**What is Azure Data Factory?**

Cloud-based integration service that allows creating data-driven workflows in the cloud for orchestrating and automating data movement and data transformation.

* Using Azure data factory, you can create and schedule the data-driven workflows(called pipelines) that can ingest data from disparate data stores.
* It can process and transform the data by using compute services such as HDInsight Hadoop, Spark, Azure Data Lake Analytics, and Azure Machine Learning.

**What is the integration runtime?**

* The integration runtime is the compute infrastructure that Azure Data Factory uses to provide the following data integration capabilities across various network environments.
* 3 Types of integration runtimes:
* **Azure Integration Run Time:** Azure Integration Run Time can copy data between cloud data stores and it can dispatch the activity to a variety of compute services such as Azure HDinsight or SQL server where the transformation takes place
* **Self Hosted Integration Run Time:**Self Hosted Integration Run Time is software with essentially the same code as Azure Integration Run Time. But you install it on an on-premise machine or a virtual machine in a virtual network. A Self Hosted IR can run copy activities between a public cloud data store and a data store in a private network. It can also dispatch transformation activities against compute resources in a private network. We use Self Hosted IR because Data factory will not be able to directly access on-primitive data sources as they sit behind a firewall.It is sometimes possible to establish a direct connection between Azure and on-premises data sources by configuring the firewall in a specific way if we do that we don’t need to use a self-hosted IR.
* **Azure SSIS Integration Run Time:**With SSIS Integration Run Time, you can natively execute SSIS packages in a managed environment. So when we lift and shift the SSIS packages to data factory, we use Azure SSIS Integration Run TIme.

### ****What is the limit on the number of integration runtimes?****

There is no hard limit on the number of integration runtime instances you can have in a data factory. There is, however, a limit on the number of VM cores that the integration runtime can use per subscription for SSIS package execution.

### ****What is the difference between Azure Data Lake and Azure Data Warehouse?****

Data Warehouse is a traditional way of storing data which is still used widely. Data Lake is complementary to Data Warehouse i.e if you have your data at a data lake that can be stored in data warehouse as well but there are certain rules that need to be followed.

|  |  |
| --- | --- |
| **DATA LAKE** | **DATA WAREHOUSE** |
| Complementary to data warehouse | Maybe sourced to the data lake |
| Data is Detailed data or Raw data. It can be in any particular form.you just need to take the data and dump it into your data lake | Data is filtered, summarised,refined |
| Schema on read (not structured, you can define your schema in n number of ways) | Schema on write(data is written in Structured form or in a particular schema) |
| One language to process data of any format(USQL) | It uses SQL |

**What is blob storage in Azure?**

[Azure Blob Storage](https://azure.microsoft.com/en-us/services/storage/blobs/) is a service for storing large amounts of unstructured object data, such as text or binary data. You can use Blob Storage to expose data publicly to the world or to store application data privately. Common uses of Blob Storage include:

* Serving images or documents directly to a browser
* Storing files for distributed access
* Streaming video and audio
* Storing data for backup and restore disaster recovery, and archiving
* Storing data for analysis by an on-premises or Azure-hosted service

### ****What is the difference between Azure Data Lake store and Blob storage?****

|  |  |  |
| --- | --- | --- |
|  | **Azure Data Lake Storage Gen1** | **Azure Blob Storage** |
| **Purpose** | Optimized storage for big data analytics workloads | General purpose object store for a wide variety of storage scenarios, including big data analytics |
| **Structure** | Hierarchical file system | Object store with flat namespace |
| **Key Concepts** | Data Lake Storage Gen1 account contains folders, which in turn contains data stored as files | Storage account has containers, which in turn has data in the form of blobs |
| **Use Cases** | Batch, interactive, streaming analytics and machine learning data such as log files, IoT data, click streams, large datasets | Any type of text or binary data, such as application back end, backup data, media storage for streaming and general purpose data. Additionally, full support for analytics workloads; batch, interactive, streaming analytics and machine learning data such as log files, IoT data, click streams, large datasets |
| **Server-side API** | [WebHDFS-compatible REST API](https://msdn.microsoft.com/library/azure/mt693424.aspx) | [Azure Blob Storage REST API](https://msdn.microsoft.com/library/azure/dd135733.aspx) |
| **Data Operations – Authentication** | Based on [Azure Active Directory Identities](https://docs.microsoft.com/en-us/azure/active-directory/develop/authentication-scenarios) | Based on shared secrets – [Account Access Keys](https://docs.microsoft.com/en-us/azure/storage/common/storage-account-manage#access-keys) and [Shared Access Signature Keys](https://docs.microsoft.com/en-us/azure/storage/common/storage-dotnet-shared-access-signature-part-1). |

**What are the top-level concepts of Azure Data Factory?**

* **Pipeline:**It acts as a carrier in which we have various processes taking place.

This individual process is an activity.

* **Activities:**Activities represent the processing steps in a pipeline. A pipeline can have one or multiple activities. It can be anything i.e process like querying a data set or moving the dataset from one source to another.
* **Datasets:**Sources of data. In simple words, it is a data structure that holds our data.
* **Linked services**: These store information that is very important when it comes to connecting an external source.

**How can I schedule a pipeline?**

* You can use the scheduler trigger or time window trigger to schedule a pipeline.
* The trigger uses a wall-clock calendar schedule, which can schedule pipelines periodically or in calendar-based recurrent patterns (for example, on Mondays at 6:00 PM and Thursdays at 9:00 PM).

**Can I pass parameters to a pipeline run?**

* Yes, parameters are a first-class, top-level concept in Data Factory.
* You can define parameters at the pipeline level and pass arguments as you execute the pipeline run on demand or by using a trigger.

### ****Can I define default values for the pipeline parameters?****

You can define default values for the parameters in the pipelines.

### ****Can an activity in a pipeline consume arguments that are passed to a pipeline run?****

Each activity within the pipeline can consume the parameter value that’s passed to the pipeline and run with the @parameter construct.

### ****Can an activity output property be consumed in another activity?****

An activity output can be consumed in a subsequent activity with the @activity construct.

### ****How do I gracefully handle null values in an activity output?****

You can use the @coalesce construct in the expressions to handle the null values gracefully.

### ****Which Data Factory version do I use to create data flows?****

Use the Data Factory V2 version to create data flows.

**How do I access data by using the other 80 dataset types in Data Factory?**

* The Mapping Data Flow feature currently allows Azure SQL Database, Azure SQL Data Warehouse, delimited text files from Azure Blob storage or Azure Data Lake Storage Gen2, and Parquet files from Blob storage or Data Lake Storage Gen2 natively for source and sink.
* Use the Copy activity to stage data from any of the other connectors, and then execute a Data Flow activity to transform data after it’s been staged. For example, your pipeline will first copy into Blob storage, and then a Data Flow activity will use a dataset in source to transform that data.

**Explain the two levels of security in ADLS Gen2?**

The two levels of security applicable to ADLS Gen2 were also in effect for ADLS Gen1. Even though this is not new, it is worth calling out the two levels of security because it’s a very fundamental piece to getting started with the data lake and it is confusing for many people just getting started.

* Role-Based Access Control (RBAC). RBAC includes built-in Azure roles such as reader, contributor, owner or custom roles. Typically, RBAC is assigned for two reasons. One is to specify who can manage the service itself (i.e., update settings and properties for the storage account). Another reason is to permit the use of built-in data explorer tools, which require reader permissions.
* Access Control Lists (ACLs). Access control lists specify exactly which data objects a user may read, write, or execute (execute is required to browse the directory structure). ACLs are POSIX-compliant, thus familiar to those with a Unix or Linux background.

POSIX does not oper-ate on a security inheritance model, which means that access ACLs are specified for every object. The concept of default ACLs is critical for new files within a directory to obtain the correct security settings, but it should not be thought of as inheritance. Because of the overhead assigning ACLs to every object, and because there is a limit of 32 ACLs for every object, it is extremely important to manage data-level security in ADLS Gen1 or Gen2 via Azure Active Directory groups.

#### **what is Backend Pool in Load balancer ?**

The backend pool will have the virtula machine which is connected with private IP address along with Availability set enabled to provide services will be connected over public IP address of the Load balancer.

## Tumbling window trigger

Tumbling window triggers are a type of trigger that fires at a periodic time interval from a specified start time, while retaining state. Tumbling windows are a series of fixed-sized, non-overlapping, and contiguous time intervals.

Data Factory offers three types of Integration Runtime (IR), and you should choose the type that best serve the data integration capabilities and network environment needs you're looking for. These three types are:

* Azure
* Self-hosted
* Azure-SSIS

The following table describes the capabilities and network support for each of the integration runtime types:

| **INTEGRATION RUNTIME TYPES** | | |
| --- | --- | --- |
| **IR type** | **Public network** | **Private network** |
| Azure | Data Flow Data movement Activity dispatch |  |
| Self-hosted | Data movement Activity dispatch | Data movement Activity dispatch |
| Azure-SSIS | SSIS package execution | SSIS package execution |

### Self-hosted IR network environment

If you want to perform data integration securely in a private network environment, which doesn't have a direct line-of-sight from the public cloud environment, you can install a self-hosted IR on premises environment behind your corporate firewall, or inside a virtual private network. The self-hosted integration runtime only makes outbound HTTP-based connections to open internet.

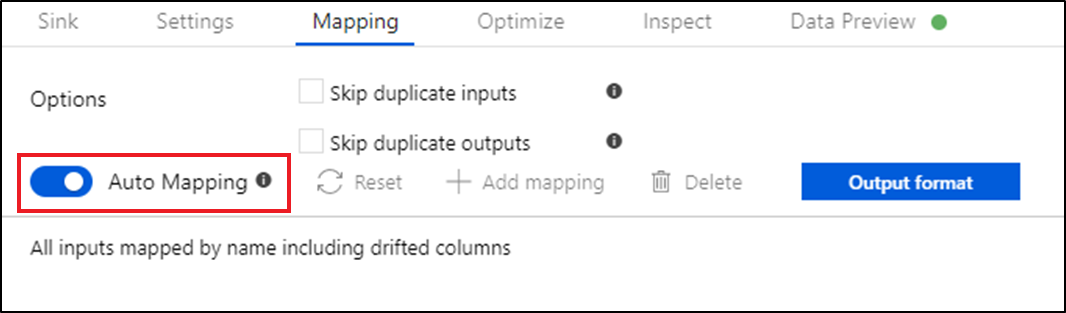
Mapping data flows are visually designed data transformations in Azure Data Factory. Data flows allow data engineers to develop graphical data transformation logic without writing code.

## Data flow canvas

The data flow canvas is separated into three parts: the top bar, the graph, and the configuration panel.

Schema drift is the case where your sources often change metadata. Fields, columns, and, types can be added, removed, or changed on the fly. Without handling for schema drift, your data flow becomes vulnerable to upstream data source changes. Typical ETL patterns fail when incoming columns and fields change because they tend to be tied to those source names.

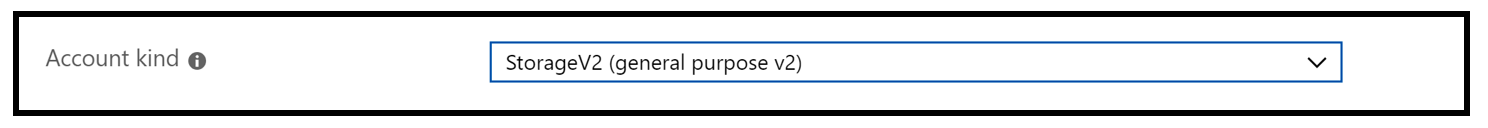
To protect against schema drift, it's important to have the facilities in a data flow tool to allow you, as a Data Engineer, to:

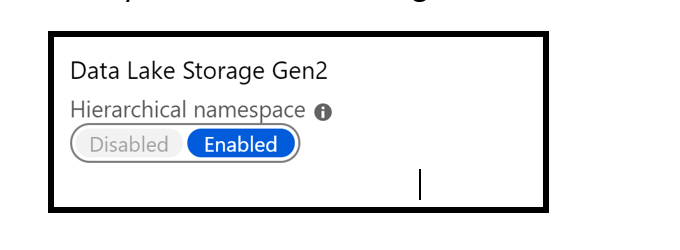
* Define sources that have mutable field names, data types, values, and sizes
* Define transformation parameters that can work with data patterns instead of hard-coded fields and values
* Define expressions that understand patterns to match incoming fields, instead of using named fields
* If schema drift is enabled, make sure the **Auto-mapping** slider in the Mapping tab is turned on. With this slider on, all incoming columns are written to your destination. Otherwise you must use rule-based mapping to write drifted columns.
* 

### Azure Data Lake Gen1 vs Azure Data Lake Gen2

|  |  |
| --- | --- |
| Azure Data Lake Gen1 | Azure Data Lake Gen2 |
| Azure Data Lake Gen 1 is file system storage in which data is distributed in blocks in a hierarchical file system. | Azure Data Lake Gen 2 contains both file system storage for performance & security and object storage for scalability. |
| Hot/Cold storage tier not supported | Supports Hot/Cold Storage tier |
| Redundant Storage not supported | Supports Redundant Storage |
| Azure data lake Analytics support available | Azure Data Lake analytics support is not available (till date 2nd July 2019). |

### Creating Azure Data Lake Gen2 and Converting Blob Storage to Gen 2

1. Go to all resource -> Click Add -> Choose Storage Account -> Choose Account Kind as Storage V2.

2. Once you created the Storage Account. Go to Configuration -> Enable Hierarchical Namespace.  


Azure Data Lake Gen 2 provides different access tier for storing the data.

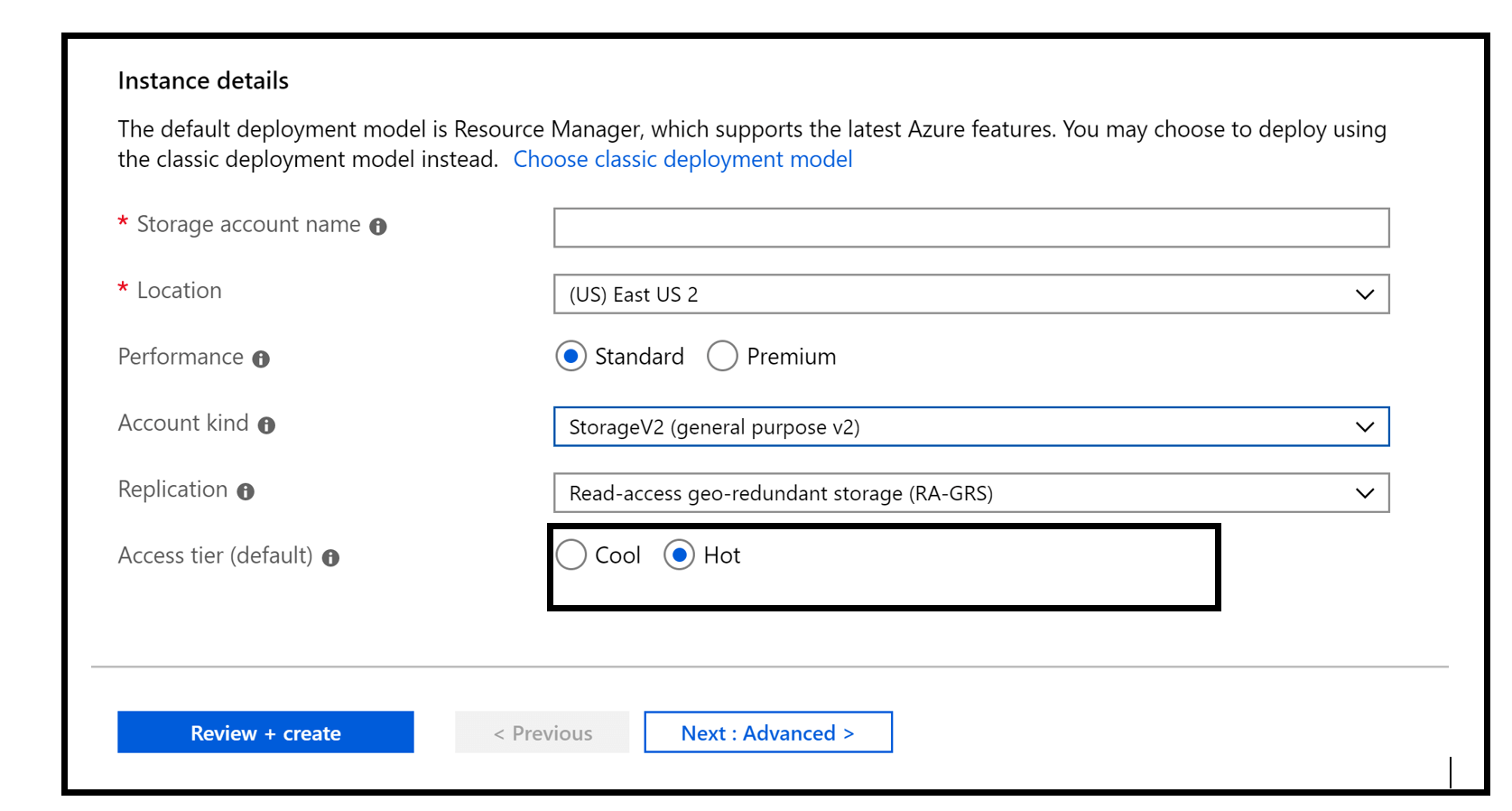
### Hot Storage

When Data Lake Gen 2 is created with Hot access tier then the file available in the storage is readily accessible. Storage Cost for hot access tier is higher whereas Access cost is lower. In case these files are not being accessed frequently it will lead to paying a lot of costs**.**

### Cool Storage

The purpose of creating the Cool access tier in Data Lake Gen 2(Storage Account V2) is that the file or storage is not accessed frequently. For example, monthly reports or annual reports which will be consumed once monthly or yearly have less access which will reduce the access cost. In the Cool Storage access tier, Storage cost is lower whereas Access cost is higher. In case these files are accessed frequently it leads to paying a lot of costs.

You can choose the hot access tier at the time of creating the Storage Account V2 (Data Lake Gen 2).

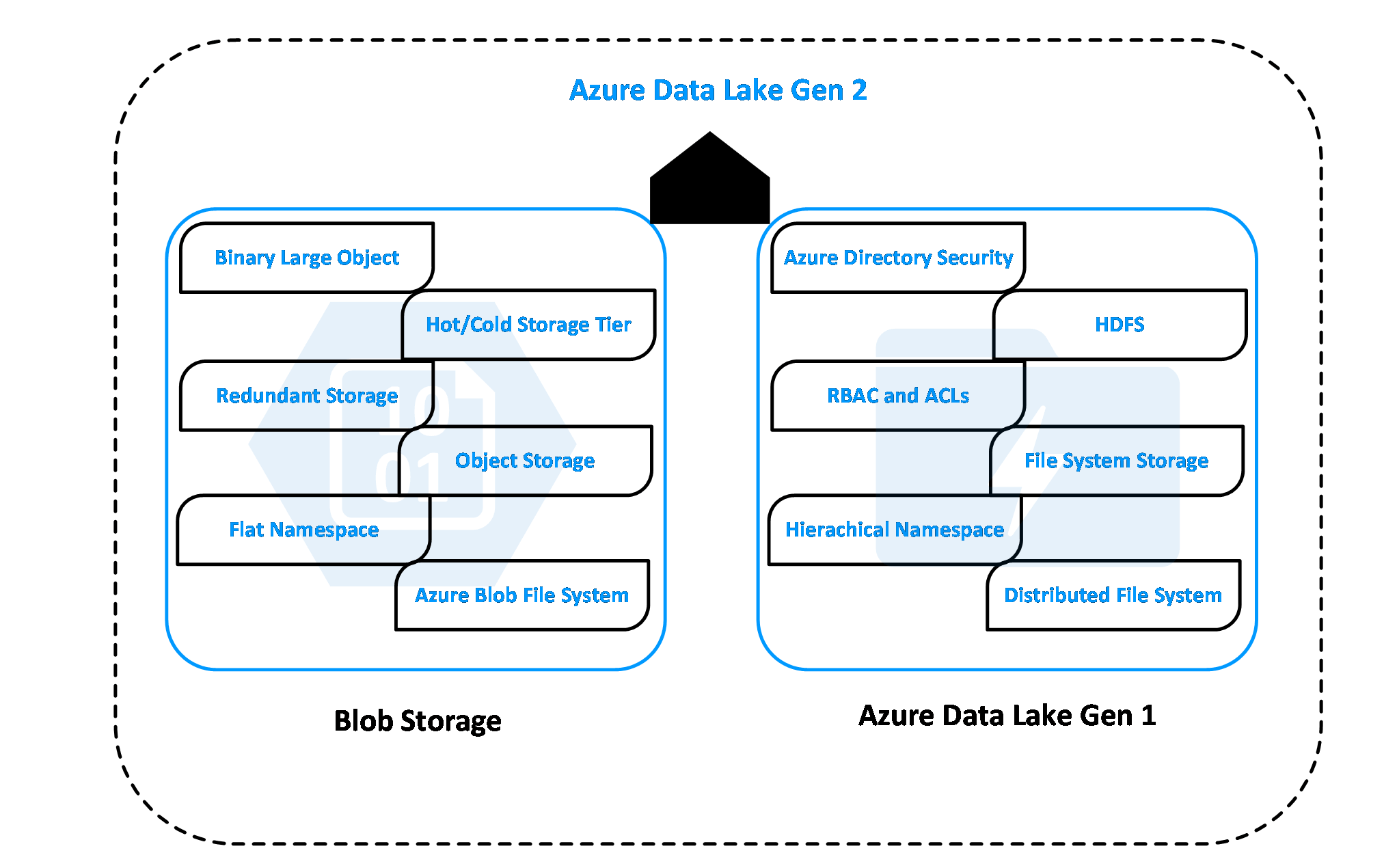


The following table compares the features of Data Factory with the features of Data Factory version 1.

| **FEATURE COMPARISON** | | |
| --- | --- | --- |
| **Feature** | **Version 1** | **Current version** |
| Datasets | A named view of data that references the data that you want to use in your activities as inputs and outputs. Datasets identify data within different data stores, such as tables, files, folders, and documents. For example, an Azure Blob dataset specifies the blob container and folder in Azure Blob storage from which the activity should read the data.  **Availability** defines the processing window slicing model for the dataset (for example, hourly, daily, and so on). | Datasets are the same in the current version. However, you do not need to define **availability** schedules for datasets. You can define a trigger resource that can schedule pipelines from a clock scheduler paradigm. For more information, see [Triggers](https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipeline-execution-triggers#trigger-execution) and [Datasets](https://docs.microsoft.com/en-us/azure/data-factory/concepts-datasets-linked-services). |
| Linked services | Linked services are much like connection strings, which define the connection information that's necessary for Data Factory to connect to external resources. | Linked services are the same as in Data Factory V1, but with a new **connectVia** property to utilize the Integration Runtime compute environment of the current version of Data Factory. For more information, see [Integration runtime in Azure Data Factory](https://docs.microsoft.com/en-us/azure/data-factory/concepts-integration-runtime) and [Linked service properties for Azure Blob storage](https://docs.microsoft.com/en-us/azure/data-factory/connector-azure-blob-storage#linked-service-properties). |
| Pipelines | A data factory can have one or more pipelines. A pipeline is a logical grouping of activities that together perform a task. You use startTime, endTime, and isPaused to schedule and run pipelines. | Pipelines are groups of activities that are performed on data. However, the scheduling of activities in the pipeline has been separated into new trigger resources. You can think of pipelines in the current version of Data Factory more as “workflow units” that you schedule separately via triggers.  Pipelines do not have “windows” of time execution in the current version of Data Factory. The Data Factory V1 concepts of startTime, endTime, and isPaused are no longer present in the current version of Data Factory. For more information, see [Pipeline execution and triggers](https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipeline-execution-triggers) and [Pipelines and activities](https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipelines-activities). |
| Activities | Activities define actions to perform on your data within a pipeline. Data movement (copy activity) and data transformation activities (such as Hive, Pig, and MapReduce) are supported. | In the current version of Data Factory, activities still are defined actions within a pipeline. The current version of Data Factory introduces new [control flow activities](https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipelines-activities#control-flow-activities). You use these activities in a control flow (looping and branching). Data movement and data transformation activities that were supported in V1 are supported in the current version. You can define transformation activities without using datasets in the current version. |
| Hybrid data movement and activity dispatch | Now called Integration Runtime, [Data Management Gateway](https://docs.microsoft.com/en-us/azure/data-factory/v1/data-factory-data-management-gateway) supported moving data between on-premises and cloud. | Data Management Gateway is now called Self-Hosted Integration Runtime. It provides the same capability as it did in V1.  The Azure-SSIS Integration Runtime in the current version of Data Factory also supports deploying and running SQL Server Integration Services (SSIS) packages in the cloud. For more information, see [Integration runtime in Azure Data Factory](https://docs.microsoft.com/en-us/azure/data-factory/concepts-integration-runtime). |
| Parameters | NA | Parameters are key-value pairs of read-only configuration settings that are defined in pipelines. You can pass arguments for the parameters when you are manually running the pipeline. If you are using a scheduler trigger, the trigger can pass values for the parameters too. Activities within the pipeline consume the parameter values. |
| Expressions | Data Factory V1 allows you to use functions and system variables in data selection queries and activity/dataset properties. | In the current version of Data Factory, you can use expressions anywhere in a JSON string value. For more information, see [Expressions and functions in the current version of Data Factory](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-expression-language-functions). |
| Pipeline runs | NA | A single instance of a pipeline execution. For example, say you have a pipeline that executes at 8 AM, 9 AM, and 10 AM. There would be three separate runs of the pipeline (pipeline runs) in this case. Each pipeline run has a unique pipeline run ID. The pipeline run ID is a GUID that uniquely defines that particular pipeline run. Pipeline runs are typically instantiated by passing arguments to parameters that are defined in the pipelines. |
| Activity runs | NA | An instance of an activity execution within a pipeline. |
| Trigger runs | NA | An instance of a trigger execution. For more information, see [Triggers](https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipeline-execution-triggers). |
| Scheduling | Scheduling is based on pipeline start/end times and dataset availability. | Scheduler trigger or execution via external scheduler. For more information, see [Pipeline execution and triggers](https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipeline-execution-triggers). |

The following table provides guidance on when to use the Copy Data tool vs. per-activity authoring in Data Factory UI:

| **TABLE 1** | |
| --- | --- |
| **Copy Data tool** | **Per activity (Copy activity) authoring** |
| You want to easily build a data loading task without learning about Azure Data Factory entities (linked services, datasets, pipelines, etc.) | You want to implement complex and flexible logic for loading data into lake. |
| You want to quickly load a large number of data artifacts into a data lake. | You want to chain Copy activity with subsequent activities for cleansing or processing data. |



To specify an expression for a property value, select **Add Dynamic Content** or click **Alt + P** while focusing on the field.

**Isolation Level**: The default for SQL sources in mapping data flow is read uncommitted. You can change the isolation level here to one of these values:

* Read Committed
* Read Uncommitted
* Repeatable Read
* Serializable
* None (ignore isolation level)

| **Metadata type** | **Description** |
| --- | --- |
| itemName | Name of the file or folder. |
| itemType | Type of the file or folder. Returned value is File or Folder. |
| size | Size of the file, in bytes. Applicable only to files. |
| created | Created datetime of the file or folder. |
| lastModified | Last modified datetime of the file or folder. |
| childItems | List of subfolders and files in the given folder. Applicable only to folders. Returned value is a list of the name and type of each child item. |
| contentMD5 | MD5 of the file. Applicable only to files. |
| structure | Data structure of the file or relational database table. Returned value is a list of column names and column types. |
| columnCount | Number of columns in the file or relational table. |
| exists | Whether a file, folder, or table exists. Note that if exists is specified in the Get Metadata field list, the activity won't fail even if the file, folder, or table doesn't exist. Instead, exists: false is returned in the output. |

A tumbling window trigger can depend on a maximum of five other triggers.

|  |  |
| --- | --- |
| offset | Offset of the dependency trigger. Provide a value in time span format and both negative and positive offsets are allowed. This property is mandatory if the trigger is depending on itself and in all other cases it is optional. Self-dependency should always be a negative offset. If no value specified, the window is the same as the trigger itself. |

## Gen1 vs Gen2 capabilities

This table compares the capabilities of Gen1 to that of Gen2.

| **GEN1 VS GEN2 CAPABILITIES** | | |
| --- | --- | --- |
| **Area** | **Gen1** | **Gen2** |
| Data organization | [Hierarchical namespace](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-namespace) File and folder support | [Hierarchical namespace](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-namespace) Container, file and folder support |
| Geo-redundancy | [LRS](https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy#locally-redundant-storage) | [LRS](https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy#locally-redundant-storage), [ZRS](https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy#zone-redundant-storage), [GRS](https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy#geo-redundant-storage), [RA-GRS](https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy#read-access-to-data-in-the-secondary-region) |
| Authentication | [AAD managed identity](https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/overview) [Service principals](https://docs.microsoft.com/en-us/azure/active-directory/develop/app-objects-and-service-principals) | [AAD managed identity](https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/overview) [Service principals](https://docs.microsoft.com/en-us/azure/active-directory/develop/app-objects-and-service-principals) [Shared Access Key](https://docs.microsoft.com/en-us/rest/api/storageservices/authorize-with-shared-key) |
| Authorization | Management - [RBAC](https://docs.microsoft.com/en-us/azure/role-based-access-control/overview) Data – [ACLs](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-access-control) | Management – [RBAC](https://docs.microsoft.com/en-us/azure/role-based-access-control/overview) Data - [ACLs](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-access-control), [RBAC](https://docs.microsoft.com/en-us/azure/role-based-access-control/overview) |
| Encryption – Data at rest | Server side – with [Microsoft-managed](https://docs.microsoft.com/en-us/azure/storage/common/storage-service-encryption?toc=/azure/storage/blobs/toc.json) or [customer-managed](https://docs.microsoft.com/en-us/azure/storage/common/encryption-customer-managed-keys?toc=/azure/storage/blobs/toc.json) keys | Server side – with [Microsoft-managed](https://docs.microsoft.com/en-us/azure/storage/common/storage-service-encryption?toc=/azure/storage/blobs/toc.json) or [customer-managed](https://docs.microsoft.com/en-us/azure/storage/common/encryption-customer-managed-keys?toc=/azure/storage/blobs/toc.json) keys |
| VNET Support | [VNET Integration](https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-network-security) | [Service Endpoints](https://docs.microsoft.com/en-us/azure/storage/common/storage-network-security?toc=/azure/storage/blobs/toc.json), [Private Endpoints](https://docs.microsoft.com/en-us/azure/storage/common/storage-private-endpoints) |
| Developer experience | [REST](https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-data-operations-rest-api), [.NET](https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-data-operations-net-sdk), [Java](https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-get-started-java-sdk), [Python](https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-data-operations-python), [PowerShell](https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-get-started-powershell), [Azure CLI](https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-get-started-cli-2.0) | Generally available - [REST](https://docs.microsoft.com/en-us/rest/api/storageservices/data-lake-storage-gen2), [.NET](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-directory-file-acl-dotnet), [Java](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-directory-file-acl-java), [Python](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-directory-file-acl-python) Public preview - [JavaScript](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-directory-file-acl-javascript), [PowerShell](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-directory-file-acl-powershell), [Azure CLI](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-directory-file-acl-cli) |
| Resource logs | Classic logs [Azure Monitor integrated](https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-diagnostic-logs) | [Classic logs](https://docs.microsoft.com/en-us/azure/storage/common/storage-analytics-logging) - Generally available Azure monitor integration – timeline TBD |
| Ecosystem | [HDInsight (3.6)](https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-hdinsight-hadoop-use-portal), [Azure Databricks (3.1 and above)](https://docs.databricks.com/data/data-sources/azure/azure-datalake.html), [SQL DW](https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-load-from-azure-data-lake-store), [ADF](https://docs.microsoft.com/en-us/azure/data-factory/load-azure-data-lake-store) | [HDInsight (3.6, 4.0)](https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-use-data-lake-storage-gen2), [Azure Databricks (5.1 and above)](https://docs.microsoft.com/en-us/azure/databricks/data/data-sources/azure/azure-datalake-gen2), [SQL DW](https://docs.microsoft.com/en-us/azure/azure-sql/database/vnet-service-endpoint-rule-overview), [ADF](https://docs.microsoft.com/en-us/azure/data-factory/load-azure-data-lake-storage-gen2) |

## Control flow activities

The following control flow activities are supported:

| **CONTROL FLOW ACTIVITIES** | |
| --- | --- |
| **Control activity** | **Description** |
| [Append Variable](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-append-variable-activity) | Add a value to an existing array variable. |
| [Execute Pipeline](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-execute-pipeline-activity) | Execute Pipeline activity allows a Data Factory pipeline to invoke another pipeline. |
| [Filter](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-filter-activity) | Apply a filter expression to an input array |
| [For Each](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-for-each-activity) | ForEach Activity defines a repeating control flow in your pipeline. This activity is used to iterate over a collection and executes specified activities in a loop. The loop implementation of this activity is similar to the Foreach looping structure in programming languages. |
| [Get Metadata](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-get-metadata-activity) | GetMetadata activity can be used to retrieve metadata of any data in Azure Data Factory. |
| [If Condition Activity](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-if-condition-activity) | The If Condition can be used to branch based on condition that evaluates to true or false. The If Condition activity provides the same functionality that an if statement provides in programming languages. It evaluates a set of activities when the condition evaluates to true and another set of activities when the condition evaluates to false. |
| [Lookup Activity](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-lookup-activity) | Lookup Activity can be used to read or look up a record/ table name/ value from any external source. This output can further be referenced by succeeding activities. |
| [Set Variable](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-set-variable-activity) | Set the value of an existing variable. |
| [Until Activity](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-until-activity) | Implements Do-Until loop that is similar to Do-Until looping structure in programming languages. It executes a set of activities in a loop until the condition associated with the activity evaluates to true. You can specify a timeout value for the until activity in Data Factory. |
| [Validation Activity](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-validation-activity) | Ensure a pipeline only continues execution if a reference dataset exists, meets a specified criteria, or a timeout has been reached. |
| [Wait Activity](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-wait-activity) | When you use a Wait activity in a pipeline, the pipeline waits for the specified time before continuing with execution of subsequent activities. |
| [Web Activity](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-web-activity) | Web Activity can be used to call a custom REST endpoint from a Data Factory pipeline. You can pass datasets and linked services to be consumed and accessed by the activity. |
| [Webhook Activity](https://docs.microsoft.com/en-us/azure/data-factory/control-flow-webhook-activity) | Using the webhook activity, call an endpoint, and pass a callback URL. The pipeline run waits for the callback to be invoked before proceeding to the next activity. |