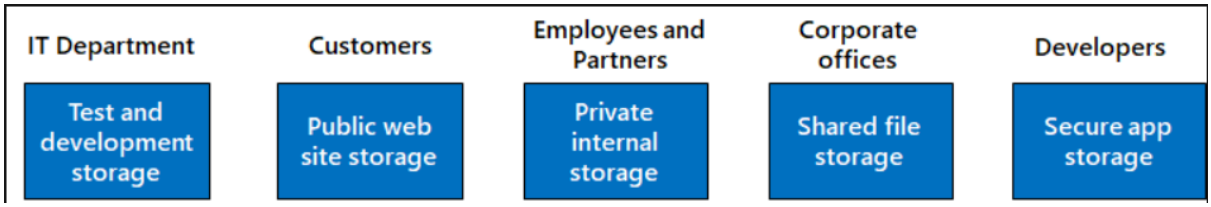
Project - Secure storage for Azure Files and Azure Blob Storage

Summary -storing business data securely by using Azure Blob Storage and Azure Files

Provide storage for different purposes such as IT department testing and training, public website, private company documents, shared file storage for the company offices, and a new company app.



Task List

Create and configure a storage account.

Create and configure blob storage.

Create and configure Azure Files.

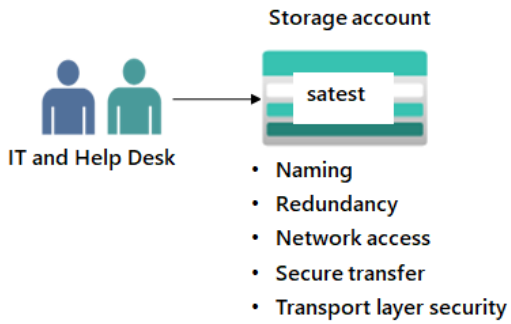
Configure encryption for storage.

Configure networking for storage.

Provide storage for the IT department testing and training

The IT department needs to prototype different storage scenarios and to train new personnel. The content isn't important enough to back up and doesn't need to be restored if the data is overwritten or removed. A simple configuration that can be easily changed is desired.

Architecture diagram

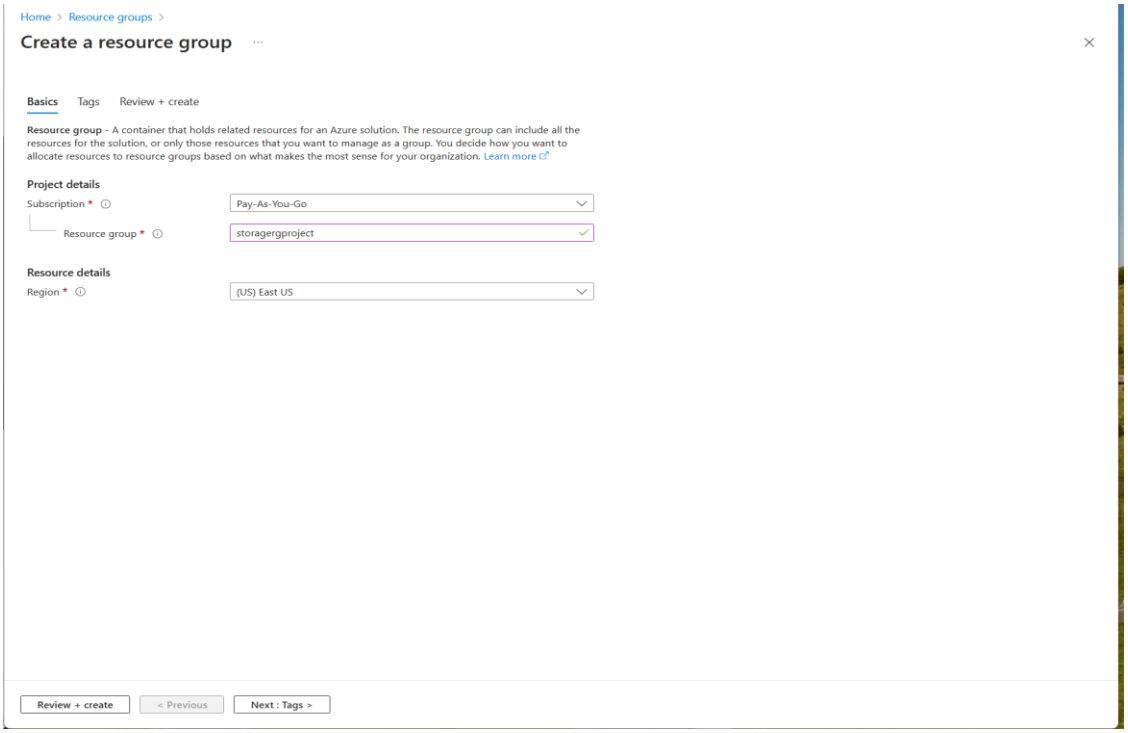


Task to be completed

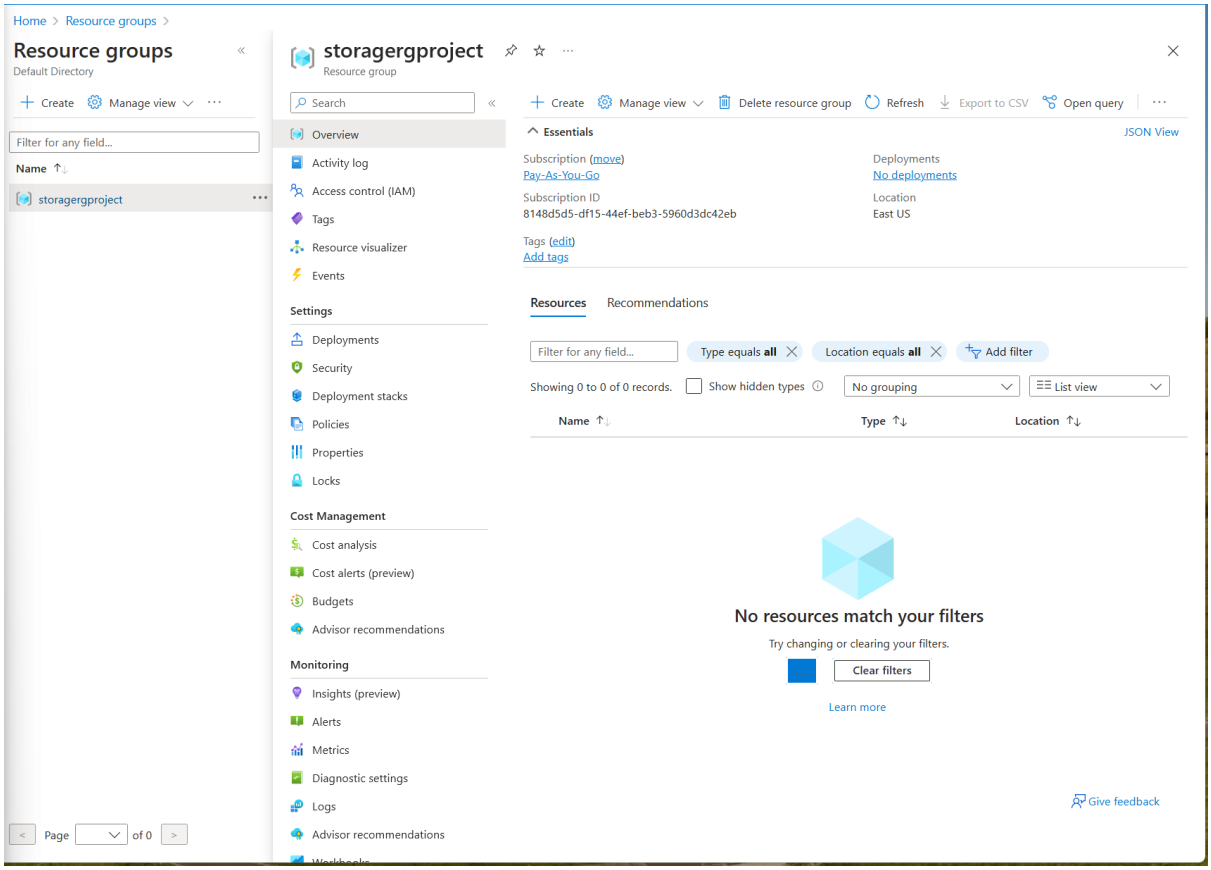
- Create a storage account.
- Configure basic settings for security and networking.

Step 1 Create a resource group and a storage account

Create the resource group with the following settings

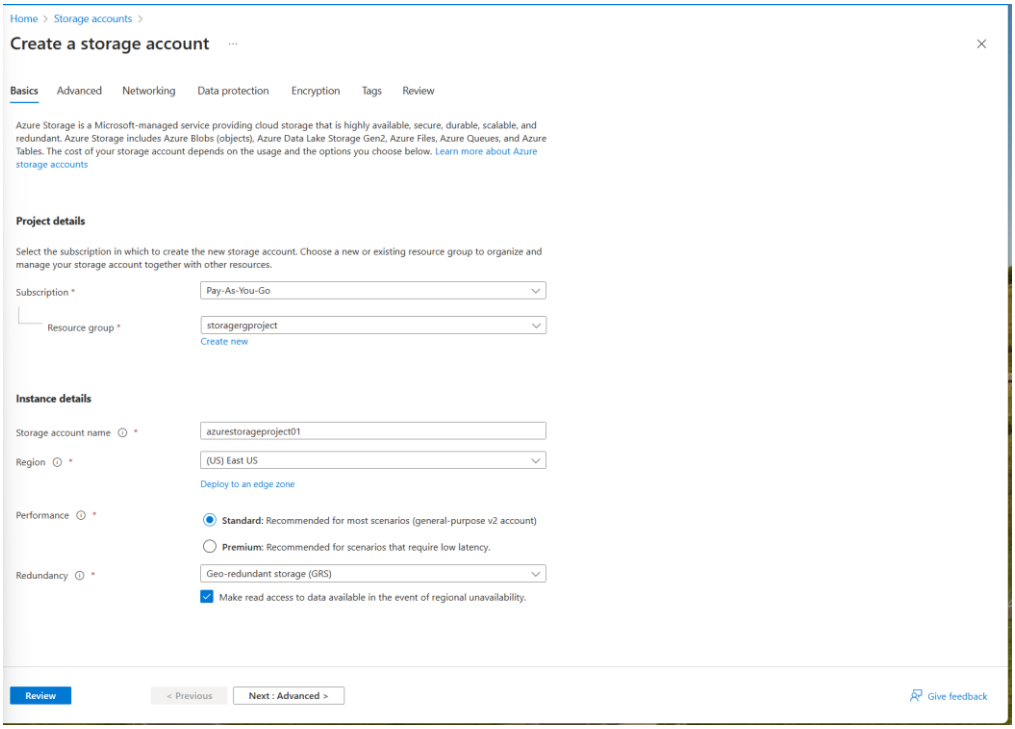


Review and create to validate the resource group

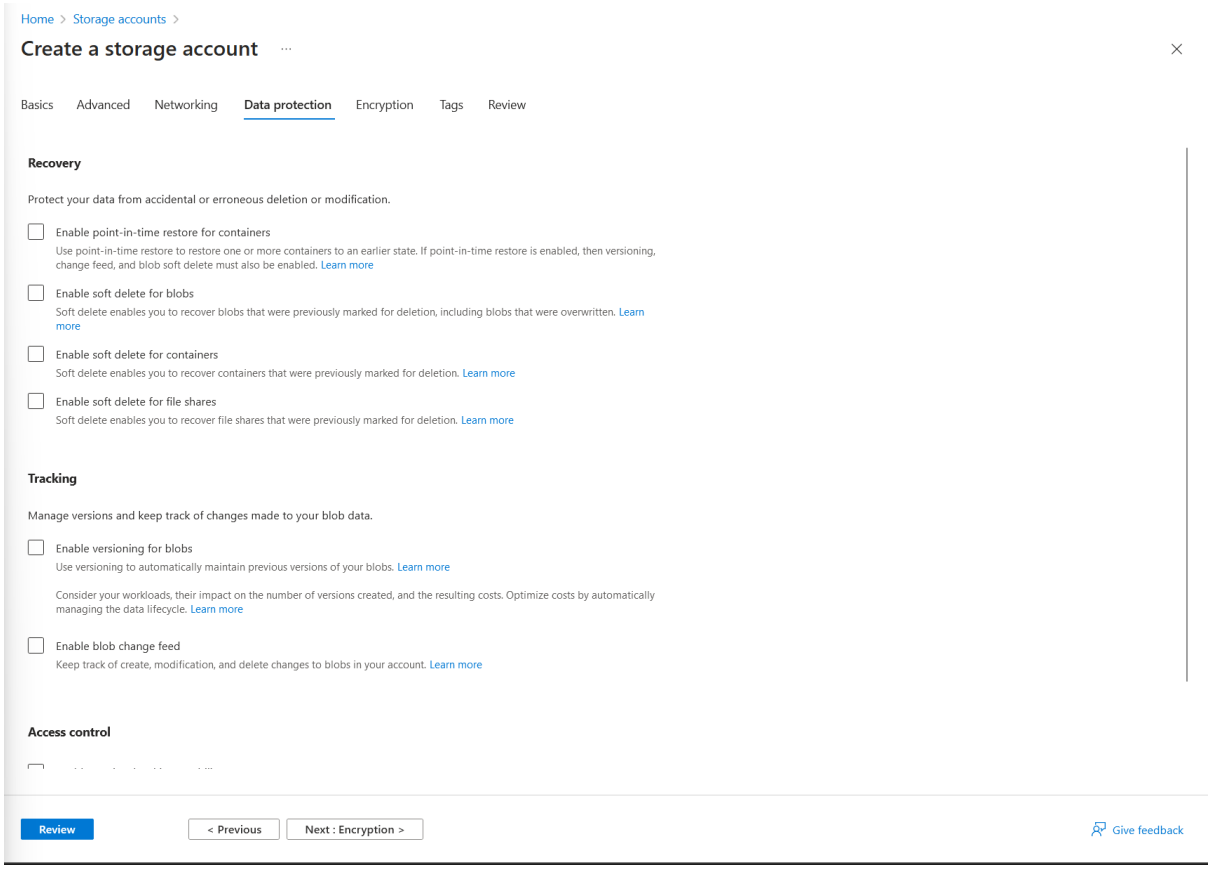


Step 2 - Create and deploy a storage account to support testing and training

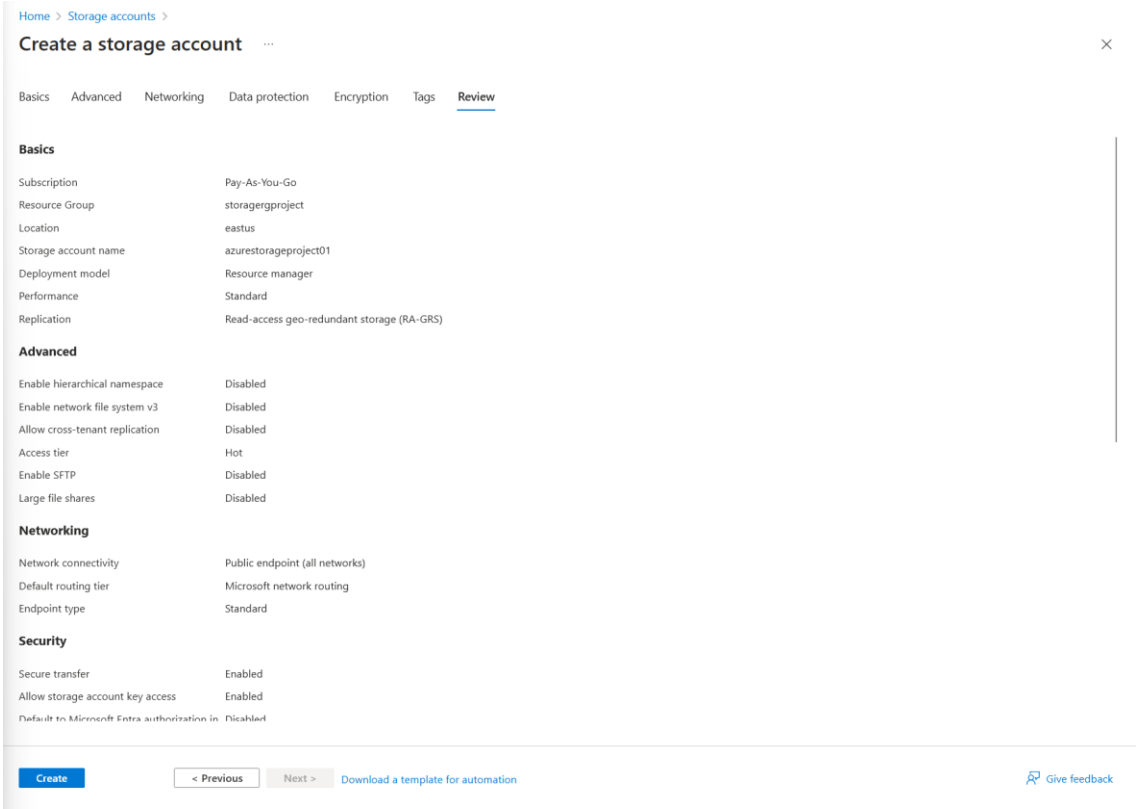
Create storage with the following setting



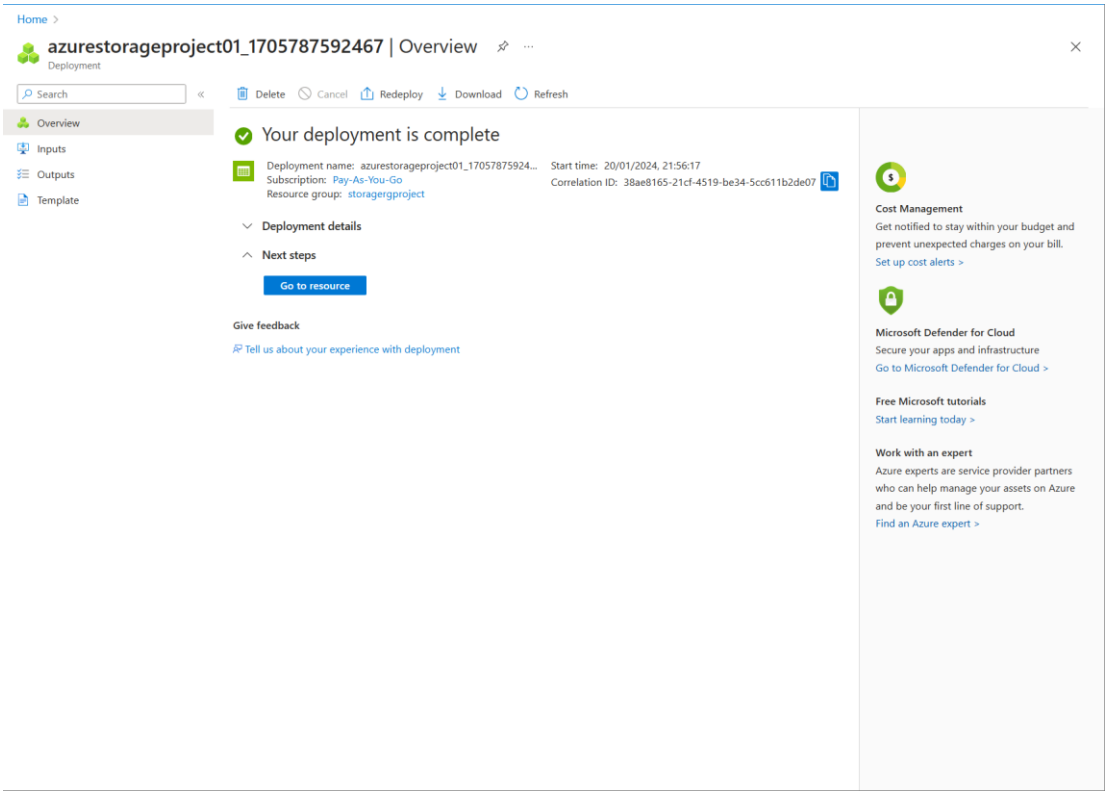
Disable the recovery for the storage account as this is not needed but if you need backups then this would be import.



Leave other settings as default and click on review then create.



Click go to resource on deployment completed.

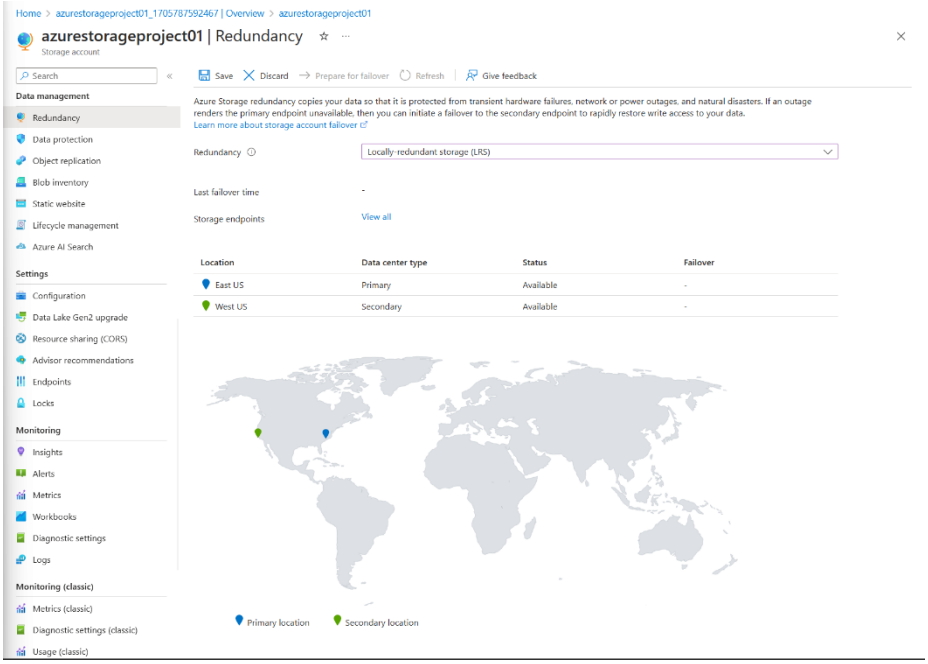


Step 3 - Configure simple settings in the storage account

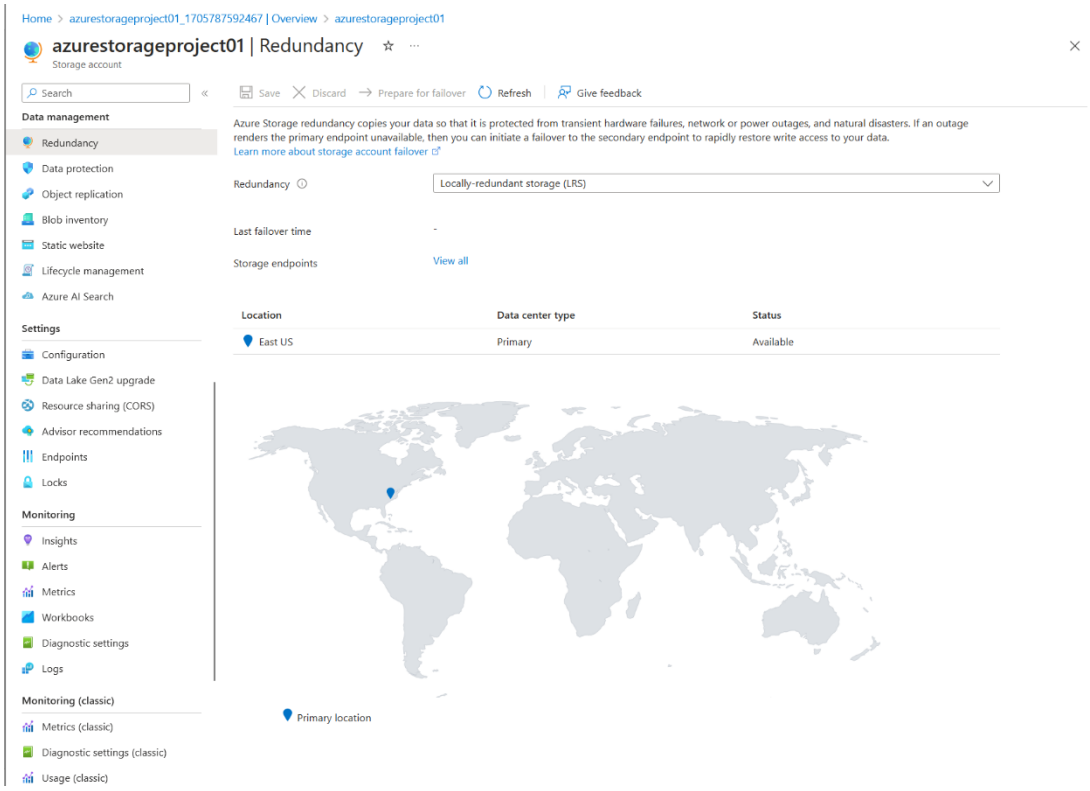
The data in this storage account doesn't require high availability or durability. A lowest cost storage solution is desired.

Under the left navigation in the **Data management** section, select the **Redundancy** blade

Select **Locally-redundant storage (LRS)** in the **Redundancy** drop-down

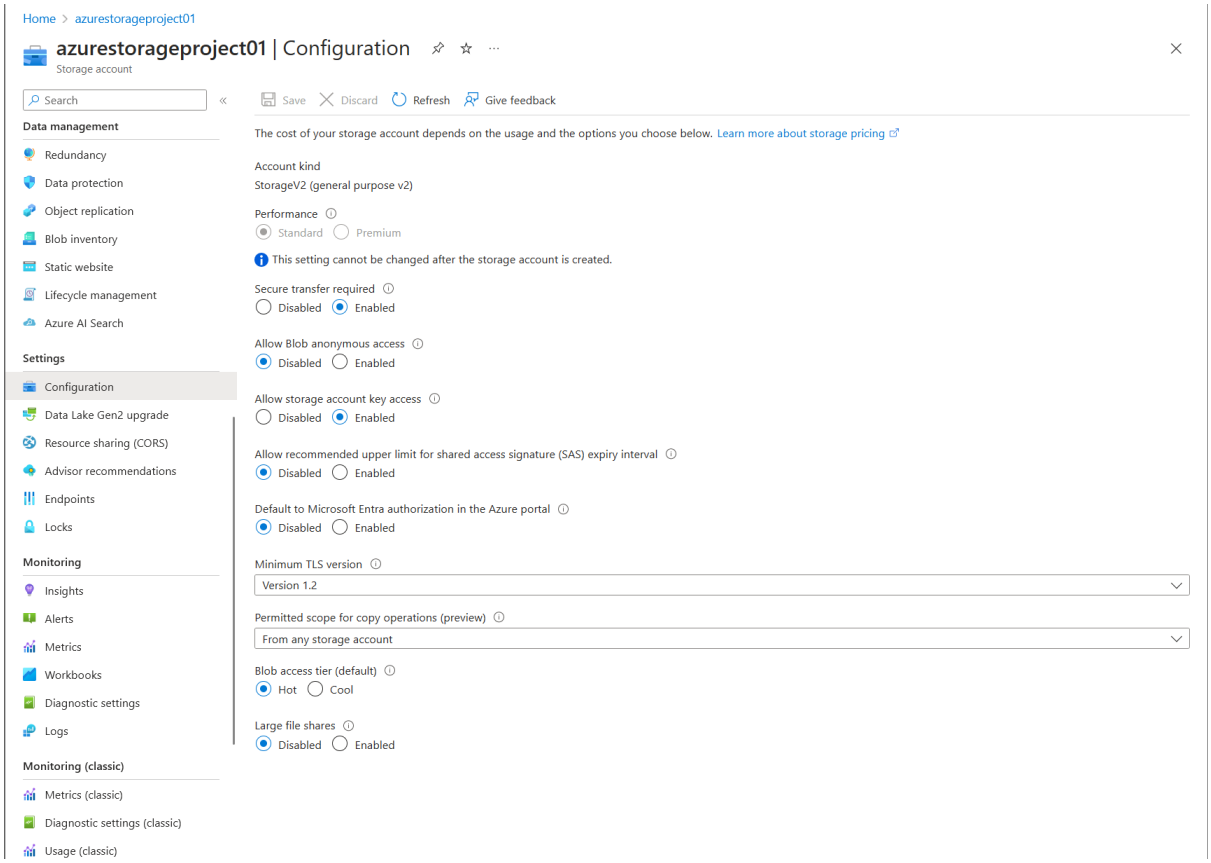


Click on save or else the settings will not be saved.

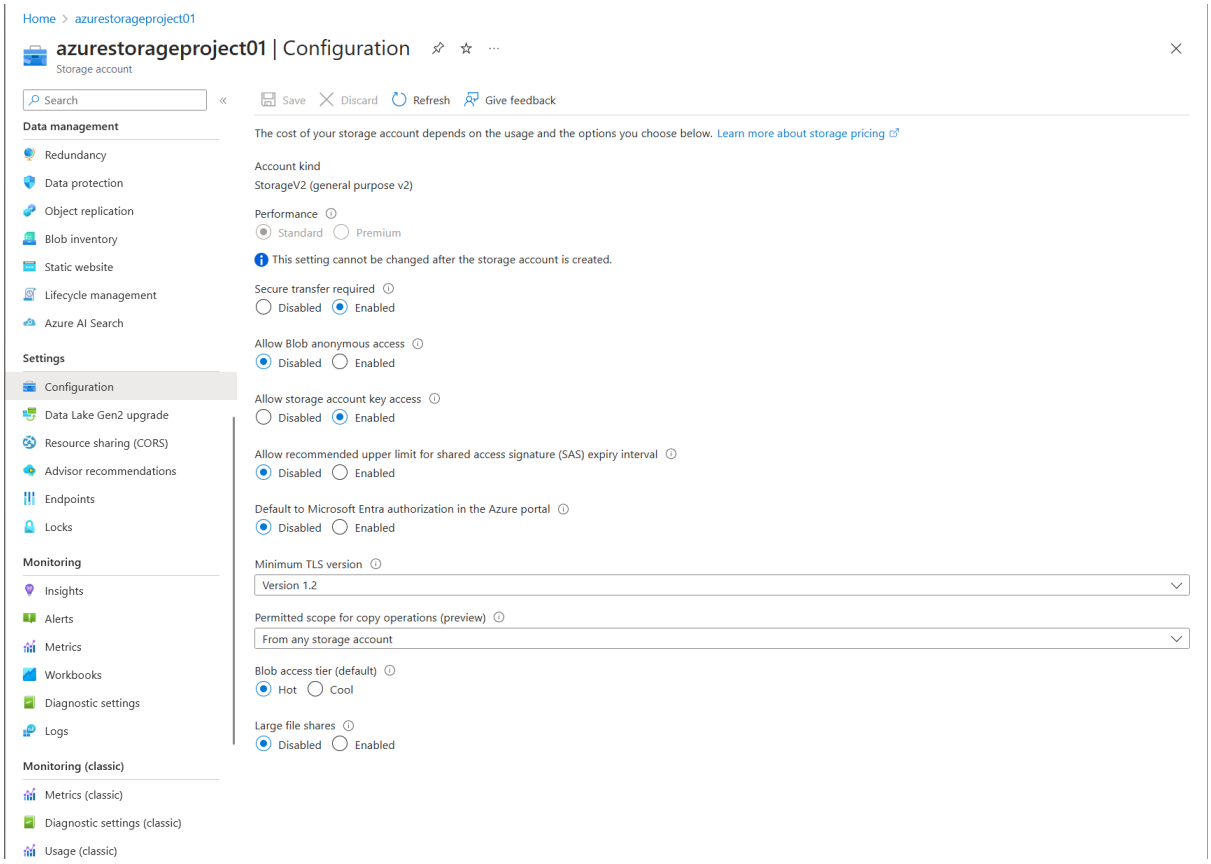


Step 4 - The storage account should only accept requests from secure connections

In the **Settings** section, select the **Configuration** blade

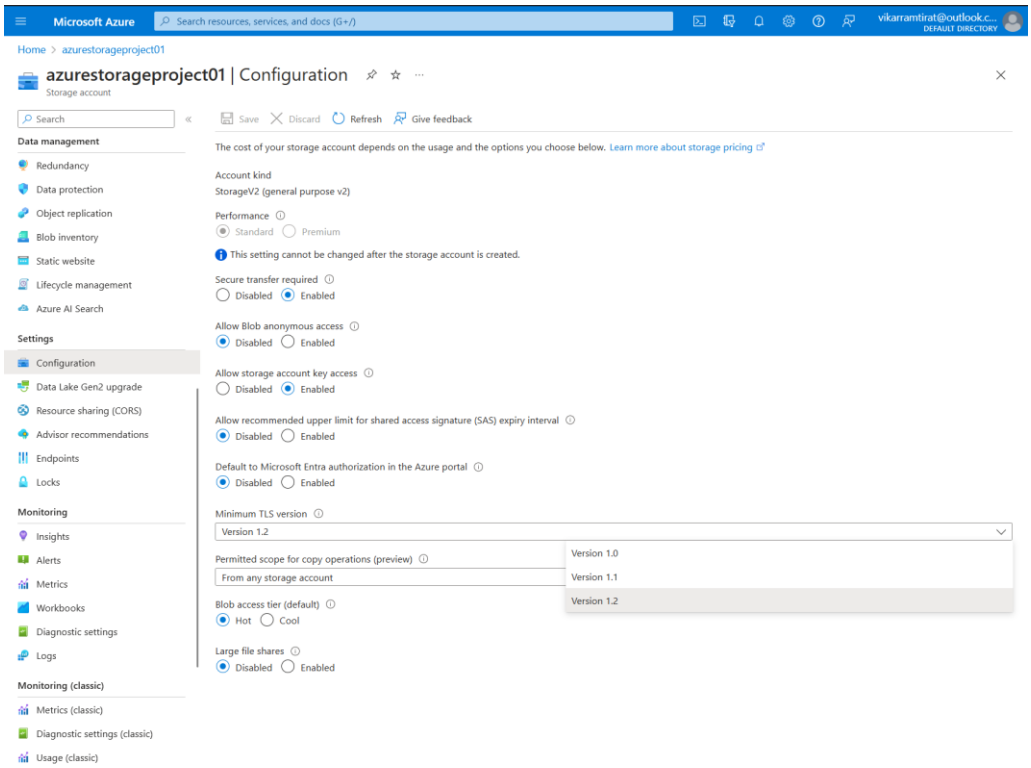


Ensure **Secure transfer required** is **Enabled**

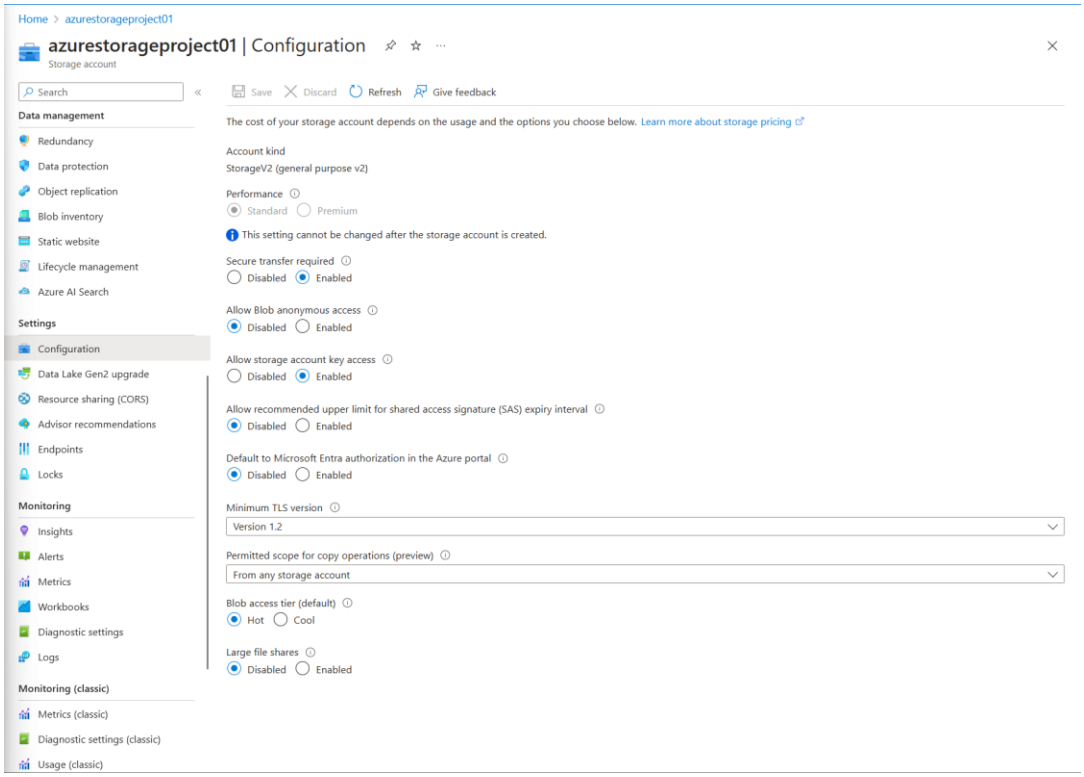


Step 5 - Developers would like the storage account to use at least TLS version 1.2

In the **Settings** section, select the **Configuration** blade

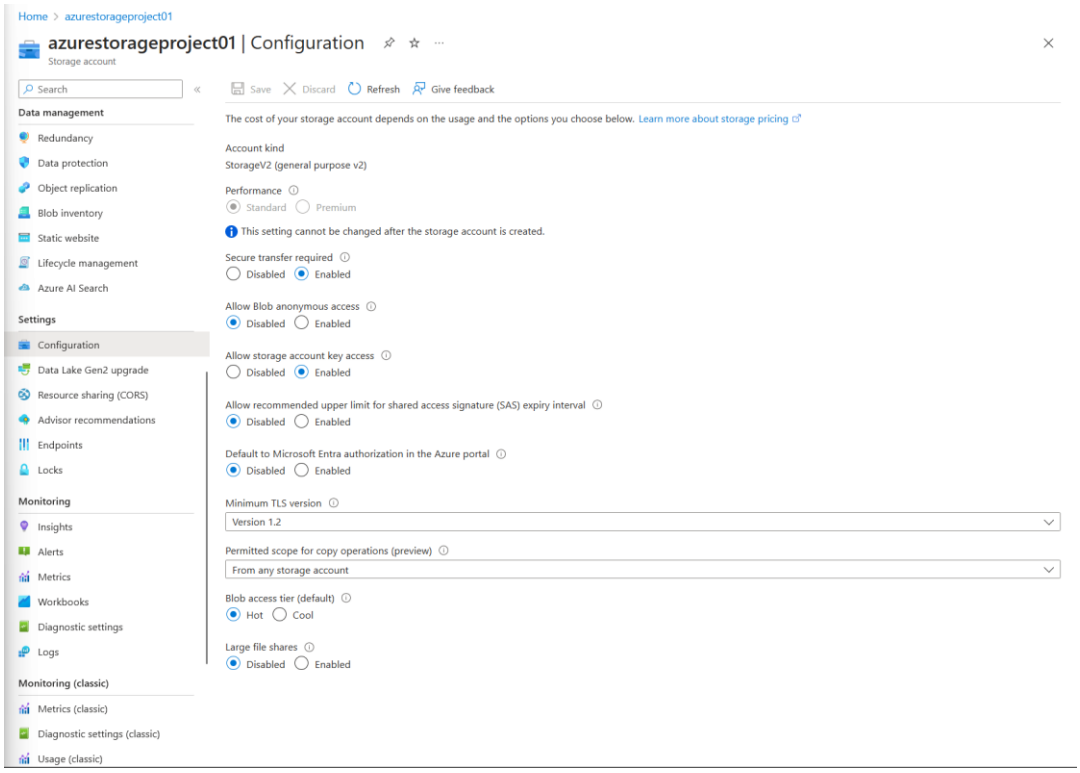


Ensure the **Minimal TLS version** is set to **Version 1.2**



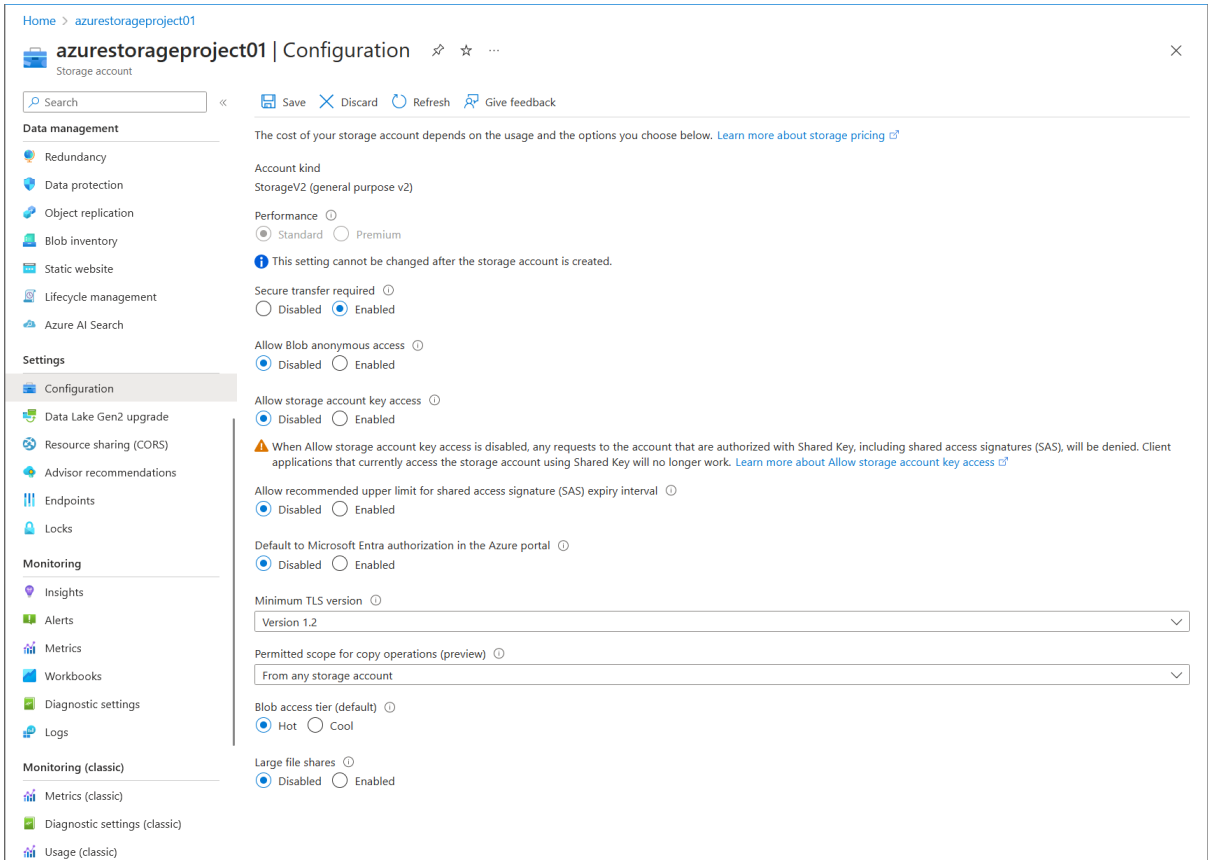
Step 6 - Until the storage is needed again, disable requests to the storage account. Learn more about disabling shared keys

In the **Settings** section, select the **Configuration** blade

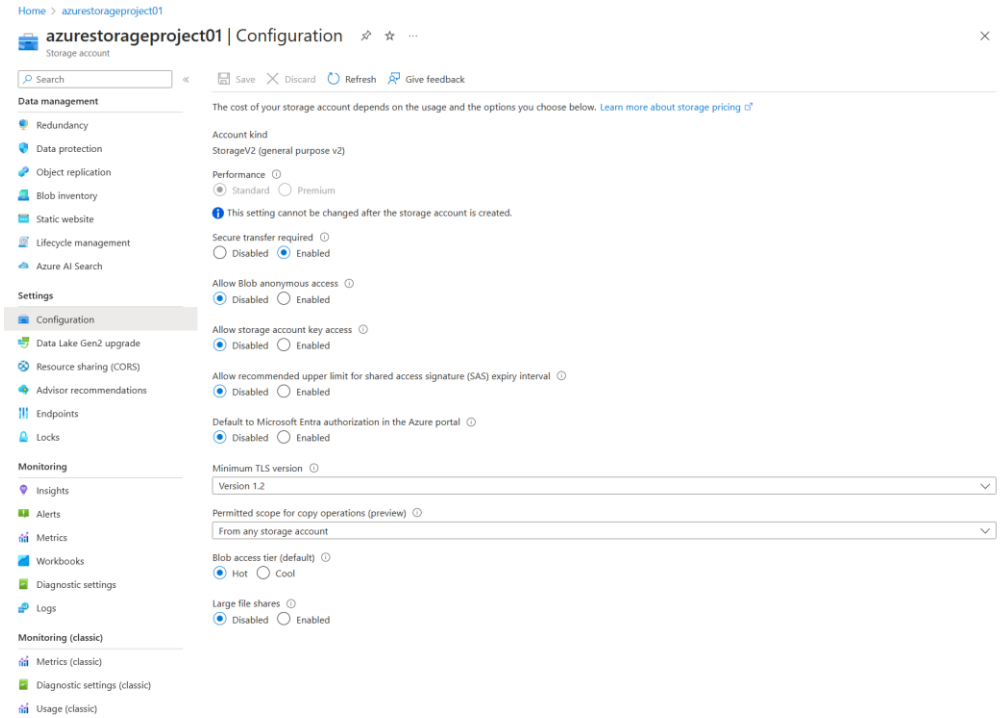


Ensure **Allow storage account key access** is **Disabled**

When Allow 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.

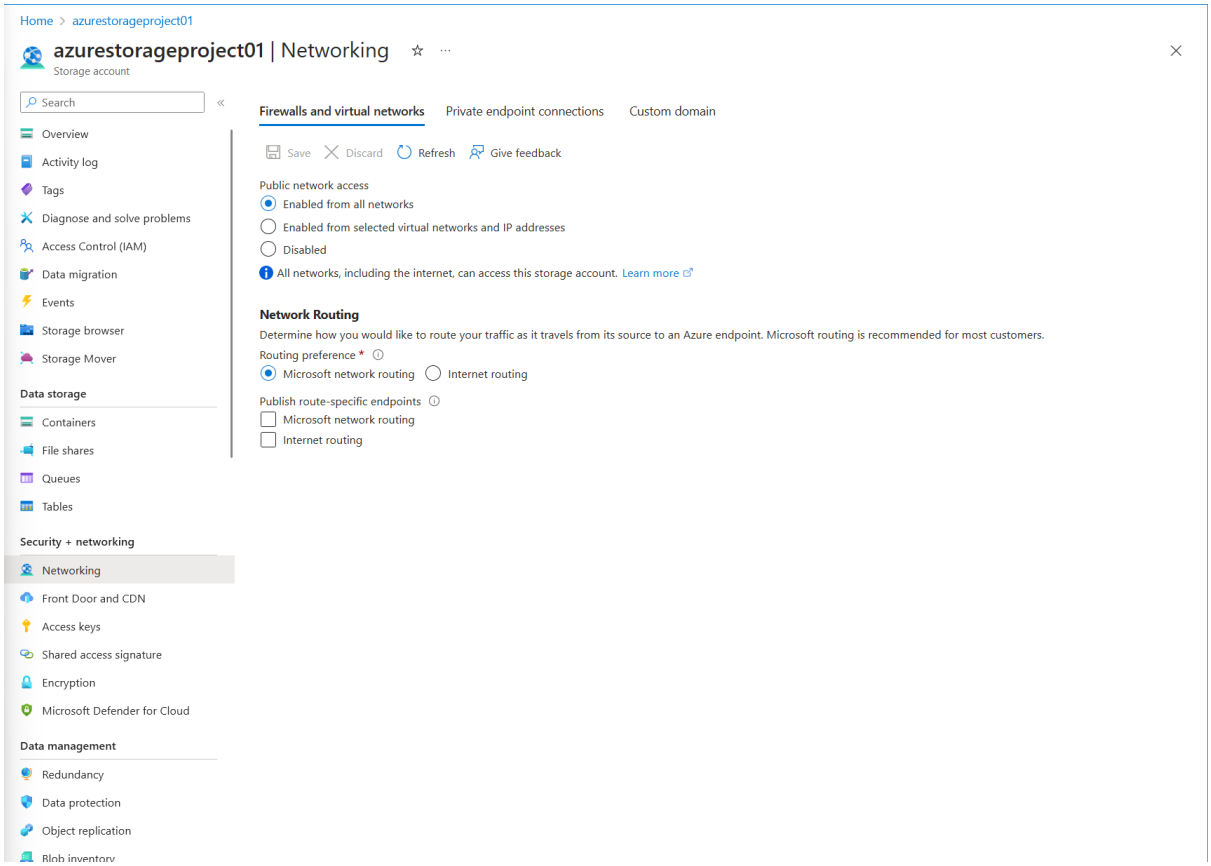


Press save to save settings made to storage account

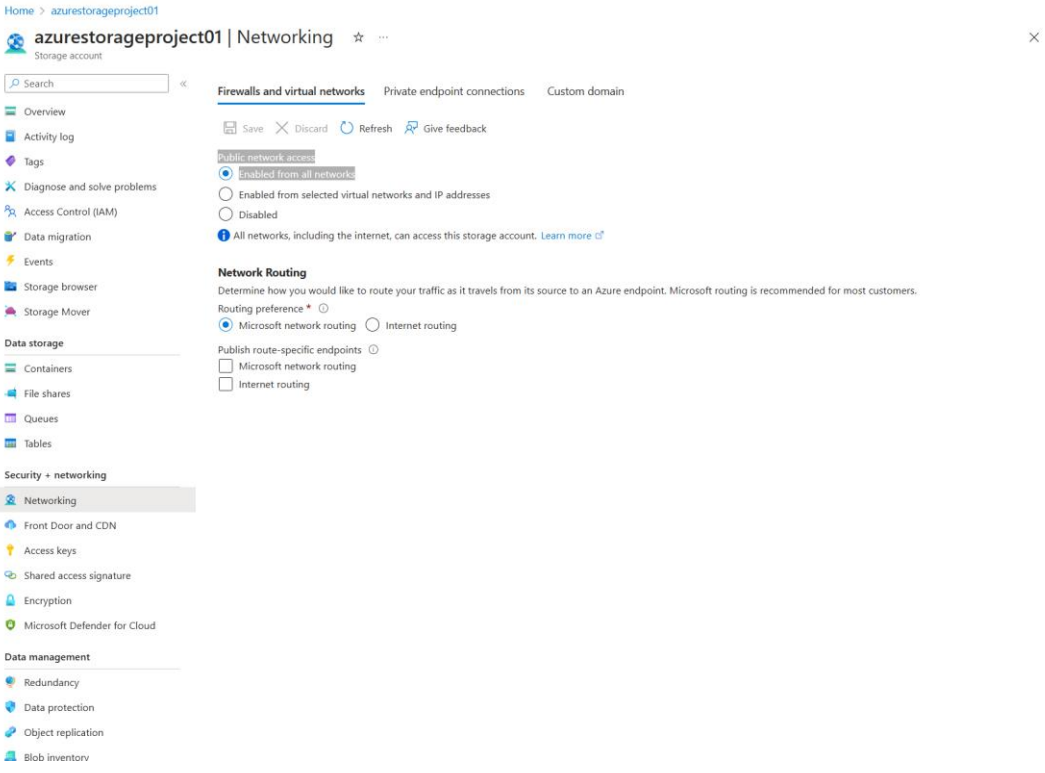


Step 7 - Ensure the storage account allows public access from all networks

In the **Security + networking** section, select the **Networking** blade



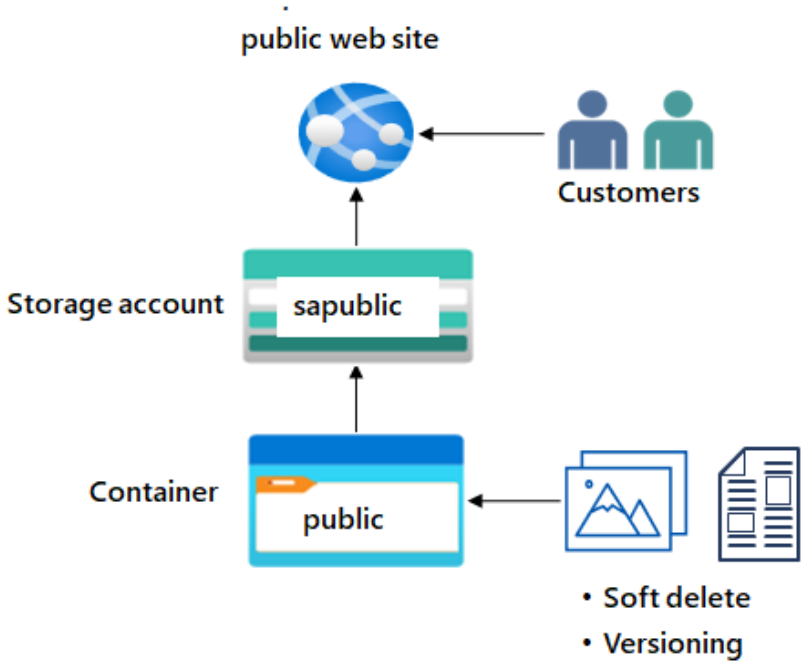
Ensure **Public network access** is set to **Enabled from all networks**



Provide storage for the public website

The company website supplies product images, videos, marketing literature, and customer success stories. Customers are located worldwide and demand is rapidly expanding. The content is mission-critical and requires low latency load times. It's important to keep track of the document versions and to quickly restore documents if they're deleted.

Architecture diagram

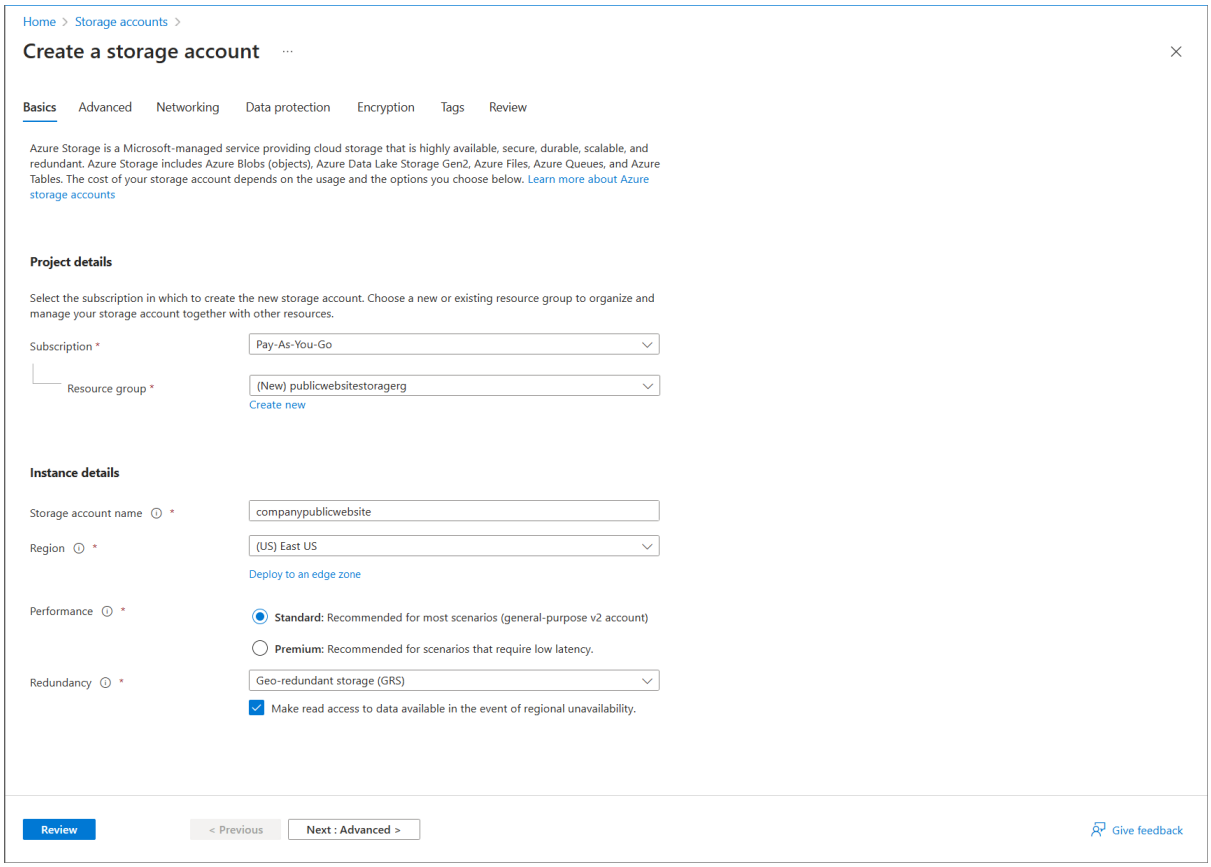


Task to be completed

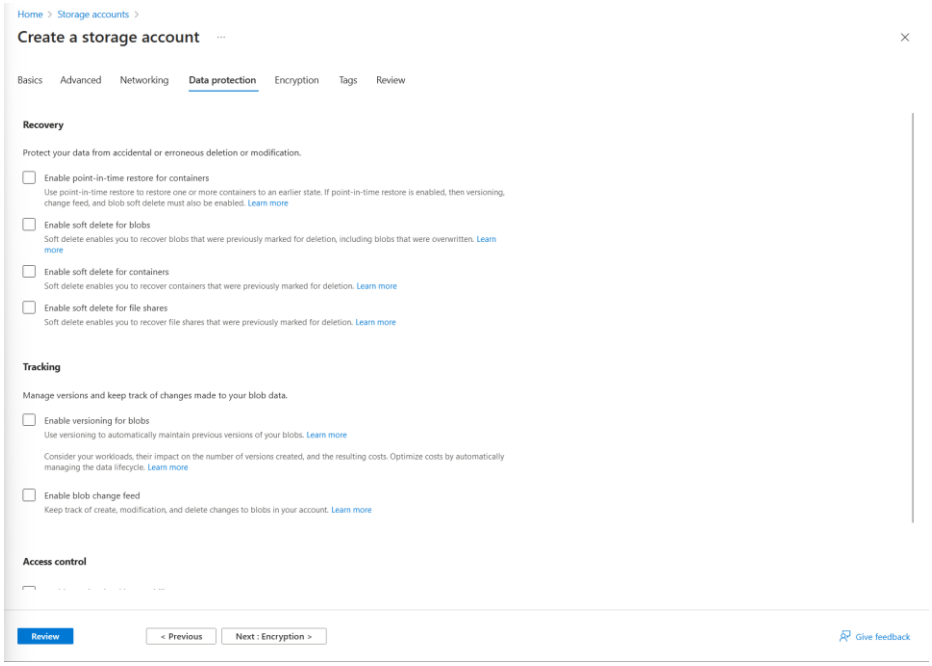
- Create a storage account with high availability.
- Ensure the storage account has anonymous public access.
- Create a blob storage container for the website documents.
- Enable soft delete so files can be easily restored.
- Enable blob versioning.

Step 1 - Create a storage account with high availability

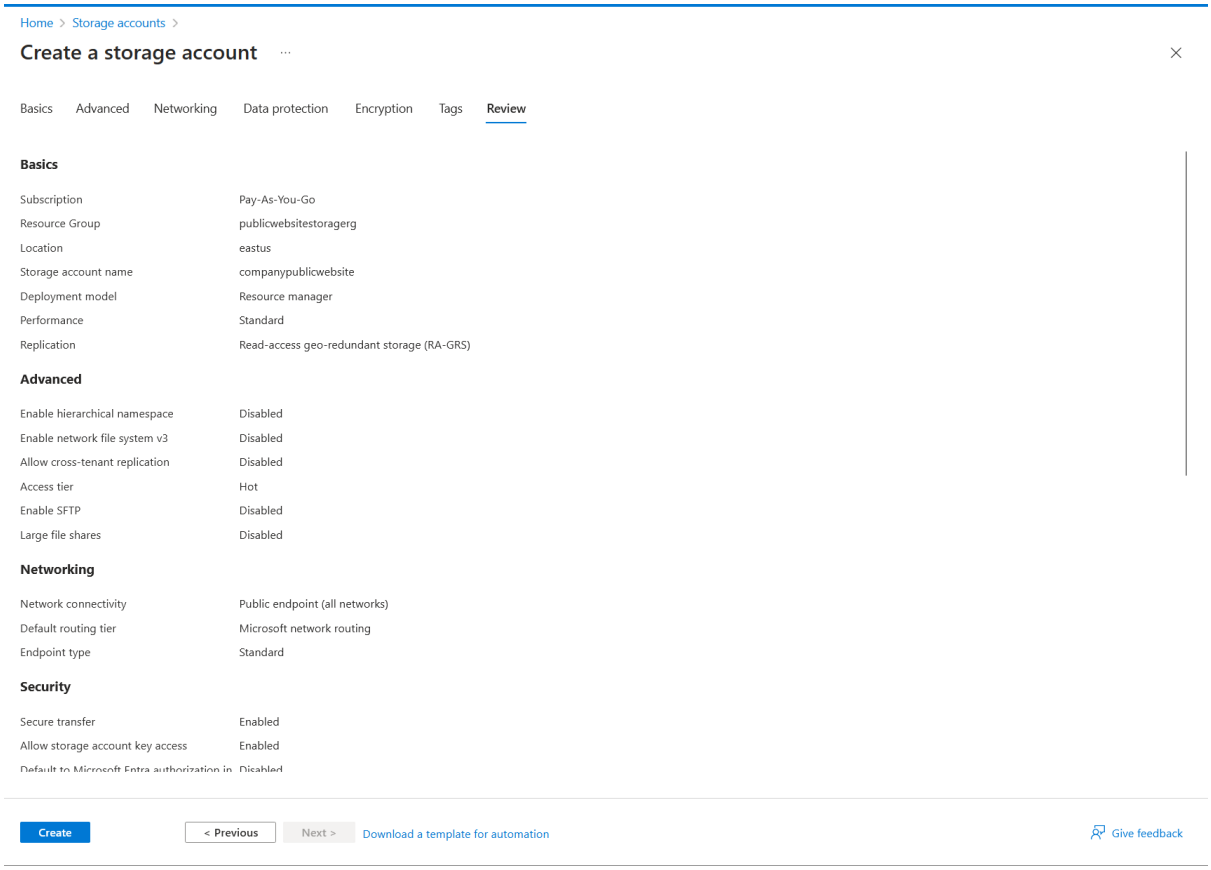
Create a storage account with the following settings to support the public website



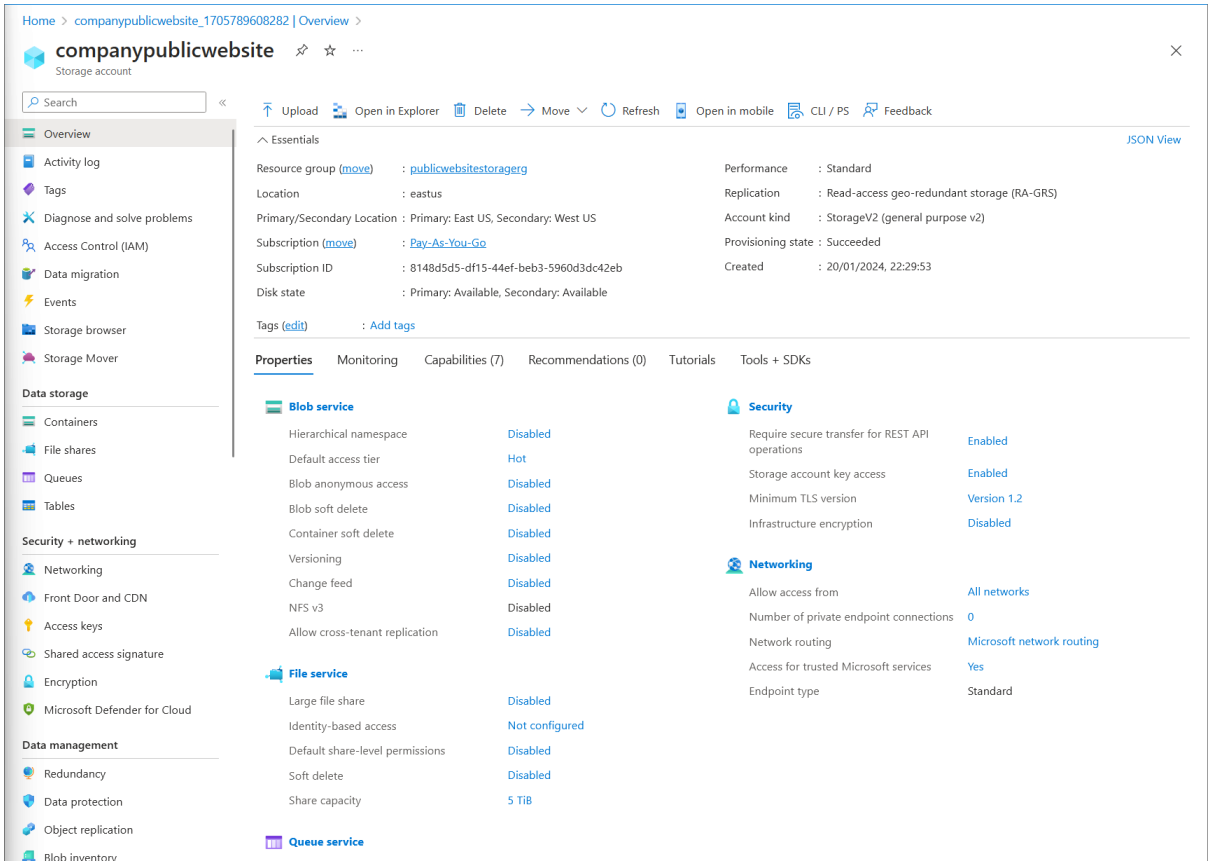
Disable all data protection this can be enable later.



All other settings default Click Review and create.

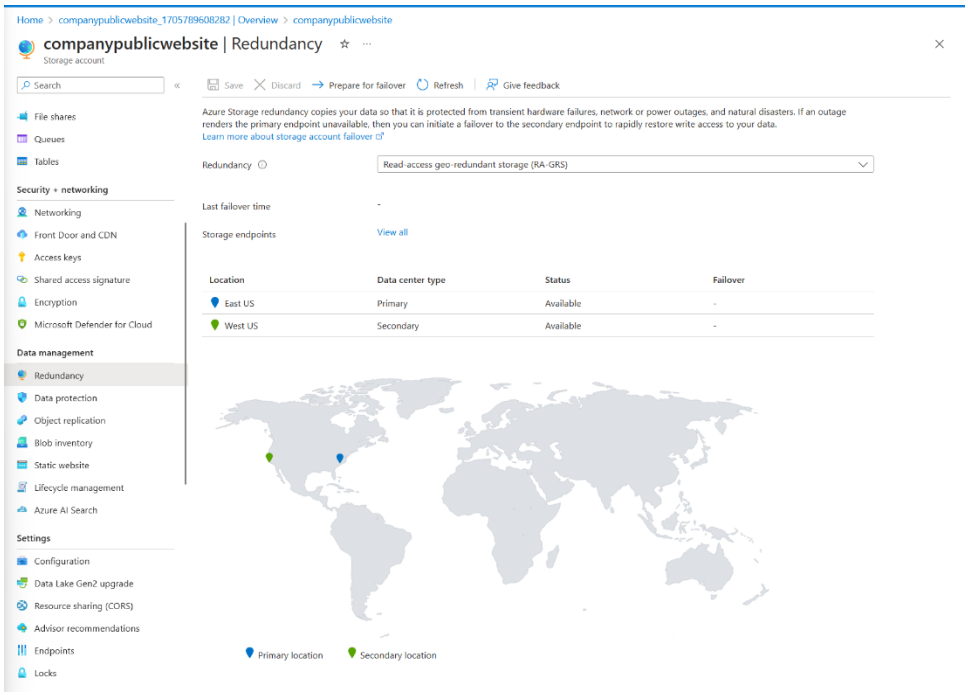


One storage account is deployed, select **Go to resource**

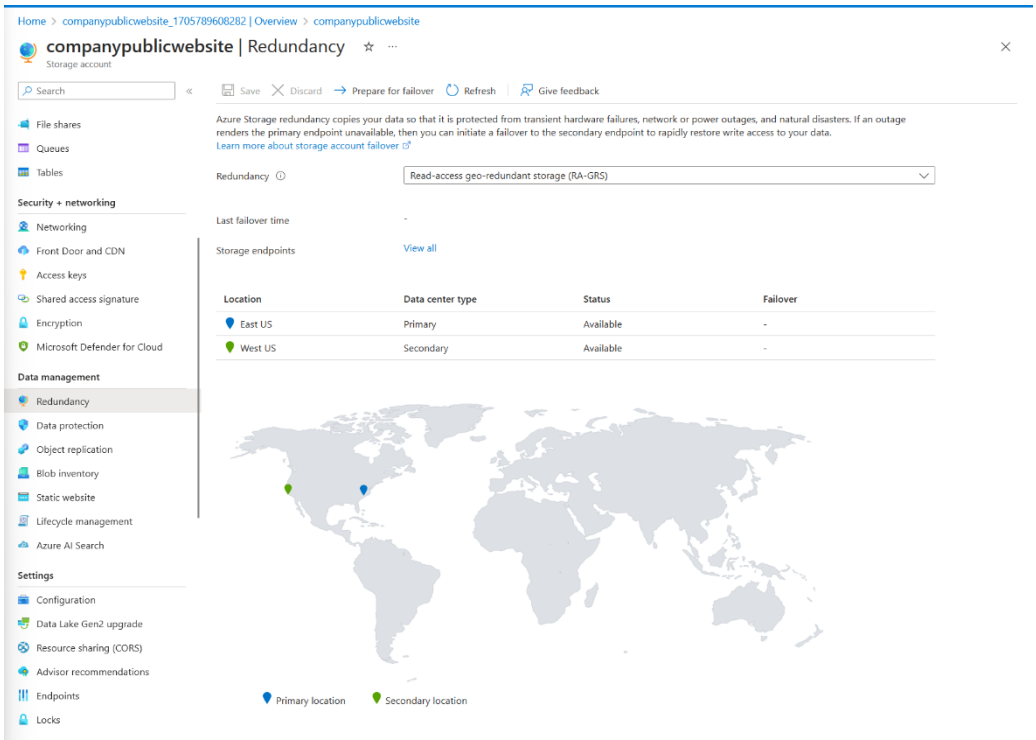


Step 2 - This storage requires high availability if there's a regional outage. Additionally, enable read access to the secondary region

In the storage account, in the **Data management** section, select the **Redundancy** blade



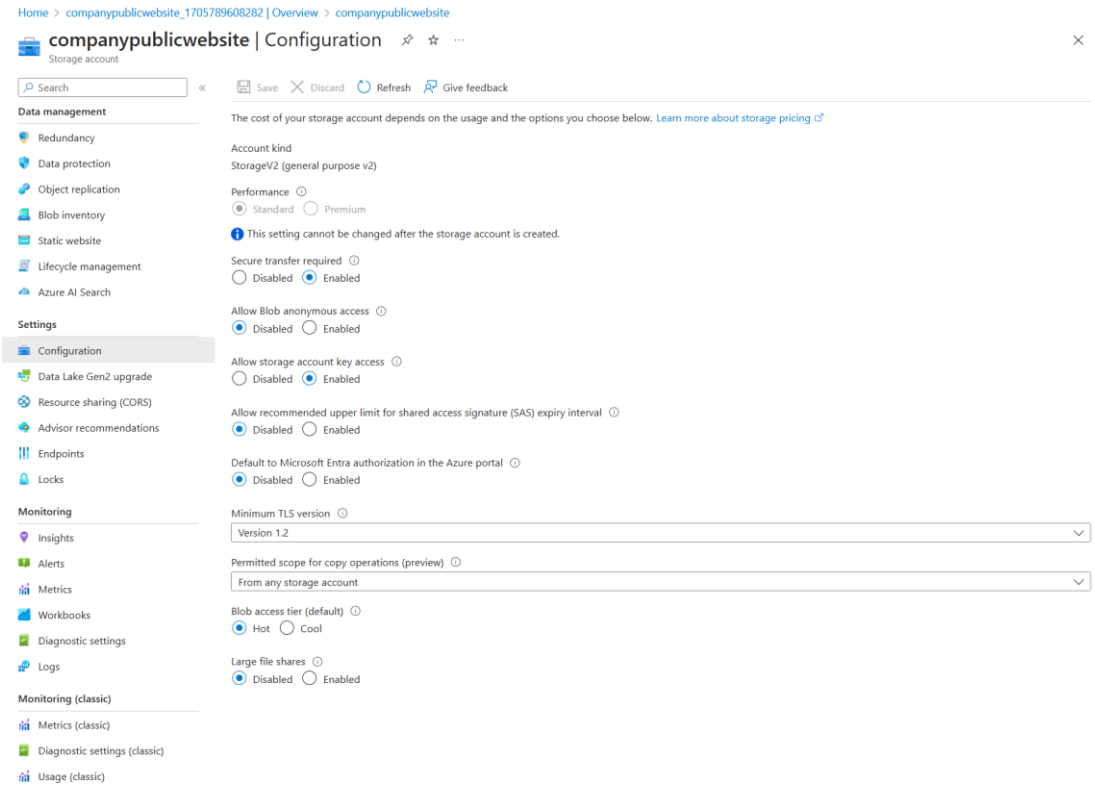
Ensure **Read-access Geo-redundant storage** is selected



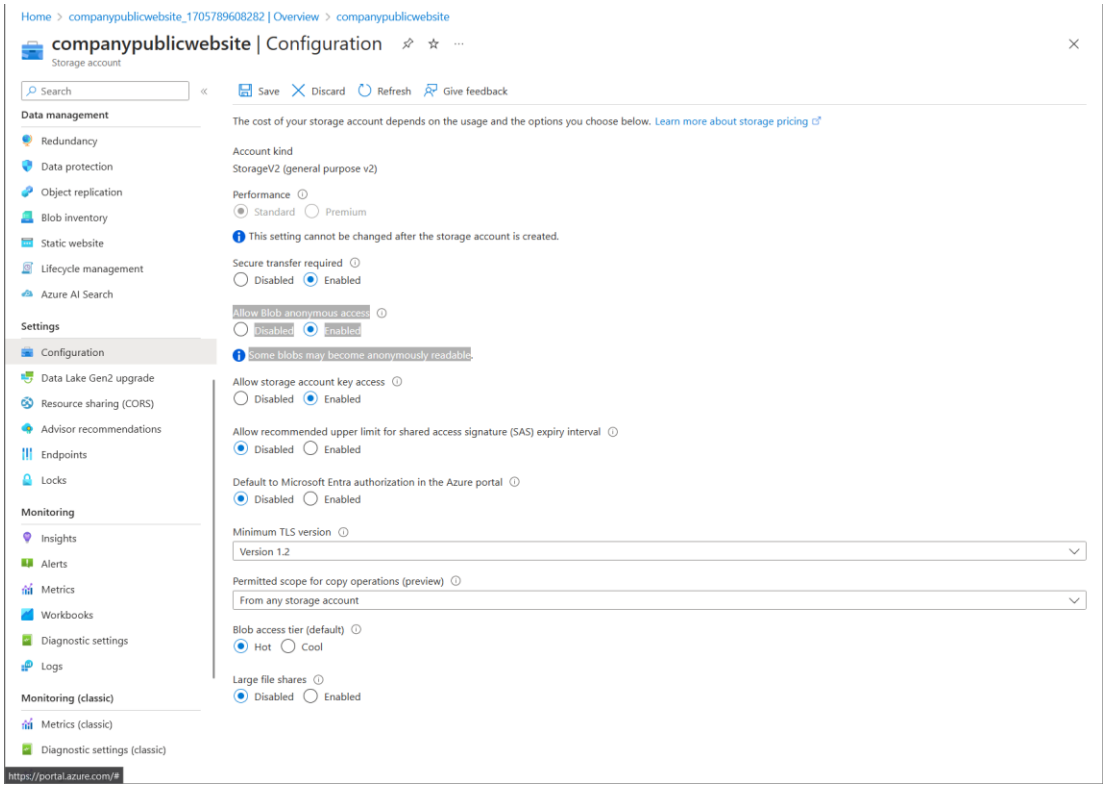
Primary Reason set to East US and secondary region set to West US

Step 3 - Information on the public website should be accessible without requiring customers to login

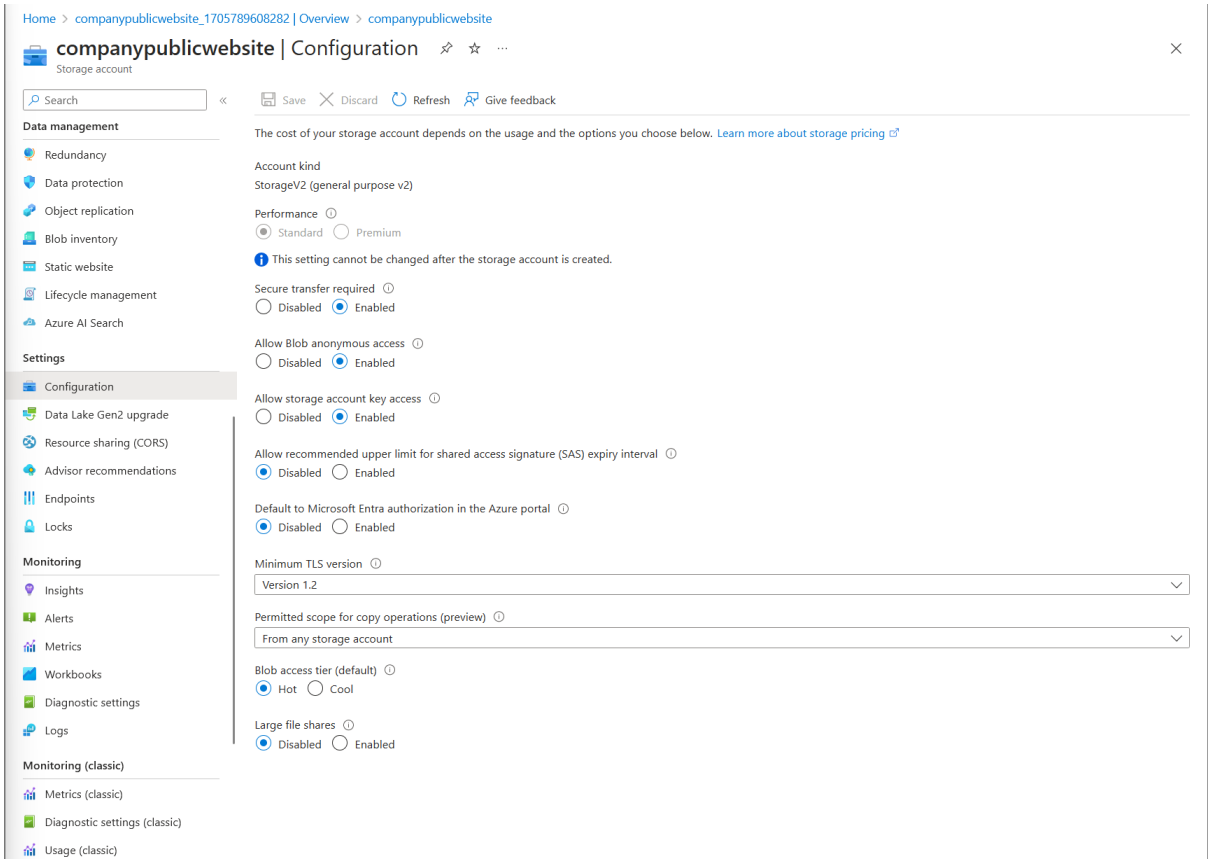
In the storage account, in the **Settings** section, select the **Configuration** blade



Ensure the **Allow blob anonymous access** setting is **Enabled**

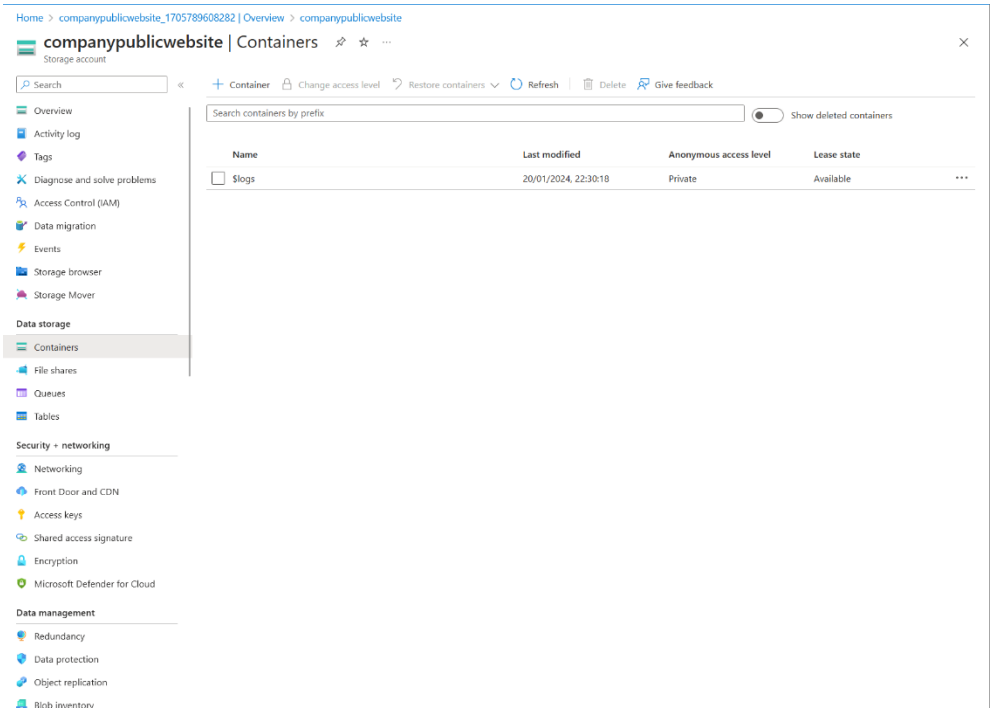


Click Save to save settings for allow blob anonymous access



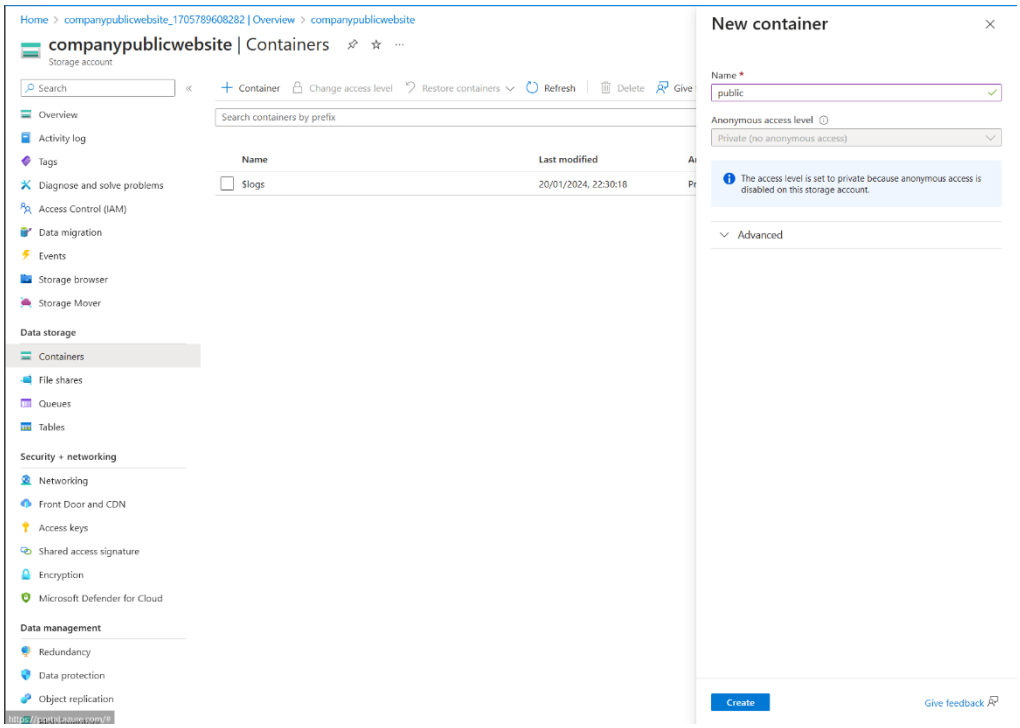
Step 4 - Create a blob storage container with anonymous read access

In the **Data storage** section, select the **Containers** blade

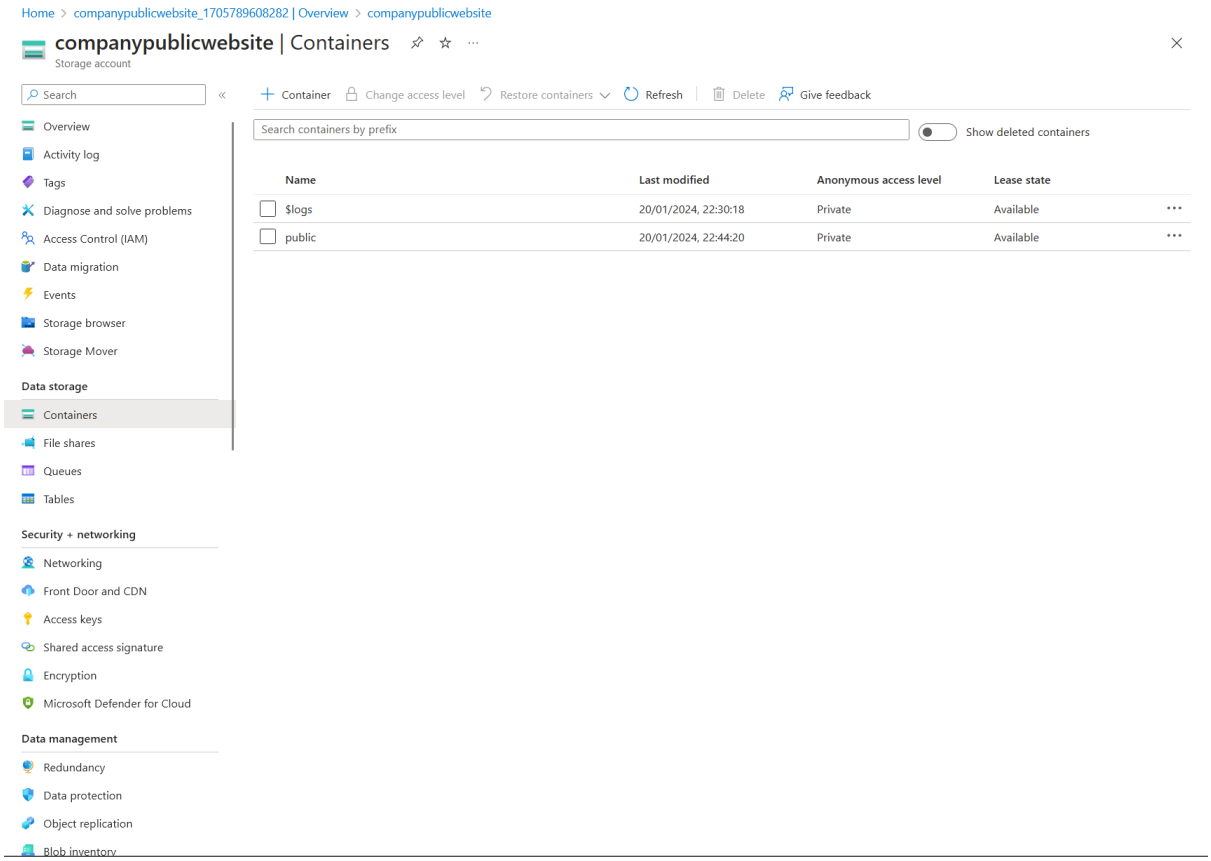


Select **+ Container**

Ensure the **Name** of the container is public

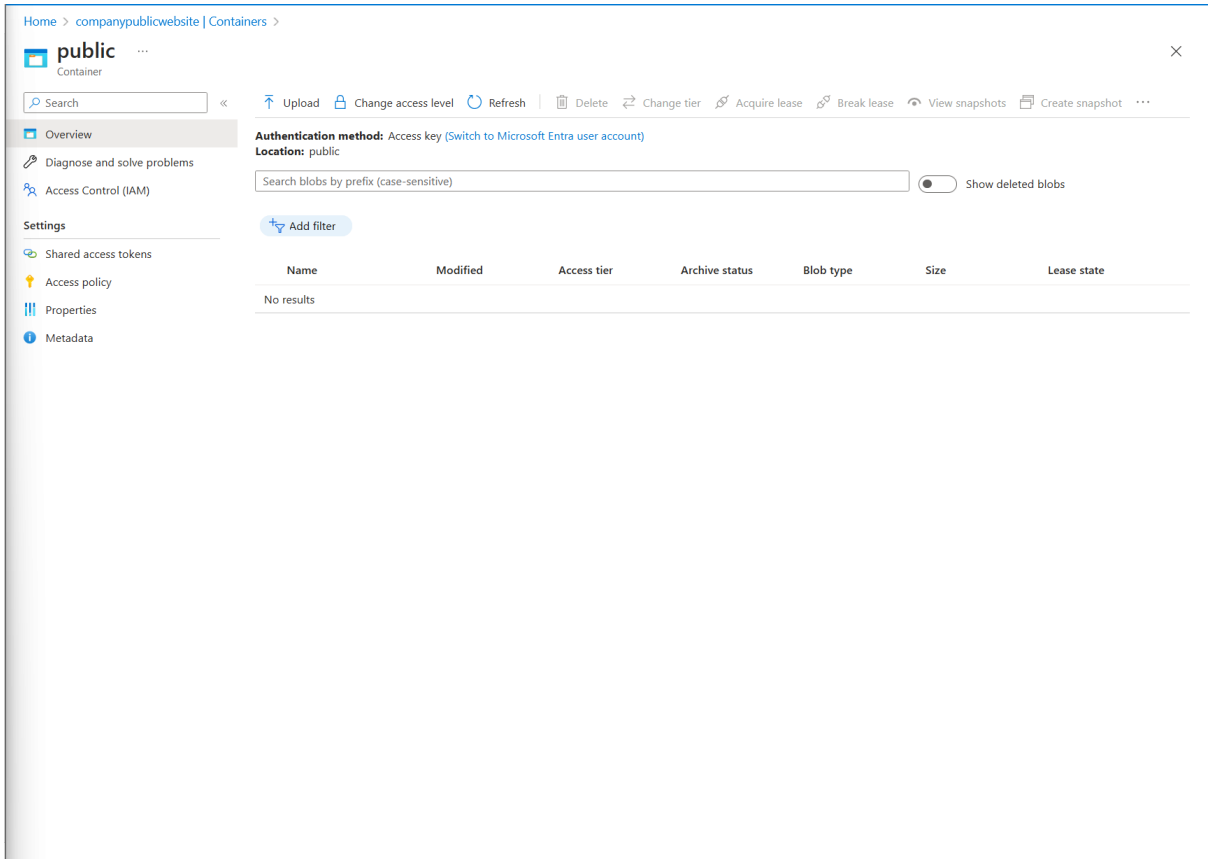


Click create to create the public container



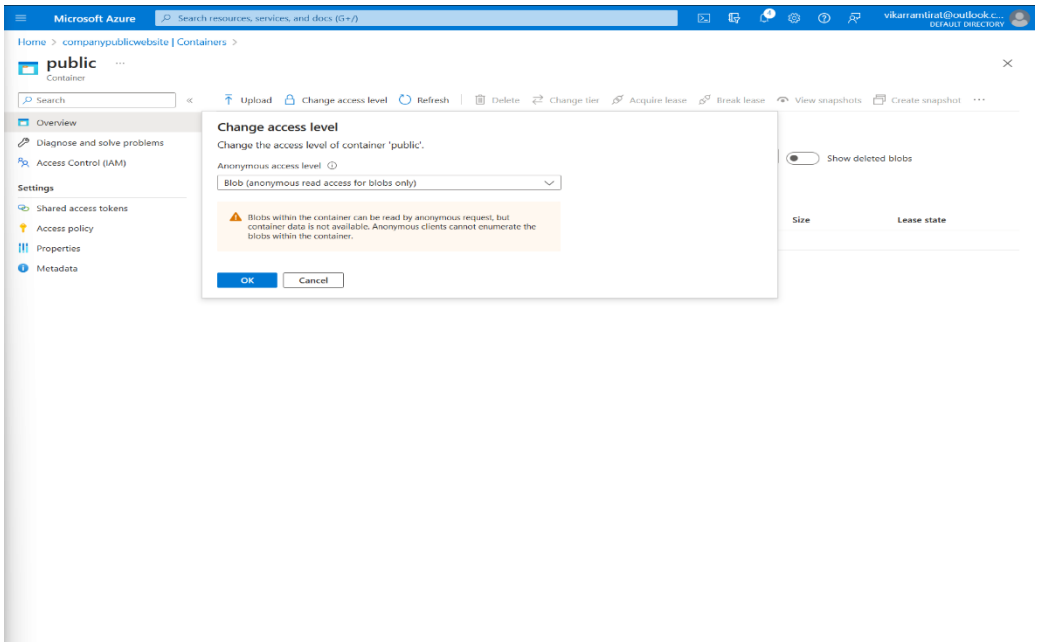
Step 5 - Customers should be able to view the images without being authenticated. Configure anonymous read access for the public container blobs.

Select the **public** container



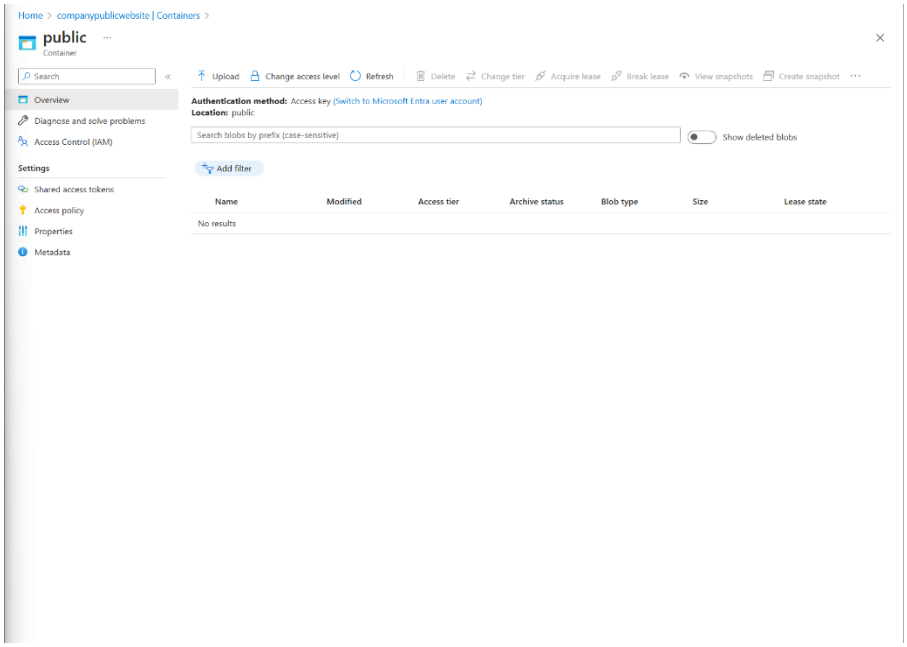
On the **Overview** blade, select **Change access level**

Ensure the **Public access level** is **Blob (anonymous read access for blobs only)**. Select **OK** to Save

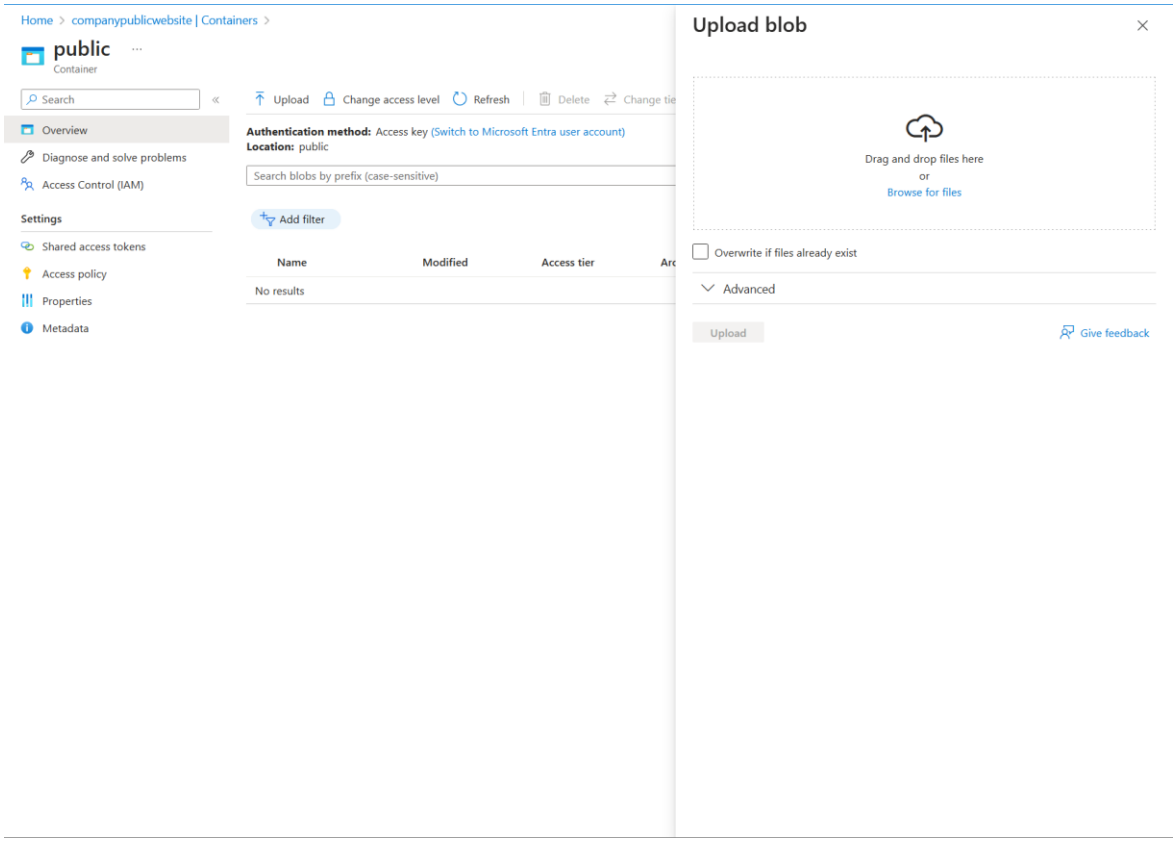


Step 6 - Uploading files and testing access

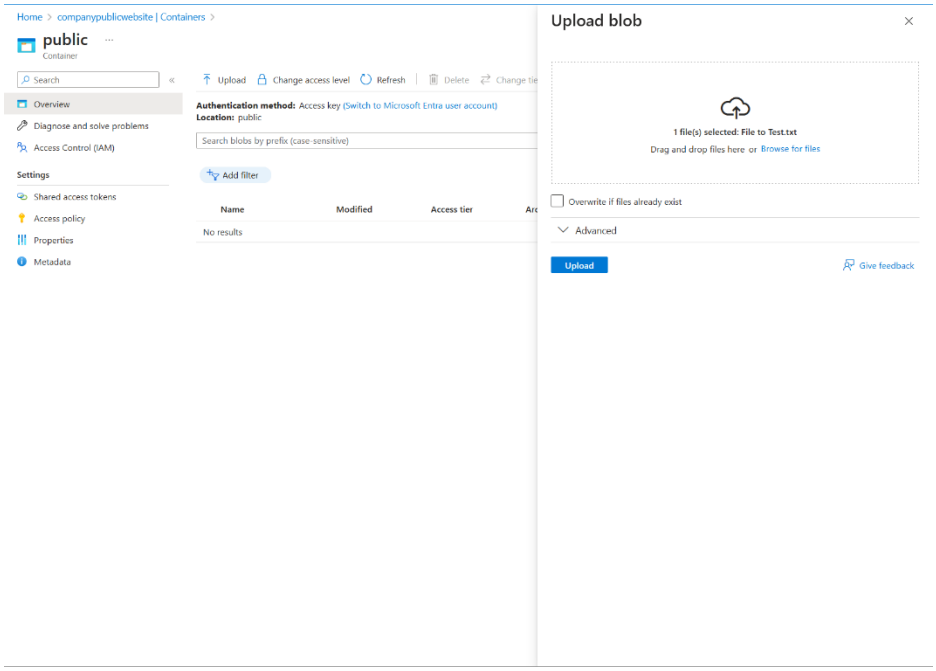
Ensure you are viewing your container



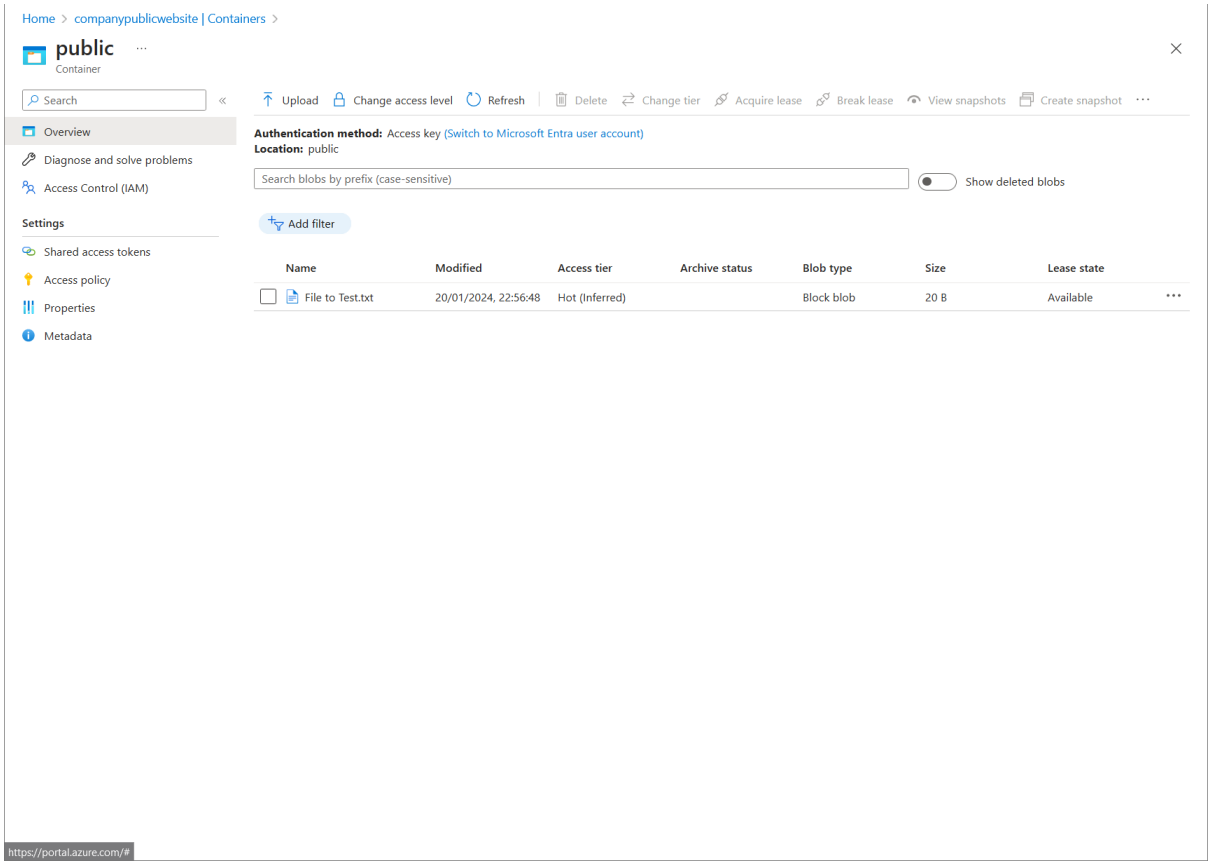
Select **Upload**



Browse to files and select a file. Browse to a file of your choice

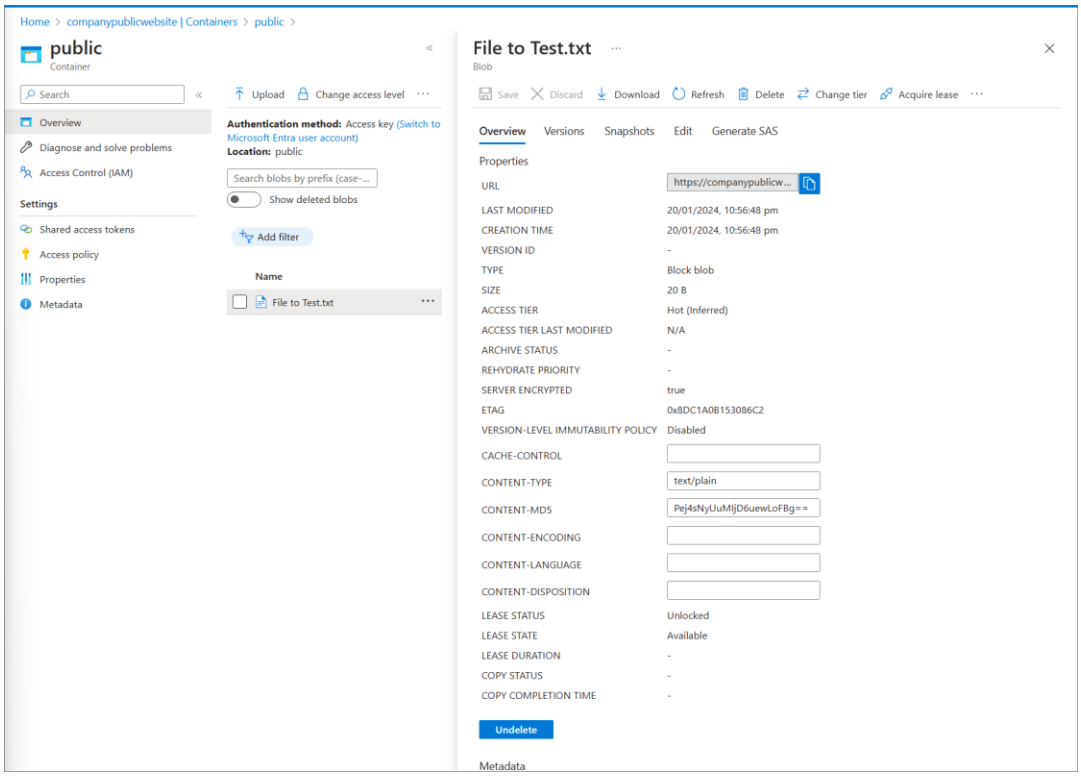


Select **Upload**

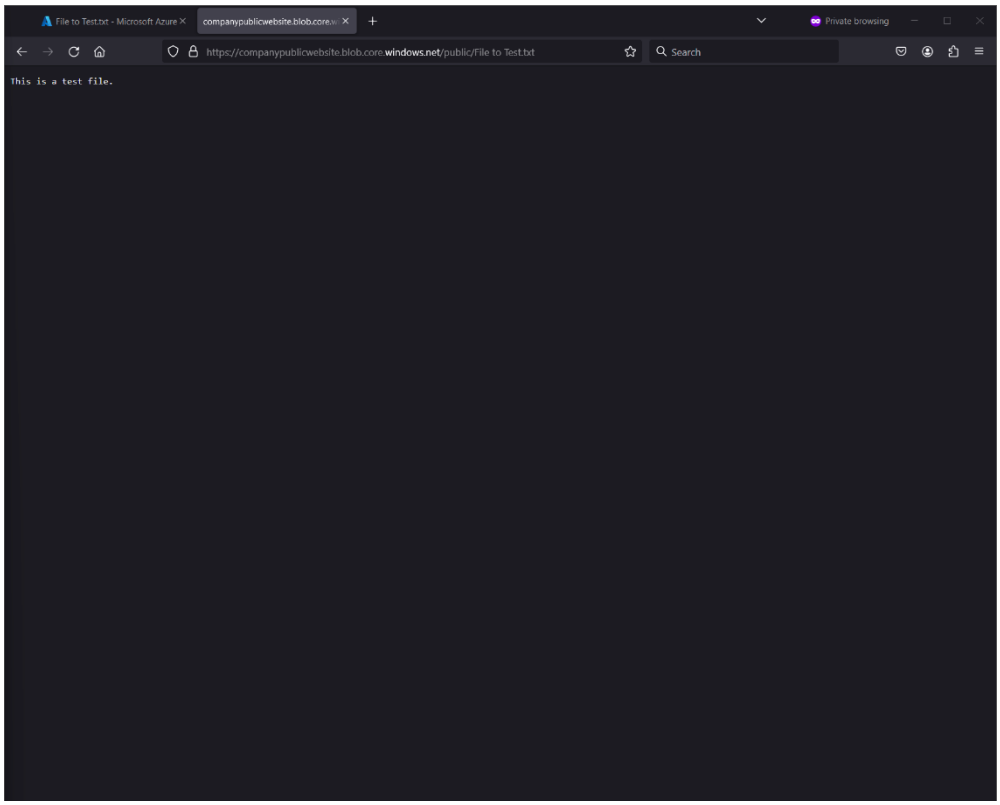


Step 7 - Determine the URL for your uploaded file. Open a browser and test the URL

Select the uploaded file On the **Overview** tab, copy the **URL**



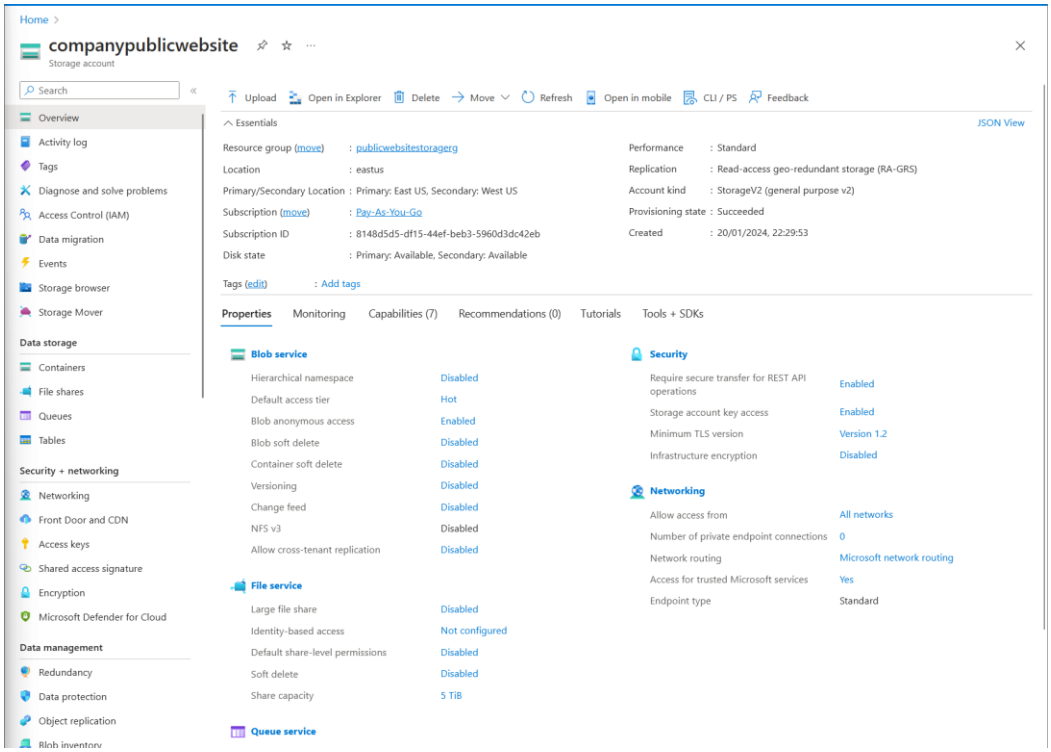
Paste the URL into a new browser tab



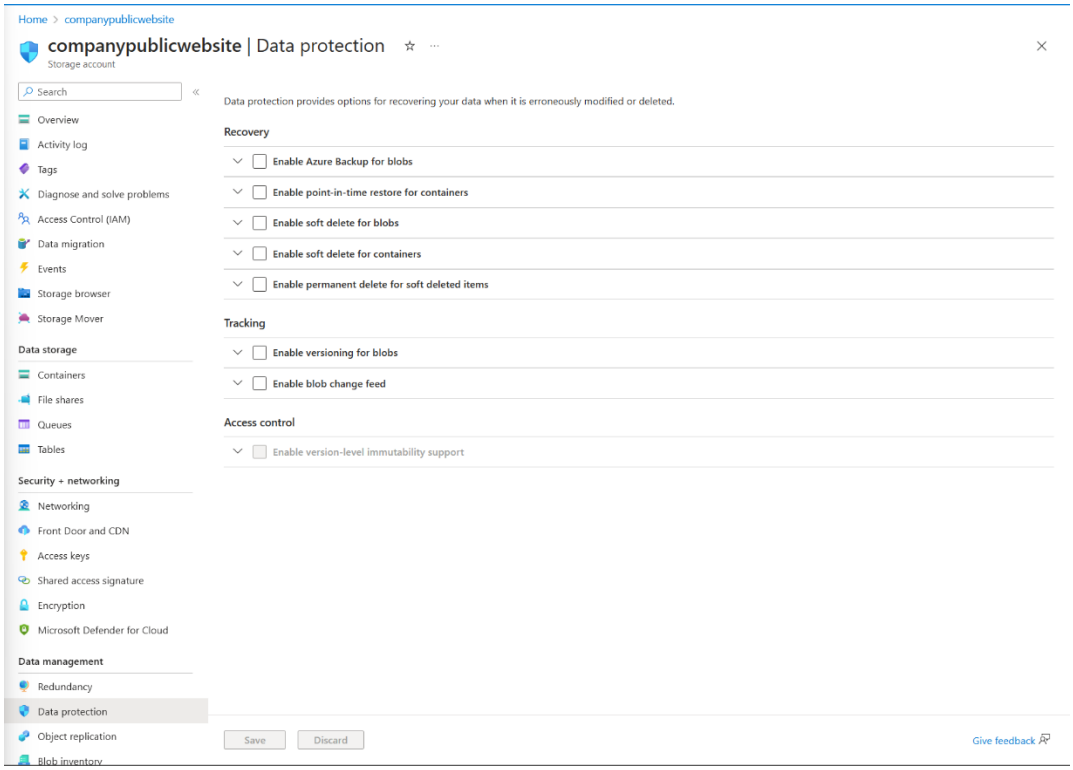
Step 8 - Configure soft delete

It's important that the website documents can be restored if they're deleted. Configure blob soft delete for 21 days.

Go to the **Overview** blade of the **storage account**



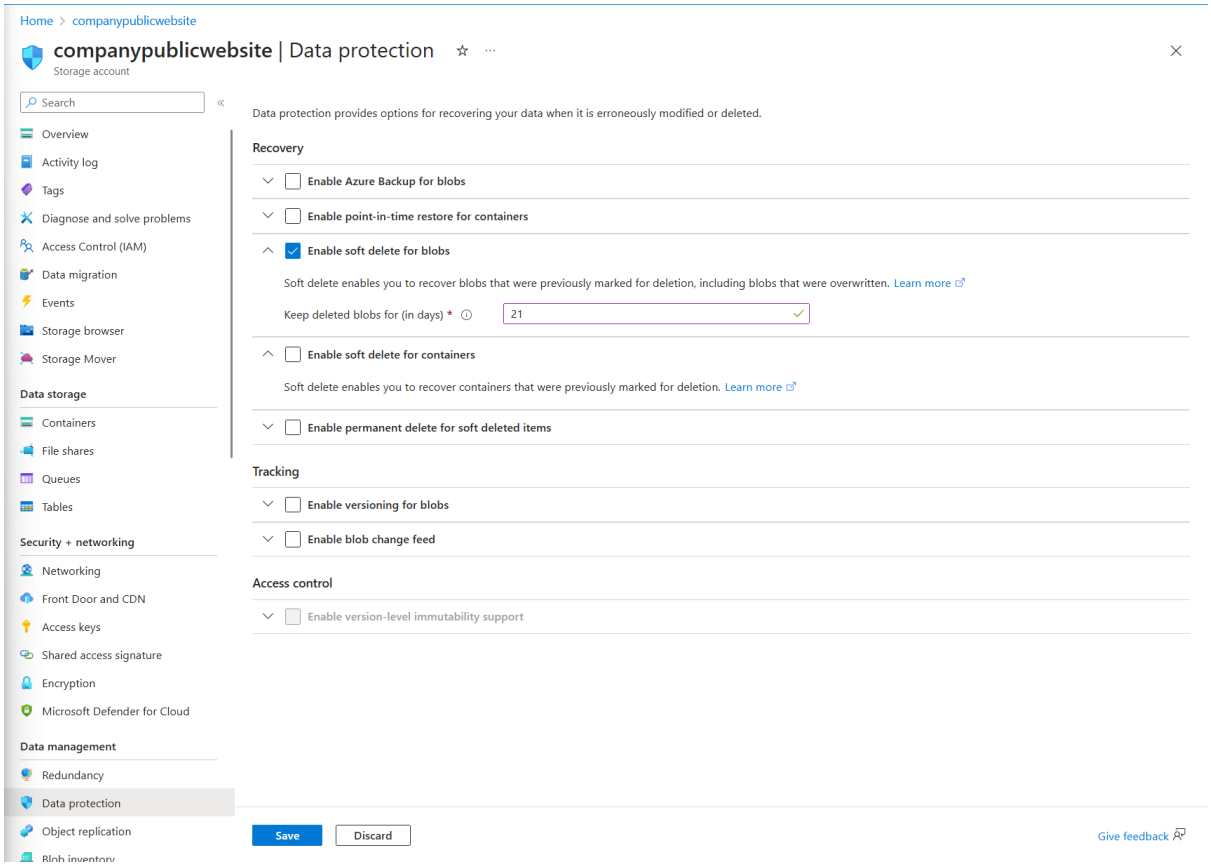
On the **Properties** page, locate the **Blob service** section Select the **Blob soft delete** setting



Ensure the **Enable soft delete for blobs** is **checked**

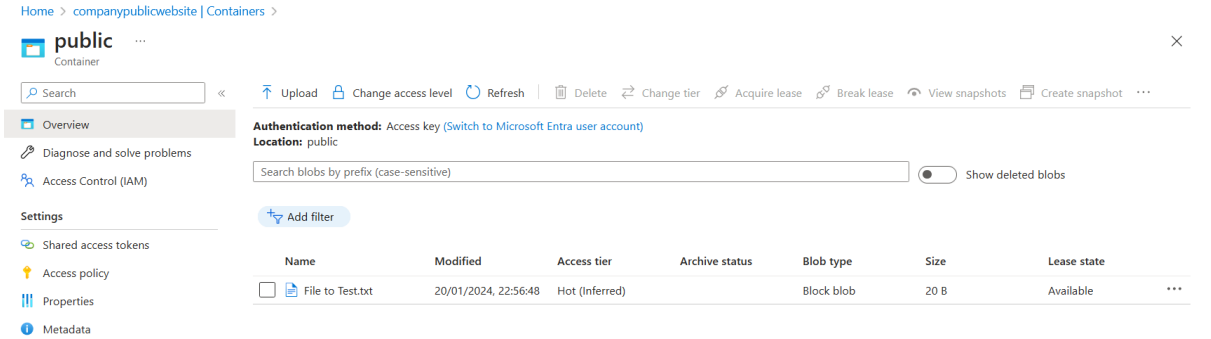
Change the **Keep deleted blobs for (in days** setting is **21**

Notice you can also **Enable soft delete for containers** leave this option unchecked and click save



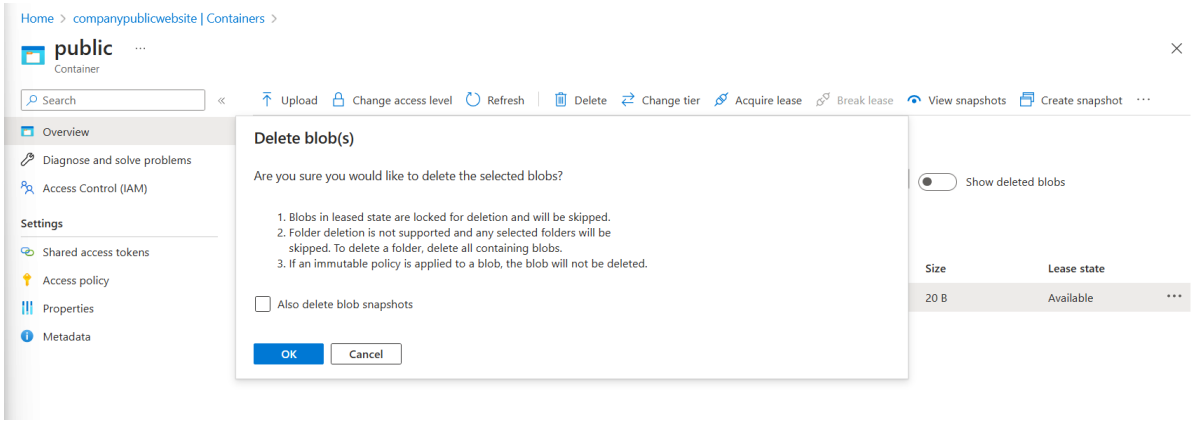
Test is soft delete is working

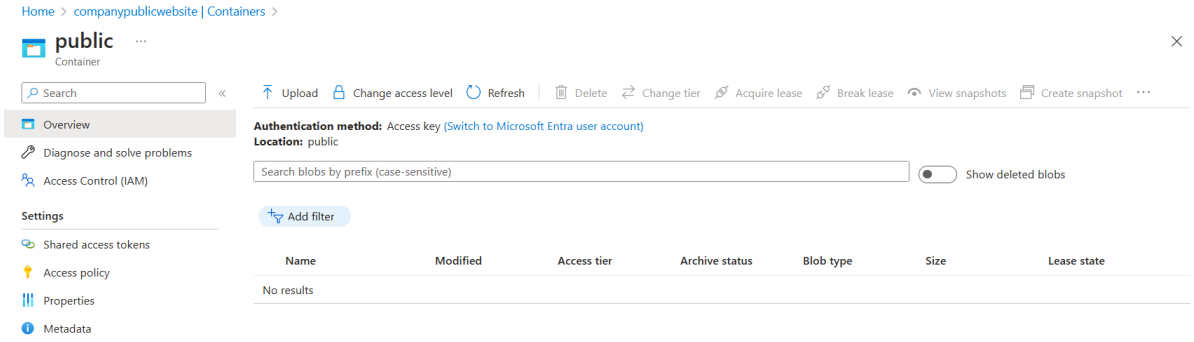
Navigate to the container where you uploaded a file



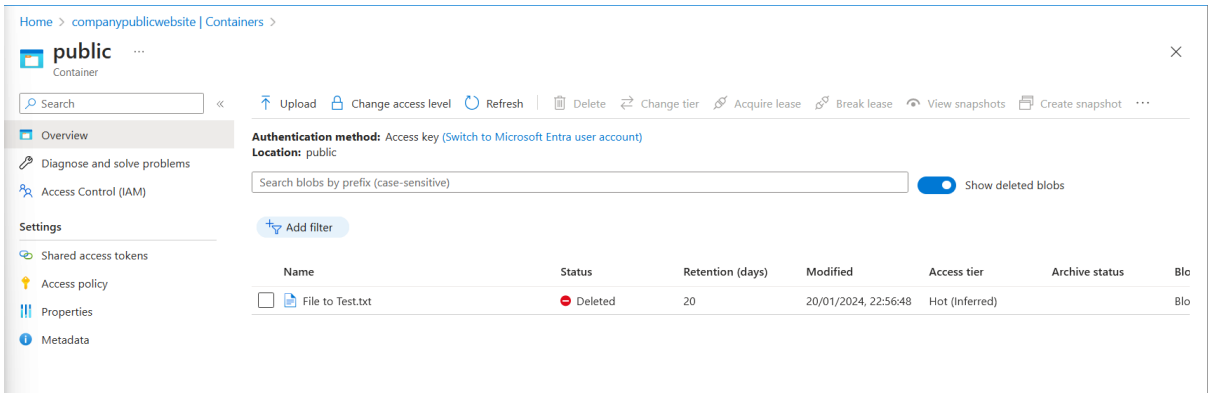
Select the file uploaded and then select **Delete**

Select **OK** to confirm deleting the file

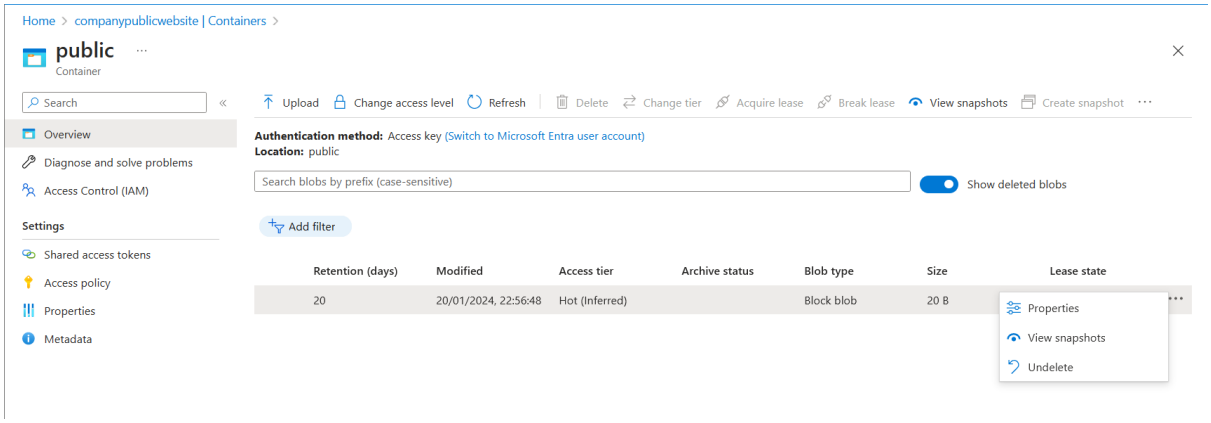




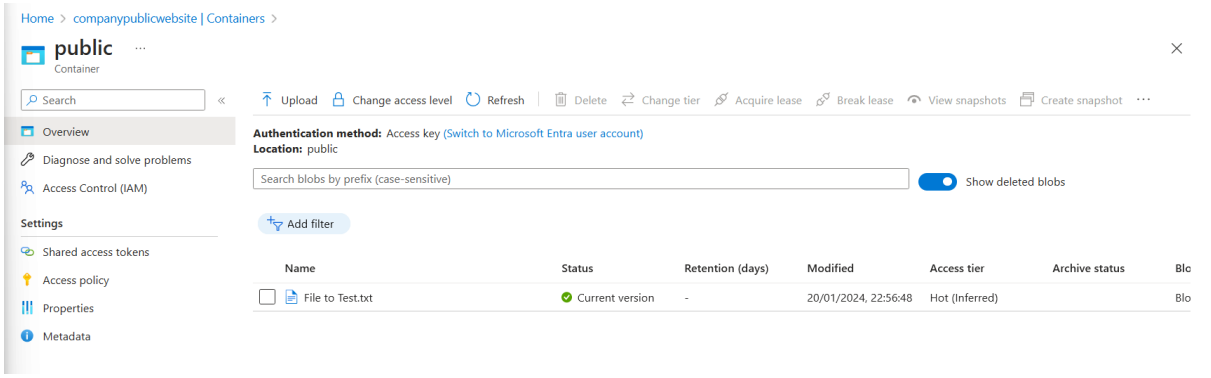
On the container **Overview** page, toggle the slider **Show deleted blobs**. This toggle is to the right of the search box



Select deleted file, and use the ellipses on the far right, to **Undelete** the file



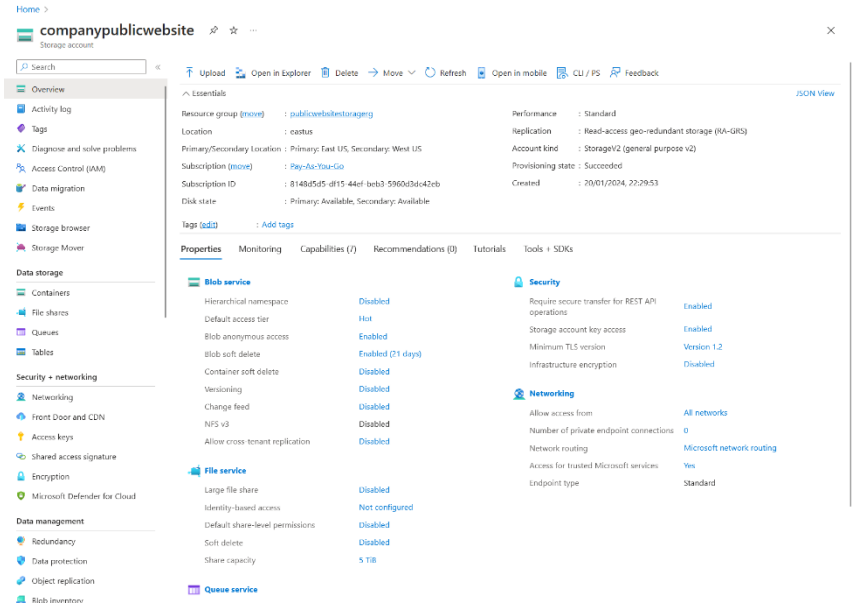
Refresh the container and confirm the file has been restored



Step 9 - Configure blob versioning

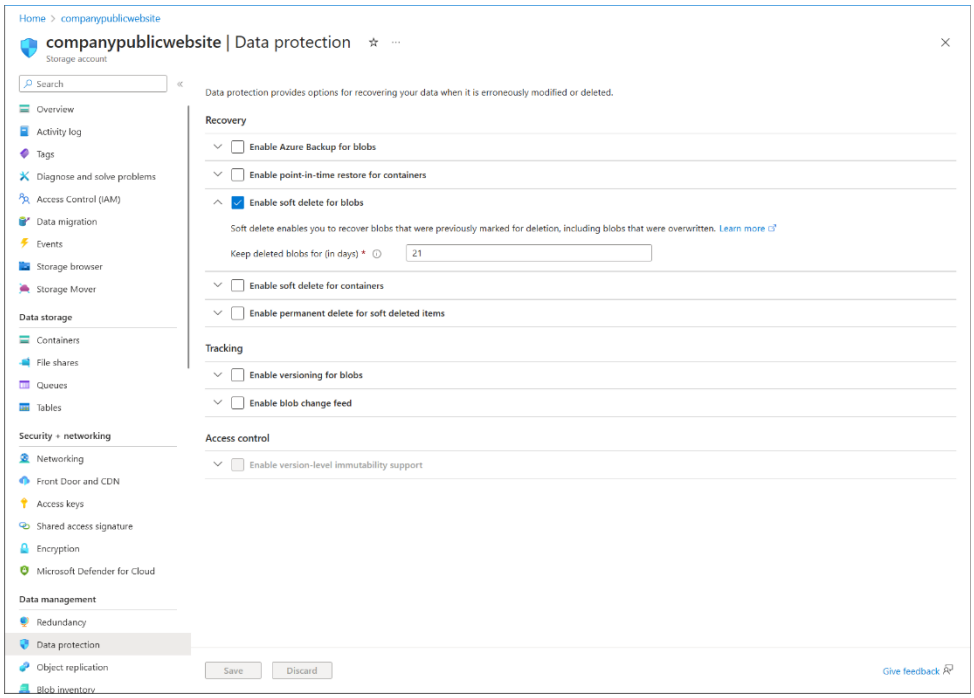
It’s important to keep track of the different website product document versions

Go to the **Overview** blade of the **storage account**



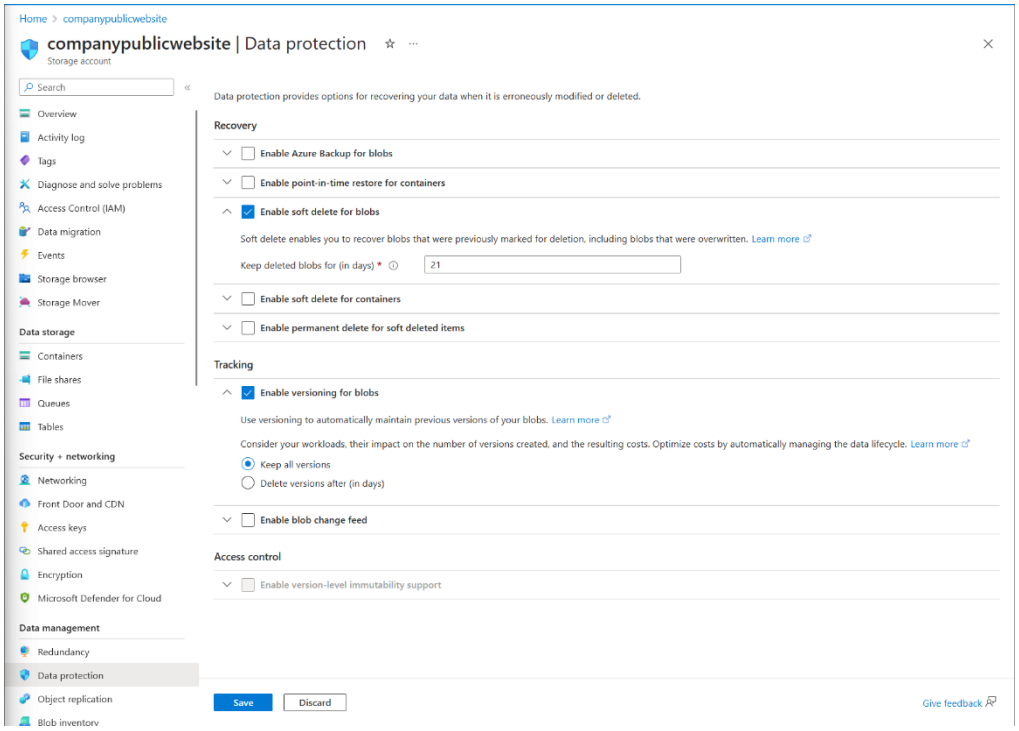
In the **Properties** section, locate the **Blob service** section

Select the **Versioning** setting



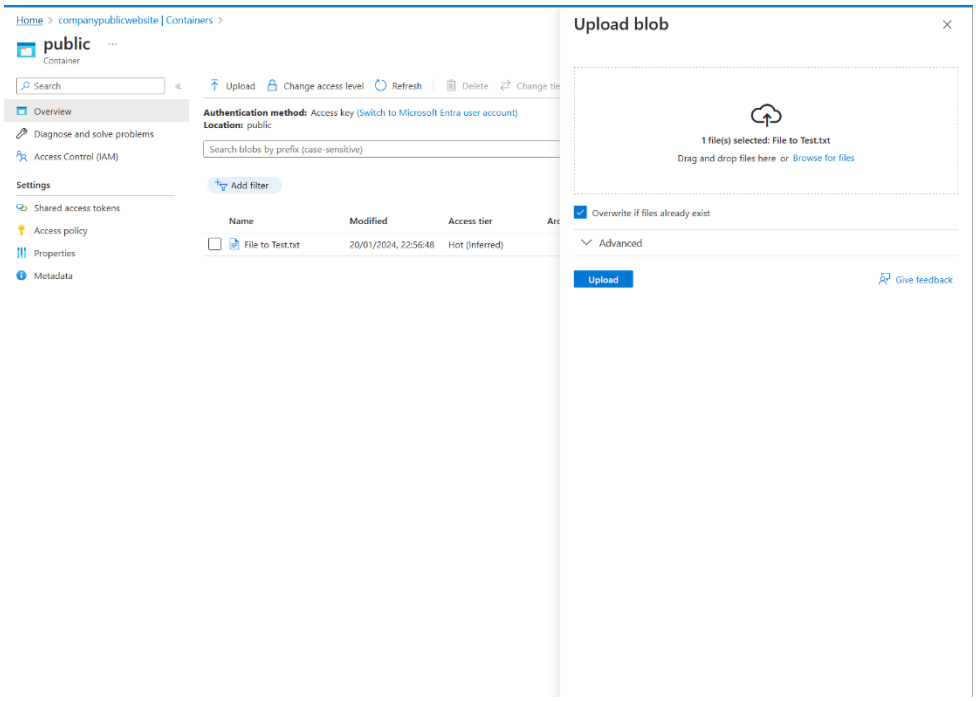
Ensure the **Enable versioning for blobs** checkbox is checked

Options to **keep all versions** or **delete versions after** select Keep all versions

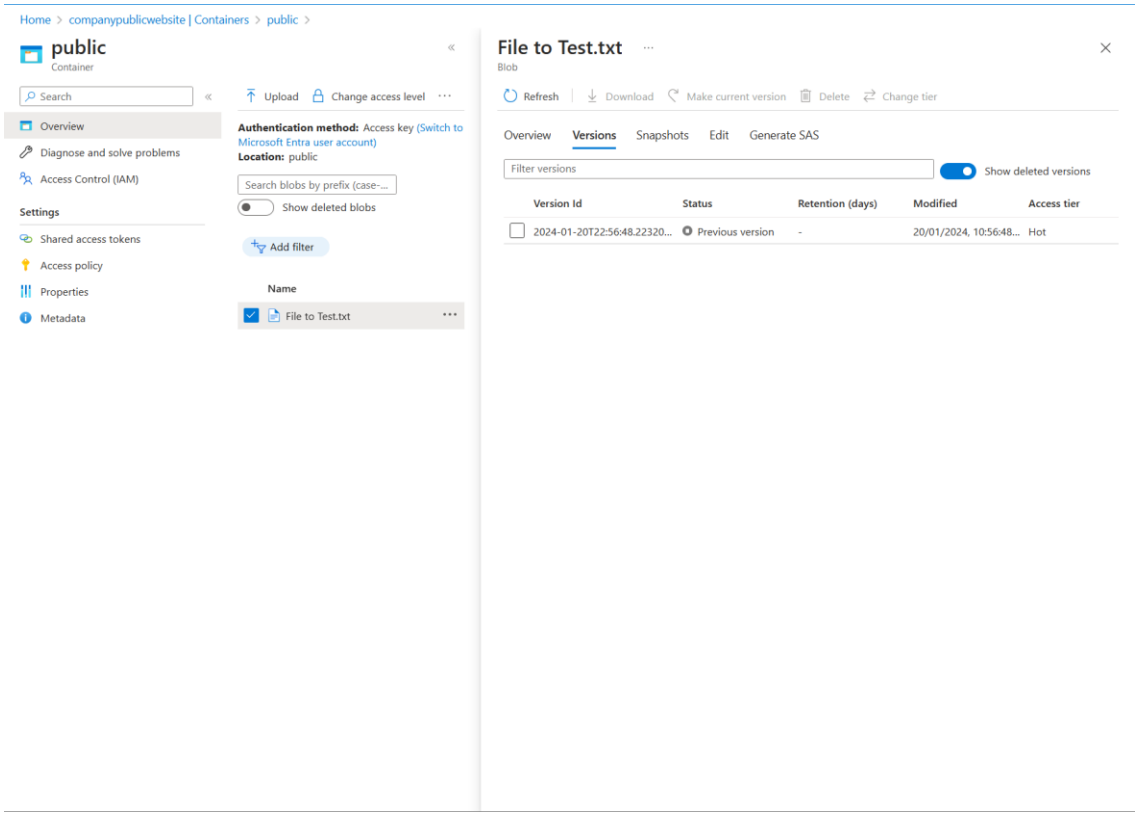


Experiment with restoring previous blob versions

Upload another version of your container file. This overwrites your existing file



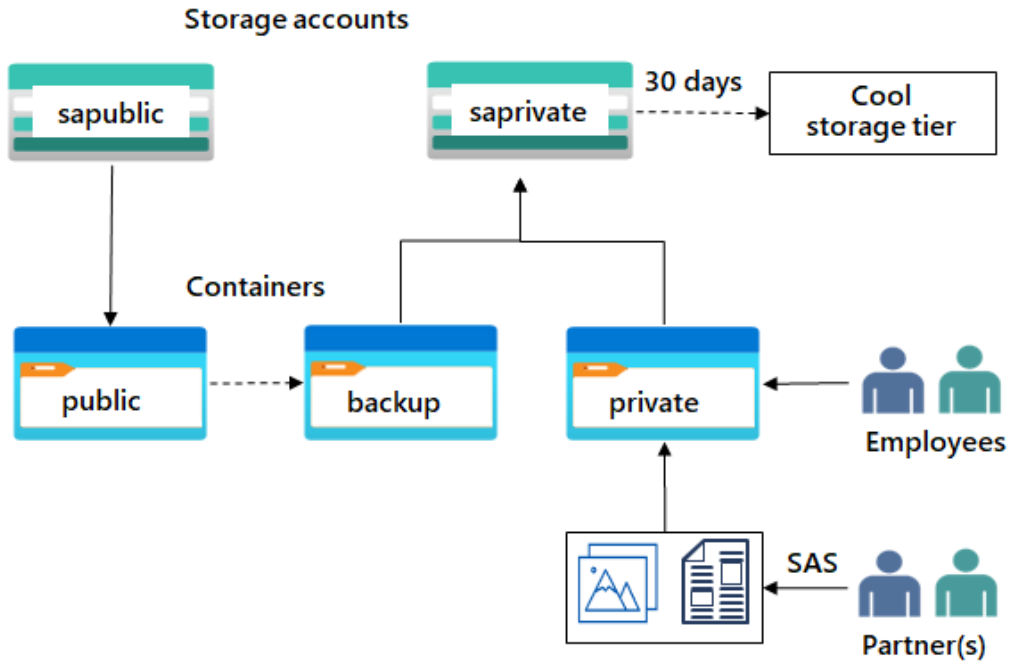
Your previous file version is listed on **Show deleted blobs** page



Provide private storage for internal company documents

The company needs storage for their offices and departments. This content is private to the company and shouldn't be shared without consent. This storage requires high availability if there's a regional outage. The company wants to use this storage to back up the public website storage.

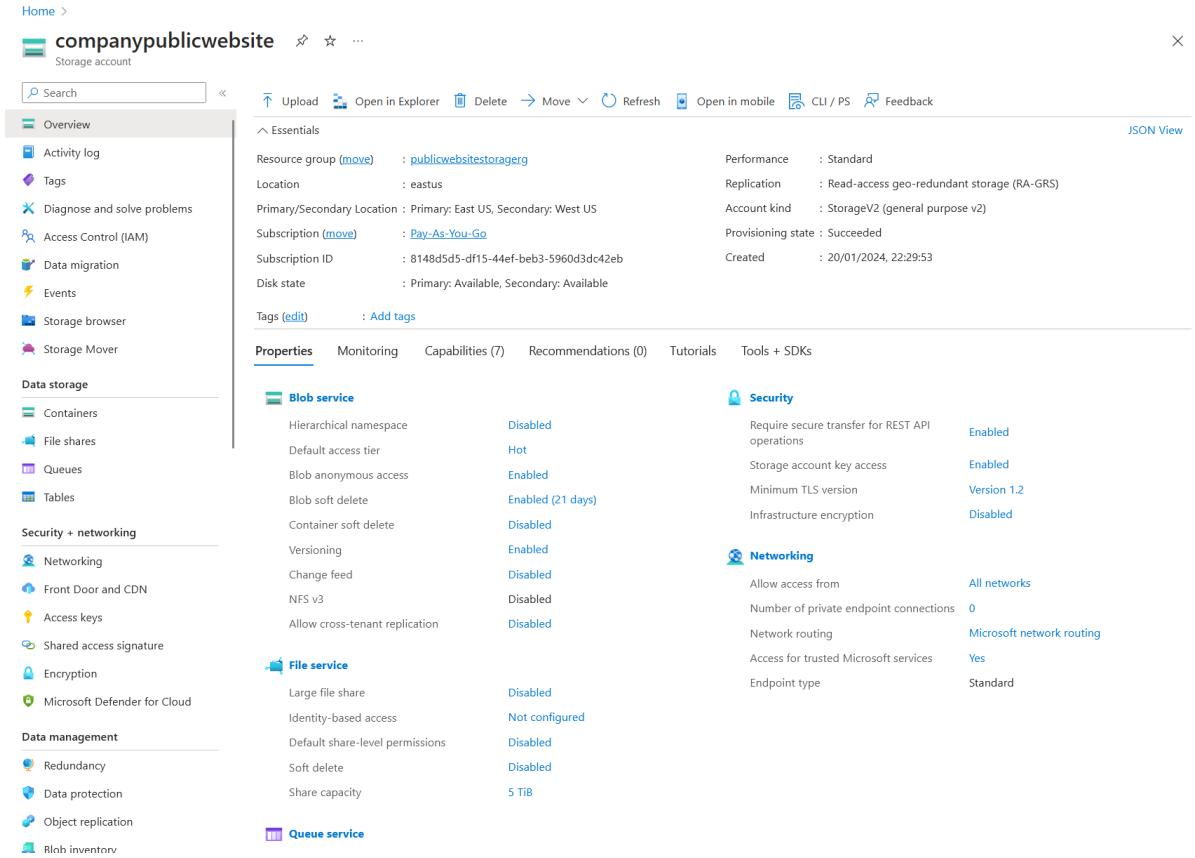
Architecture diagram



Task to be Completed

- Create a storage account for the company private documents.
- Configure redundancy for the storage account.
- Configure a shared access signature so partners have restricted access to a file.
- Back up the public website storage.
- Implement lifecycle management to move content to the cool tier.

*Step 1 - Create storage account – Same storage account used for **companypublicwebsite***



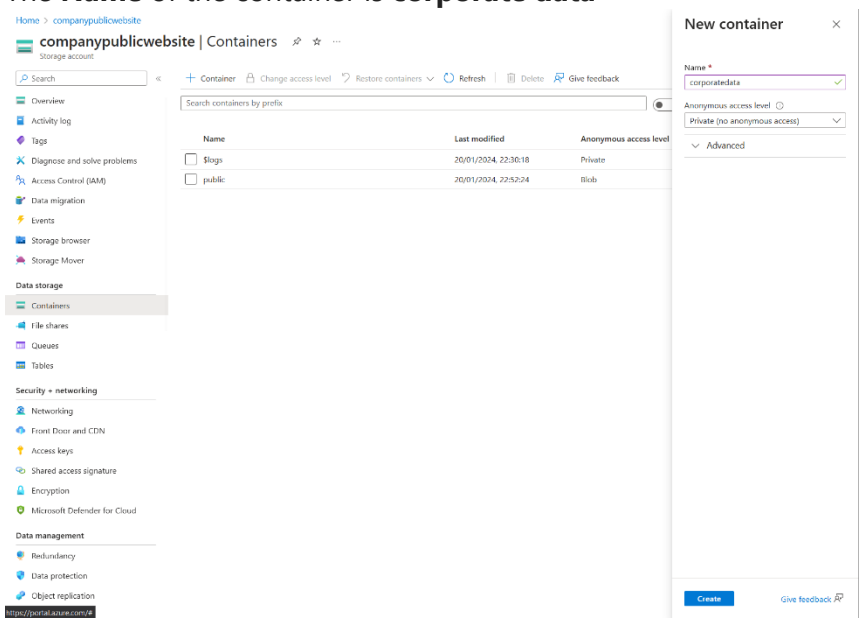
Step 2 Create a storage container, upload a file, and restrict access to the file

Create a private storage container for the corporate data

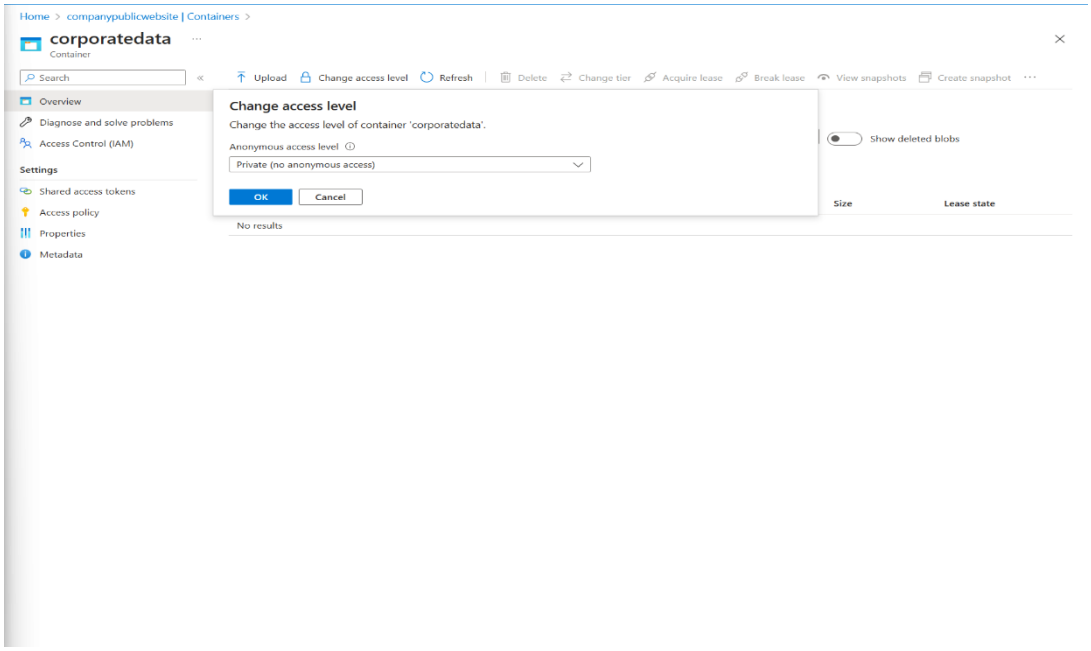
In the storage account, in the **Data storage** section, select the **Containers** blade

Select **+ Container**

The **Name** of the container is **corporate data**



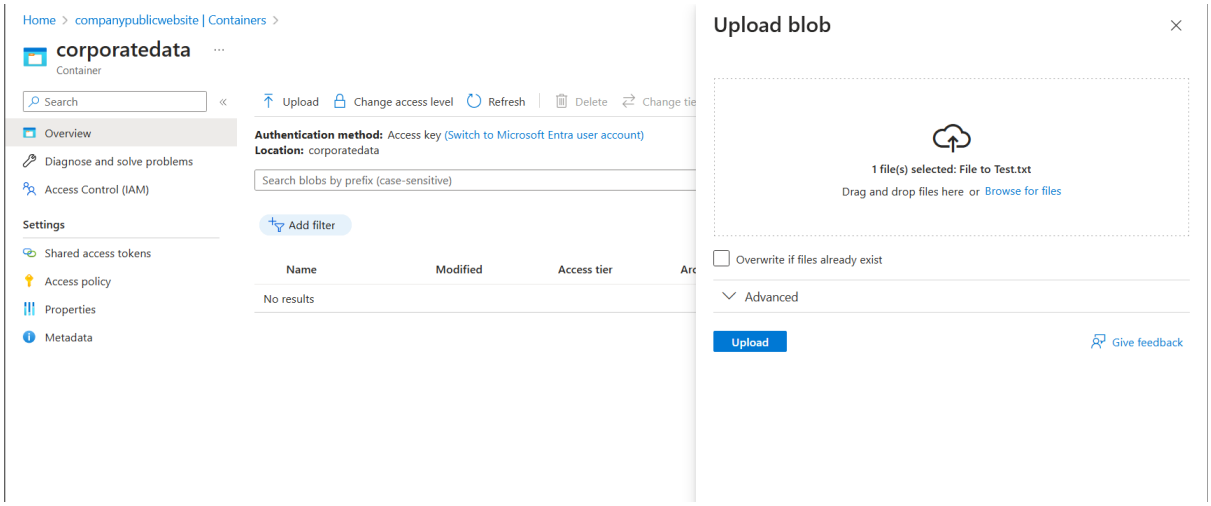
Ensure the **Public access level** is **Private (no anonymous access)**



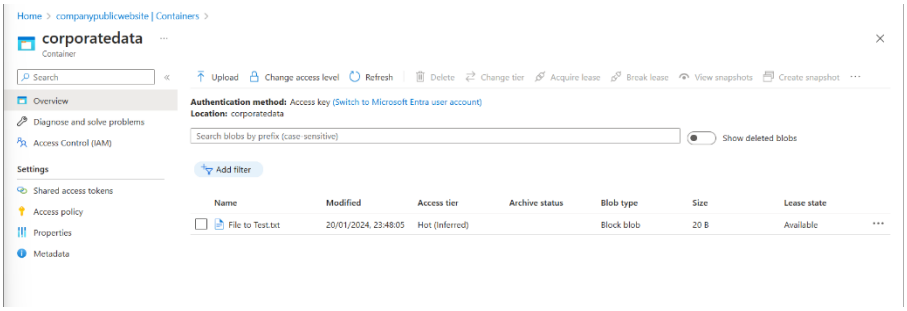
Step 3 - For testing, upload a file to the private container. The type of file doesn't matter. A small image or text file is a good choice. Test to ensure the file isn't publicly accessible

Select the container

Select **Upload**

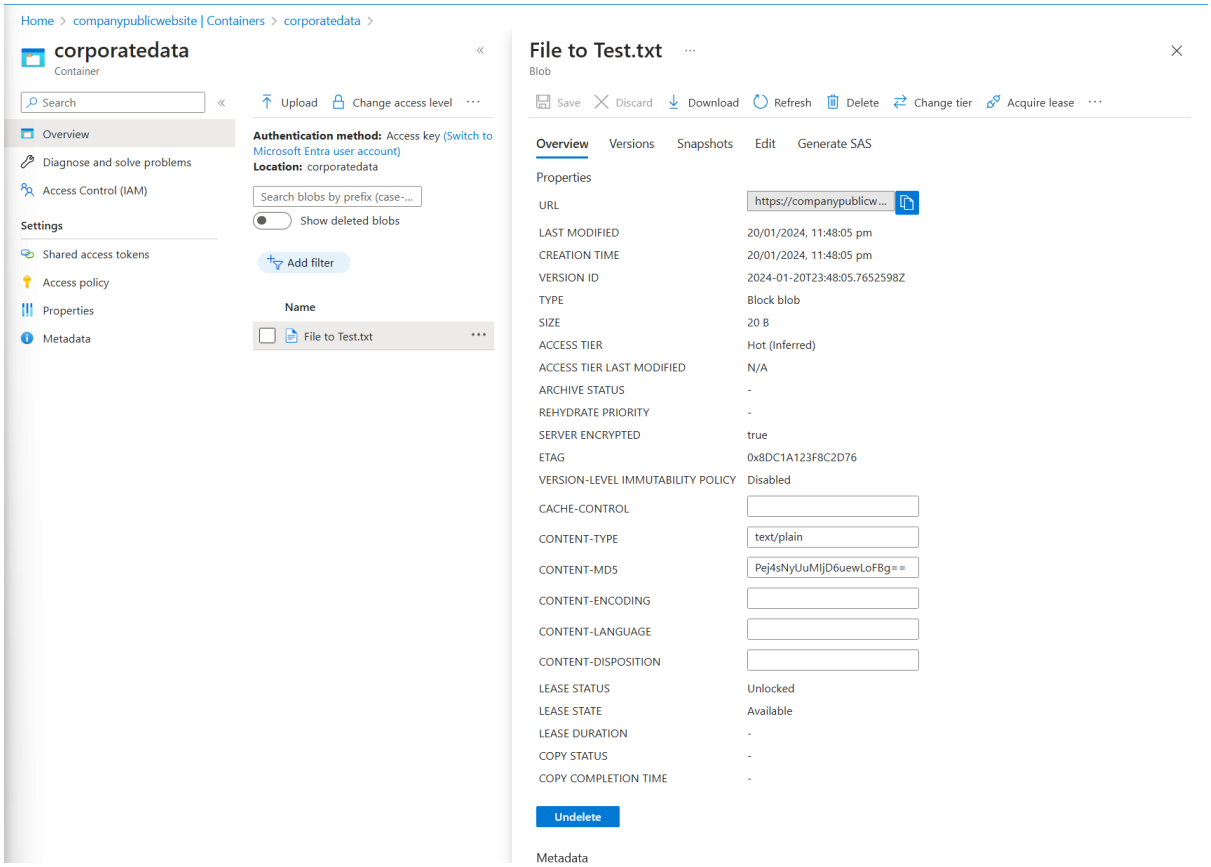


Upload the file

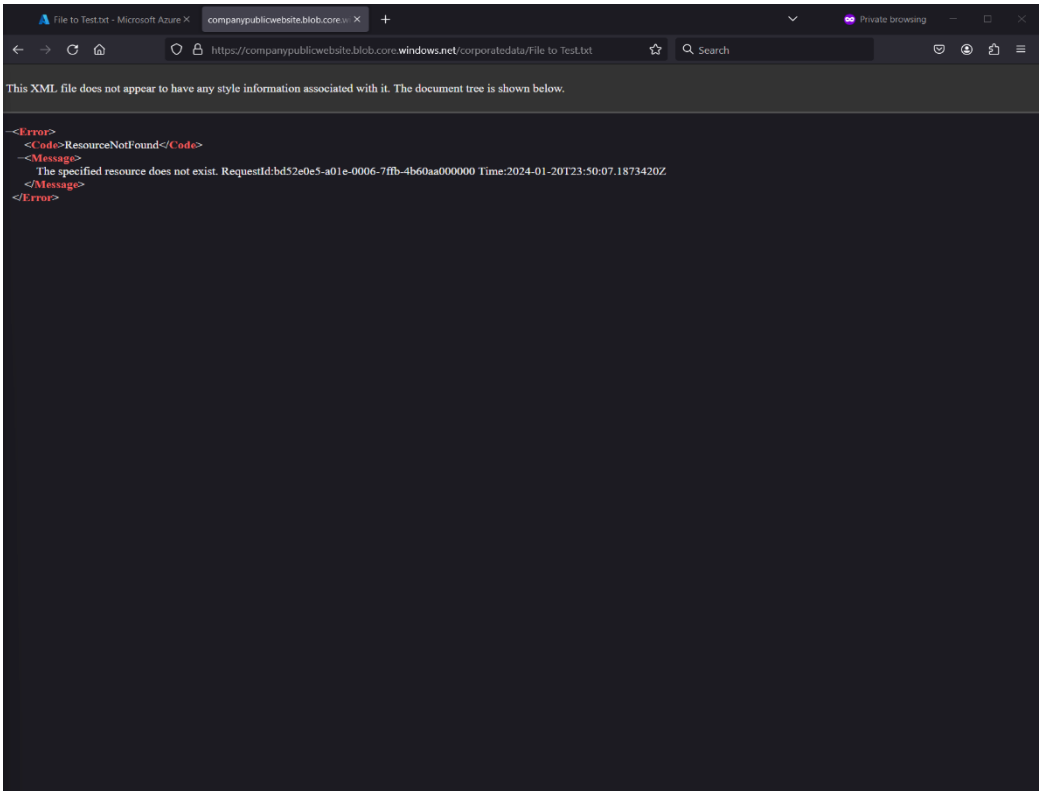


Select the uploaded file

On the **Overview** tab, copy the **URL**



Paste the **URL** into a new browser tab Verify the file doesn't display and you receive an error



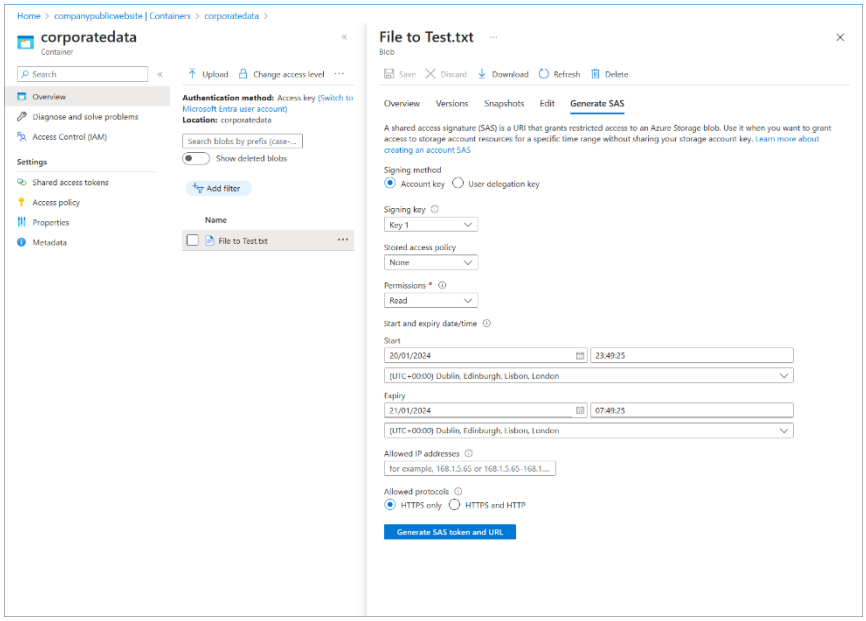
Step 4 - An external partner requires read and write access to the file for at least the next 24 hours. Configure and test a shared access signature (SAS)

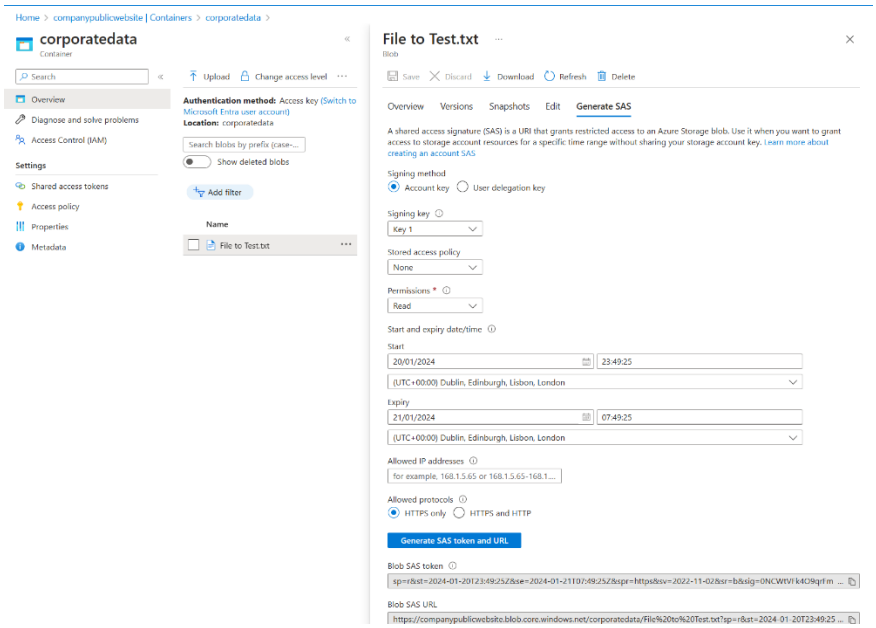
Select uploaded blob file and move to the **Generate SAS** tab

In the **Permissions** drop-down, ensure the partner has only **Read** permission

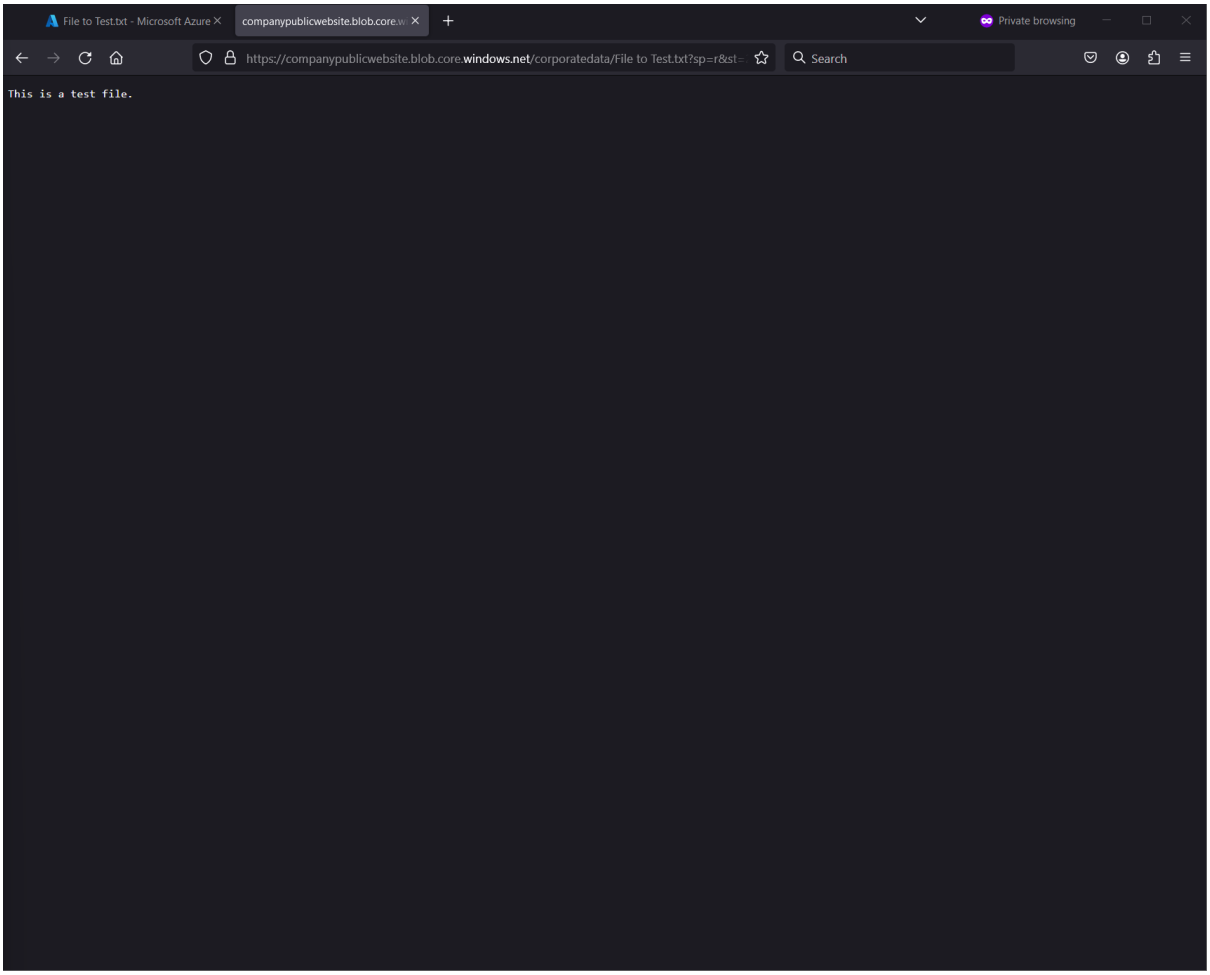
Verify the **Start and expiry date/time** is for the next 24 hours

Select **Generate SAS token and URL**





Copy the **Blob SAS URL** to a new browser tab



Verify you can access the file. If you have uploaded an image file it will display in the browser. Other file types will be downloaded.

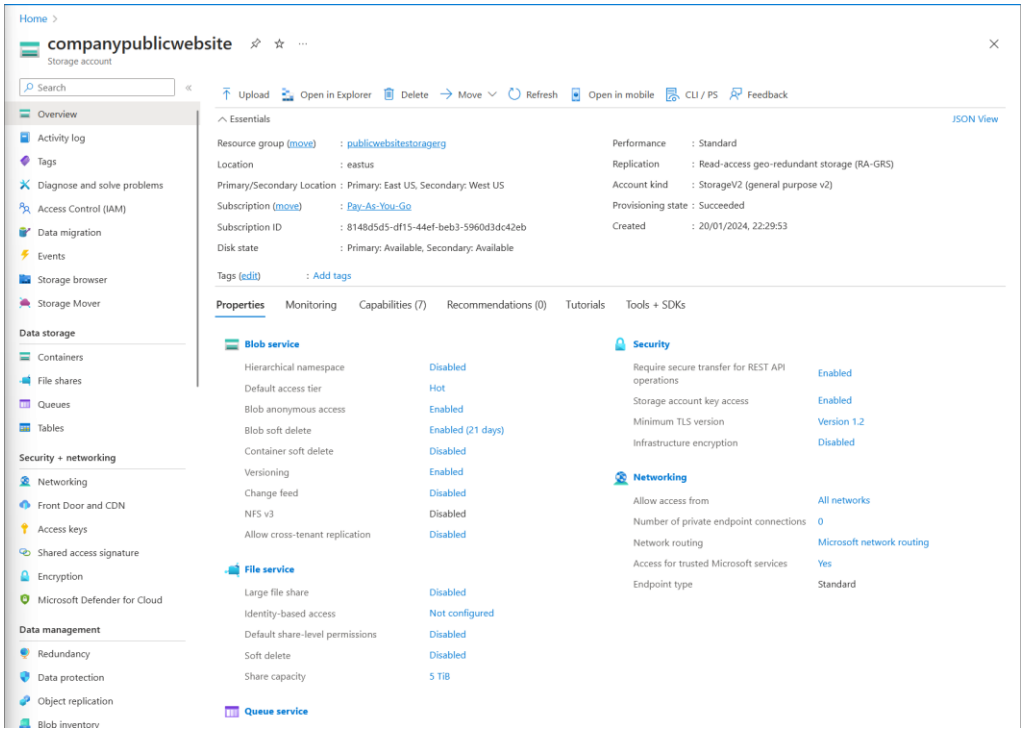
Step 5 - Configure storage access tiers and content replication

To save on costs, after 30 days, move blobs from the hot tier to the cool tier.

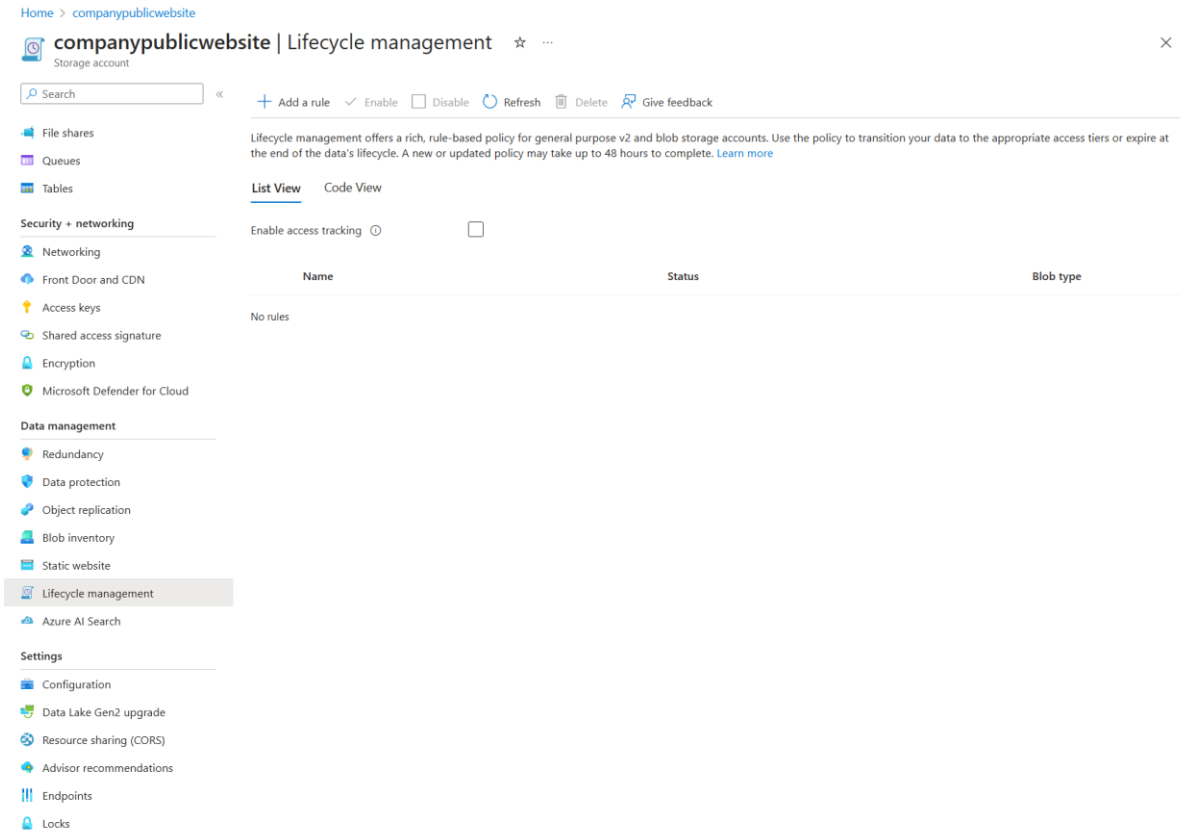
Return to the **storage account**

In the **Overview** section, notice the **Default access tier** is set to **Hot**

Next Image



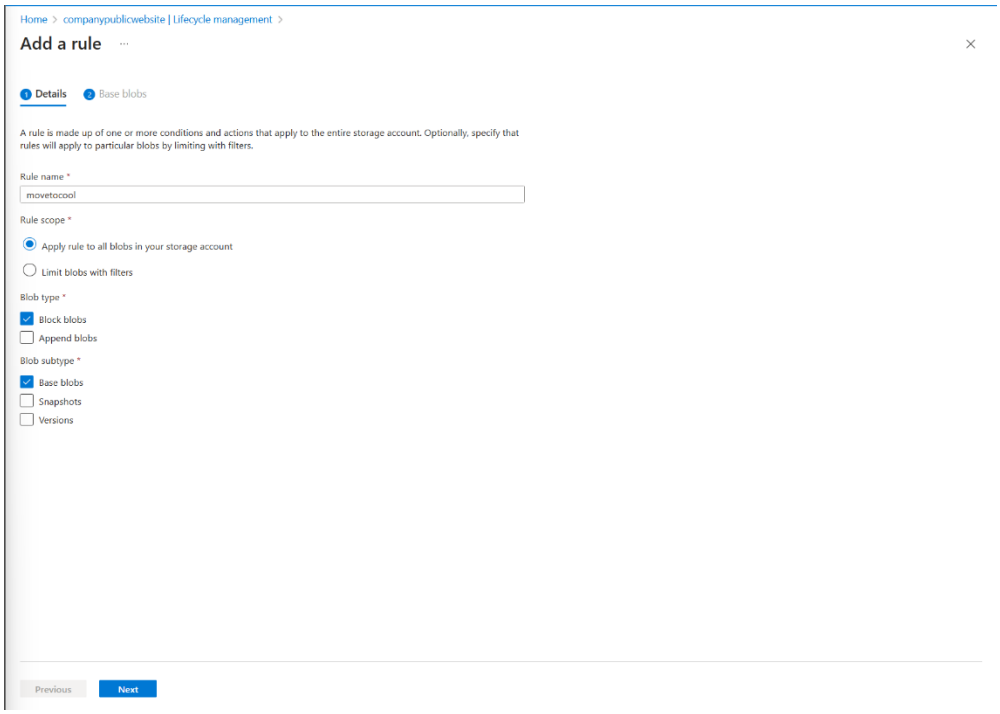
In the **Data management** section, select the **Lifecycle management blade**



Select **Add rule**

Set the **Rule name** to movetocool

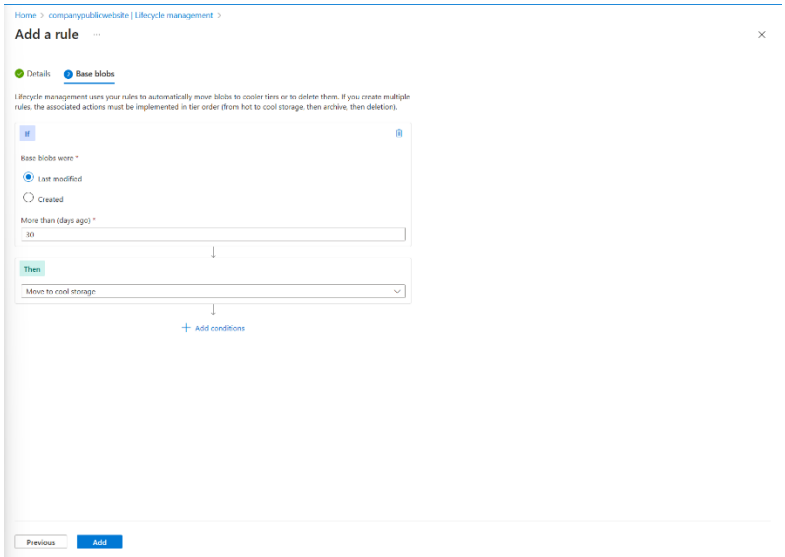
Set the **Rule scope** to **Apply rule to all blobs in the storage account**



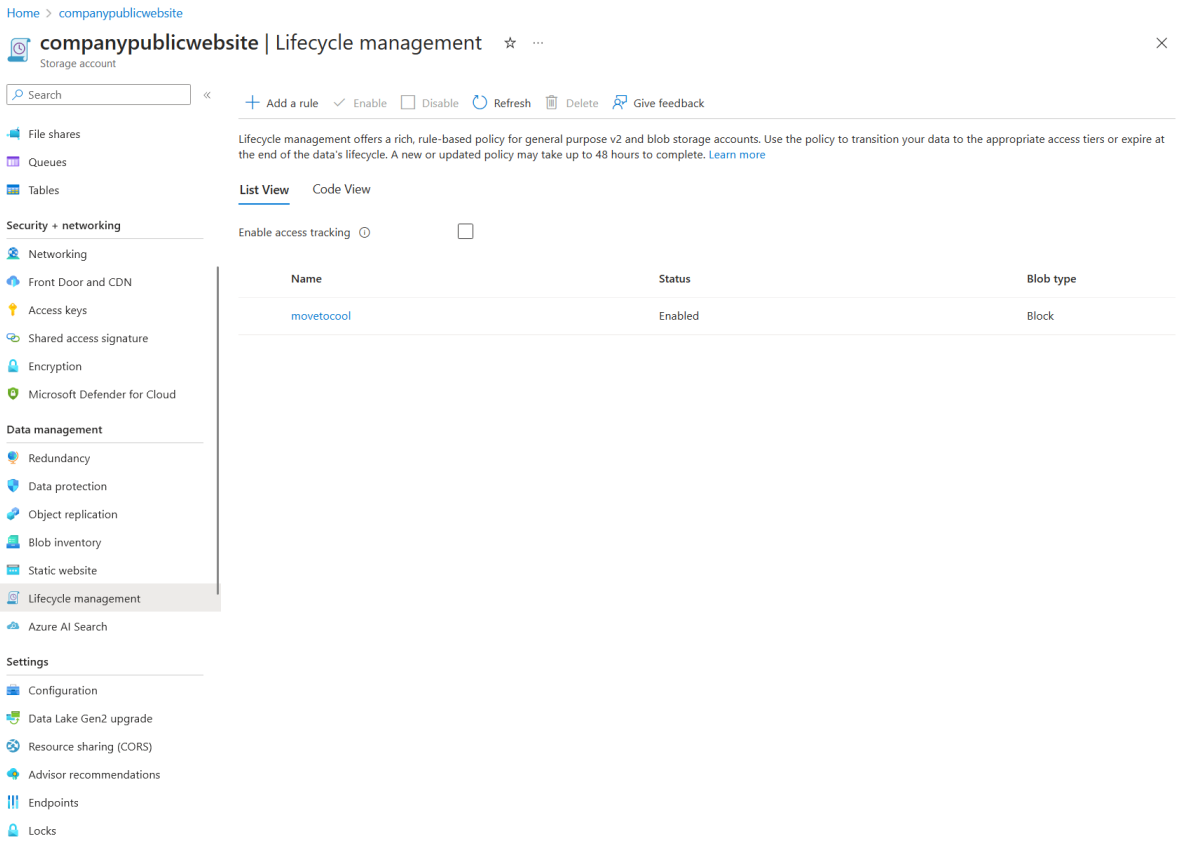
Select **Next**

Set **More than (days ago)** to **30**

In the **Then** drop-down select **Move to cool storage**

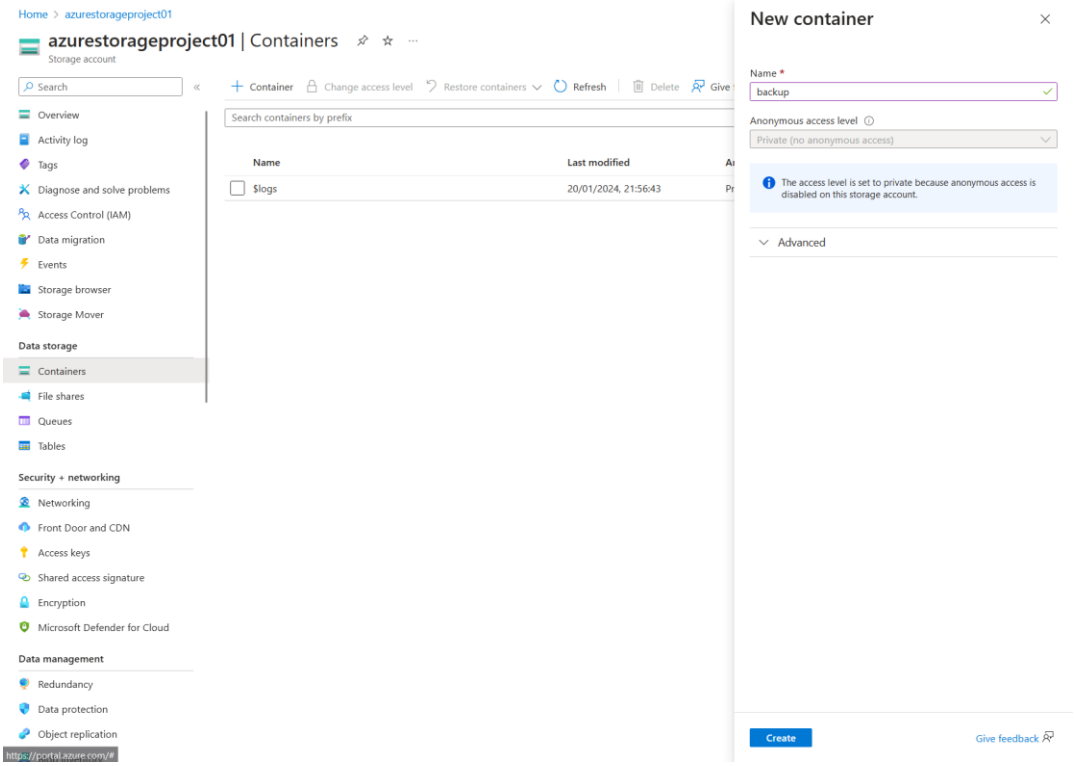


Add the rule.

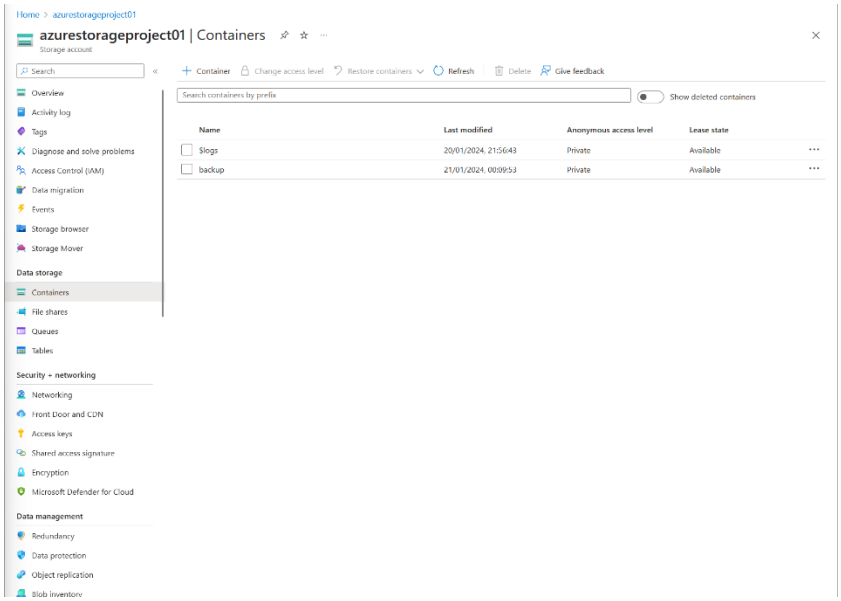


Step 5 - The public website files need to be backed up to another storage account – Configure object replication

Create a new container called backup in **azurestorageproject01** storage account

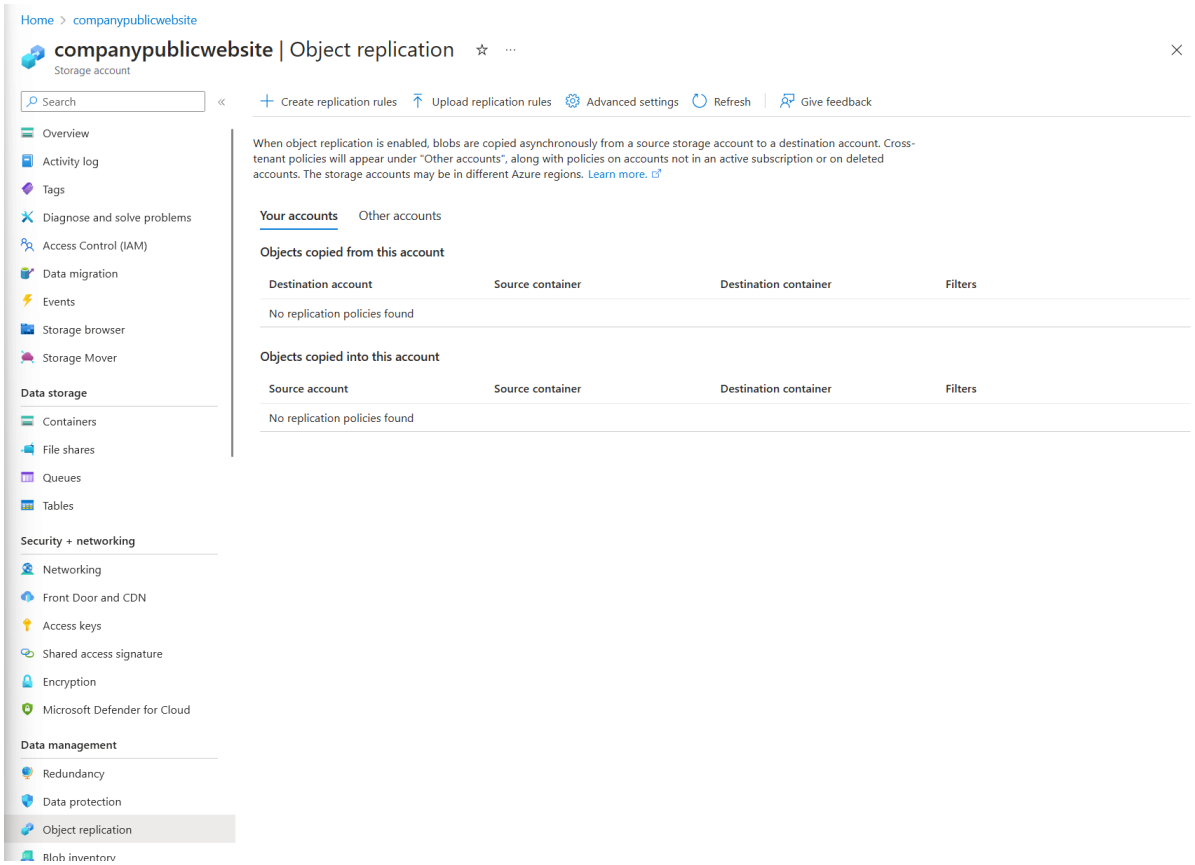


Confirm back up container created.



Step 5 – Configure object replication settings on companypublicwebsite storage account

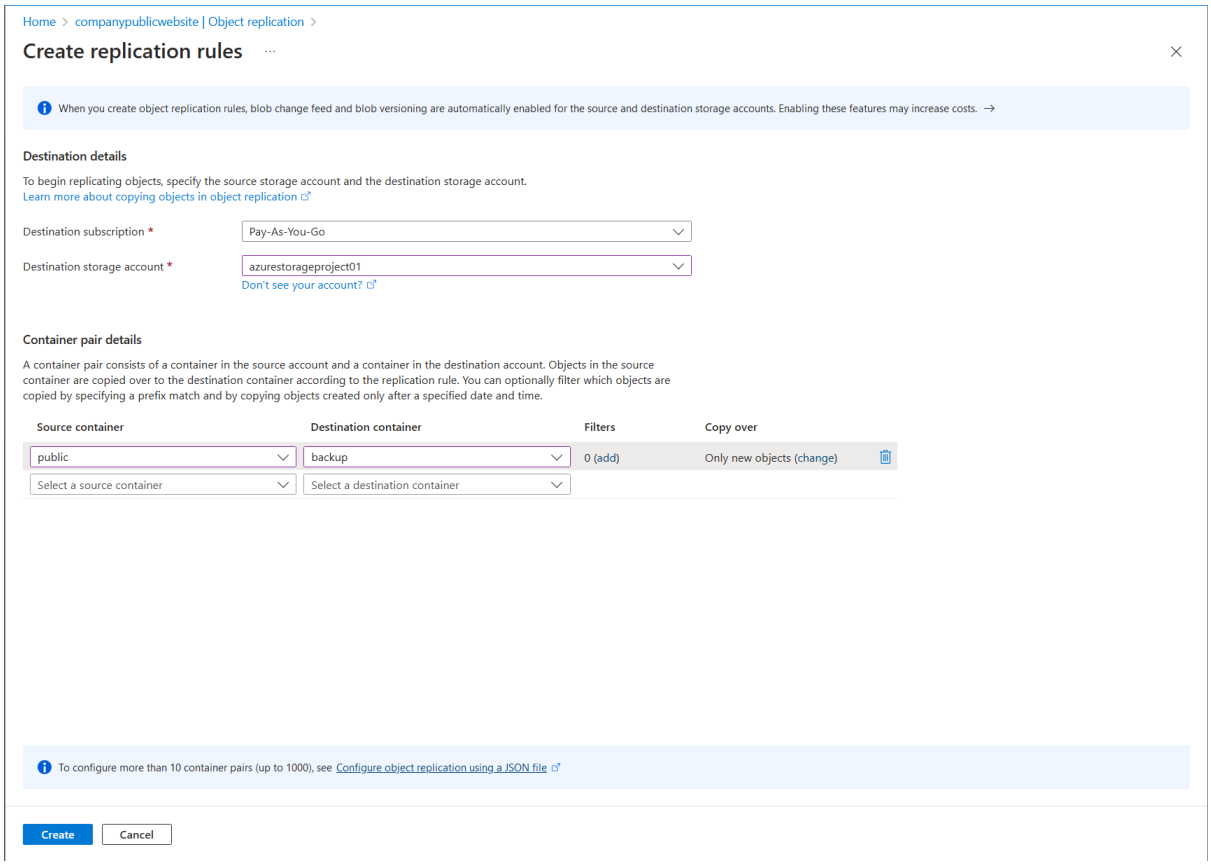
In the **Data management** section, select the **Object replication** blade.



Select **Create replication rules**

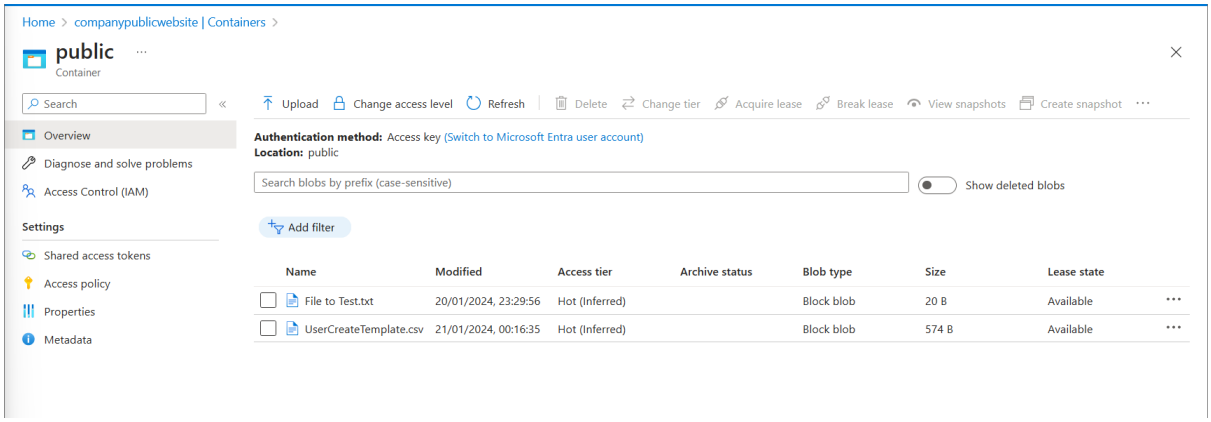
Set the **Destination storage account** to the **private** storage account

Set the **Source container** to **public** and the **Destination container** to **backup**

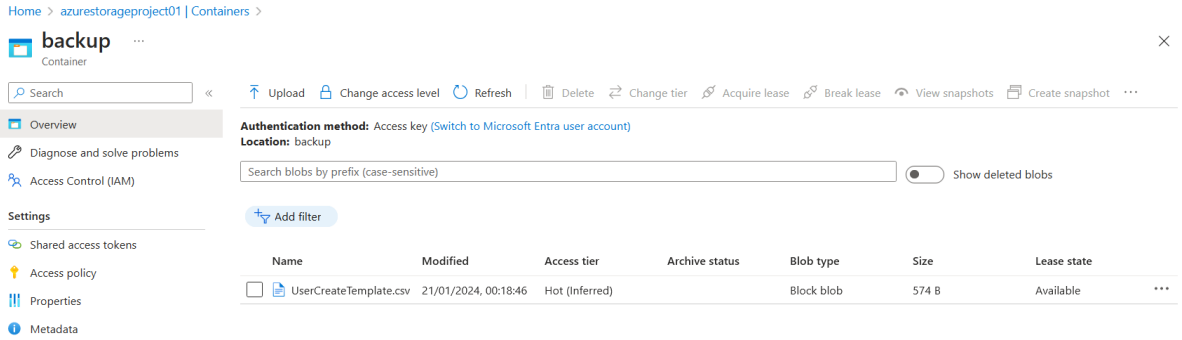


Create the replication rule

Uploaded file UserCreateTemplate.csv to public container on company public website as test



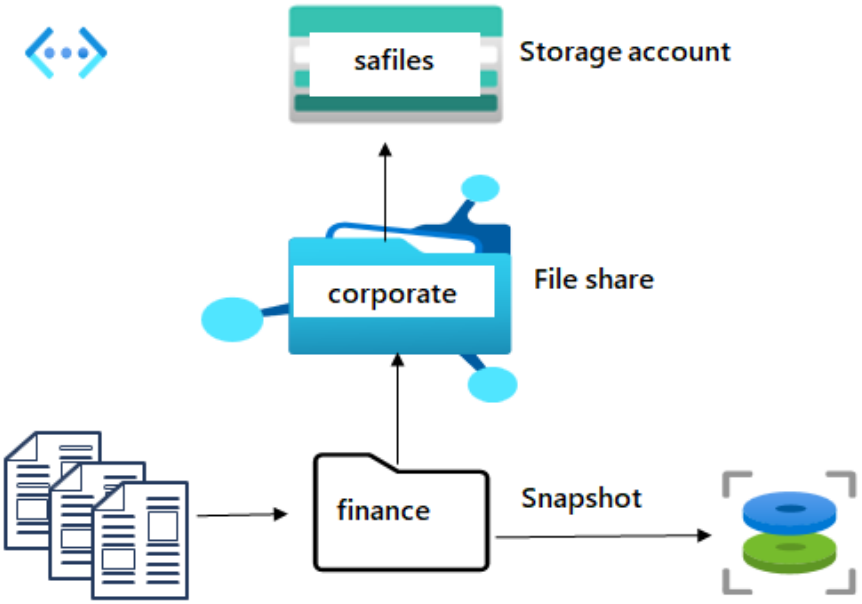
To check replication went back to destination back up storage account azurestorageproject01 and in the back up container file UserCreateTemplate.csv is listed in container showing it was replicated to storage account successfully.



Provide shared file storage for the company offices

The company is geographically dispersed with offices in different locations. These offices need a way to share files and disseminate information. For example, the Finance department needs to confirm cost information for auditing and compliance. This file shares should be easy to access and load without delay. Some content should only be accessed from selected corporate virtual networks.

Architecture diagram

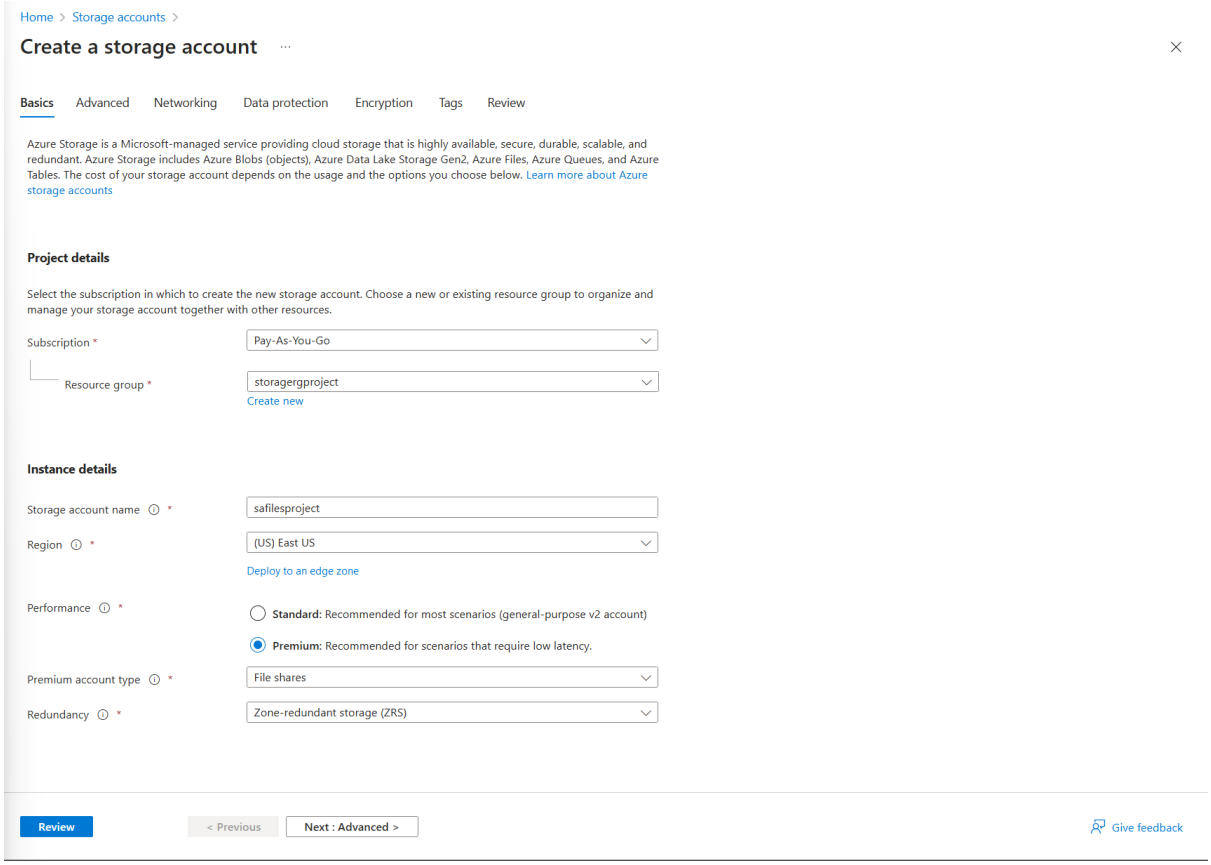


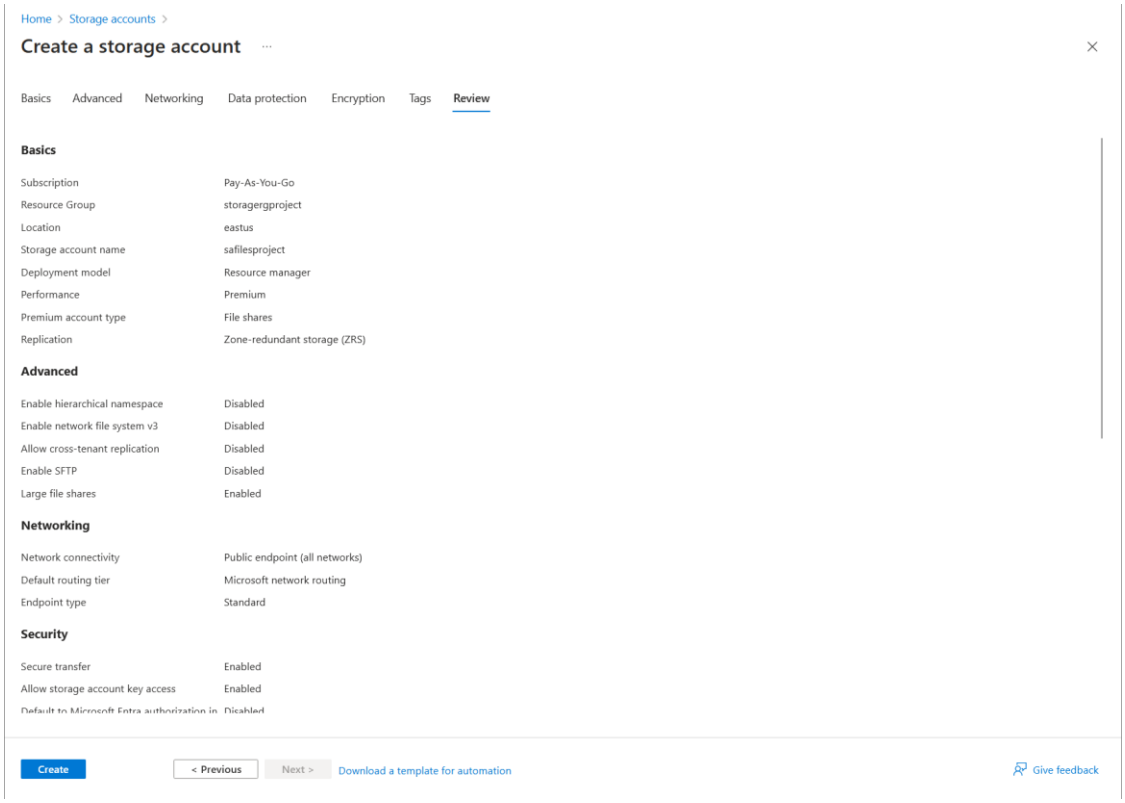
Task to be completed

- Create a storage account specifically for file shares.
- Configure a file share and directory.
- Configure snapshots and practice restoring files.
- Restrict access to a specific virtual network and subnet.

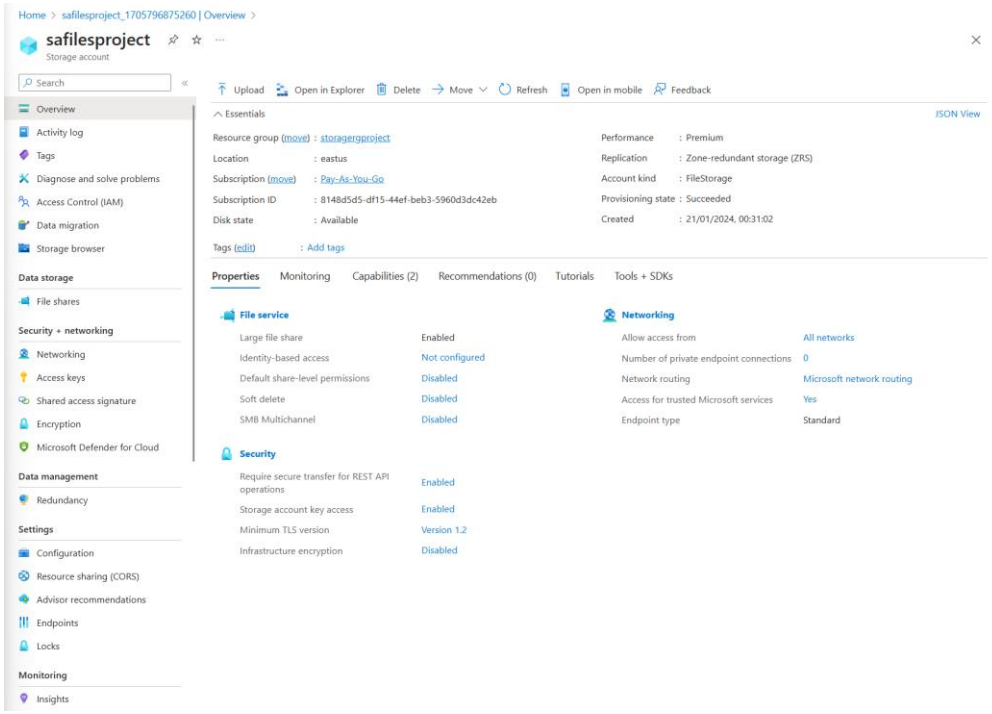
Step 1 - Create and configure a storage account for Azure Files.

Create a storage account for the finance department's shared files with the following options





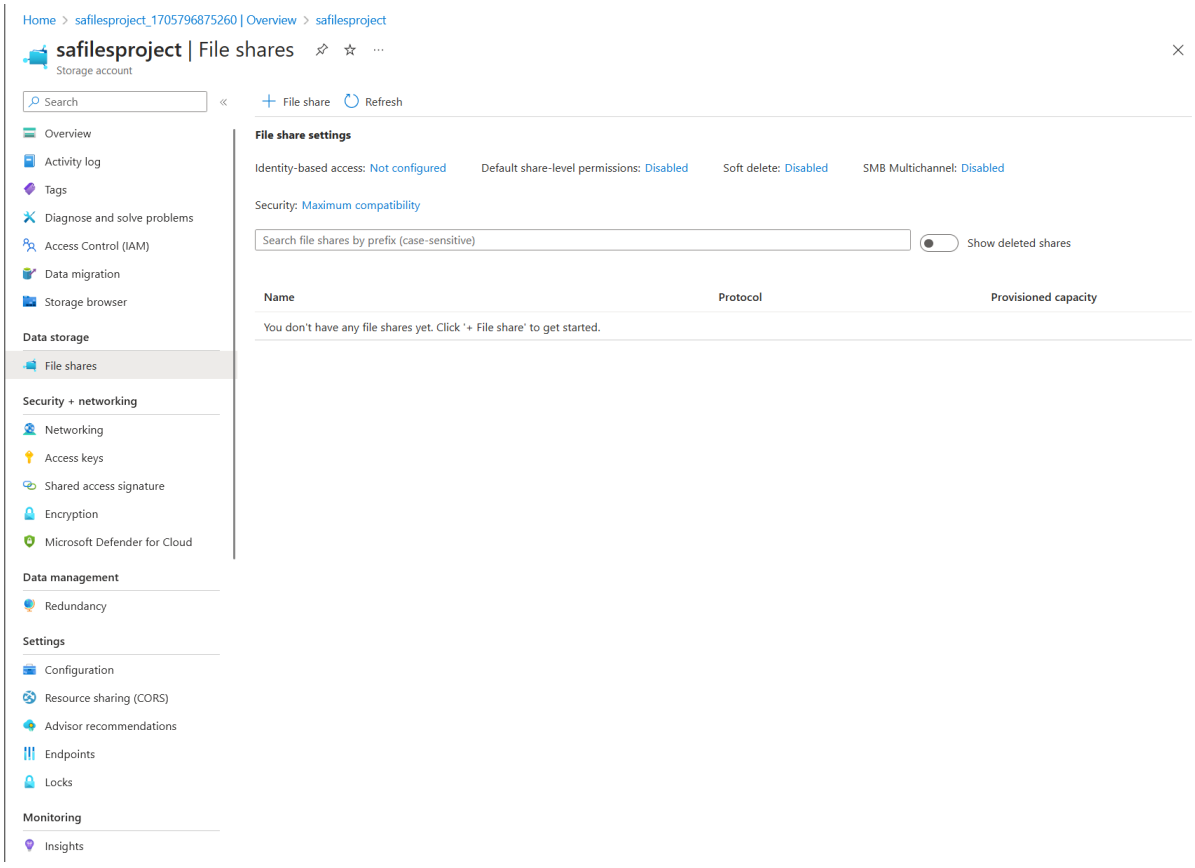
Review and create



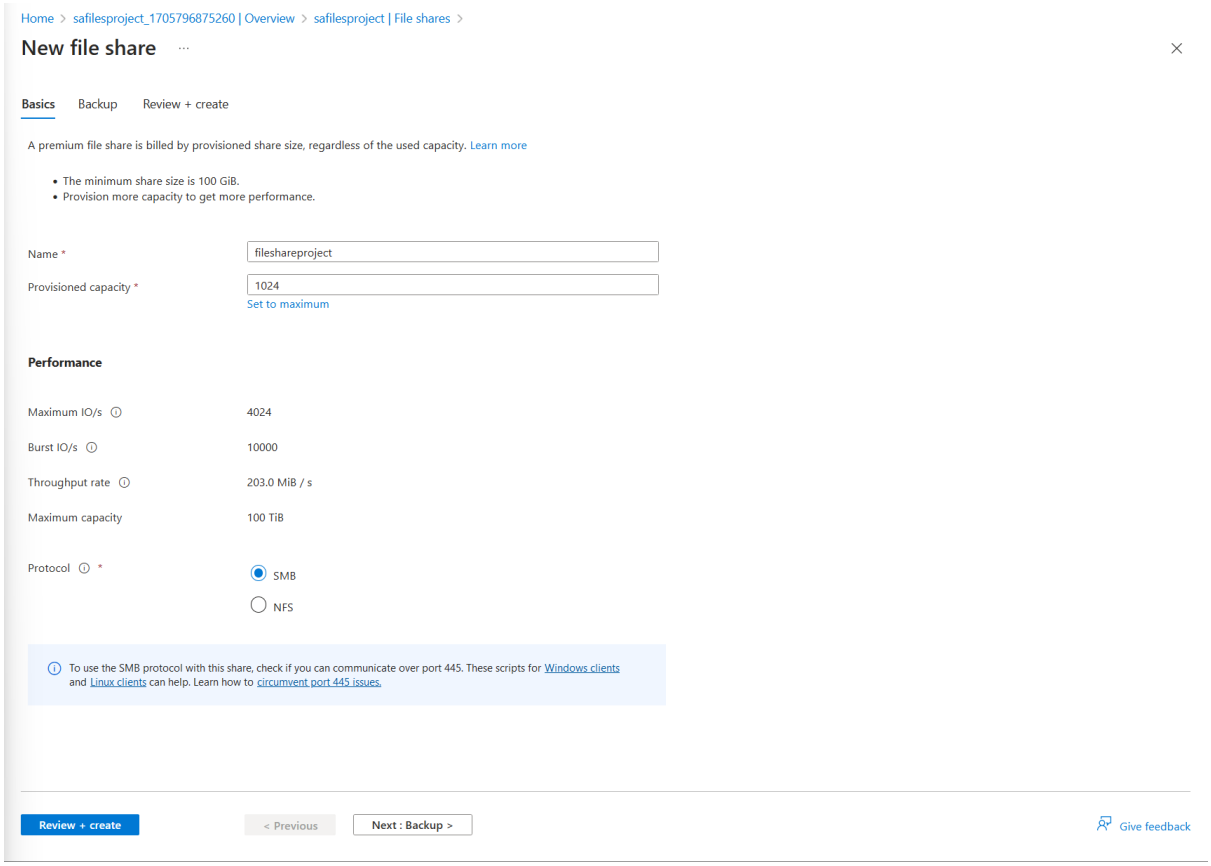
Step 2 - Create and configure a file share with directory

Create a file share for the corporate office

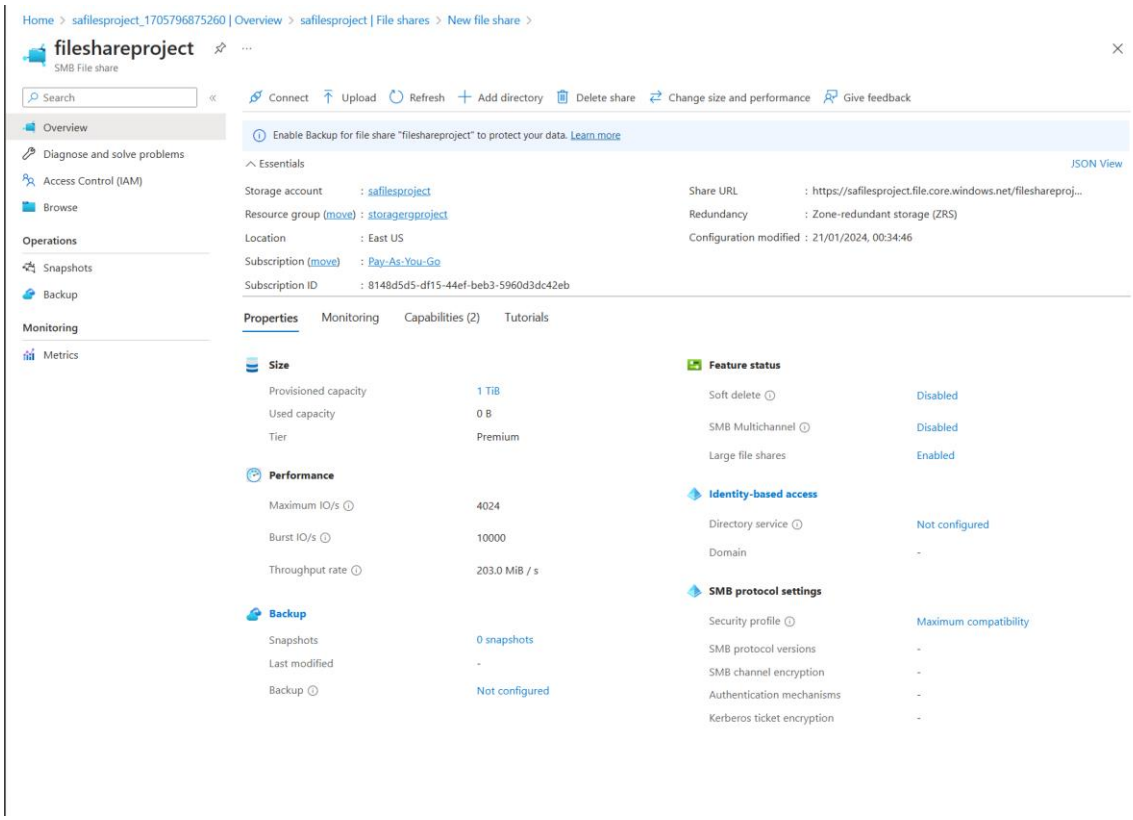
In the storage account, in the **Data storage** section, select the **File shares** blad



Select **+ File share** and provide a **Name**



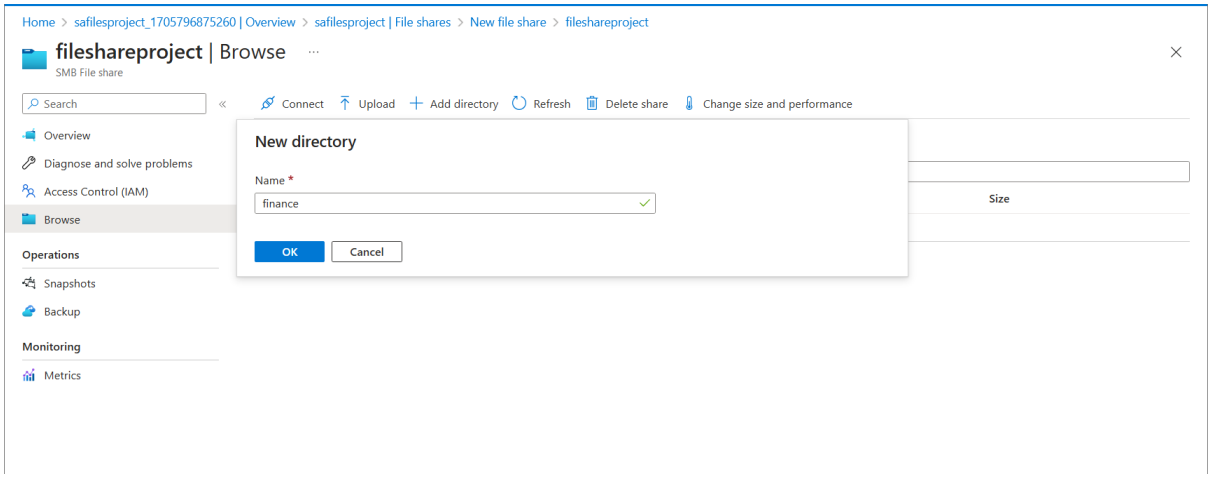
Review and create

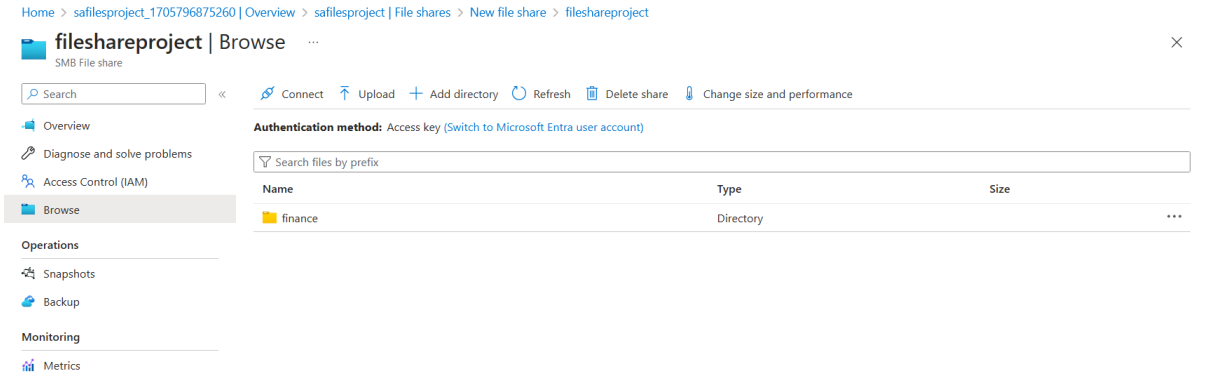


Step 3 - Add a directory to the file share for the finance department. For future testing, upload a file

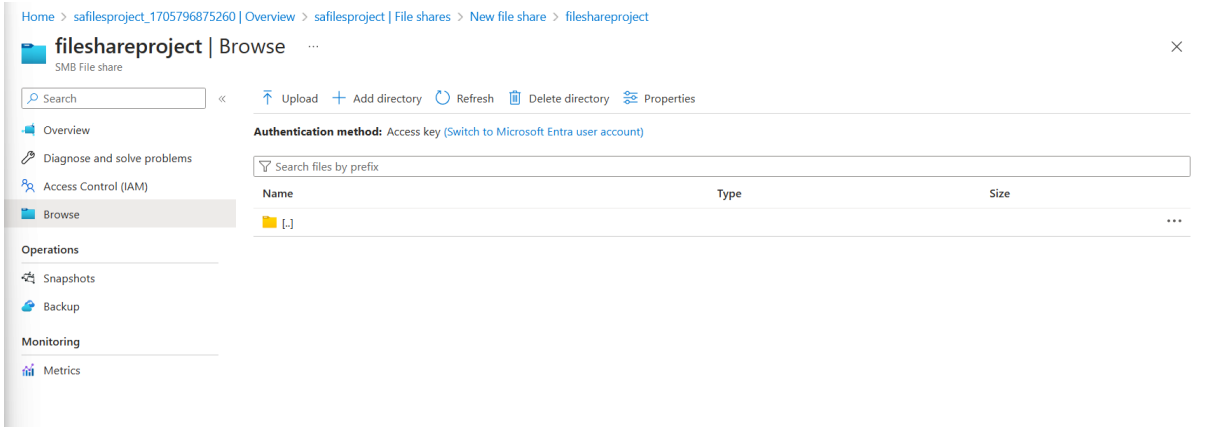
Select your file share and select **+ Add directory**

Name the new directory finance



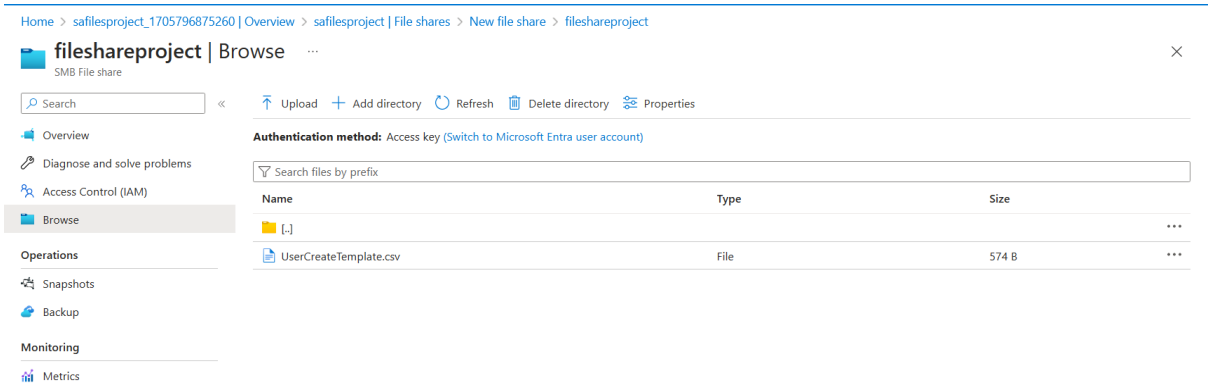


Select **Browse** and then select the **finance** directory



Notice you can **Add directory** to further organize your file share

Upload a file of your choosing



Step 4 - Configure and test snapshots

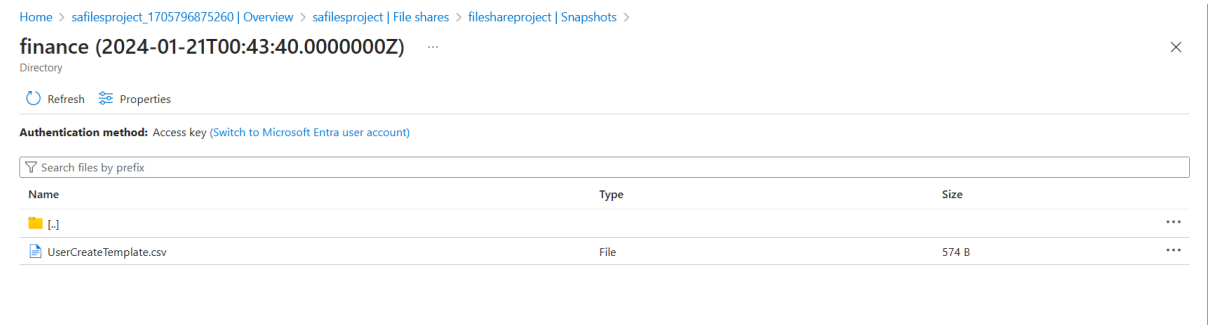
Similar to blob storage, you need to protect against accidental deletion of files

Select file share

In the **Operations** section, select the **Snapshots** blade

Select **+ Add snapshot**. The comment is optional. Select **OK**

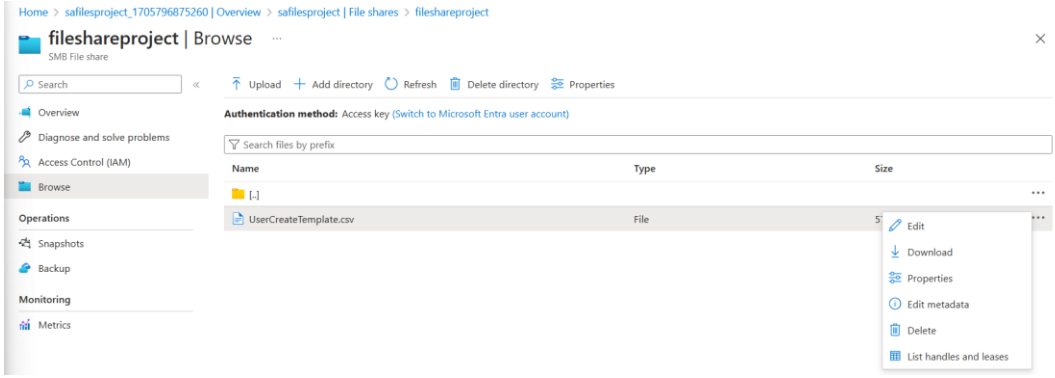
Select your snapshot and verify your file directory and uploaded file are included



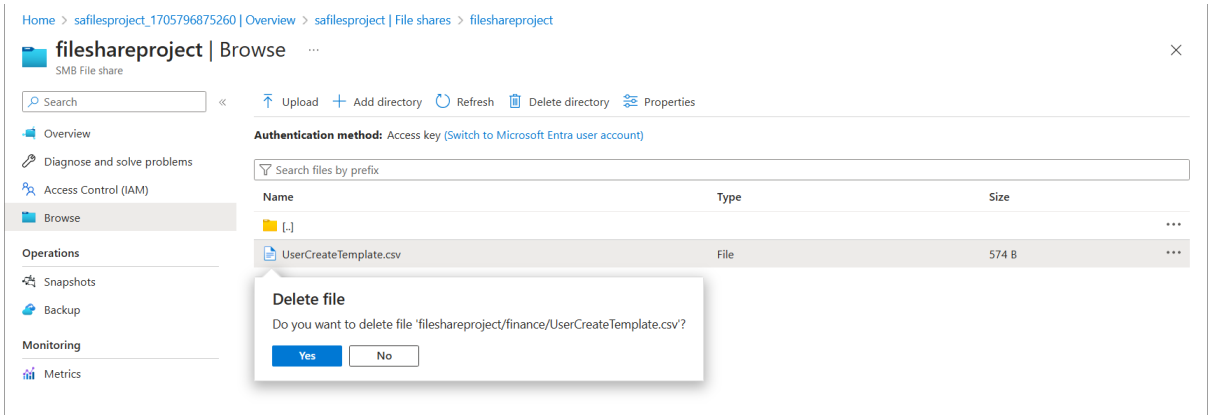
Step 5 - using snapshots to restore a file

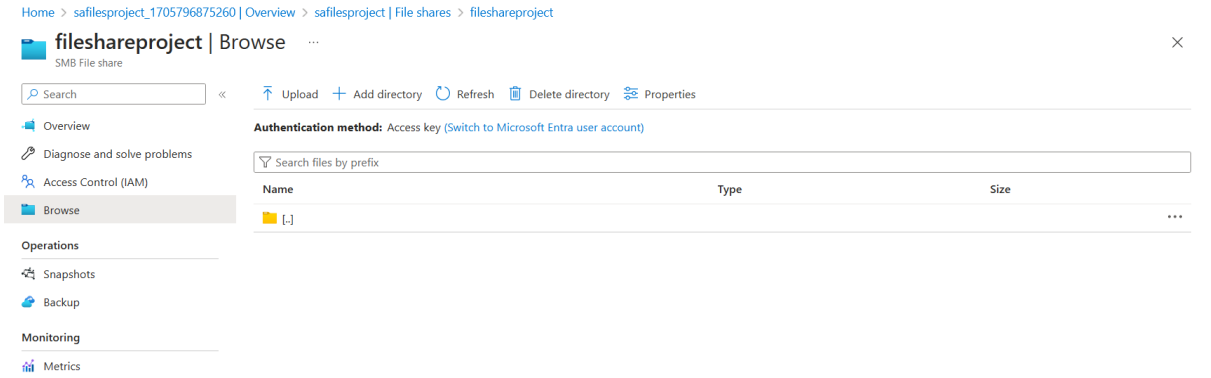
Return to your **file share**

Browse to your file directory

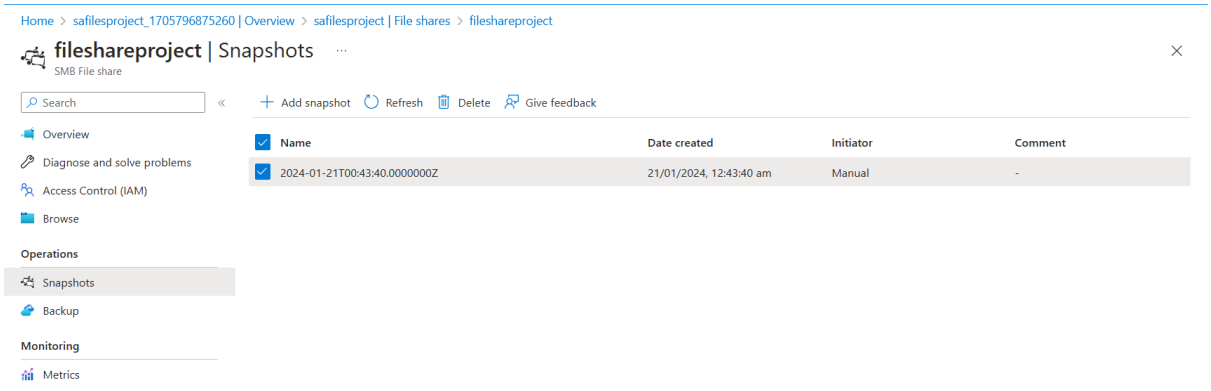


Locate your uploaded file and in the **Properties** pane select **Delete**. Select **Yes** to confirm the deletion

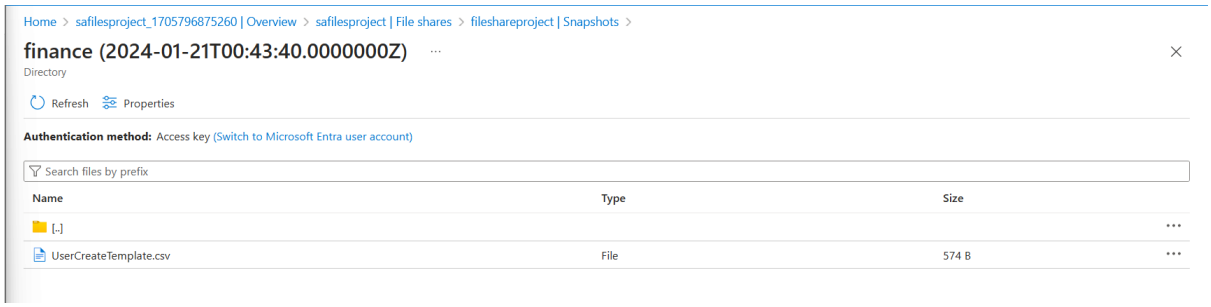




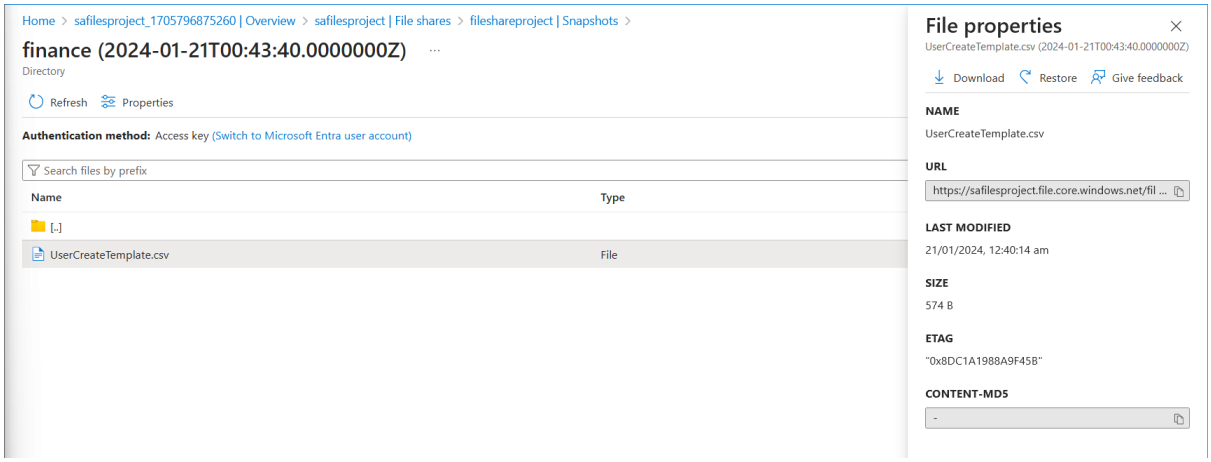
Select the **Snapshots** blade and then select your snapshot



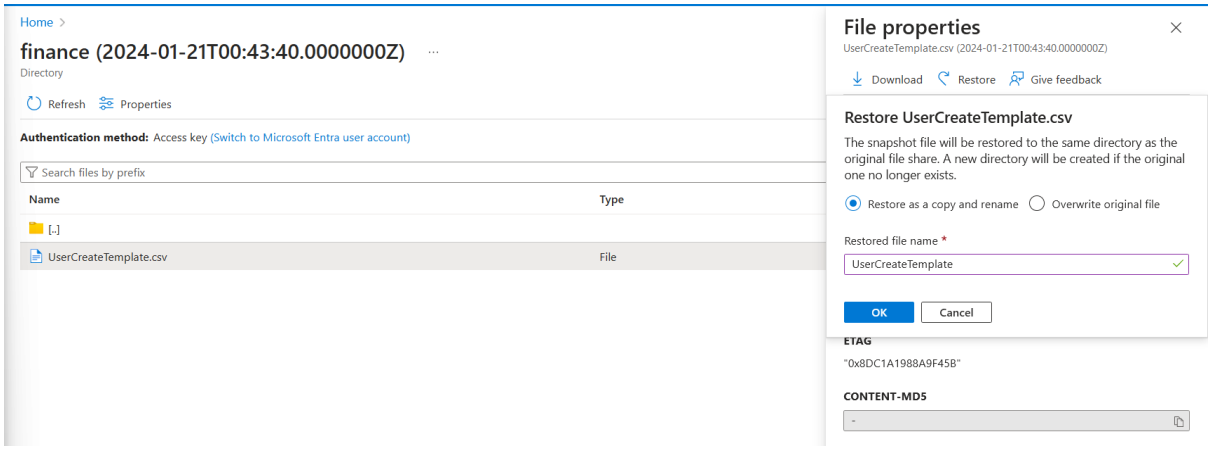
Navigate to the file you want to restore



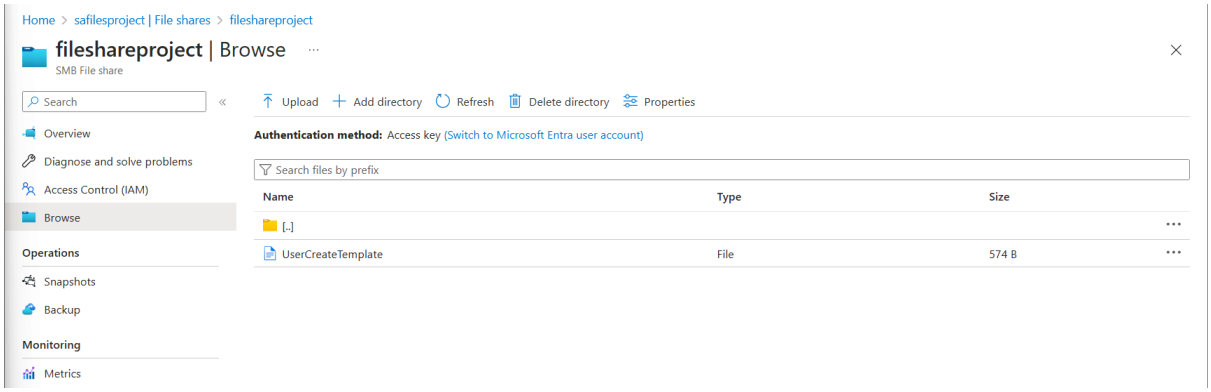
Select the file and the select **Restore**



Provide a **Restored file name**



Verify file directory has the restored file

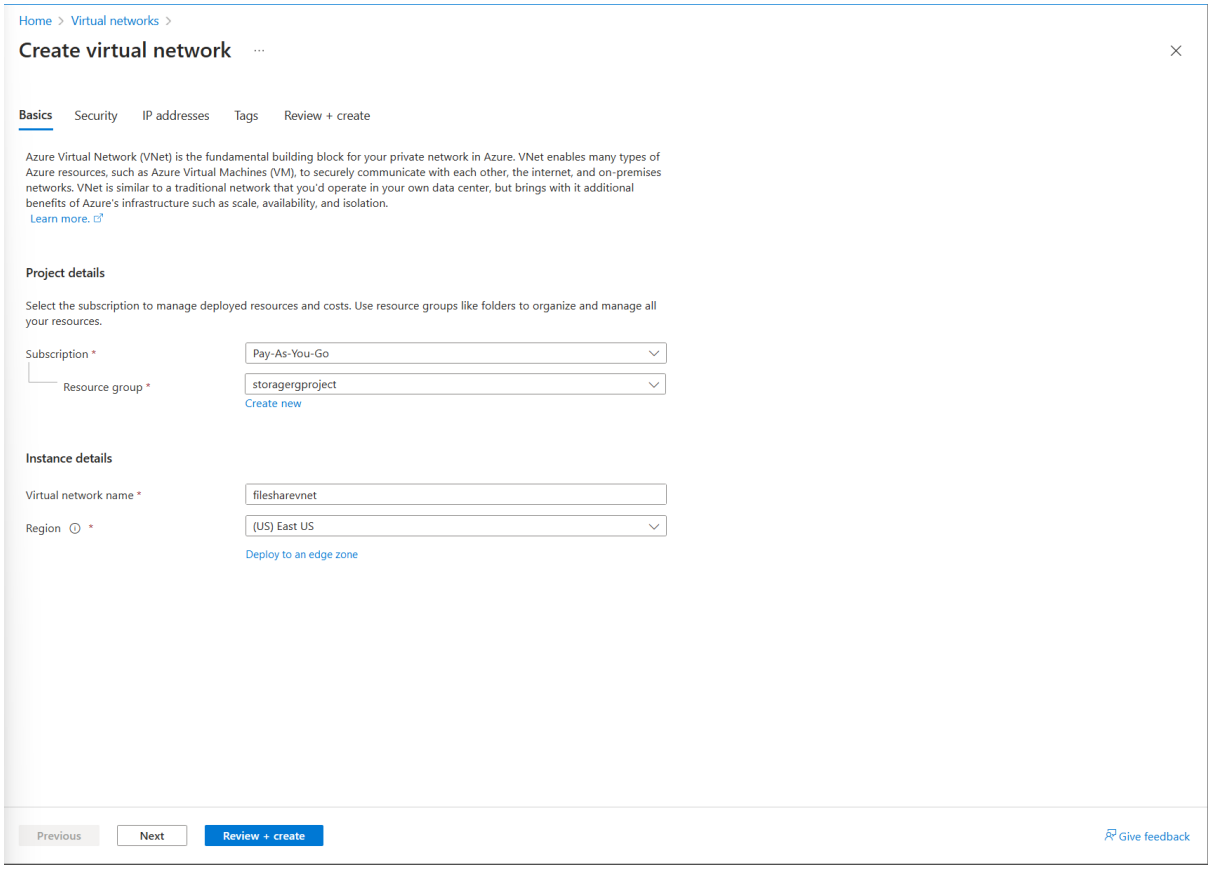


Step 6 - Configure restricting storage access to selected virtual networks

This section requires a virtual network with subnet. In a production environment these resources would already be created.

Search for and select **Virtual networks**

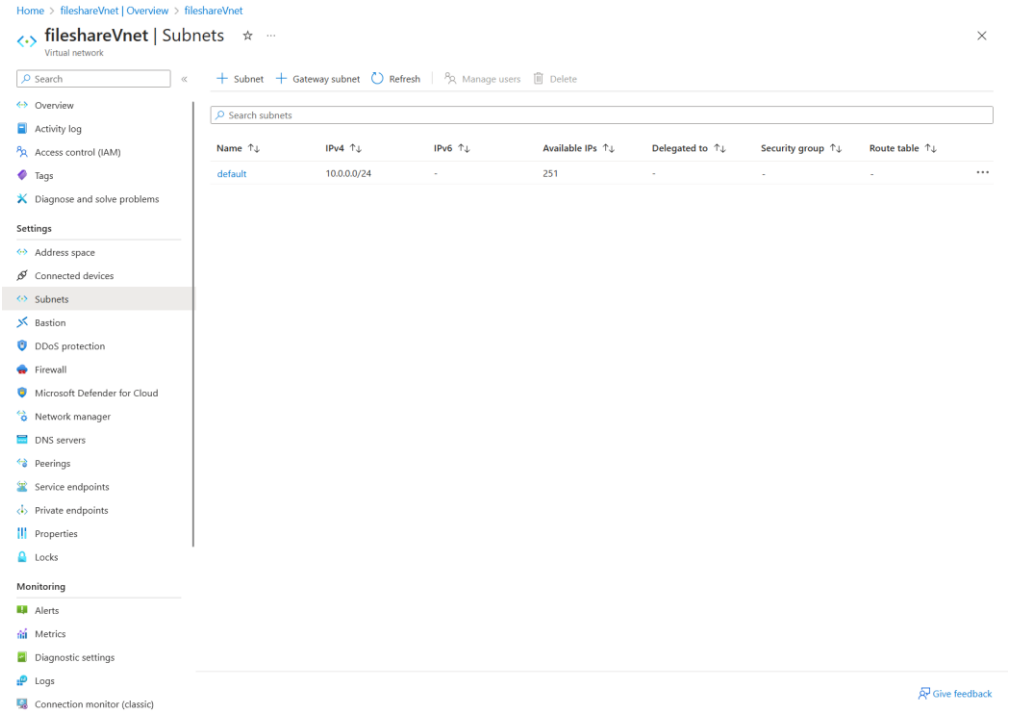
Select **Create**. Select your resource group. and give the virtual network a **name**



Review and create

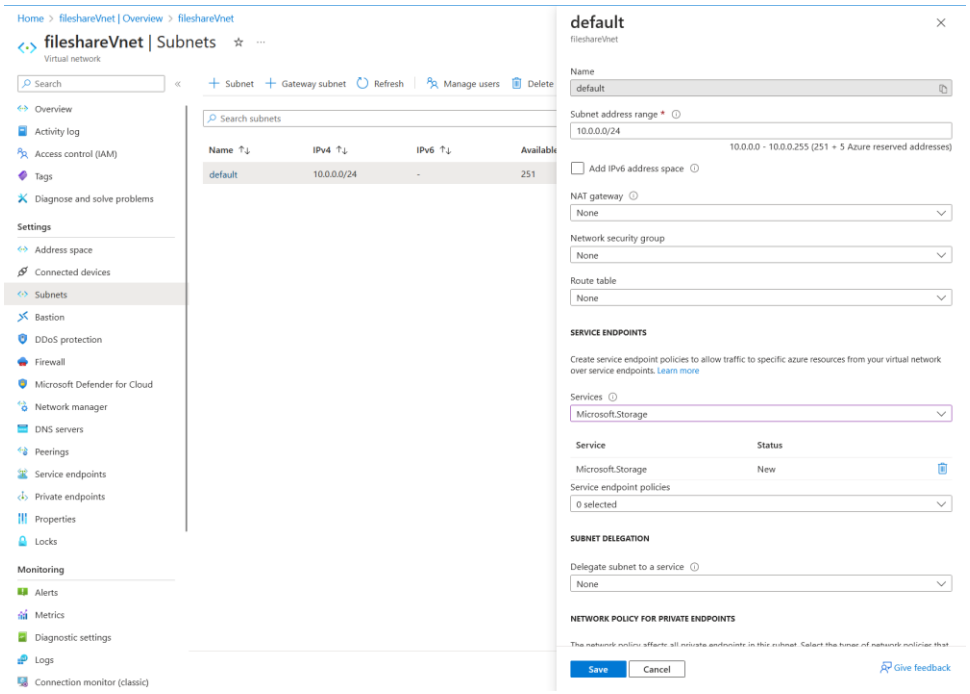
Next section configure Vnet settings for file share.

In the **Settings** section, select the **Subnets** blade



Select the **default** subnet

In the **Service endpoints** section choose **Microsoft.Storage** in the **Services** drop-down

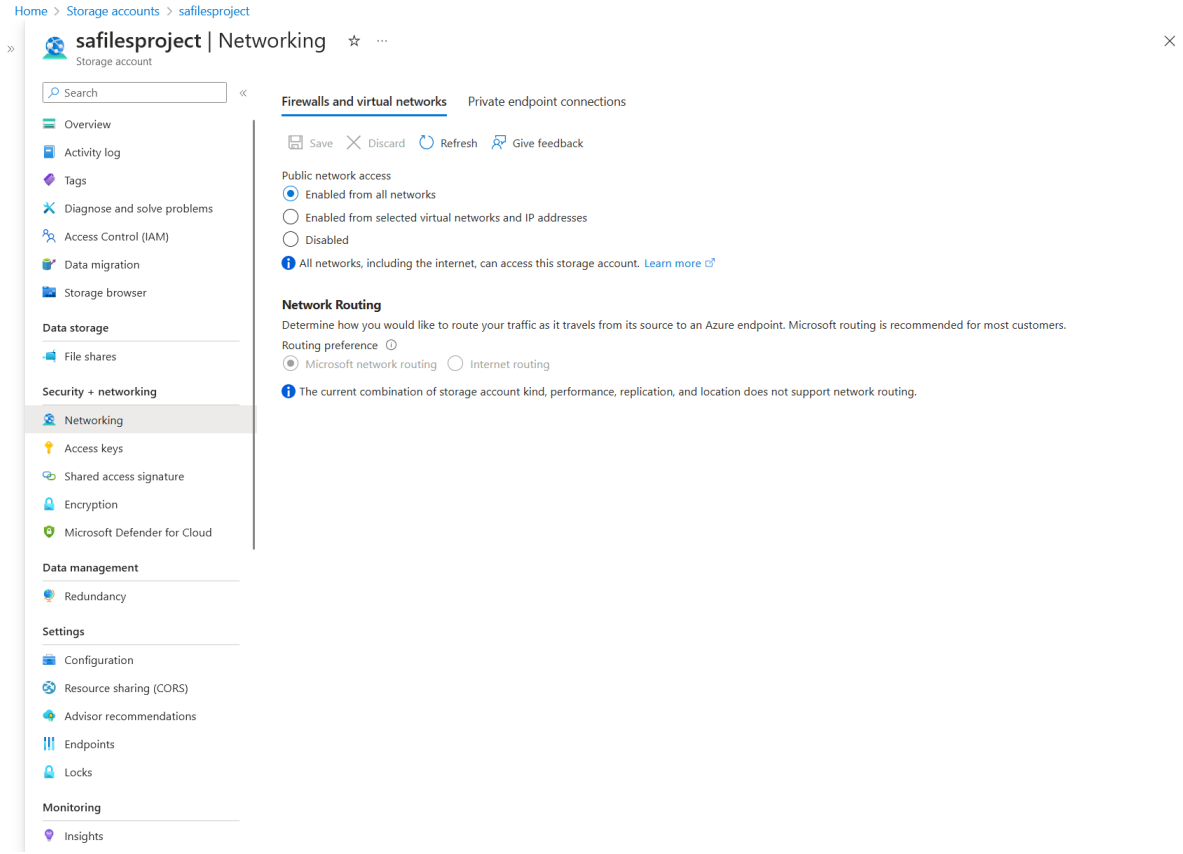


Save to save settings to redirect storage traffic to Microsoft backbone network to storage account

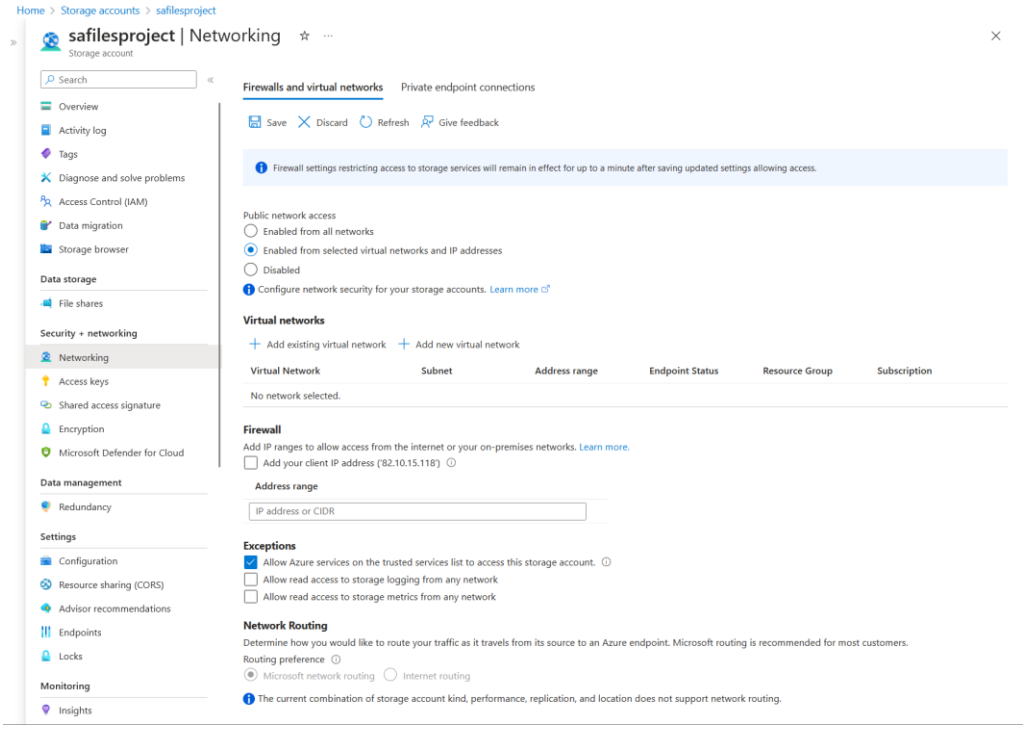
Step 7 - The storage account should only be accessed from the virtual network just created

Back to storage account **safilesproject**

In the **Security + networking** section, select the **Networking** blade

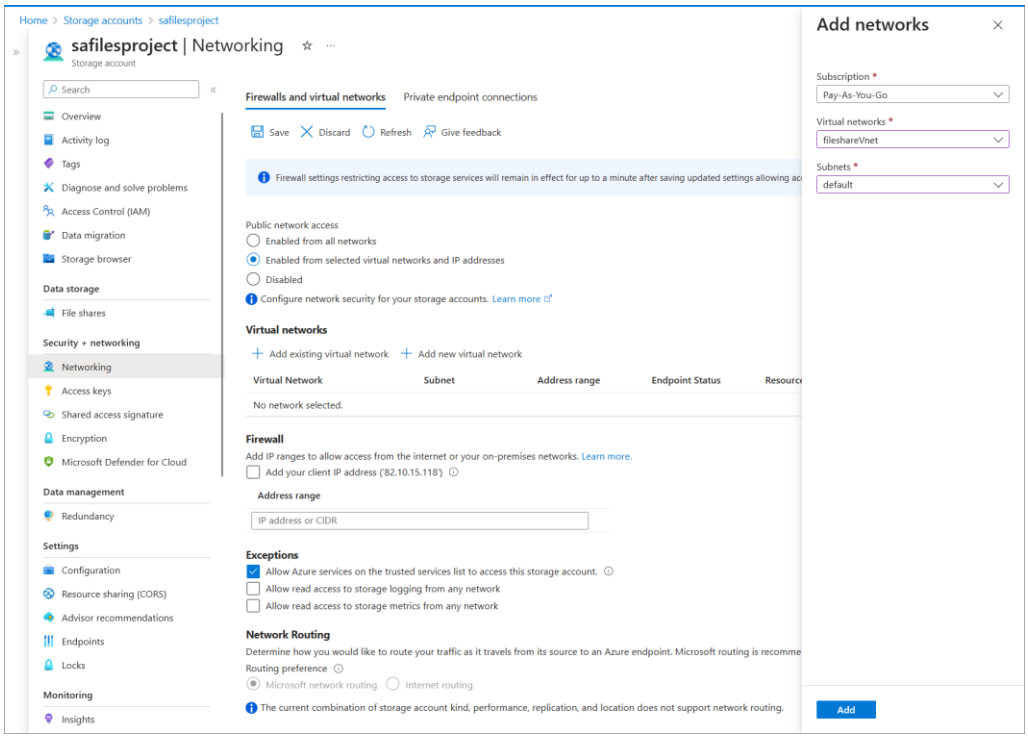


Change the **Public network access** to **Enabled from selected virtual networks and IP addresses**

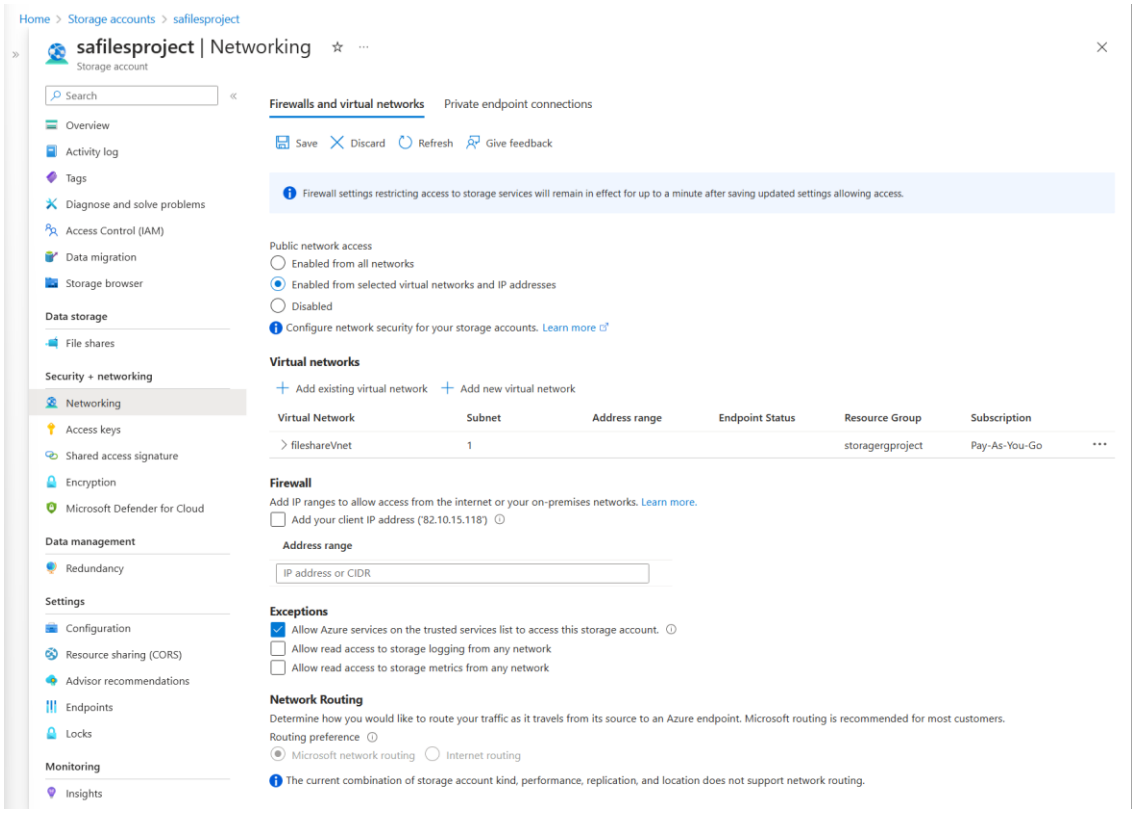


In the **Virtual networks** section, select **Add existing virtual network**

Select your virtual network and subnet, select **Add**

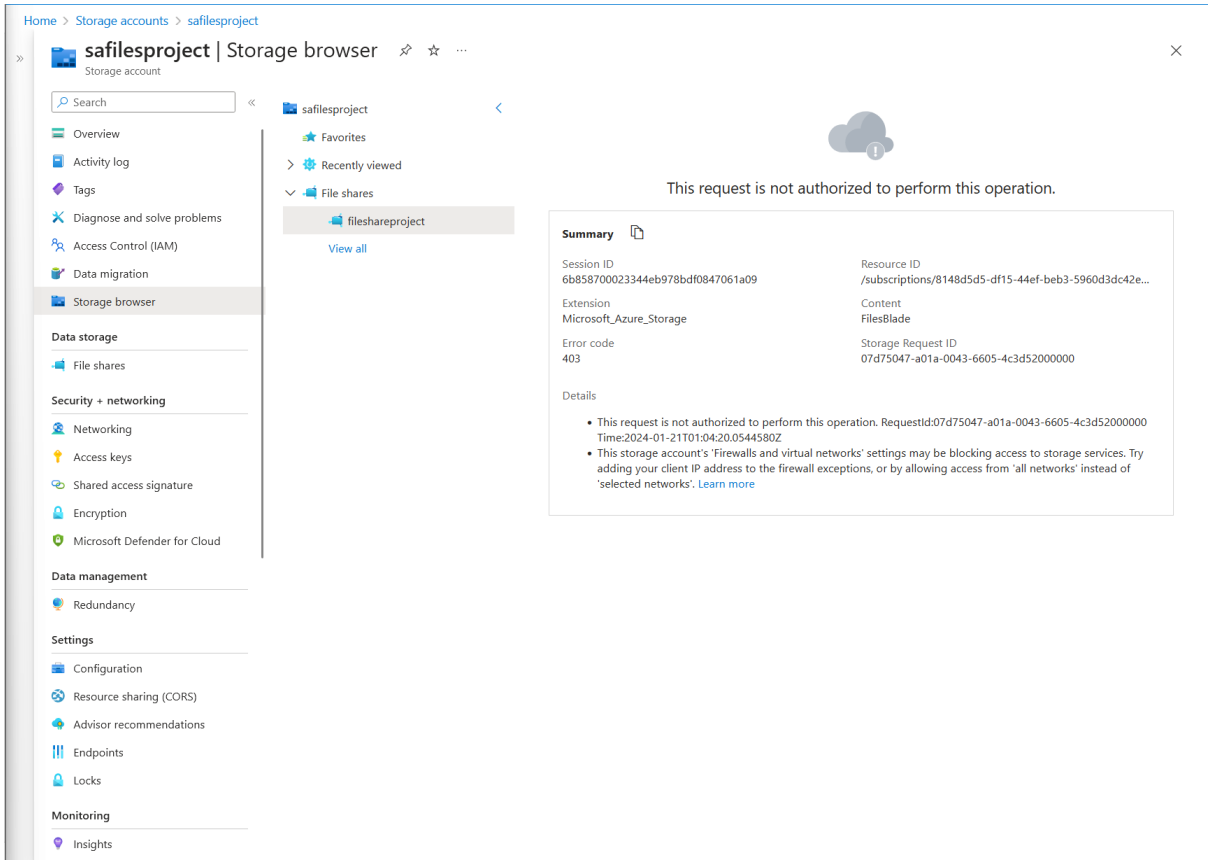


Save settings



Select the **Storage browser** and navigate to your file share

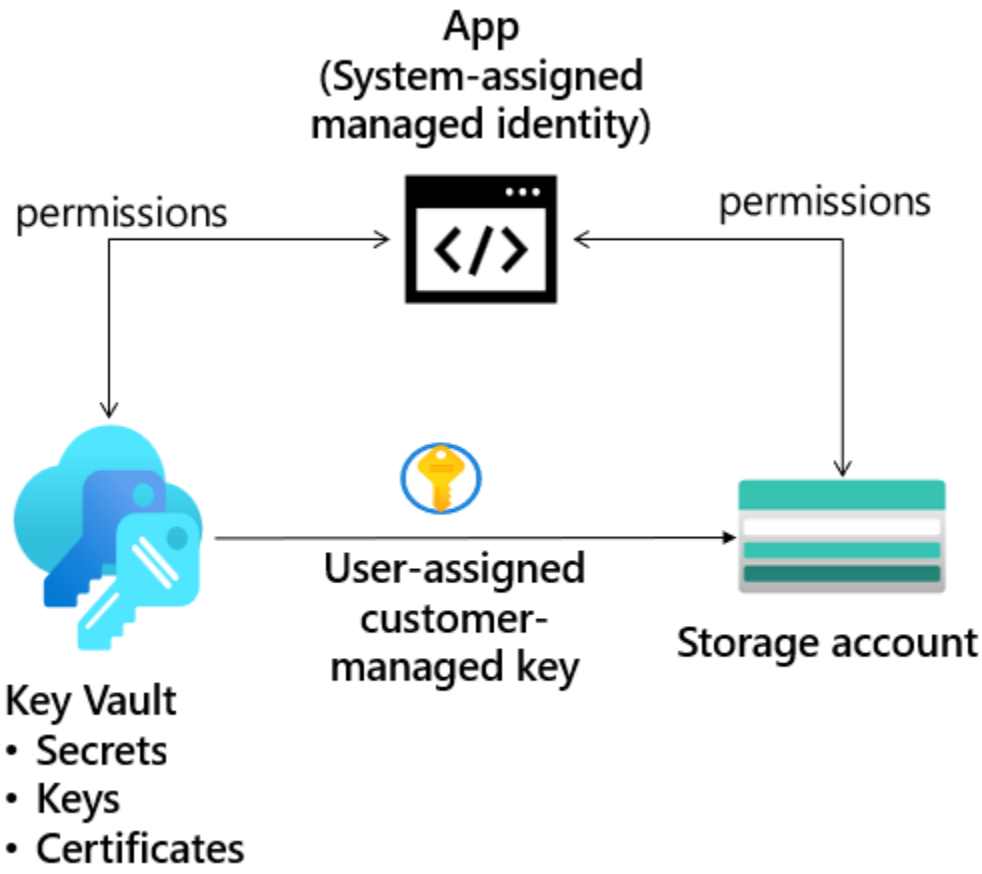
Verify the message *not authorized to perform this operation*. You are not connecting from the virtual network



Provide storage for a new company app

The company is designing and developing a new app. Developers need to ensure the storage is only accessed using keys and managed identities. The developers would like to use role-based access control. To help with testing, protected immutable storage is needed.

Architecture diagram

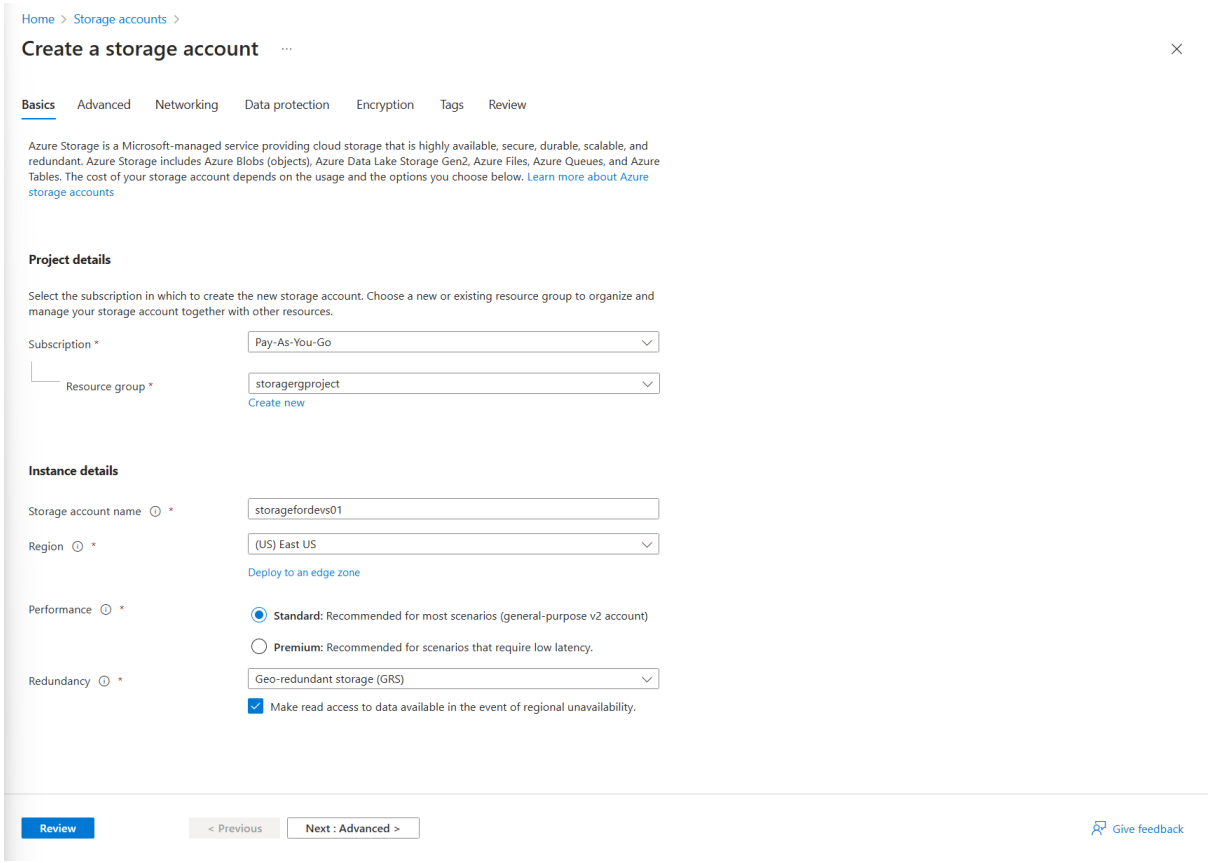


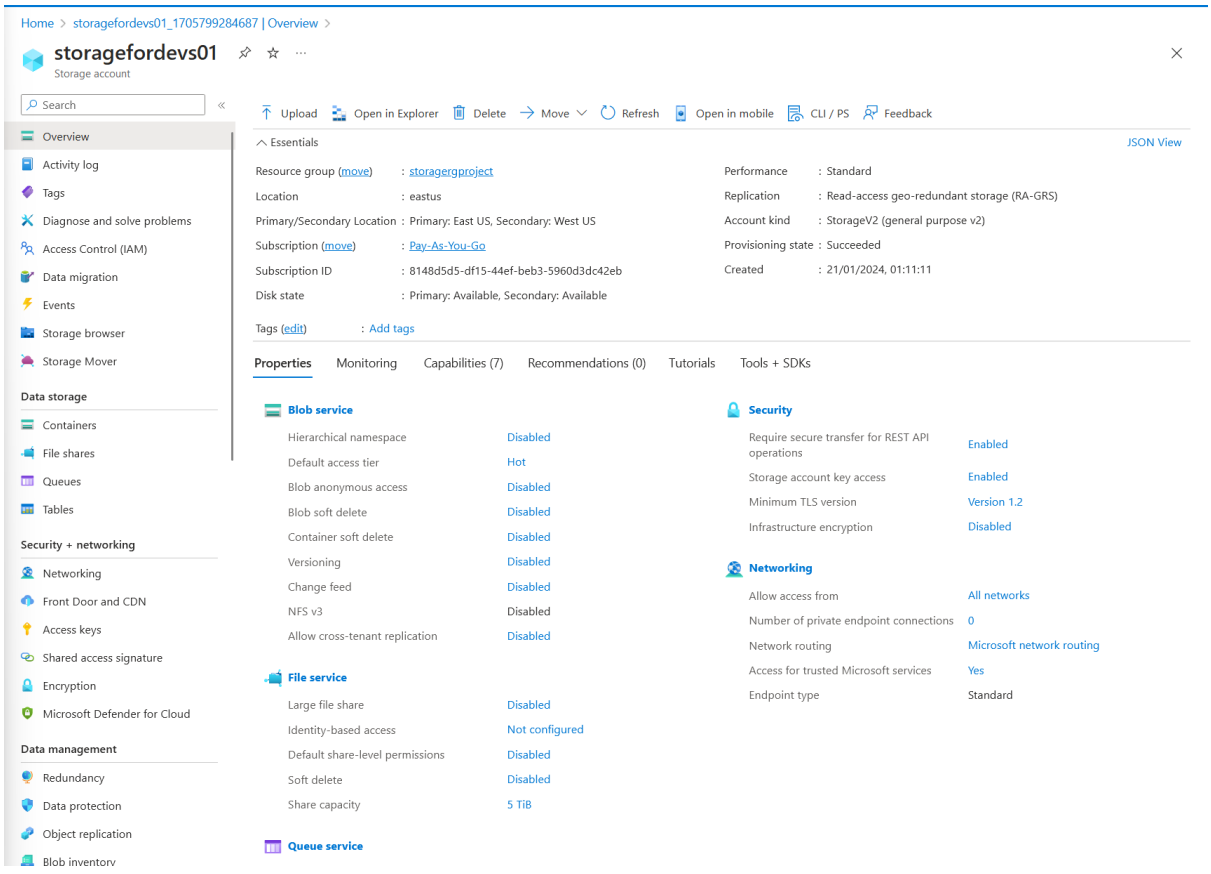
Task to be completed

- Create the storage account and managed identity.
- Secure access to the storage account with a key vault and key.
- Configure the storage account to use the customer managed key in the key vault
- Configure a time-based retention policy and an encryption scope.

Step 1 - Create the storage account and managed identity

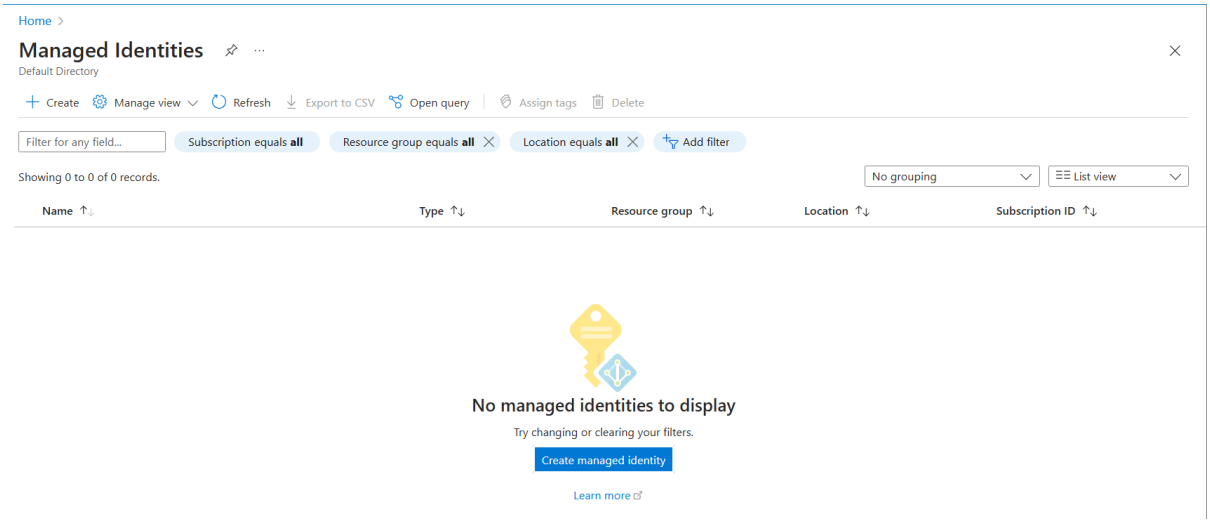
Create a storage account for the web app with the following options





Step 2 - Provide a managed identity for the web app to use

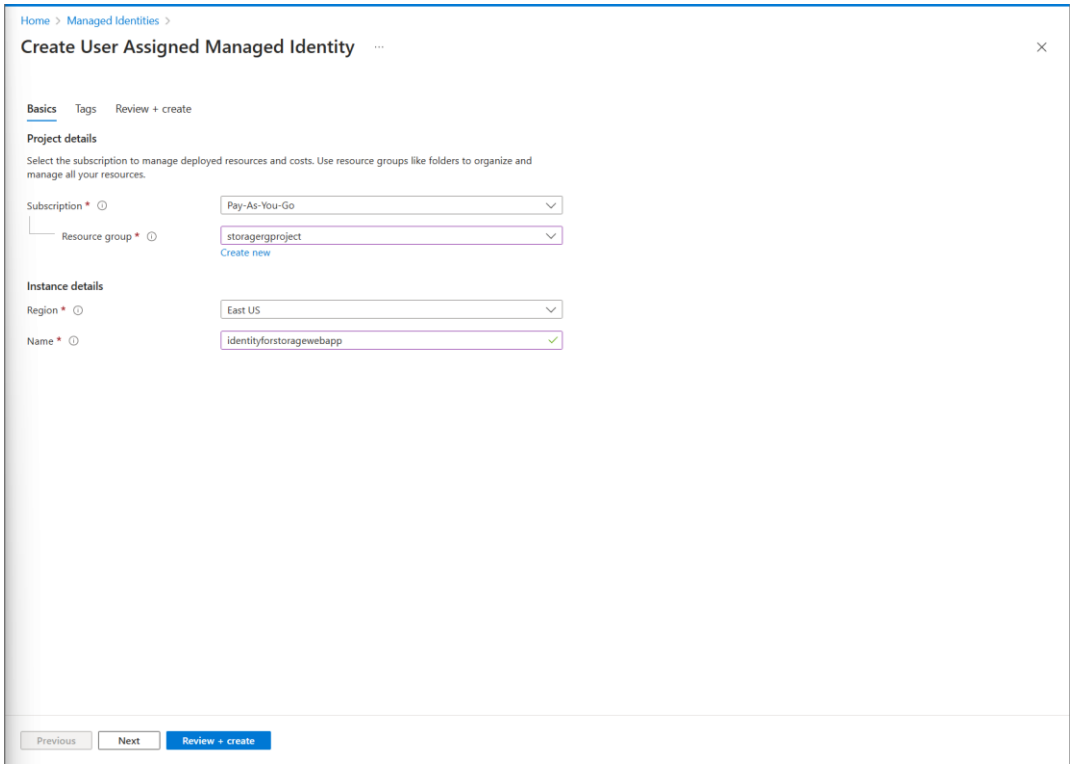
Search for and select **Managed identities**



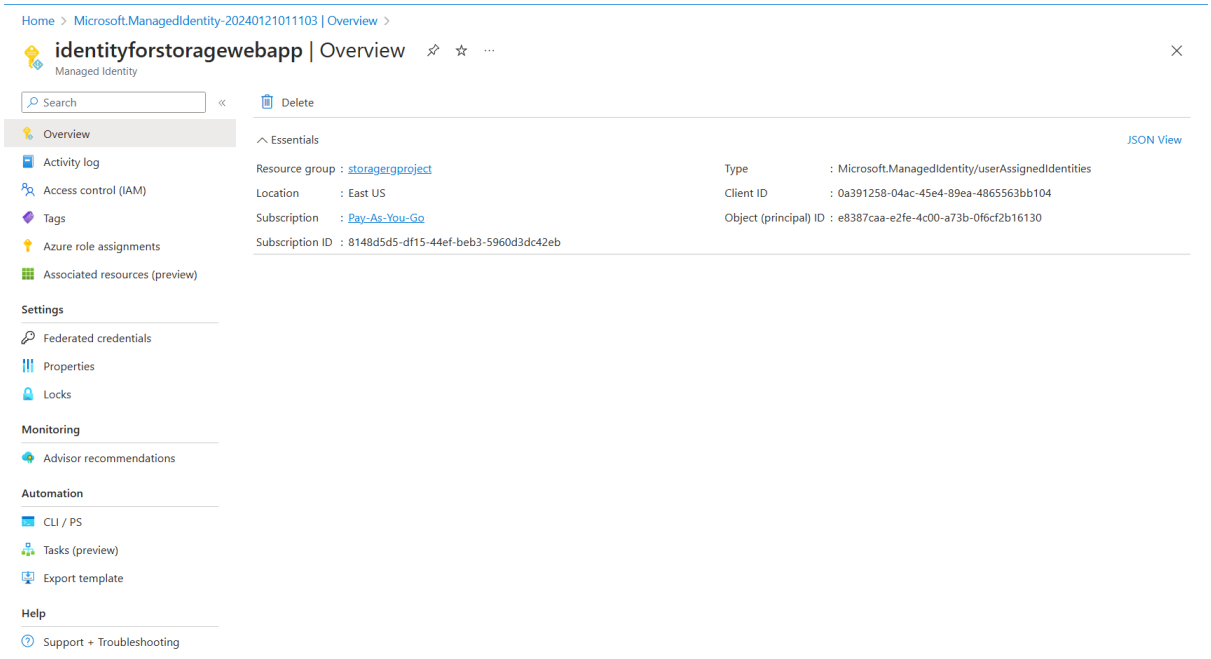
Select **Create**

Select **resource group**

Give your managed identity a name

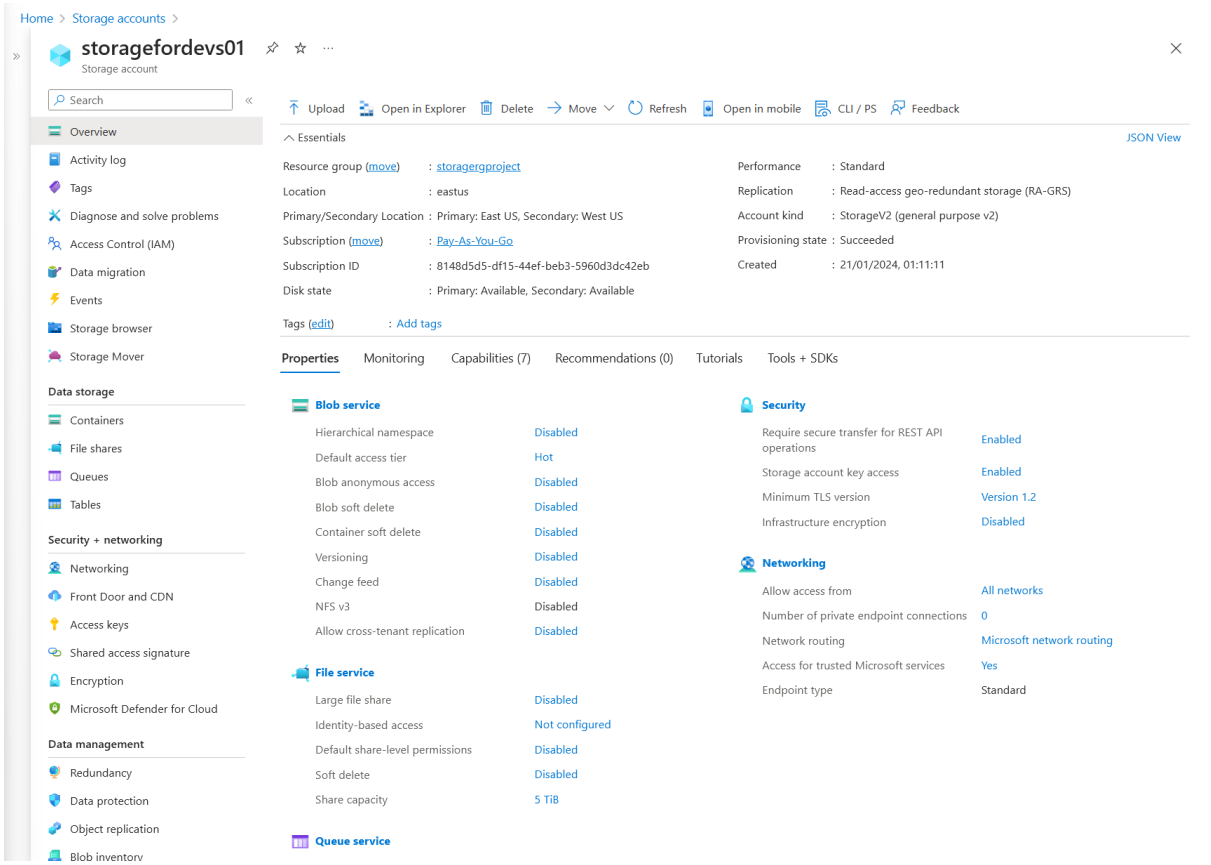


Review and create

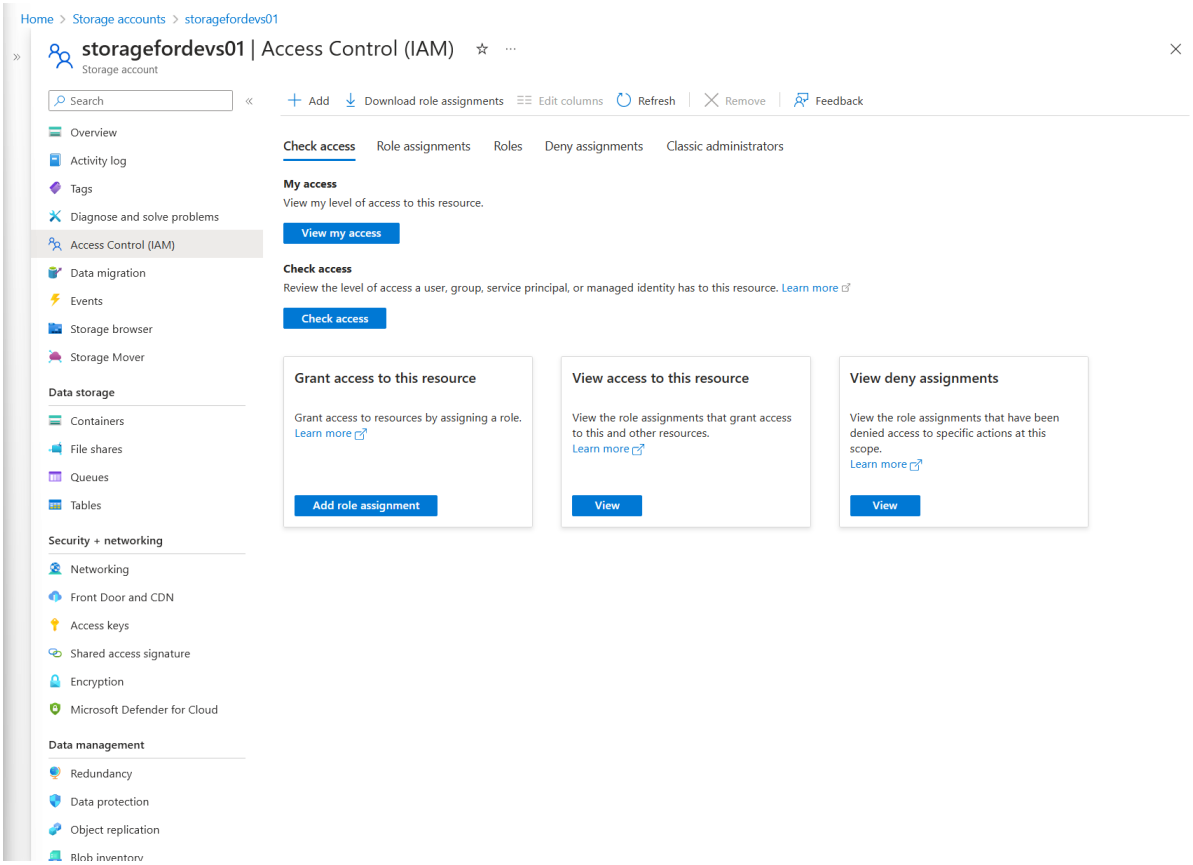


Step 3 - Assign the correct permissions to the managed identity. The identity only needs to read and list containers and blobs

Search for and select your **storage account**

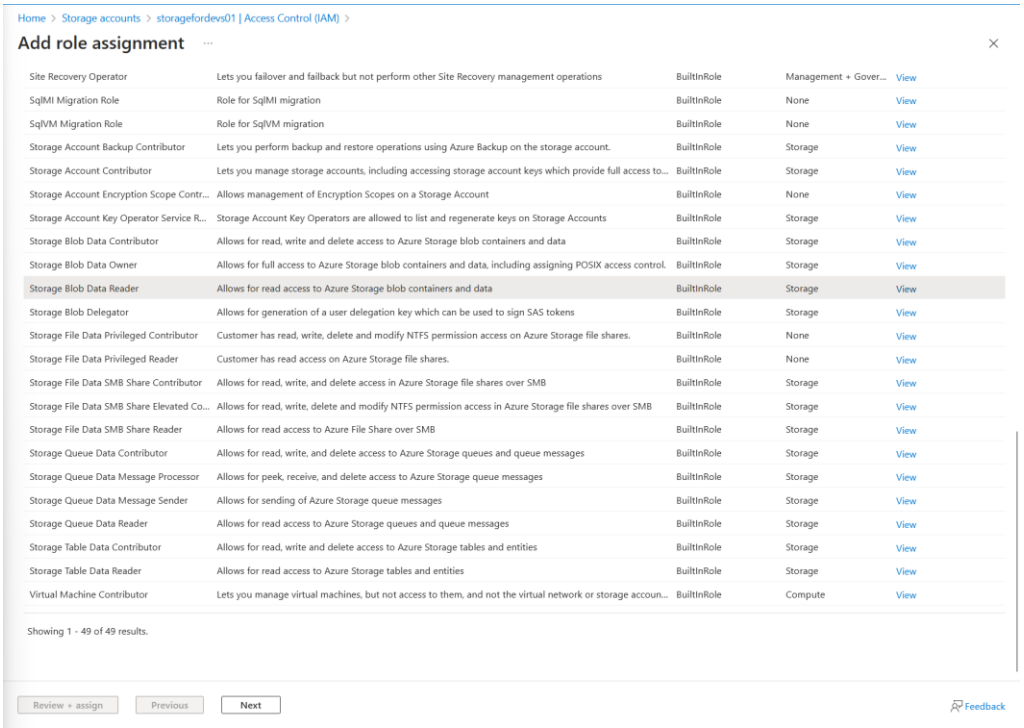


Select the **Access Control (IAM)** blade

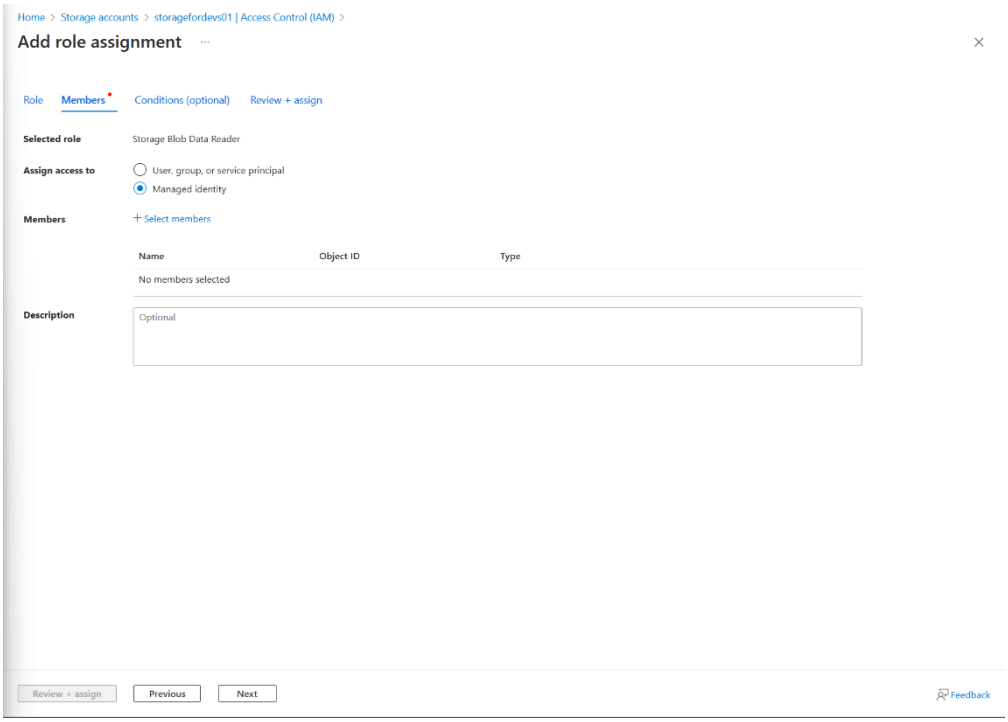


Select **Add role assignment**

On the **Job functions roles** page, search for and select the **Storage Blob Data Reader** role

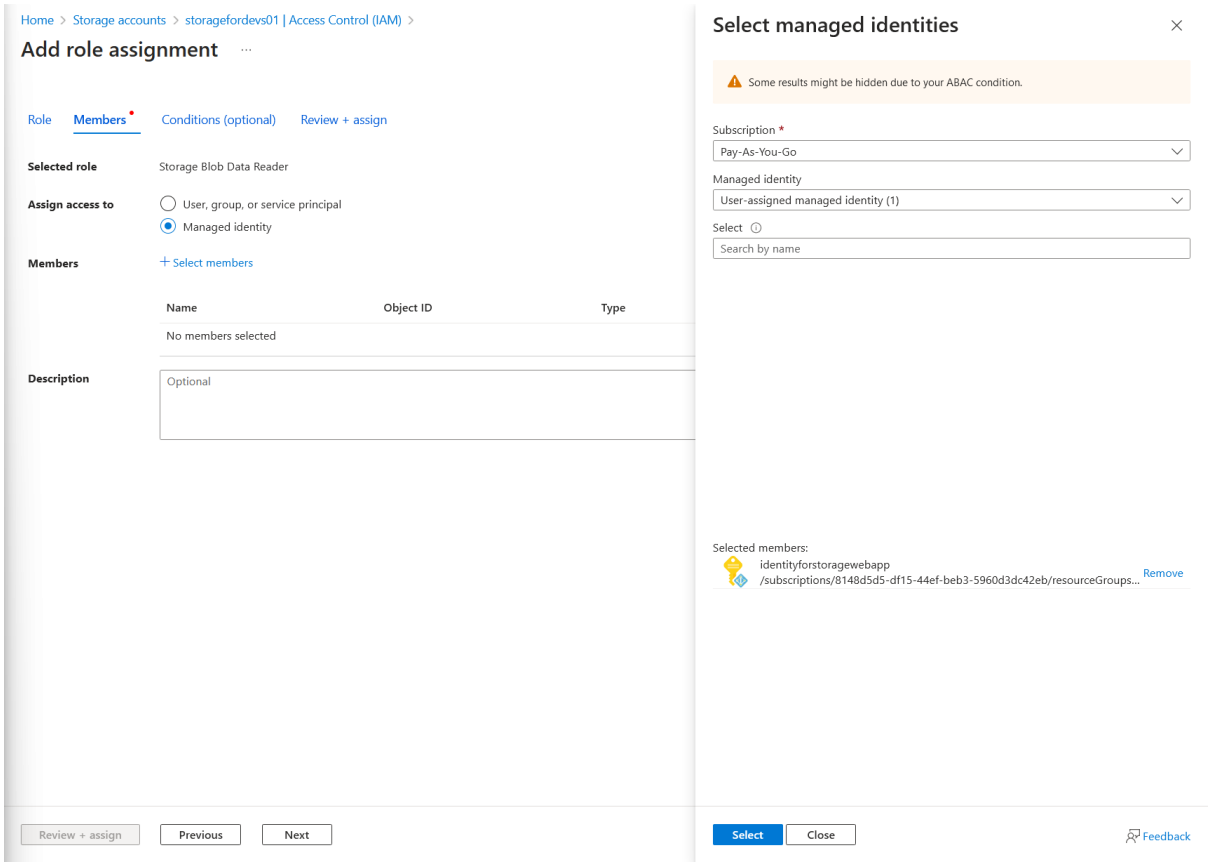


On the **Members** page, select **Managed identity**

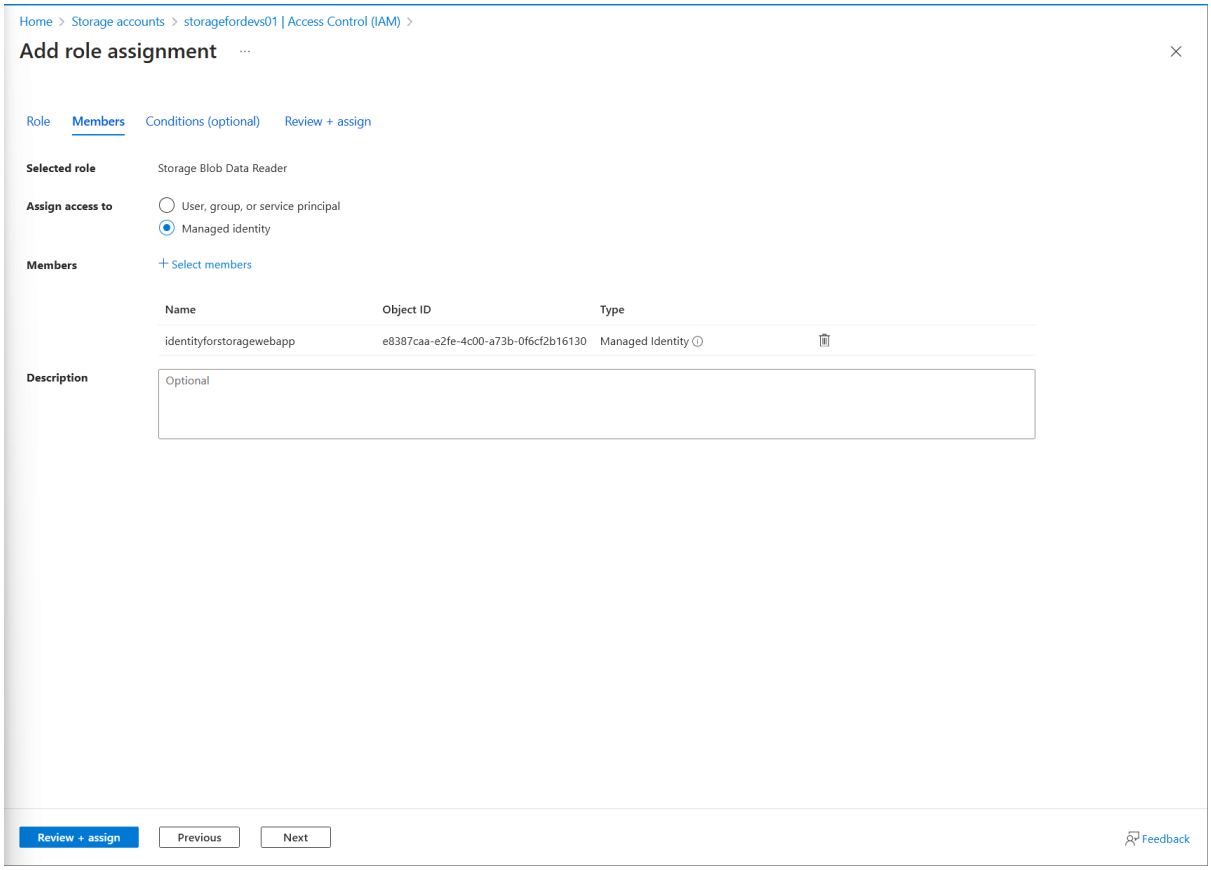


Select **Select members**, in the **Managed identity** drop-down select **User-assigned managed identity**

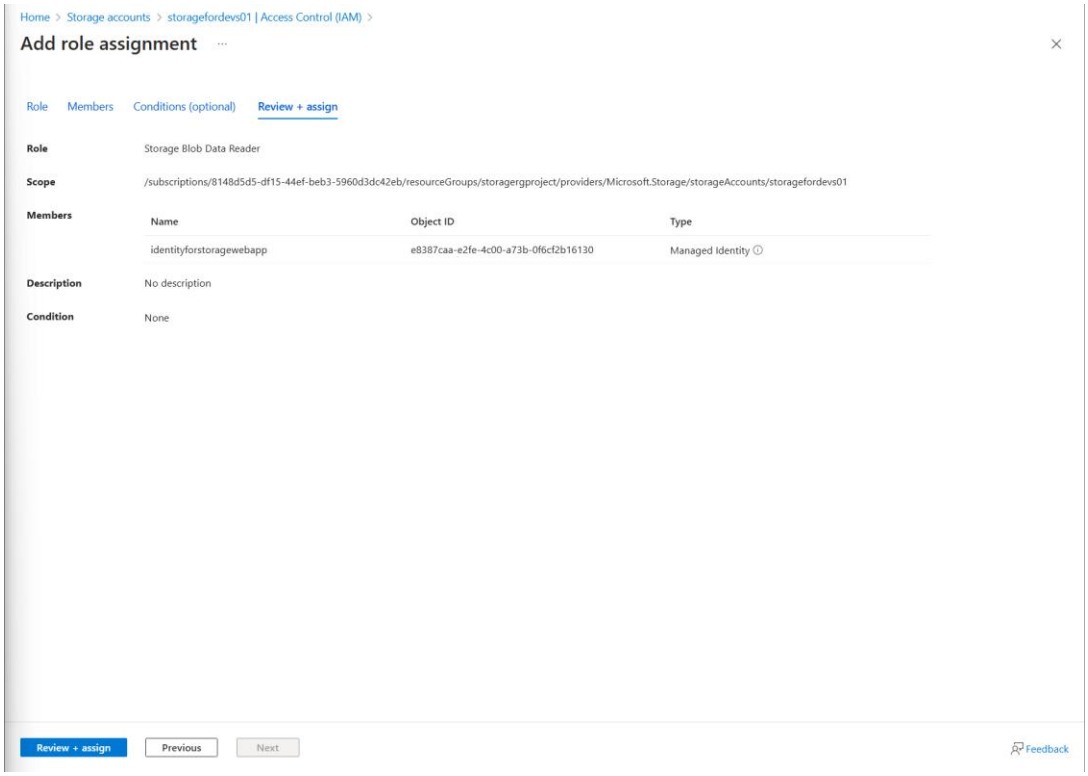
Select the managed identity



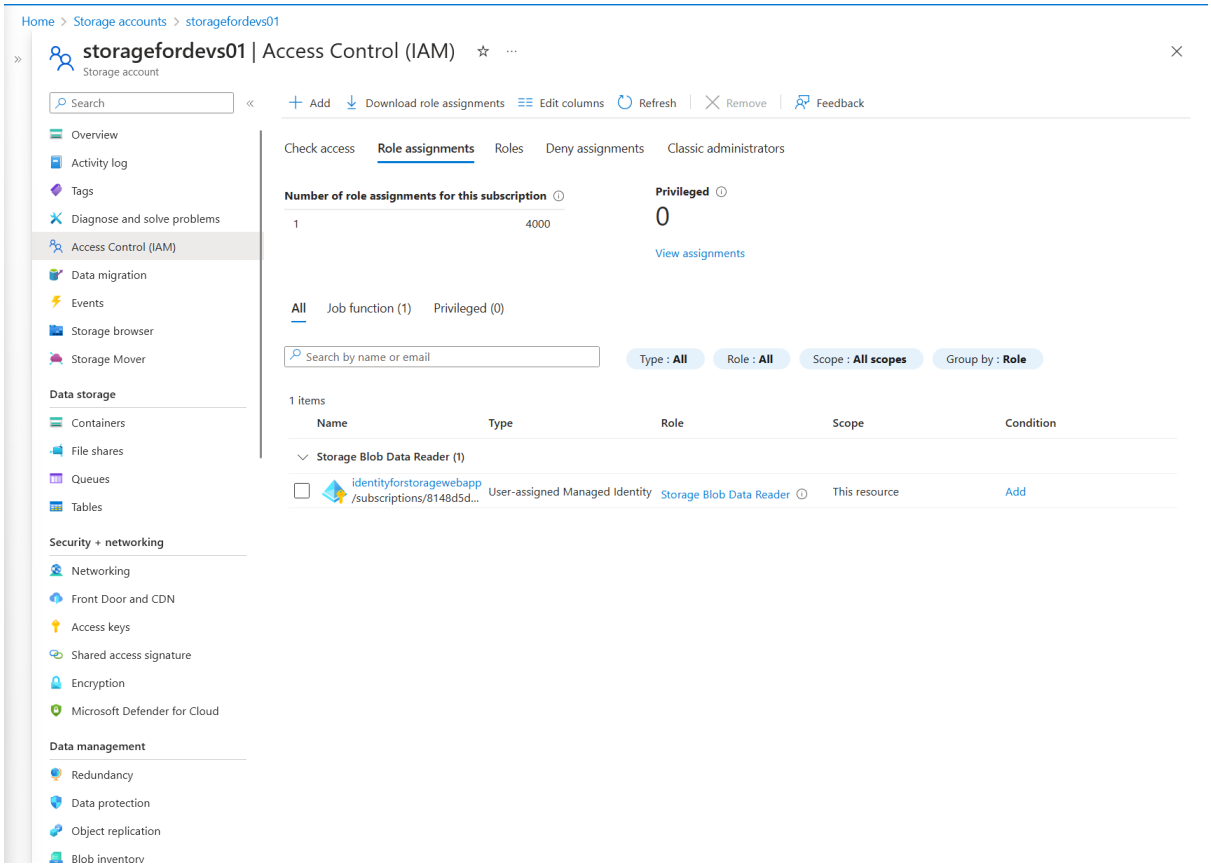
Click **Select** and then **Review + assign** the role



Select **Review + assign** a second time to add the role assignment



Storage account can now be accessed by a managed identity with the Storage Data Blob Reader permissions



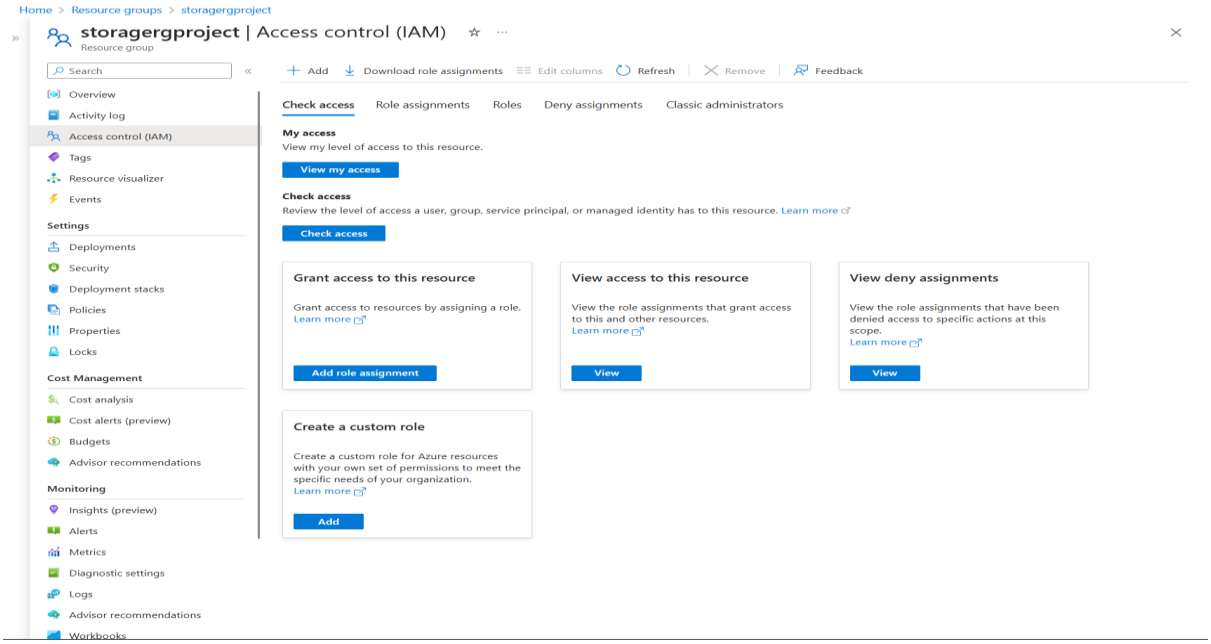
Step 4 - Secure access to the storage account with a key vault and key

To create the key vault and key needed for this part your user account must have Key Vault Administrator permissions

In the portal, search for and select **Resource groups**

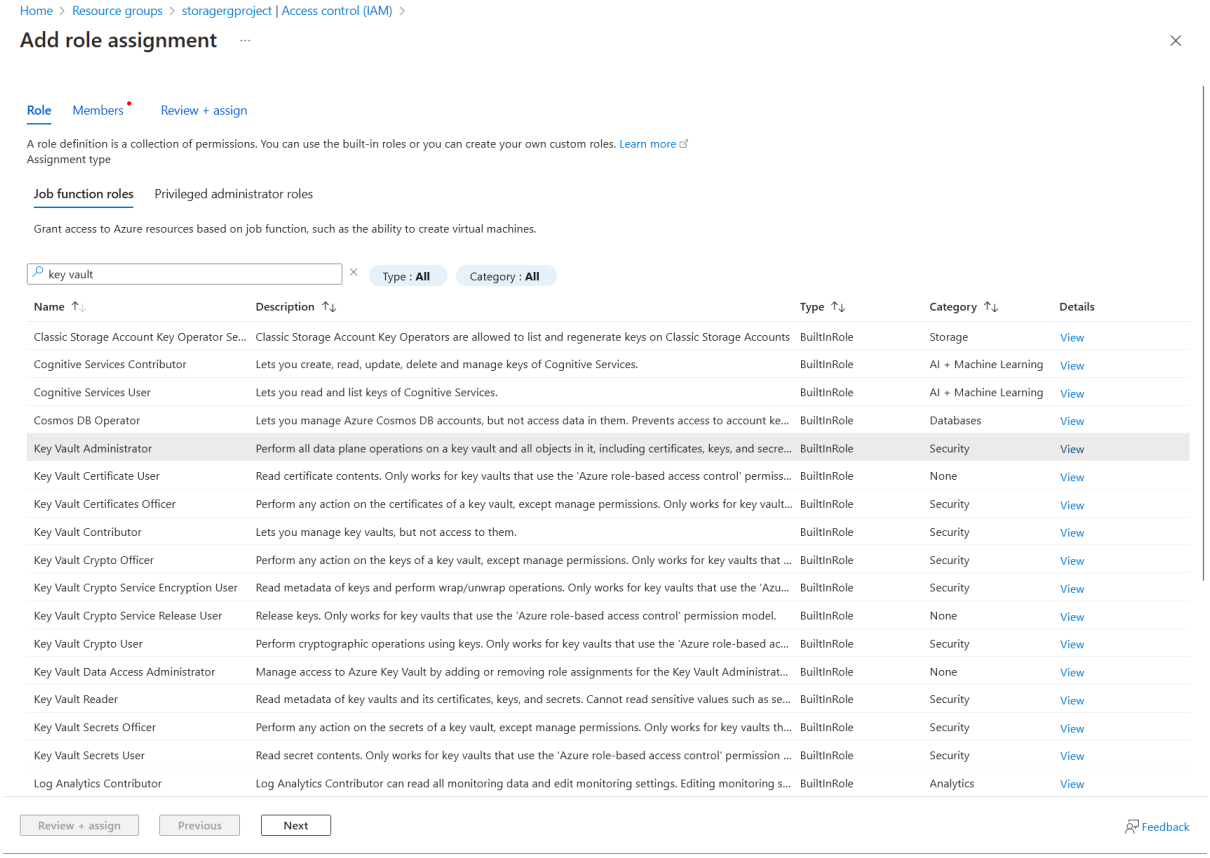
Select your **resource group**, and then the **Access Control (IAM)** blade.

Next Image



Select **Add role assignment**

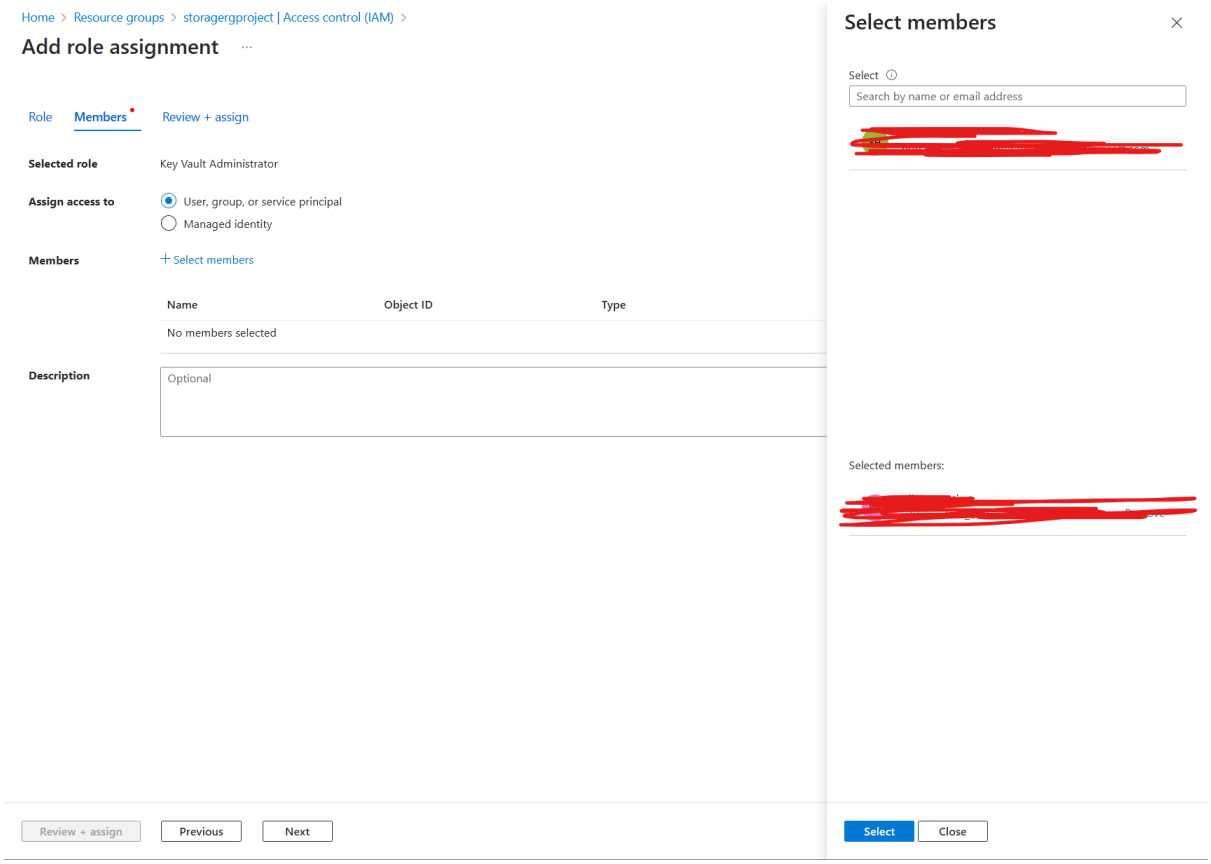
On the **Job functions roles** page, search for and select the **Key Vault Administrator** role



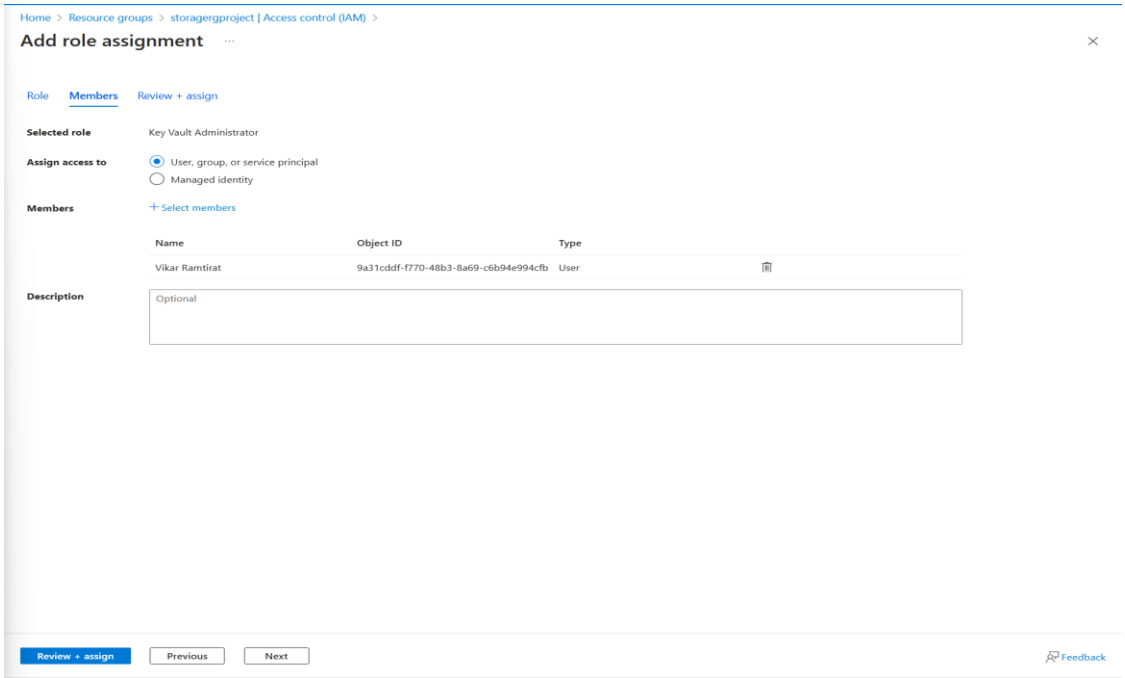
On the **Members** page, select **User, group, or service principal**

Select **Select members**

Search for and select your user account. Your user account is shown in the top right of the portal

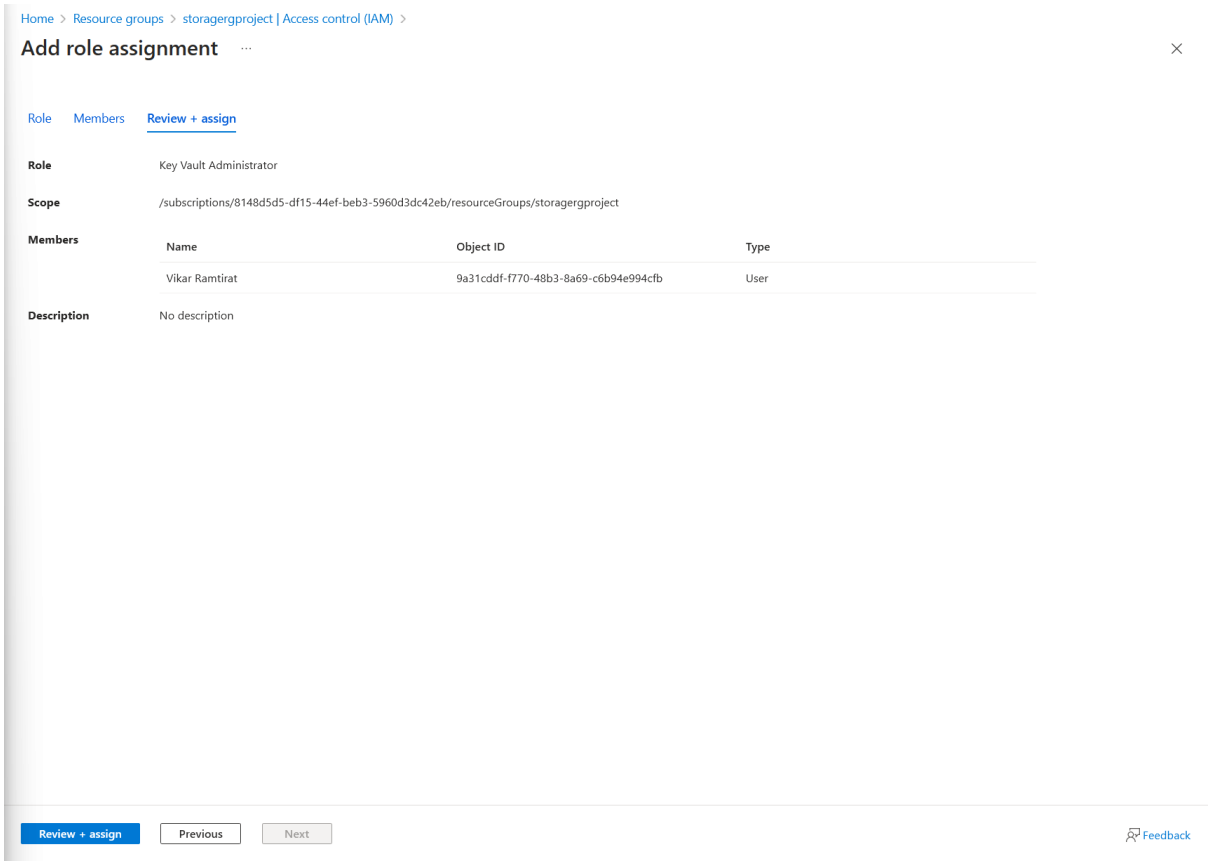


Hid my azure account for security reasons.



Click **Select** and then **Review + assign**

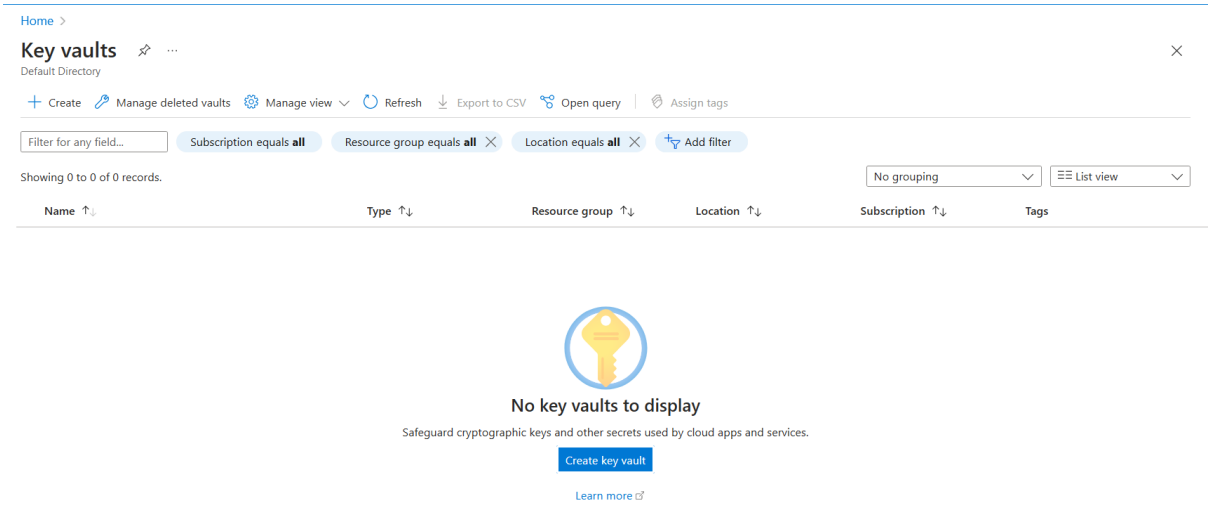
Select **Review + assign** a second time to add the role assignment

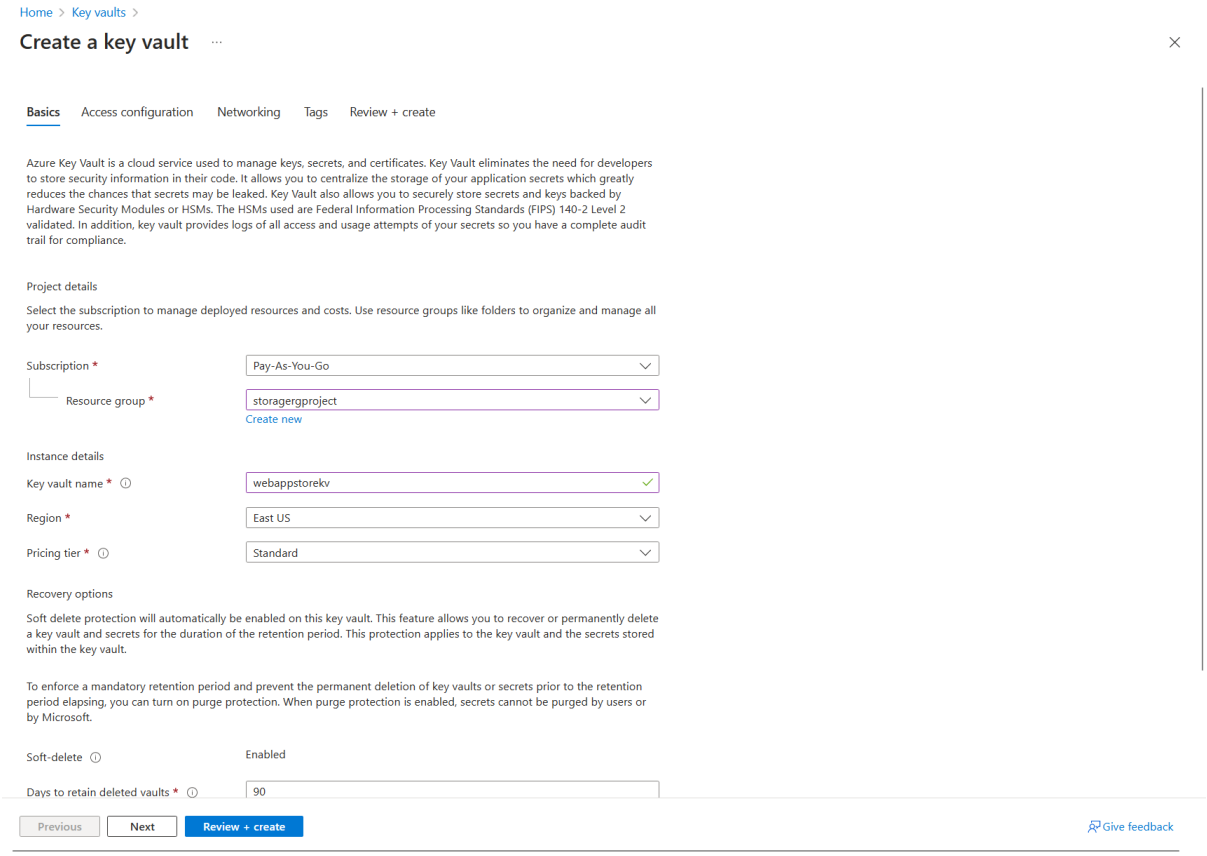


Step 5 - Create a key vault to store the access keys

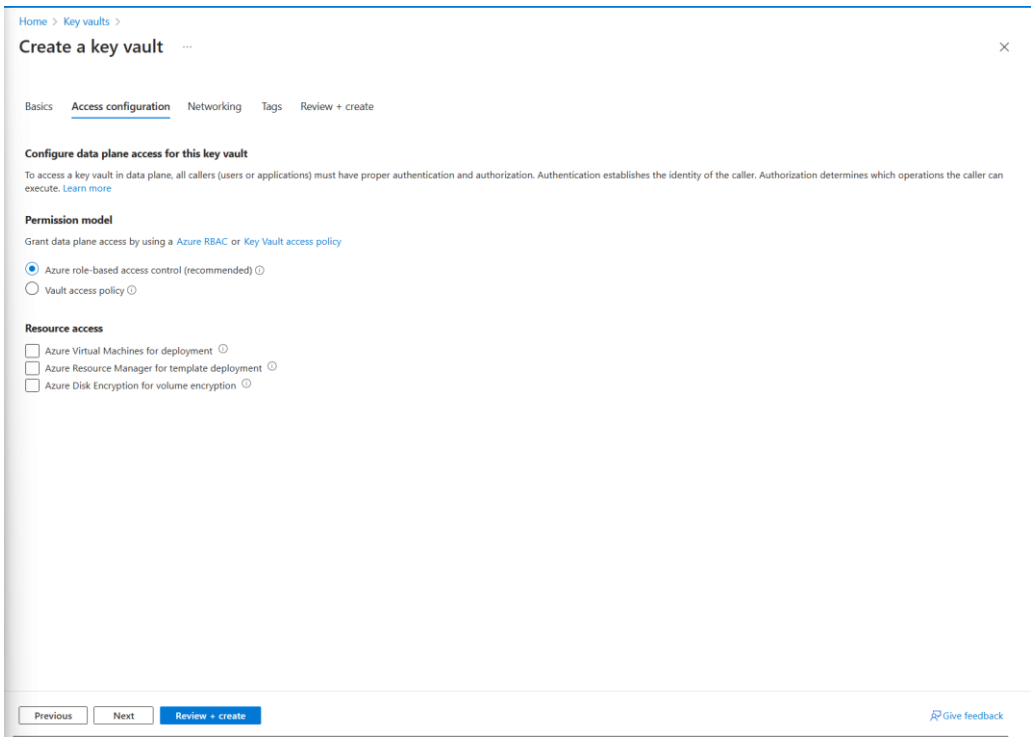
In the portal, search for and select **Key vaults**

Select **Create**



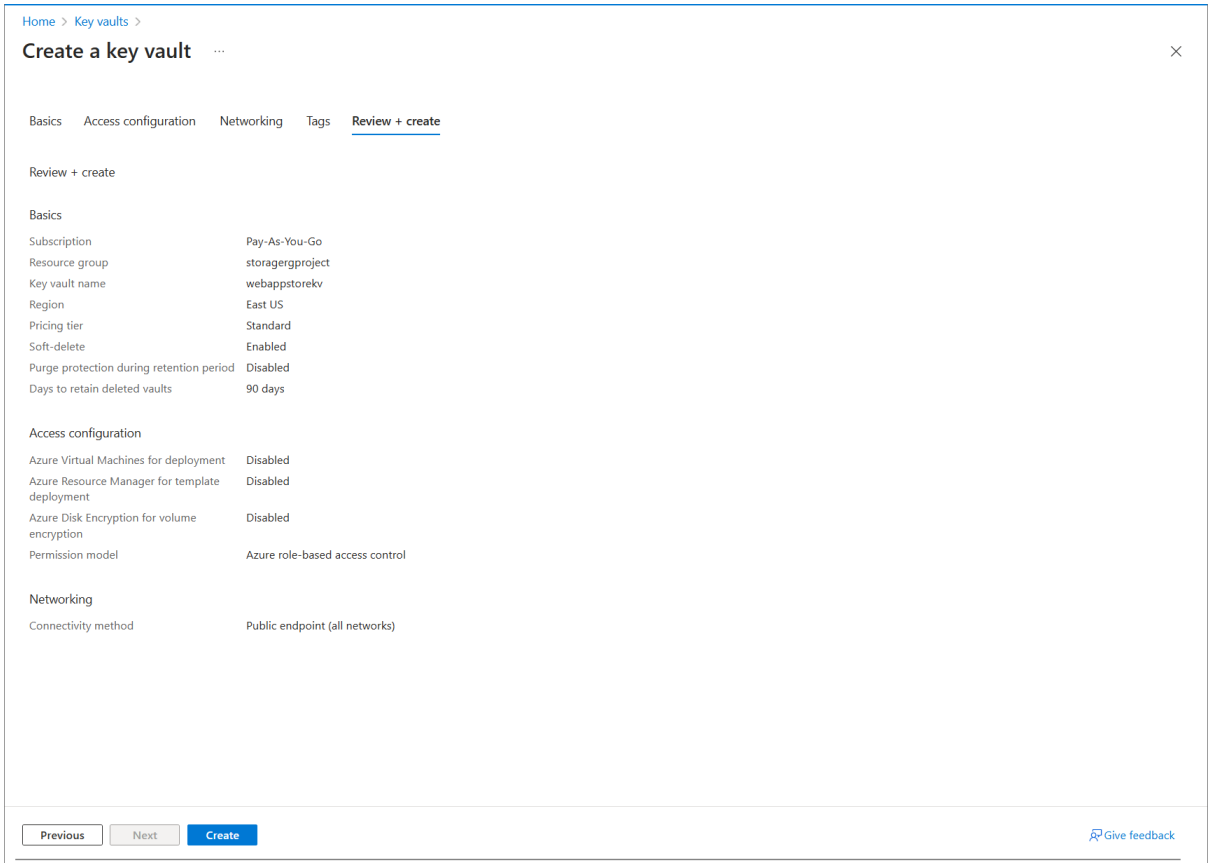


Ensure on the **Access configuration** tab that **Azure role-based access control (recommended)** is selected

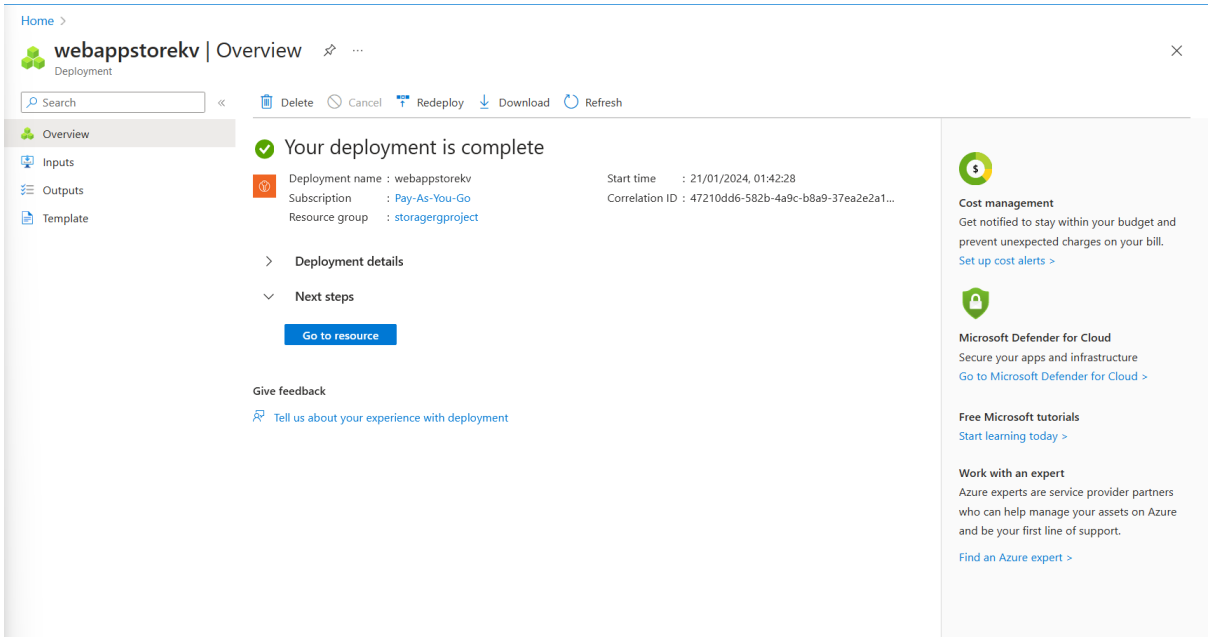


Review and create

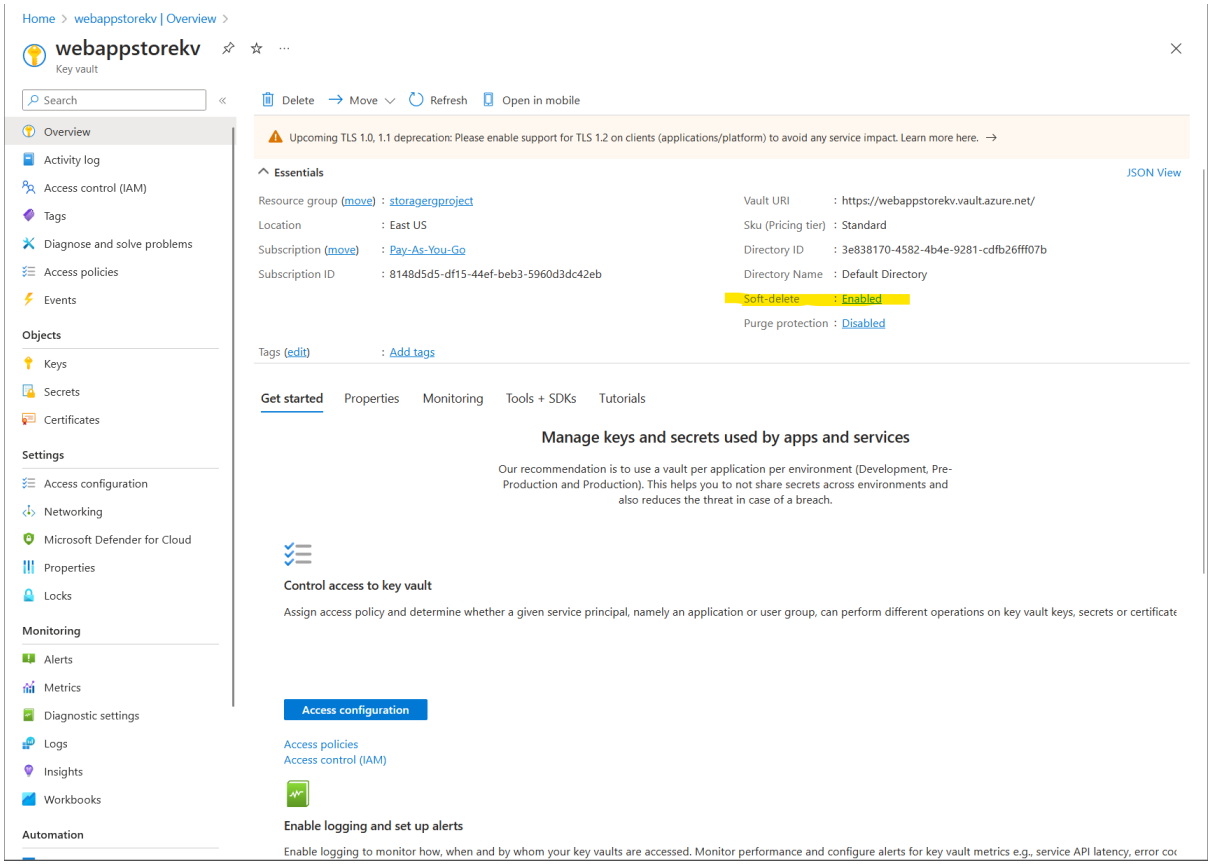
Wait for the validation checks to complete and then select **Create**



After the deployment, select **Go to resource**

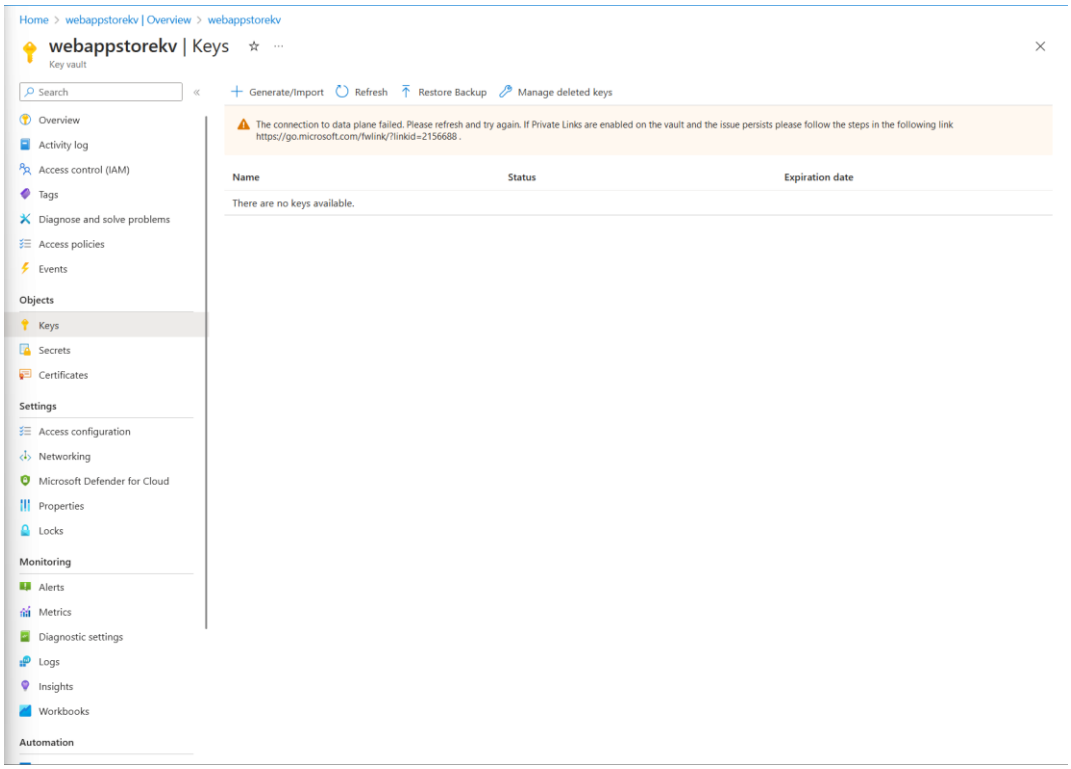


On the **Overview** blade ensure both **Soft-delete** and **Purge protection** are **enabled**



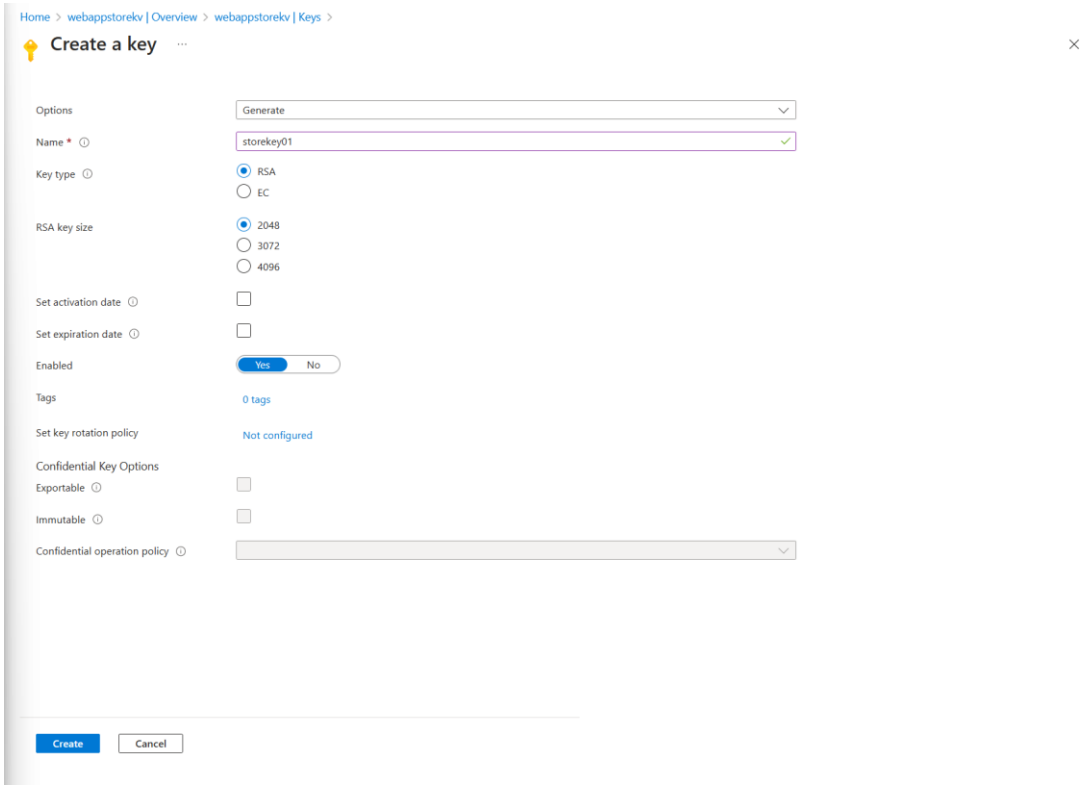
Step 5 - Create a customer-managed key in the key vault

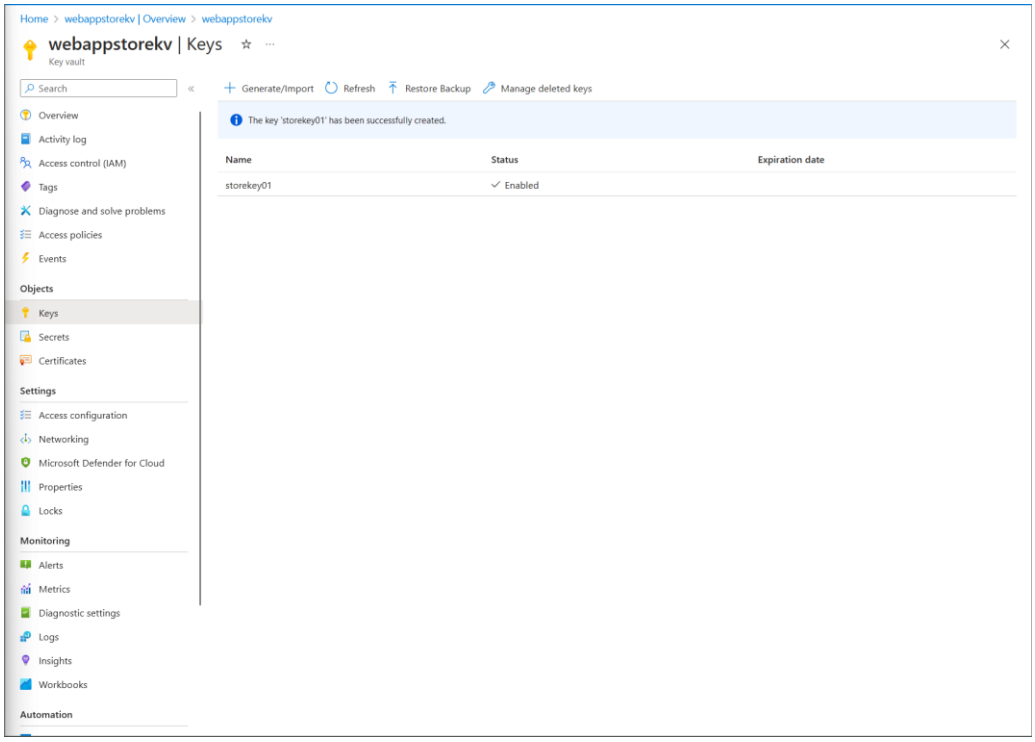
In your **key vault**, in the **Objects** section, select the **Keys** blade



Select **Generate/Import** and **Name** the key

Take the defaults for the rest of the parameters, and **Create** the key



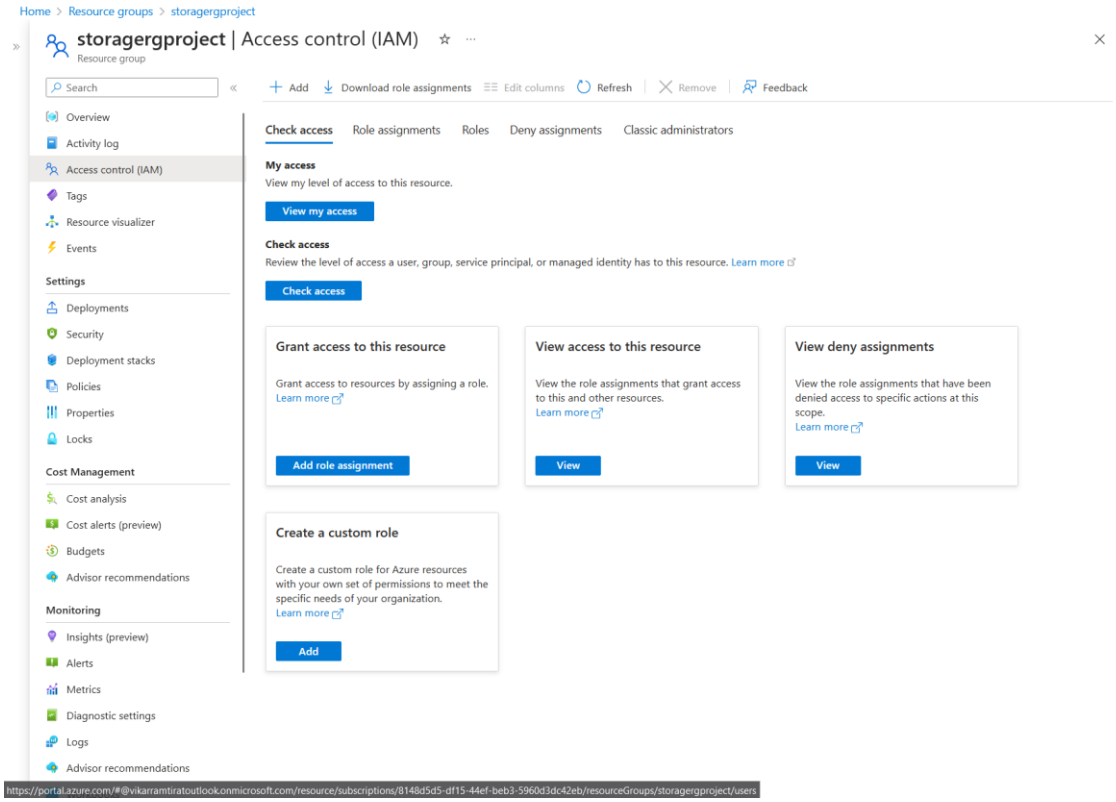


Step 6 - Configure the storage account to use the customer managed key in the key vault

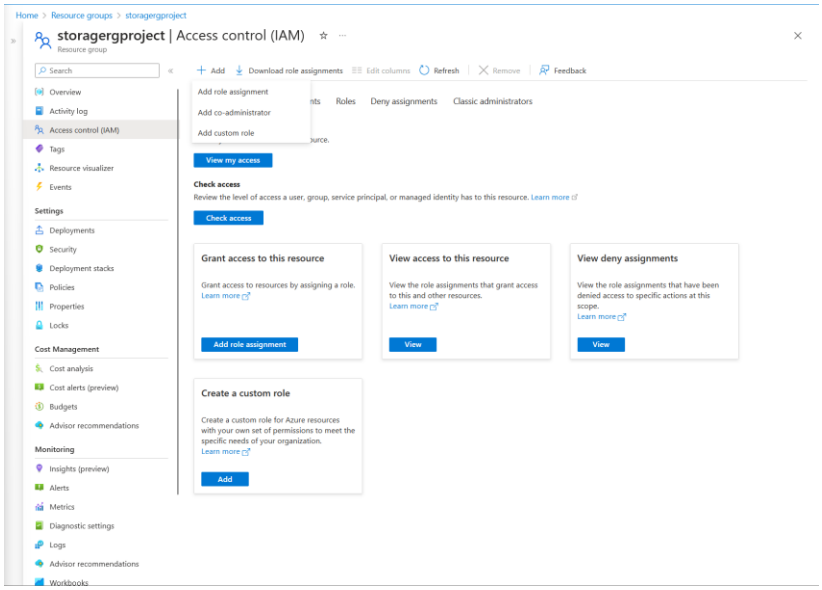
Before you can complete the next steps, you must assign the Key Vault Crypto Service Encryption User role to the managed identity

In the portal, search for and select **Resource groups**

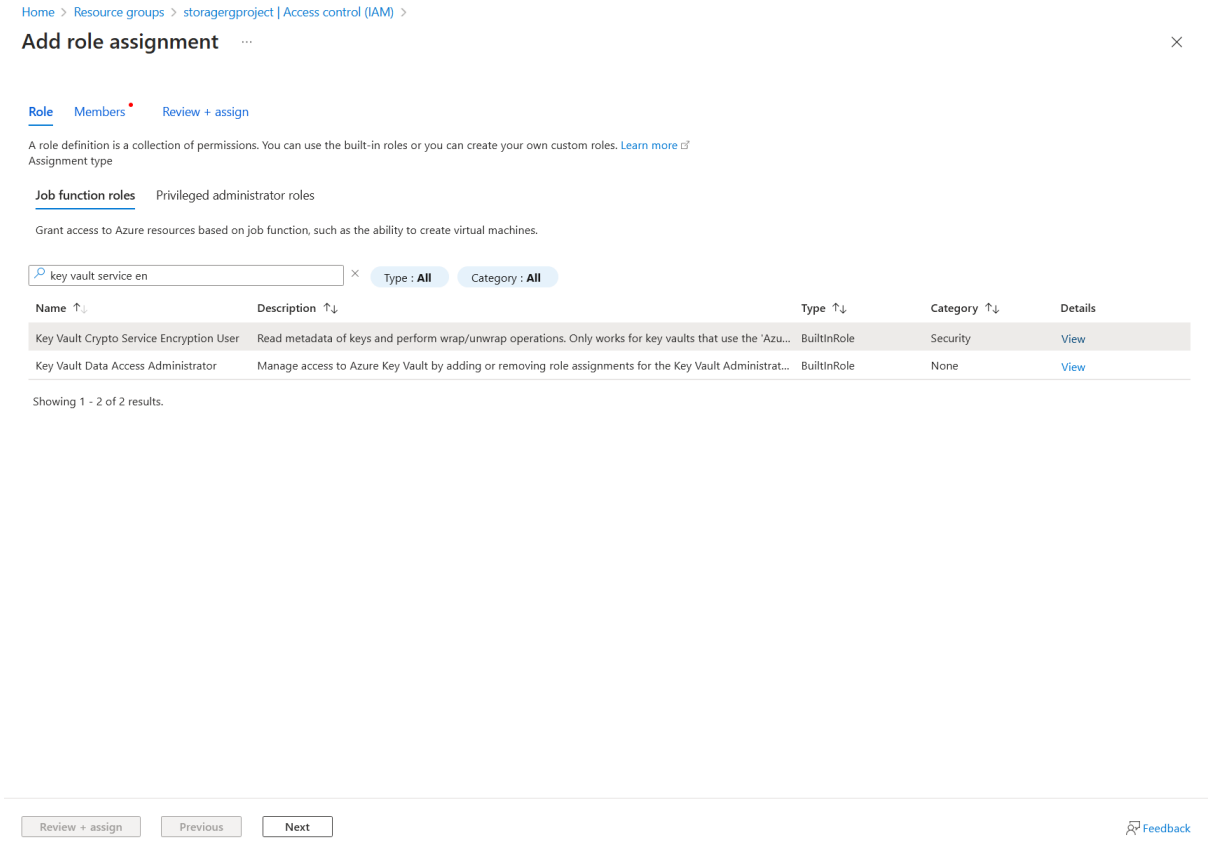
Select your **resource group**, and then the **Access Control (IAM)** blade



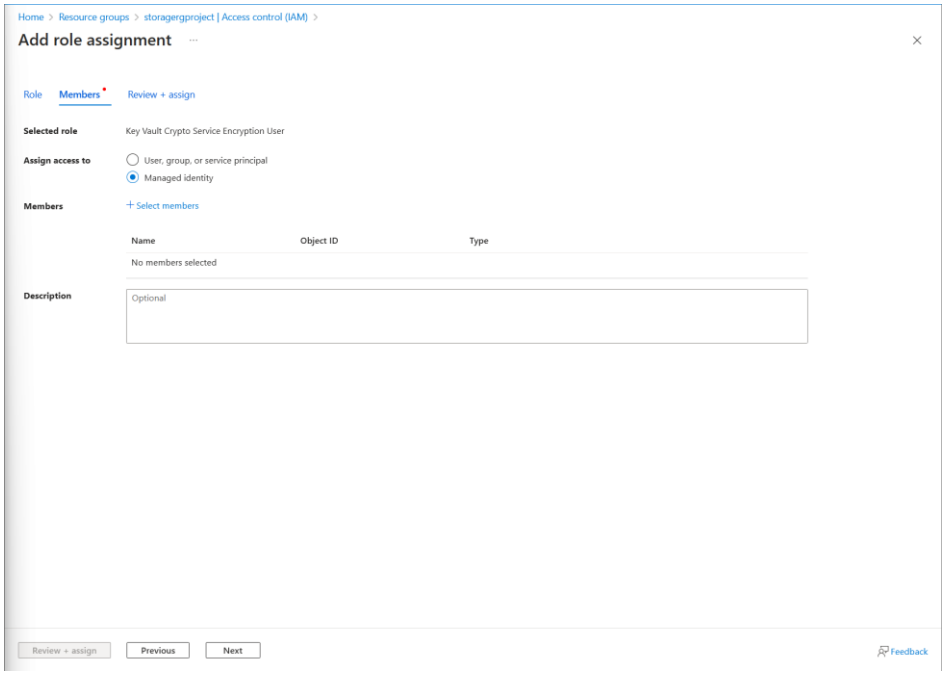
Select **Add role assignment**



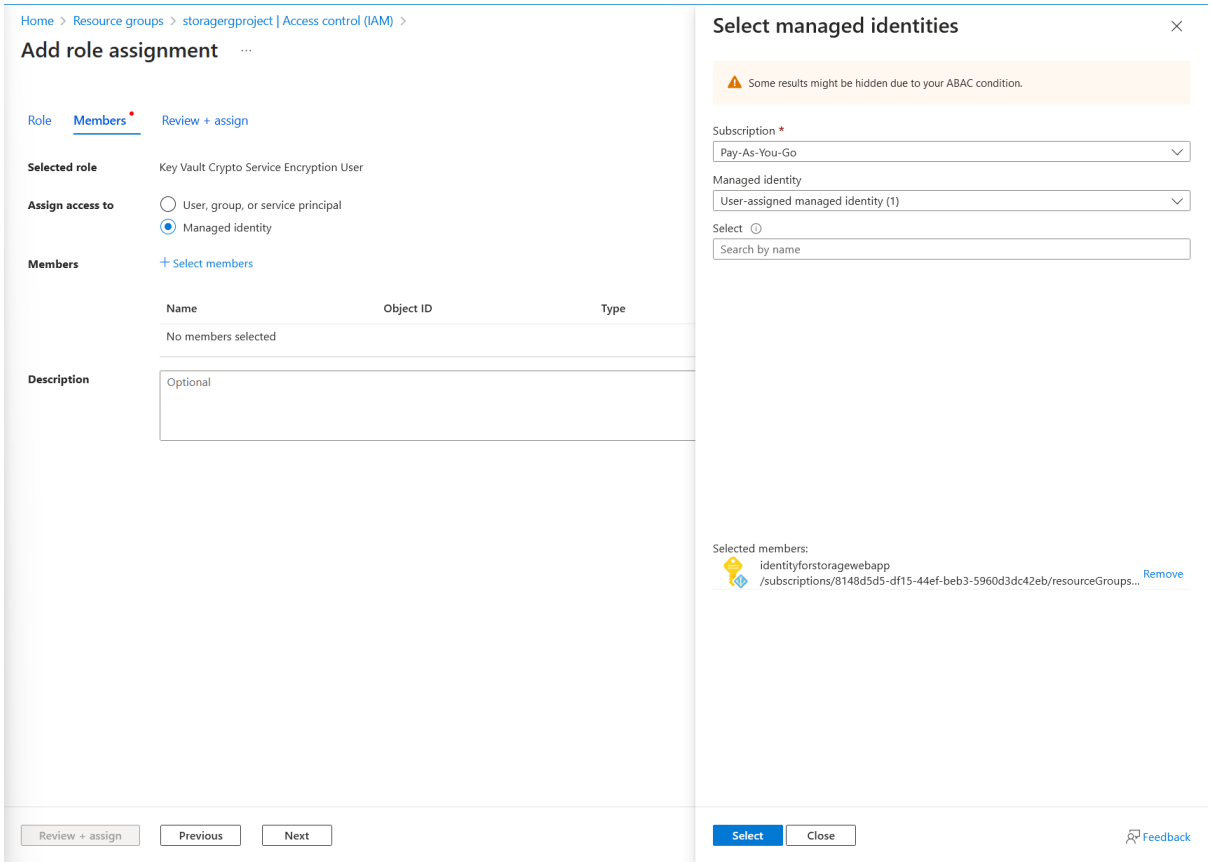
On the **Job functions roles** page, search for and select the **Key Vault Crypto Service Encryption User** role



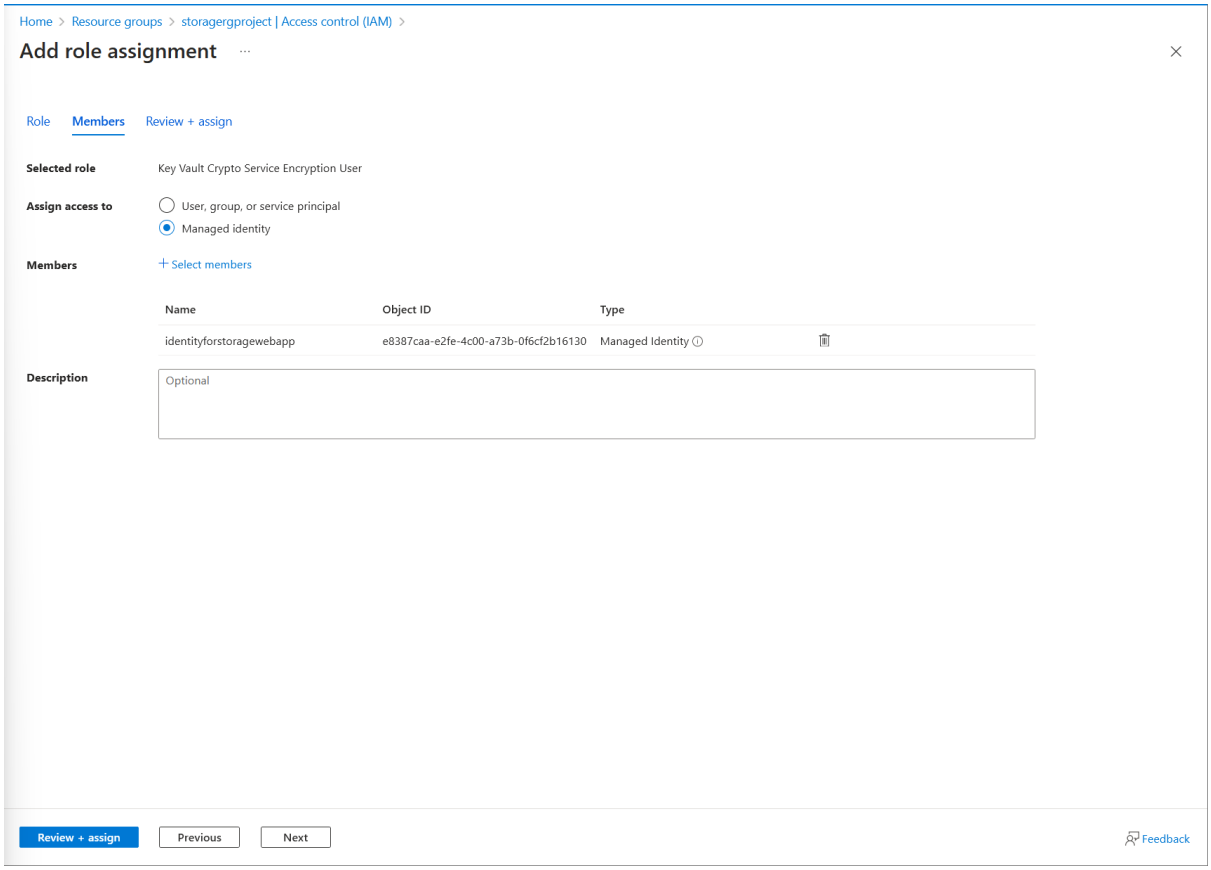
On the **Members** page, select **Managed identity**



Select **Select members**, in the **Managed identity** drop-down select **User-assigned managed identity**

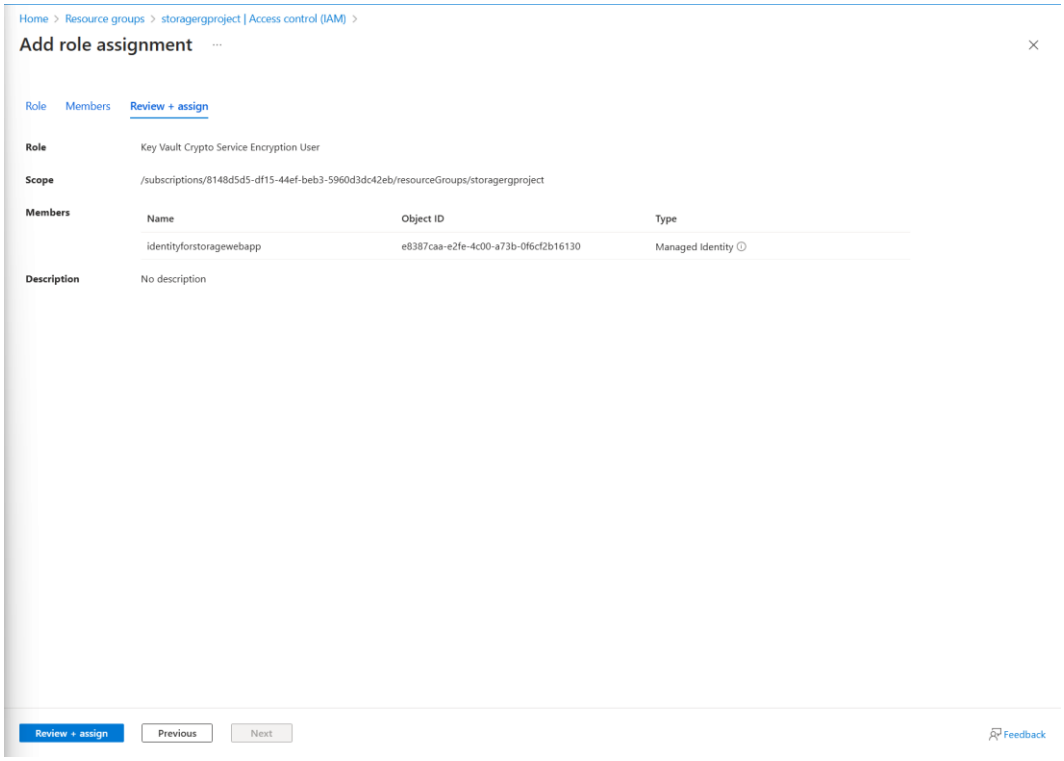


Select the managed identity



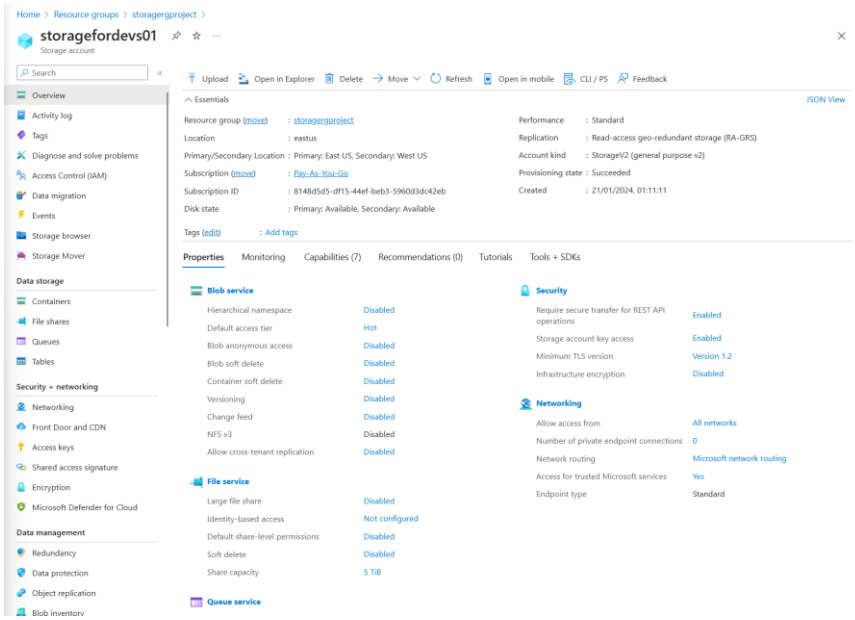
Click **Select** and then **Review + assign**

Select **Review + assign** a second time to add the role assignment

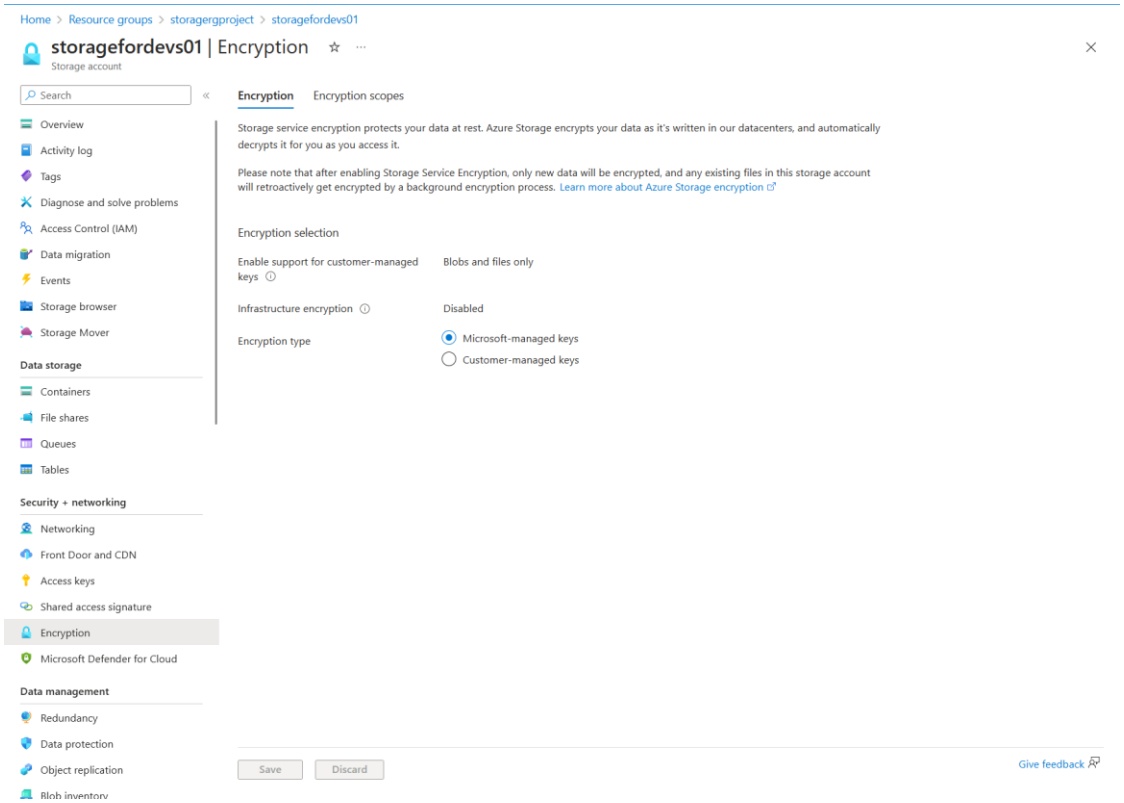


Step 7 - Configure the storage account to use the customer managed key in your key vault

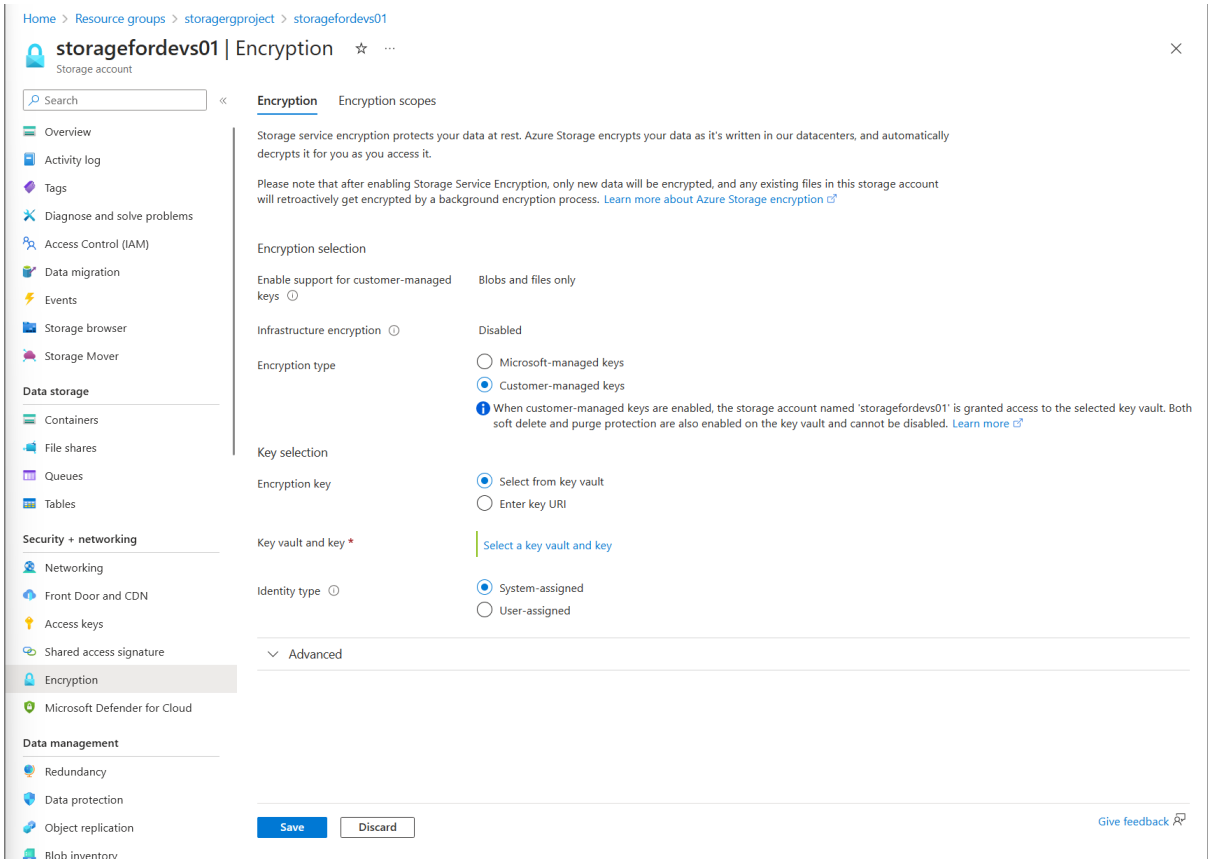
Return to the storage account



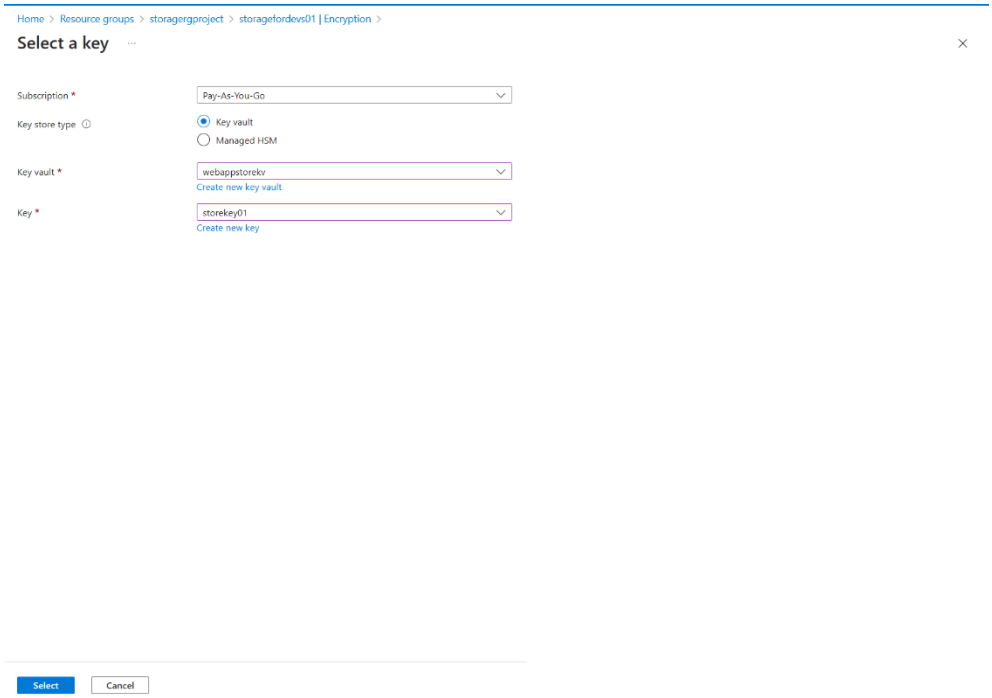
In the **Security + networking** section, select the **Encryption** blade



Select **Customer-managed keys**

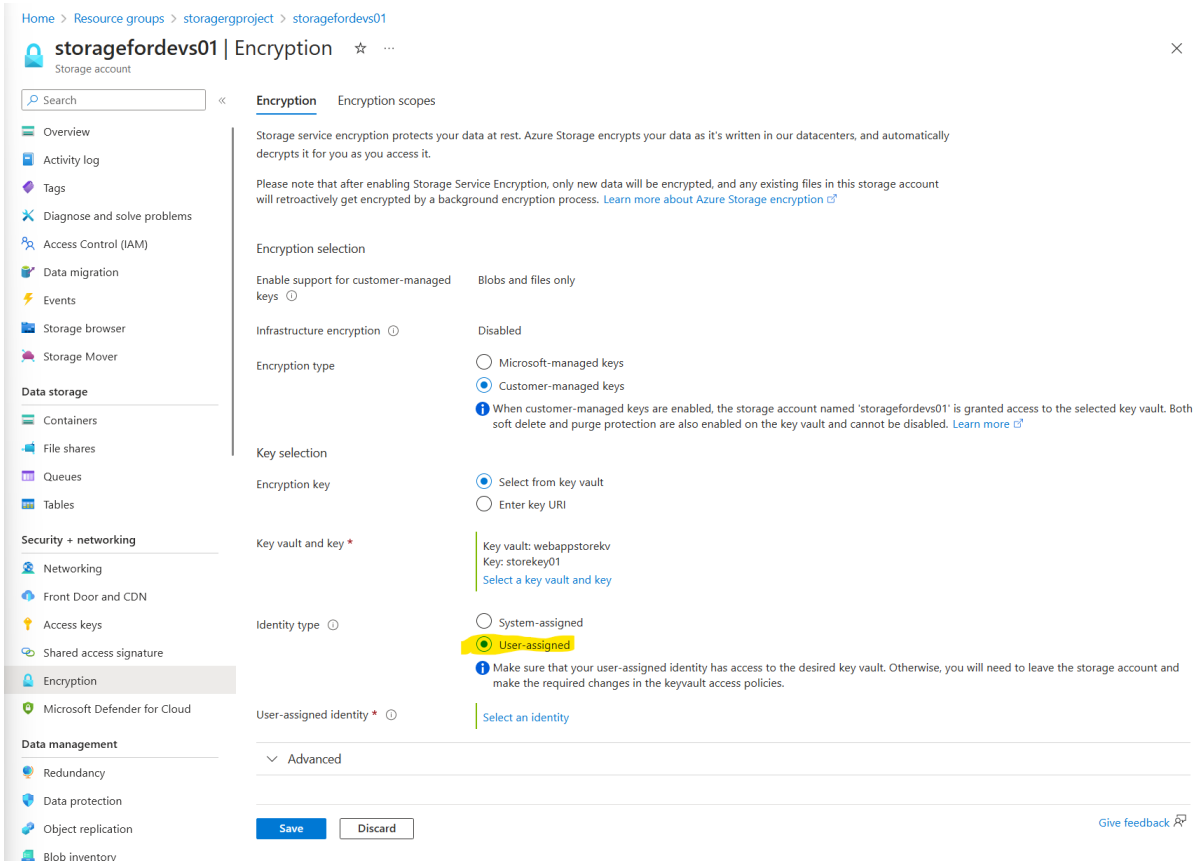


Select a key vault and key. Select your key vault and key



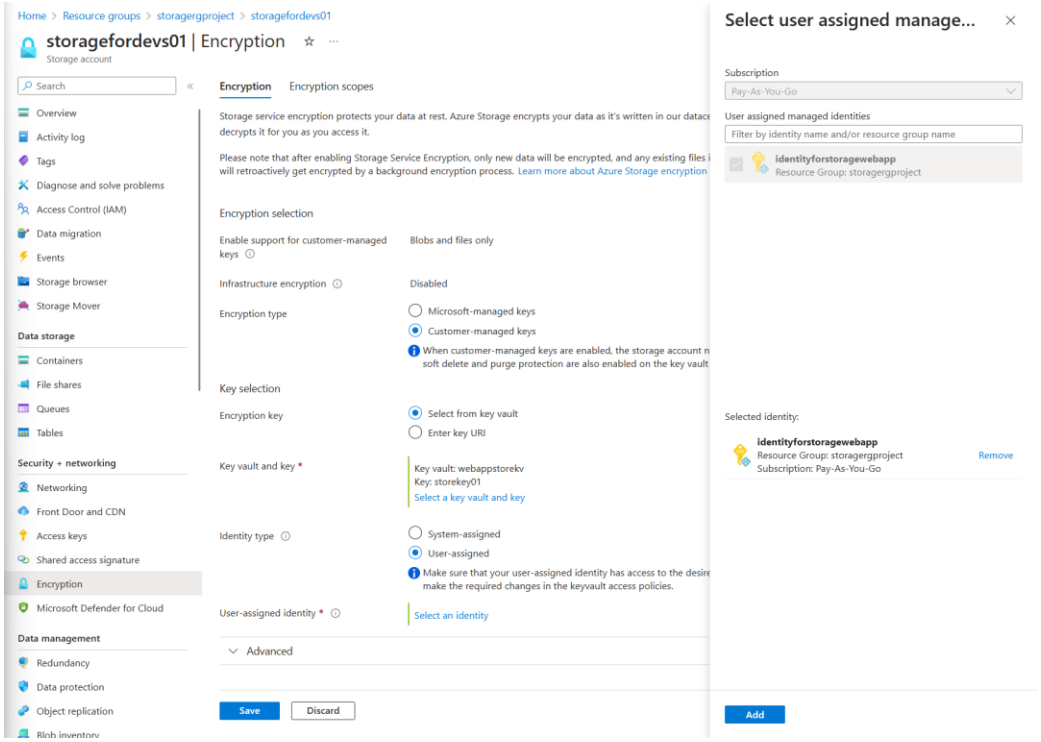
Select to confirm your choices

Ensure the **Identity type** is **User-assigned**



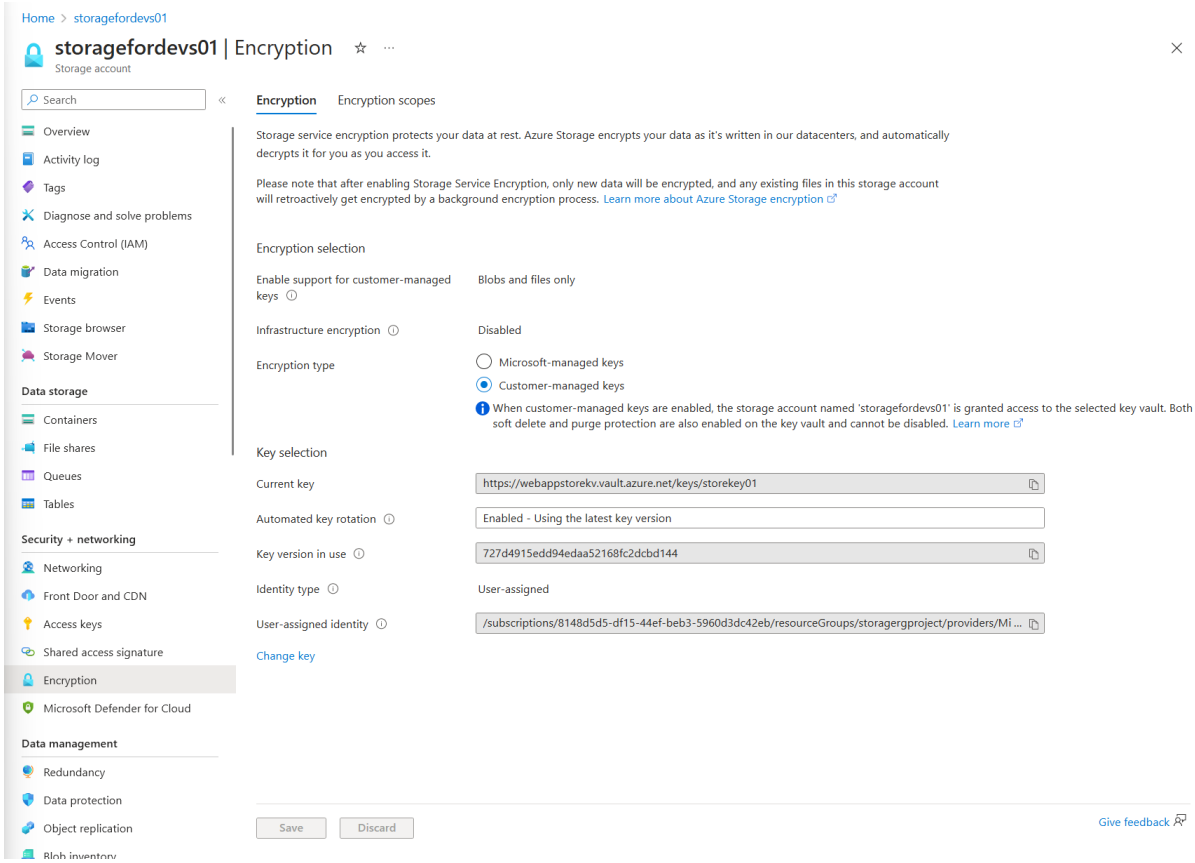
Select an identity

Select your managed identity then select **Add**



Click Save to save settings

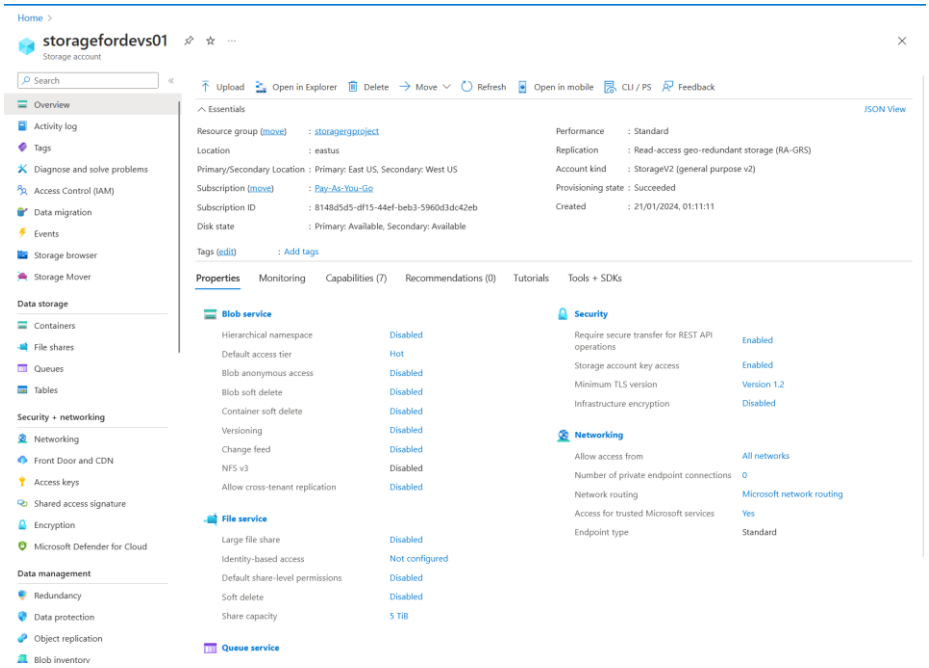
If an error occurs purge protection on key vault needs to be turned on key vault.



Step 8 - Configure a time-based retention policy and an encryption scope

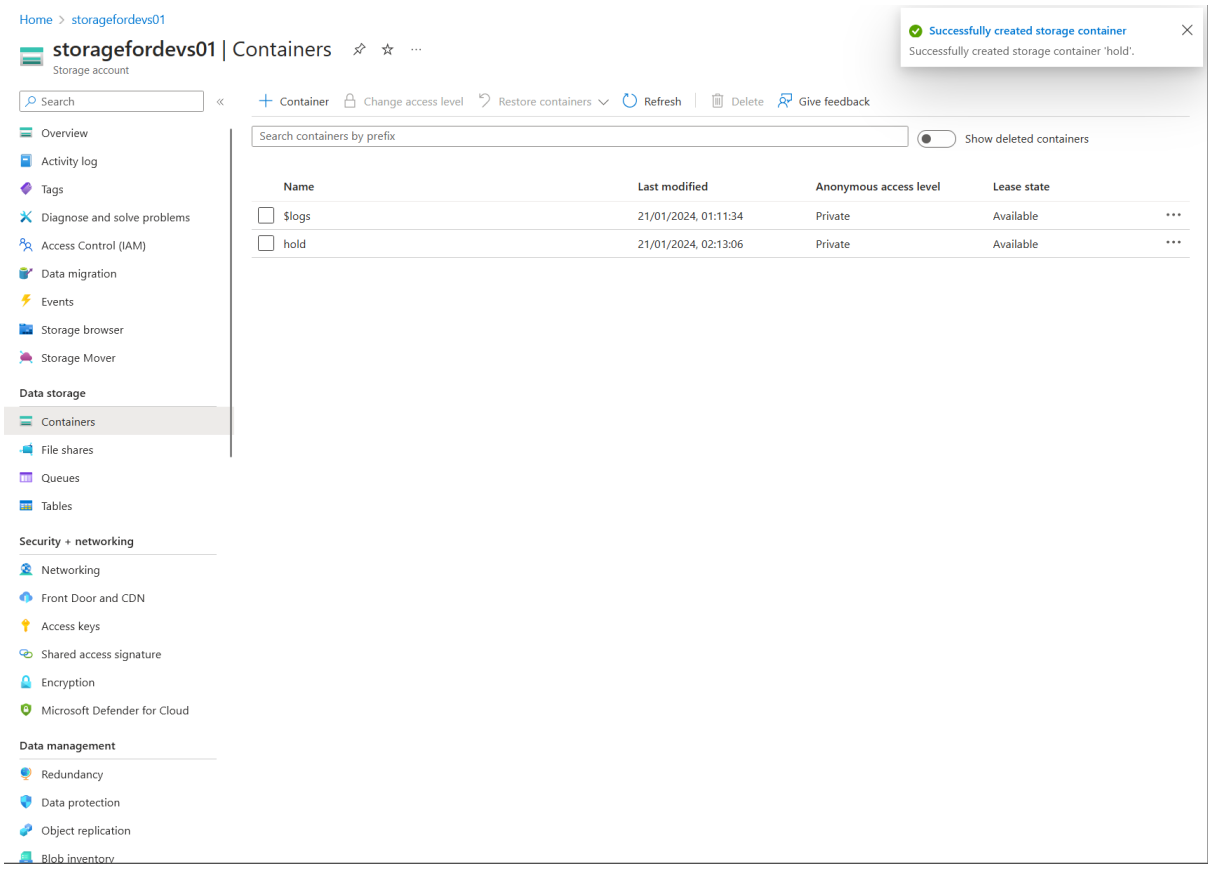
The developers require a storage container where files can't be modified, even by the administrator

Navigate to your **storage account**

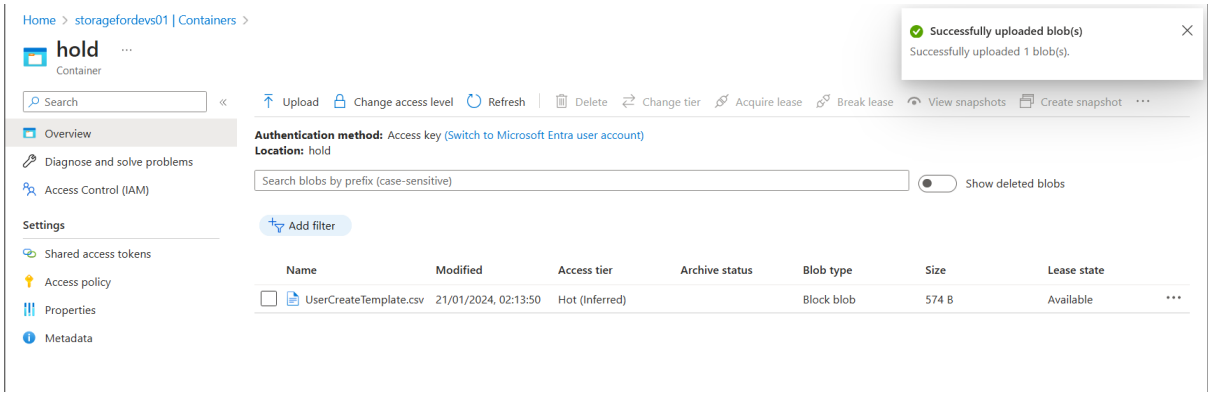


In the **Data storage** section, select the **Containers** blade

Create a container called **hold**. Take the defaults. Be sure to **Create** the container

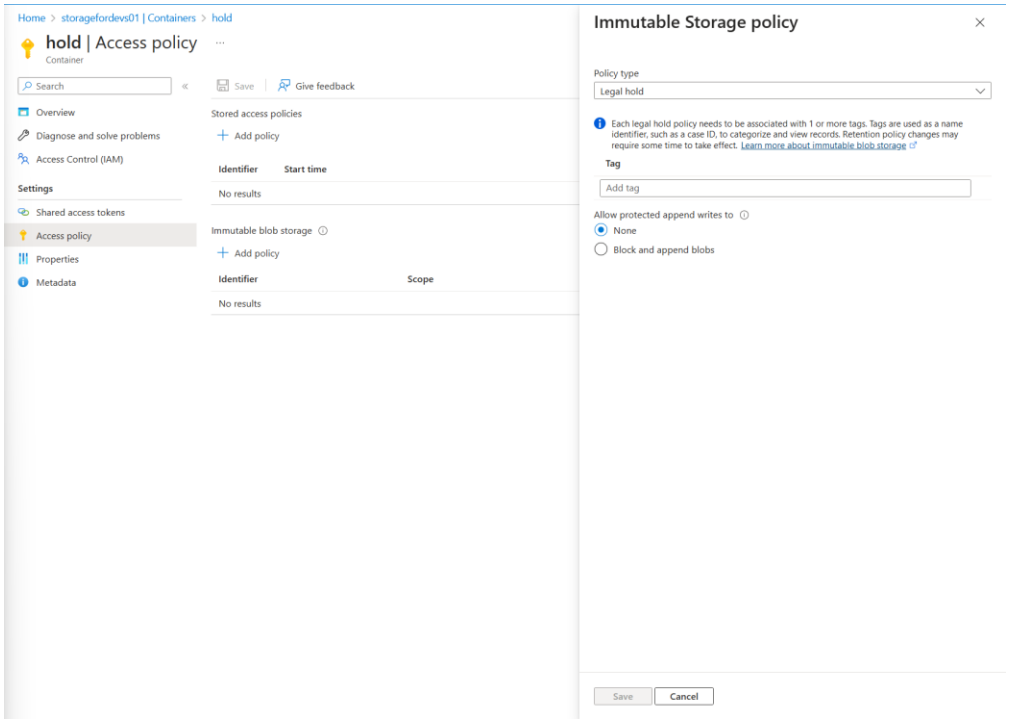


Upload a file to the container



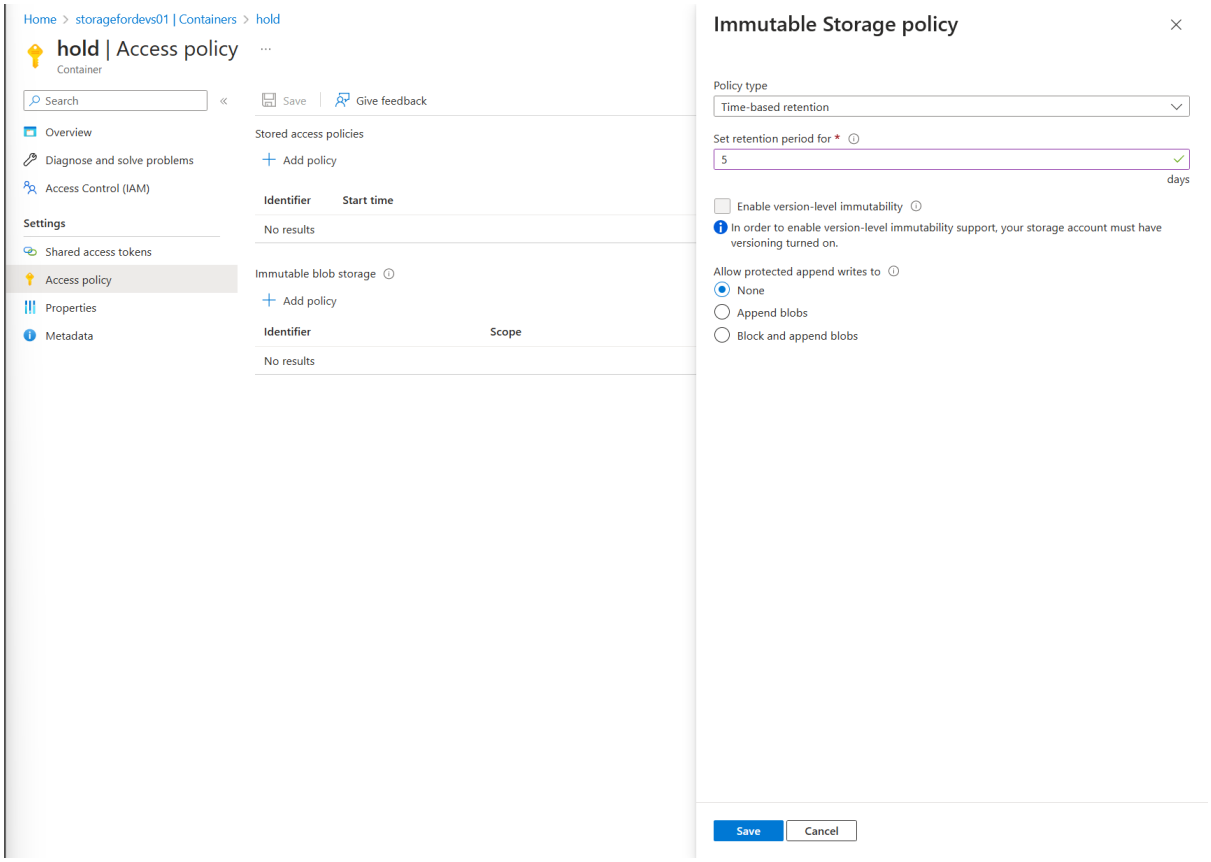
In the **Settings** section, select the **Access policy** blade

In the **Immutable blob storage** section, select **+ Add policy**

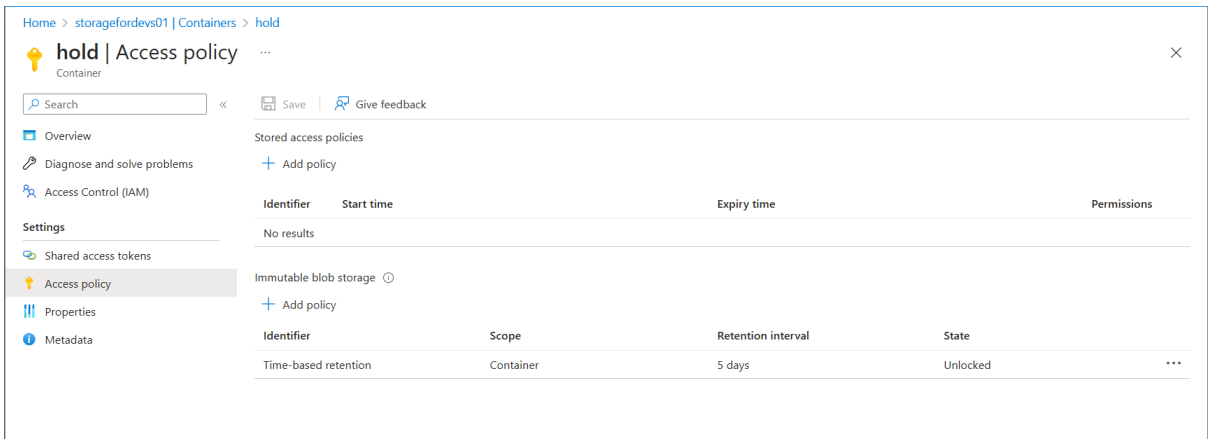


For the **Policy type**, select **time-based retention** & Set the **Retention period** to **5** days

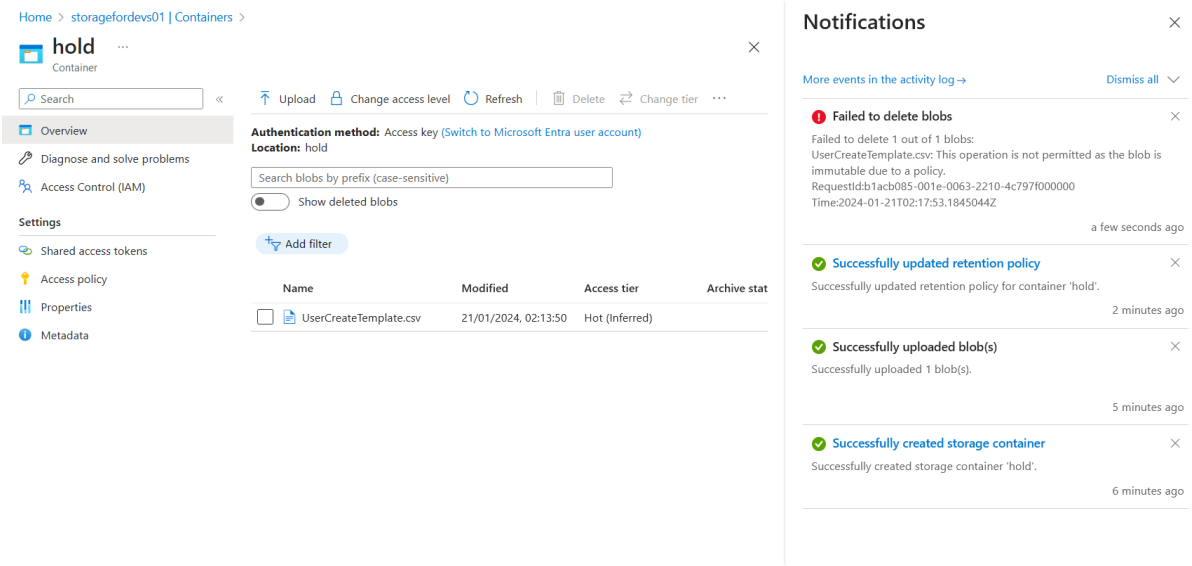
See image next



Click on save



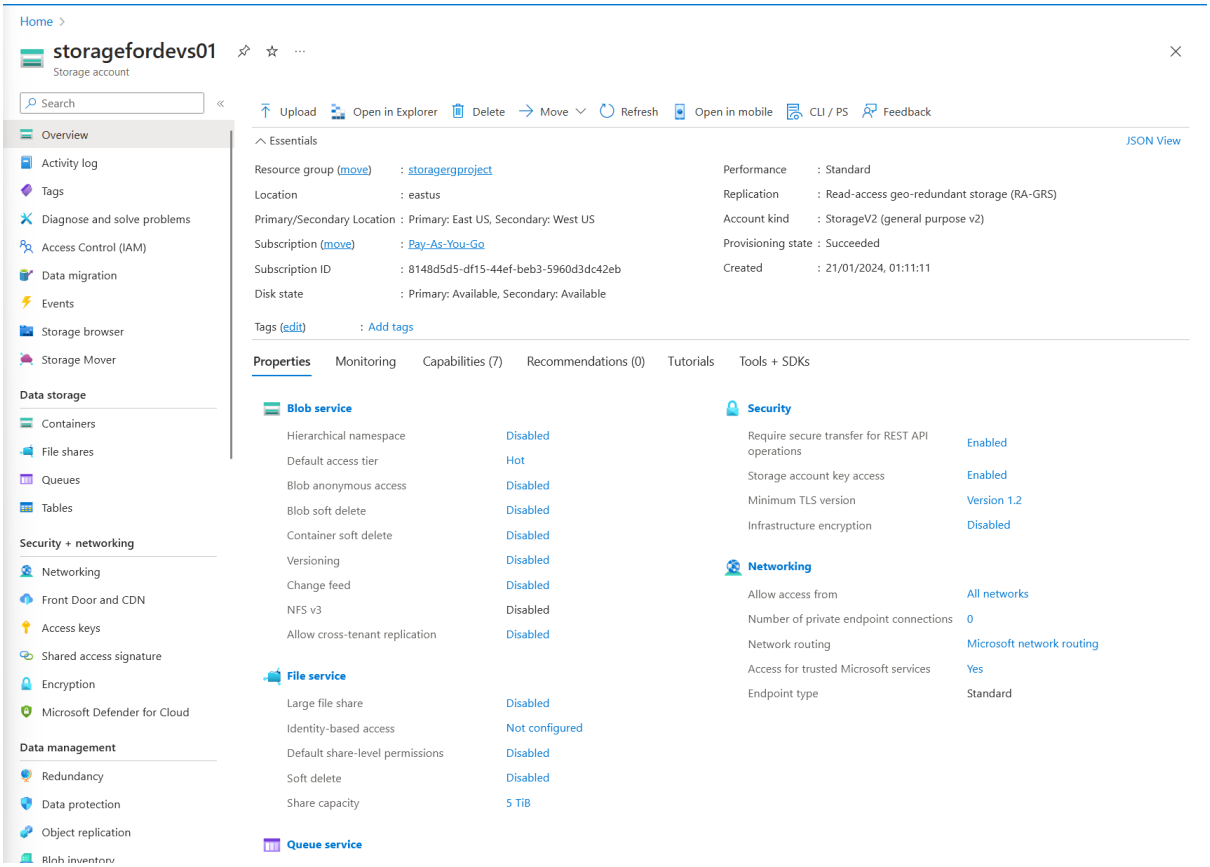
To test Try to **delete** the file in the container



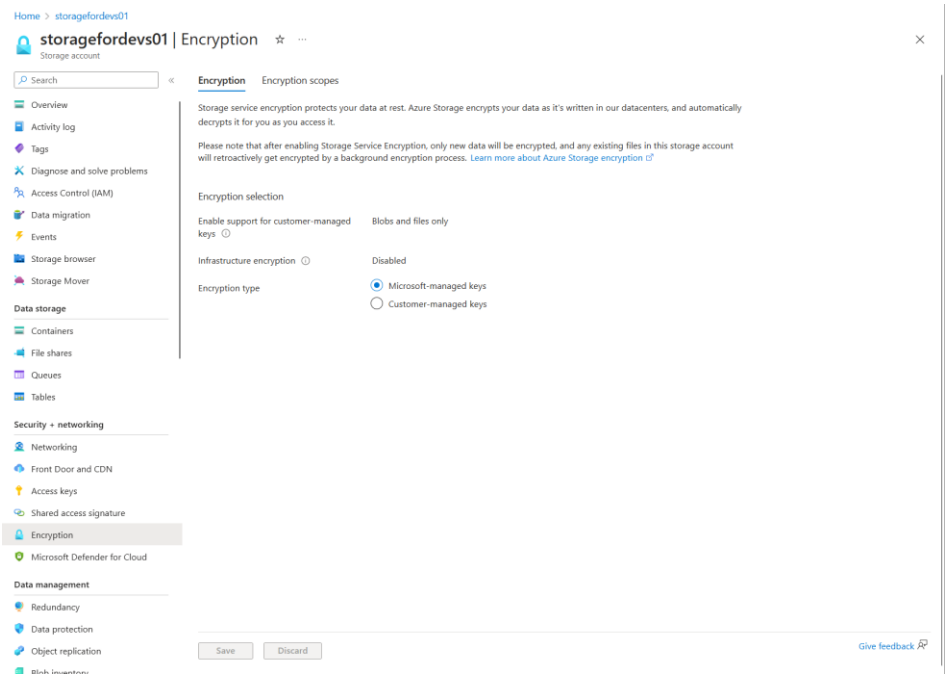
Verify you are notified **failed to delete blobs** due to policy

Step 9 - The developers require an encryption scope that enables infrastructure encryption

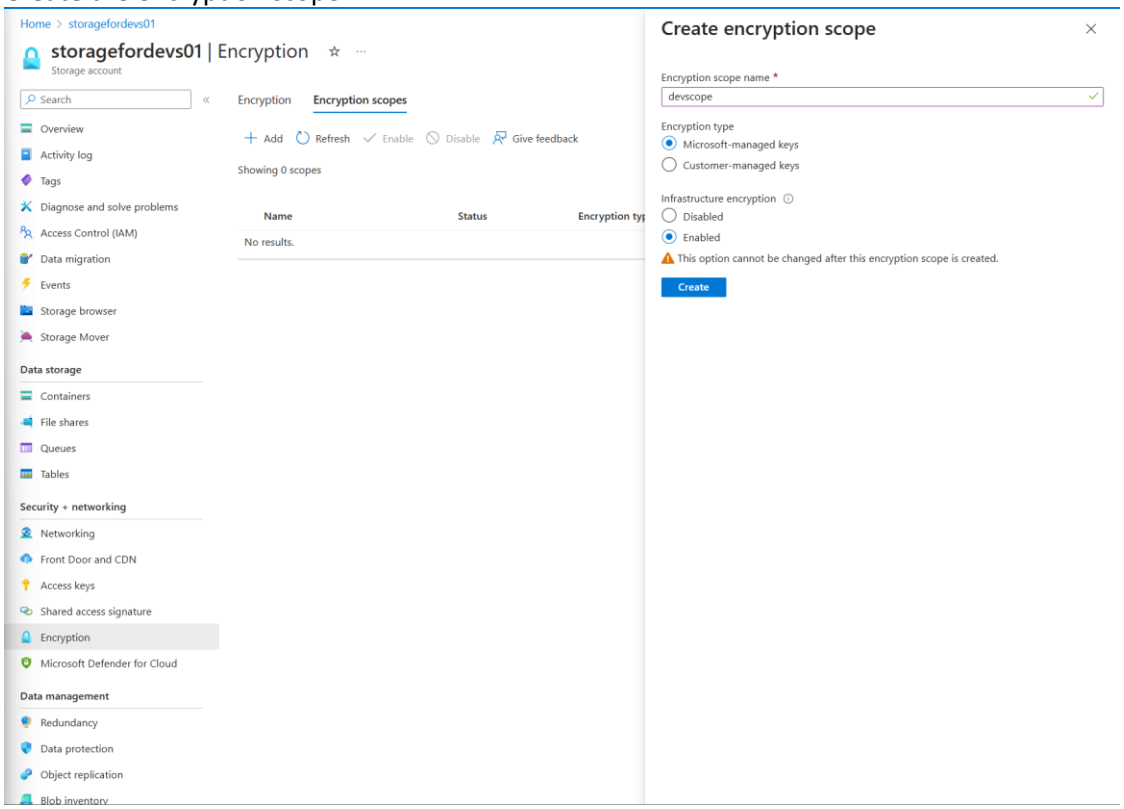
Navigate back to the storage account **storagefordevs01**



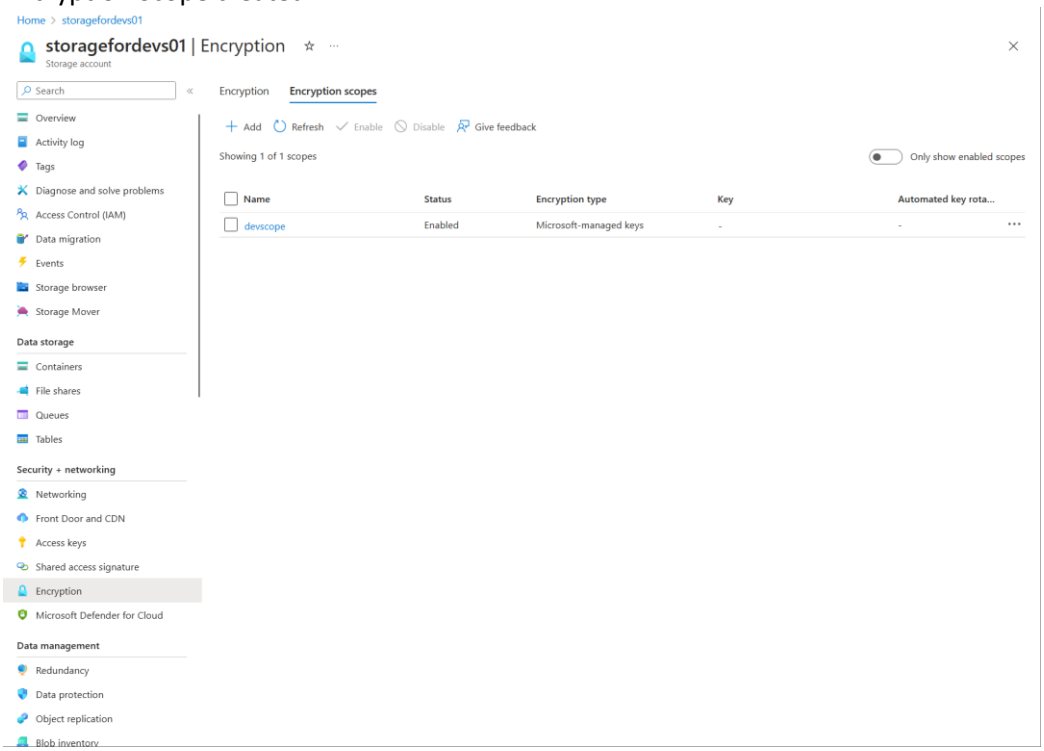
In the **Security + networking** blade, select **Encryption**



In the **Encryption scopes** tab, select **Add**
Give your encryption scope a **name**
The **Encryption type** is **Microsoft-managed key**
Set **Infrastructure encryption** to **Enable**
Notice the warning that enabling infrastructure encryption can not be changed after the scope is created
Create the encryption scope



Encryption Scope created



Return to your storage account and create a new container

Notice on the **New container** page, there is the **Name** and **Public access level**

Notice in the **Advanced** section you can select the **Encryption scope** you created and apply it to all blobs in the container