Different file types to know –

- Avro - this is a row-based file format, each record in the file contains a header that describes the structure of the data in the record, the data itself is stored in the binary format, the format is idea for compressing data, results in less storage and requires less bandwidth
- Parquet – is a columnar data format, data for each column is stored together in something knows as a row group, it supports compression and different encoding schemes

Azure Data Lake Gen2:

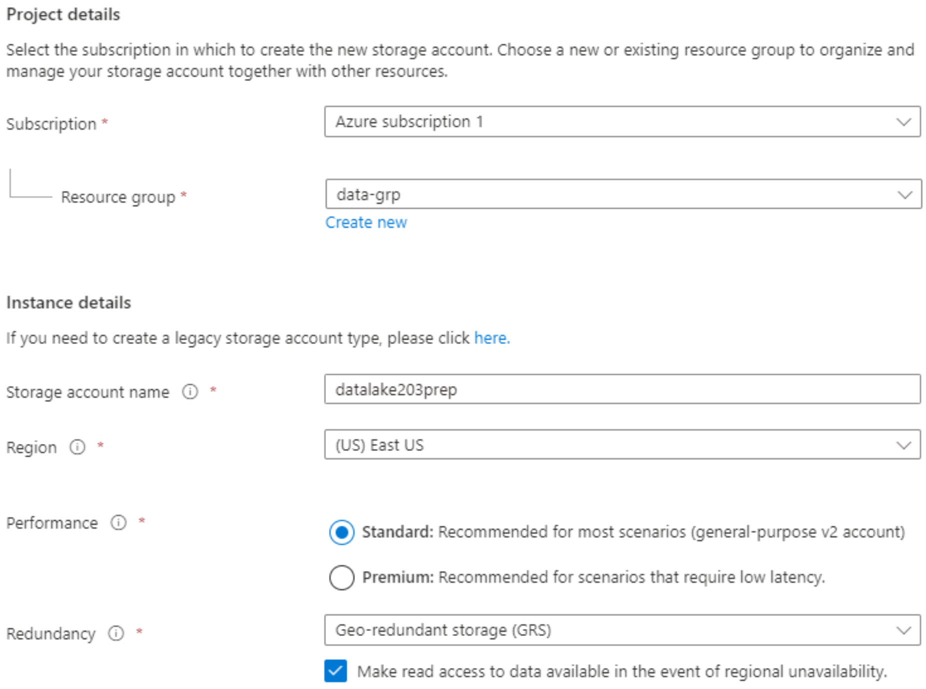
Built on top of the Azure blob storage. Gives the ability to host an enterprise data lake on azure. Can also get the feature of a hierarchical namespace on top of blob storage. Helps organize objects/files into hierarchy of directories for efficient data access.

A data lake is used to store large amounts of data in its native, raw format. They're optimized for storing TBs of data. The data could come form a variety of sources and in a variety of forms.

Creating an Azure Data Lake Gen2

From the home dashboard, go to create a resource and select Storage Account.

Select your subscription, resource group, give a unique storage account name and leave everything else as is. Proceed to next.



Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription * : Azure subscription 1

Resource group * : data-grp

Create new

Instance details

If you need to create a legacy storage account type, please click here.

Storage account name ⓘ * : datalake203prep

Region ⓘ * : (US) East US

Performance ⓘ *

● **Standard:** Recommended for most scenarios (general-purpose v2 account)

○ **Premium:** Recommended for scenarios that require low latency.

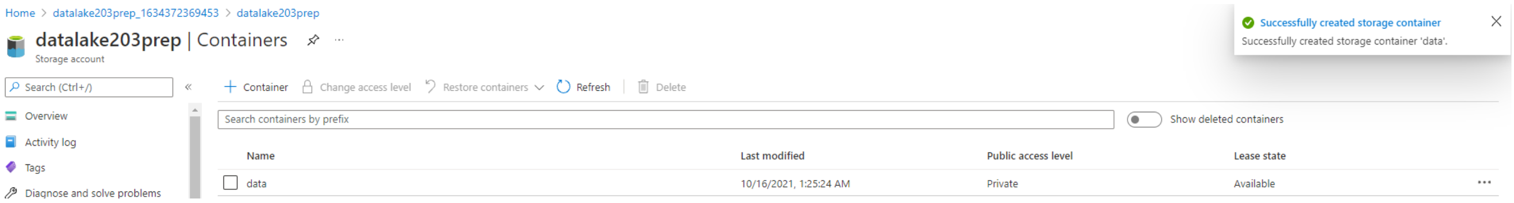
Redundancy ⓘ * : Geo-redundant storage (GRS)

☑ Make read access to data available in the event of regional unavailability.

In the next screen scroll down to Data Lake Storage Gen2 and enable hierarchical namespace. Leave all the other settings as they are and create the resource.

After the storage account has been created, you'd see that the overall layout is the same as a normal storage account that we had created earlier.

Now click on containers, create container with private access.



Home > datalake203prep_1634372369453 > datalake203prep

datalake203prep | Containers

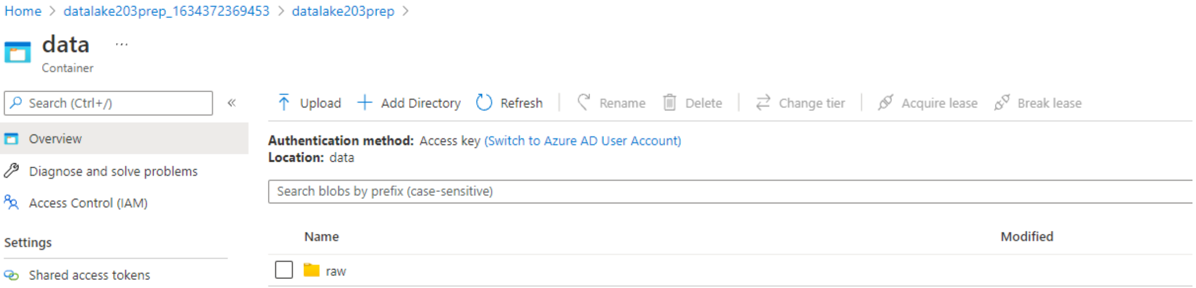
Storage account

Successfully created storage container

Successfully created storage container 'data'.

	Name	Last modified	Public access level	Lease state
□	data	10/16/2021, 1:25:24 AM	Private	Available

In the container we can now create a folder.



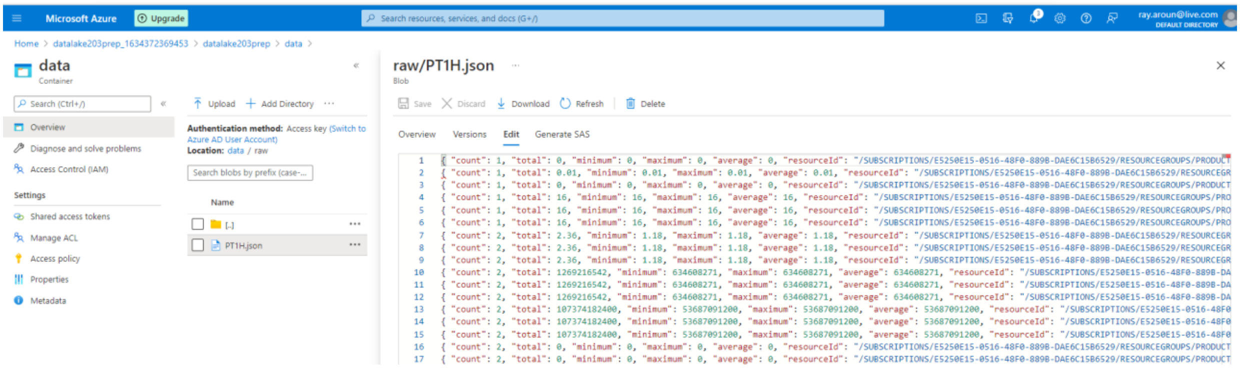
Authentication method: Access key (Switch to Azure AD User Account)
Location: data

Search blobs by prefix (case-sensitive)

Name	Modified
□ raw	

In the raw folder we upload a json file.

Then we can see the file uploaded –



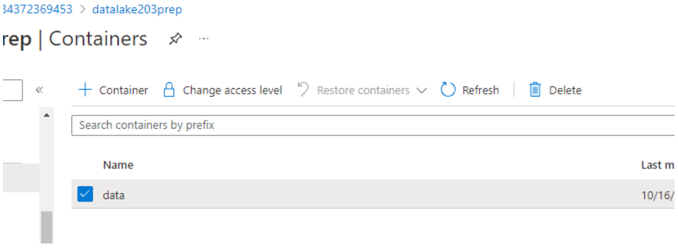
```
{ "count": 1, "total": 0, "minimum": 0, "maximum": 0, "average": 0, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DAE6C15B6529/RESOURCEGROUPS/PRODUCT
{ "count": 1, "total": 0.01, "minimum": 0.01, "maximum": 0.01, "average": 0.01, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DAE6C15B6529/RESOURCEGR
{ "count": 1, "total": 0, "minimum": 0, "maximum": 0, "average": 0, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DAE6C15B6529/RESOURCEGROUPS/PRODUCT
{ "count": 1, "total": 16, "minimum": 16, "maximum": 16, "average": 16, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DAE6C15B6529/RESOURCEGROUPS/PRO
{ "count": 1, "total": 16, "minimum": 16, "maximum": 16, "average": 16, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DAE6C15B6529/RESOURCEGROUPS/PRO
{ "count": 1, "total": 16, "minimum": 16, "maximum": 16, "average": 16, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DAE6C15B6529/RESOURCEGROUPS/PRO
{ "count": 2, "total": 2.36, "minimum": 1.18, "maximum": 1.18, "average": 1.18, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DAE6C15B6529/RESOURCEGR
{ "count": 2, "total": 2.36, "minimum": 1.18, "maximum": 1.18, "average": 1.18, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DAE6C15B6529/RESOURCEGR
{ "count": 2, "total": 2.36, "minimum": 1.18, "maximum": 1.18, "average": 1.18, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DAE6C15B6529/RESOURCEGR
{ "count": 2, "total": 1269216542, "minimum": 634608271, "maximum": 634608271, "average": 634608271, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DA
{ "count": 2, "total": 1269216542, "minimum": 634608271, "maximum": 634608271, "average": 634608271, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DA
{ "count": 2, "total": 1269216542, "minimum": 634608271, "maximum": 634608271, "average": 634608271, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DA
{ "count": 2, "total": 107374182400, "minimum": 53687091200, "maximum": 53687091200, "average": 53687091200, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0
{ "count": 2, "total": 107374182400, "minimum": 53687091200, "maximum": 53687091200, "average": 53687091200, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0
{ "count": 2, "total": 107374182400, "minimum": 53687091200, "maximum": 53687091200, "average": 53687091200, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0
{ "count": 2, "total": 0, "minimum": 0, "maximum": 0, "average": 0, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DAE6C15B6529/RESOURCEGROUPS/PRODUCT
{ "count": 2, "total": 0, "minimum": 0, "maximum": 0, "average": 0, "resourceId": "/SUBSCRIPTIONS/E5250E15-0516-48F0-889B-DAE6C15B6529/RESOURCEGROUPS/PRODUCT
```

This was just a quick demo of how we can upload files to Data lakes.

Visualizing Data on your Azure Data Lake using PowerBI

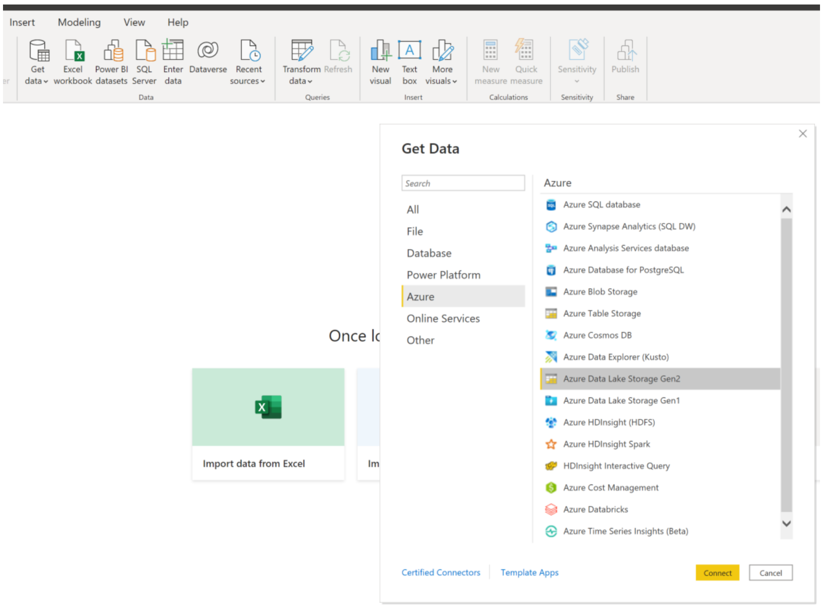
In the raw folder upload the Log.csv file. It contains the information of the Azure activity done by a user (obtained from the Monitor section > view all activity logs).

Now after going back to the containers menu and change the access level to blob for anonymous access.



Name	Last m
☑ data	10/16/

Open up your PowerBI desktop. Select Get Data > Azure > Azure Data Lake Storage Gen2



Get Data

Search

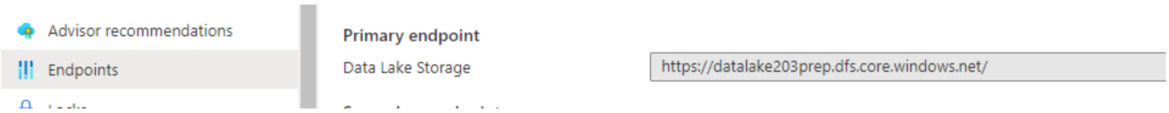
- All
- File
- Database
- Power Platform
- Azure
- Online Services
- Other

Azure

- Azure SQL database
- Azure Synapse Analytics (SQL DW)
- Azure Analysis Services database
- Azure Database for PostgreSQL
- Azure Blob Storage
- Azure Table Storage
- Azure Cosmos DB
- Azure Data Explorer (Kusto)
- Azure Data Lake Storage Gen2
- Azure Data Lake Storage Gen1
- Azure HDInsight (HDFS)
- Azure HDInsight Spark
- HDInsight Interactive Query
- Azure Cost Management
- Azure Databricks
- Azure Time Series Insights (Beta)

Certified Connectors Template Apps Connect Cancel

When it asks for URL, go to your Datalake Gen2 landing page, on the left hand side menu, scroll down to Endpoints and copy the primary endpoint for the Datalake storage.

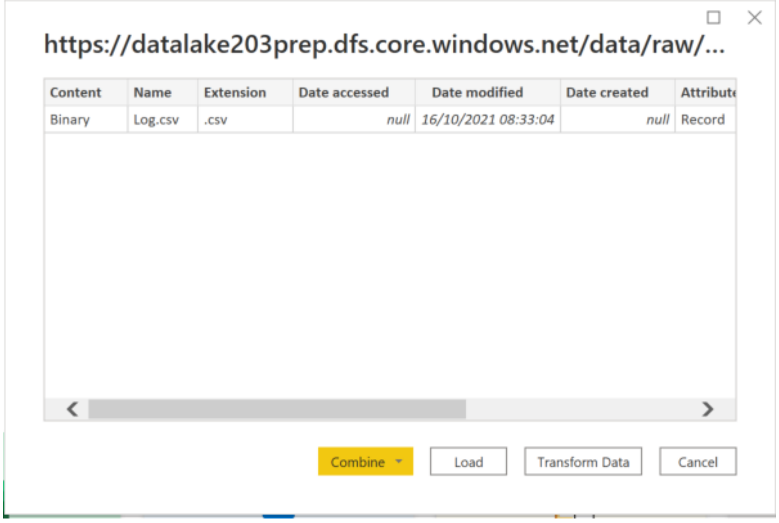


Primary endpoint

Data Lake Storage https://datalake203prep.dfs.core.windows.net/

Paste this link back on PowerBI and add /data/raw/Log.csv or the path to where your Log.csv is residing. Click okay.

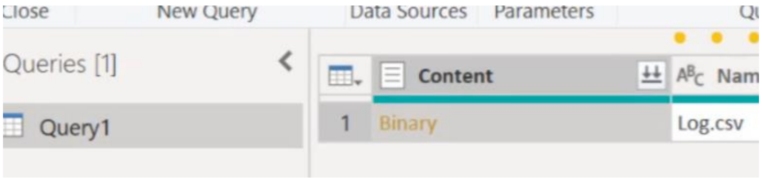
You can then provide the Account key. This is obtained from Access Keys in the left had menu from the Data Lake landing page. After pasting the key and moving forward, you get the following screen where we click transform data.



https://datalake203prep.dfs.core.windows.net/data/raw/...

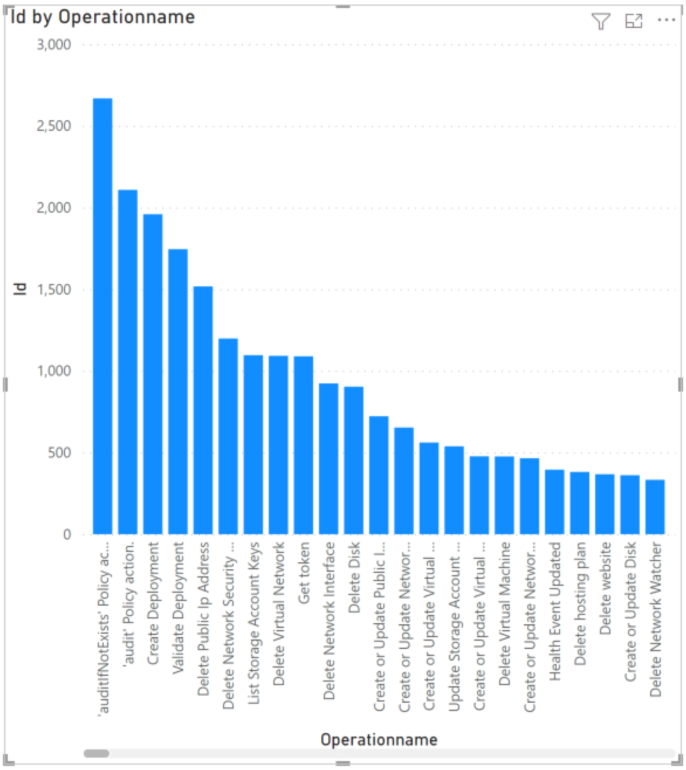
Content	Name	Extension	Date accessed	Date modified	Date created	Attribute
Binary	Log.csv	.csv	null	16/10/2021 08:33:04	null	Record

Click on the Binary under content to display all the data.



	Content	Name
1	Binary	Log.csv

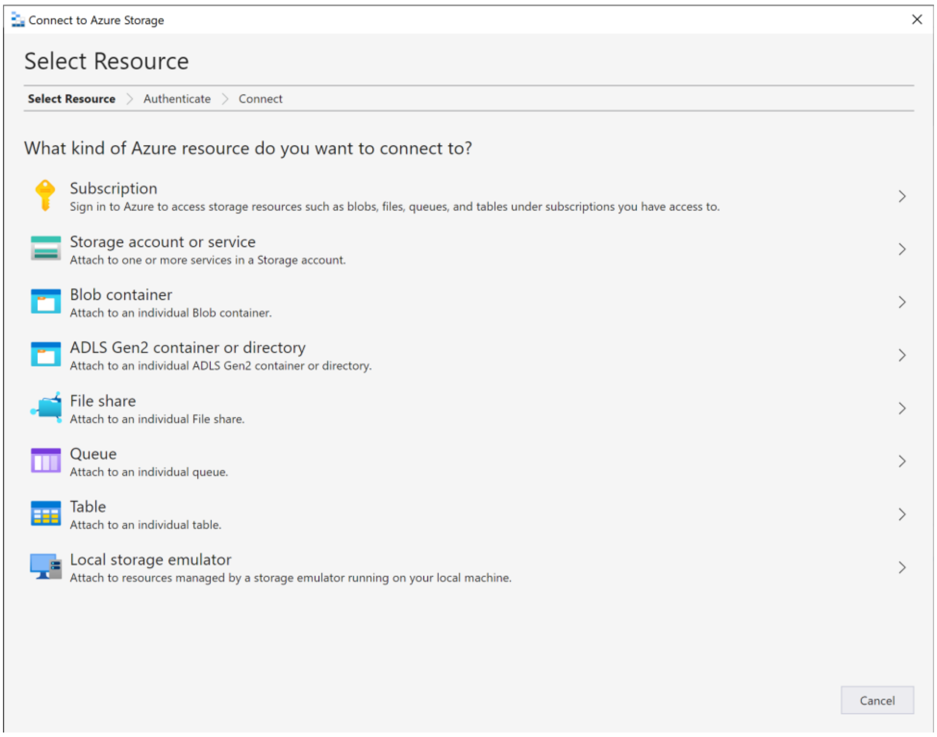
Click close and apply to get to the dashboard screen. There you can now create visualizations for the data that is residing on the Azure Data Lake Gen2. An example bar char of Id by Operationname would look like this.



Azure storage explorer

Like the name suggests, used to explore storage accounts. If you have employees in an organization that only need to access data in the storage accounts, then this is what they can use.

This is how it looks like when you first download the Azure storage explorer application.



Connect to Azure Storage

Select Resource

Select Resource > Authenticate > Connect

What kind of Azure resource do you want to connect to?

Subscription
Sign in to Azure to access storage resources such as blobs, files, queues, and tables under subscriptions you have access to.

Storage account or service
Attach to one or more services in a Storage account.

Blob container
Attach to an individual Blob container.

ADLS Gen2 container or directory
Attach to an individual ADLS Gen2 container or directory.

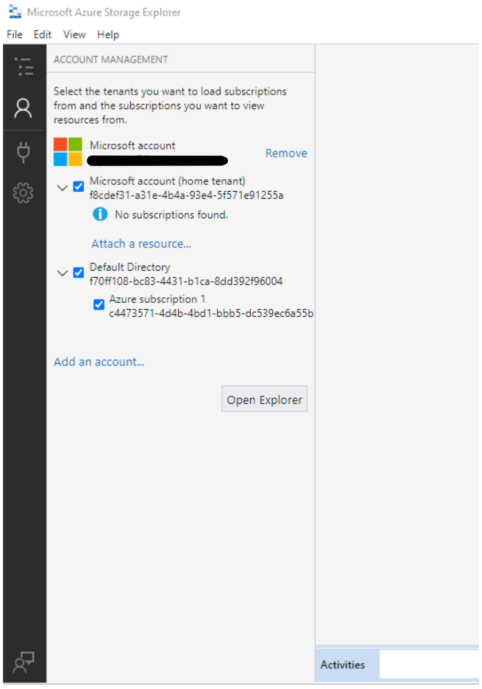
File share
Attach to an individual File share.

Queue
Attach to an individual queue.

Table
Attach to an individual table.

Local storage emulator
Attach to resources managed by a storage emulator running on your local machine.

To connect, I select Subscription > choose Azure > Log in using your Azure credentials.



Microsoft Azure Storage Explorer

ACCOUNT MANAGEMENT

Select the tenants you want to load subscriptions from and the subscriptions you want to view resources from.

Microsoft account
[redacted]

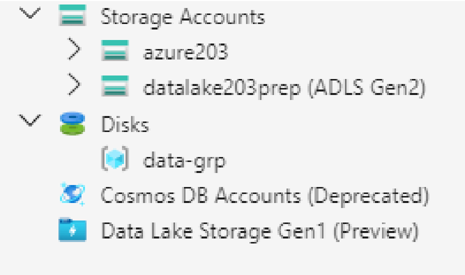
☑ Microsoft account (home tenant)
f8cdef31-a31e-4b4a-93e4-5f571e91255a

No subscriptions found.

☑ Default Directory
f70ff108-bc83-4431-b1ca-8dd392f96004

☑ Azure subscription 1
c4473571-4d4b-4bd1-bbb5-dc539ec6a55b

After reaching the above screen, select your subscription and click open explorer. In the next screen it would show all the different storage accounts that you have setup with the account.



- Storage Accounts
  - azure203
  - datalake203prep (ADLS Gen2)
- Disks
  - data-grp
- Cosmos DB Accounts (Deprecated)
- Data Lake Storage Gen1 (Preview)

Using this explorer, you can download the files from and upload files to the azure storage accounts.

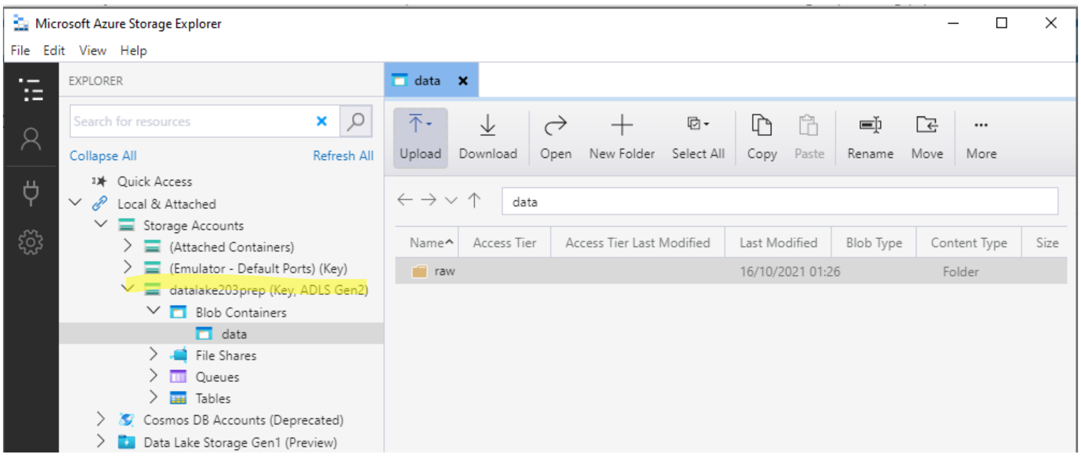
Let's say if a user is only supposed to access only a particular storage account then we can make use of access keys and grant them access to only the data they are supposed to access.

From the left-hand menu click on the profile button to add another account -

From Add an account > Choose Storage account or service > Select account name and key.

Open up the Storage Account that you want to give access to. From the left hand menu, scroll down to access keys, copy the first key and paste it in the Azure storage explorer. Copy the storage account name and paste that as well. Click next.

The data lake shows up under the local and attached Storage accounts option.

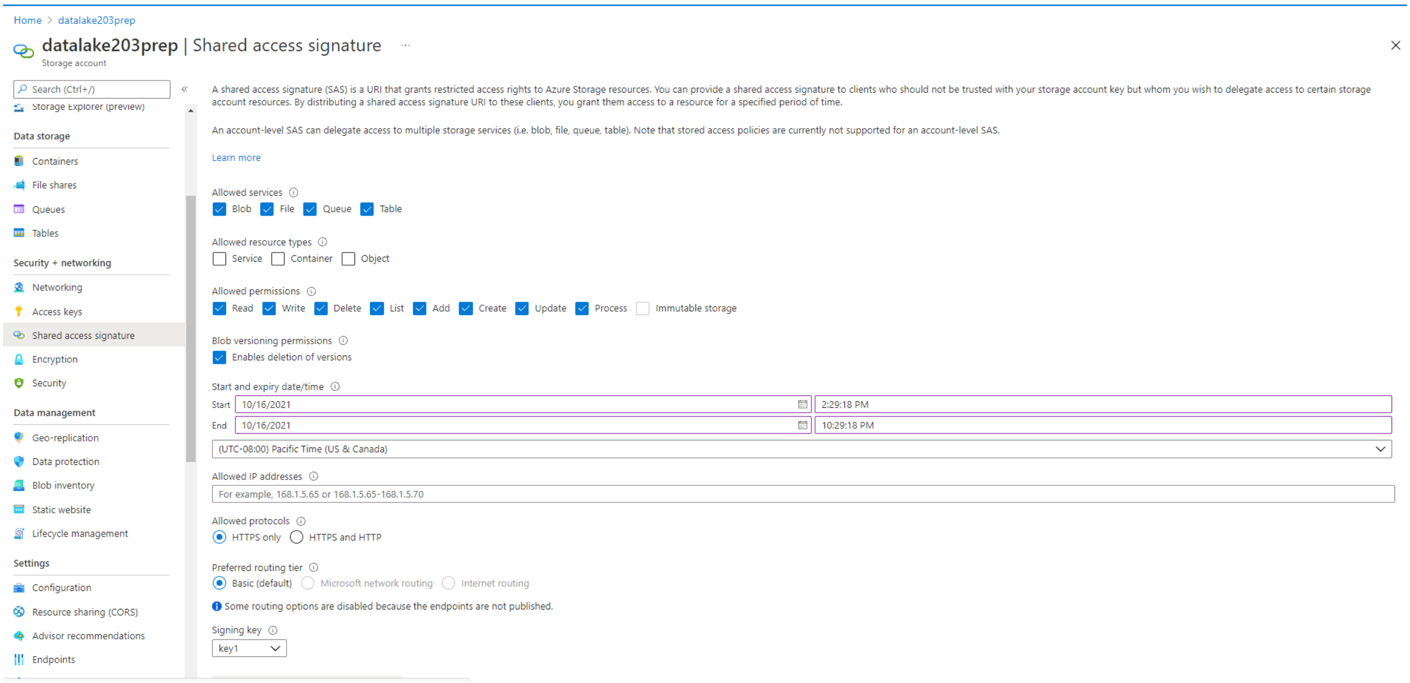


data

Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Type	Size
raw			16/10/2021 01:26		Folder	

Shared Access Signatures

With this you can provide selective access to the services that are present in the storage account. To see the Shared Access Signature menu, open up your storage account and from the left-hand menu scroll down to 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.

Allowed services: ☑ Blob ☑ File ☑ Queue ☑ Table

Allowed resource types: □ Service □ Container □ Object

Allowed permissions: ☑ Read ☑ Write ☑ Delete ☑ List ☑ Add ☑ Create ☑ Update ☑ Process □ Immutable storage

Blob versioning permissions: ☑ Enables deletion of versions

Start and expiry date/time
Start: 10/16/2021 2:29:18 PM
End: 10/16/2021 10:29:18 PM
(UTC-08:00) Pacific Time (US & Canada)

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

Preferred routing tier: ● Basic (default) ○ Microsoft network routing ○ Internet routing

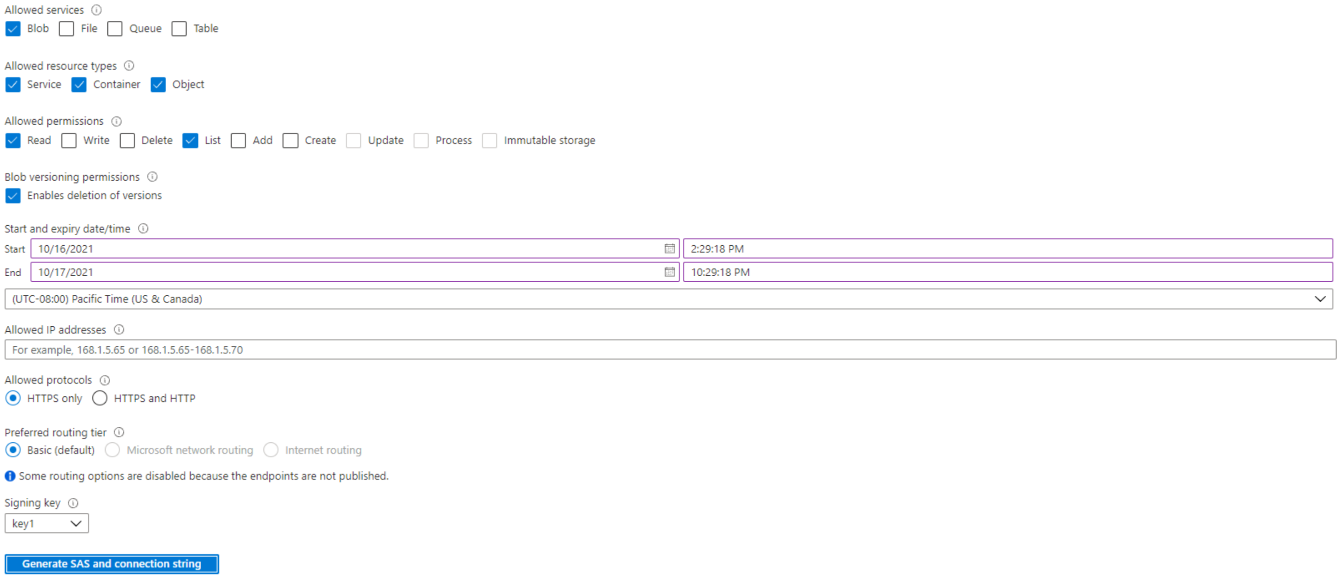
Some routing options are disabled because the endpoints are not published.

Signing key: key1

So probably the main difference between sharing Shared access signatures and Access keys is that:

- With the access keys the user has access to blob containers, files, queues and tables
- With Shared access signature you can decide what service to share within the storage account

For example if you want someone to access the blob container, and be able to access the service, the container and the objects in the container. If they can only Read the files. Expiring the next day. Then it would look like this –



Allowed services ⓘ

☑ Blob □ File □ Queue □ Table

Allowed resource types ⓘ

☑ Service ☑ Container ☑ Object

Allowed permissions ⓘ

☑ Read □ Write □ Delete ☑ List □ Add □ Create □ Update □ Process □ Immutable storage

Blob versioning permissions ⓘ

☑ Enables deletion of versions

Start and expiry date/time ⓘ

Start 10/16/2021 2:29:18 PM

End 10/17/2021 10:29:18 PM

(UTC-08:00) Pacific Time (US & Canada)

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

Preferred routing tier ⓘ

● Basic (default) ○ Microsoft network routing ○ Internet routing

ⓘ Some routing options are disabled because the endpoints are not published.

Signing key ⓘ

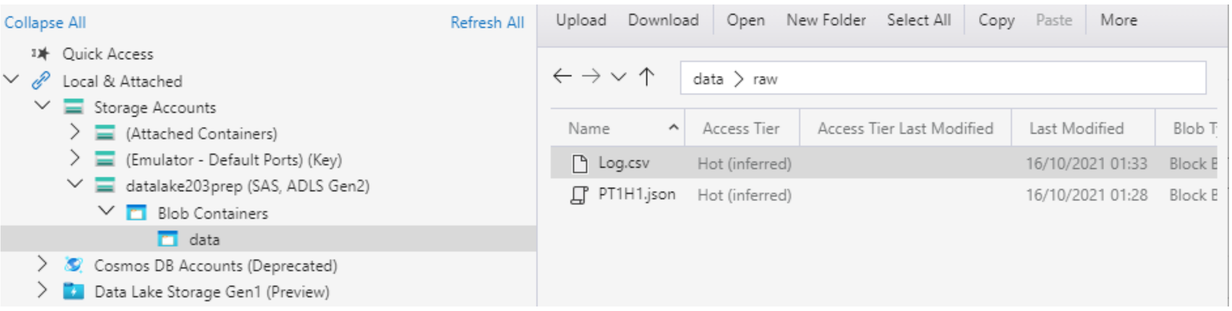
key1

Generate SAS and connection string

Upon clicking the Generate SAS and connection string, it would generate Connection string , SAS token and Blob service SAS URL.

To connect to the storage account with the above specifications, we can again go to the Azure storage explorer. Remove the previously connected Storage account with the access key (right click > detach).

Go to add account again > Storage account or service > Select Shared Access Signature URL > Copy and paste the Blob service SAS URL from Azure to the Azure storage explorer. It would show up as :



Name	Access Tier	Access Tier Last Modified	Last Modified	Blob T
Log.csv	Hot (inferred)		16/10/2021 01:33	Block E
PT1H1.json	Hot (inferred)		16/10/2021 01:28	Block E

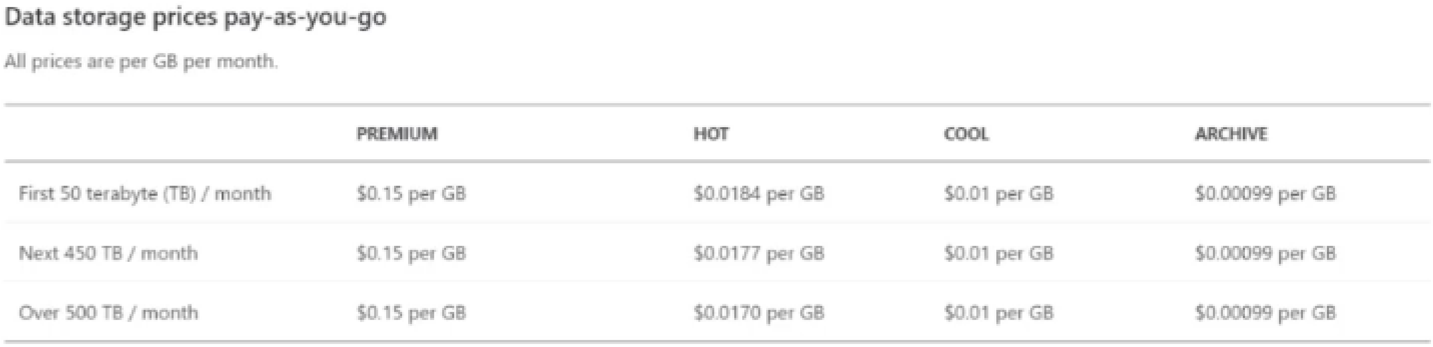
The user would only be able to read and list the files / objects in the containers.

Azure storage account – Redundancy

1. **Locally redundant storage** – When this option is chosen, **three copies** of your data are made or your data is stored at just **one data center.** This helps to protect against server rack of drive failures. So if the data center goes down then the data would be in accessible.
2. **Zone redundant storage** – Helps protects data against data center level failures. Here data is **replicated synchronously across three Azure availability zones.** Each availability zone is a separate physical location with independent power, cooling and networking. If the entire region goes down, then the data would be inaccessible.
3. **Geo redundant storage** – The data is replicated to another region. So if your primary region is Central US then **three copies of your data** are made using LRS. Then the data would be replicated in the East US 2 region where there would also be three copies made of your data.
   - Because of this option there would be increased costs involved as we would be paying for the cost to store data in two locations, for bandwidth to keep the data consistent between two locations, etc.
   - But here the data would always be accessed from the primary location and would only be accessed from the secondary location if the primary location goes down.
4. **Read-access geo redundant storage** – This is just like the geo redundant storage but the data is available at both the primary and secondary locations at the same time.
5. **Geo-zone redundant storage** – Here the data is made available in the primary region across multiple availability zones. But in the secondary region, data is replicated using LRS only.

Azure storage account – Access tier

There are three access tiers – hot, cool and archive (available only on the individual object level). Storage accounts can be configured to have hot or cold access tiers.



Data storage prices pay-as-you-go

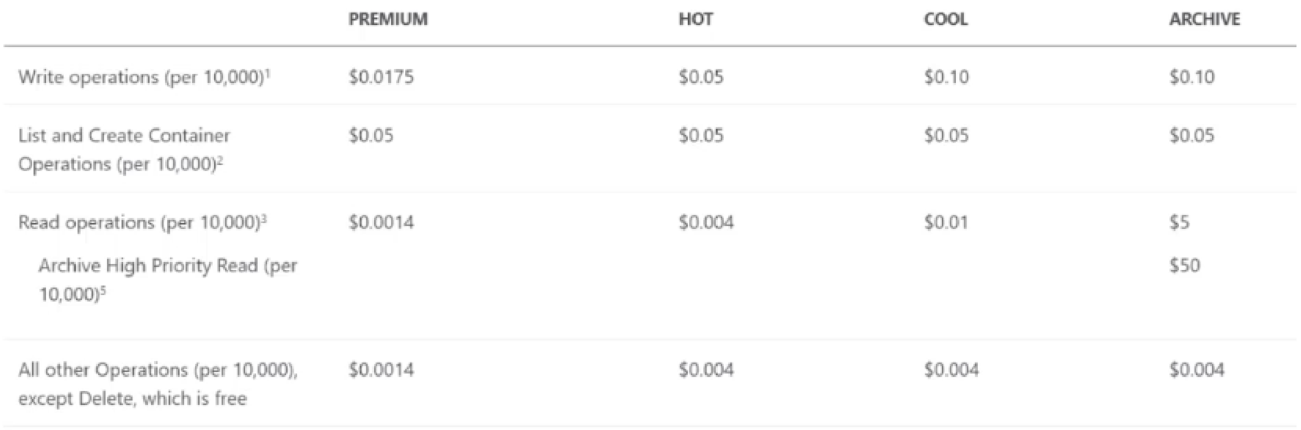
All prices are per GB per month.

	PREMIUM	HOT	COOL	ARCHIVE
First 50 terabyte (TB) / month	$0.15 per GB	$0.0184 per GB	$0.01 per GB	$0.00099 per GB
Next 450 TB / month	$0.15 per GB	$0.0177 per GB	$0.01 per GB	$0.00099 per GB
Over 500 TB / month	$0.15 per GB	$0.0170 per GB	$0.01 per GB	$0.00099 per GB

When data is frequently accessed it should be put in the hot access tier, if slightly less then it could be put in the cool access tier. And if the data would not be accessed but a backup of the data is still required then Archive access tier can be used.

When data is accessed from the Archive tier then it needs to be rehydrated – which probably means that it need to brought back online and would not be instant (similar to the data storage options in Amazon S3). The tier of the file is changed to hot or cold to access the file.

There are prices associated with different operations in different tiers.



	PREMIUM	HOT	COOL	ARCHIVE
Write operations (per 10,000)[1]	$0.0175	$0.05	$0.10	$0.10
List and Create Container Operations (per 10,000)[2]	$0.05	$0.05	$0.05	$0.05
Read operations (per 10,000)[3]	$0.0014	$0.004	$0.01	$5
Archive High Priority Read (per 10,000)[5]				$50
All other Operations (per 10,000), except Delete, which is free	$0.0014	$0.004	$0.004	$0.004

Cool access tier is sued when data is not accessed frequently and is stored for at least 30 days.

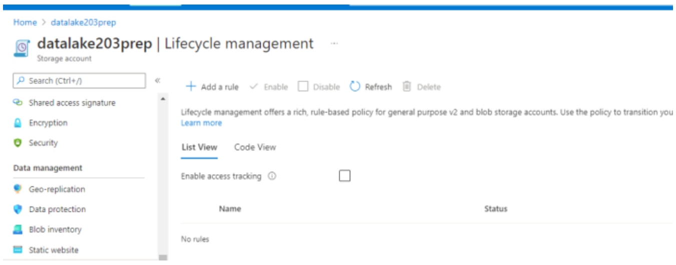
Archive tier is used for that data which is rarely access and stored for at least 180 days.

There are early deletion fees associated with the cool access tier and archive access tier if the tier is change to hot before 30 days and 180 days respectively.

Azure storage account – lifecycle management

If there are a lot of objects that are being managed and if we want to make sure that some of the objects are moved from the hot access tier to the cool access tier based on access timeframe. For this the **lifecycle management** tool can be used.

It can be accessed from the left hand menu on your storage account under data management. Click Add rule –

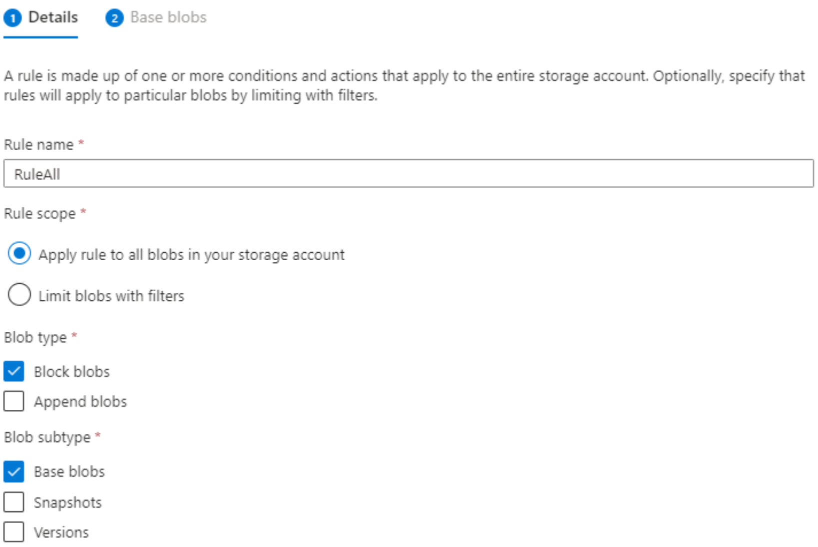


Lifecycle management offers a rich, rule-based policy for general purpose v2 and blob storage accounts. Use the policy to transition you

Enable access tracking □

No rules

Fill out the details as required –



1 Details 2 Base blobs

A rule is made up of one or more conditions and actions that apply to the entire storage account. Optionally, specify that rules will apply to particular blobs by limiting with filters.

Rule name *

RuleAll

Rule scope *

● Apply rule to all blobs in your storage account

○ Limit blobs with filters

Blob type *

☑ Block blobs

□ Append blobs

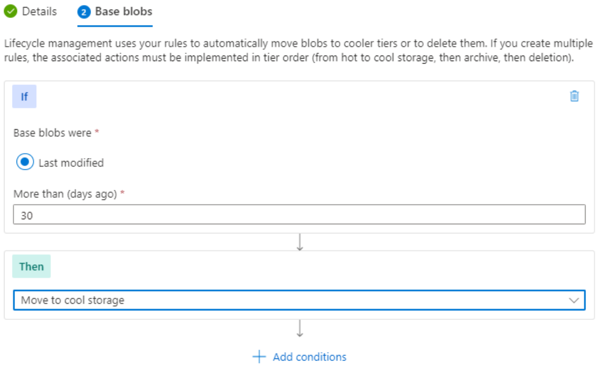
Blob subtype *

☑ Base blobs

□ Snapshots

□ Versions

In the next screen give a number of days after which a certain action needs to occur. Here we are choosing 30 days after which the blob should be moved to the cool storage.



✓ Details 2 Base blobs

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

If

Base blobs were *

● Last modified

More than (days ago) *

30

Then

Move to cool storage

Finish creating the rule and this rule would move the blobs to the cool tier when data hasn't been accessed in the last 30 days.