Cloudera Runtime 7.3.1

atlas-extract-adls

Date published: 2020-07-28
Date modified: 2024-12-10

https://docs.cloudera.com/

Legal Notice

Contents

Before you start

It is assumed that you are familiar with CDP, Azure Environments, and Azure Data Lake Storage Service.

Refer to Resources for Onboarding Azure for CDP to familiarize yourself with the terms used for this feature.

It is assumed that you have performed the steps mentioned in Azure quick start and are familiar with the functioning of Atlas.

Follow the Azure environment documentation to register your Azure environment to CDPD and create required resources.

Introduction to Atlas ADLS Extractor

The Atlas metadata extractor, adls-extractor, for Azure Data Lake Storage is a stand-alone application that you can run on the Atlas host as needed to provide comprehensive metadata for data assets stored in ADLS Gen2.

The adls-extractor reads metadata relevant for Atlas from ADLS artifacts like blobs, directories, containers, storage accounts and publishes them to Atlas.

In order for extraction to function, the following prerequisites must be met:

- Determine the Type of Extraction on page 5
- Updating Extractor Configuration with ADLS Authentication on page 5
- Setting up Azure managed Identity for Extraction on page 12

To perform the extraction follow Running ADLS Metadata Extractor on page 13.

Once the extraction process is completed, you must verify the data that has been published to Atlas. For more information, see Verifying Atlas for the extracted data on page 17.

Terminologies

When you use this document for performing ADLS metadata extraction, there are terminologies and usage of technical terms that you could come across. Use the following information for more information.

ADLS

Azure Data Lake Storage. For more information, see Azure Data Lake Storage.

ADLS

Storage and Blob: Definitions are available in Introduction to Azure Blob Storage.

Atlas

Metadata management Service on CDP. For more information, see Using metadata for cluster governance.

IDBroker

Authentication services within CDP Public Cloud. For more information, see Cloud identity federation

Extraction Prerequisites

Before you perform Atlas metadata extraction, you must understand the prerequisites of this feature, that includes the types of extraction.

Determine the Type of Extraction

Choose Running Bulk Extraction on page 14 if you need to get the complete snapshot of the storage account, container or specific path (configured at allow list in extractor configuration file) metadata at Atlas. Normally recommended for the very first time.

Choose Running Incremental Extraction on page 14 if you need to keep Atlas synchronized with the changes happening in ADLS. This takes more time to process the individual changes compared to bulk extraction. It needs to be run periodically if lineage has to be kept up to date.

If incremental extractions need to be set up to run periodically, it can be done using your own scheduling tools or using crontabs. As you get familiar with the nature of changes within ADLS, viz. volume of changes and the speed of the extraction, you can determine the frequency with which to run the extractor.

For incremental extractions to function, you will need to set up Azure storage queues. Please see Configuring ADLS Gen2 Storage Queue on page 5 for more details.

Note: The success of this operation is dependent on ensuring that the extraction is done before the events are deleted from the ADLS queue.

Updating Extractor Configuration with ADLS Authentication

In the CDP environment Knox IDBroker is used for federated authentication.

The extractor employs the Atlas' kerberized session to authenticate to Knox IDBroker. It receives the delegation token from IDBroker and uses it to request cloud credentials. No configuration change is necessary in this case.

To override this behaviour, the ADLS secret keys can be configured at:

/opt/cloudera/parcels/CDH/lib/atlas/extractors/conf/adls.conf Default

Example: atlas.adls.extraction.account.key=<key>

Configuring ADLS Gen2 Storage Queue

You must create and configure a storage queue for the ADLS Gen2 storage account which was created as part of Extraction Prerequisites on page 4 for storing blob and directory event notifications.

Note: The ADLS storage queue is created only for Incremental Extraction.

Create or Identify a Storage queue

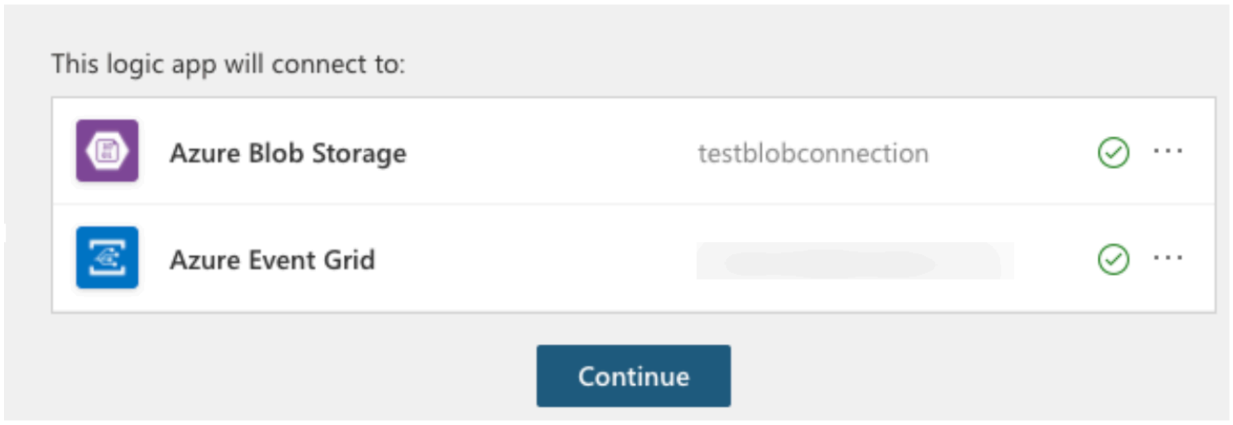
1. On Azure Portal > Storage Accounts > Queue service > Queues > Queue.
2. Provide a name for your queue and click OK. For example, testqueue.

Configuring the Storage queue for Storage events

Configure the above storage queue for the ADLS Gen2 storage account so that it gets all the Blob and Directory events like create, delete, and rename operations.

1. On Azure Portal, navigate to Storage Accounts > Events > Get Started > When a new blob is uploaded > Create. The Logic App Designer window is displayed.

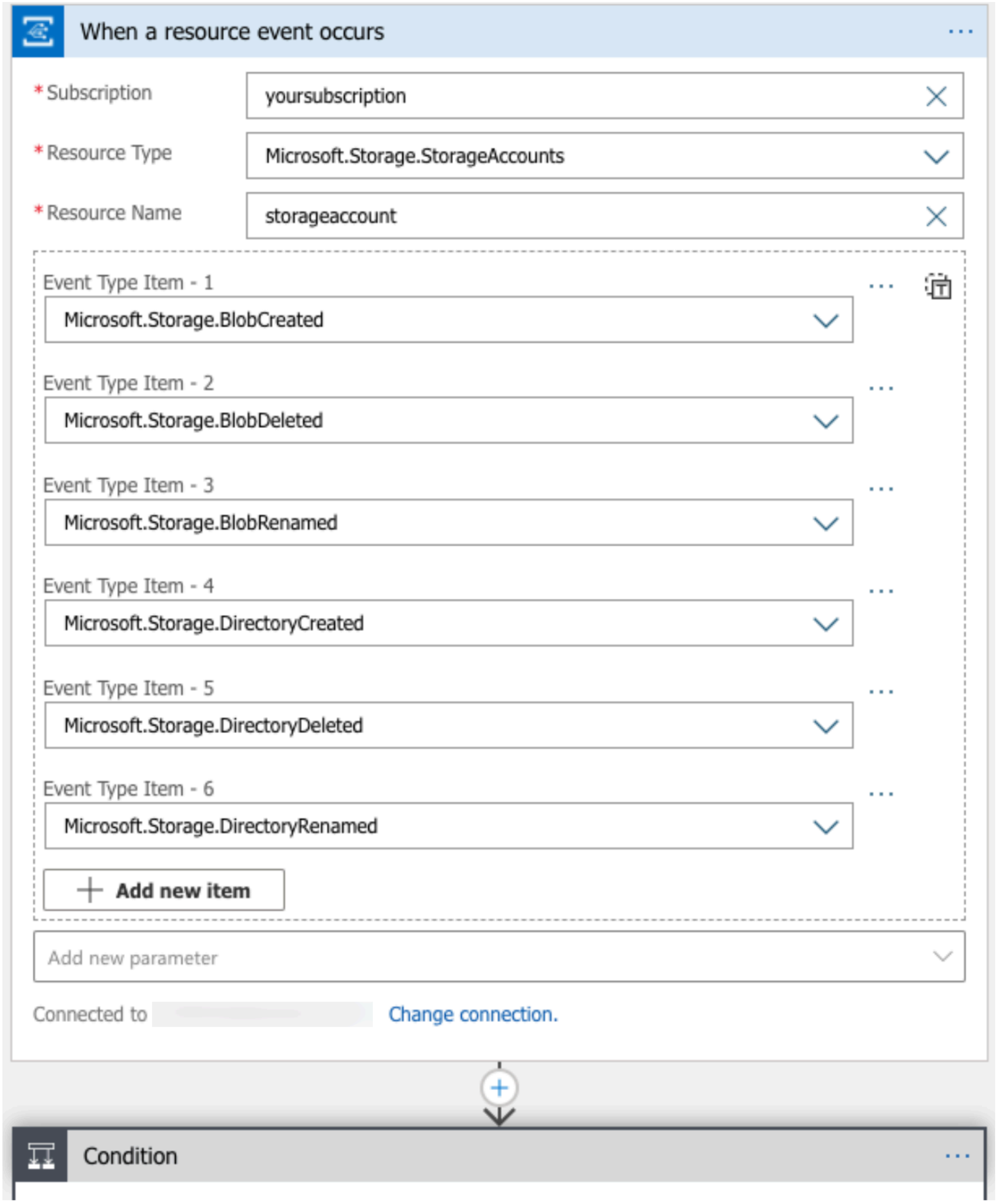
2. Click Create Azure Blob Storage and Azure Event Grid > Continue.


This logic app will connect to:

Azure Blob Storage	testblobconnection	✓
Azure Event Grid		✓

3. Choose all the events for When a resource event occurs:

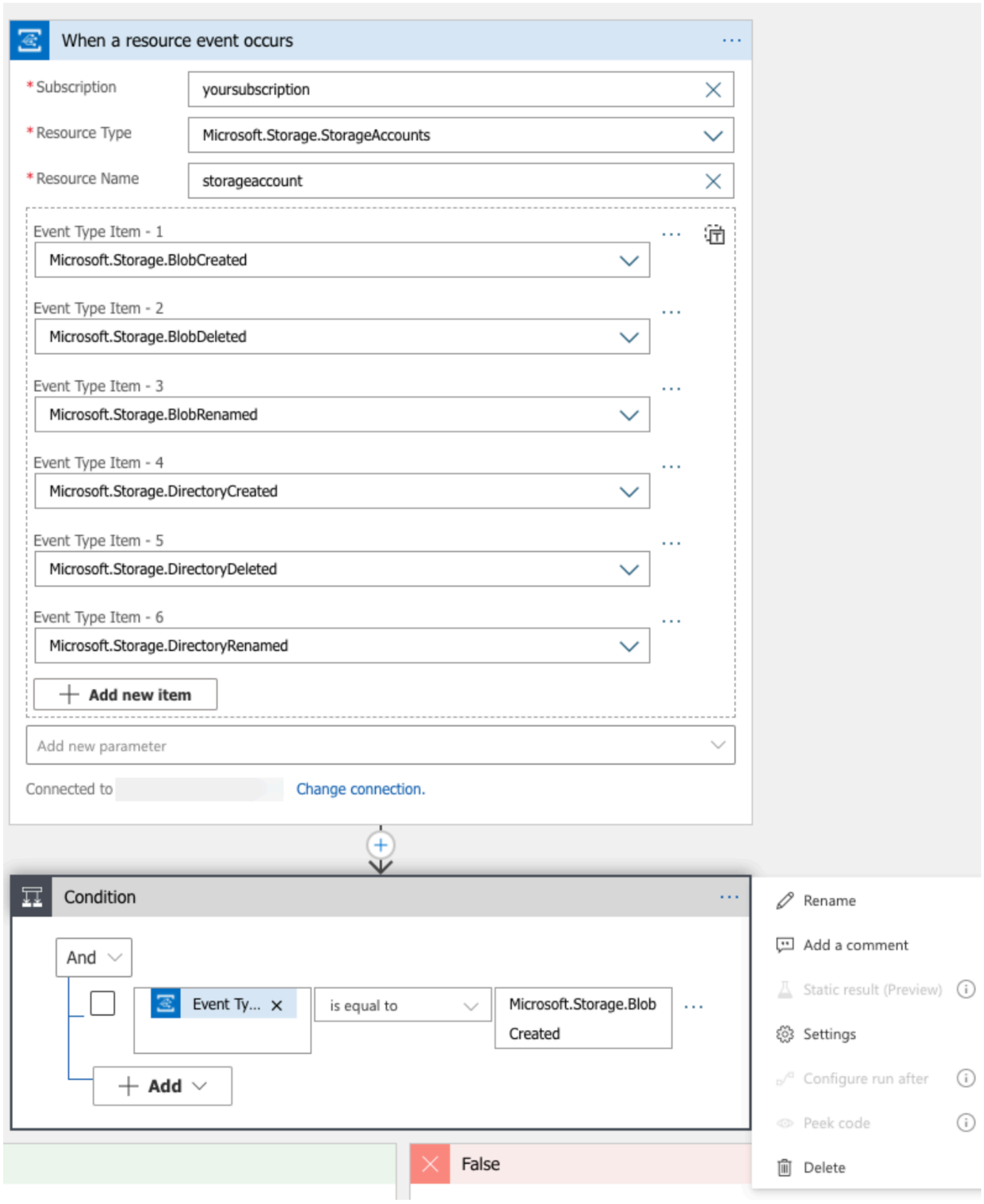
- Microsoft.Storage.BlobCreated
- Microsoft.Storage.BlobDeleted
- Microsoft.Storage.BlobRenamed
- Microsoft.Storage.DirectoryCreated
- Microsoft.Storage.DirectoryDeleted
- Microsoft.Storage.DirectoryRenamed



When a resource event occurs

*Subscription	yoursubscription
*Resource Type	Microsoft.Storage.StorageAccounts
*Resource Name	storageaccount
Event Type Item - 1	Microsoft.Storage.BlobCreated
Event Type Item - 2	Microsoft.Storage.BlobDeleted
Event Type Item - 3	Microsoft.Storage.BlobRenamed
Event Type Item - 4	Microsoft.Storage.DirectoryCreated
Event Type Item - 5	Microsoft.Storage.DirectoryDeleted
Event Type Item - 6	Microsoft.Storage.DirectoryRenamed

4. Delete the Condition block and click New step.



When a resource event occurs

* Subscription: yoursubscription
* Resource Type: Microsoft.Storage.StorageAccounts
* Resource Name: storageaccount

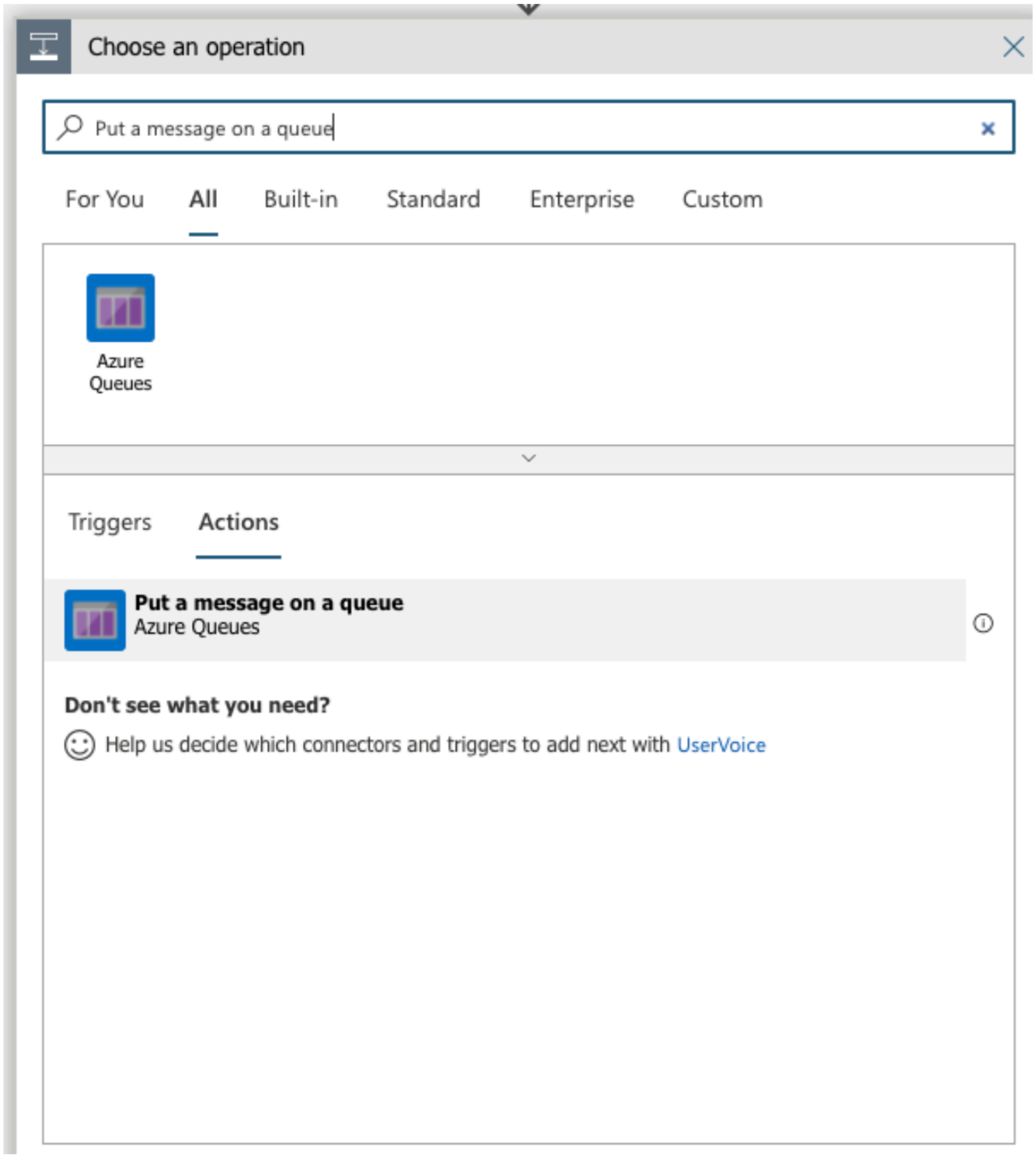
Event Type Item - 1	Microsoft.Storage.BlobCreated
Event Type Item - 2	Microsoft.Storage.BlobDeleted
Event Type Item - 3	Microsoft.Storage.BlobRenamed
Event Type Item - 4	Microsoft.Storage.DirectoryCreated
Event Type Item - 5	Microsoft.Storage.DirectoryDeleted
Event Type Item - 6	Microsoft.Storage.DirectoryRenamed

Connected to Change connection.

Condition

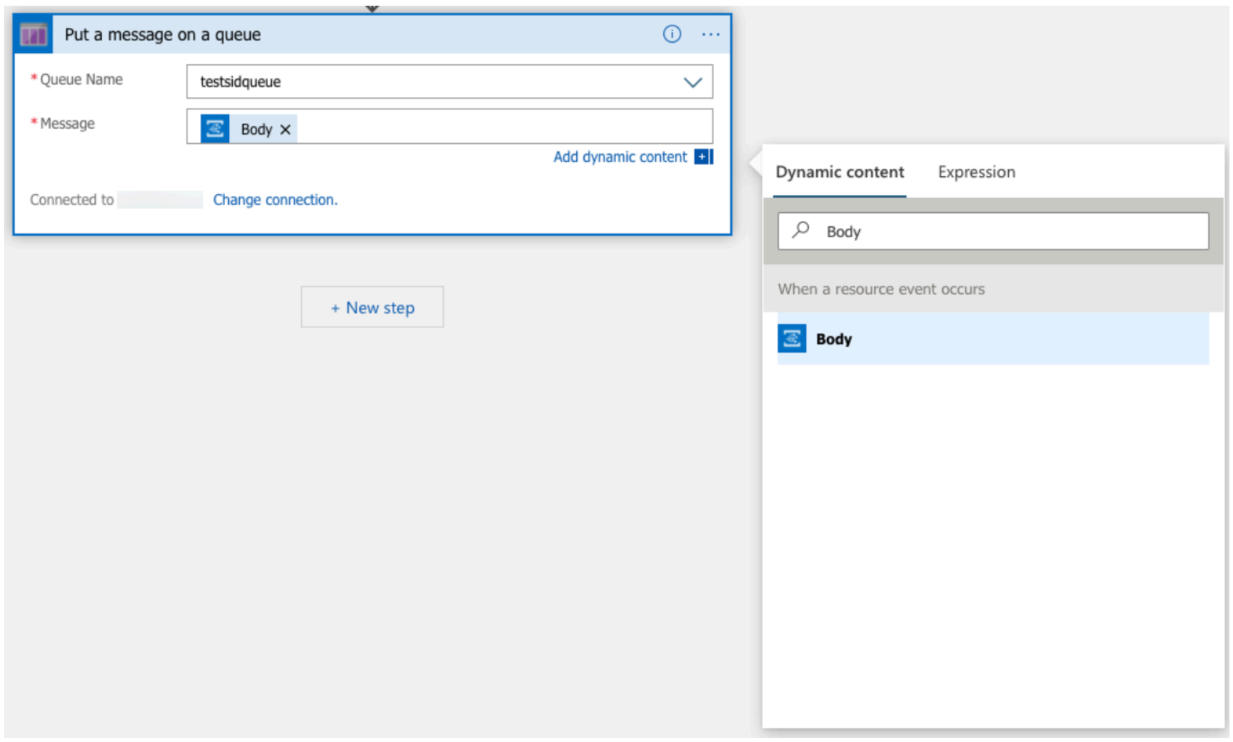
And: Event Ty... is equal to Microsoft.Storage.Blob Created

5. Under Choose an operation window search Put a message on a queue and choose the Action > Azure Queues > Put a message on a queue.



6. Choose the queue name and Messages format:

- Queue Name: testqueue. (Example)
- Message format: Body.



Queue Name: testsidqueue

Message: Body

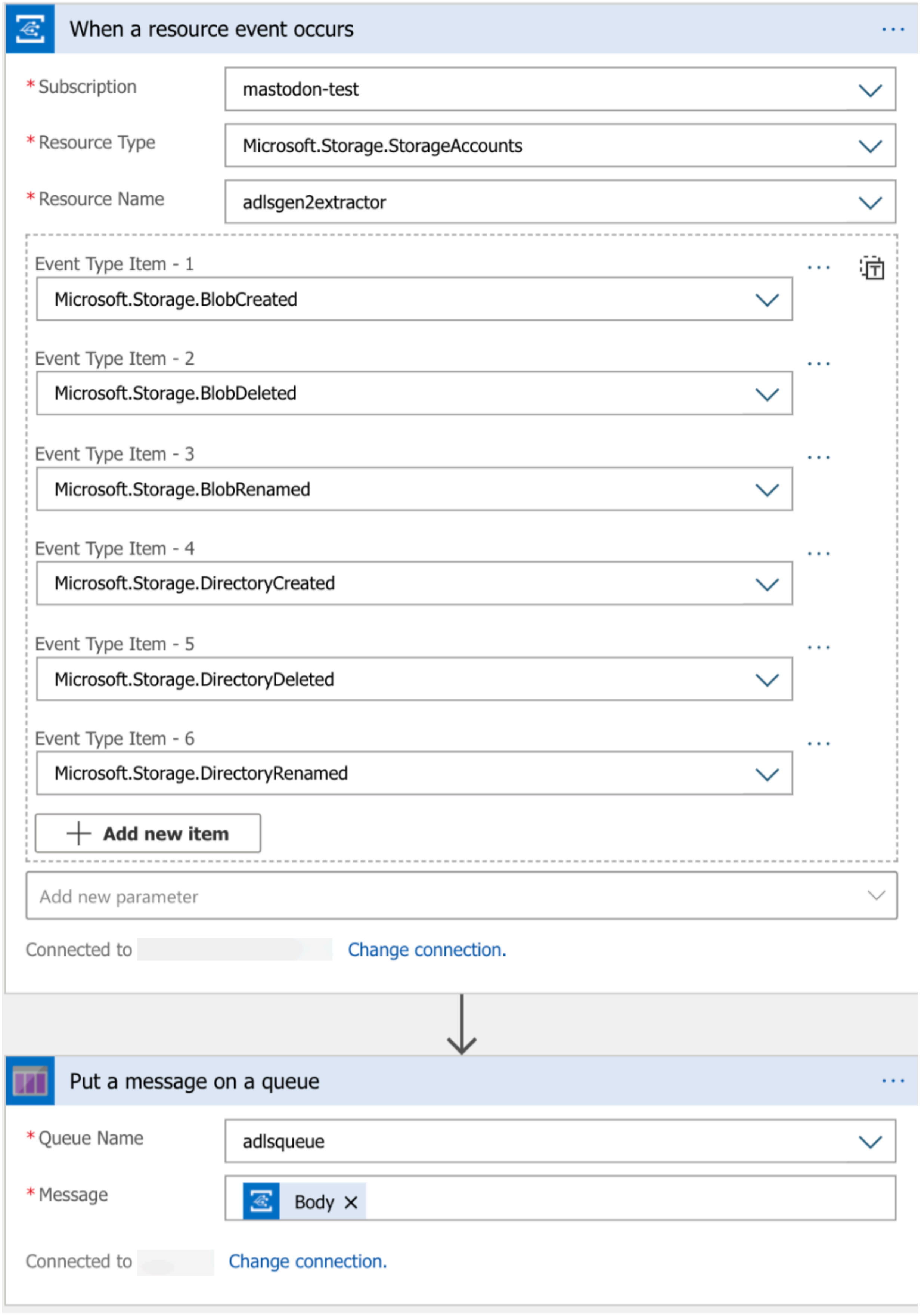
Connected to

Body

When a resource event occurs

Body

7. Click Save As and provide an appropriate name to Logic App. For example: newLogicAppForEventQueuetest.



When a resource event occurs

* Subscription	mastodon-test
* Resource Type	Microsoft.Storage.StorageAccounts
* Resource Name	adlsgen2extractor

Event Type Item - 1	Microsoft.Storage.BlobCreated
Event Type Item - 2	Microsoft.Storage.BlobDeleted
Event Type Item - 3	Microsoft.Storage.BlobRenamed
Event Type Item - 4	Microsoft.Storage.DirectoryCreated
Event Type Item - 5	Microsoft.Storage.DirectoryDeleted
Event Type Item - 6	Microsoft.Storage.DirectoryRenamed

Put a message on a queue

* Queue Name	adlsqueue
* Message	Body

8. Click Create.

Setting up Azure managed Identity for Extraction

Once you have created and configured the storage account and setup the queue (for incremental extraction only), you must create the managed identities and later assign roles with specific scopes to these identities.

You can reuse the same resource group that you created for the storage account or you can create a new resource group that can act as a logical grouping of managed identities.

Note: It is assumed that the minimal secure setup, where the managed identity will have at least one role assigned.

Creating Managed Identity

You must create a new managed identity or use the existing one based on the requirement.

1. On the Azure portal > Azure services homepage select Managed Identities > Create.

   The Create User Assigned Managed Identity page is displayed.
2. Under the Basics tab, provide information for the following:

   - Subscription
   - Resource Group (You can optionally create a new Resource Group as well)

     - Region
     - Name (Specify the name of a managed identity). For example, testatlasmanagedidentity
3. Click Review + Create.
4. Review the entered information and click Create.

   The deployment is initialized and submitted. Later, it is successfully deployed to the selected Resource Group.

Assigning Roles for the Managed Identities

Once you have created the Managed Identities, assign the roles with specific scoopes to the identity.

Procedure

1. In ADLS > Storage Accounts > Your storage account Access Control (IAM) > Add role assignments.
2. Based on the type of extraction selected, you can add the type of role:

   For Bulk extraction, you can add the following:

   - Click Add > Add role assignment
   - - Role > Select Storage Blob Data Contributor.
     - Assign Access to > Select User assigned managed identity.

    - Subscription > As applicable.
- Click Save.

For Incremental extraction, you can add the following:

- Click Add > Add role assignment
    - Role > Storage Queue Data Contributor.
    - Assign Access to > User assigned managed identity.
    - Subscription > As applicable.
- Click Save.

Mapping Atlas Identity to CDP users

In order to use the created Managed Identity, a Atlas user must be mapped to the managed identity.

The option to add or modify these mappings is available in the Cloudera Manager UI.

1. Select Management Console under Environments > click on an environment > Actions > Manage Access > IDBroker Mappings > Edit.
2. Under IDBroker Mappings, you can change the mappings of the Atlas user to managed identity. The user or group dropdown is prepopulated with CDP users and groups. On the right hand side, specify the Resource ID (copied from the Azure Portal) for the Atlas user.

For example:

1. Cloudera Manager > Configuration > Search for the user role and add Knox IDBroker Azure User Mapping information.
2. Restart the Knox service.

Note: If a user is mapped to multiple roles via group membership, the specific role to be used needs to be provided at runtime. If the user is mapped directly to a role, the direct mapping takes precedence over mapping via group membership. For information on how to specify the role, refer to Specifying a group when user belongs to multiple groups .

Running ADLS Metadata Extractor

You must note about how you can run the ADLS Metadata Extractor.

The extractor script is located at:

/opt/cloudera/parcels/CDH/lib/atlas/extractors/bin/adls-extractor.sh

The configuration is located at:

/opt/cloudera/parcels/CDH/lib/atlas/extractors/conf/adls.conf

Note: The extractor configuration file includes properties that require ADLS Gen2 configuration values. Be sure to set those values in the configuration file before running the command.

Before running the extraction script, you must log in to Atlas using “atlas” user or a “root” user.

Attention: Ensure that a valid Atlas Kerberos session is in place before running the script.

For more information, see Extraction Configuration on page 15.

Running Bulk Extraction

Bulk extraction mechanism employs direct calls on Azure API calls to fetch all blobs and containers in ADLS Gen2 storage account.

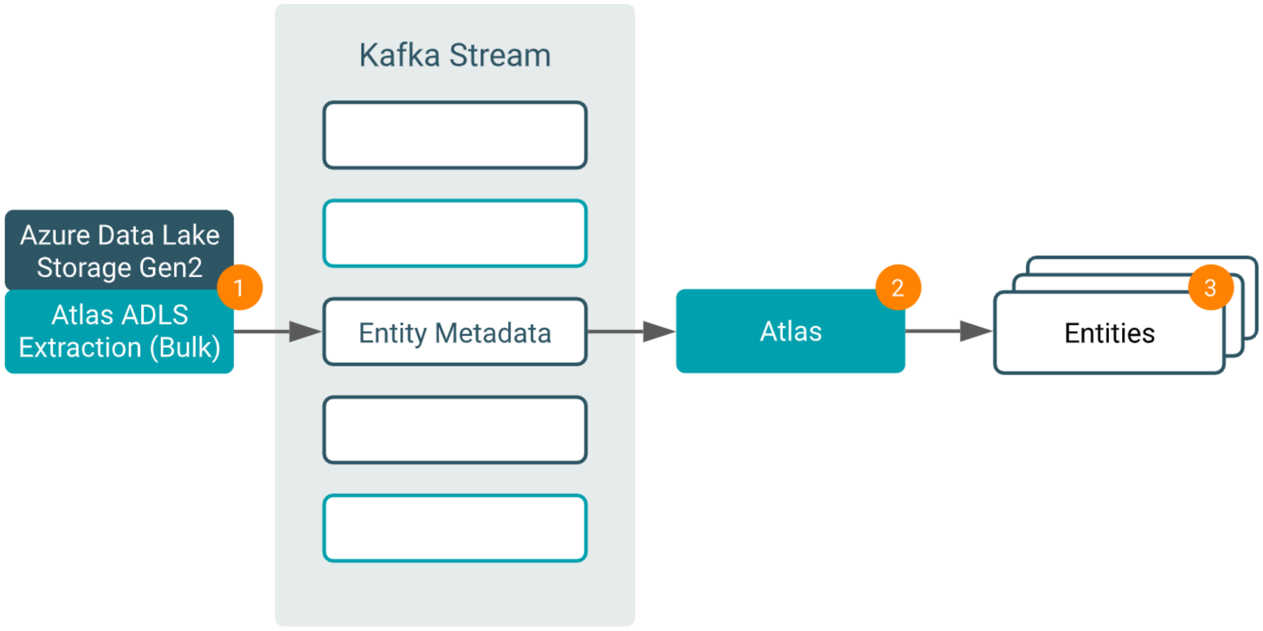
If there is a failure while extracting the complete Azure metadata, the bulk extraction must be resumed from the last checkpoint by changing atlas.adls.extraction.resume.from.progress.file=true configuration at adls.conf.

The following command line example runs the bulk extraction. Assuming the mandatory properties are set in the default configuration file, only the parameter to enable bulk mode is required:

/opt/cloudera/parcels/CDH/lib/atlas/extractors/bin/adls-extractor.sh

Or

/opt/cloudera/parcels/CDH/lib/atlas/extractors/bin/adls-extractor.sh -e BULK



Refer to Extraction Configuration on page 15for more details on different optional configurations.

Running Incremental Extraction

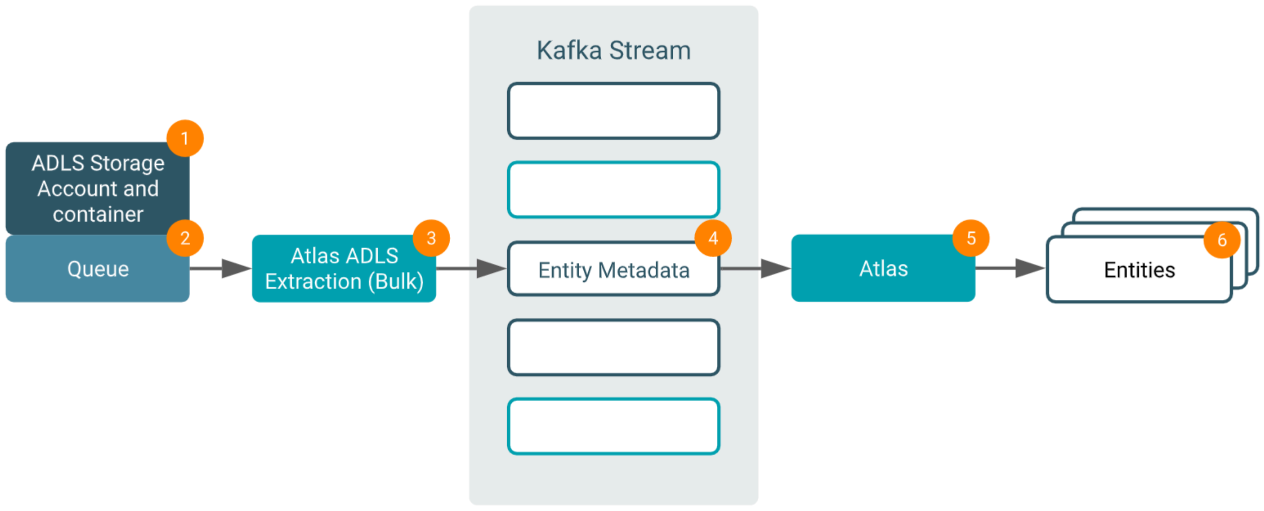
With incremental extraction, you can keep Atlas updated with changes happening within ADLS.

You must configure the queue for the storage account so that extraction receives all blob, create, delete, and rename events.

In addition, you can optionally set the timeout or retry time so the extractor does not keep running even when it is not able to connect to the queue.

The following command line example runs the incremental extraction. Assuming the mandatory properties are set in the default configuration file, only the parameter to enable incremental mode is required:

```
/opt/cloudera/parcels/CDH/lib/atlas/extractors/bin/adls-extractor.sh -e INCR
EMENTAL
```


Kafka Stream
ADLS Storage Account and container
1
Queue
2
Atlas ADLS Extraction (Bulk)
3
Entity Metadata
4
Atlas
5
Entities
6

You must plan whether to set up a repeating job to run incremental extraction. Assuming you have a lot of changes that you want to track.

Command-line options to run Extraction

You can use the command-line options to run the Atlas ADLS metadata extraction.

Command-line option	Parameters	Default Value
-c, or--config	The path of the configuration file containing the extraction type specific configurations parameters.	By default it considers $ATLAS_HOME/ extractors/conf/adls.conf. This can be changed using this parameter.
-e,--extraction <arg>	The type of extraction to be done. Two values allowed Incremental or Bulk.	Bulk
-f, -failOnError	Setting this option causes extraction to stop when an error occurs. Not setting this option, will cause the exception to be logged and operation to proceed.	false

Note: Progress of the extraction can be monitored via the log file /var/log/atlas/azure-adls-extractor.log

Extraction Configuration

Some of the configurations that must be set-up before you perform the metadata extraction.

The configuration file is located at: /opt/cloudera/parcels/CDH/lib/atlas/extractor/adls.conf

Configuration Parameter	Purpoose	Default Value
atlas.adls.extraction.account.name	ADLS Gen2 storage account which was created as part of Extraction Prerequisites on page 4.	Mandatory.
atlas.adls.extraction.account.key	ADLS account key if IDBroker is not configured.	To be specified if Knox IDBroker is not configured at CDP.

Configuration Parameter	Purpoose	Default Value
atlas.adls.extraction.access.token	Access token for token based authentication.	If Knox IDBroker is not configured at CDP, token based authentication is required. It must be configured.
atlas.adls.extraction.allowlist.paths=abfs://<containername>@<accountname>.dfs.core.windows.net/<path>	Comma separated ABFS paths or patterns from which ADLS metadata (directory, blob) needs to be extracted. Multiple values can be configured by ',' separated. Example: abfs://testcontainer@teststorageaccount.dfs.core.windows.net/testdir1/	
atlas.adls.extraction.denylist.paths=abfs://<containername>@<accountname>.dfs.core.windows.net/<path>	Comma separated ABFS paths or patterns from which ADLS metadata should be excluded from extraction. Multiple values can be configured by ',' separated. Example: abfs://testcontainer@teststorageaccount.dfs.core.windows.net/testdir2/	
atlas.adls.extraction.max.blob.per.call	Number of blob storage to be fetched in one call to Azure ADLS by bulk extraction.	1000
atlas.adls.extraction.timeout.per.call.in.sec	The timeout (seconds), used for each ADLS SDK call wherever it is required.	30
atlas.adls.extraction.resume.from.progress.file	Resume from the last run in case of failure feature.	Set to false by default. Set it to true if resuming an extract.
atlas.adls.extraction.progress.file	Progress file used for extraction in case the user wants to resume.	adls_extractor_progress_file.props
atlas.adls.extraction.max.reconnect.count	Specify the maximum number of retries to: • IDBroker in case of credentials expiry. • Retryable exception for Azure ADLS.	
atlas.adls.extraction.fs.system	File System used in Azure ADLS.	Default set to: abfs
atlas.adls.extraction.incremental.queueNames	Azure list of Account:QueueName which is configured as part of Configuring ADLS Gen2 Storage Queue on page 5 to get the blob/directory create, delete events. Example: teststorageaccount:testqueue	
atlas.adls.extraction.incremental.messagesPerRequest	The number of messages Incremental Extractor tries to fetch from ADLS Queue in a single call.	Default is 10. It ranges from 1 to 32.
atlas.adls.extraction.incremental.requestWaitTime	The wait time in seconds in a single call to ADLS Queue to fetch atlas.adls.extraction.incremental.messagesPerRequest messages.	20
atlas.adls.extraction.incremental.max.retry	Maximum retry count in case of Idle while reading Queue Messages in Incremental Extraction.	20
atlas.adls.extraction.incremental.delete.needed.for.rename	Does an entity need deletion if it has been renamed to something which should not be created at Atlas due to allow and deny list.	false
atlas.notification.hook.asynchronous	This setting should be set to "true" only when extracting a large number of adls metadata (directory, blob) where there is a possibility of a lag when publishing messages to ATLAS_HOOK Kafka topic.	Defaults to asynchronous sending of events: true To set synchronous sending of events: false (Synchronous)

Verifying Atlas for the extracted data

Once the metadata extraction is completed, you can verify the new types of data that are added to Atlas.

Data Type	Description
adls_gen2_account	Represents a StorageAccount (not the Azure Account).
adls_gen2_container	Represents the container
adls_gen2_directory	Represents the directory.
adls_gen2_blob	Represents a blob or file with the name.

Note: From Atlas' web UI, use Basic Search to search for entities added to Atlas as part of the extraction.

Resources for on-boarding Azure for CDP users

Refer to the following sections before you commence the Atlas ADLS Extraction feature.

- Introduction to Azure Data Lake Storage Gen2
- Azure quick start
- Azure Subscription Requirements
- Register an Azure Environment
- Creating ADLS Gen2 storage account and containers
- Setting up ADLS Gen2 and Managed Identities