Code-Free ETL as a service

Ingest

- Multi-cloud and on-premise hybrid copy data
- 100+ native connectors
- Serverless and auto-scale
- Use wizard for quick copy jobs

Control Flow

- Design code-free data pipelines
- Generate pipelines via SDK
- Utilize workflow constructs: loops, branches, conditional execution, variables, parameters, ...

Data Flow

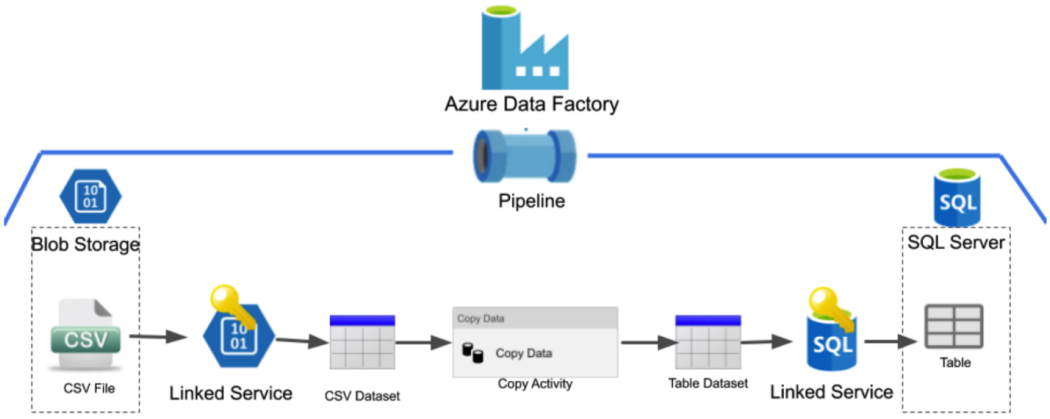
- Code-free data transformations that execute in Spark
- Scale-out with Azure Integration Runtimes
- Generate data flows via SDK
- Designers for data engineers and data analysts

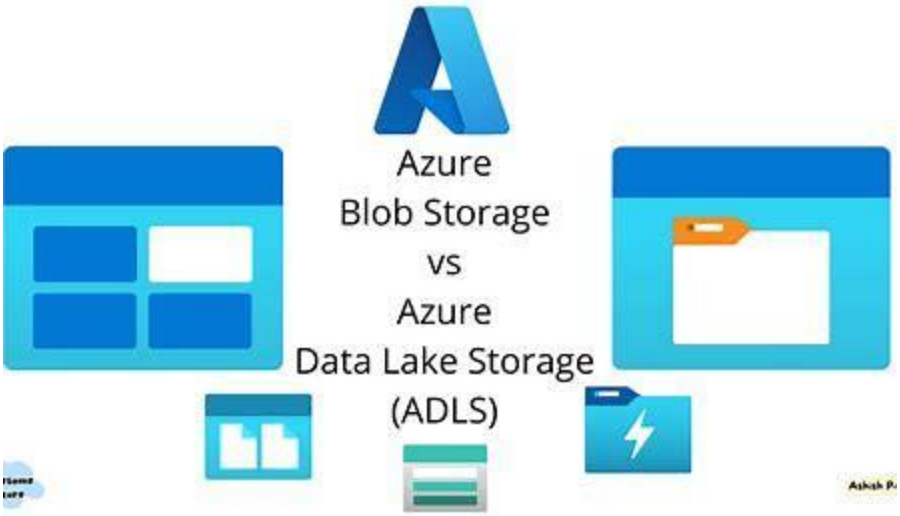
Schedule

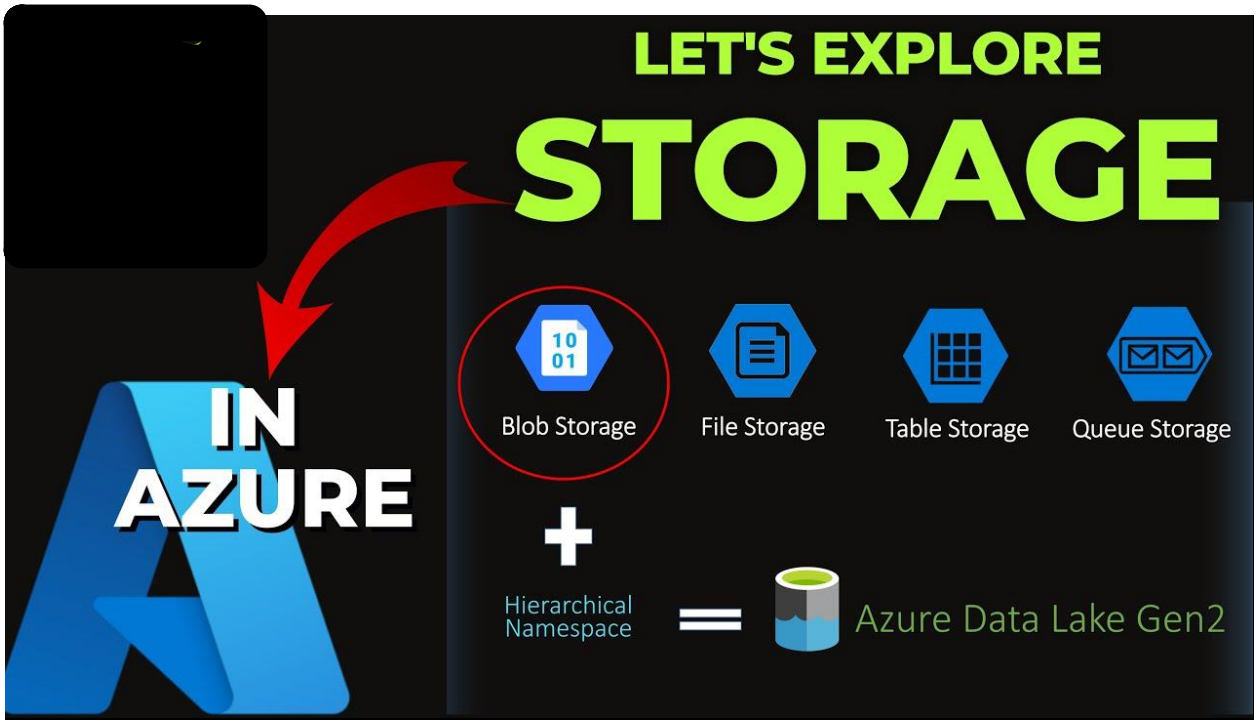
- Build and maintain operational schedules for your data pipelines
- Wall clock, event-based, tumbling windows, chained

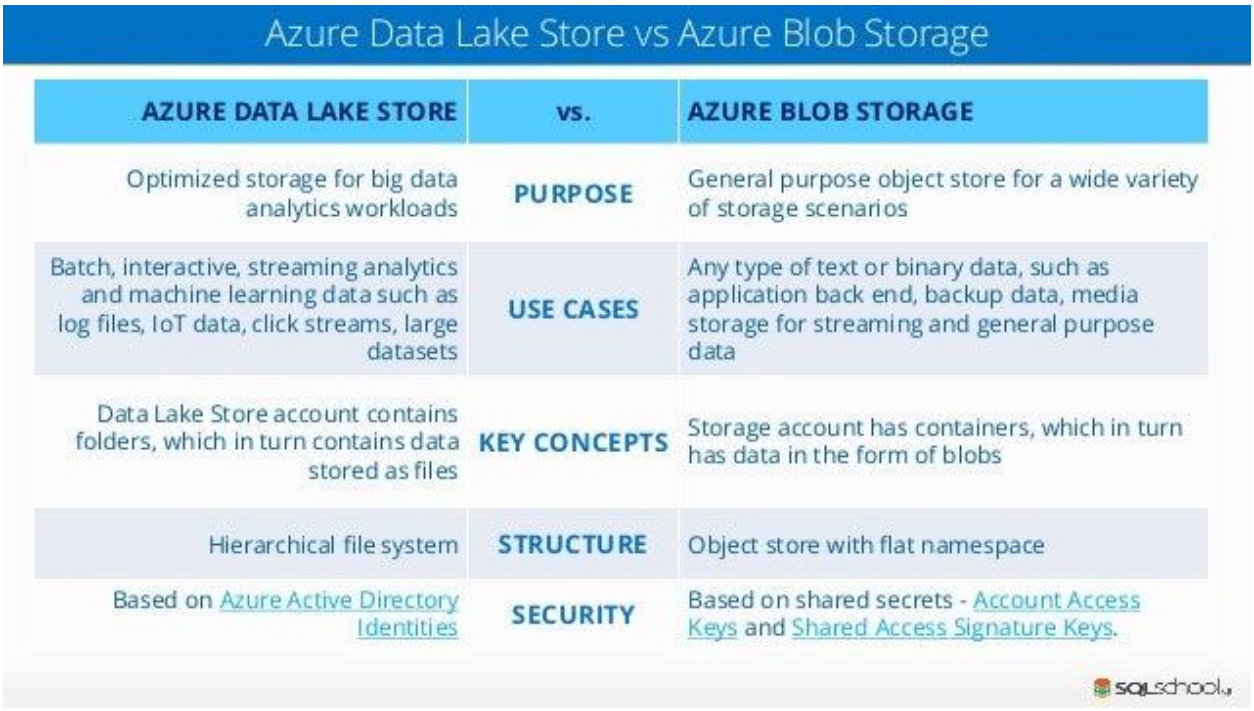
Monitor

- View active executions and pipeline history
- Detail activity and data flow executions
- Establish alerts and notifications




Azure
Blob Storage
vs
Azure
Data Lake Storage
(ADLS)


LET'S EXPLORE
STORAGE
IN
AZURE
Blob Storage
File Storage
Table Storage
Queue Storage
+
Hierarchical
Namespace
=
Azure Data Lake Gen2



Comparison Table

Feature	Azure Data Lake Storage Gen2 (ADLS Gen2)	Azure Blob Storage
Data Organization	Hierarchical namespace (directories and files)	Flat namespace (containers and blobs)
Performance	Optimized for high-performance analytics and large-scale data processing	General-purpose with high availability and scalability
Scalability	Designed to handle petabytes of data with high throughput	Scalable to handle massive amounts of unstructured data
Security	Advanced security features like fine-grained access control and encryption	Basic security with encryption at rest and in transit
Pricing	Typically higher cost due to advanced features and performance	Cost-effective with various pricing tiers for different needs
Use Cases	Big Data Analytics, Data Lakes, Machine Learning Data Storage	Backup and Restore, Archiving, Content Distribution and Streaming
Integration	Seamlessly integrates with Azure Synapse, Azure Databricks, and more	Integrates with a wide range of Azure services and external tools

Step-by-Step Guide to Create an ADLS Gen2 Account

1. Sign In to Azure Portal

- Go to the Azure Portal.
- Sign in with your Azure credentials.

2. Navigate to the Storage Accounts

- In the Azure portal, select **"Storage accounts"** from the left-hand menu. If you don't see it, use the search bar at the top to search for "Storage accounts".

3. Create a New Storage Account

- Click on **"+ Create"** or **"Add"** to start the process of creating a new storage account.

4. Configure Basic Settings

- **Subscription**: Select the Azure subscription under which you want to create the storage account.
- **Resource Group**: Choose an existing resource group or create a new one to organize your resources.
- **Storage Account Name**: Enter a unique name for your storage account. The name must be globally unique and adhere to Azure's naming conventions.
- **Region**: Choose the region where you want to deploy your storage account. It's recommended to select a region close to your users or data.

5. Select Storage Account Type

- **Performance**: Choose between **Standard** (HDD) and **Premium** (SSD). ADLS Gen2 is available with Standard performance.
- **Replication**: Choose the replication strategy that fits your needs (e.g., **LRS** - Locally Redundant Storage, **GRS** - Geo-Redundant Storage).
- **Storage Account Kind**: Ensure that you select **StorageV2 (general-purpose v2)**. ADLS Gen2 features are available in this kind.

6. Enable Hierarchical Namespace

- Under the **"Advanced"** tab, look for the **"Hierarchical namespace"** setting.

- **Enable** the hierarchical namespace by toggling the switch. This setting is essential for using ADLS Gen2 features like directory and file-level access control.

7. Configure Networking (Optional)

- If you need to restrict access to your storage account, configure the network settings by choosing **“Networking”** and setting up access controls, such as Virtual Network and IP firewall rules.

8. Set Up Data Protection (Optional)

- Configure **data protection** options such as soft delete and change feed if required. These settings help protect your data from accidental deletions and track changes.

9. Review + Create

- Review all the settings you’ve configured in the previous steps.
- Click on **“Create”** to start the deployment of your storage account.

10. Access Your ADLS Gen2 Account

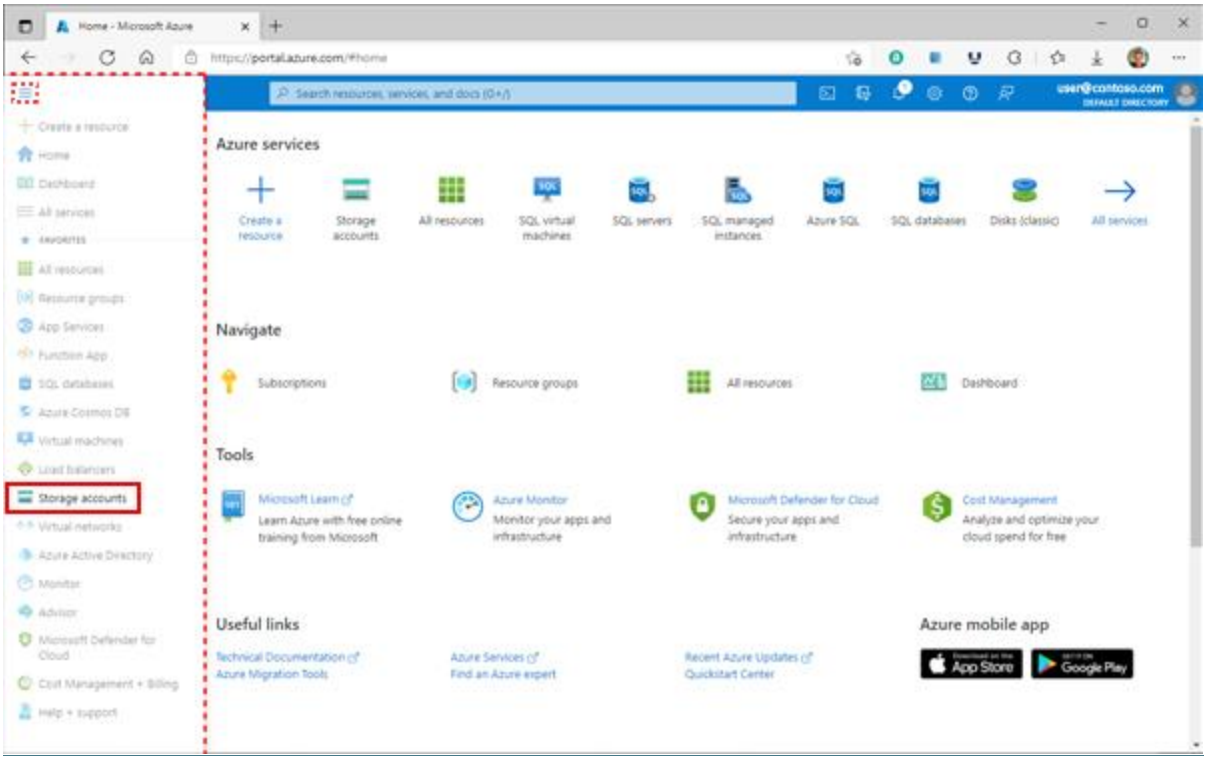
- Once the deployment is complete, navigate to **"Storage accounts"** in the Azure portal.
- Select your newly created storage account from the list.
- You can now use the **“Containers”** section to create and manage data containers or use other features like Access Control and Storage Explorer.

Additional Tips

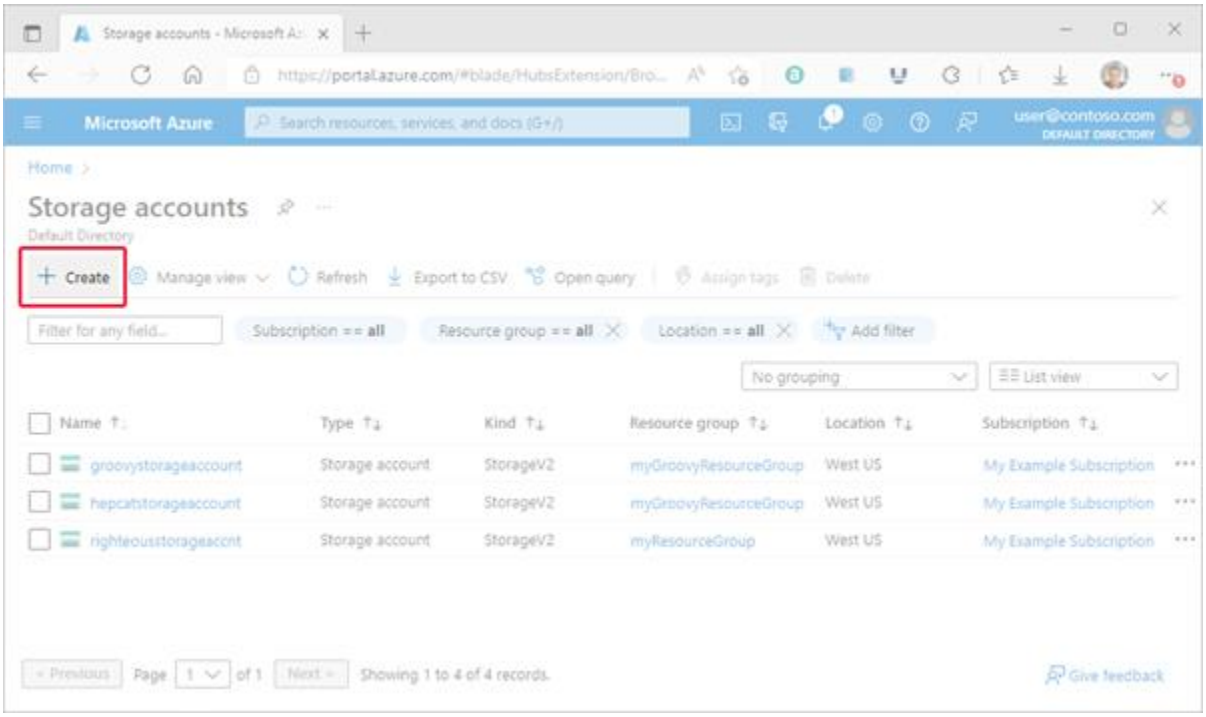
- **Access Control**: Configure access control policies to manage permissions and security for your ADLS Gen2 account.
- **Data Explorer**: Use Azure Storage Explorer or similar tools to interact with your ADLS Gen2 account and manage files and directories.
- **Integration**: ADLS Gen2 integrates well with Azure Synapse Analytics, Azure Databricks, and other Azure services for advanced data processing and analytics.

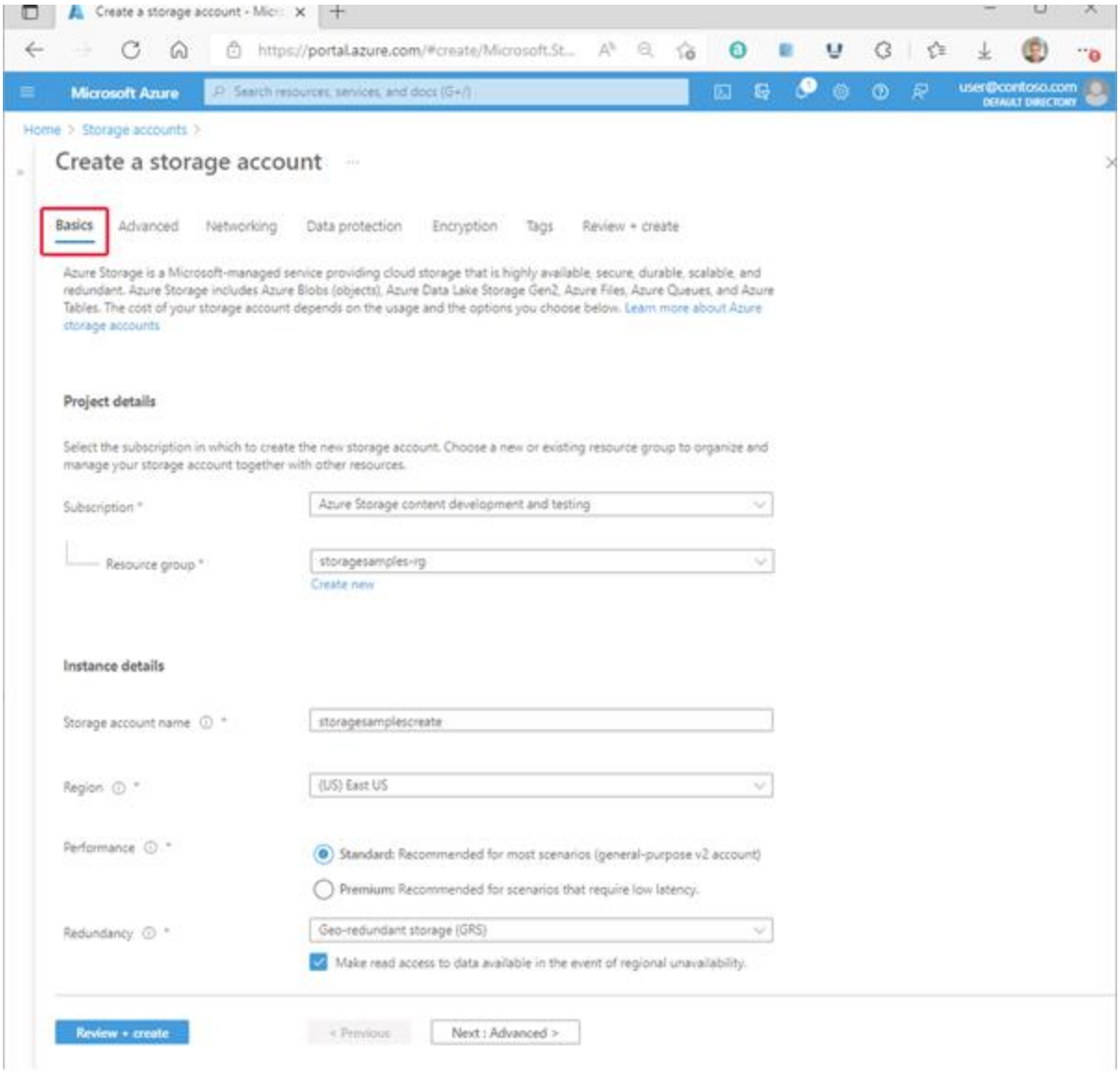
By following these steps, you should be able to set up an Azure Data Lake Storage Gen2 account and start leveraging its capabilities for scalable data storage and management.

1.



2. On the **Storage accounts** page, select **Create**.




Create a storage account - Micr
https://portal.azure.com/#create/Microsoft.St...
Microsoft Azure
Search resources, services, and docs (G+/)
user@contoso.com
DEFAULT DIRECTORY
Home > Storage accounts >
Create a storage account
Basics
Advanced
Networking
Data protection
Encryption
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 about Azure storage accounts
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 Storage content development and testing
Resource group *
storagesamples-rg
Create new
Instance details
Storage account name *
storagesamplescreate
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.
Review + create
< Previous
Next : Advanced >

All services > Storage accounts >

Create a storage account ...

Basics | **Advanced** | Networking | Data protection | Encryption | Tags | Review

Certain options have been disabled by default due to the combination of storage account performance, redundancy, and region.

Security

Configure security settings that impact your storage account.

Require secure transfer for REST API operations ☑

Allow enabling anonymous access on individual containers ☐

Enable storage account key access ☑

Default to Azure Active Directory authorization in the Azure portal ☐

Minimum TLS version: Version 1.2

Permitted scope for copy operations (preview): From any storage account

Hierarchical Namespace

Hierarchical namespace, complemented by Data Lake Storage Gen2 endpoint, enables file and directory semantics, accelerates big data analytics workloads, and enables access control lists (ACLs) Learn more

Enable hierarchical namespace ☐

Access protocols

Blob and Data Lake Gen2 endpoints are provisioned by default Learn more

Enable SFTP ☐

To enable SFTP, 'hierarchical namespace' must be enabled.

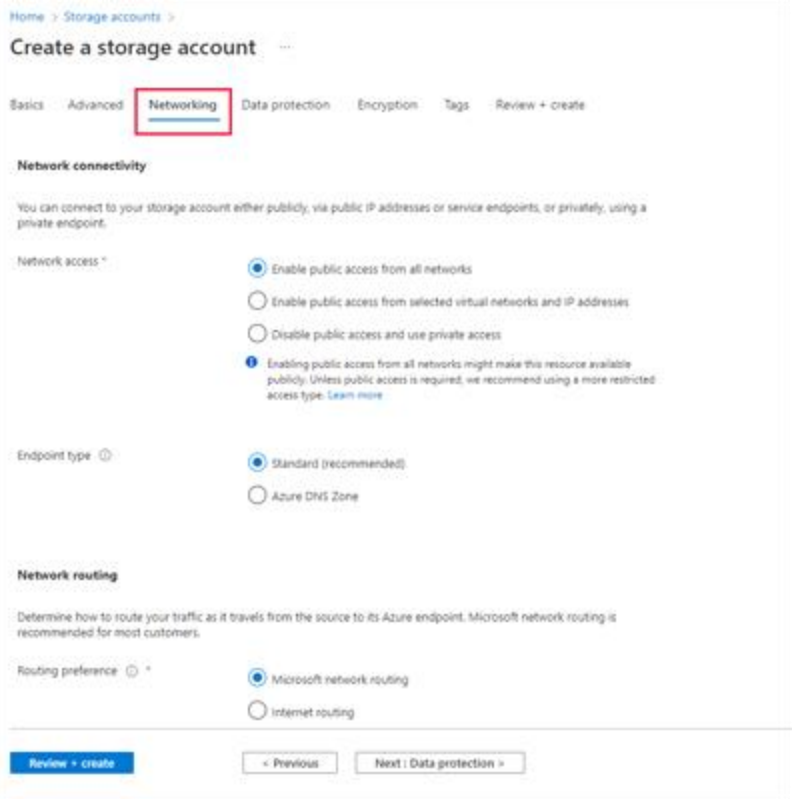
Enable network file system v3 ☐

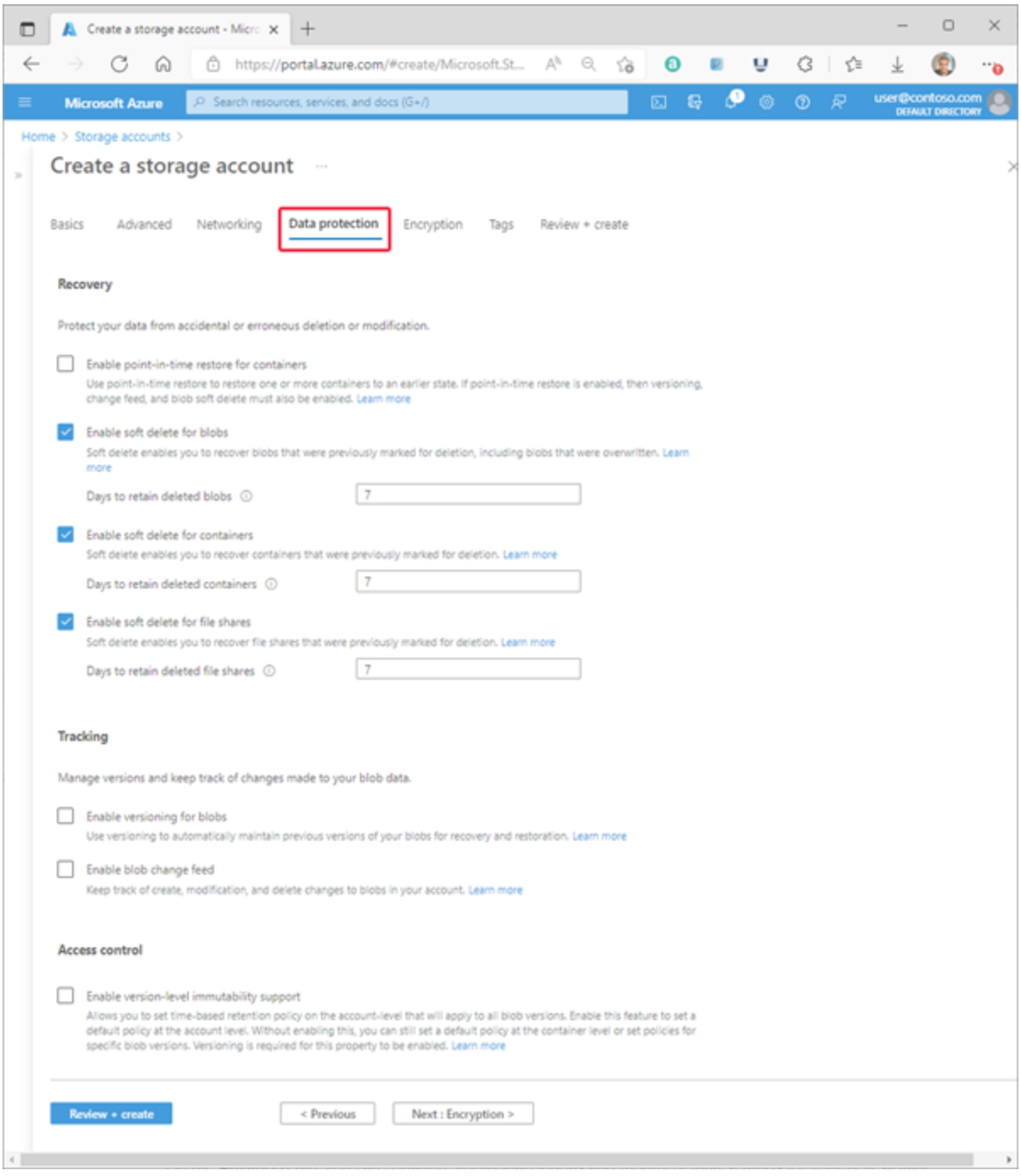
To enable NFS v3 'hierarchical namespace' must be enabled. Learn more about NFS v3

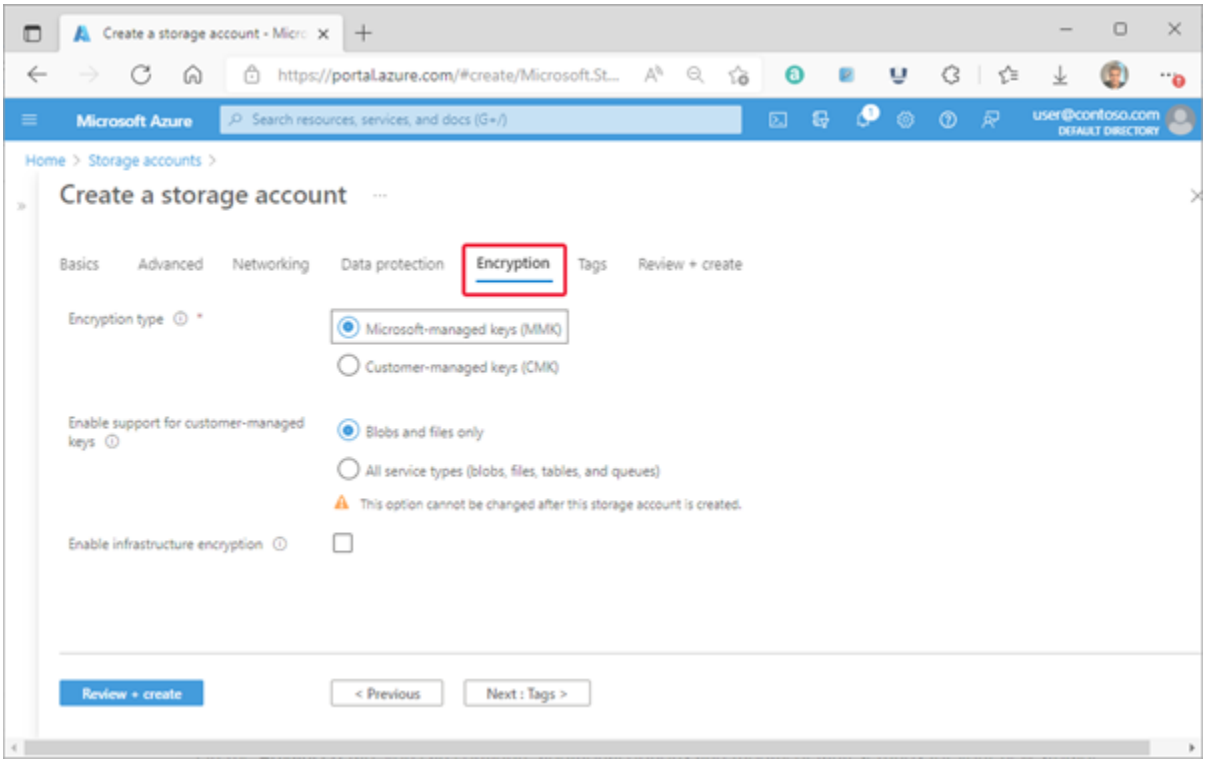
Blob storage

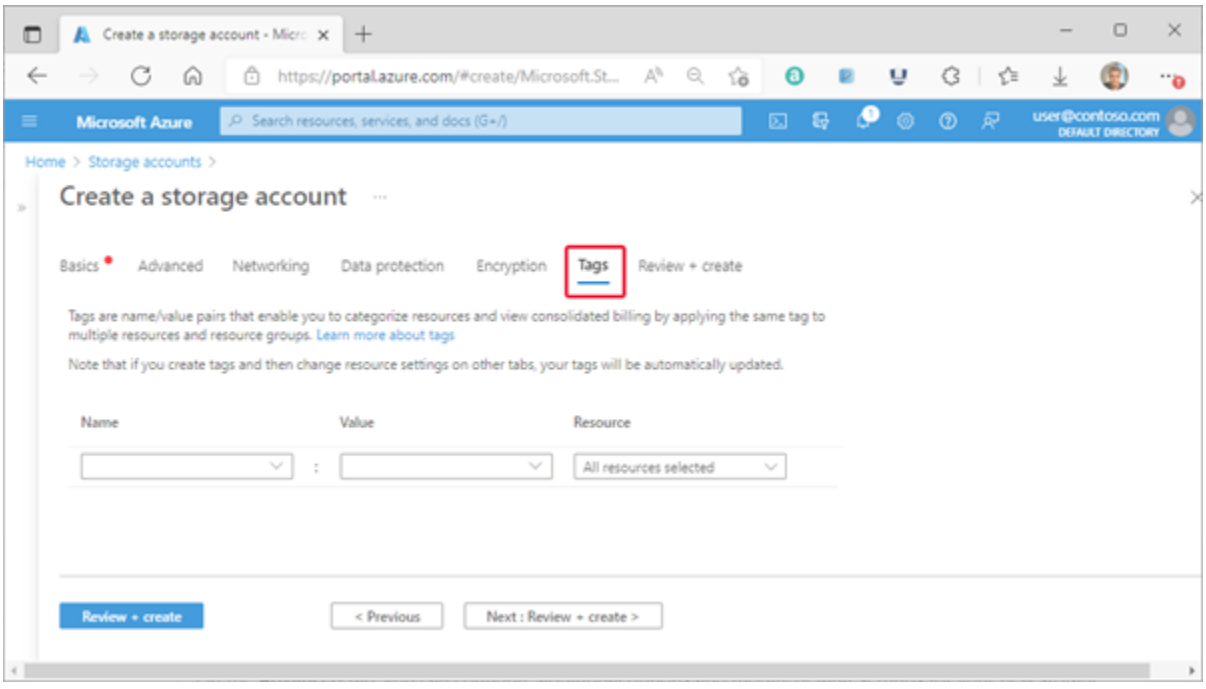
Allow cross-tenant replication ☐

Review | < Previous | Next : Networking >


Home > Storage accounts >
Create a storage account
Basics
Advanced
Networking
Data protection
Encryption
Tags
Review + create
Network connectivity
You can connect to your storage account either publicly, via public IP addresses or service endpoints, or privately, using a private endpoint.
Network access *
Enable public access from all networks
Enable public access from selected virtual networks and IP addresses
Disable public access and use private access
Enabling public access from all networks might make this resource available publicly. Unless public access is required, we recommend using a more restricted access type. Learn more
Endpoint type
Standard (recommended)
Azure DNS Zone
Network routing
Determine how to route your traffic as it travels from the source to its Azure endpoint. Microsoft network routing is recommended for most customers.
Routing preference *
Microsoft network routing
Internet routing
Review + create
< Previous
Next : Data protection >


Create a storage account - Micro
https://portal.azure.com/#create/Microsoft.St...
Microsoft Azure
Search resources, services, and docs (G+/)
user@contoso.com
DEFAULT DIRECTORY
Home > Storage accounts >
Create a storage account
Basics
Advanced
Networking
Data protection
Encryption
Tags
Review + create
Recovery
Protect your data from accidental or erroneous deletion or modification.
Enable point-in-time restore for containers
Use point-in-time restore to restore one or more containers to an earlier state. If point-in-time restore is enabled, then versioning, change feed, and blob soft delete must also be enabled. Learn more
Enable soft delete for blobs
Soft delete enables you to recover blobs that were previously marked for deletion, including blobs that were overwritten. Learn more
Days to retain deleted blobs
7
Enable soft delete for containers
Soft delete enables you to recover containers that were previously marked for deletion. Learn more
Days to retain deleted containers
7
Enable soft delete for file shares
Soft delete enables you to recover file shares that were previously marked for deletion. Learn more
Days to retain deleted file shares
7
Tracking
Manage versions and keep track of changes made to your blob data.
Enable versioning for blobs
Use versioning to automatically maintain previous versions of your blobs for recovery and restoration. Learn more
Enable blob change feed
Keep track of create, modification, and delete changes to blobs in your account. Learn more
Access control
Enable version-level immutability support
Allows you to set time-based retention policy on the account-level that will apply to all blob versions. Enable this feature to set a default policy at the account level. Without enabling this, you can still set a default policy at the container level or set policies for specific blob versions. Versioning is required for this property to be enabled. Learn more
Review + create
< Previous
Next : Encryption >


Create a storage account - Micr
https://portal.azure.com/#create/Microsoft.St...
Microsoft Azure
Search resources, services, and docs (G+/)
user@contoso.com
DEFAULT DIRECTORY
Home > Storage accounts >
Create a storage account
Basics
Advanced
Networking
Data protection
Encryption
Tags
Review + create
Encryption type
Microsoft-managed keys (MMK)
Customer-managed keys (CMK)
Enable support for customer-managed keys
Blobs and files only
All service types (blobs, files, tables, and queues)
This option cannot be changed after this storage account is created.
Enable infrastructure encryption
Review + create
< Previous
Next : Tags >


Create a storage account - Micr
https://portal.azure.com/#create/Microsoft.St...
Microsoft Azure
Search resources, services, and docs (G+/)
user@contoso.com
DEFAULT DIRECTORY
Home > Storage accounts >
Create a storage account
Basics
Advanced
Networking
Data protection
Encryption
Tags
Review + create
Tags are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources and resource groups. Learn more about tags
Note that if you create tags and then change resource settings on other tabs, your tags will be automatically updated.
Name
Value
Resource
All resources selected
Review + create
< Previous
Next : Review + create >

All services > Storage accounts >

Create a storage account ...

Basics | Advanced | Networking | Data protection | Encryption | Tags | Review

Basics

Subscription	Visual Studio Enterprise Subscription
Resource Group	AkashTestRG
Location	eastus
Storage account name	disabledaccounttest
Deployment model	Resource manager
Performance	Standard
Replication	Read-access geo-redundant storage (RA-GRS)

Advanced

Enable hierarchical namespace	Disabled
Enable network file system v3	Disabled
Allow cross-tenant replication	Disabled
Access tier	Hot
Enable SFTP	Disabled
Large file shares	Disabled

Networking

Network connectivity	Public endpoint (all networks)
Default routing tier	Microsoft network routing
Endpoint type	Standard

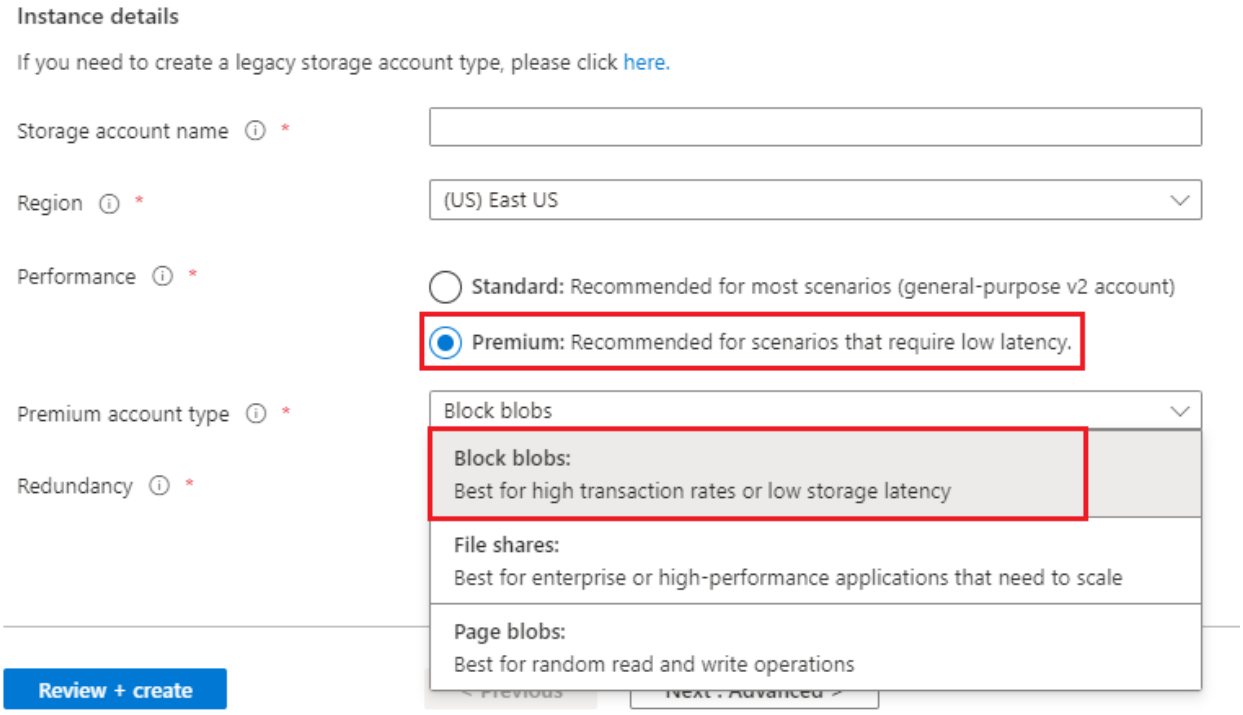
Security

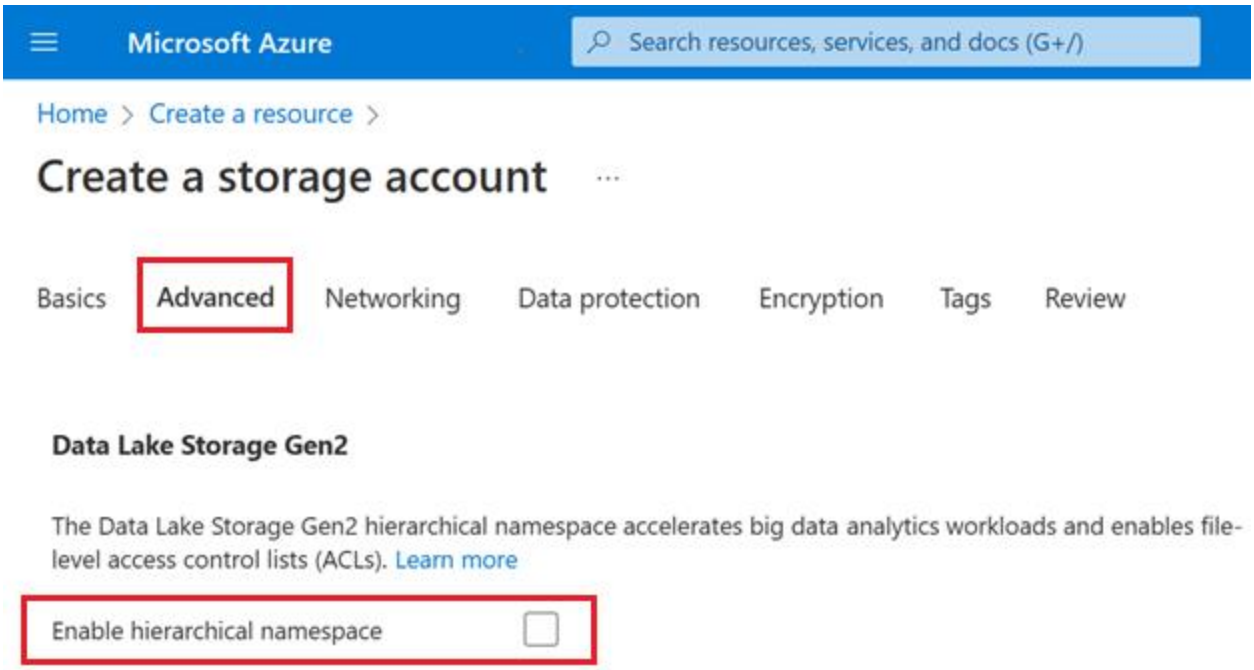
Secure transfer	Enabled
Allow storage account key access	Enabled
Default to Azure Active Directory authorization in the Azure portal	Disabled
Blob anonymous access	Disabled
Minimum TLS version	Version 1.2
Permitted scope for copy operations (preview)	From any storage account

Data protection

Point-in-time restore	Disabled
Blob soft delete	Enabled
Blob retainment period in days	7
Container soft delete	Enabled
Container retainment period in days	7

Create | < Previous | Next > | Download a template for automation


Instance details
If you need to create a legacy storage account type, please click here.
Storage account name
Region
(US) East US
Performance
Standard: Recommended for most scenarios (general-purpose v2 account)
Premium: Recommended for scenarios that require low latency.
Premium account type
Block blobs
Redundancy
Block blobs:
Best for high transaction rates or low storage latency
File shares:
Best for enterprise or high-performance applications that need to scale
Page blobs:
Best for random read and write operations
Review + create


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Create a resource >
Create a storage account
Basics
Advanced
Networking
Data protection
Encryption
Tags
Review
Data Lake Storage Gen2
The Data Lake Storage Gen2 hierarchical namespace accelerates big data analytics workloads and enables file-level access control lists (ACLs). Learn more
Enable hierarchical namespace

🌟 What is Azure Data Factory (ADF)?

Azure Data Factory (ADF) is a cloud-based data integration service provided by Microsoft Azure. It enables seamless and scalable data movement and orchestration across various sources, including Azure Storage accounts and Azure SQL databases. ADF acts as a powerful **Extract, Transform, Load** (ETL) tool, allowing users to efficiently transfer, transform, and process large volumes of data. With its robust capabilities and intuitive user interface, ADF empowers organizations to streamline their data workflows, automate data pipelines, and ensure data integrity throughout the entire process. Whether you need to ingest data from multiple sources, transform and enrich it, or load it into various target destinations, ADF provides the flexibility, scalability, and reliability to handle complex data integration scenarios effectively

🌟 🌟 Features of Azure Data Factory

💬 **Data Compression:** During the Data Copy activity, it is possible to compress the data and write the compressed data to the target data source. This feature helps optimize bandwidth usage in data copying.

💬 **Extensive Connectivity Support for Different Data Sources: Azure** Data Factory provides broad connectivity support for connecting to different data sources. This is useful when you want to pull or write data from different data sources.

💬 **Custom Event Triggers:** Azure Data Factory allows you to automate data processing using custom event triggers. This feature allows you to automatically execute a certain action when a certain event occurs.

💬 **Data Preview and Validation:** During the Data Copy activity, tools are provided for previewing and validating data. This feature helps you ensure that data is copied correctly and written to the target data source correctly.

💬 **Customizable Data Flows:** Azure Data Factory allows you to create customizable data flows. This feature allows you to add custom actions or steps for data processing.

💬 **Integrated Security:** Azure Data Factory offers integrated security features such as Entra ID integration and role-based access control to control access to dataflows. This feature increases security in data processing and protects your data.

🌟 🌟 🌟 Prerequisites to Deploying Azure Data Factory

Before you proceed with the article, you need to have the following prerequisites in place:

- **Azure Account with an active subscription**: The first thing you need to do is sign up for an Azure account if you don't already have one. You can sign up for a free trial account that will give you access to a limited amount of Azure resources for a limited time.
- **Permission to create resources in the subscription**: Once you have an Azure account, you need to make sure that you have the enough permissions to create and interact with Azure Data Factory, Azure Storage Account and Azure SQL Database resource. You can work with a contributor role on the subscription or resource group to create these resources.

Througout the article, you'll interact with the following resources:

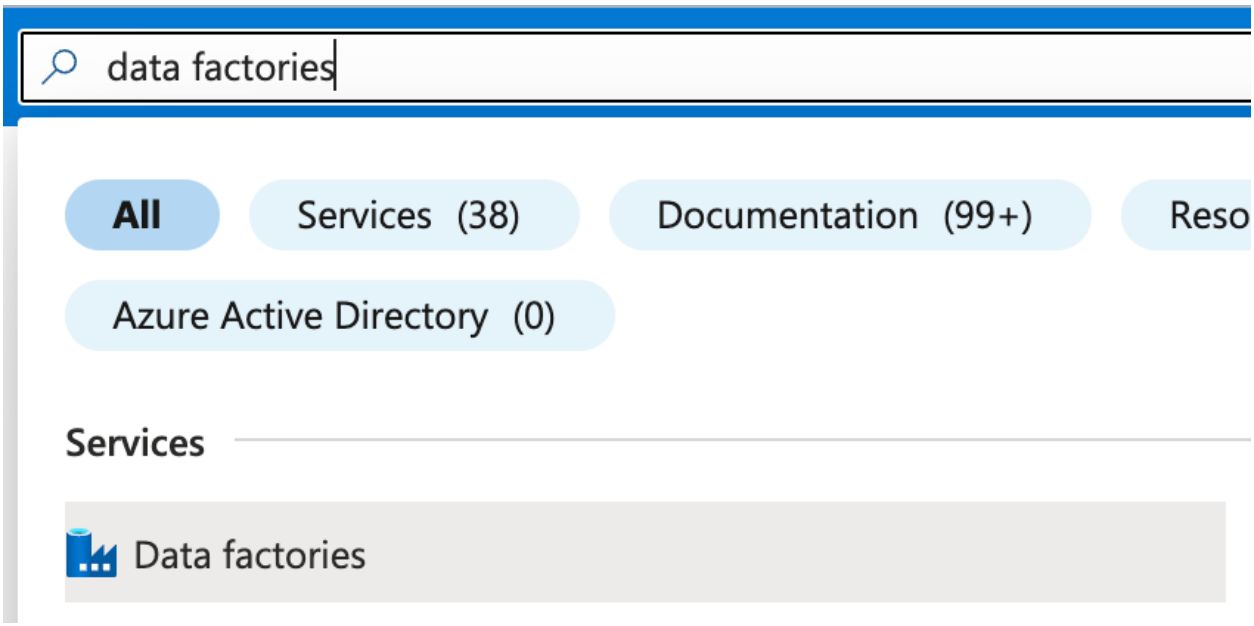
- Azure Storage Account and Container
- Azure SQL Server & Database
- Azure Data Factory and Pipelines

🌟 🌟 🌟 🌟 Creating an Azure Data Factory using Azure Portal

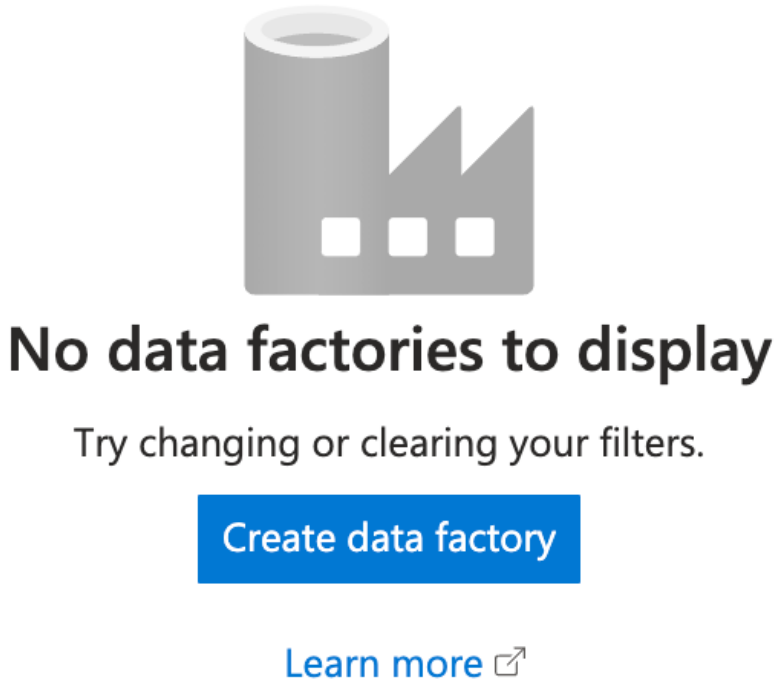
The first step in building a data pipeline with Azure Data Factory is to plan out your pipeline. This involves identifying the data sources and destinations, as well as any transformations or processing that needs to occur along the way. You'll also need to consider the frequency of data updates and any dependencies between different components of the pipeline. By taking the time to plan out your pipeline in advance, you can ensure that it meets your business requirements and is scalable for future growth.

In this section, you will create an Azure Data Factory using Azure Portal. Follow these steps below to begin:

- While in the Azure Portal, type *Azure Data Factory* in the search bar and click **Data factories** under the **Services**:

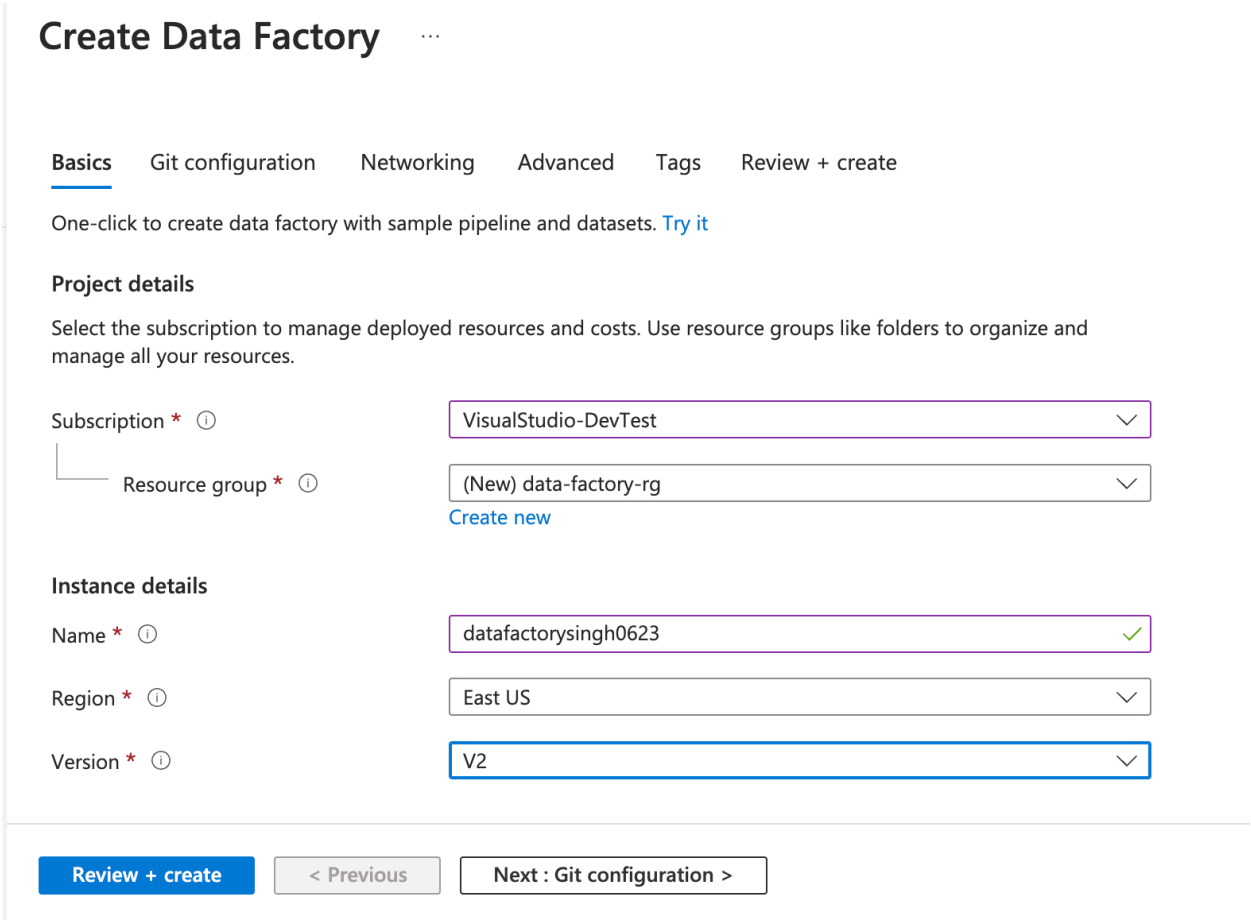


- Click the **Create data factory** button to create a new Azure Data Factory instance:

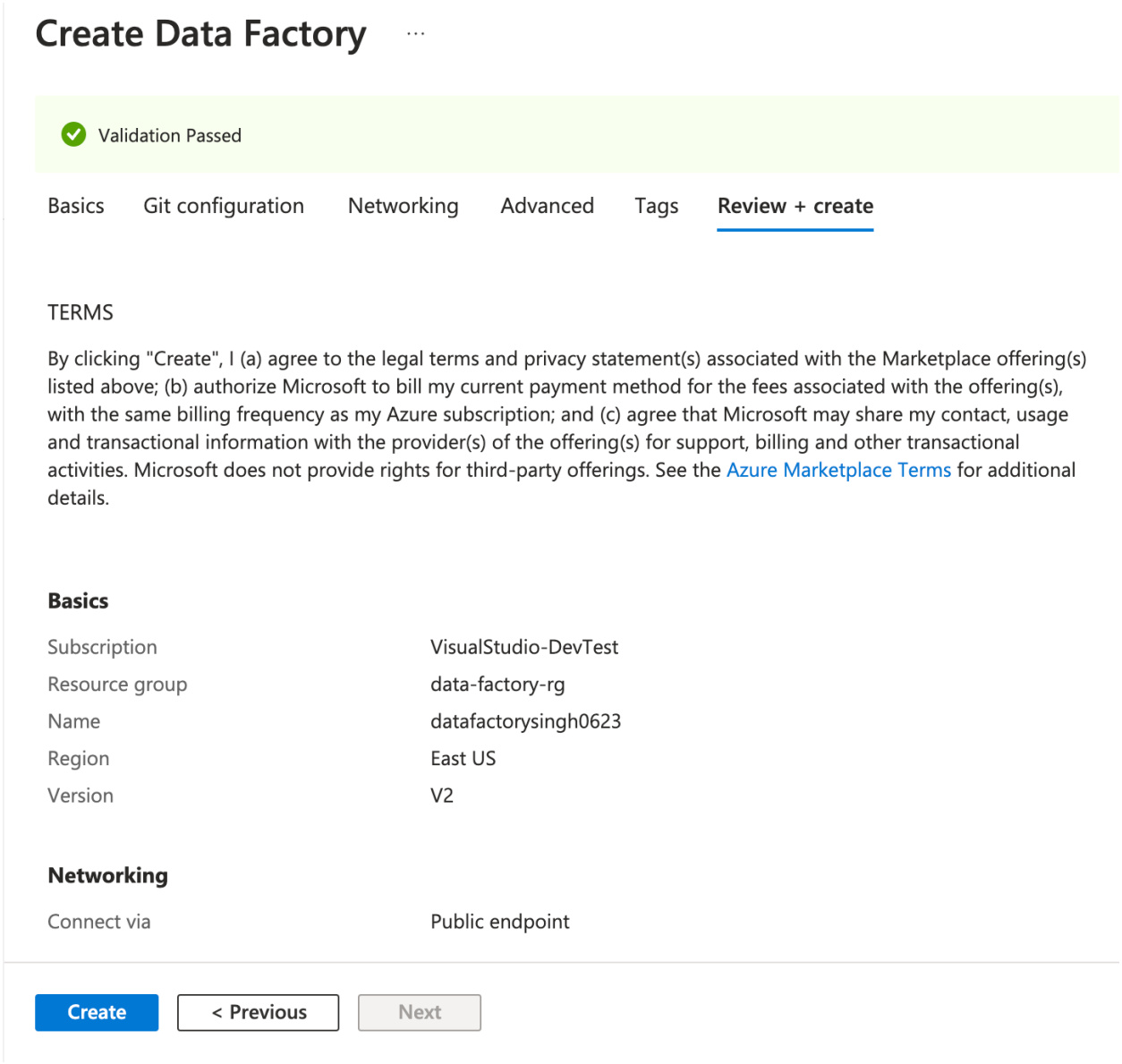


- Fill out the following on the **Create Data Factory** popup under the **Basics** tab and click **Review + create**:

- **Resource group**: Select the resource group created earlier.
- **Name**: Enter any name of your choice. Ensure that the name is globally unique.
- **Region**: Choose a location of your choice. (**East US** in this case)
- **Version**: Select **V2** version from the dropdown



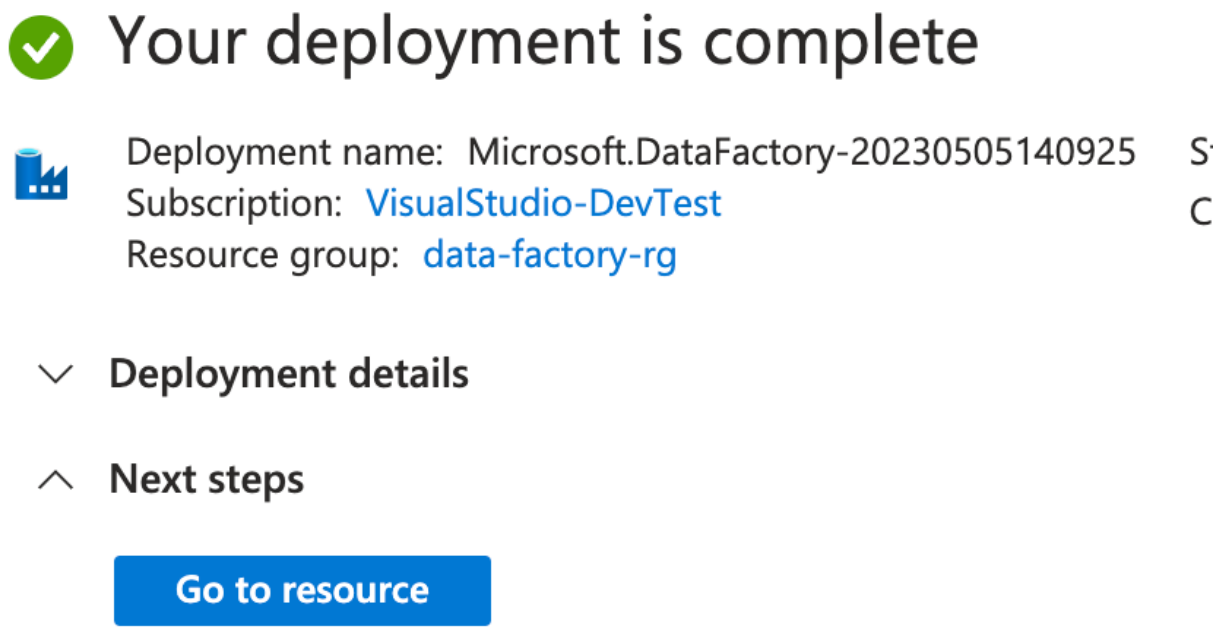
- Click **Create** once validation passes:



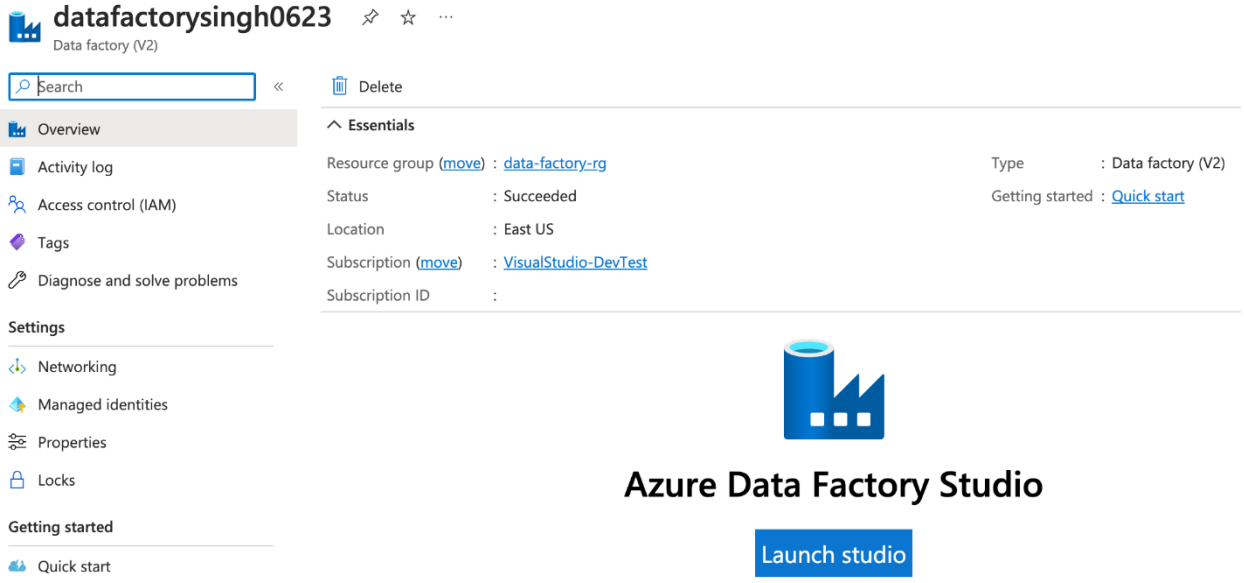
Note: If you are creating ADF in your production environment you can use Networking tab to adjust the network settings. The following options are available for the endpoint:

- **Public Endpoint:** Traffic can reach the service resource from on premises without using public endpoints. A Service Endpoint remains a publicly routable IP address.
- **Private Endpoint:** A Private Endpoint is a private IP in the address space of the virtual network where the private endpoint is configured.

- The deployment process will take a few minutes. Wait for the deployment to finish and click **Go to resource**.



- You will be taken to the Data Factory resource **Overview** tab where you will see the resource configuration and an overview of the data factory resource.



INTEGRATION RUNTIME

WHAT IS INTEGRATION RUNTIME?

- Integration runtime is the compute infrastructure used by Azure Data Factory (ADF) to provide various data integration capabilities across different network environments.

- An integration runtime provides the bridge between activities and linked services. It provides the compute environment where the activity is either run directly or dispatched.

- Integration runtime provides the following functions

    1. Data Flow
    2. Data Movement
    3. Activity Dispatch
    4. SSIS package execution

- There are 3 types of Integration runtimes

    1.**Self-hosted IR** 2. **Azure IR** 3. **Azure-SSIS IR**

SELF-HOSTED IR

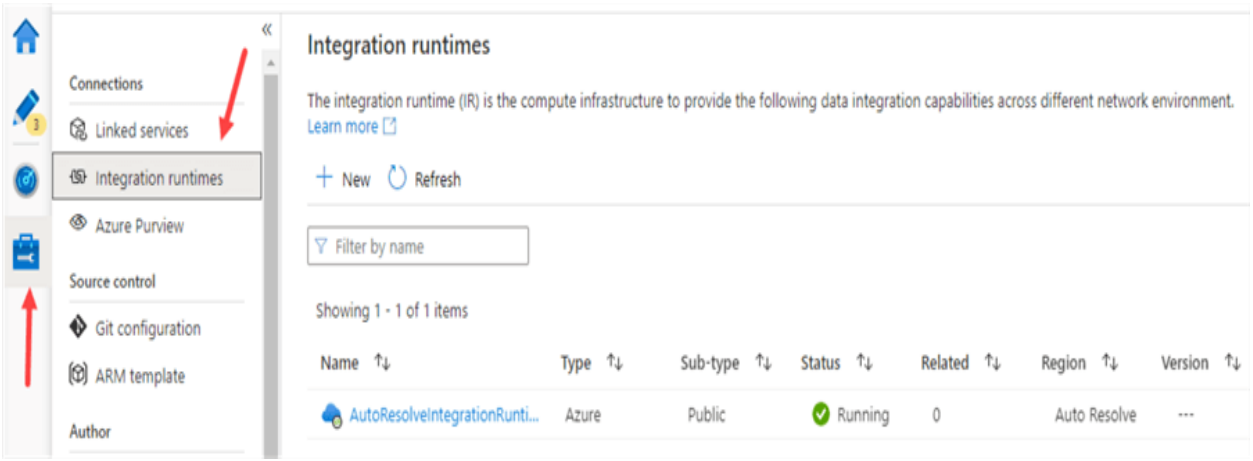
- **Description:** This option allows you to run data integration tasks on your own hardware or in a virtual machine. It's essential when you need to access data that resides in a private network or on-premises.
- **Use Cases:** Useful for hybrid data scenarios where data needs to be transferred between on-premises environments and cloud services.

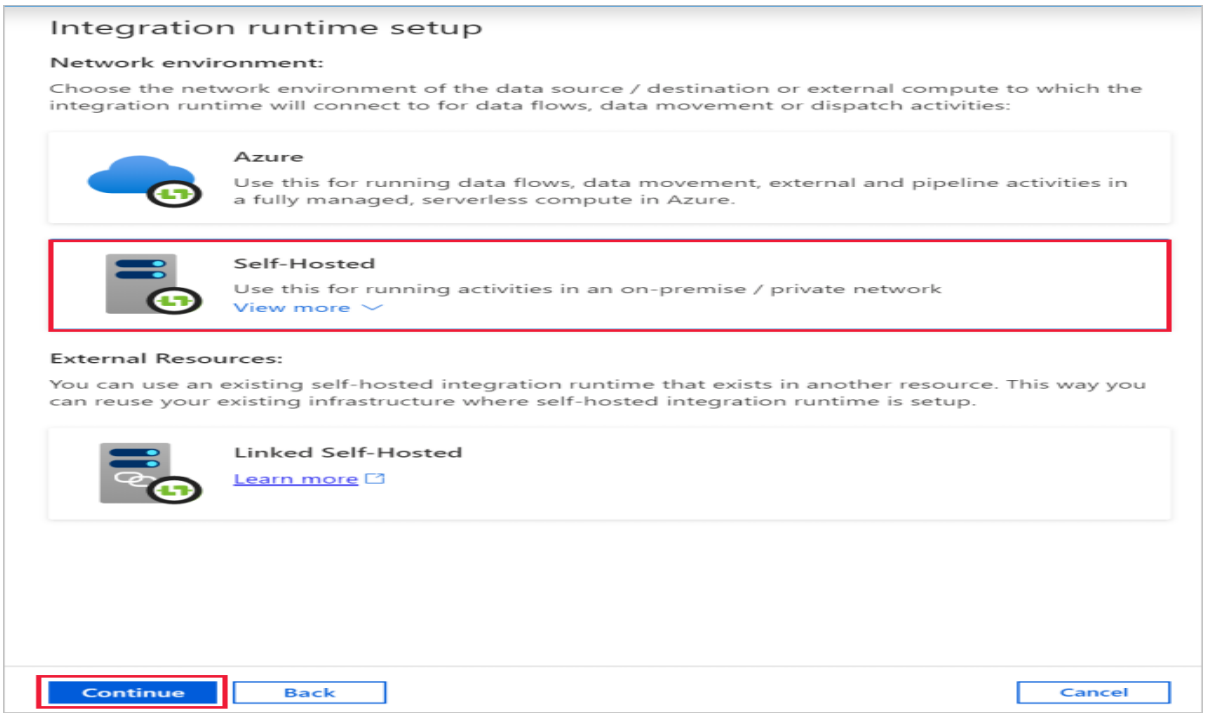
AZURE IR

- **Description:** This is a fully managed, serverless option provided by Azure. It is used for data movement, data transformation, and activity dispatch within the Azure cloud.
- **Use Cases:** Ideal for copying data between Azure data stores and performing data transformations using Azure services.

AZURE-SSIS IR

- **Description:** This runtime is specifically designed for running SQL Server Integration Services (SSIS) packages in Azure. It allows for the migration of existing SSIS packages to Azure.
- **Use Cases:** Ideal for organizations that have existing SSIS workloads and want to lift and shift these to the cloud without rewriting them.




Integration runtime setup
Network environment:
Choose the network environment of the data source / destination or external compute to which the integration runtime will connect to for data flows, data movement or dispatch activities:
Azure
Use this for running data flows, data movement, external and pipeline activities in a fully managed, serverless compute in Azure.
Self-Hosted
Use this for running activities in an on-premise / private network
View more
External Resources:
You can use an existing self-hosted integration runtime that exists in another resource. This way you can reuse your existing infrastructure where self-hosted integration runtime is setup.
Linked Self-Hosted
Learn more
Continue
Back
Cancel

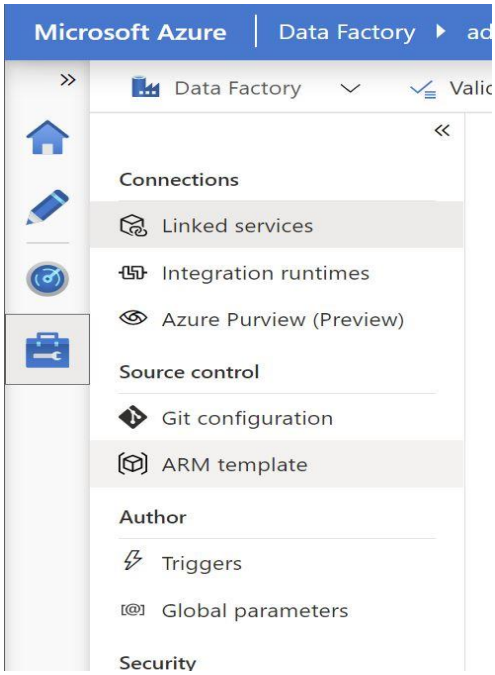
AZURE DATA FACTORY: LINKED SERVICES AND DATASETS

Linked Services 🔗

Linked Services are essential in ADF for connecting to various data sources and destinations. They define the connection information needed to access your data. Think of them as connection strings or configurations for your data stores.

A linked service contains the connection details (connection string), e.g. Database server, database name, file path, URL etc. A Linked Service might include authentication information related to the connection e.g. login id, passwords, API Keys etc. in an encrypted format. Linked Services can be created under the Manage tab in UI (screenshot below) . Linked Services can be parameterized, so that one Linked Service can be reused with similar datastore types e.g. a generic Database Linked Service to access multiple Databases. This will be discussed in detail in a future post.

- **Purpose:** Establish connections between ADF and your data stores or compute environments.
- **Key Features:**
    - **Connection Details:** Specify parameters like server names, authentication methods, and database names.
    - **Integration Runtimes:** Linked Services use integration runtimes to connect to on-premises or cloud data sources.
    - **Types:** Includes Azure Blob Storage, SQL databases, SaaS applications, and more.

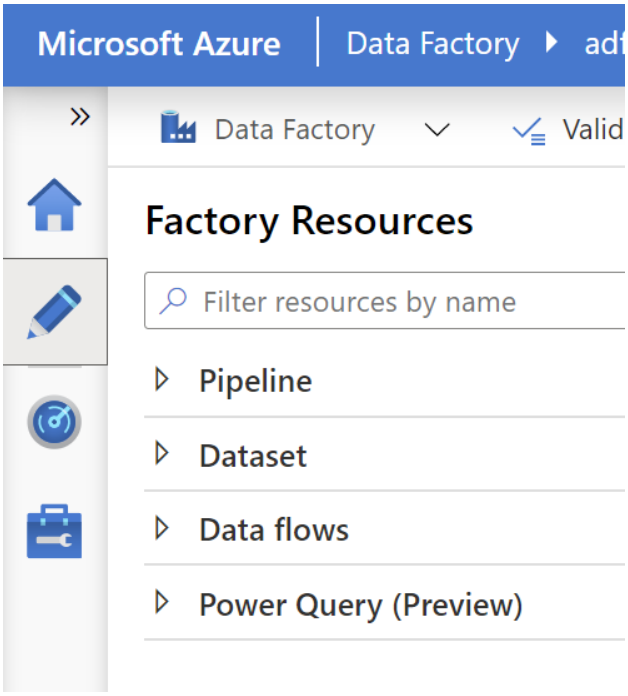


Datasets 📂

Datasets represent the data structures within your data stores. They define the schema and provide the metadata needed for data operations. Datasets act as references to the data within your Linked Services.

A Dataset is a reference to a data store and provides a very specific pointer to an object within the Linked Service. E.g. If a Linked Service points to a Database instance, the dataset can refer to a specific table that we would like to use as source or sink in the Data Factory Pipeline. Datasets can be created on top of an existing Linked Service in the Author tab.

- **Purpose:** Represent data in a structured format for activities within your pipelines.
- **Key Features:**
    - **Schema:** Defines the structure of the data (columns, data types, etc.).
    - **Parameters:** Allows for dynamic data sources by passing parameters to datasets.
    - **Types:** Includes structured data (tables, files) and unstructured data (logs, blobs).



How They Work Together 🤝

1. **Create a Linked Service:** Configure the connection to your data source (e.g., an Azure SQL Database or Azure Blob Storage).
2. **Define a Dataset:** Reference the data you want to work with, specifying the structure and any parameters.
3. **Use in Pipelines:** Integrate Linked Services and Datasets into your pipelines for data movement, transformation, and orchestration.

By mastering Linked Services and Datasets, you can build more robust and flexible data workflows in Azure Data Factory, seamlessly connecting to and working with diverse data sources.

PIPELINES AND ACTIVITIES IN AZURE DATA FACTORY

In Azure Data Factory (ADF), **Pipelines** and **Activities** are core components that help you build, orchestrate, and manage data workflows. Let's break down what each term means and how they function:

1. Pipelines

A **Pipeline** in Azure Data Factory is a logical grouping of activities that perform a specific task or set of tasks. It serves as a container that holds activities, orchestrates the execution of those activities, and provides the workflow for data processing.

- **Purpose:** Pipelines enable you to manage and execute complex data workflows, handling data movement and transformation efficiently.
- **Components:** ○ **Activities:** The tasks executed within a pipeline.
    - **Triggers:** Define when and how pipelines should run, such as on a schedule or in response to events.
    - **Parameters:** Allow you to pass dynamic values into the pipeline, making it flexible and reusable.
    - **Integration Runtimes:** Define the compute environment used for executing the activities in a pipeline.

2. Activities

Activities are the building blocks of a pipeline. They represent individual tasks that perform operations such as data movement, transformation, and control flow.

- **Types of Activities:**
    - **Data Movement Activities:** Move data from a source to a destination. Examples include Copy Activity and Data Flow Activity.
        - **Copy Activity:** Copies data from a source dataset to a destination dataset. Useful for ETL (Extract, Transform, Load) processes.
        - **Data Flow Activity:** Provides a way to design and execute data transformations using a visual interface.
    - **Data Transformation Activities:** Transform data using compute services. Examples include Mapping Data Flow and Wrangling Data Flow.

        - **Mapping Data Flow:** Allows for data transformation through a visual interface where you can map, aggregate, and filter data.
        - **Wrangling Data Flow:** Allows data preparation and transformation using a code-free interface, leveraging Power Query.
    - **Control Flow Activities:** Manage the execution sequence and dependencies of other activities. Examples include If Condition, ForEach, and Execute Pipeline.
        - **If Condition Activity:** Executes activities based on a specified condition.
        - **ForEach Activity:** Executes a set of activities for each item in a collection.
        - **Execute Pipeline Activity:** Invokes another pipeline, allowing for modular design and reuse.
    - **Miscellaneous Activities:** Perform additional operations, such as managing resources or custom actions. Examples include Web Activity and Stored Procedure Activity.
        - **Web Activity:** Calls an HTTP endpoint and processes the response.
        - **Stored Procedure Activity:** Executes a stored procedure in a database.

Putting It All Together

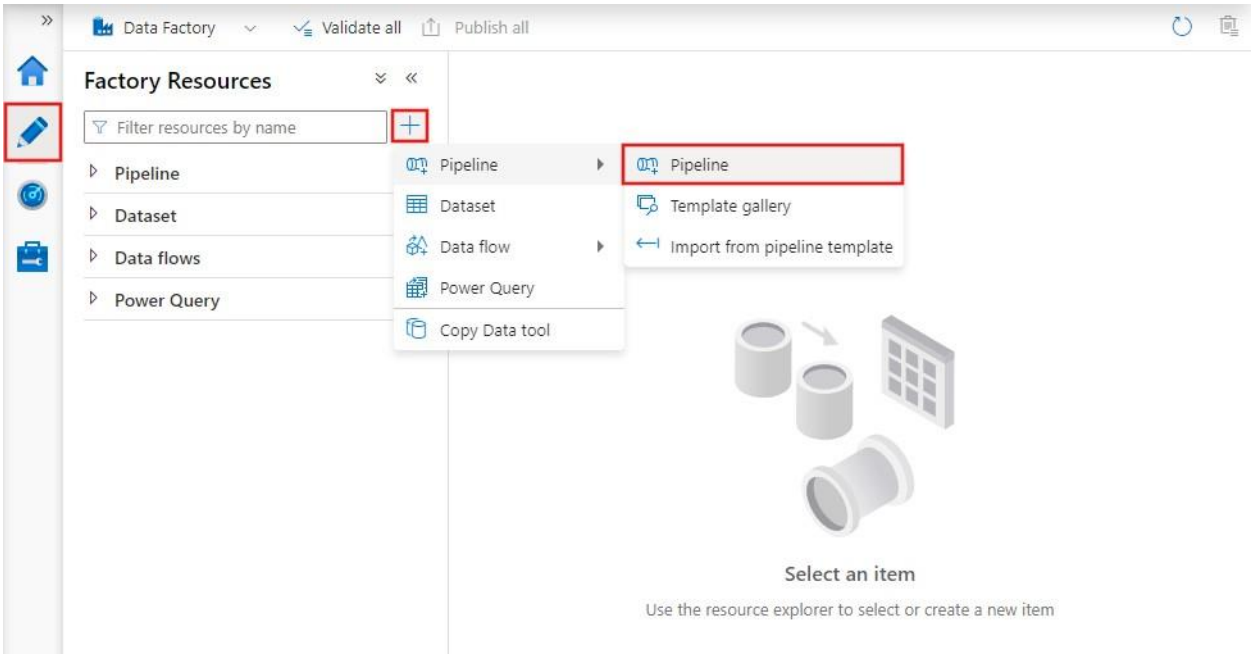
To create a data workflow, you design a pipeline that orchestrates a series of activities. For example, a pipeline might use a Copy Activity to move data from an on-premises database to an Azure Data Lake, followed by a Data Flow Activity to transform the data, and finally a

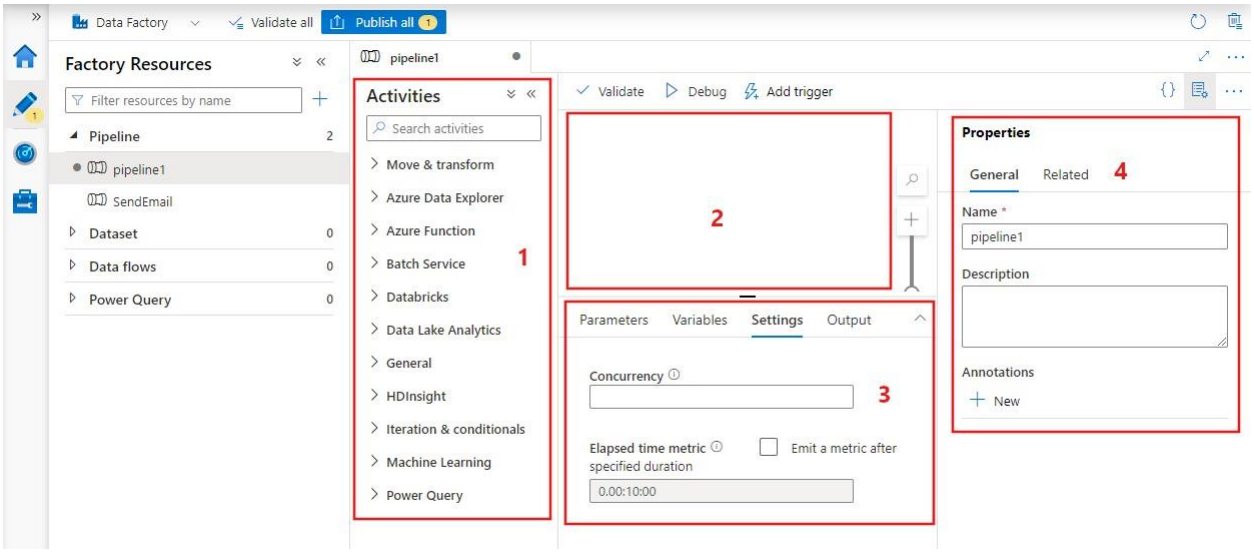
Web Activity to trigger an external process based on the results.

By leveraging pipelines and activities, you can build robust and scalable data integration solutions in Azure Data Factory, automating complex data workflows and ensuring efficient data processing.

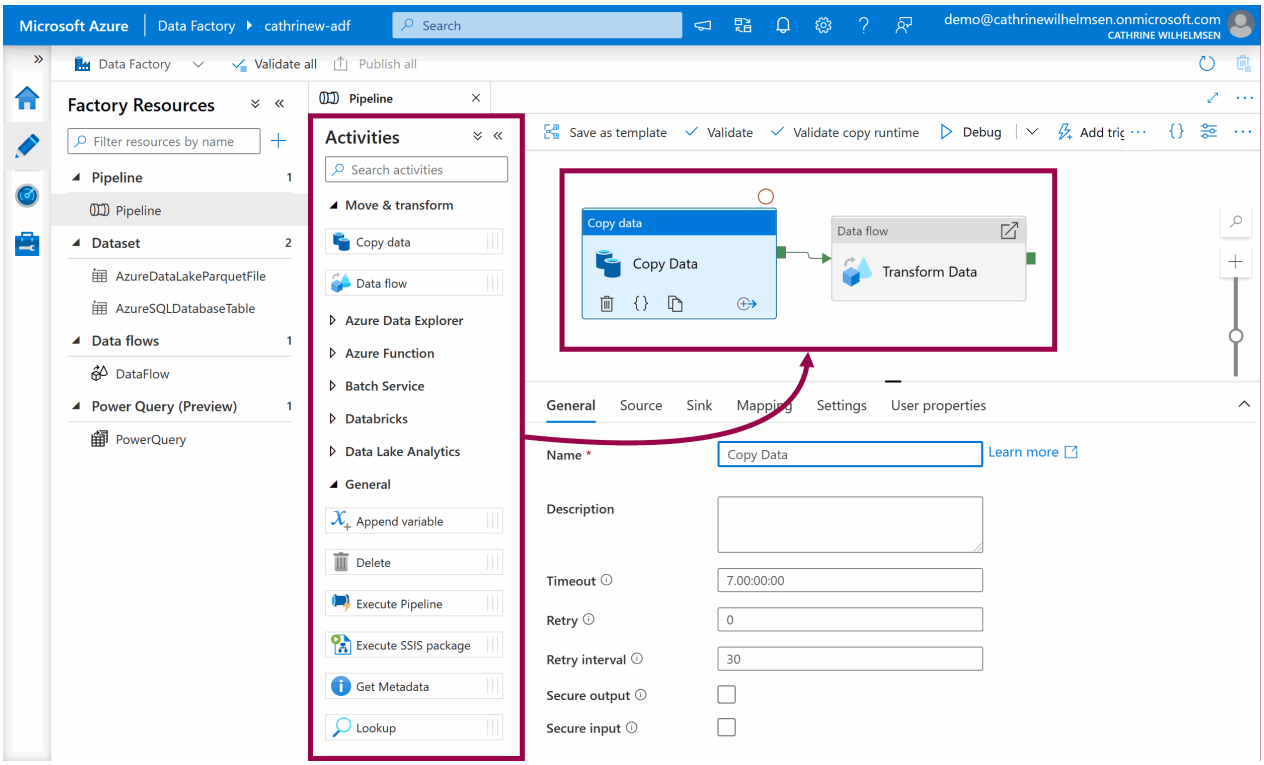
Azure Data Factory and Azure Synapse Analytics have three groupings of activities: data movement activities, data transformation activities, and control activities. An activity can take zero or more input datasets and produce one or more output datasets. The following diagram shows the relationship between pipeline, activity, and dataset:

Creating a pipeline with UI





Activities


Microsoft Azure
Data Factory
cathrinew-adf
Search
demo@cathrinewilhelmsen.onmicrosoft.com
CATHRINE WILHELMSEN
Data Factory
Validate all
Publish all
Factory Resources
Filter resources by name
Pipeline 1
Pipeline
Dataset 2
AzureDataLakeParquetFile
AzureSQLDatabaseTable
Data flows 1
DataFlow
Power Query (Preview) 1
PowerQuery
Pipeline
Activities
Search activities
Move & transform
Copy data
Data flow
Azure Data Explorer
Azure Function
Batch Service
Databricks
Data Lake Analytics
General
Append variable
Delete
Execute Pipeline
Execute SSIS package
Get Metadata
Lookup
Save as template
Validate
Validate copy runtime
Debug
Add trig
Copy data
Copy Data
Data flow
Transform Data
General
Source
Sink
Mapping
Settings
User properties
Name *
Copy Data
Learn more
Description
Timeout
7.00:00:00
Retry
0
Retry interval
30
Secure output
Secure input

TRIGGERS IN AZURE DATA FACTORY

A **Trigger** is a feature that enables you to automate the execution of your pipelines based on specific conditions or schedules. Triggers determine when a pipeline should run, allowing for greater control and efficiency in data integration and transformation processes.

Types of Triggers in Azure Data Factory:

- Scheduled Trigger
- Tumbling Window Trigger
- Storage Event Trigger
- Custom Event Trigger

1. **Scheduled Trigger**:

- **Description**: Runs a pipeline at specified intervals (e.g., daily, hourly, weekly).
- **Use Cases**: Ideal for recurring tasks such as daily data loading or periodic reports.
- **Configuration**: You can set the frequency and the start time.

2. Tumbling Window Trigger:

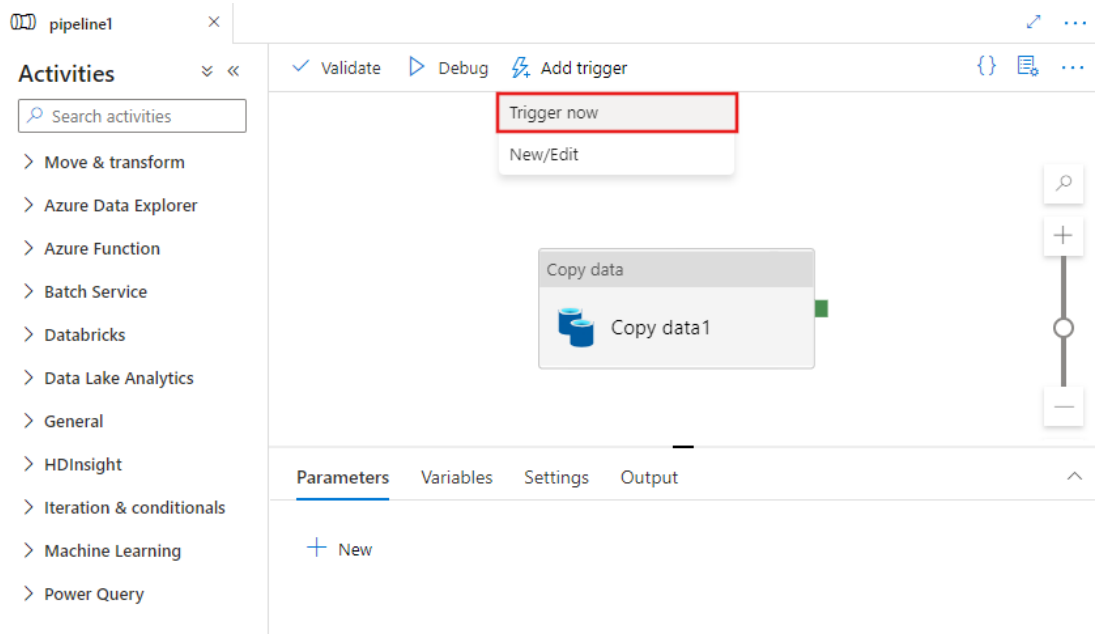
- **Description**: Executes a pipeline on a recurring schedule with non-overlapping time windows (e.g., every hour, processing data for the last hour).
- **Use Cases**: Suitable for batch processing, where each run handles a distinct set of data.
- **Configuration**: Define the duration of the window and the start time.

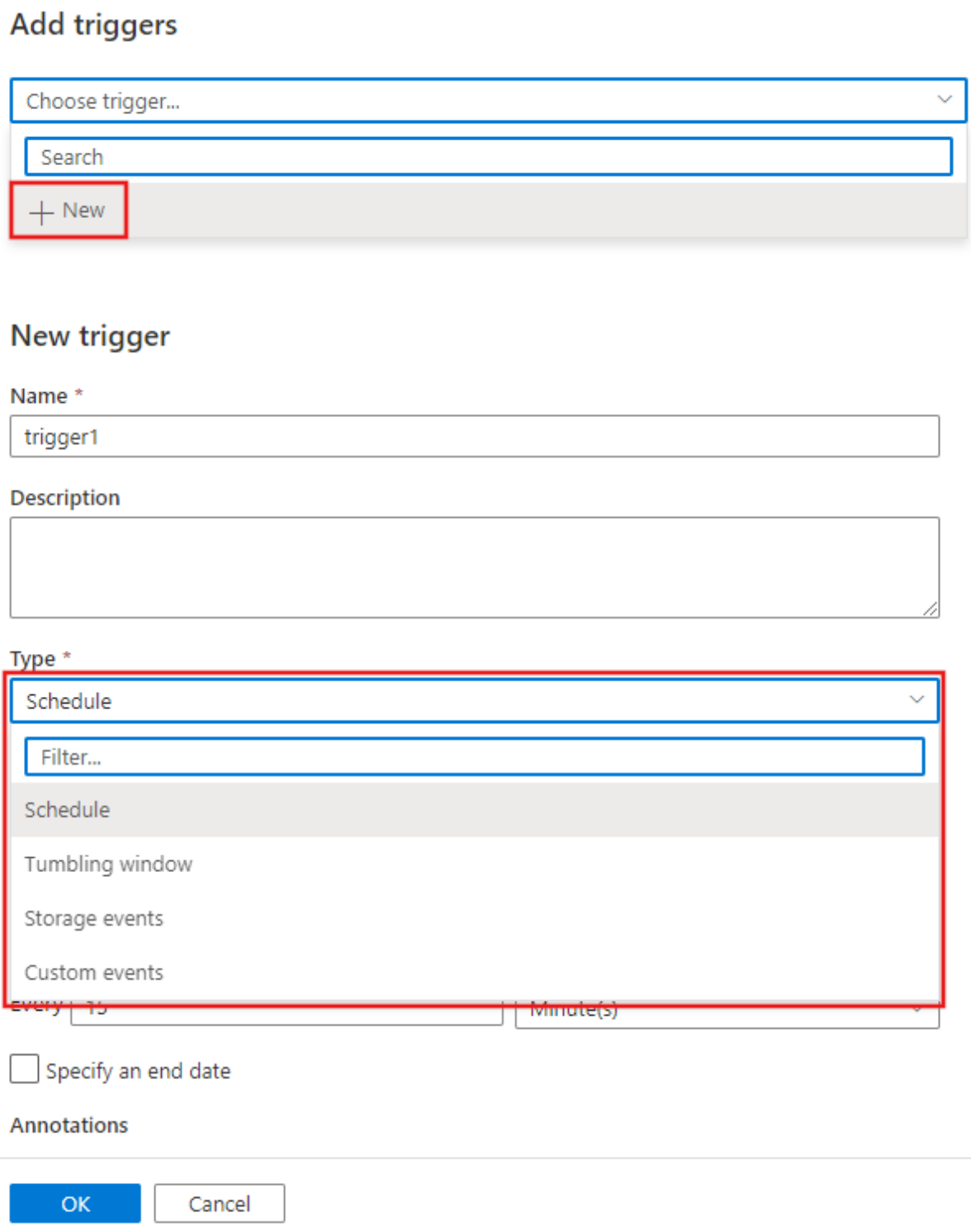
3. **Storage Event Trigger**:

- **Description**: Initiates a pipeline in response to events in Azure Storage, such as when a new file is created or an existing file is deleted.
- **Use Cases**: Useful for data ingestion workflows that react to changes in data availability.
- **Configuration**: Specify the storage account, event types, and paths to monitor.

4. **Custom Event Trigger**:

- **Description**: Triggers a pipeline based on events from external sources, such as Azure Event Grid or Service Bus.
- **Use Cases**: Ideal for scenarios where workflows need to respond to application-specific events or API calls.
- **Configuration**: Set up using Azure Event Grid or other event sources, defining the event criteria.




Add triggers
Choose trigger...
Search
New
New trigger
Name *
trigger1
Description
Type *
Schedule
Filter...
Schedule
Tumbling window
Storage events
Custom events
Minute(s)
Specify an end date
Annotations
OK
Cancel