Brian Dekker

Brian.dekker@hva.nl

Learing story 2

My recap of what I learned during my second learing story.

Inhoud

[Storage accounts. 2](#_Toc100310735)

[What is a storage account. 2](#_Toc100310736)

[How to setup a storage account. 3](#_Toc100310737)

[Single region redundancy 4](#_Toc100310738)

[LRS 4](#_Toc100310739)

[ZRS 5](#_Toc100310740)

[Redundancy with a secondary region. 5](#_Toc100310741)

[GRS 6](#_Toc100310742)

[GZRS 6](#_Toc100310743)

[Advanced settings 7](#_Toc100310744)

[Security 8](#_Toc100310745)

[Blob Storage 12](#_Toc100310746)

[What is blob storage 12](#_Toc100310747)

[Cosmo DB 13](#_Toc100310748)

[API problemen 13](#_Toc100310749)

[Managed identity’s 14](#_Toc100310750)

[Fisrt watched the 48min video. 14](#_Toc100310751)

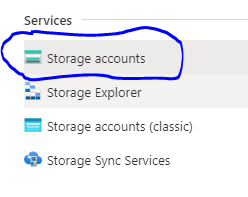
[Recap of the video. 14](#_Toc100310752)

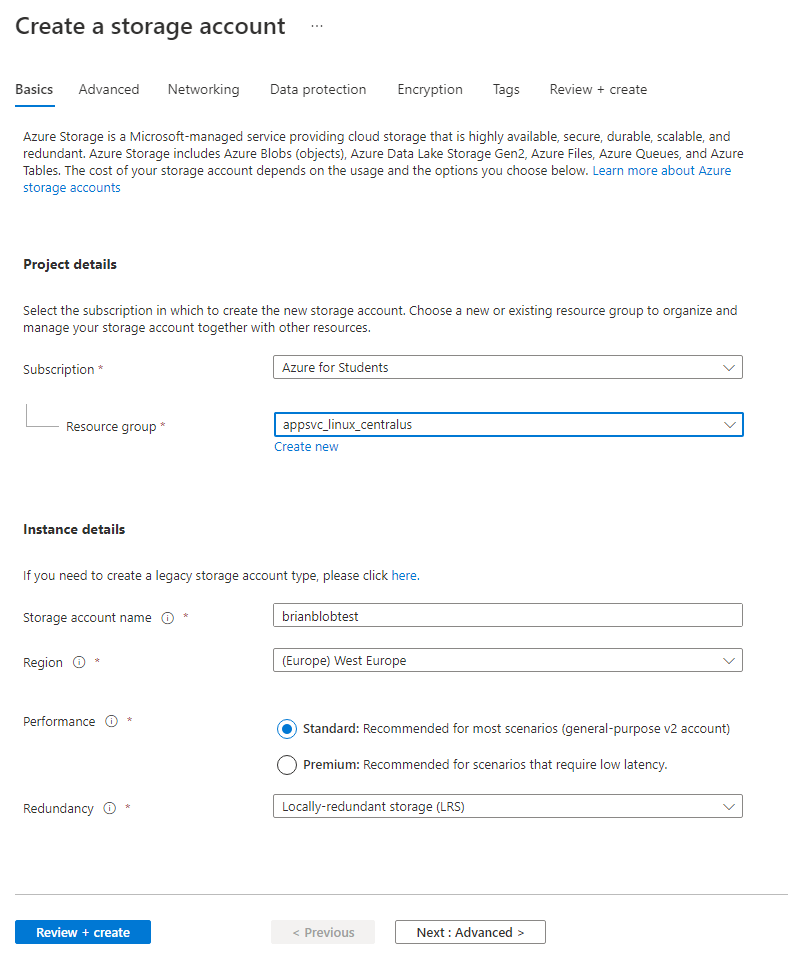
# Storage accounts.

## What is a storage account.

## How to setup a storage account.

When you want to create a blob storage in azure u need to create a storage account. Make sure you choose storage account and not storage account (classic).



Now we need to create our account when u click create this screen will open.

Now all the options here we should already be familiar with from other services we created however the two at the bottom are unique to storage account so I will explain.

**Performance:** The performance option is pretty straight forward do you want high speed access to your files all the time or is it ok for a slower experience from time to time.

**Redundancy:** This option is a bit more complicated u get to choose between 4 different redundancy options. Here are the options in short  
- **Locally-Redundant storage (LRS):** This is the most basic option that provides protection against server rack and drive failures. This is also the cheapest option.  
- **Geo-redundant storage (GRS**) : This option makes sure that if something goes wrong a backup is ready in a secondary region.   
- **Zone-redundant storage (ZRS)** : This option gives protection against datacentre level failures  
- **Geo-zone-redundant storage** : This option gives you both the backup capability’s or GRS and the protection of ZRS.

Below you can see these options represented in picture form with a more detailed explanation.

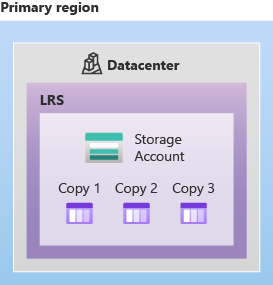
## Single region redundancy

When using blob storage in a single region u have two redundancy option

Local redundancy storage(LRS) and Zero redundancy storage(ZRS).

### LRS

Local redundancy storage replicates your data three times within a datacentre in the primary region.

LRS is the lowest cost option of the 4 options and offers the least durability compared to the other options. LRS protects data against server rack and drive failures. But lets say for example a massive fire breaks out in the datacentre all 3 copies will be lost forever since you wont be able to recover them in any way. To combat this risk you need to at least use one of the other 3 redundancy options.

LRS is a good option when:

* Your data can easily be reconstructed if data loss occurs.
* If your data is restricted to a certain region or country because of government requirements. It can happen that if you pick the GRS or ZGRS option the data gets geo replicated in a region that goes against those rules.

### ZRDiagram showing how data is replicated in the primary region with ZRSS

Zero-redundant storage (ZRS) replicates your Azure Storage data synchronously across three Azure availability zones in the primary region. Each zone has its own location with its own power source, cooling and network. So if one zone catches fire the other zones will still function and your data is safe.

ZRS is recommended when you require high availability of you data. ZRS is basically a upgraded version of LRS since u can still use ZRS under the same government rules as LRS because ZRS does not backup your data in different regions or countries.

## Redundancy with a secondary region.

When using storage with a secondary region u have two redundancy option.

Geo redundancy storage(GRS) and Geo zone redundancy storage(GZRS).

For Data that requires high durability you can choose one of these two options. The secondary region is hundreds of miles away so even in the case of a complete region collapse your data is still safe in the secondary region.

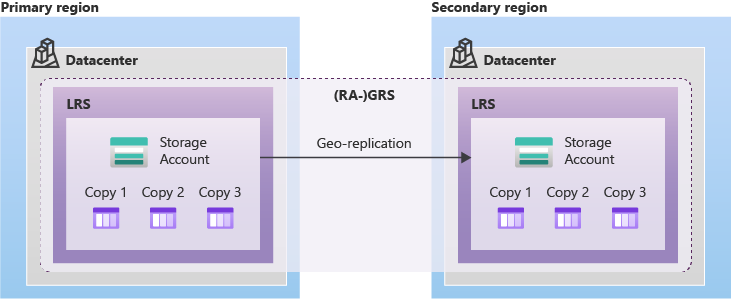
The secondary region is based on what primary region you picked when creating your storage account the secondary region can’t be changed.

You cant write or read any of the data in the secondary region unless there is a failover to that region. For read access to the secondary region you need to configure your storage account to allow read access on the secondary region this is not on by default.

If a failover happens the secondary region becomes the new primary region. The data in the secondary region can miss some data compared to the primary region in case of a failover because both region sync asynchronously so it can happen that when the primary region fails the secondary region does not have the newest version of the data. At the moment the time it takes for the secondary to be updated after the primary region is less than 15 minutes.

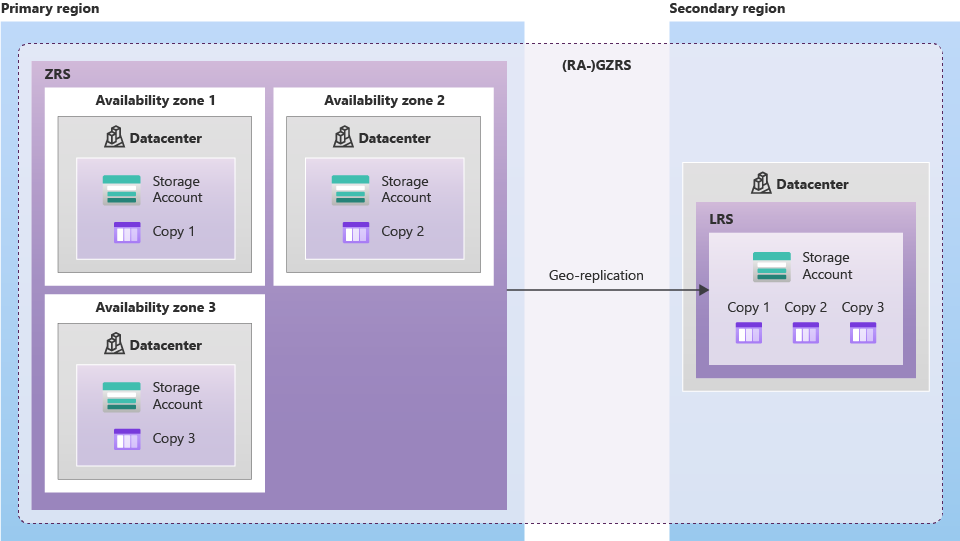
### GRS

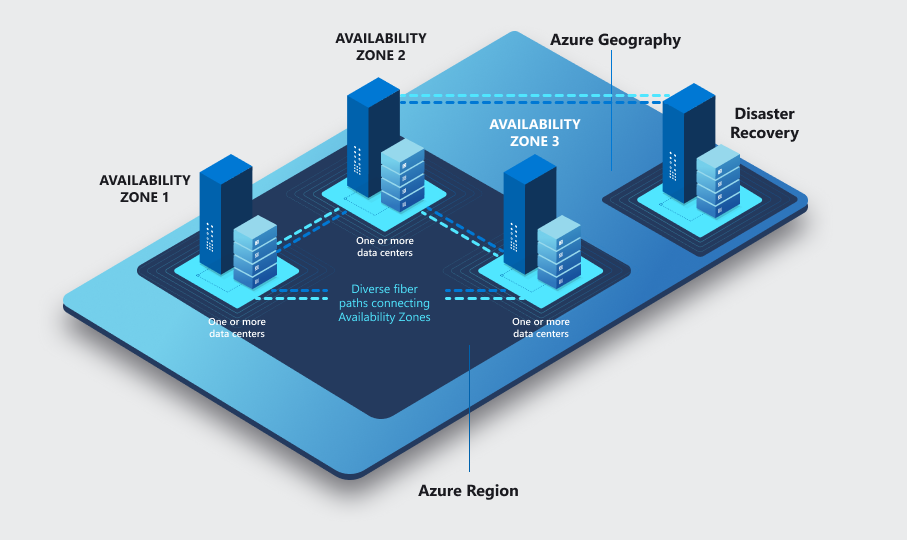
Geo-redundant storage (GRS) copies your data 3 times like LRS but also makes a backup of you data and copies in a secondary region. When you write data to the primary region first the data will be replicated in that region after that the data will be written to the secondary region and replicated after it has been written.



### GZRS

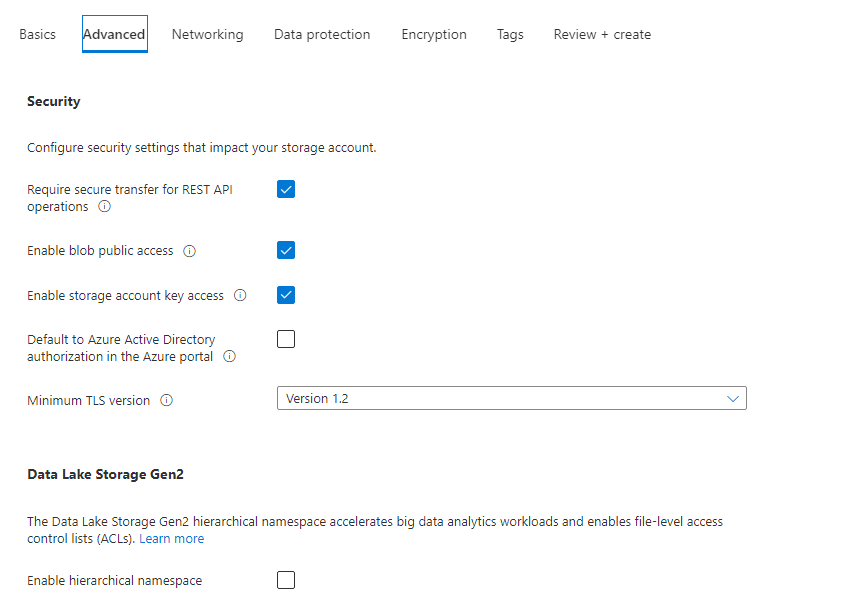
Geo-zone-redundancy storage is ZRS and GRS combined. Data in a GZRS storage is copied across three zones in the primary region based on ZRS and then replicated in a secondary region based on LRS this option gives you everything maximum durability and excellent performance speed.





Ok now that we know what all the redundancy option do lets get back to creating our test storage account.

## Advanced settings



### Security

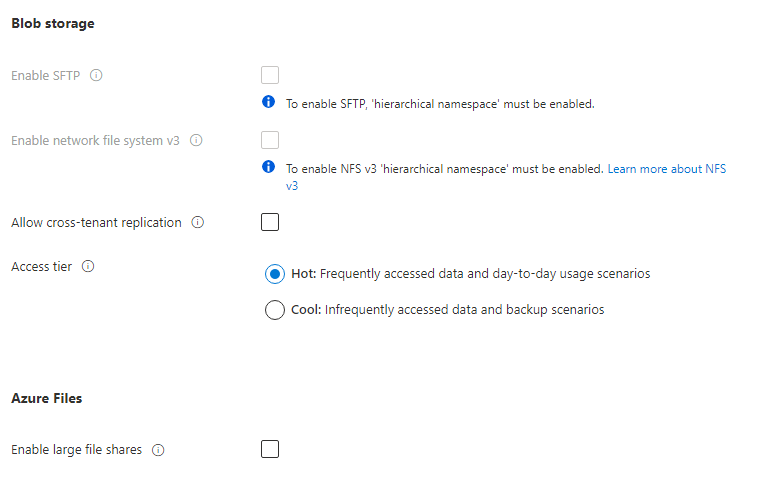
**Require secure transfer for rest api operations**: The secure transfer option enhances the security of your storage account by only allowing REST API operations on the storage account using HTTPs. Any requests using HTTP will be rejected when this setting is enabled. When you are using the Azure file service, connections without encryption will fail, including scenarios using SMB 2.1, SMB 3.0 without encryption, and some flavors of the Linux SMB client. Because Azure storage doesn’t support HTTPs for custom domain names, this option is not applied when using a custom domain name. Connections via NFSv3 for blobs over TCP will succeed but will not be secured

**Enable blob public access**: When blob public access is enabled, one is permitted to configure container ACLs (Access control list) to allow anonymous access to blobs within the storage account. When disabled, no anonymous access to blobs within the storage account is permitted, regardless of underlying ACL configurations

**Enable storage account key access:** When storage account key access is disabled, any requests to the account that are authorized with Shared Key, including shared access signatures (SAS), will be denied. Client applications that currently access the storage account using shared key will no longer work.

**Default to azure directory:** When this property is enabled, the Azure portal authorizes requests to blobs, queues, and tables with Azure Active Directory by default

**Minimum TLS(Transport layer security version:** Set the minimum TLS version needed by applications using your storage account's data

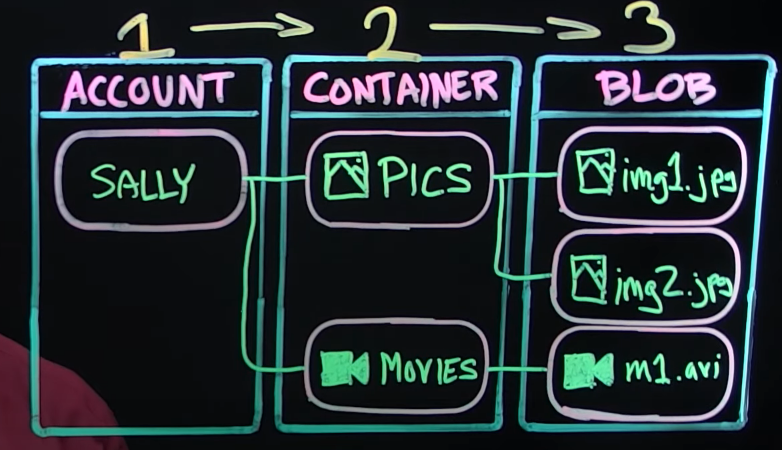


# Blob Storage

I did some research on azure blob storage

## What is blob storage

Blob storage can save all kinds of data. Blob storage is made up of 3 pillars that account, containers and blobs. Accounts can create containers and in those containers the user can save their blobs in this example sally has access or created the containers Pics and Movies and she can now add blob (files) to those containers these blobs are usually unstructured. In simple terms its like you normal file system in windows.



Azure allows you to decide who has access to what container. Blob storage is useful when u need to share files or keep track of files with the people you work with. It is a way to centralize all your files in an organization in a safe and secure space that can be accessed from different locations.

## Cosmo DB

## API problemen

# Managed identity’s

Here i wil recap my progress of following the managed identity workshop.

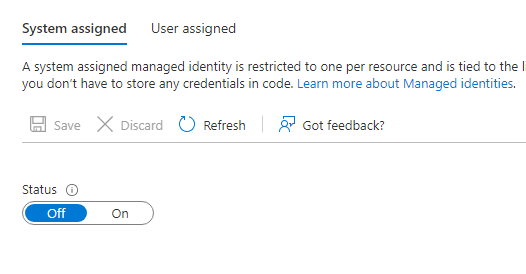
## Fisrt watched the 48min video.

<https://www.youtube.com/watch?v=rC1TV0_sIrM&ab_channel=JohnSavill%27sTechnicalTraining>

### Recap of the video.

Managed identity’s fix the issue of having to use secrets or certificates to access service principles. A service principle is a representation of a app instance. So let’s say I want to use a Netflix AAD in my AAD(Azure Active Directory) it will create a service principle in my AAD that points to the Netflix AAD. Now for the app in my resource to use my AAD we need to connect the two one way of doing this is by using secret keys or certificates but we don’t want to do that so we will use managed identity’s instead.

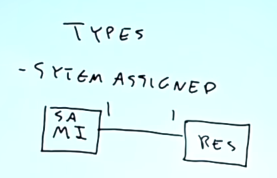
In a resource you can turn on a identity that azure is going to manage.



When you turn this on it will create service principle in you AAD with the same name as the azure resource. Now this service principle and Resource are linked with a shared life cycle so if I delete the resource the SP is also deleted. Now if we create a second resource we can give permissions to Resource 1 service principle to do things in this resource with RBAC(Resource based access control).

Now there are two different types of managed identity’s

### System assigned

System assigned managed identity’s are managed identity’s that can only have one resource connected to it and a resource can only have one managed identity. The life cycle between the two is shared if you delete the resource the managed identity is also deleted. System identity is a One to One relationship with resources.

### User assigned

With this managed identity the user creates a managed identity as a separate resource. A user can create a managed identity in a AAD and then assign it to multiple resources. So the life cycle between the two is different if you delete a resource the managed identity stays. So user identity is a many to many relationship with resources.

So lets say for example I have a storage account with some blob storage I can now create a managed identity and give it permission to read the blob data in that storage account now any resource that I give that manged identity to has permission to read the blob data.