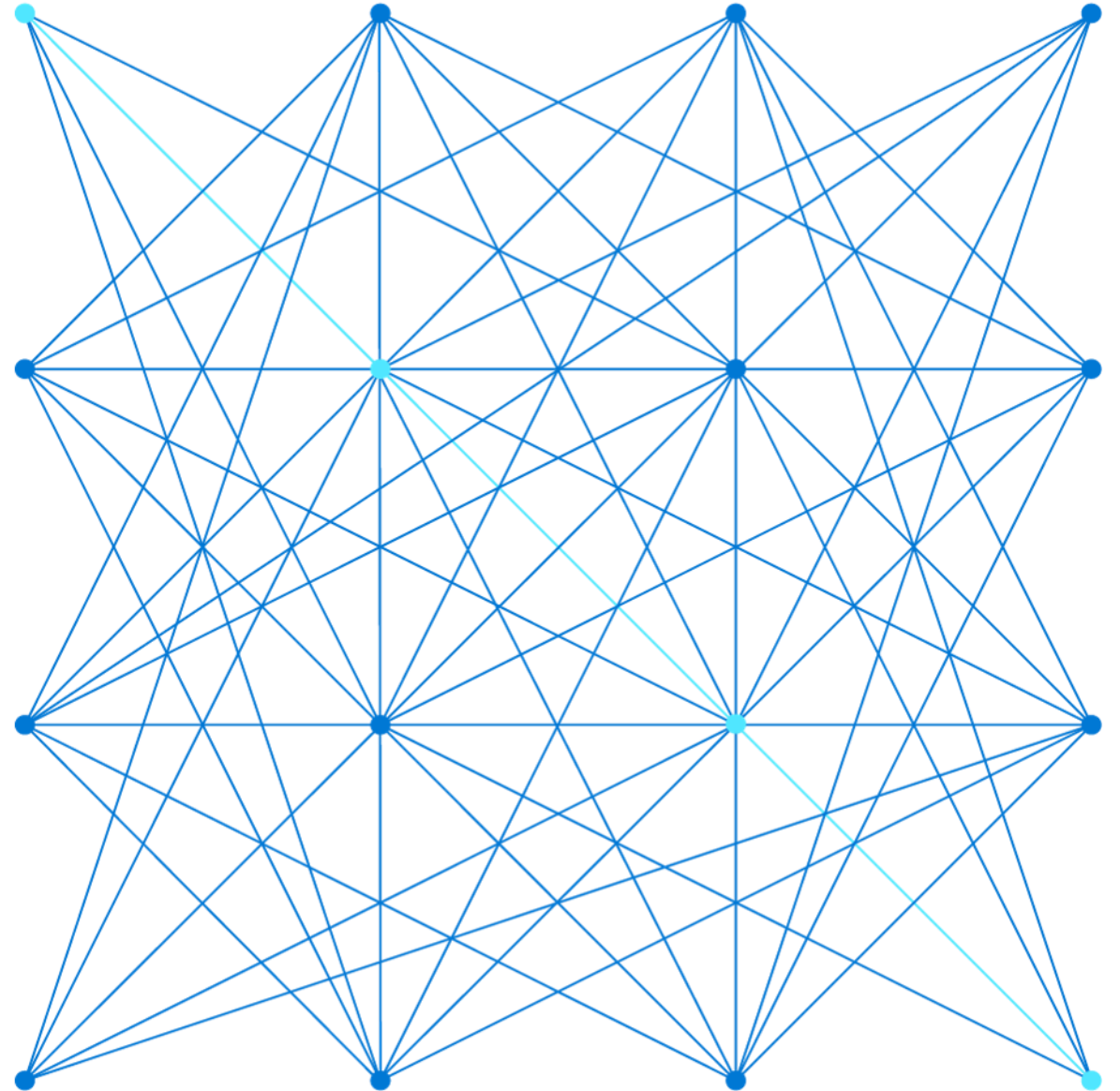


# AZ-104

## Administer Azure Storage

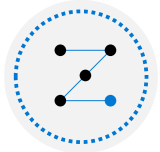


# About this course: Course Outline



01: Administer Identity

---



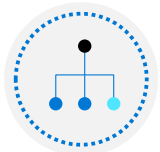
02: Administer Governance and Compliance

---



03: Administer Azure Resources

---



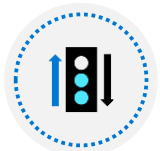
04: Administer Virtual Networking

---

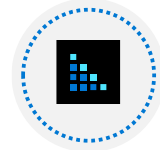


05: Administer Intersite Connectivity

---



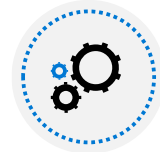
06: Administer Network Traffic Management



07: Administer Azure Storage

---

Lab 7



08: Administer Azure Virtual Machines

---

Lab 8



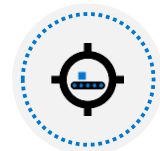
09: Administer PaaS Compute Options

---



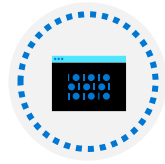
10: Administer Data Protection

---



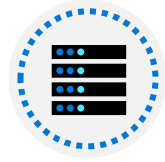
11: Administer Monitoring

# Administer Azure Storage Introduction



[Configure Storage Accounts](#)

---



[Configure Blob Storage](#)

---



[Configure Storage Security](#)

---



[Configure Azure Files](#)

---

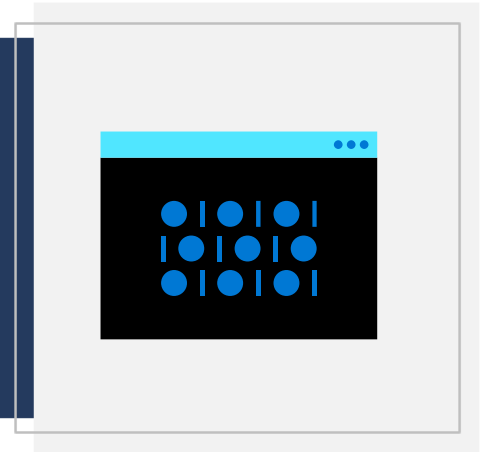


[Lab 07 – Manage Azure Storage](#)

---

LRS  
GRS  
GRS-RA

# Configure Storage Accounts



# Configure Storage Accounts Introduction



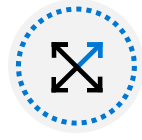
Implement Azure Storage



Explore Azure Storage Services



Determine Storage Account Kinds



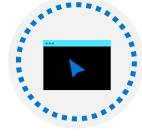
Determine Replication Strategies



Access Storage



Secure Storage Endpoints



Demonstration – Configure a storage account



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

Disk  
Ultra

Two tiers: Premium and  
Standard

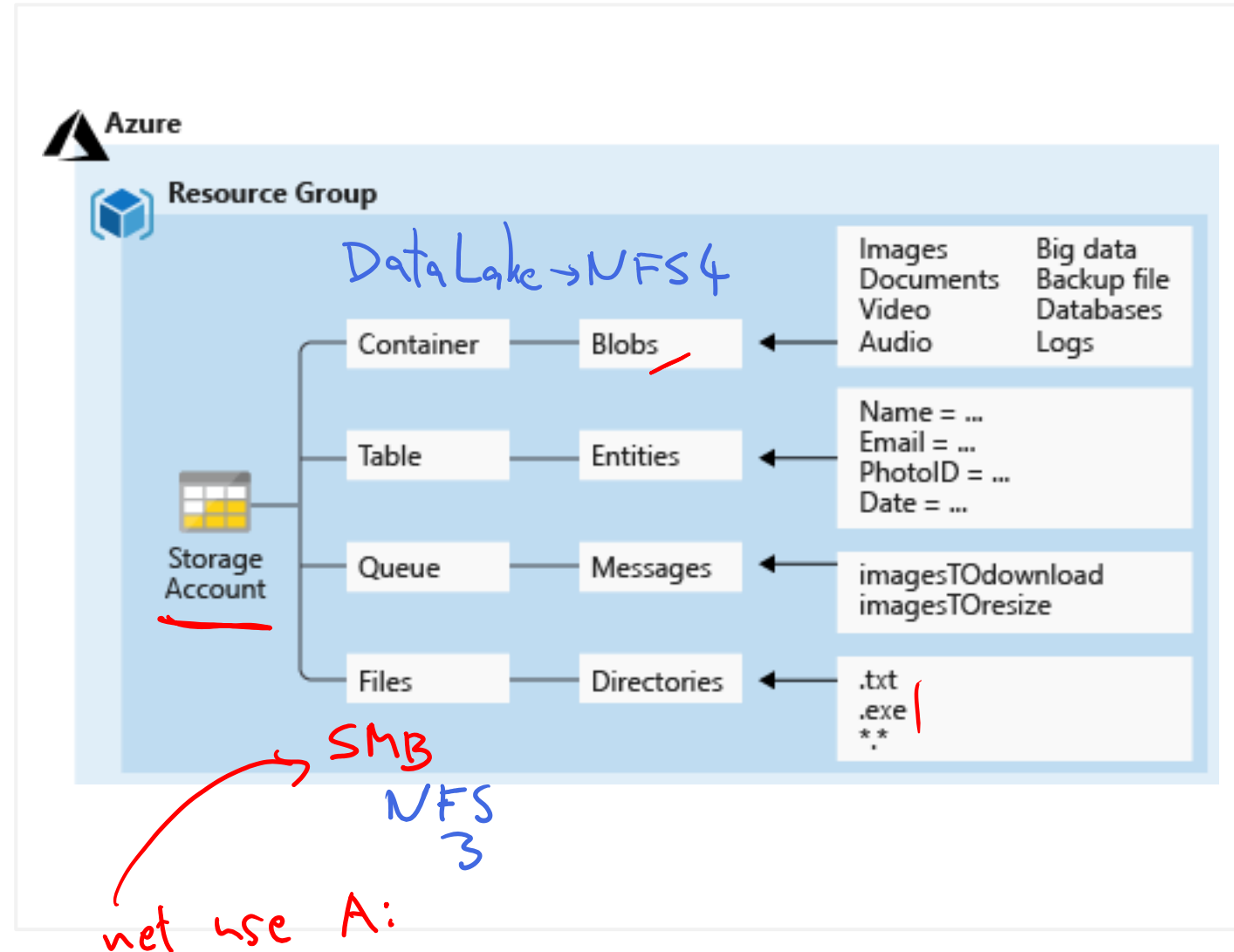
# Explore Azure Storage Services

**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

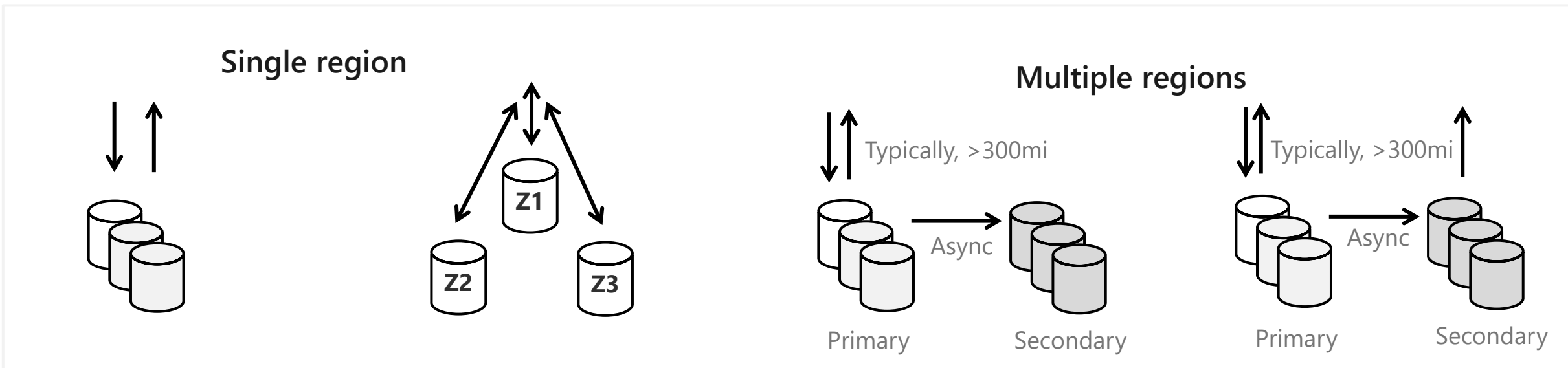
Storage Account	Recommended usage
Standard general-purpose v2	Most scenarios including Blob, File, Queue, Table, and Data Lake Storage.
Premium block blobs	Block blob scenarios with high transactions rates, or scenarios that use smaller objects or require consistently low storage latency.
Premium file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

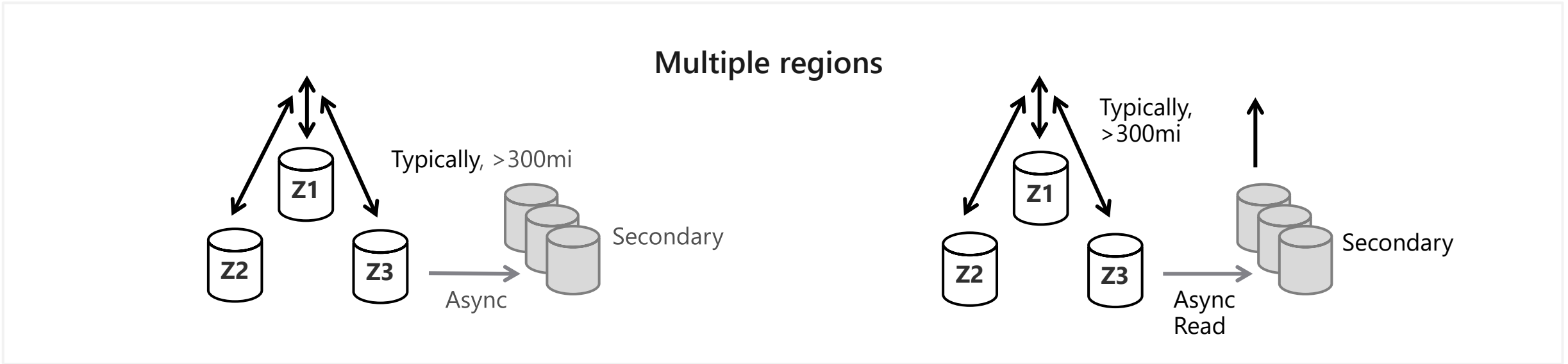
## RA-GRS

- 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

Custom domain

 Save

 Discard


 Refresh

Public network access

☒ Enabled from all networks


☐ Enabled from selected virtual networks and IP addresses

☐ Disabled

 All networks, including the internet, can access this storage account. [Learn more](#)


Network Routing

Determine how you would like to route your traffic as it travels from its source to an Azure endpoint. Microsoft routing is recommended for most customers.

Routing preference 

☒ Microsoft network routing

☐ Internet routing

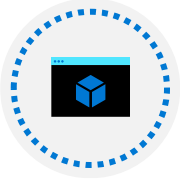
 The current combination of storage account kind, performance, replication, and location does not support network routing.

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

© Copyright Microsoft Corporation. All rights reserved.

# Demonstration – Configure a storage account



Create a storage account

---



Configure storage account settings

---

# Summary and Resources – Configure Storage Accounts

Knowledge Check Questions



Microsoft Learn Modules ([docs.microsoft.com/Learn](https://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.*

LRS



westenrope Region Location

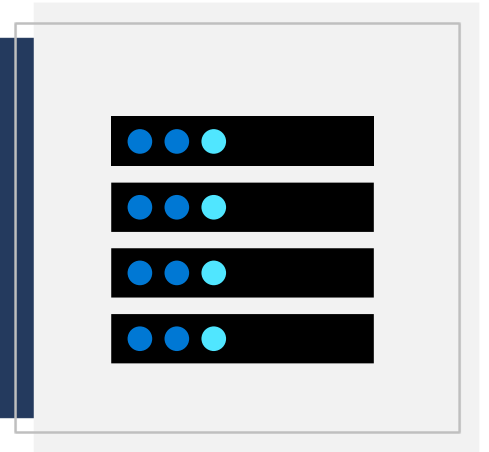
Sync.

ZRS



Avail Zones

## Configure Blob Storage



GRS



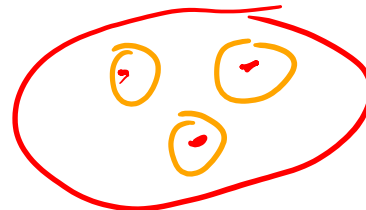
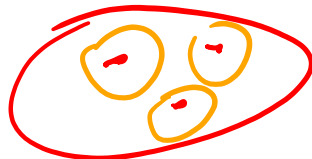
west

async.



northenrope

GZRS



GZRS - RA

# Configure Blob Storage Introduction



Implement Blob Storage



Create Blob Containers



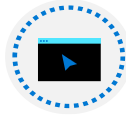
Create Blob Access Tiers



Add Blob Lifecycle Management Rules



Determine Blob Object Replication



Demonstration – Configure Blob Storage



Summary and Resources

\* Upload Blobs and Determine Storage Pricing are not covered.



AWS Dr. Vogels S3

# Implement Blob Storage

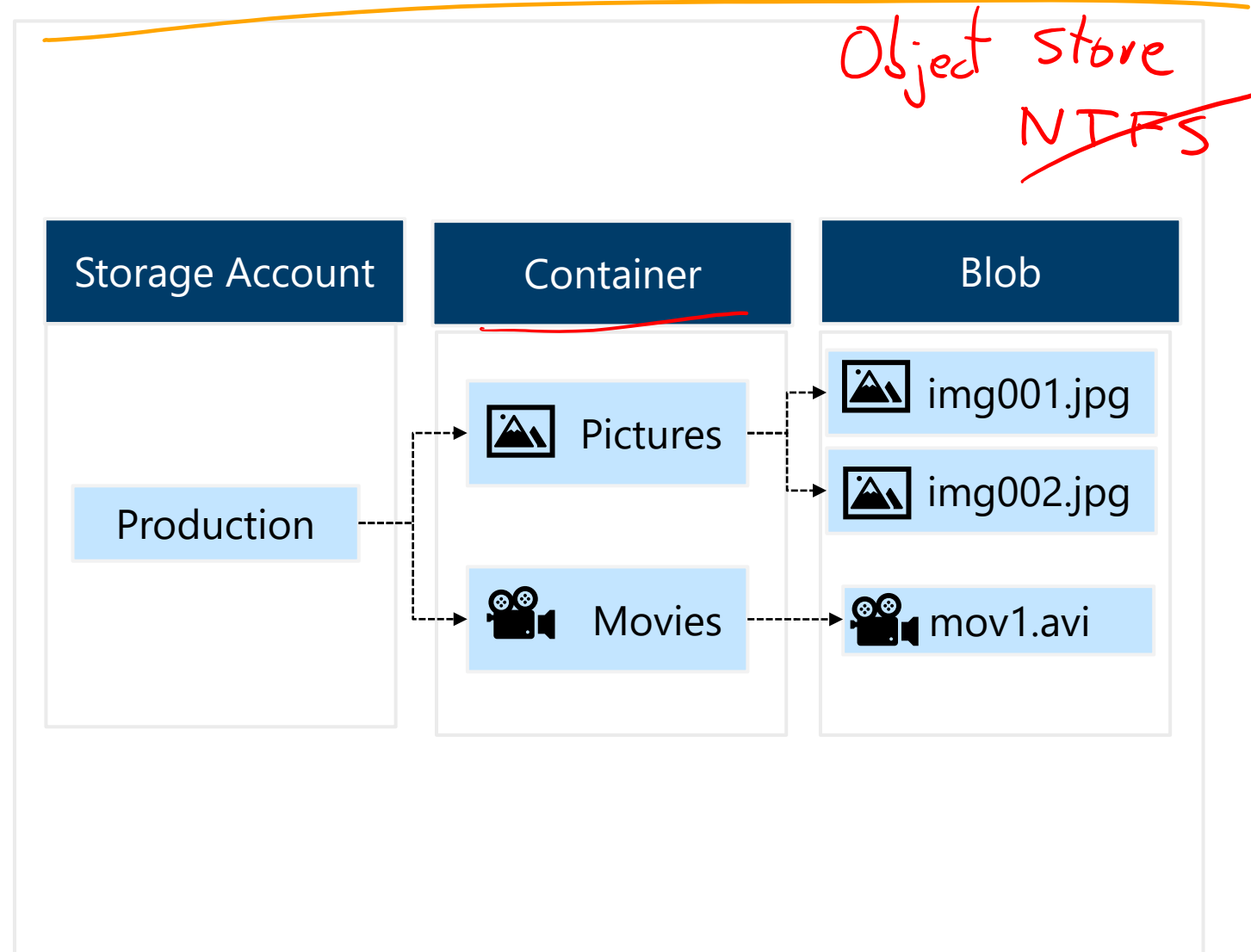
Stores unstructured data in the cloud

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

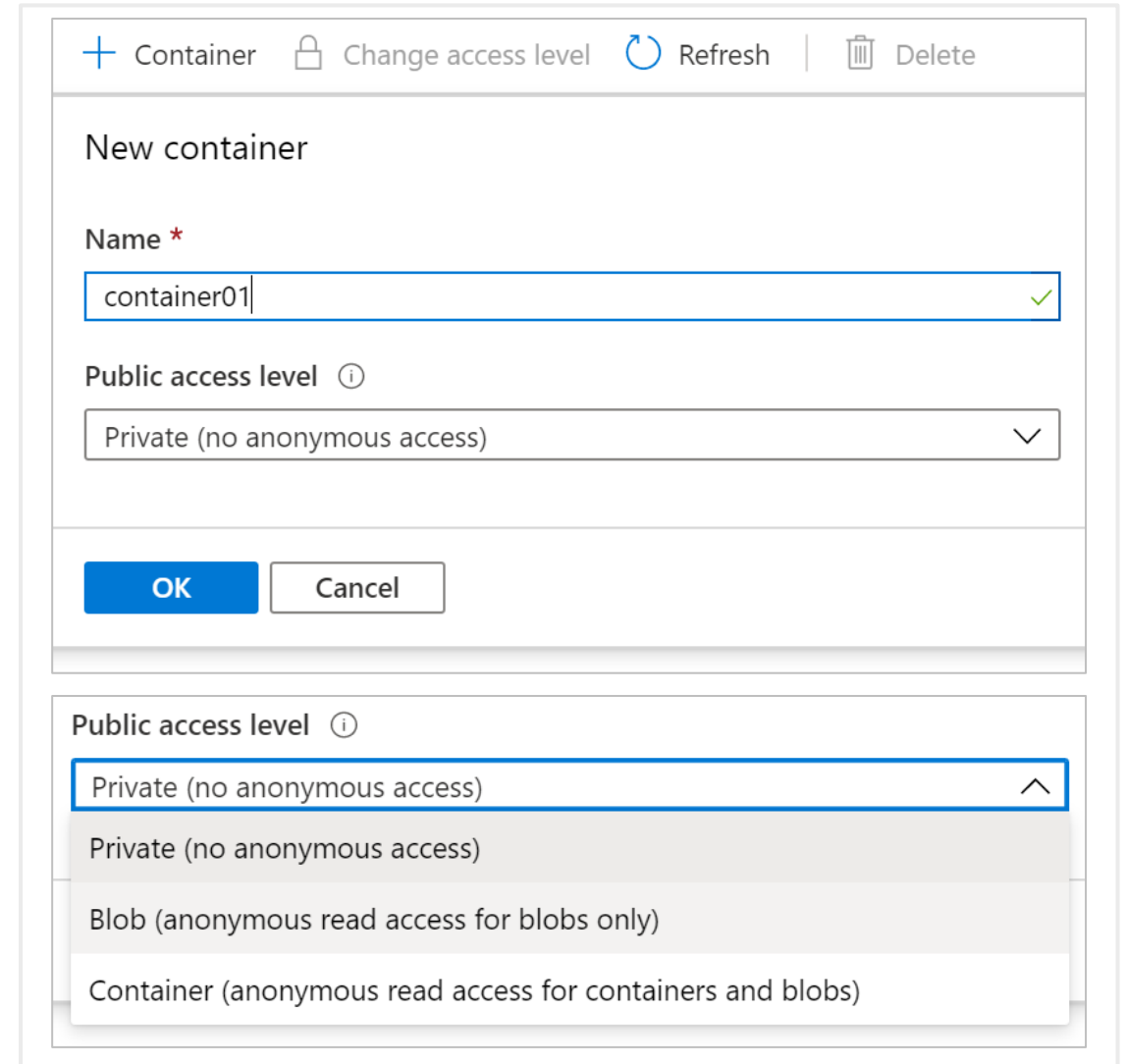
Accounts have unlimited containers

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



The screenshot shows the 'New container' dialog box in the Azure portal. At the top, there is a toolbar with icons for '+ Container', 'Change access level', 'Refresh', and 'Delete'. The main section is titled 'New container' and contains a 'Name' field with a red asterisk, where 'container01' is entered and a green checkmark is visible. Below this is a 'Public access level' dropdown menu with an information icon, currently set to 'Private (no anonymous access)'. At the bottom of the dialog are 'OK' and 'Cancel' buttons. Below the dialog, a separate view of the 'Public access level' dropdown menu is shown, listing four options: 'Private (no anonymous access)' (selected), 'Private (no anonymous access)', 'Blob (anonymous read access for blobs only)', and 'Container (anonymous read access for containers and 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

## Access Tier

Optimize storage costs by placing your data in the appropriate access tier. |

Hot (Inferred)

Hot (Inferred)

Cool

Archive



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

## Add a rule

✓ Details 2 Base blobs

Lifecycle management uses your rules to automatically move blobs to cooler tiers or to delete them. If you create multiple rules, the associated actions must be implemented in tier order (from hot to cool storage, then archive, then deletion).

+ Add if-then block

If

Base blobs were \*

Last modified

More than (days ago) \*

Enter a value

Then

Delete the blob

**Move to cool storage**  
This is the most reliable option if cost is not a priority.

**Move to archive storage**  
Archive storage does not fully delete the blob. However, it cannot be moved back to cool storage.

**Delete the blob**  
This is the most efficient option if backing up a blob is not a priority.

# Determine Blob Object Replication

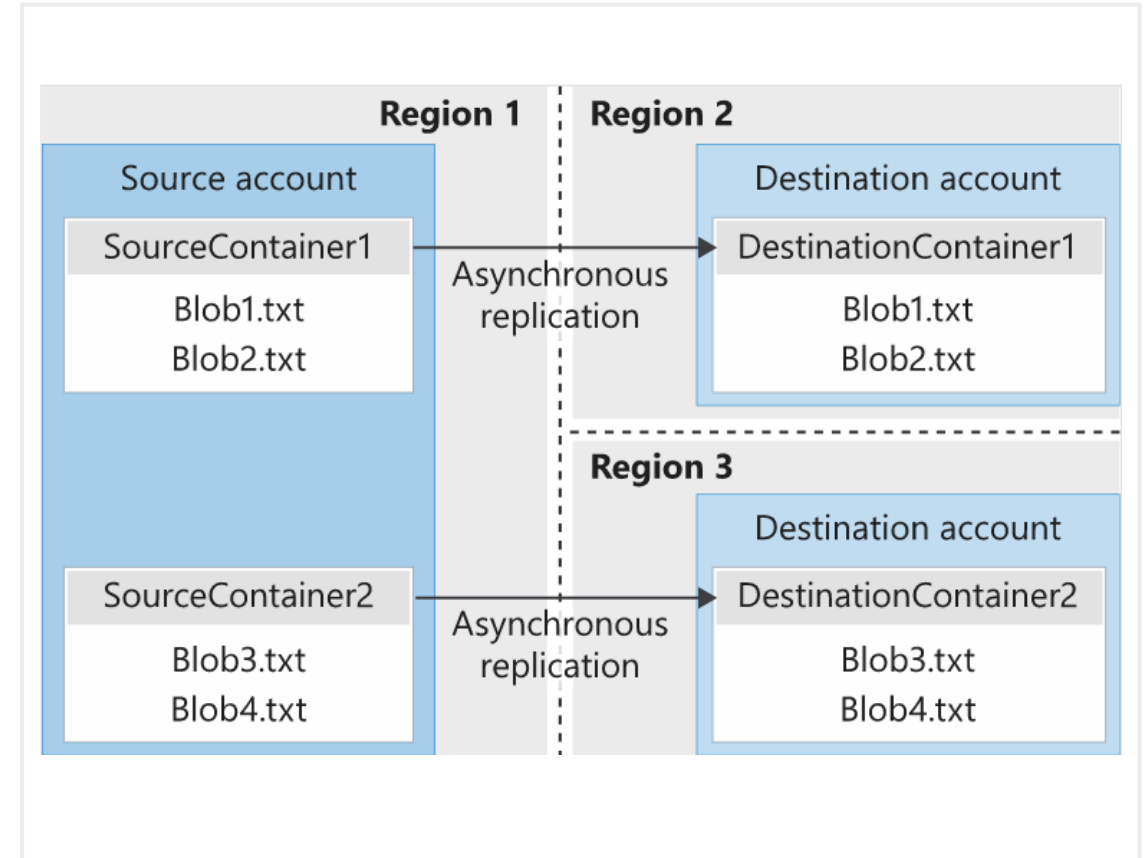
Asynchronous to any other Region

Minimizes latency for read requests

Increases efficiency for compute workloads

Optimizes data distribution

Optimizes costs



# Demonstration – Configure Blob Storage



**Create a  
container**

**Configure the  
container**

**Upload a file**

# Summary and Resources - Configure Blob Storage

## Knowledge Check Questions



## Microsoft Learn Modules ([docs.microsoft.com/Learn](https://docs.microsoft.com/Learn))

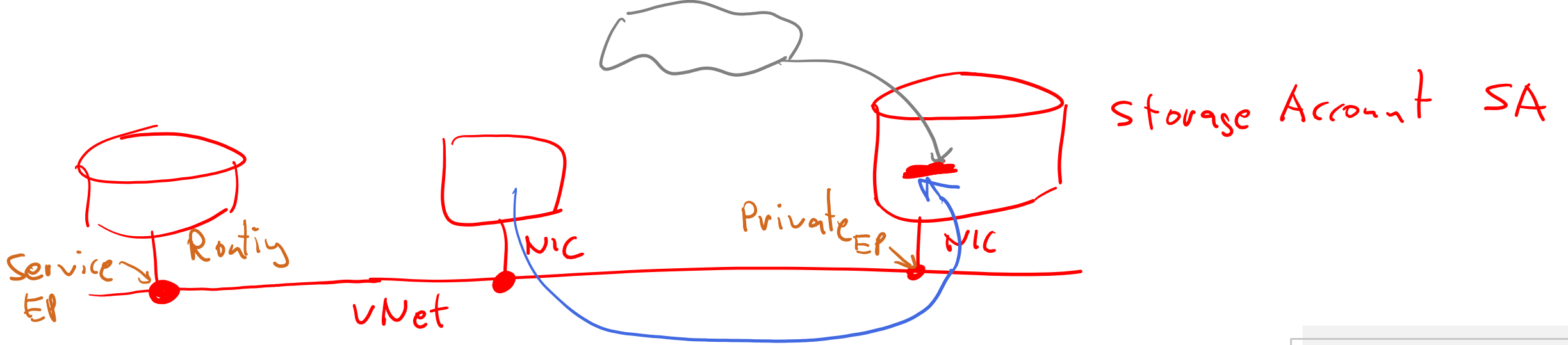
[Optimize storage performance and costs using Azure Blob storage tiers \(Sandbox\)](#)

---

[Gather metrics from your Azure Blob Storage containers \(Sandbox\)](#)

---

*A sandbox indicates a hands-on exercise.*



# Configure Storage Security

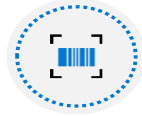




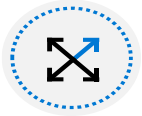
# Configure Storage Security Introduction



Review Storage Security Strategies



Create Shared Access Signatures



Identify URI and SAS Parameters



Demonstration – Configure storage security



Determine Storage Service Encryption



Create Customer Managed Keys



Apply Storage Security Best Practices



Summary and Resources

# Review Storage Security Strategies



Storage Service Encryption

---



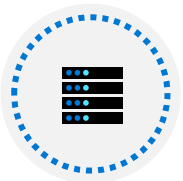
Authentication with Azure AD  
and RBAC

---



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

---

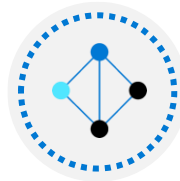


Azure disk encryption



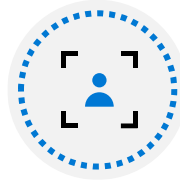
Shared Access Signatures –  
delegated access

---



Shared Key – encrypted  
signature string

---



Anonymous access to containers  
and blobs

# 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

Signing method ⓘ

☒ Account key ☐ User delegation key

Signing key ⓘ

Key 1 ▼

Permissions \* ⓘ

Read ▼

Start and expiry date/time ⓘ

Start

02/01/2021 

(UTC-08:00) Coordinated Universal Time-08 ▼

Expiry

02/02/2021 

(UTC-08:00) Coordinated Universal Time-08 ▼

Allowed IP addresses ⓘ

for example, 168.1.5.65 or 168.1.5.65-168.1....

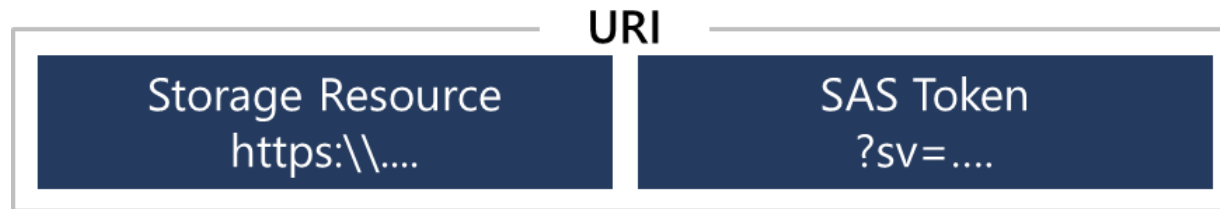
Allowed protocols ⓘ

☒ HTTPS ☐ HTTP

**Generate SAS token and URL**

# Identify URI and SAS Parameters

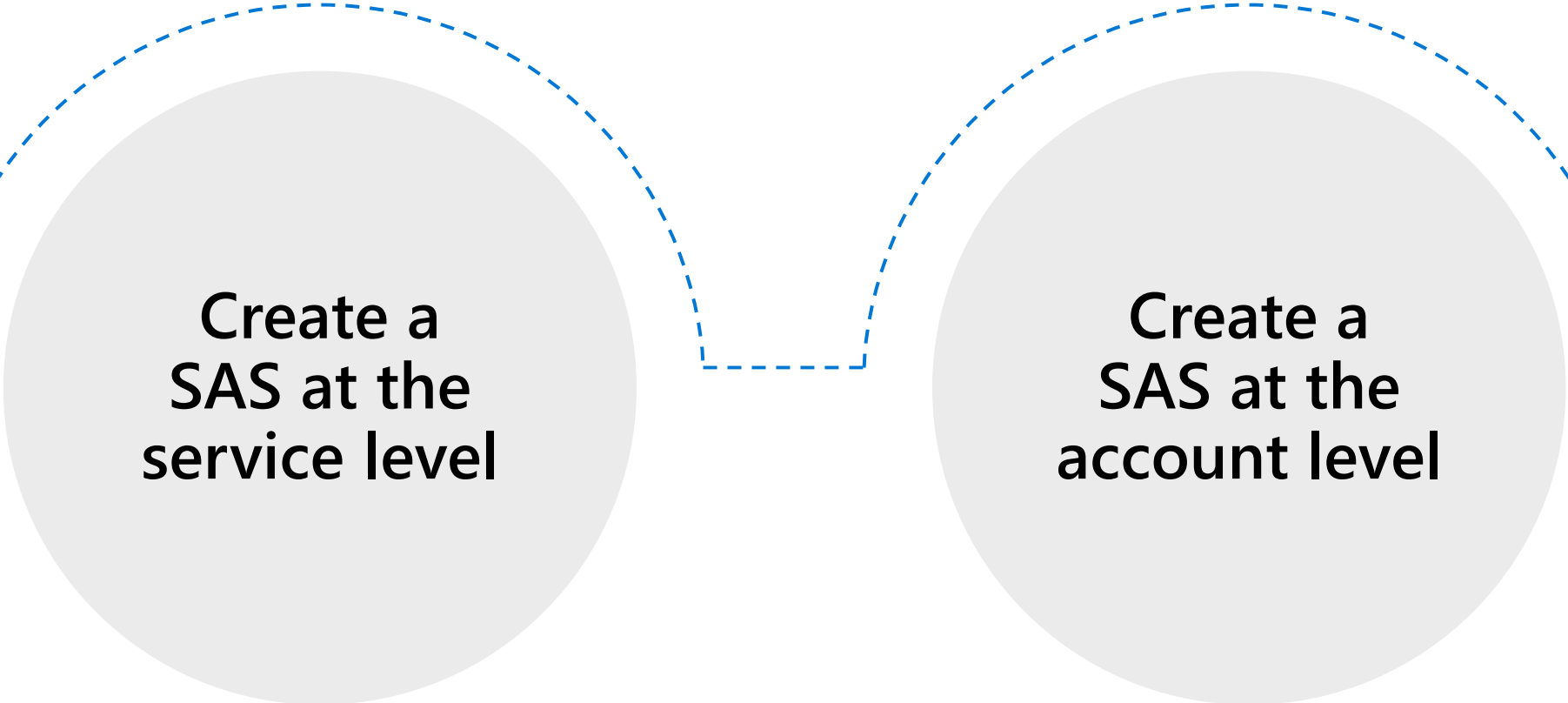
- A SAS is a signed URI that points to one or more storage resources
- Consists of a storage resource URI and the SAS token



<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=j0qABJZHfUVEBQ3yVn7kWiCKl00sxCiK1rzEchfAz8U%3D>

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

# Demonstration – Configure storage security



The diagram consists of two light gray circles arranged horizontally. A dashed blue line starts from the left, goes up and over the first circle, then down and under the second circle, and finally goes up and over the second circle before continuing to the right. This line connects the two circles, indicating a sequence of steps.

**Create a  
SAS at the  
service level**

**Test and verify  
the SAS**

# 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


**Encryption**

 Save  Discard

Storage service encryption protects your data at rest. Azure Storage encrypts your data as it's written in our datacenters, and automatically decrypts it for you as you access it.

By default, data in the storage account is encrypted using Microsoft Managed Keys. You may choose to bring your own key.

Please note that after enabling Storage Service Encryption, only new data will be encrypted, and any existing files in this storage account will retroactively get encrypted by a background encryption process.

[Learn More about Azure Storage Encryption](#) 

**Encryption type**

☒ Microsoft Managed Keys

☐ Customer Managed Keys



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

## Encryption type

- ☐ Microsoft Managed Keys
- ☒ Customer Managed Keys

**i** The storage account named 'storage987123' will be granted access to the selected key vault. Both soft delete and purge protection will be enabled on the key vault and cannot be disabled. [Learn more about customer managed keys](#) 

## Encryption key

- ☐ Enter key URI
- ☒ Select from Key vault

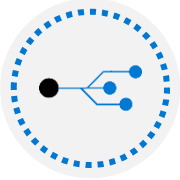
## Key vault and key \*

Key vault: keyvault987123

Key: storagekey

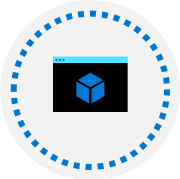
[Select a key vault and key](#)

# 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

---

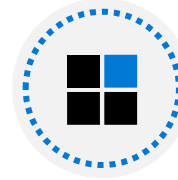


Use Storage Analytics to monitor your application

---



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](https://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.

Windows 11

SMB  
3.1 :445

net use A: → 

# Configure Azure Files



# Configure Azure Files Introduction



Compare Files to Blobs



Manage File Shares



Create File Share Snapshots



Demonstration – Configure File Shares



Configure Storage with Tools (optional)



Summary and Resources

\* File Sync is part of the Learn module but not included here

# 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	<ul style="list-style-type: none"><li>• Lift and shift an application to the cloud</li><li>• Store shared data across multiple virtual machines</li><li>• Store development and debugging tools that need to be accessed from many virtual machines</li></ul>
<u>Azure Blobs</u>	Client libraries and a REST interface that allows unstructured data (flat namespace) to be stored and accessed at a massive scale in block blobs	<ul style="list-style-type: none"><li>• Support streaming and random-access scenarios</li><li>• Access application data from anywhere</li></ul>

Queue

Service Bus  
pub data sub  
Fifo

Kafka (Apache)  
→ Event Hub

# Manage File Shares

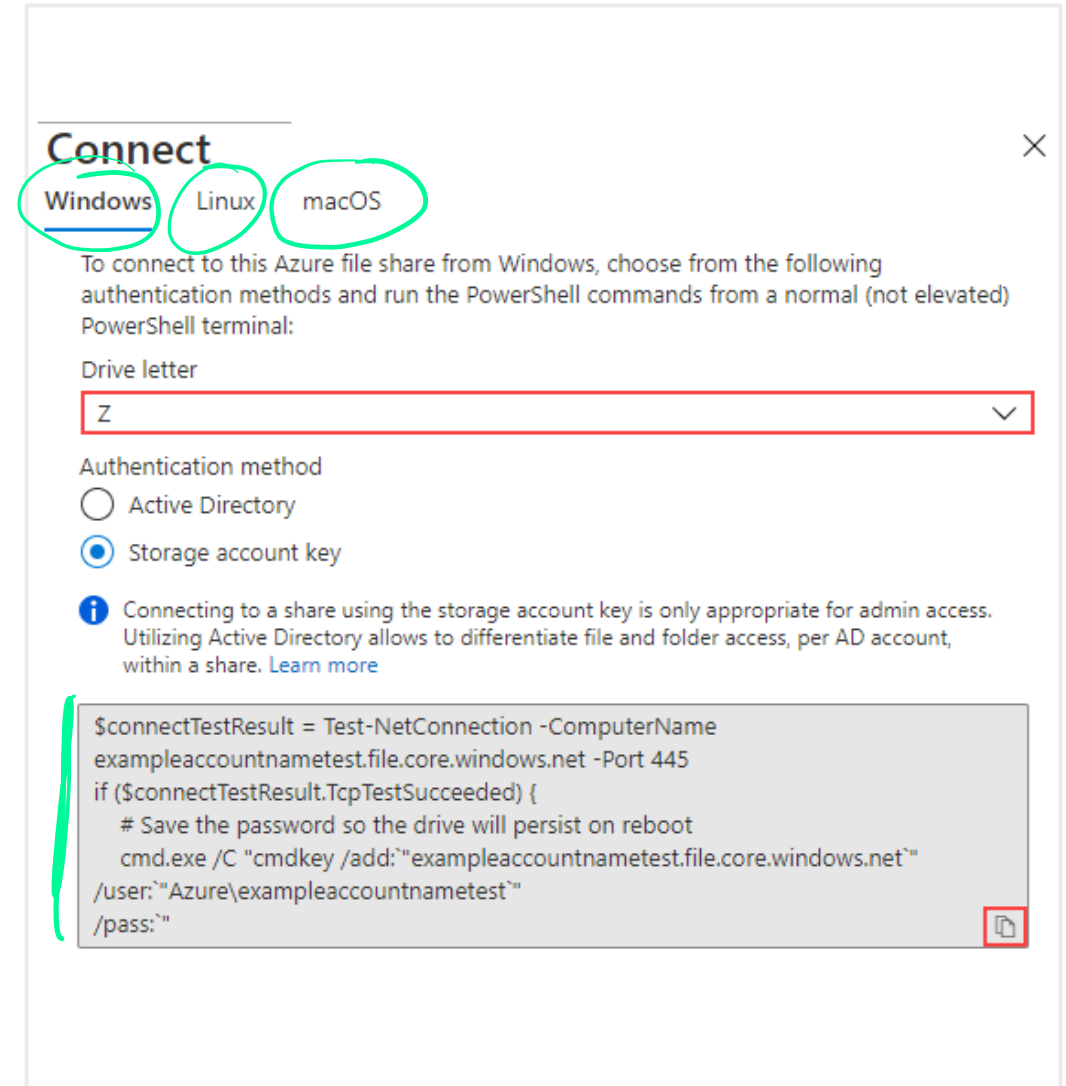
File share quotas

Windows – ensure port 445 is open

Linux – mount the drive

MacOS – mount the drive

Secure transfer required – SMB 3.0 encryption



The screenshot shows the 'Connect' dialog box for connecting to an Azure file share from Windows. The 'Windows' tab is selected and highlighted with a green circle. Below the tabs, there is a text box for the 'Drive letter' with 'Z' selected, highlighted by a red rectangle. Under 'Authentication method', the 'Storage account key' option is selected with a blue radio button. An information icon (i) is next to a note about using the storage account key for admin access. At the bottom, a PowerShell command is shown in a text box, highlighted by a green bracket on the left and a red square on the right. The command is: `$connectTestResult = Test-NetConnection -ComputerName exampleaccountnametest.file.core.windows.net -Port 445`  
`if ($connectTestResult.TcpTestSucceeded) {`  
 `# Save the password so the drive will persist on reboot`  
 `cmd.exe /C "cmdkey /add:"exampleaccountnametest.file.core.windows.net"`  
 `/user:"Azure\exampleaccountnametest"`  
 `/pass:""`

**Connect**

Windows Linux macOS

To connect to this Azure file share from Windows, choose from the following authentication methods and run the PowerShell commands from a normal (not elevated) PowerShell terminal:

Drive letter

Z

Authentication method




☐ Active Directory

☒ Storage account key

**i** Connecting to a share using the storage account key is only appropriate for admin access. Utilizing Active Directory allows to differentiate file and folder access, per AD account, within a share. [Learn more](#)

```
$connectTestResult = Test-NetConnection -ComputerName
exampleaccountnametest.file.core.windows.net -Port 445
if ($connectTestResult.TcpTestSucceeded) {
    # Save the password so the drive will persist on reboot
    cmd.exe /C "cmdkey /add:"exampleaccountnametest.file.core.windows.net"
    /user:"Azure\exampleaccountnametest"
    /pass:""
```

# Create File Share Snapshots

<div><div> Add snapshot</div><div> Refresh</div><div> Delete</div></div>		
Name		Initiator
<input type="checkbox"/>	2020-03-12T00:58:38.00000000Z	3/11/2020, 8:58:38 PM -

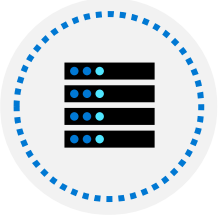
Incremental snapshot that captures the share state at a point in time

Is read-only copy of your data

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

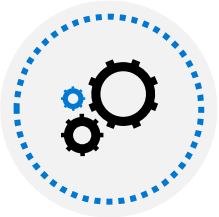
- Protection against application error and data corruption
- Protection against accidental deletions or unintended changes
- General backup purposes

# Demonstration – Configure File Shares



Create a file share and upload a file

---

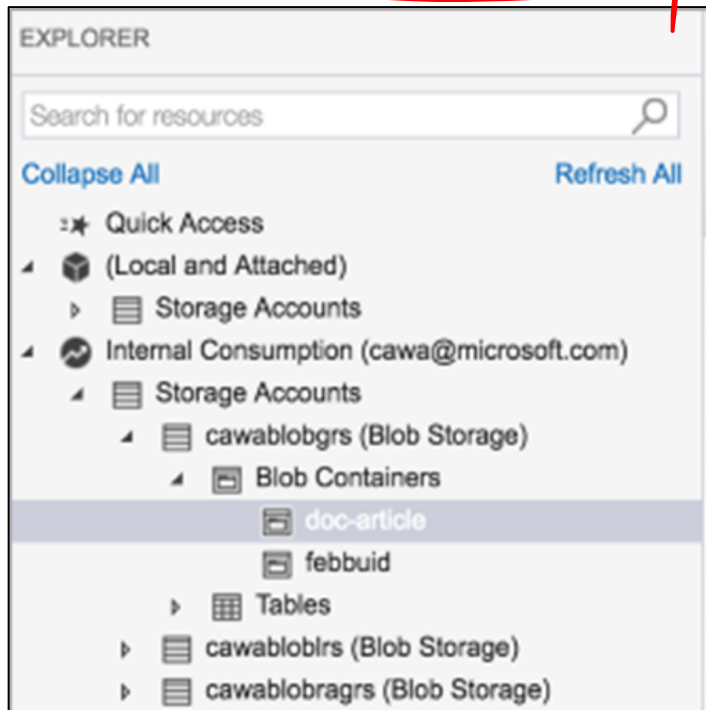


Manage snapshots

---

## Configure Storage with Tools (optional)

Azure Storage Explorer



The Import and Export service

Create import/export job ...

Create import/export job

Basics Job details Shipping Tags Review + create

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ASC DEMO

Resource group \* [Create new](#)

Name \*

Type ☒ Import into Azure ☐ Export from Azure

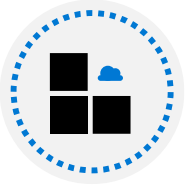
Destination Azure region \*

AzCopy

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



# Demonstration – Storage Tools (optional)



---

Work with Storage Explorer or Storage Browser

---

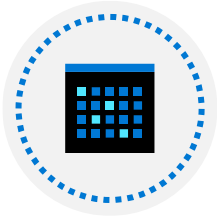


---

Work with AzCopy

---

# Demonstration – AzCopy (optional)



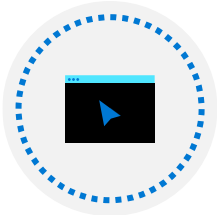
Install the AzCopy tool

---



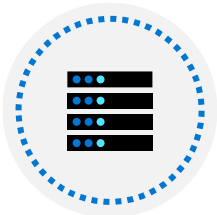
Explore the help

---



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](https://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](#)

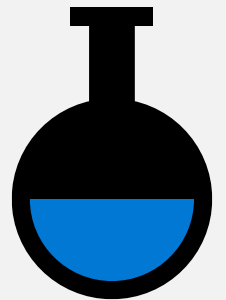
---

[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 – 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:

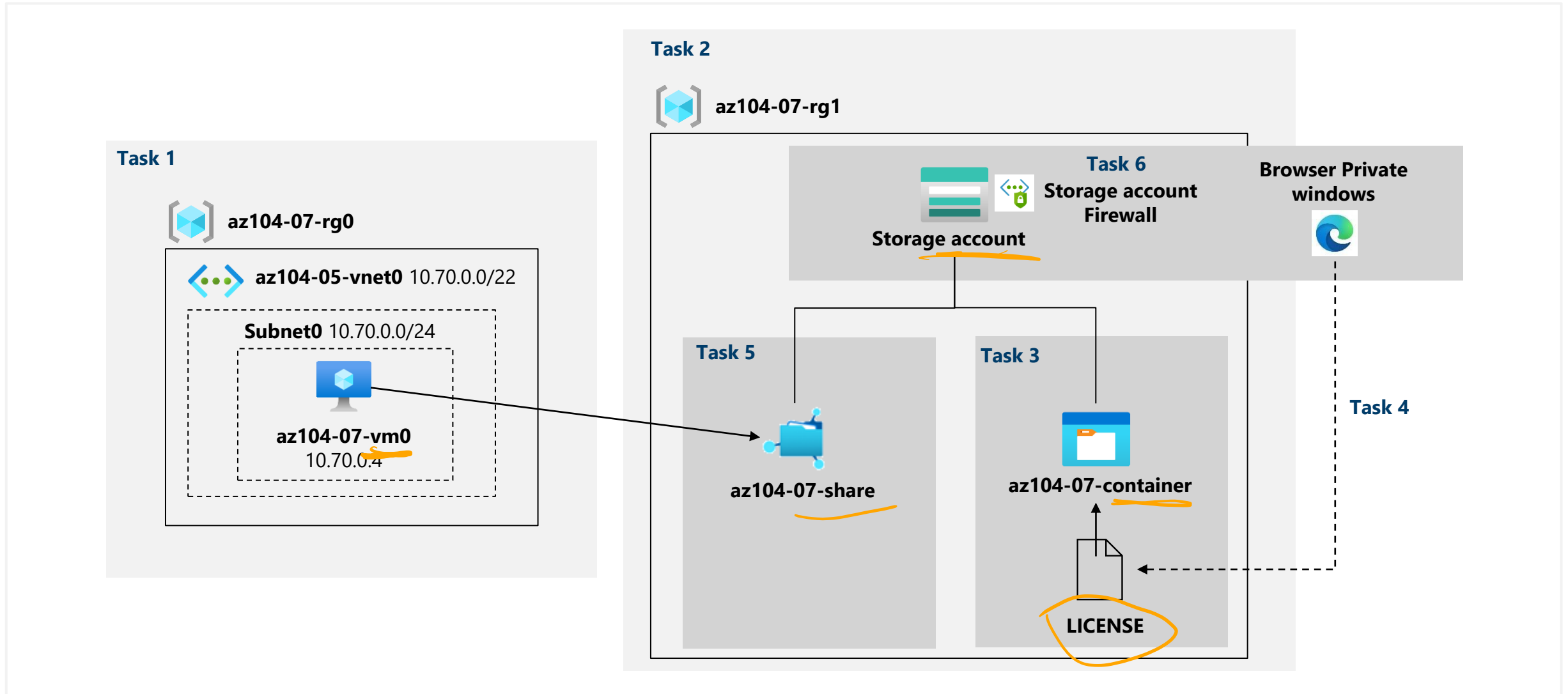
Create and configure an Azure Files shares

### Task 6:

Manage network access for Azure Storage

Next slide for an architecture diagram 

# Lab 07 – Architecture diagram



# End of presentation

