

Azure File Storage Overview

In this tutorial I'm going to show how to:

- Create a Resource Group
- Create a Storage Account
- Create a Container inside Blob Storage
- Upload Blobs to the Container
- Generate SAS from Azure Portal
- Use of Windows Azure Storage Explorer
- Generate SAS from Windows Azure Storage Explorer

Important:

This tutorial is Part 1, of a series of tutorials.

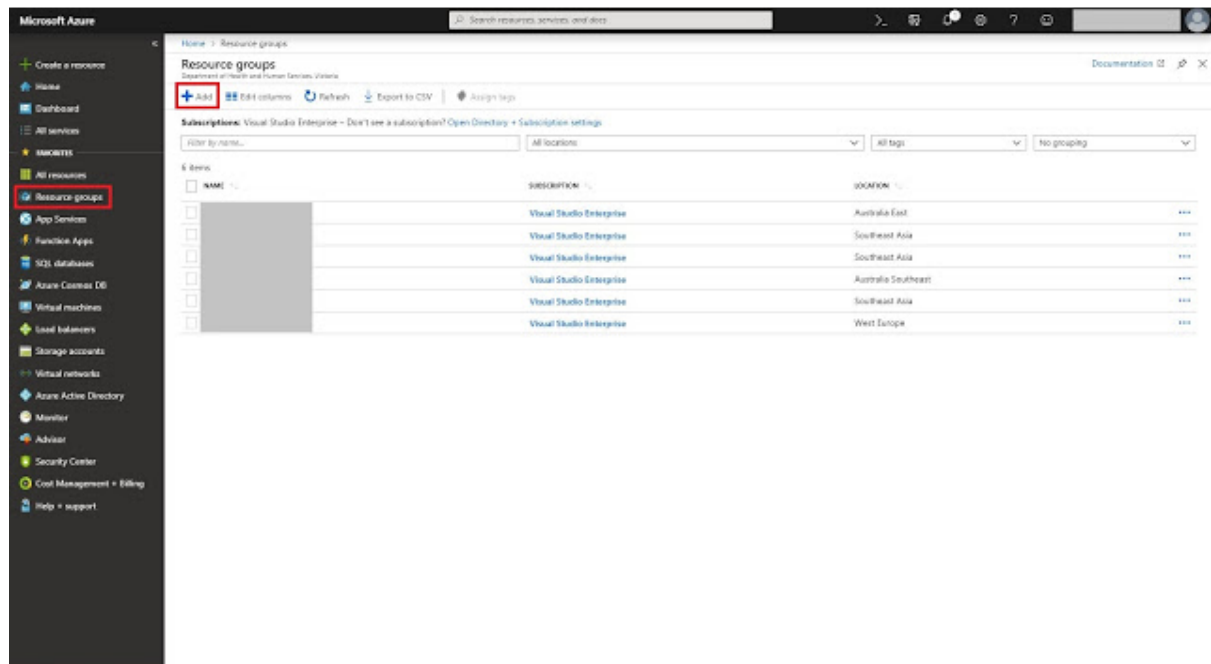
- *Part 1: Azure File Storage Overview*
- *Part 2: Azure File Storage using VB.Net*

Step 01: Create a Resource Group

As the first step, we need to create Resource Group.

If you have an existing Resource Group that you want to associate, then you can skip this step.

Click on Add button. (Menu → Resources Groups)



Give a name to the Resource Group: "rg-azurefilestorage"

Create a resource group

[Basics](#) [Tags](#) [Review + Create](#)

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#)

PROJECT DETAILS

* Subscription ⓘ Visual Studio Enterprise

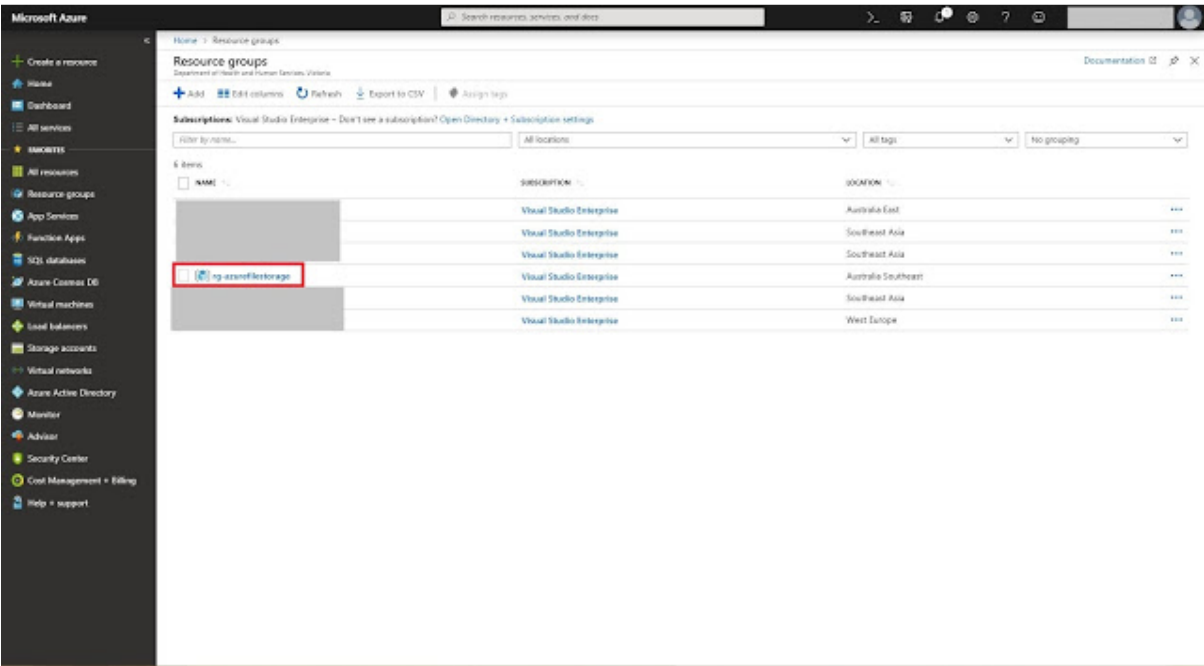
* Resource group ⓘ rg-azurefilestorage ✓

RESOURCE DETAILS

* Region ⓘ Australia Southeast

[Review + Create](#) [Next : Tags](#)

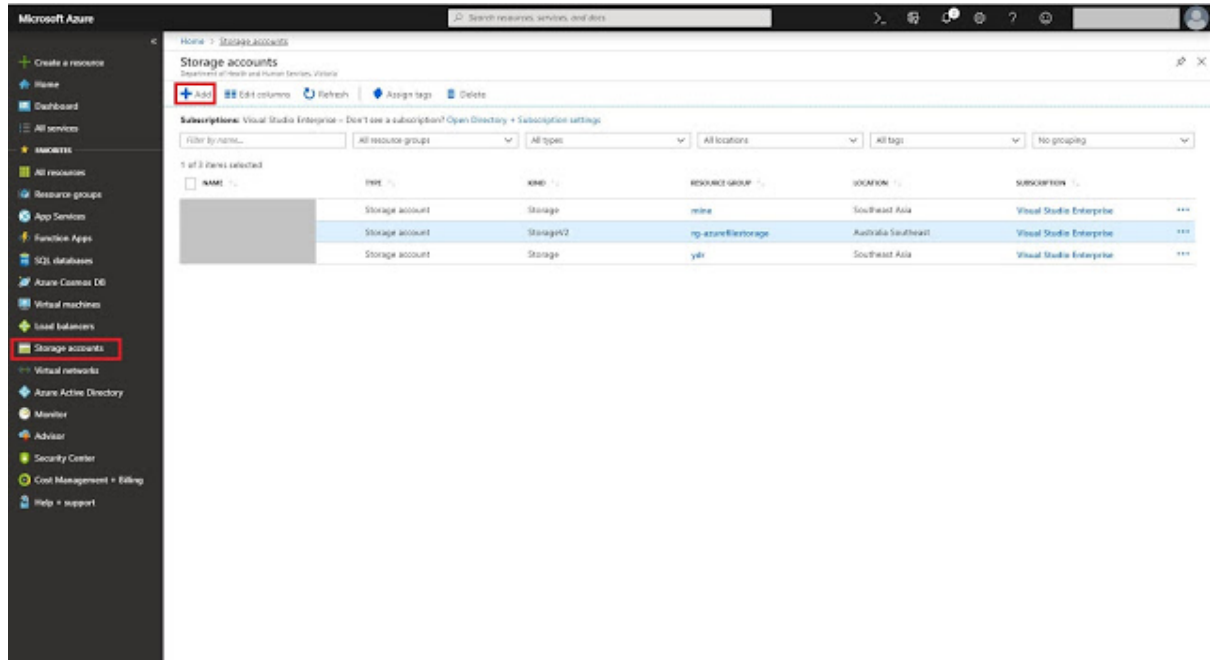
We can see the newly created Resource Group.



Step 02: Create a Storage Account

Next, we need to create a Storage Account.

Click on + Add button. (Menu → Storage Accounts)



Select the Resource Group we created earlier from the drop-down menu

Give a name to the File Storage "saazurefilestorage"

Create storage account

[Basics](#) [Advanced](#) [Tags](#) [Review + create](#)

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more](#)

PROJECT DETAILS

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

* Subscription

* Resource group [Create new](#)

INSTANCE DETAILS

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

* Storage account name ✓

* Location

Performance ☒ Standard ☐ Premium

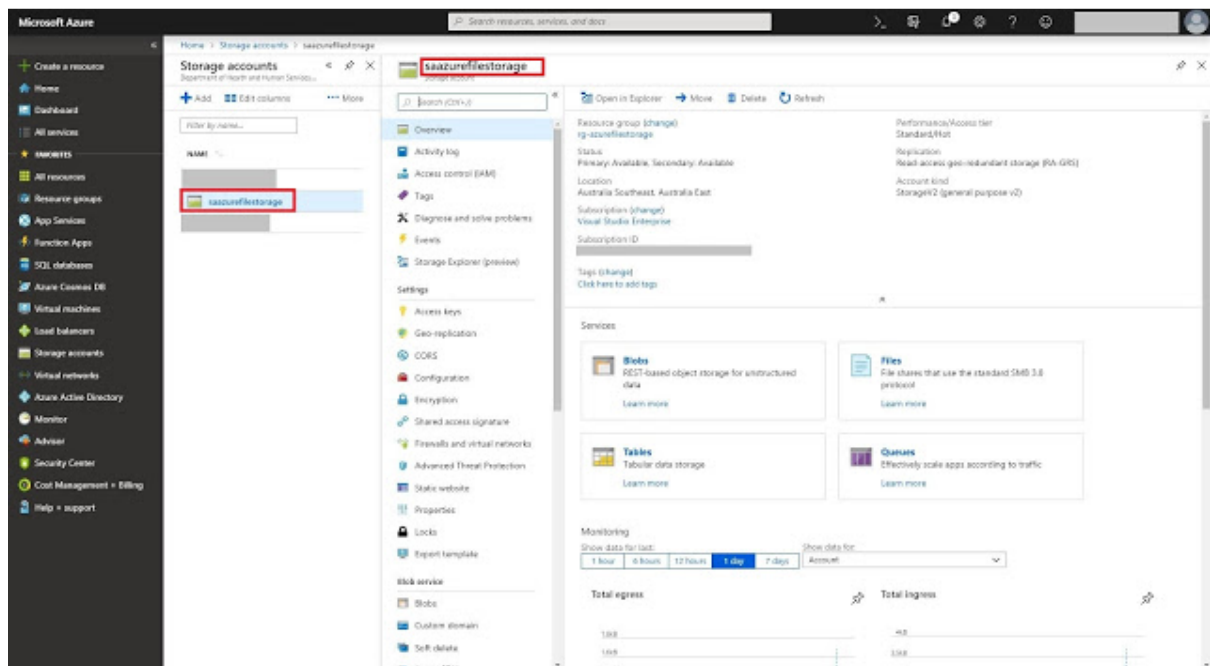
Account kind

Replication

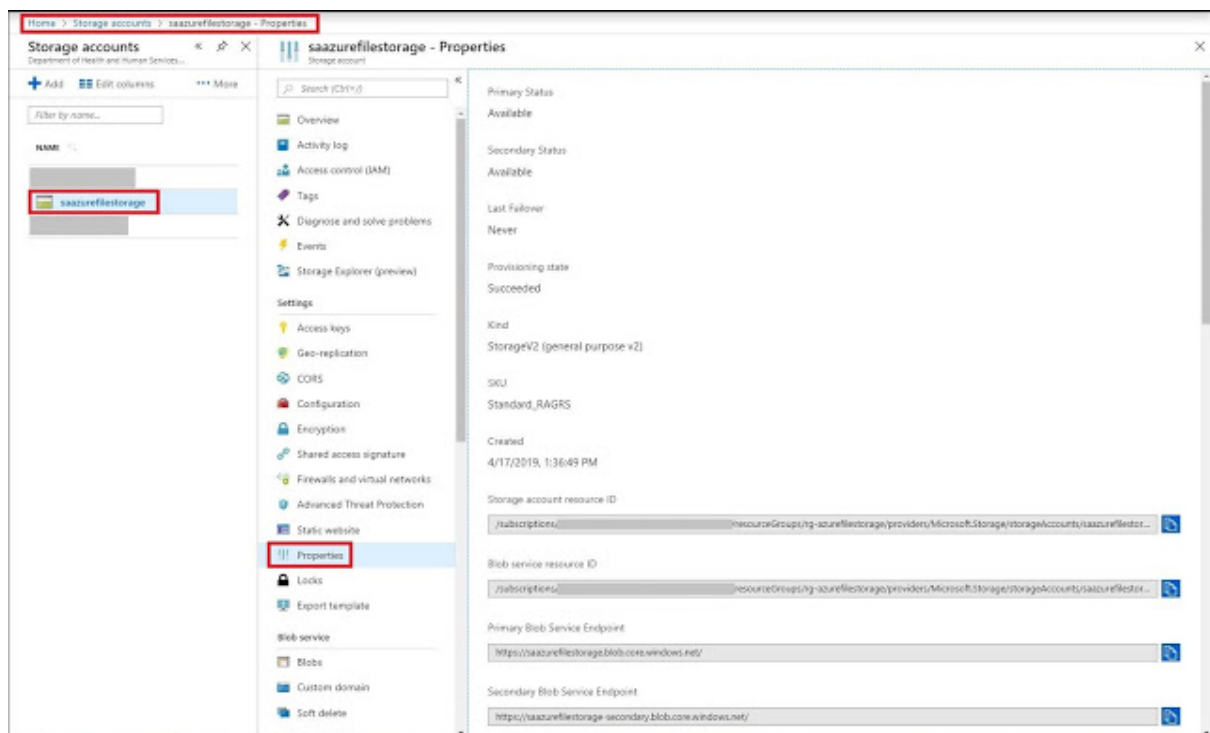
Access tier (default) ☐ Cool ☒ Hot

[Review + create](#) [Previous](#) [Next : Advanced >](#)

We can see the newly created Storage Account.

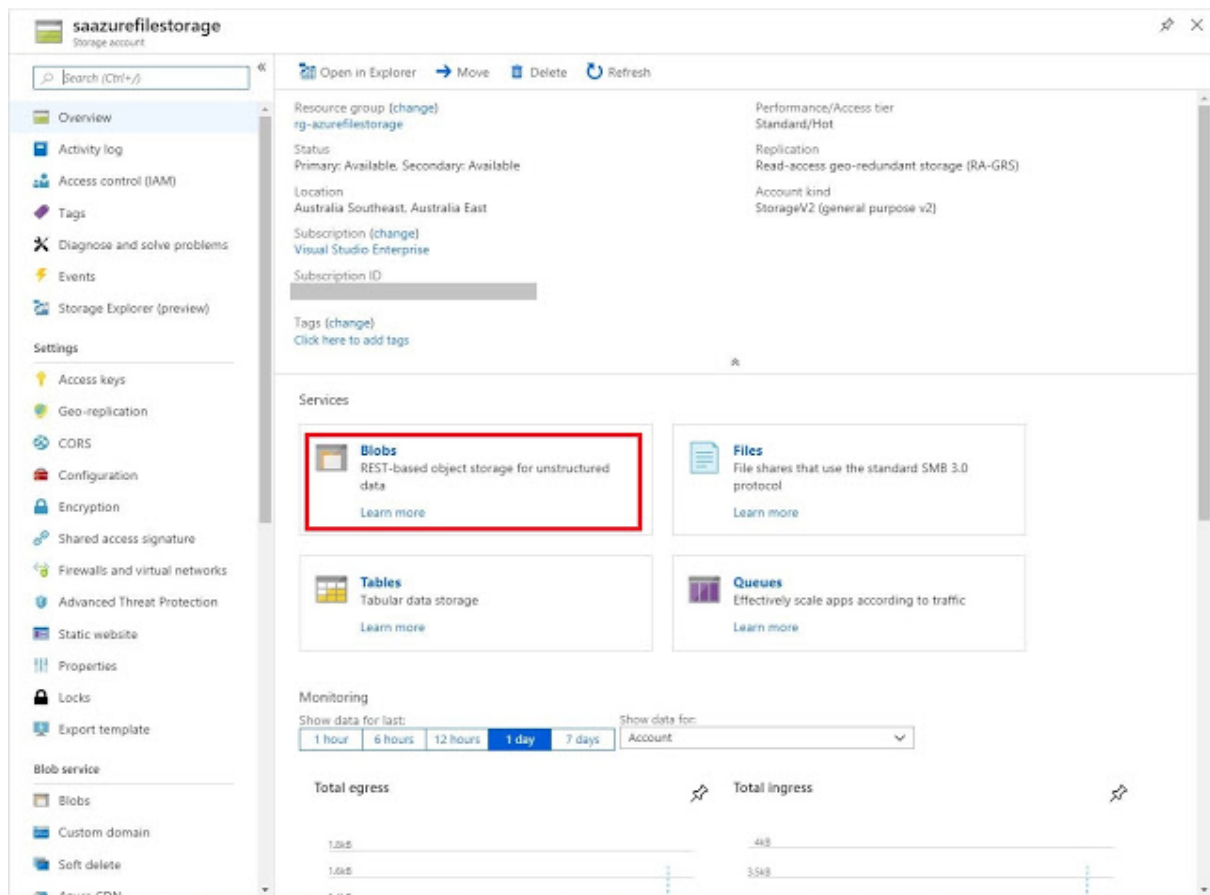


We can view the Properties for the Storage Account.

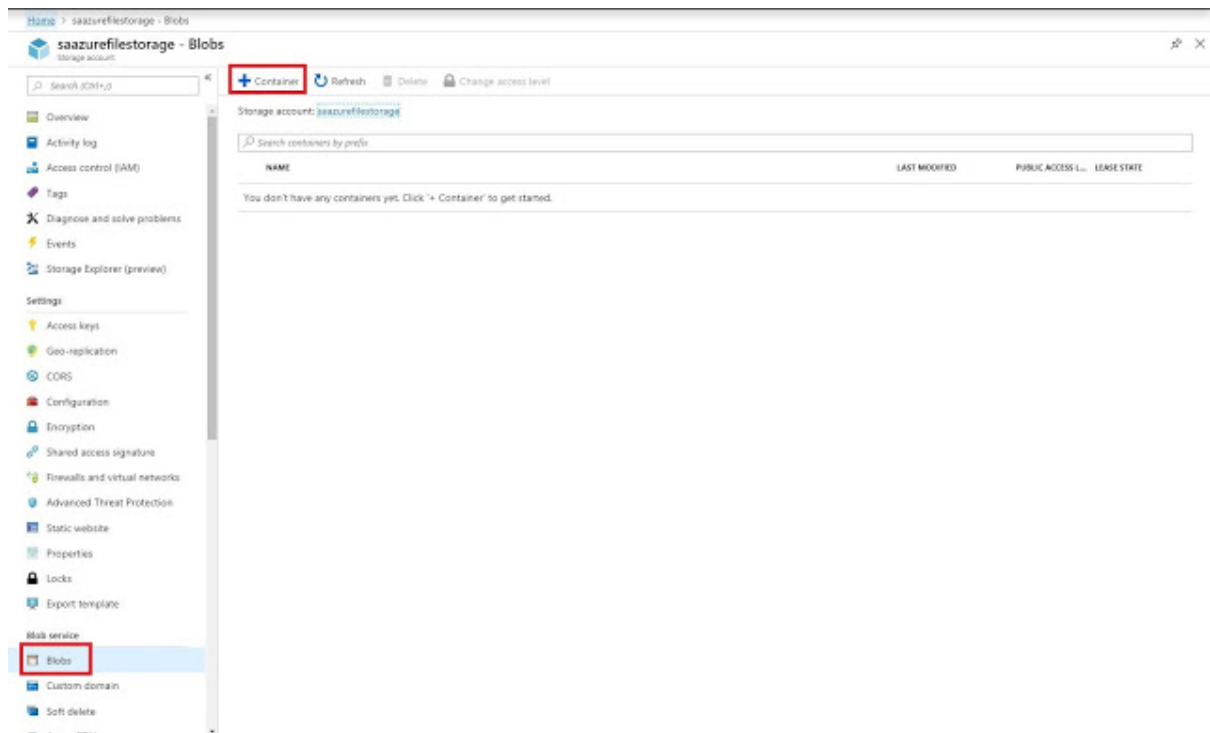


Step 03: Create a Container inside Blob Storage

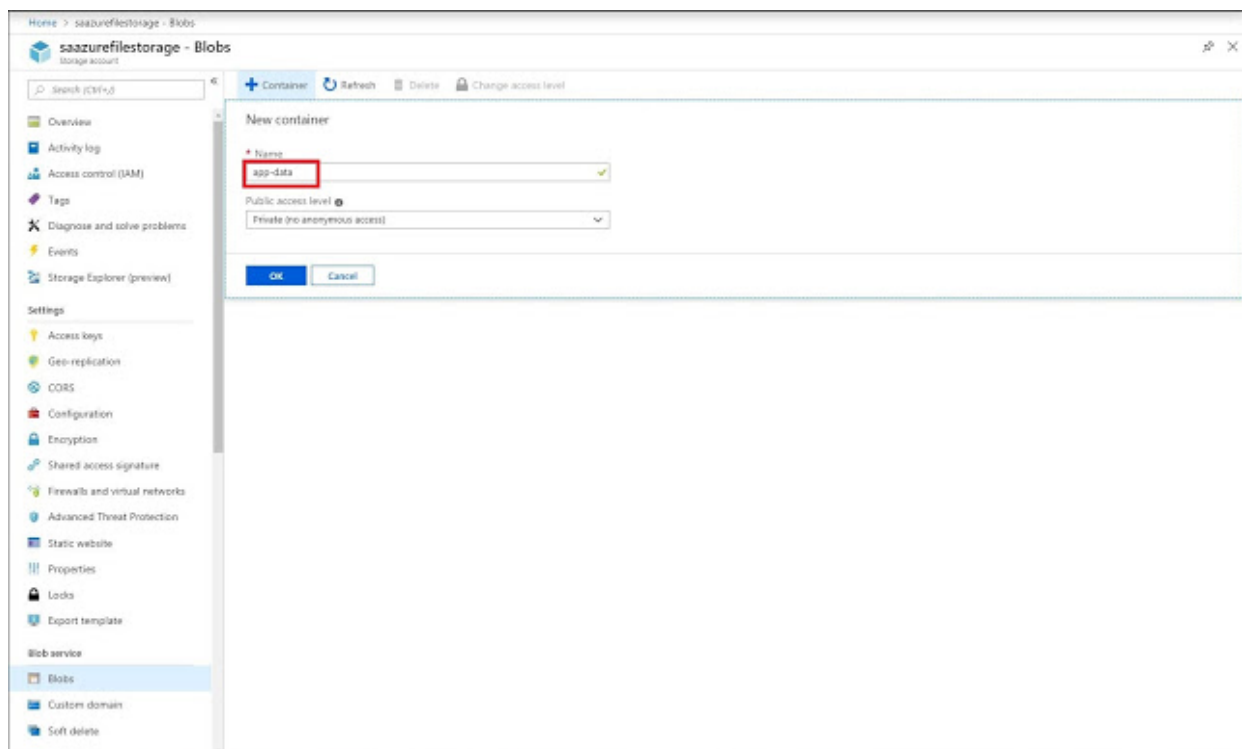
On the new Storage Account we created click on Blobs.



Click on Add Container button.



Give a name to the Container "app-data"



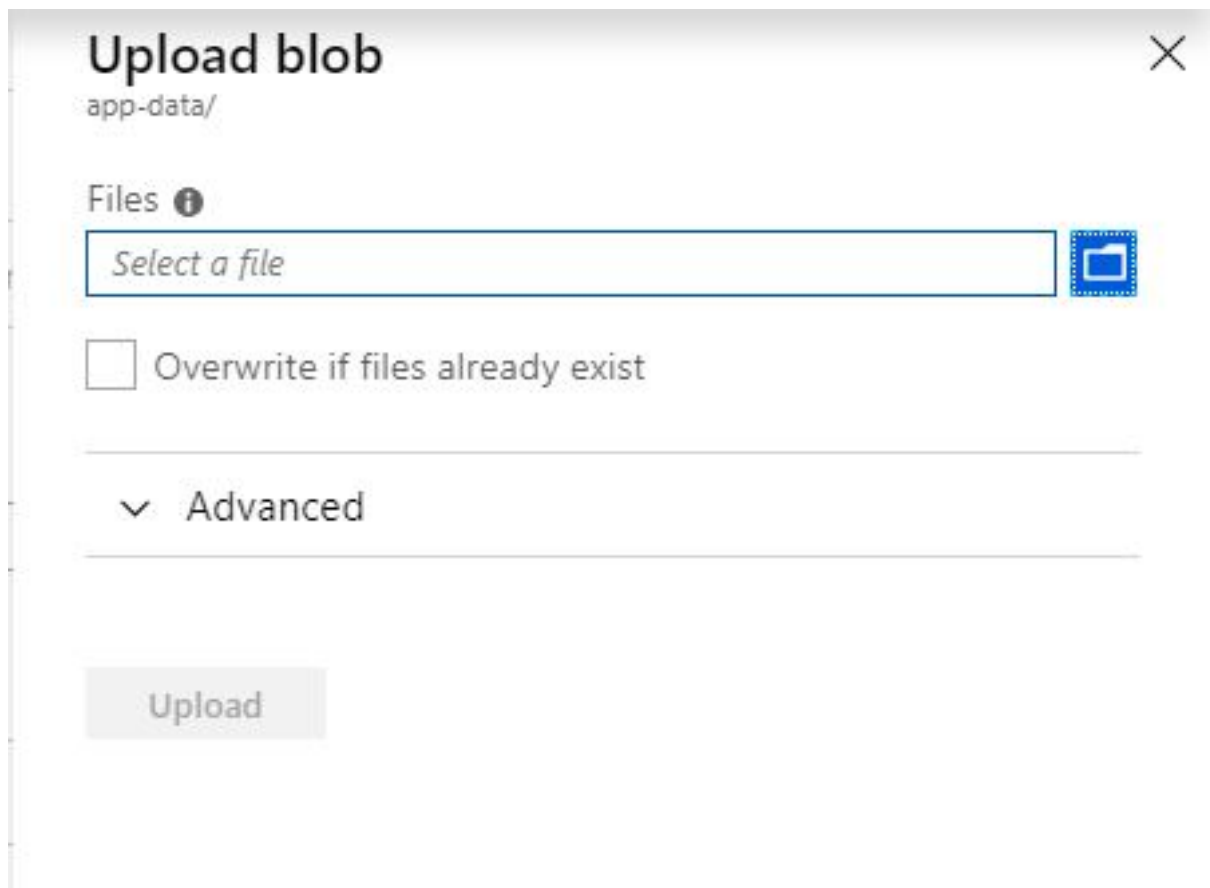
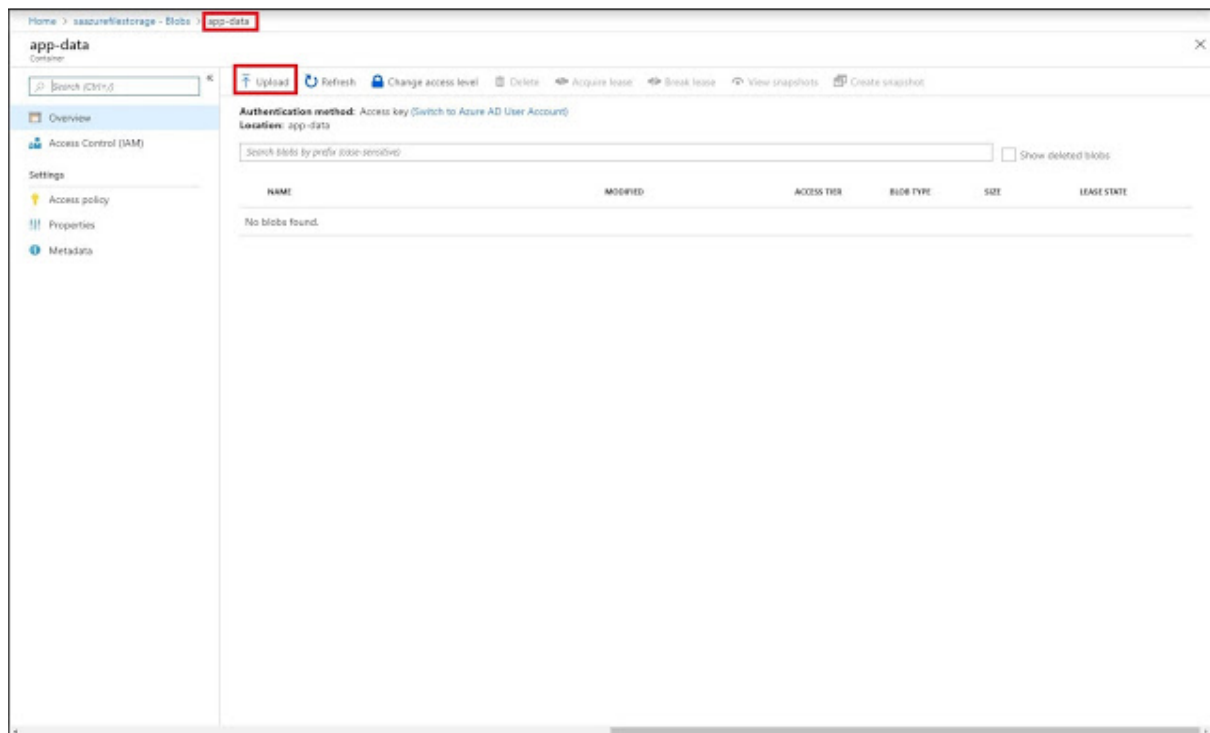
We can see the newly created Container.

The screenshot displays the Azure Storage Explorer application. The top navigation bar shows the path 'Home > saazurefilestorage - Blobs'. The left sidebar contains a list of navigation options, with 'Blobs' selected. The main pane shows the 'Storage account: saazurefilestorage' and a table of containers. A search bar at the top of the table is labeled 'Search containers by prefix'. The table has four columns: 'NAME', 'LAST MODIFIED', 'PUBLIC ACCESS L...', and 'LEASE STATE'. A single container named 'app-data' is listed, with its name highlighted by a red box. The 'LAST MODIFIED' column shows '4/17/2019, 1:49:24 PM', 'PUBLIC ACCESS L...' shows 'Private', and 'LEASE STATE' shows 'Available'.

NAME	LAST MODIFIED	PUBLIC ACCESS L...	LEASE STATE
app-data	4/17/2019, 1:49:24 PM	Private	Available

Step 04: Upload blobs to the Container

Go inside the new Container we created earlier and click on Upload button.



Upload blob

app-data/

Files ⓘ

"azure file storage example.txt"

☐ Overwrite if files already exist

Advanced

Upload

We can see the uploaded blobs.

Home > azurefilestorage - Blobs > app-data

app-data

Container

Search (Ctrl+F)

Upload Refresh Change access level Delete Acquire lease Break lease View snapshots Create snapshot

Overview

Access Control (IAM)

Settings

Access policy

Properties

Metadata

Authentication method: Access key (Switch to Azure AD User Account)

Location: app-data

Search blobs by prefix (case-sensitive)

Show deleted blobs

NAME	MODIFIED	ACCESS THIS	BLOB TYPE	SIZE	LEASE STATE	
azure file storage example.txt	4/17/2019, 1:54:09 PM	Hot (Inferred)	Block blob	27 B	Available	...

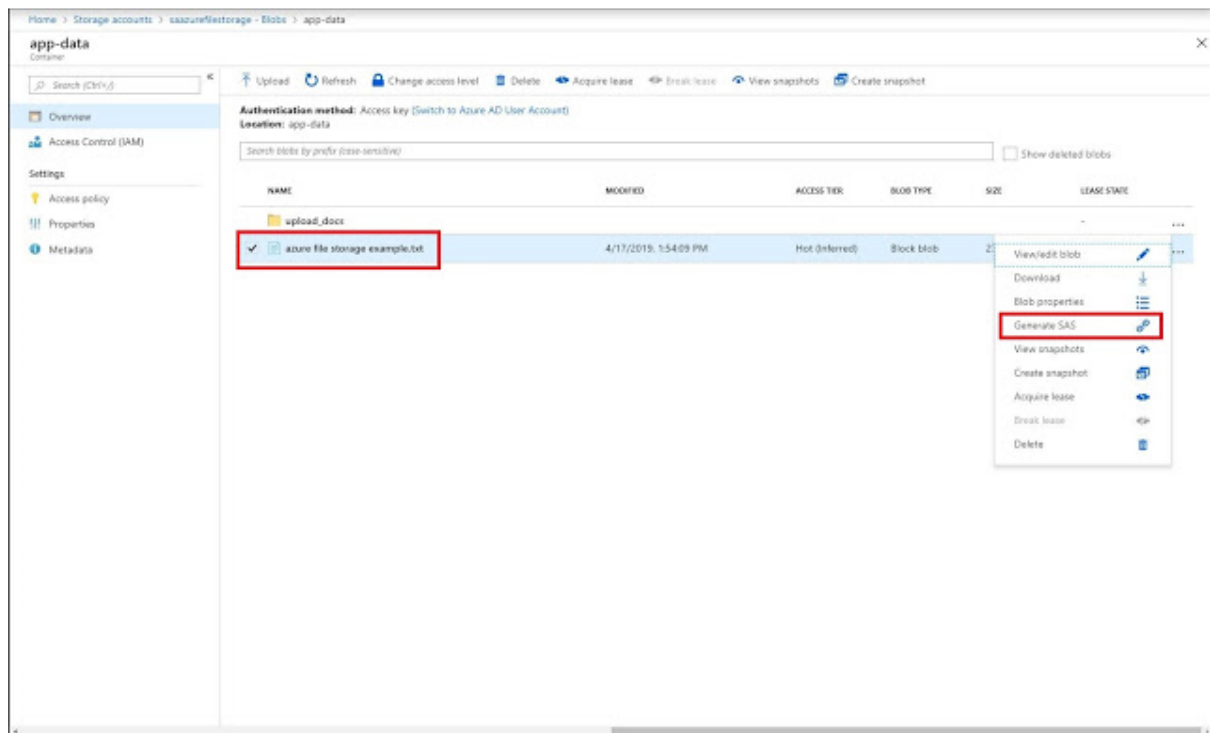
Blob properties can be viewed by clicking on "... " button.

The screenshot displays the Azure Storage portal interface for a container named 'app-data'. The left sidebar shows navigation options: Overview, Access Control (IAM), Settings, Access policy, Properties, and Metadata. The main area shows a table of blobs. A red box highlights the row for 'azure file storage example.txt'. Another red box highlights the three-dot menu button at the end of this row. The menu is open, showing options: View/edit blob, Download, Blob properties, Generate SAS, View snapshots, Create snapshot, Acquire lease, Break lease, and Delete. The 'Blob properties' option is highlighted.

NAME	MODIFIED	ACCESS TIER	BLOB TYPE	SIZE	LEASE STATE
upload_docs					
✓ azure file storage example.txt	4/17/2019, 1:54:09 PM	Hot (Inferred)	Block blob	20	

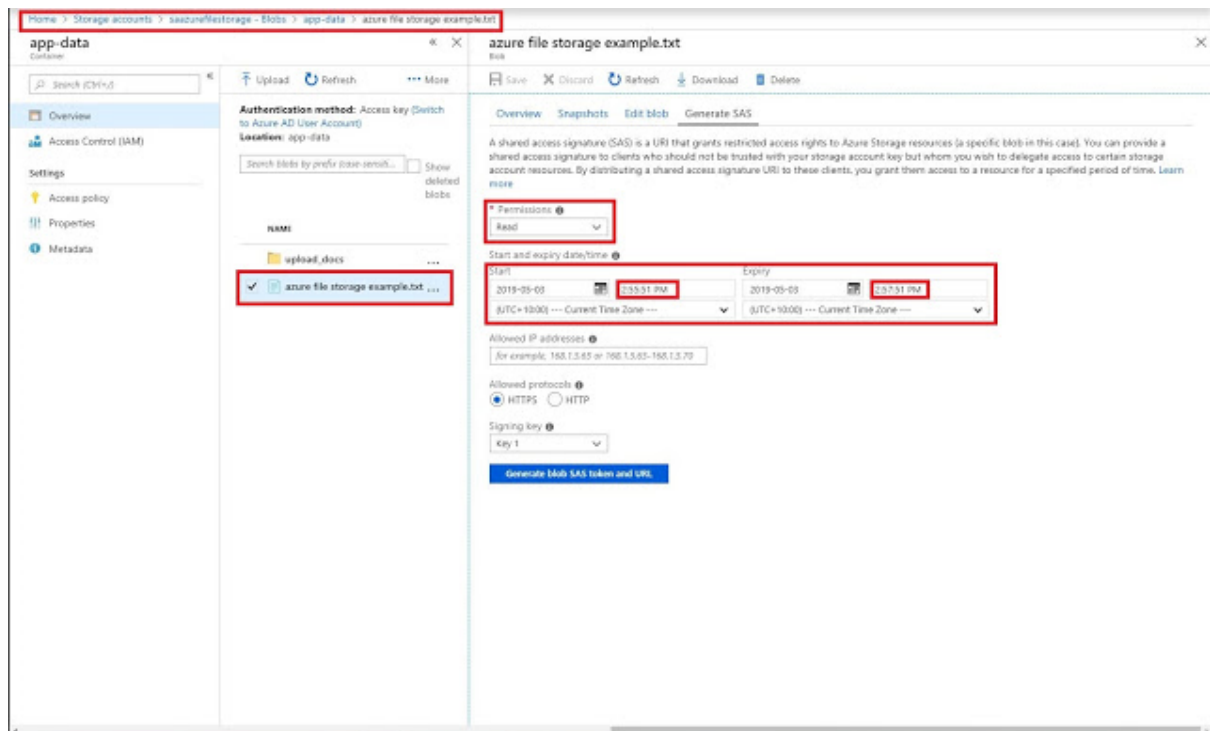
Step 05: Generate SAS from Azure Portal

We can share a file in a Blob Storage by generating a SAS.



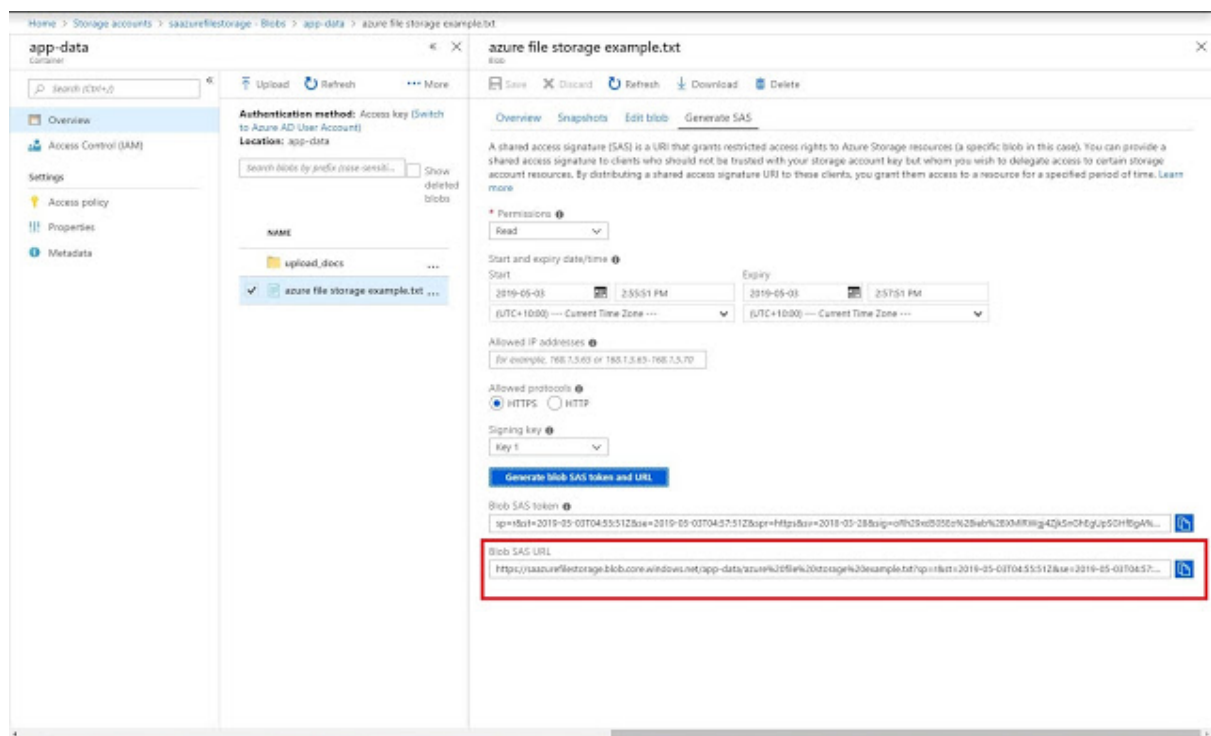
We can set below properties on SAS:

- Permission - (Read, Create, Write, Delete) What permissions we provide to the shared file URL
- Start and expiry date/time - From which date/time To which date/time shared file URL is valid
- Allowed IP addresses
- Allowed protocols - HTTPS or HTTP
- Signing key - Key 1 or Key 2



So above Shared File URL is valid:

From 2019-05-03 2:55.51 PM To 2019-05-03 2:57:51 PM



Shared URL will look something like this:

<https://saazurefilestorage.blob.core.windows.net/app-data/azure%20file%20storage%20example.txt?sp=r&st=2019-05-03T04:55:51Z&se=2019-05-03T04:57:51Z&spr=https&sv=2018-03-28&sig=oRh29xd505Eo%2Bieb%2BXMIRWqj4ZjkSnGhEqUpSGHfBgA%3D&sr=b>

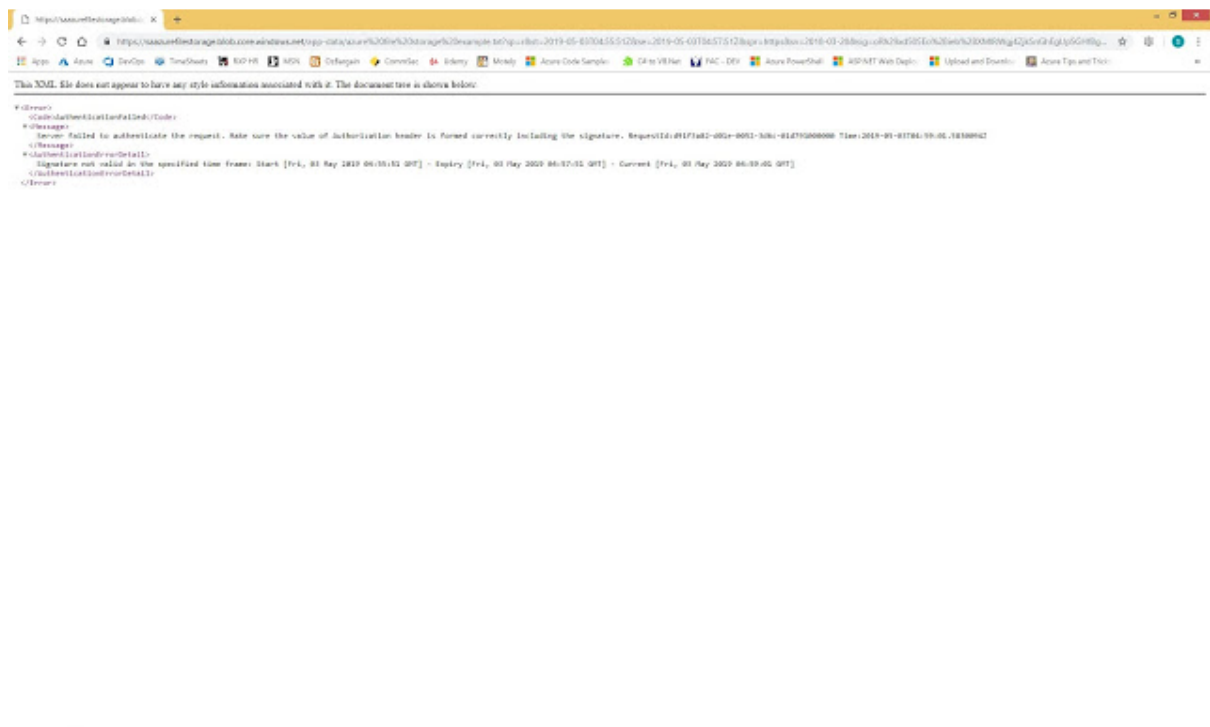
Bellow will be the token:

[sp=r&st=2019-05-03T04:55:51Z&se=2019-05-03T04:57:51Z&spr=https&sv=2018-03-28&sig=oRh29xd505Eo%2Bieb%2BXMIRWqj4ZjkSnGhEqUpSGHfBgA%3D&sr=b](#)

We can view the file content by pasting URL to a browser:

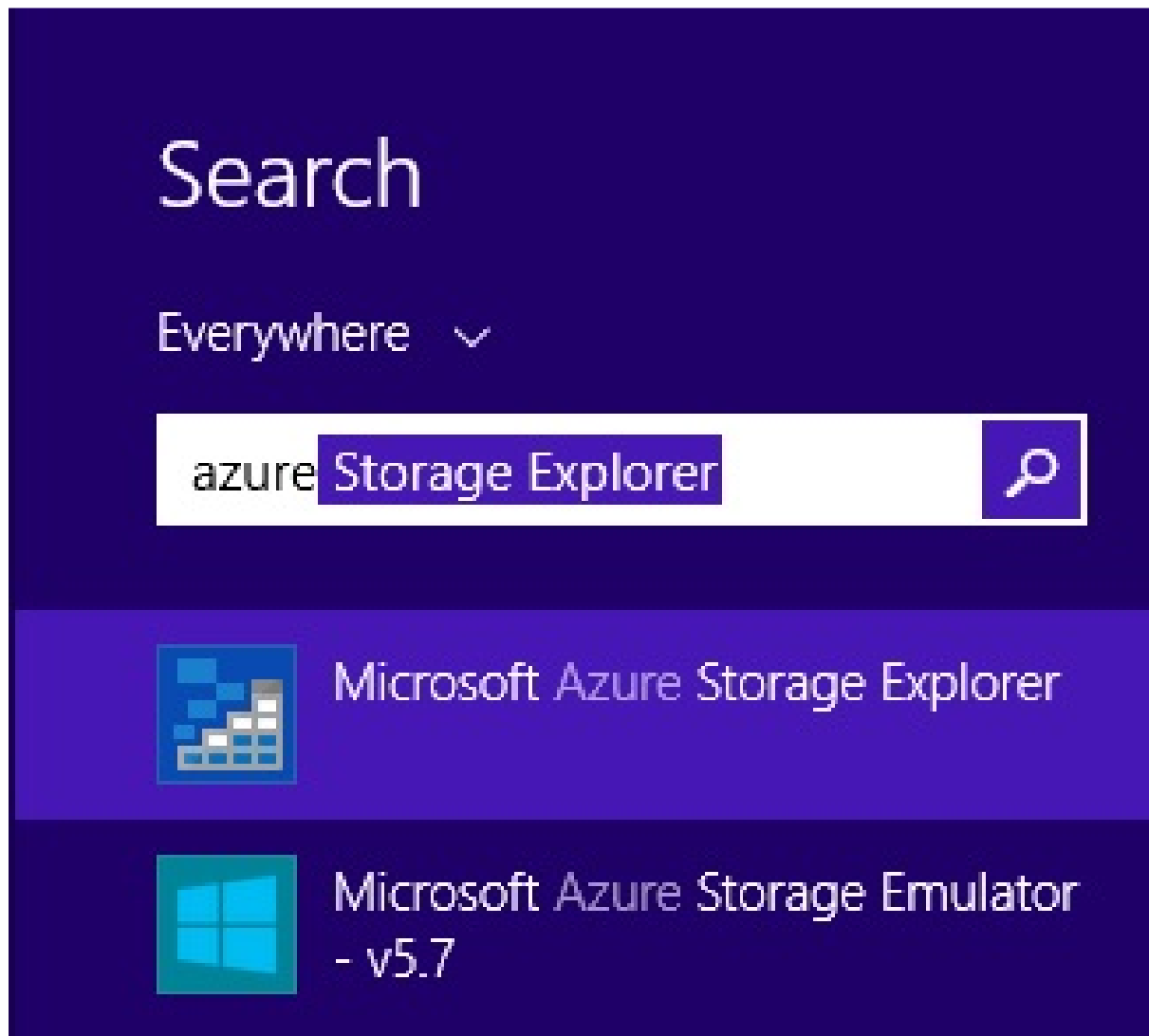


If we try to view that file outside the specified valid time frame we will be prompted with bellow error.

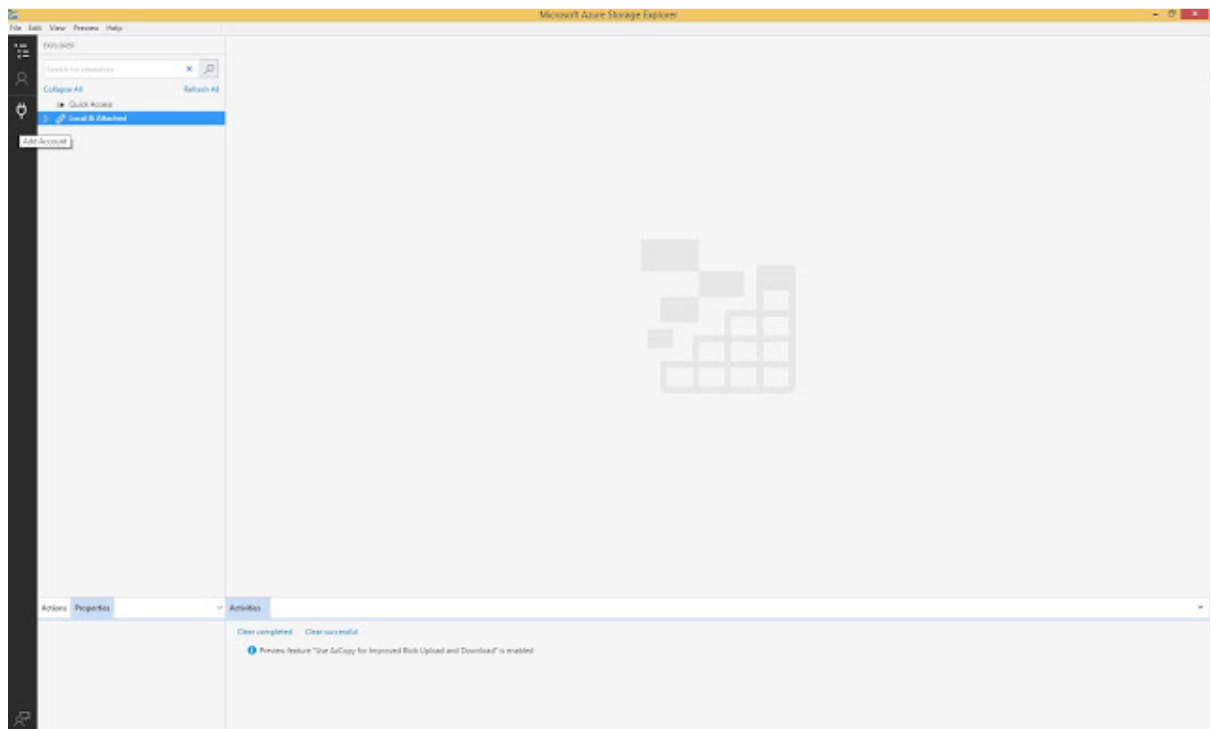


Step 06: Use of Windows Azure Storage Explorer

Got to Windows and type Azure Storage Explorer on the Windows search.



To add an Storage Account click on Add Account.



Select the option "Use a storage account name and key"

Microsoft Azure Storage Explorer - Connect

Connect to Azure Storage

How do you want to connect to your Storage Account or service?

☐ Add an Azure Account

Azure environment:

Azure

☐ Use a connection string

☐ Use a shared access signature (SAS) URI

☒ Use a storage account name and key


☐ Attach to a local emulator

Back Next Connect Cancel

You can get the storage account name and key for the previously created Storage Account through Azure Portal.

Go to Access keys under Storage Account.

The screenshot displays the Azure Portal interface for a storage account named 'saazurefilestorage'. The breadcrumb navigation at the top indicates the path: Home > Storage accounts > saazurefilestorage > Access keys. The left-hand navigation pane shows the 'Access keys' option highlighted under the 'Settings' section. The main content area provides instructions on using access keys and lists the account details. The 'Storage account name' is 'saazurefilestorage'. Two access keys are shown: 'key1' and 'key2', each with its corresponding 'Key' and 'Connection string'. The 'key1' connection string is partially visible as 'DefaultEndpointsProtocol=http;AccountName=saazurefilestorage;AccountKey=...'. The 'key2' connection string is also partially visible as 'DefaultEndpointsProtocol=http;AccountName=saazurefilestorage;AccountKey=...'. Red boxes highlight the 'Access keys' menu item, the 'Storage account name' field, and the 'key1' label.

 Microsoft Azure Storage Explorer - Connect

Connect with Name and Key

Display name:

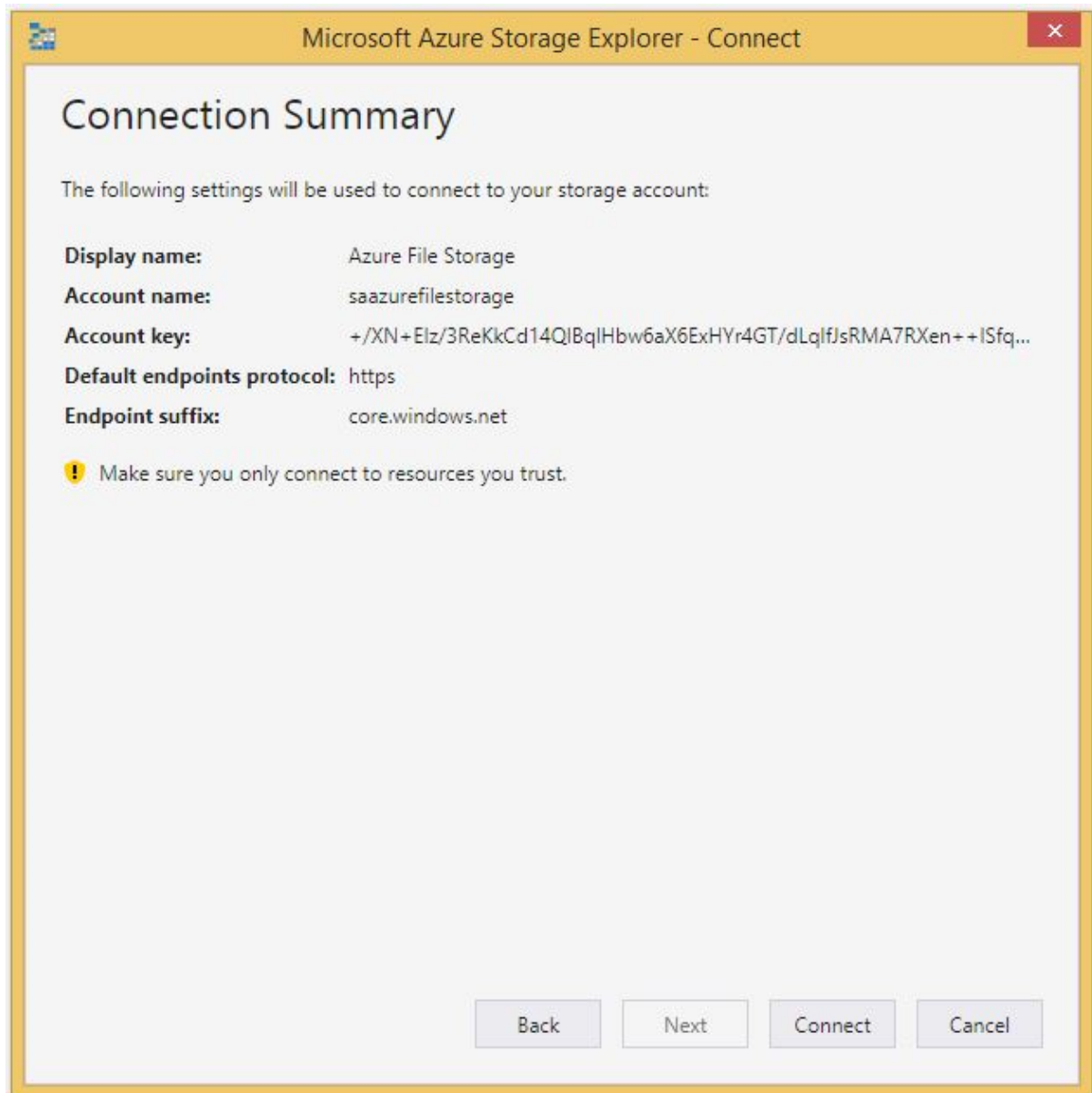
Account name:

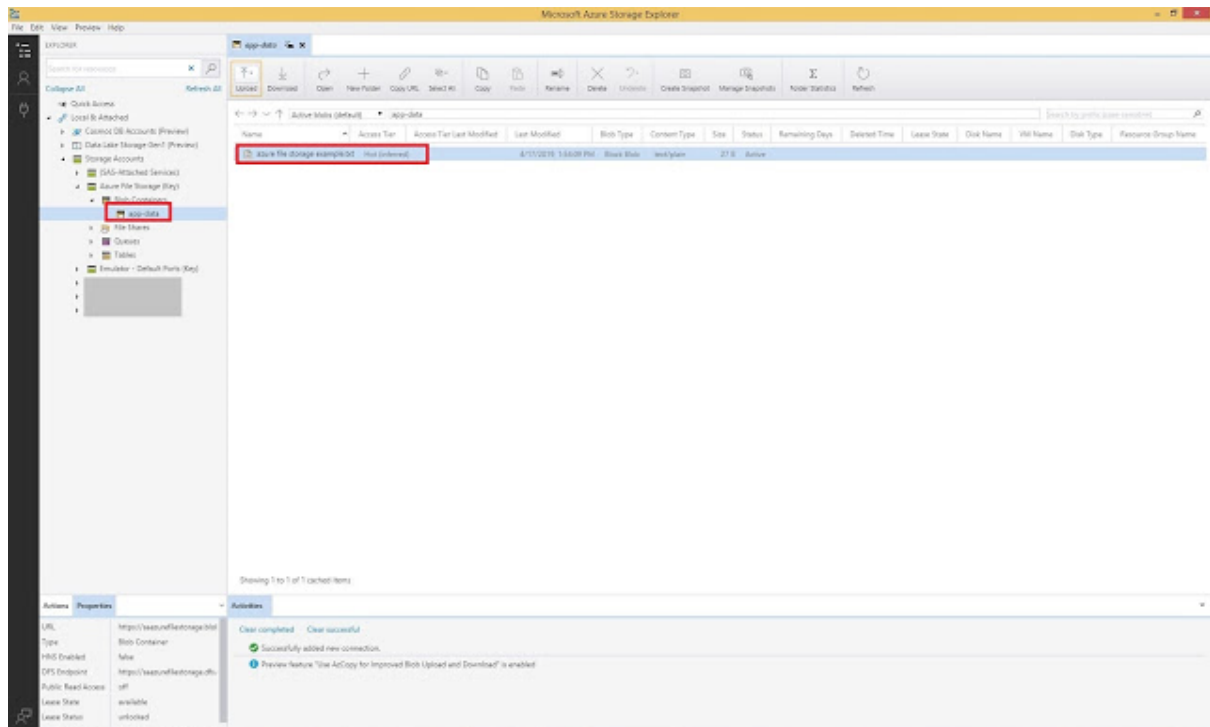
Account key:

Storage domain:

☐ Use HTTP (not recommended)

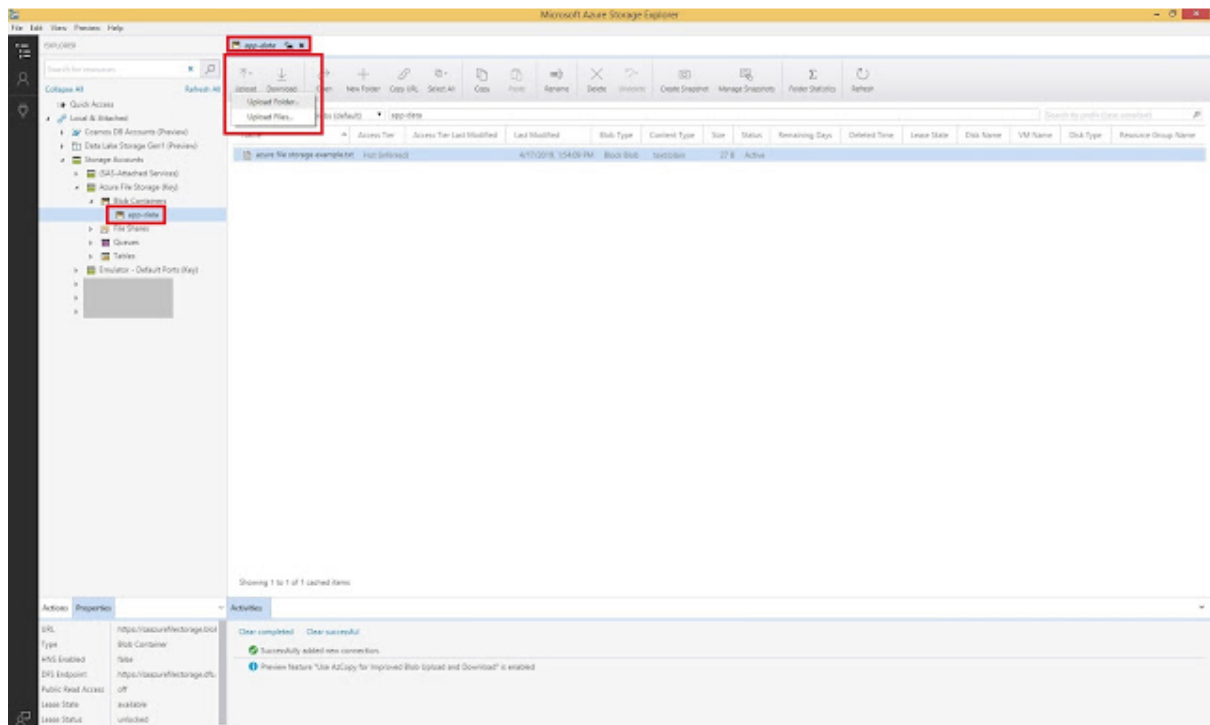
[Online privacy statement](#)






Also, you can upload folders through Windows Azure Storage Explorer. (folder upload is not available through portal at the moment)

Select Upload Folder.



 Microsoft Azure Storage Explorer - Upload to Blob Container ✕

Upload Folder

Select the folder you want to upload, the type of blob you want to upload as, and the destination directory.

Selected folder:

upload_docs ...

Blob Type

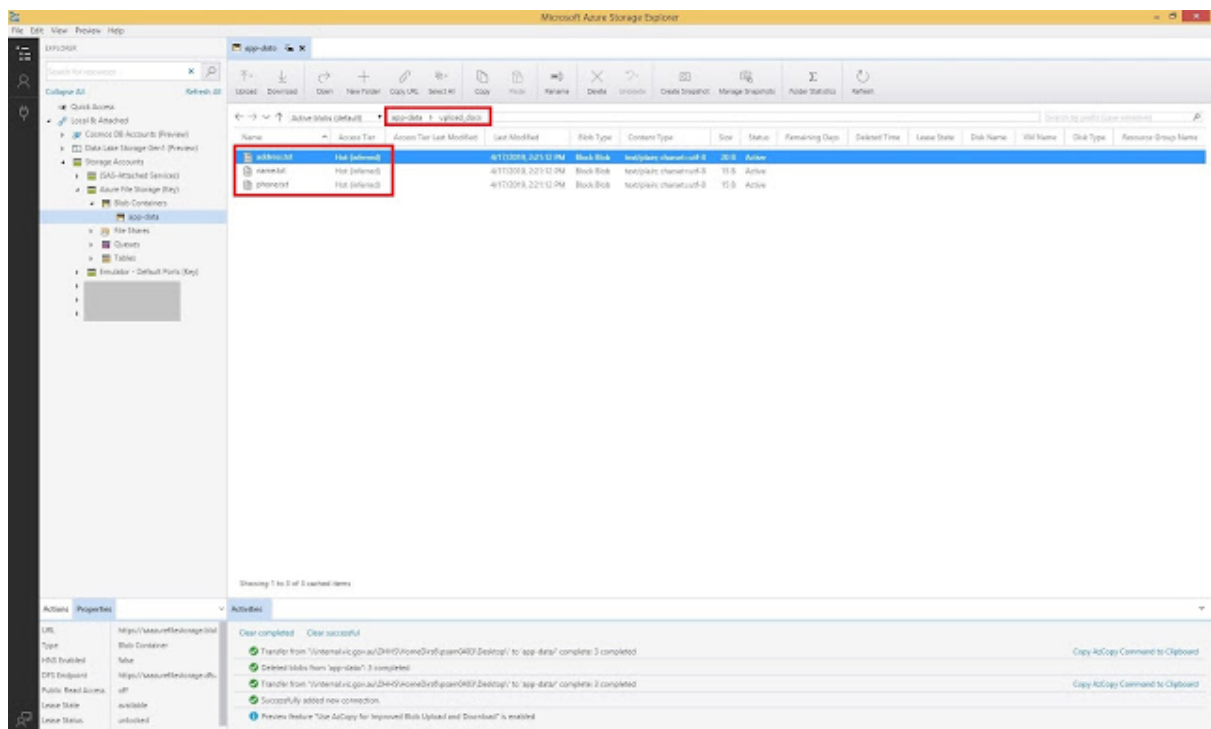
Block Blob ▼

Destination directory:

/

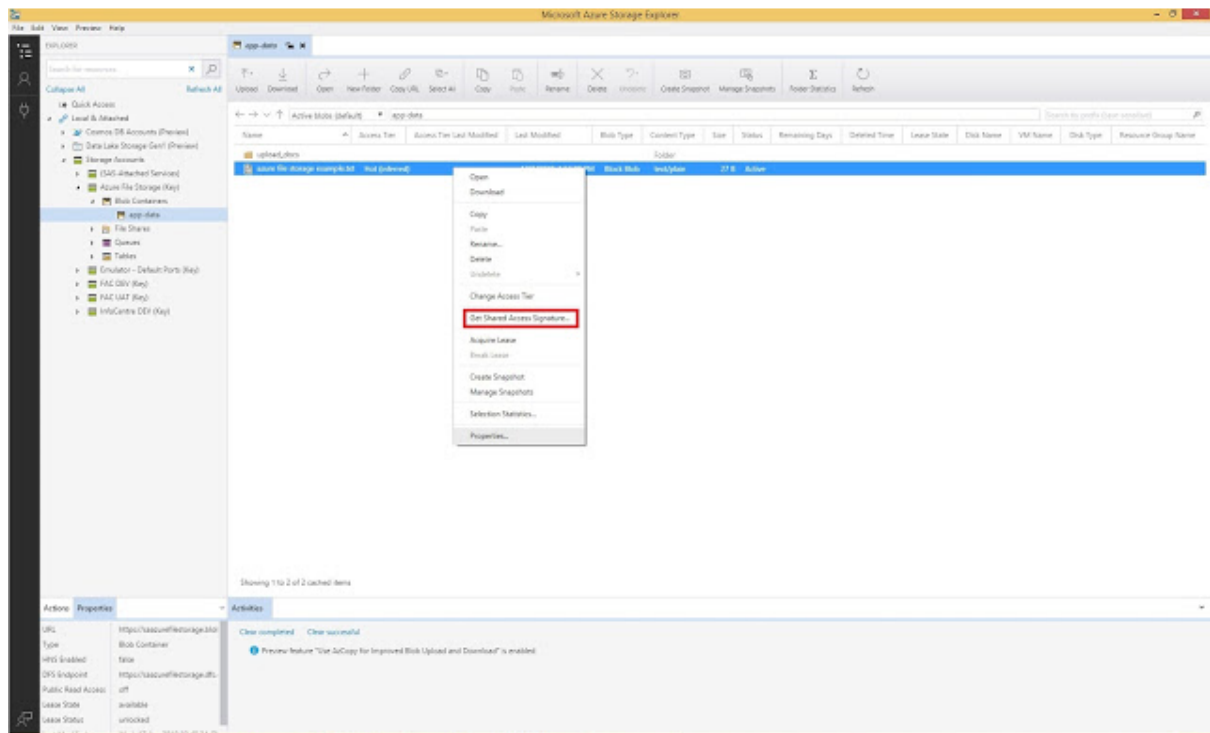
Upload

Cancel



Step 07: Generate SAS from Widows Azure Storage Explorer

Right click on the file.



Microsoft Azure Storage Explorer - Generate Shared Access Sig... ✕

Shared Access Signature

Access policy: (none) ▼

Start time: 03/05/2019 02:54 PM

Expiry time: 04/05/2019 02:54 PM

Time zone:

☒ Local

☐ UTC

Permissions:

☒ Read

☐ Write

☐ Delete

☒ List

☐ Generate container-level shared access signature URI

Create Cancel