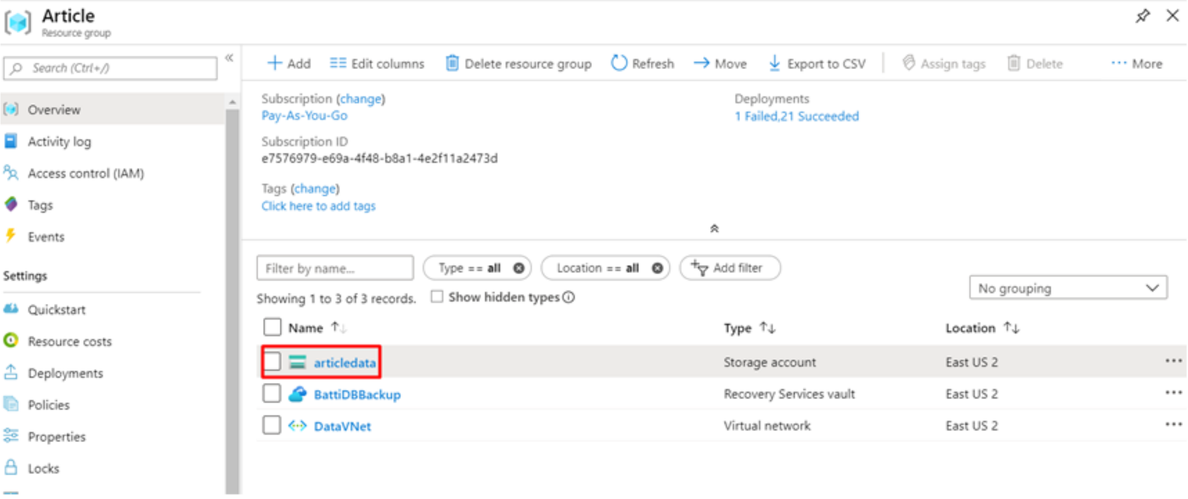
How to Protect an Azure Storage Account using Shared Access Signatures?

The Shared Access Signature (SAS) is the mechanism that restricts access to Azure Storage. It is one of the more secure ways to provide access to our storage account. It eliminates the need for Access Keys to gain access to your Azure storage account.

The two types of SAS

- **Service Level** – Gives access to a resource in just one of the storage services: Blob, Queue, Table and File
- **Account Level** – Gives access to resources in one or more of the storage services. All of the operations available via a Service Level SAS are also available via an Account Level SAS.

Step 1: For this example, we have the Storage Account Name, "**articledata**", under the Article Resource Group, click "**articledata**".



Subscription (change)
Pay-As-You-Go

Subscription ID
e7576979-e69a-4f48-b8a1-4e2f11a2473d

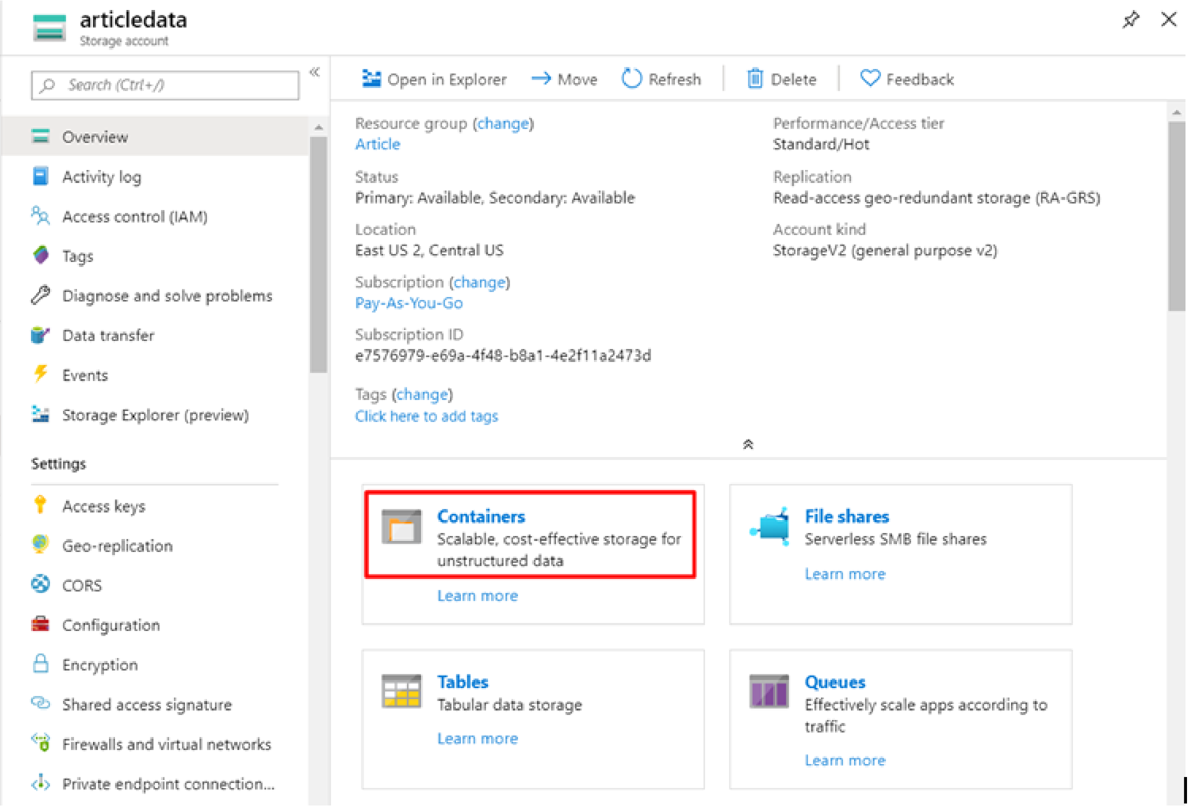
Tags (change)
Click here to add tags

Deployments
1 Failed,21 Succeeded

Showing 1 to 3 of 3 records.

Name	Type	Location
articledata	Storage account	East US 2
BattiDBBackup	Recovery Services vault	East US 2
DataVNet	Virtual network	East US 2

Step 2: In the "**articledata**" Storage Account, Click Containers.



Resource group (change)
Article

Status
Primary: Available, Secondary: Available

Location
East US 2, Central US

Subscription (change)
Pay-As-You-Go

Subscription ID
e7576979-e69a-4f48-b8a1-4e2f11a2473d

Tags (change)
Click here to add tags

Performance/Access tier
Standard/Hot

Replication
Read-access geo-redundant storage (RA-GRS)

Account kind
StorageV2 (general purpose v2)

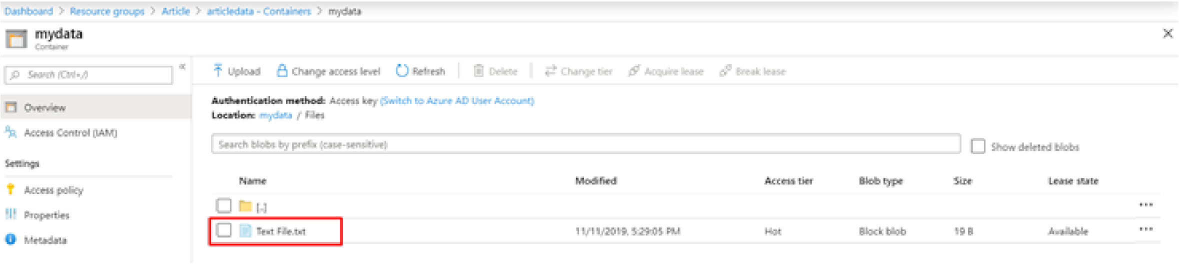
Containers
Scalable, cost-effective storage for unstructured data
Learn more

File shares
Serverless SMB file shares
Learn more

Tables
Tabular data storage
Learn more

Queues
Effectively scale apps according to traffic
Learn more

Step 3: Under the “Container”, we have already uploaded one text file named TextFile.

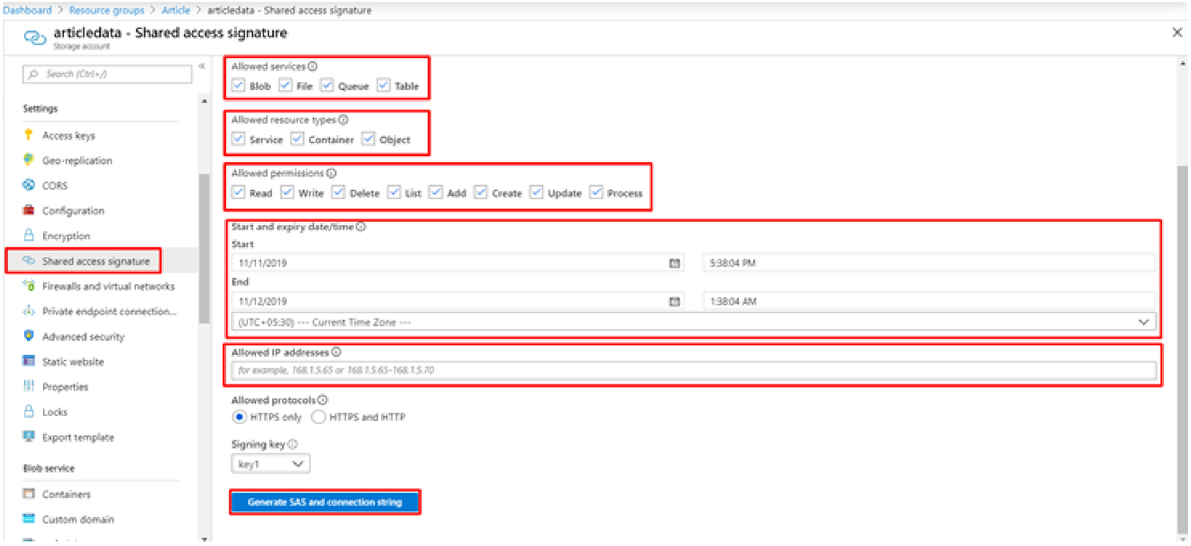


Authentication method: Access key (Switch to Azure AD User Account)
Location: mydata / Files

	Name	Modified	Access tier	Blob type	Size	Lease state
□	[..]					
□	Text File.txt	11/11/2019, 5:29:05 PM	Hot	Block blob	19 B	Available

Step 4: Now step back to our Storage Account Name “**article data**”, and then click “**Shared Access Signature**” under the settings.

- **Allowed Service** – We can select what are the services that we can allow to the user.
- **Allowed Permission** – We can select what kind of permission to allow to the user.
- **Start and End** – We can set the availability time period.
- **Allowed IP Address** – We can whitelist the IP access to our storage account.



articledata - Shared access signature

Allowed services
☑ Blob ☑ File ☑ Queue ☑ Table

Allowed resource types
☑ Service ☑ Container ☑ Object

Allowed permissions
☑ Read ☑ Write ☑ Delete ☑ List ☑ Add ☑ Create ☑ Update ☑ Process

Start and expiry date/time
Start
11/11/2019 5:38:04 PM
End
11/12/2019 1:38:04 AM
(UTC+05:30) --- Current Time Zone ---

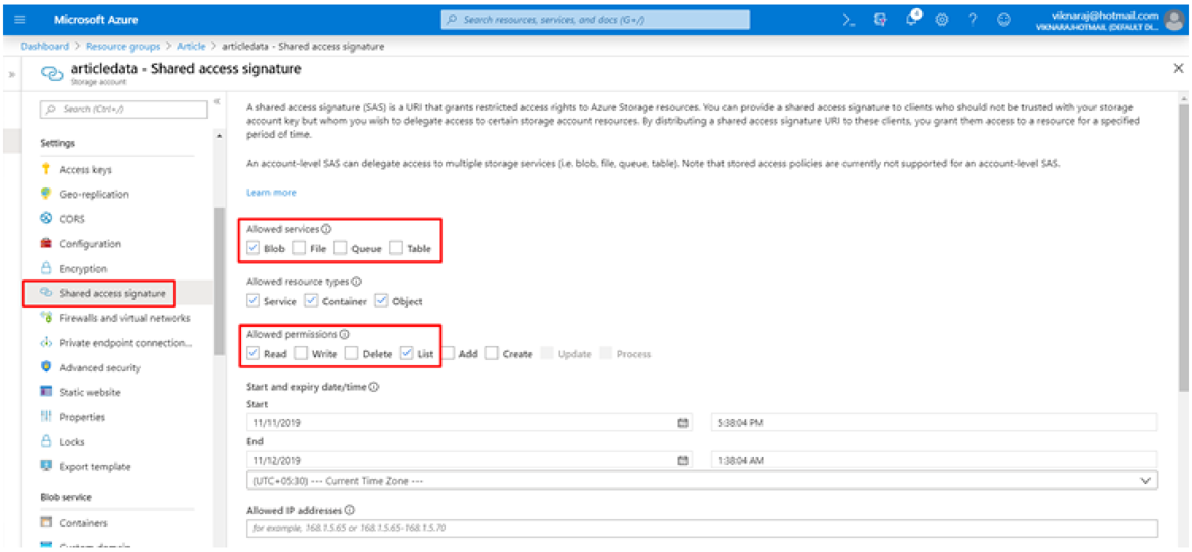
Allowed IP addresses
for example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Allowed protocols
● HTTPS only ○ HTTPS and HTTP

Signing key
key1

Generate SAS and connection string

Step 5: Now we are going to grant permission so users can “**Read and List**” the Documents under the Storage account, but they can’t “**Delete, Write, Add or Create**” any documents in the Storage Account. Click “**Generate SAS and Connection String**”.



articledata - Shared access signature

A shared access signature (SAS) is a URI that grants restricted access rights to Azure Storage resources. You can provide a shared access signature to clients who should not be trusted with your storage account key but whom you wish to delegate access to certain storage account resources. By distributing a shared access signature URI to these clients, you grant them access to a resource for a specified period of time.

An account-level SAS can delegate access to multiple storage services (i.e. blob, file, queue, table). Note that stored access policies are currently not supported for an account-level SAS.

Learn more

Allowed services
☑ Blob ☐ File ☐ Queue ☐ Table

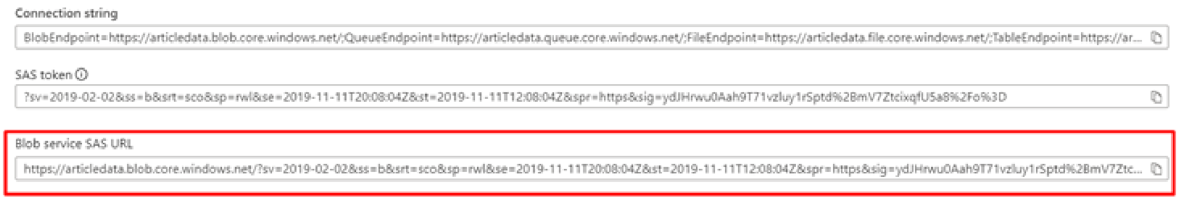
Allowed resource types
☑ Service ☑ Container ☑ Object

Allowed permissions
☑ Read ☐ Write ☐ Delete ☑ List ☐ Add ☐ Create ☐ Update ☐ Process

Start and expiry date/time
Start
11/11/2019 5:38:04 PM
End
11/12/2019 1:38:04 AM
(UTC+05:30) --- Current Time Zone ---

Allowed IP addresses
for example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Step 6: After Generating the SAS and Connection String, copy the “**Blob Service SAS URL**”.

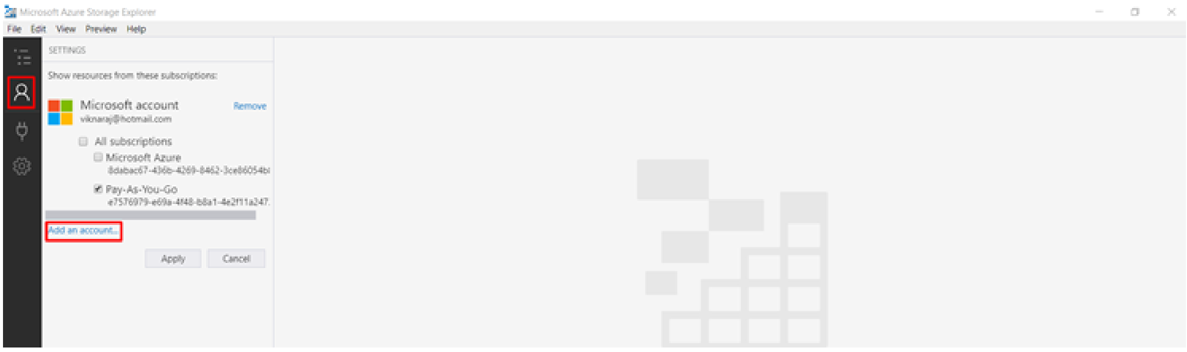


Connection string
```
BlobEndpoint=https://articledata.blob.core.windows.net/;QueueEndpoint=https://articledata.queue.core.windows.net/;FileEndpoint=https://articledata.file.core.windows.net/;TableEndpoint=https://ar...
```

SAS token
```
?sv=2019-02-02&ss=b&srt=sco&sp=rwl&se=2019-11-11T20:08:04Z&st=2019-11-11T12:08:04Z&spr=https&sig=ydJHrwu0Aah9T71vzluy1rSptd%2BmV7ZtcixqfU5a8%2Fo%3D
```

Blob service SAS URL
```
https://articledata.blob.core.windows.net/?sv=2019-02-02&ss=b&srt=sco&sp=rwl&se=2019-11-11T20:08:04Z&st=2019-11-11T12:08:04Z&spr=https&sig=ydJHrwu0Aah9T71vzluy1rSptd%2BmV7Ztc...
```

Step 7: Open the Microsoft Azure Storage Explorer, and then click “**Add an Account**”.



Show resources from these subscriptions:

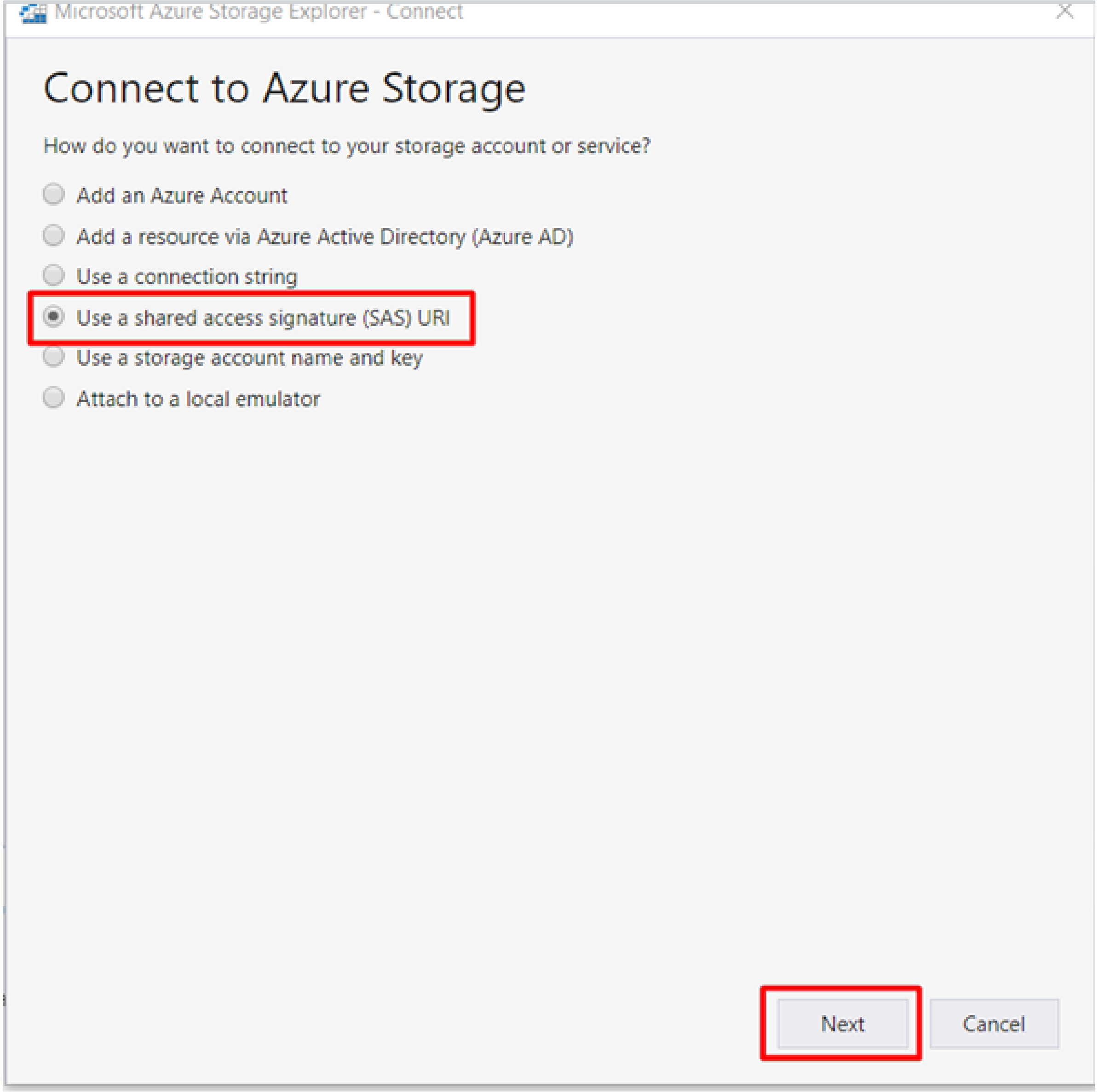
Microsoft account
viknaraj@hotmail.com

□ All subscriptions

□ Microsoft Azure
8dabac67-436b-4269-8462-3ce86054b

☑ Pay-As-You-Go
e7576979-e69a-4f48-b8a1-4e2f11a247.

Step 8: In the Connect to Azure Storage select “**Use a shared access signature (SAS) URI**”, and then click Next.

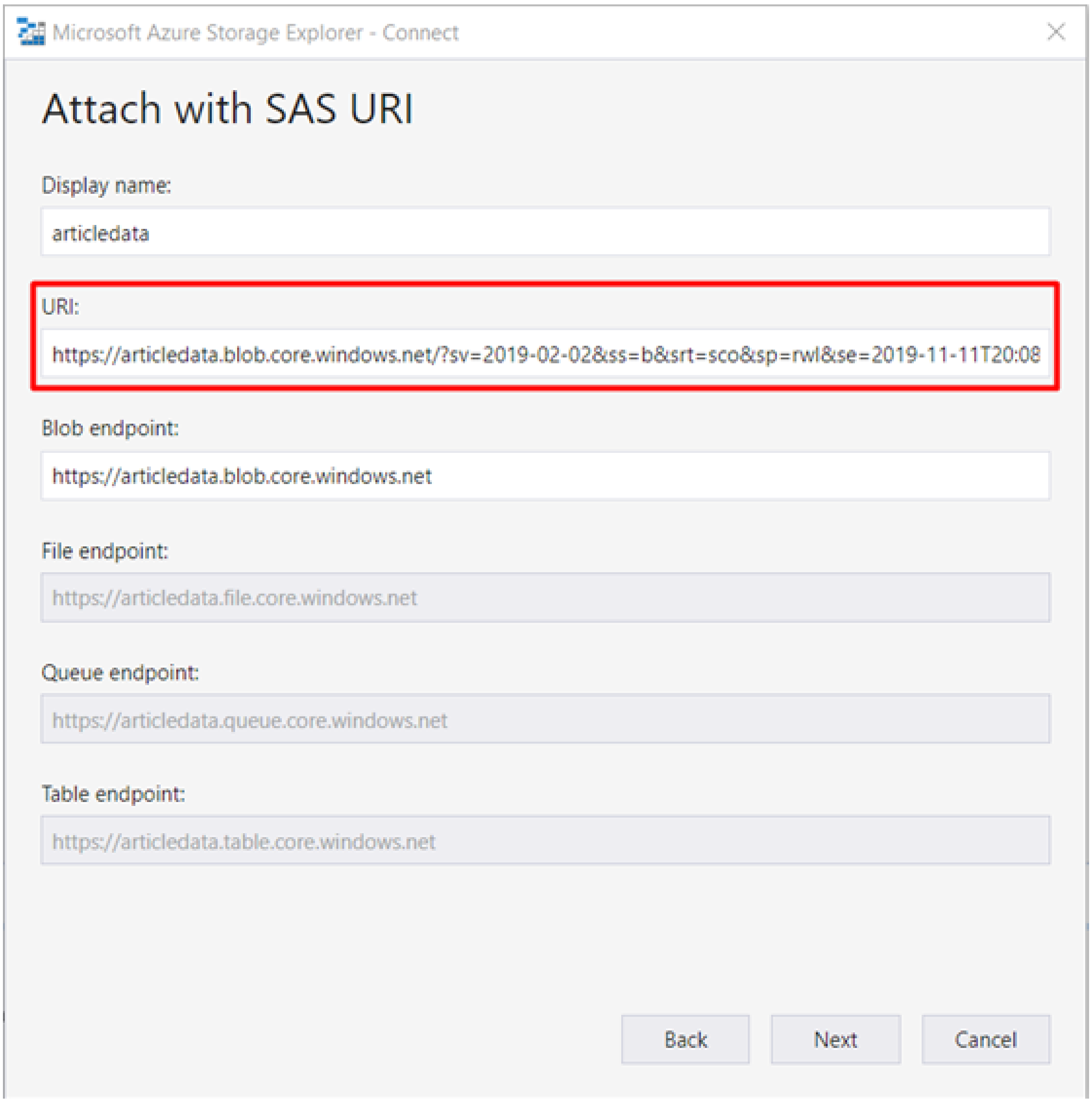


Connect to Azure Storage

How do you want to connect to your storage account or service?

○ Add an Azure Account
○ Add a resource via Azure Active Directory (Azure AD)
○ Use a connection string
● Use a shared access signature (SAS) URI
○ Use a storage account name and key
○ Attach to a local emulator

Step 9: Paste the URL that we copied in step 6. When we paste the URL, it automatically updates other text boxes, then click Next.



Attach with SAS URI

Display name:

articledata

URI:

https://articledata.blob.core.windows.net/?sv=2019-02-02&ss=b&srt=sco&sp=rwl&se=2019-11-11T20:08

Blob endpoint:

https://articledata.blob.core.windows.net

File endpoint:

https://articledata.file.core.windows.net

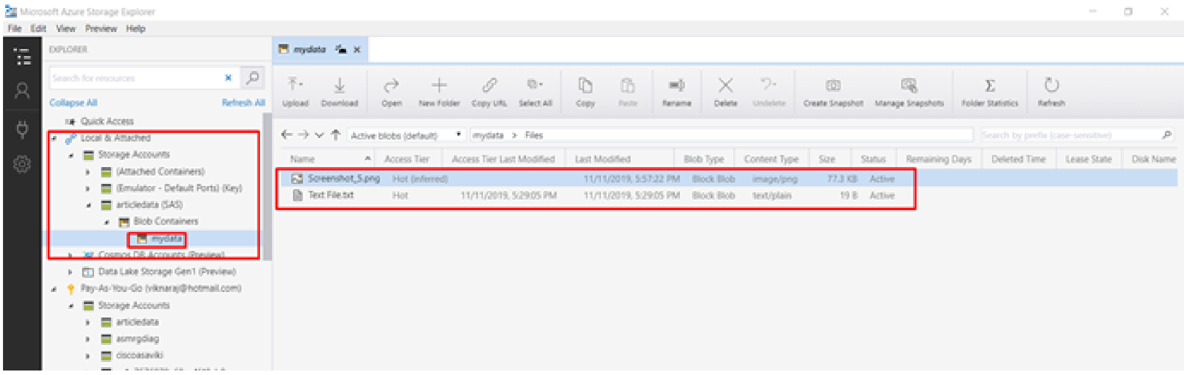
Queue endpoint:

https://articledata.queue.core.windows.net

Table endpoint:

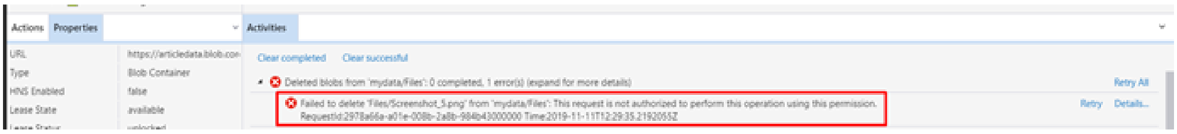
https://articledata.table.core.windows.net

Step 10: In our Storage Account we can find the “mydata” folder. Under the folder we have two files, we can read these because we have selected the “Read” permission. Now we try to remove the “Screenshot_5.png” file, select the file and click Delete.



Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Type	Size	Status	Remaining Days	Deleted Time	Lease State	Disk Name
Screenshot_5.png	Hot (inferred)		11/11/2019, 5:57:22 PM	Block Blob	image/png	77.3 KB	Active				
Text File.txt	Hot	11/11/2019, 5:29:05 PM	11/11/2019, 5:29:05 PM	Block Blob	text/plain	19 B	Active				

Step 11: When we click Delete, we can check the “Activities” it’s saying that we can’t perform this operation because we don’t have the permission to delete.

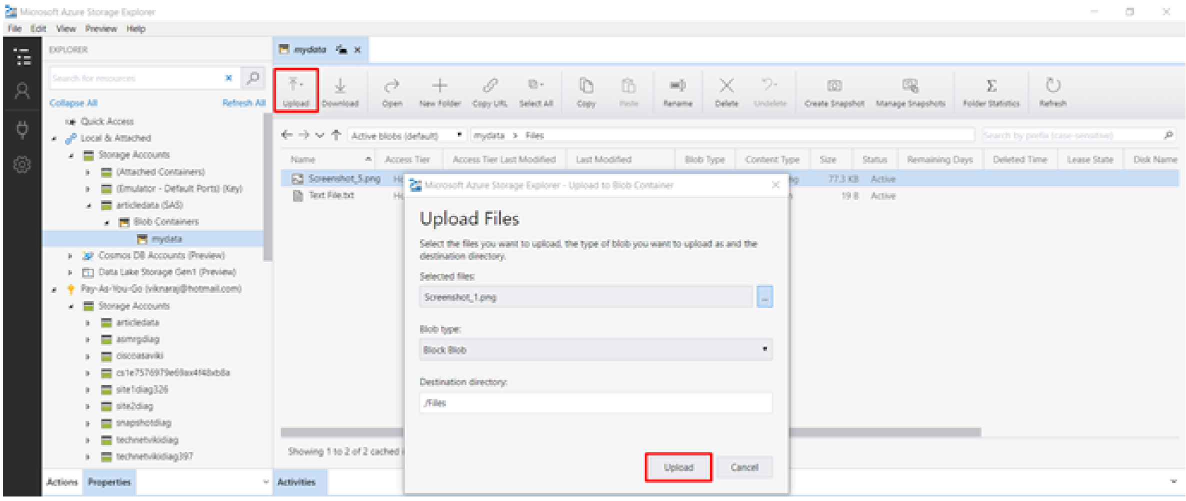


URL	https://articledata.blob.cor
Type	Blob Container
HNS Enabled	false
Lease State	available
Lease Status	unlocked

Deleted blobs from 'mydata/Files': 0 completed, 1 error(s) (expand for more details)

Failed to delete 'Files/Screenshot_5.png' from 'mydata/Files': This request is not authorized to perform this operation using this permission. RequestId:2978a66a-a01e-008b-2a8b-984b43000000 Time:2019-11-11T12:29:35.2192055Z

Step 12: We can try to upload the file to our storage account, so click “Upload” and then select the files and click the “Upload” button. Also, you get the same error.



Upload Files

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

Selected files: Screenshot_1.png

Blob type: Block Blob

Destination directory: /Files

Summary

In this demo, we have learned how to protect our Storage account using Shared Access Signature (SAS)