Copy data from Azure Blob storage to a SQL Database by using the Copy Data tool

9/25/2020 • 5 minutes to read • Edit Online

APPLIES TO: Azure Data Factory Azure Synapse Analytics (Preview)

In this tutorial, you use the Azure portal to create a data factory. Then you use the Copy Data tool to create a pipeline that copies data from Azure Blob storage to a SQL Database.

> **NOTE**
> If you're new to Azure Data Factory, see Introduction to Azure Data Factory.

In this tutorial, you perform the following steps:

- Create a data factory.
- Use the Copy Data tool to create a pipeline.
- Monitor the pipeline and activity runs.

Prerequisites

- **Azure subscription**: If you don't have an Azure subscription, create a free account before you begin.
- **Azure Storage account**: Use Blob storage as the *source* data store. If you don't have an Azure Storage account, see the instructions in Create a storage account.
- **Azure SQL Database**: Use a SQL Database as the *sink* data store. If you don't have a SQL Database, see the instructions in Create a SQL Database.

Create a blob and a SQL table

Prepare your Blob storage and your SQL Database for the tutorial by performing these steps.

Create a source blob

1. Launch **Notepad**. Copy the following text and save it in a file named **inputEmp.txt** on your disk:

```
FirstName|LastName
John|Doe
Jane|Doe
```

2. Create a container named **adfv2tutorial** and upload the inputEmp.txt file to the container. You can use the Azure portal or various tools like Azure Storage Explorer to perform these tasks.

Create a sink SQL table

1. Use the following SQL script to create a table named **dbo.emp** in your SQL Database:

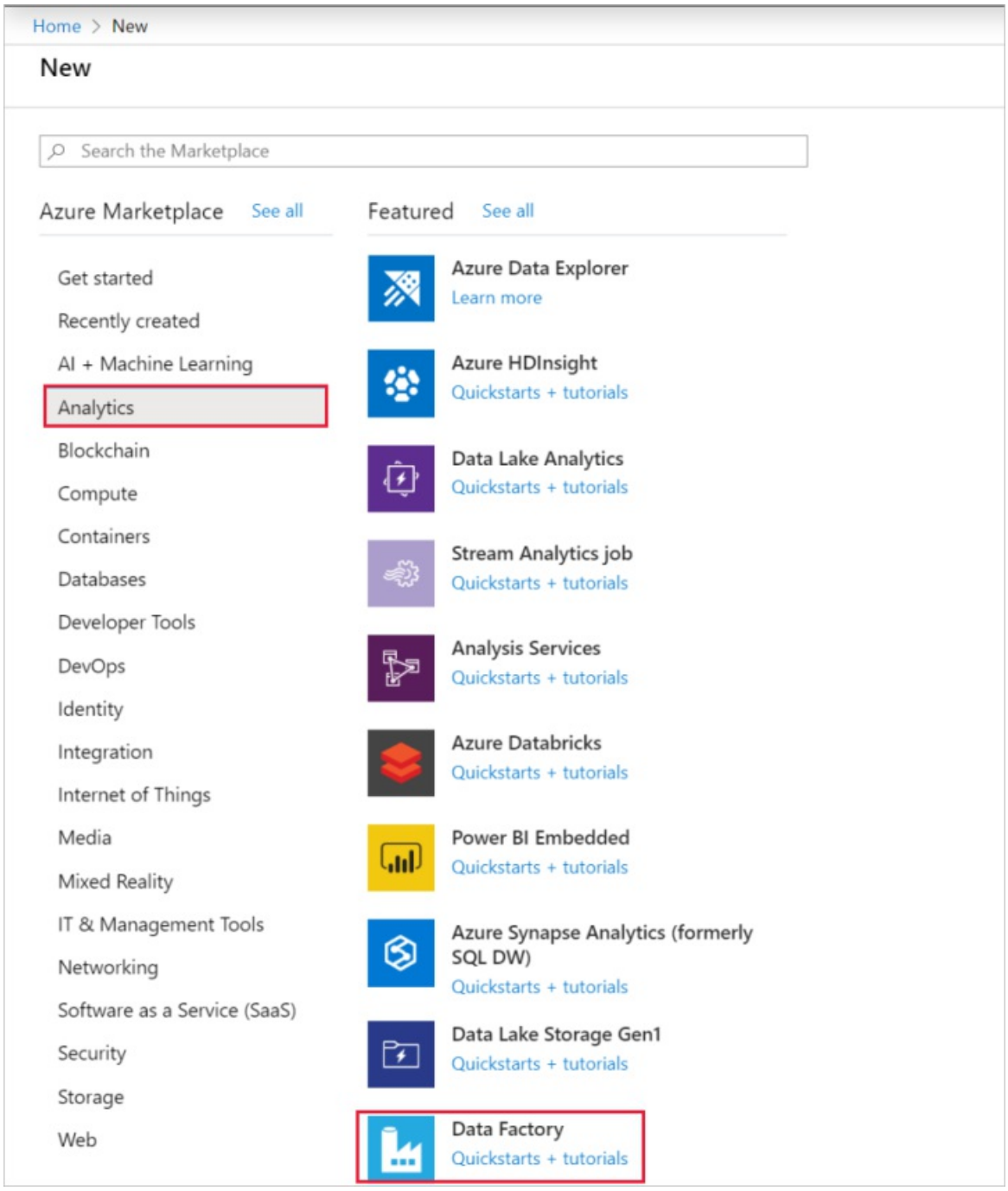
```
CREATE TABLE dbo.emp
(
    ID int IDENTITY(1,1) NOT NULL,
    FirstName varchar(50),
    LastName varchar(50)
)
GO

CREATE CLUSTERED INDEX IX_emp_ID ON dbo.emp (ID);
```

2. Allow Azure services to access SQL Server. Verify that the setting **Allow Azure services and resources to access this server** is enabled for your server that's running SQL Database. This setting lets Data Factory write data to your database instance. To verify and turn on this setting, go to logical SQL server > Security > Firewalls and virtual networks > set the **Allow Azure services and resources to access this server** option to **ON**.

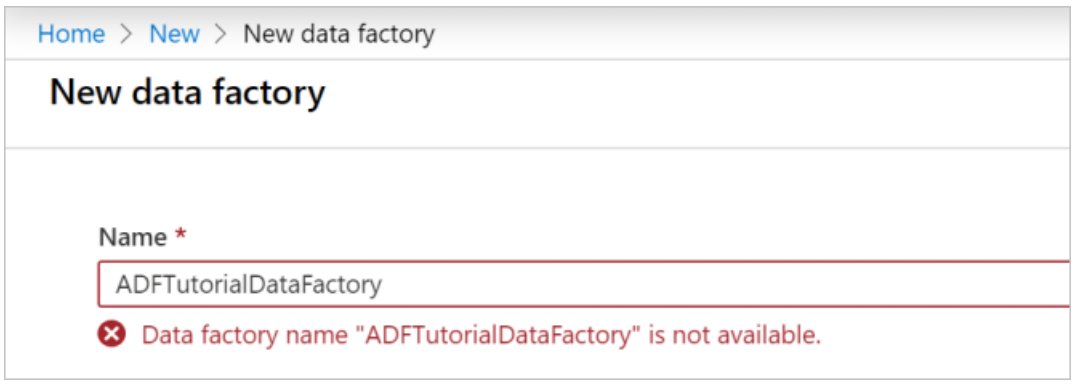
Create a data factory

1. On the left menu, select **Create a resource** > **Analytics** > **Data Factory**:



2. On the **New data factory** page, under **Name**, enter **ADFTutorialDataFactory**.

   The name for your data factory must be *globally unique*. You might receive the following error message:



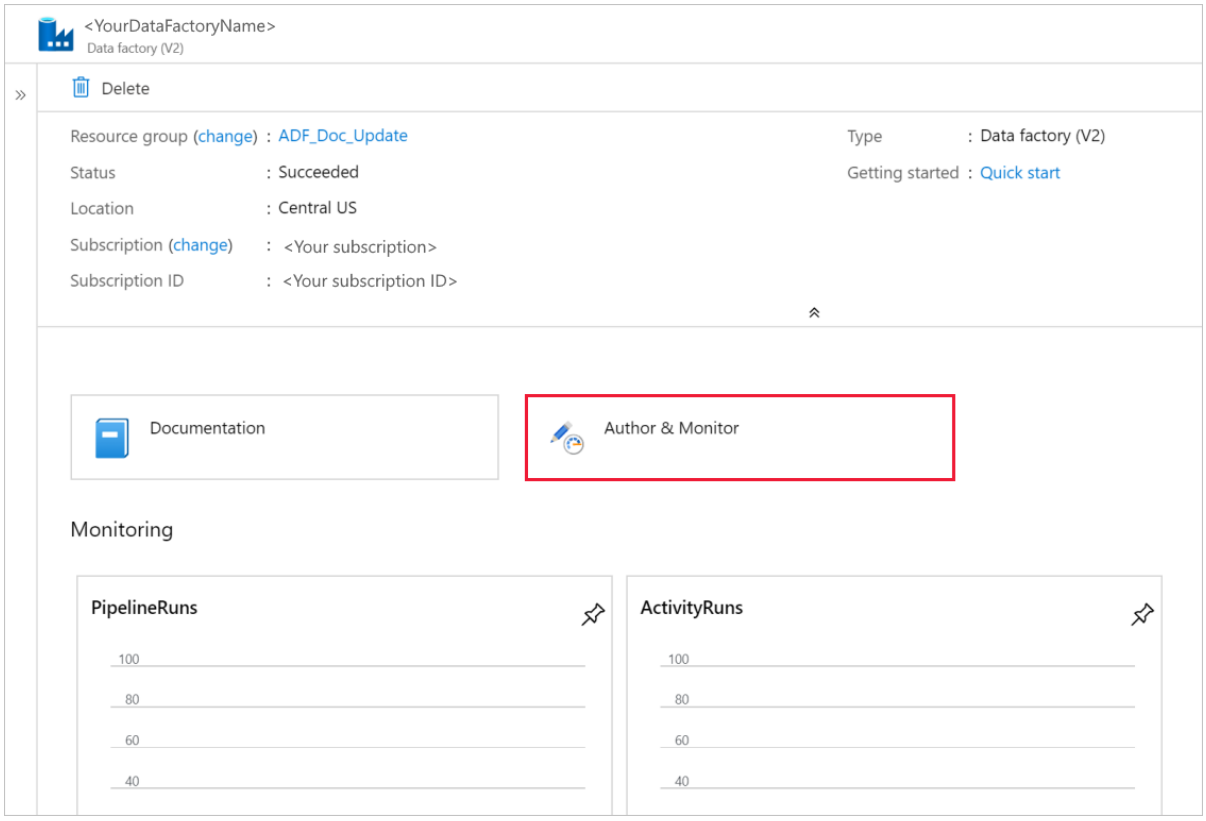
   If you receive an error message about the name value, enter a different name for the data factory. For example, use the name ***yourname*ADFTutorialDataFactory**. For the naming rules for Data Factory artifacts, see Data Factory naming rules.

3. Select the Azure **subscription** in which to create the new data factory.
4. For **Resource Group**, take one of the following steps:

   a. Select **Use existing**, and select an existing resource group from the drop-down list.

   b. Select **Create new**, and enter the name of a resource group.

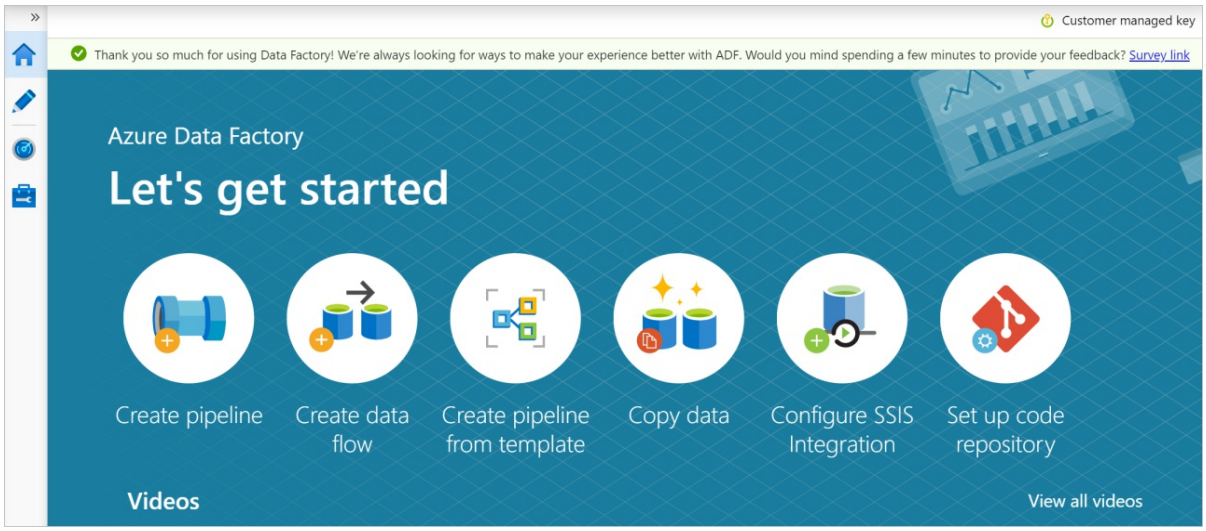
   To learn about resource groups, see Use resource groups to manage your Azure resources.
5. Under **version**, select **V2** for the version.
6. Under **location**, select the location for the data factory. Only supported locations are displayed in the drop-down list. The data stores (for example, Azure Storage and SQL Database) and computes (for example, Azure HDInsight) that are used by your data factory can be in other locations and regions.
7. Select **Create**.
8. After creation is finished, the **Data Factory** home page is displayed.



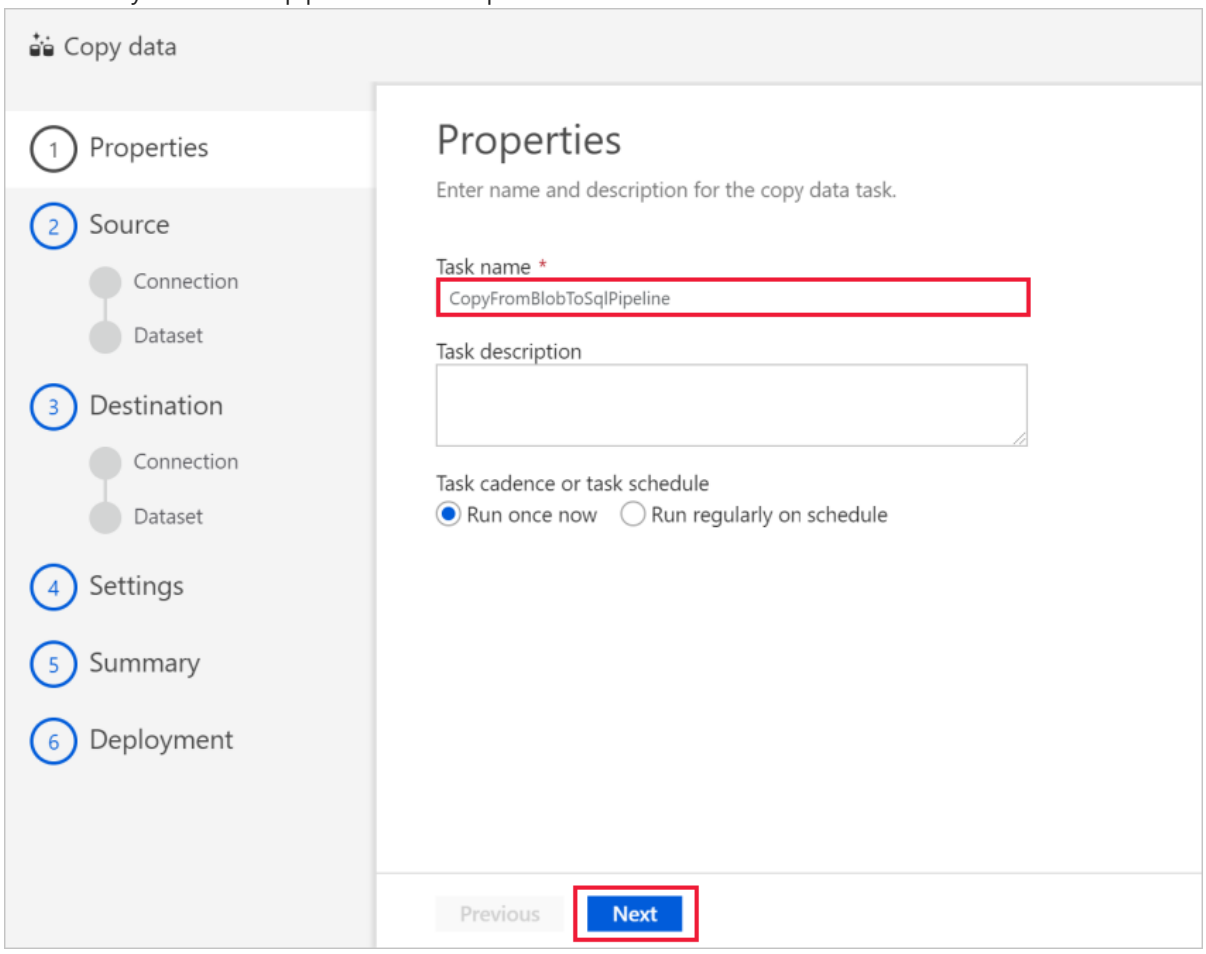
9. To launch the Azure Data Factory user interface (UI) in a separate tab, select the **Author & Monitor** tile.

Use the Copy Data tool to create a pipeline

1. On the **Let's get started** page, select the **Copy Data** tile to launch the Copy Data tool.



2. On the **Properties** page, under **Task name**, enter **CopyFromBlobToSqlPipeline**. Then select **Next**. The Data Factory UI creates a pipeline with the specified task name.



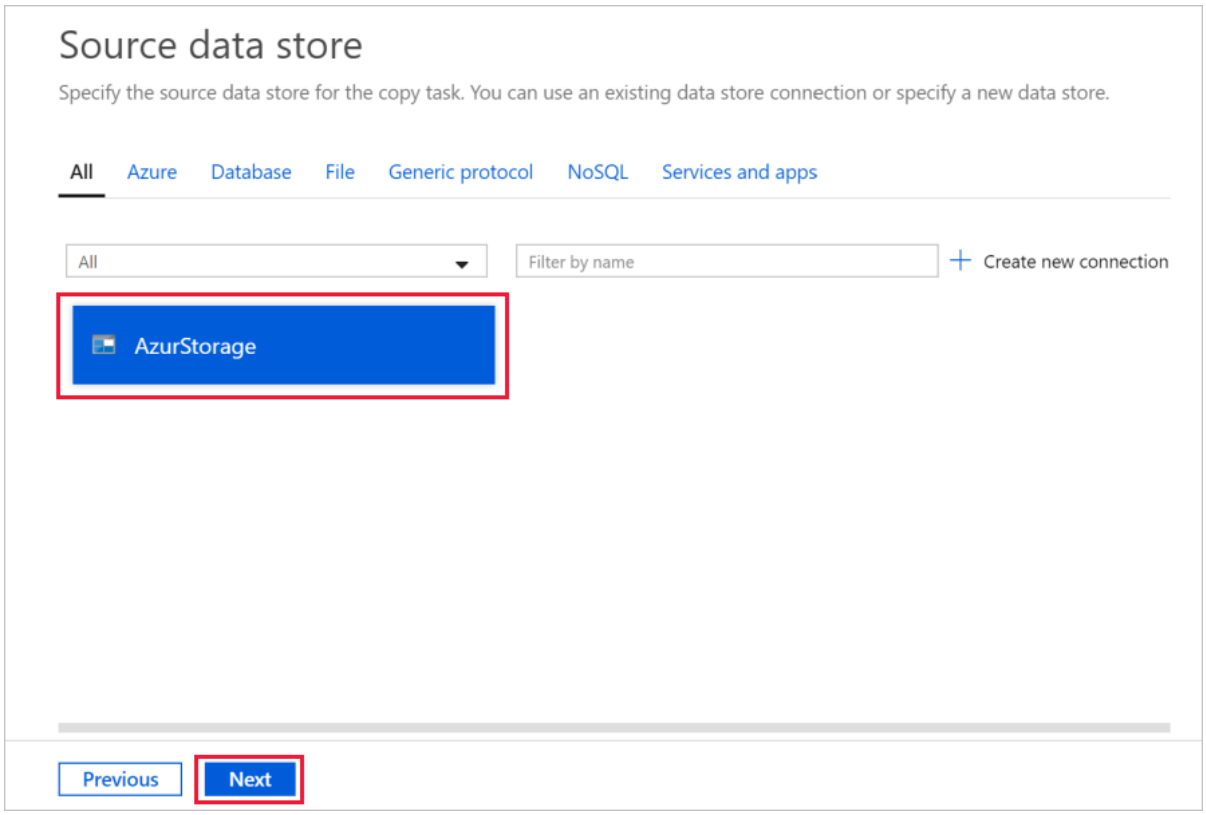
3. On the **Source data store** page, complete the following steps:

   a. Click **+ Create new connection** to add a connection

   b. Select **Azure Blob Storage** from the gallery, and then select **Continue**.

   c. On the **New Linked Service** page, select your Azure subscription, and select your storage account from the **Storage account name** list. Test connection and then select **Create**.

   d. Select the newly created linked service as source, then click **Next**.

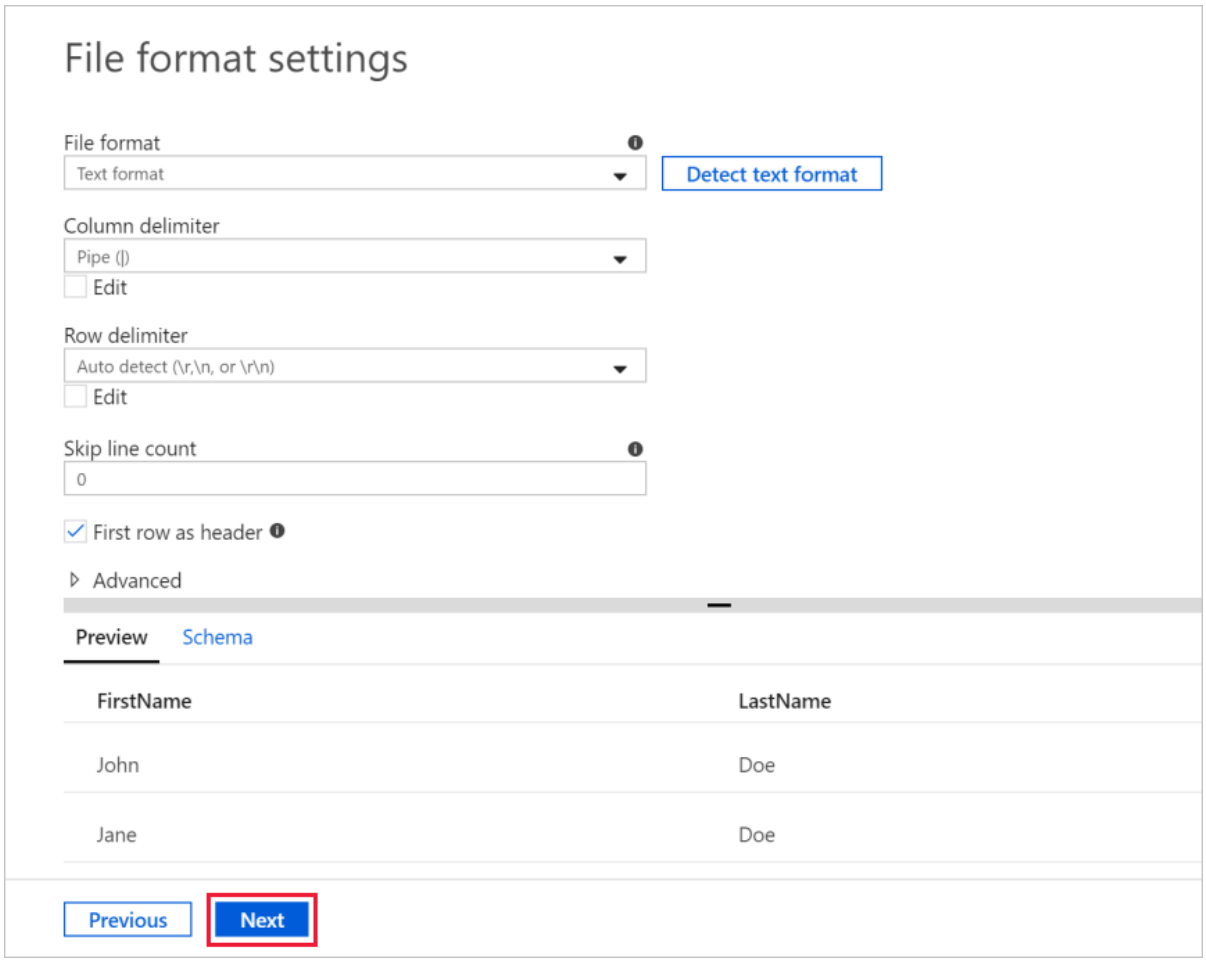


4. On the **Choose the input file or folder** page, complete the following steps:

   a. Click **Browse** to navigate to the **adfv2tutorial/input** folder, select the **inputEmp.txt** file, then click **Choose**.

   b. Click **Next** to move to next step.

5. On the **File format settings** page, enable the checkbox for *First row as header*. Notice that the tool automatically detects the column and row delimiters. Select **Next**. You can also preview data and view the schema of the input data on this page.

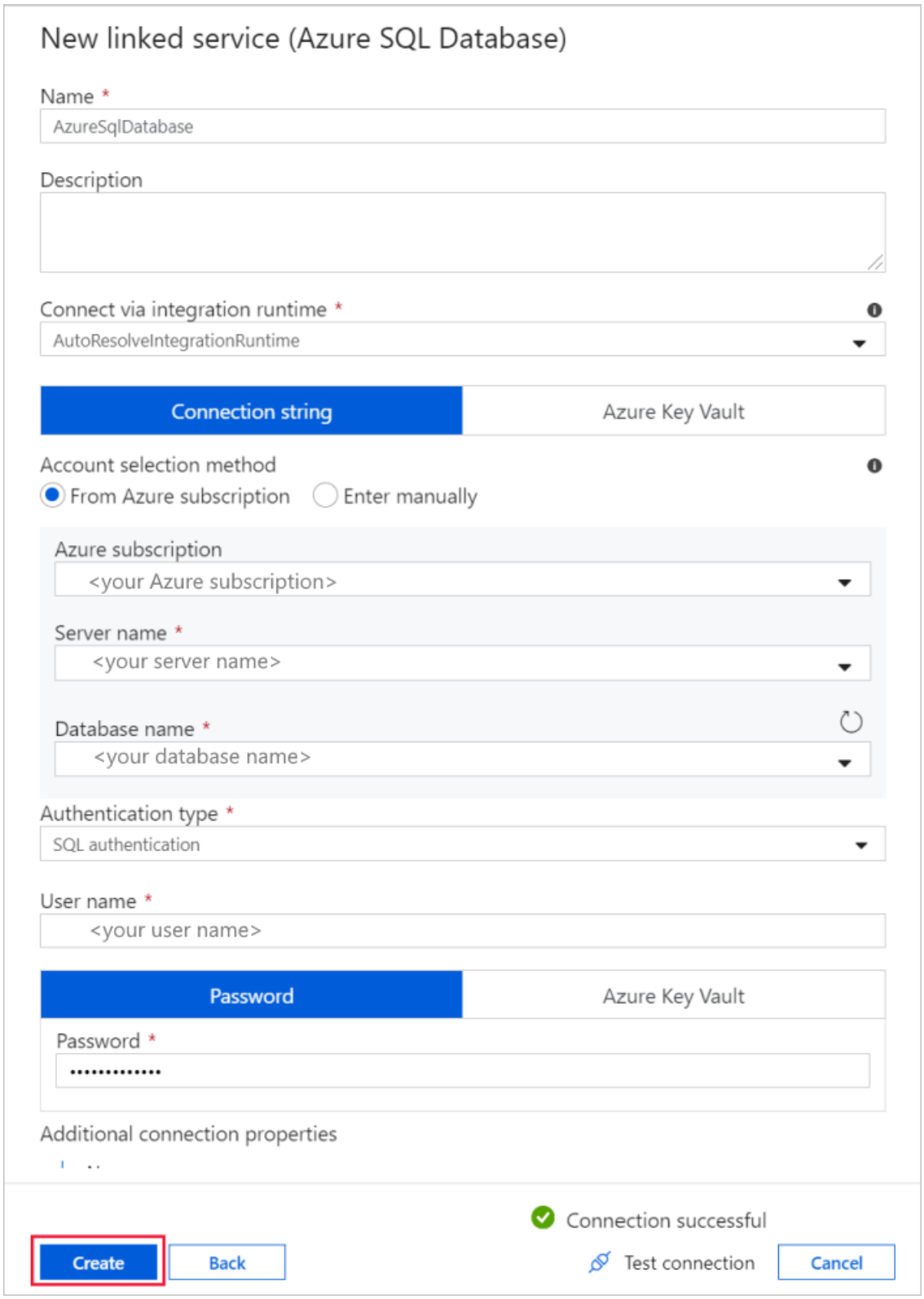


6. On the **Destination data store** page, completes the following steps:

   a. Click **+ Create new connection** to add a connection

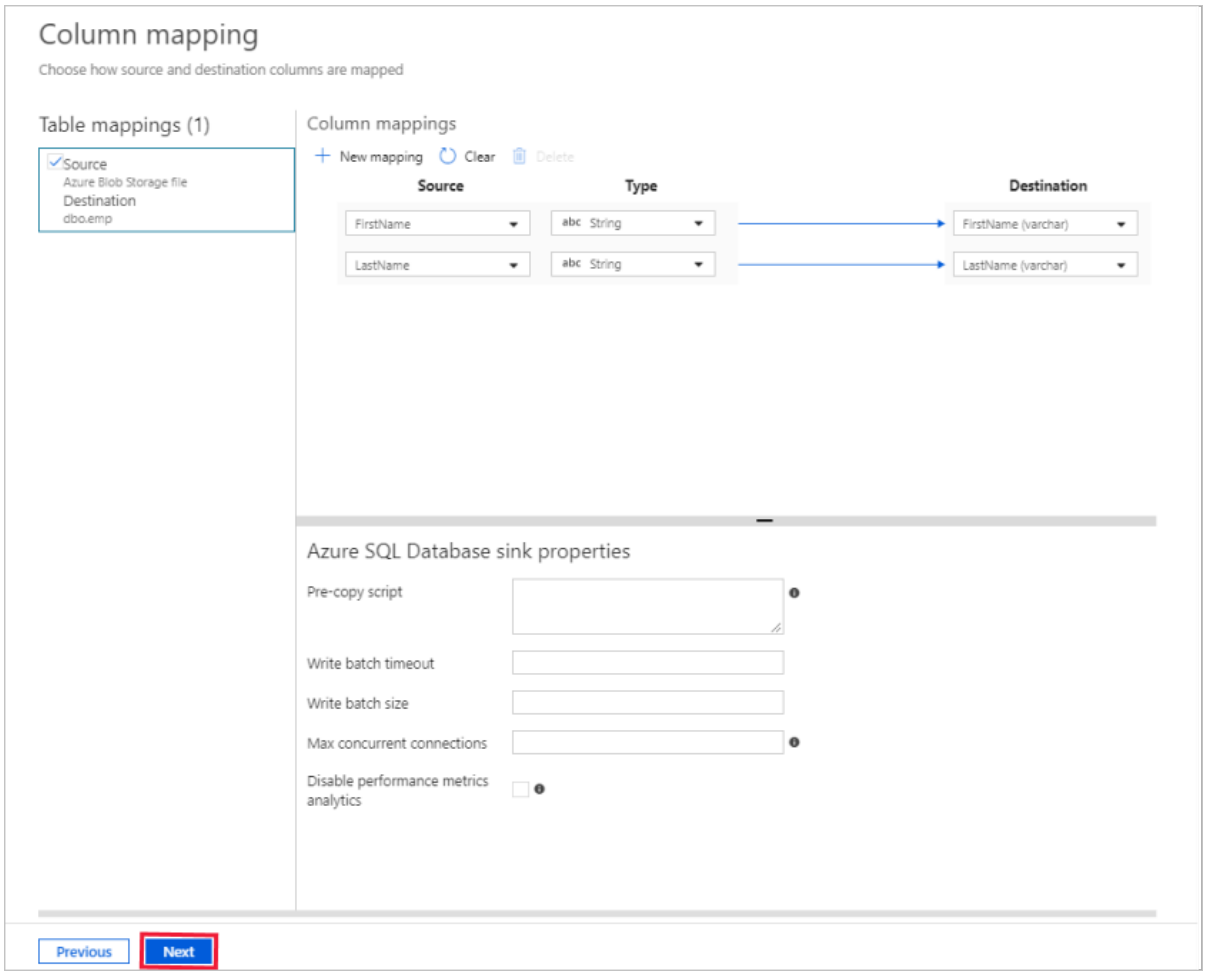
   b. Select **Azure SQL Database** from the gallery, and then select **Continue**.

   c. On the **New Linked Service** page, select your server name and DB name from the dropdown list, and specify the username and password, then select **Create**.

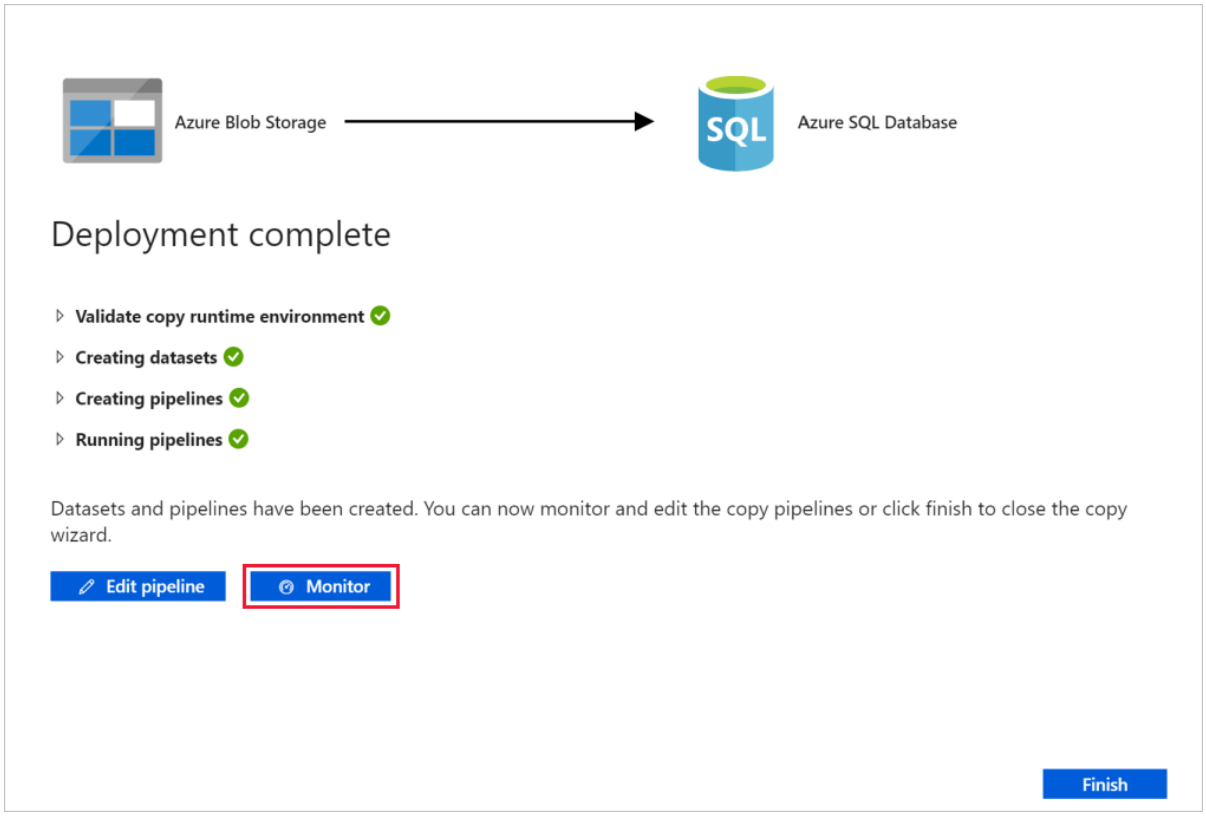


d. Select the newly created linked service as sink, then click **Next**.

7. On the **Table mapping** page, select the **[dbo].[emp]** table, and then select **Next**.
8. On the **Column mapping** page, notice that the second and the third columns in the input file are mapped to the **FirstName** and **LastName** columns of the **emp** table. Adjust the mapping to make sure that there is no error, and then select **Next**.

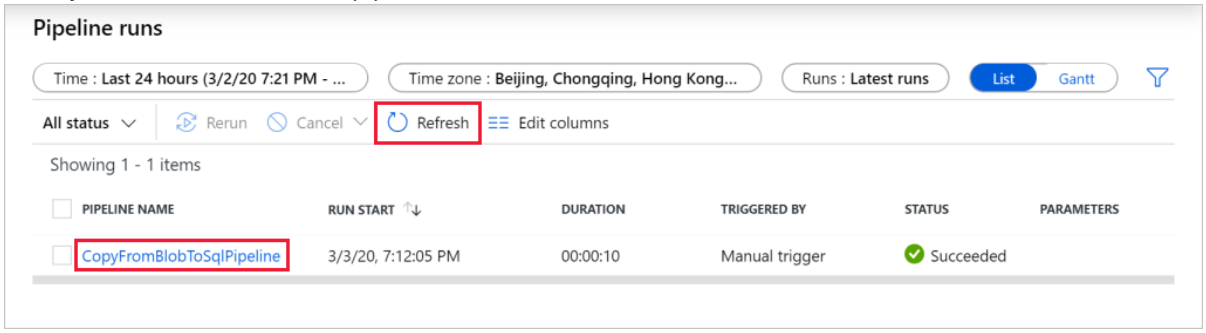


9. On the **Settings** page, select **Next**.
10. On the **Summary** page, review the settings, and then select **Next**.
11. On the **Deployment page**, select **Monitor** to monitor the pipeline (task).

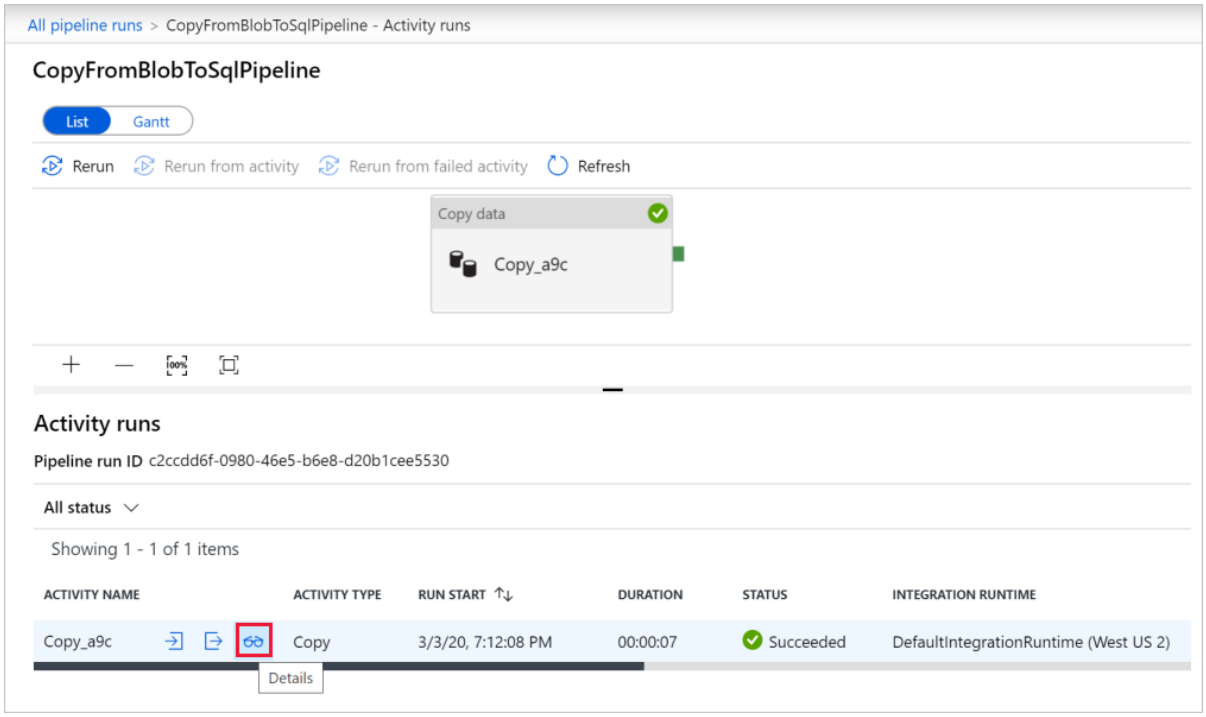


12. On the Pipeline runs page, select **Refresh** to refresh the list. Click the link under **PIPELINE NAME** to view

activity run details or rerun the pipeline.

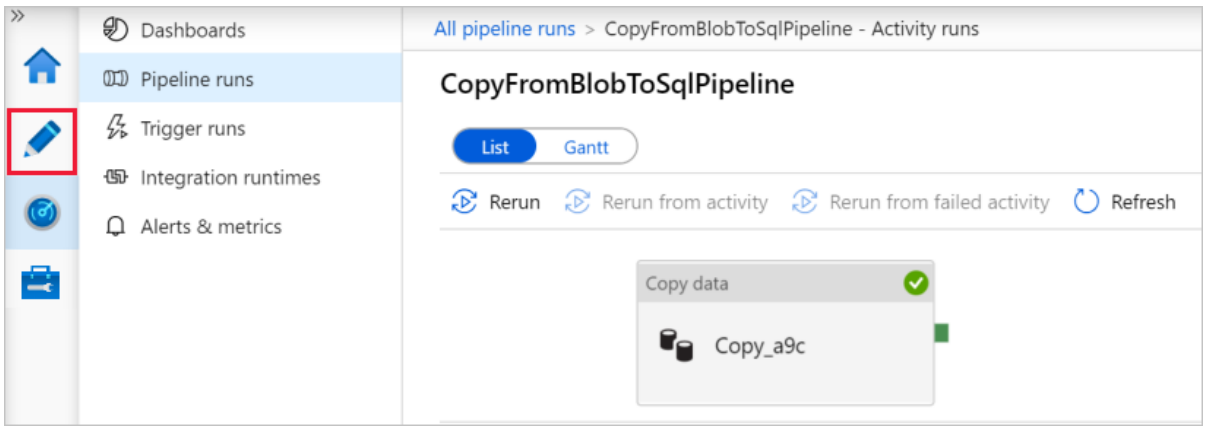


13. On the Activity runs page, select the **Details** link (eyeglasses icon) under the **ACTIVITY NAME** column for more details about copy operation. To go back to the Pipeline Runs view, select the **ALL pipeline runs** link in the breadcrumb menu. To refresh the view, select **Refresh**.



14. Verify that the data is inserted into the **dbo.emp** table in your SQL Database.

15. Select the **Author** tab on the left to switch to the editor mode. You can update the linked services, datasets, and pipelines that were created via the tool by using the editor. For details on editing these entities in the Data Factory UI, see the Azure portal version of this tutorial.



Next steps

The pipeline in this sample copies data from Blob storage to a SQL Database. You learned how to:

- Create a data factory.
- Use the Copy Data tool to create a pipeline.
- Monitor the pipeline and activity runs.

Advance to the following tutorial to learn how to copy data from on-premises to the cloud:

Copy data from on-premises to the cloud

Copy data from Azure Blob storage to a database in Azure SQL Database by using Azure Data Factory

9/25/2020 • 10 minutes to read • Edit Online

APPLIES TO: Azure Data Factory Azure Synapse Analytics (Preview)

In this tutorial, you create a data factory by using the Azure Data Factory user interface (UI). The pipeline in this data factory copies data from Azure Blob storage to a database in Azure SQL Database. The configuration pattern in this tutorial applies to copying from a file-based data store to a relational data store. For a list of data stores supported as sources and sinks, see the supported data stores table.

> **NOTE**
> - If you're new to Data Factory, see Introduction to Azure Data Factory.

In this tutorial, you perform the following steps:

- Create a data factory.
- Create a pipeline with a copy activity.
- Test run the pipeline.
- Trigger the pipeline manually.
- Trigger the pipeline on a schedule.
- Monitor the pipeline and activity runs.

Prerequisites

- **Azure subscription**. If you don't have an Azure subscription, create a free Azure account before you begin.
- **Azure storage account**. You use Blob storage as a *source* data store. If you don't have a storage account, see Create an Azure storage account for steps to create one.
- **Azure SQL Database**. You use the database as a *sink* data store. If you don't have a database in Azure SQL Database, see the Create a database in Azure SQL Database for steps to create one.

Create a blob and a SQL table

Now, prepare your Blob storage and SQL database for the tutorial by performing the following steps.

Create a source blob

1. Launch Notepad. Copy the following text, and save it as an **emp.txt** file on your disk:

```
FirstName,LastName
John,Doe
Jane,Doe
```

2. Create a container named **adftutorial** in your Blob storage. Create a folder named **input** in this container. Then, upload the **emp.txt** file to the **input** folder. Use the Azure portal or tools such as Azure Storage Explorer to do these tasks.

Create a sink SQL table

1. Use the following SQL script to create the **dbo.emp** table in your database:

```
CREATE TABLE dbo.emp
(
    ID int IDENTITY(1,1) NOT NULL,
    FirstName varchar(50),
    LastName varchar(50)
)
GO

CREATE CLUSTERED INDEX IX_emp_ID ON dbo.emp (ID);
```

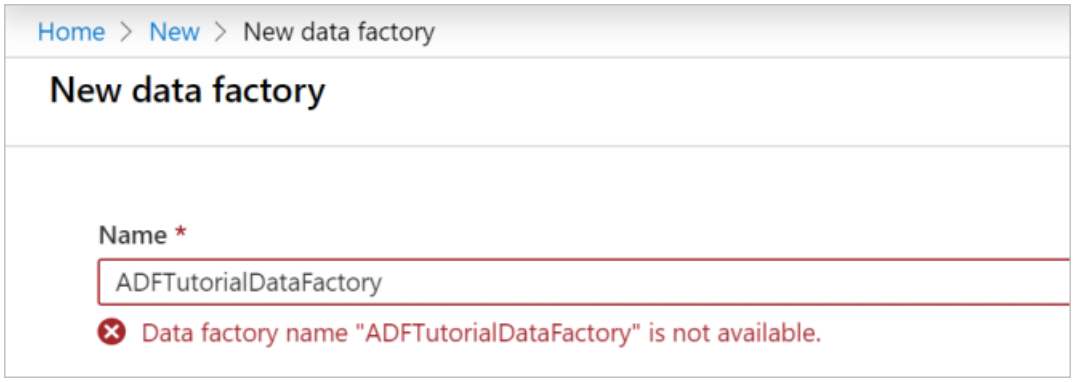
2. Allow Azure services to access SQL Server. Ensure that **Allow access to Azure services** is turned **ON** for your SQL Server so that Data Factory can write data to your SQL Server. To verify and turn on this setting, go to logical SQL server > Overview > Set server firewall> set the **Allow access to Azure services** option to **ON**.

Create a data factory

In this step, you create a data factory and start the Data Factory UI to create a pipeline in the data factory.

1. Open **Microsoft Edge** or **Google Chrome**. Currently, Data Factory UI is supported only in Microsoft Edge and Google Chrome web browsers.
2. On the left menu, select **Create a resource** > **Analytics** > **Data Factory**.
3. On the **New data factory** page, under **Name**, enter **ADFTutorialDataFactory**.

   The name of the Azure data factory must be *globally unique*. If you receive an error message about the name value, enter a different name for the data factory. (for example, yournameADFTutorialDataFactory). For naming rules for Data Factory artifacts, see Data Factory naming rules.



4. Select the Azure **subscription** in which you want to create the data factory.
5. For **Resource Group**, take one of the following steps:

   a. Select **Use existing**, and select an existing resource group from the drop-down list.

   b. Select **Create new**, and enter the name of a resource group.

   To learn about resource groups, see Use resource groups to manage your Azure resources.
6. Under **Version**, select **V2**.
7. Under **Location**, select a location for the data factory. Only locations that are supported are displayed in the drop-down list. The data stores (for example, Azure Storage and SQL Database) and computes (for example, Azure HDInsight) used by the data factory can be in other regions.
8. Select **Create**.
9. After the creation is finished, you see the notice in Notifications center. Select **Go to resource** to navigate

to the Data factory page.

10. Select **Author & Monitor** to launch the Data Factory UI in a separate tab.

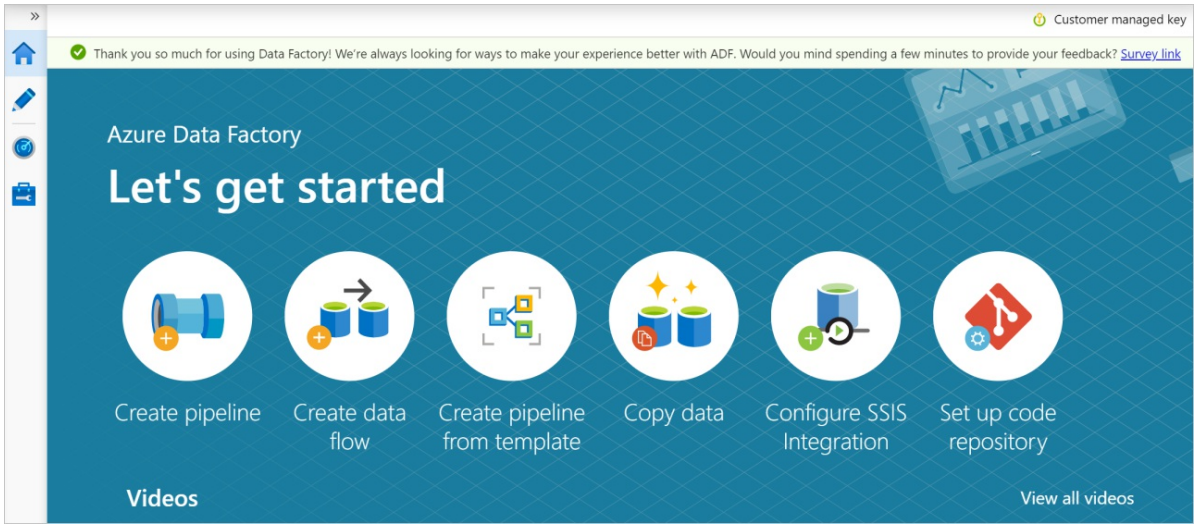
Create a pipeline

In this step, you create a pipeline with a copy activity in the data factory. The copy activity copies data from Blob storage to SQL Database. In the Quickstart tutorial, you created a pipeline by following these steps:

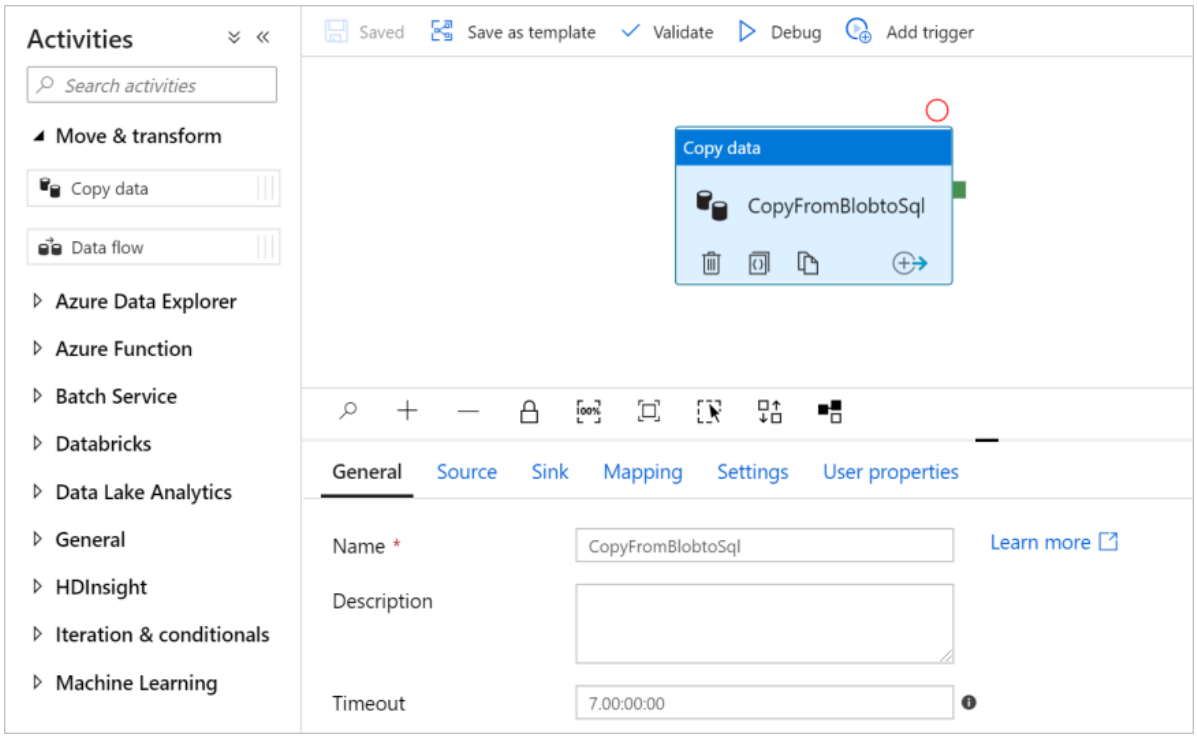
1. Create the linked service.
2. Create input and output datasets.
3. Create a pipeline.

In this tutorial, you start with creating the pipeline. Then you create linked services and datasets when you need them to configure the pipeline.

1. On the **Let's get started** page, select **Create pipeline**.



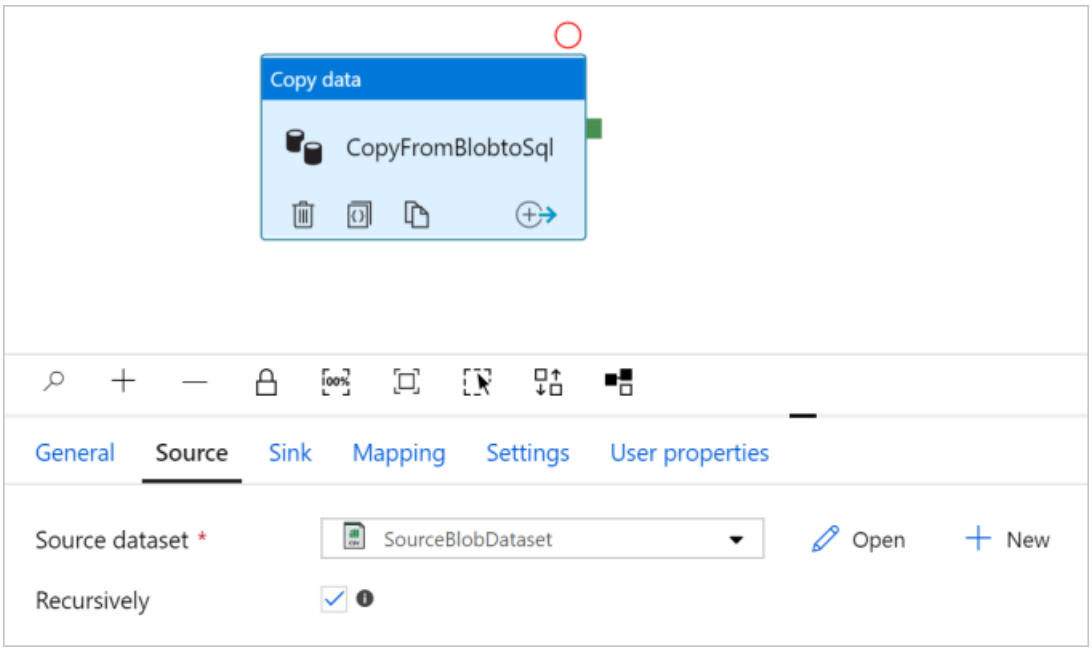
2. a. In the General panel under **Properties**, specify **CopyPipeline** for **Name**. Then collapse the panel by clicking the Properties icon in the top-right corner.
3. In the **Activities** tool box, expand the **Move and Transform** category, and drag and drop the **Copy Data** activity from the tool box to the pipeline designer surface. Specify **CopyFromBlobToSql** for **Name**.



Configure source

> **TIP**
>
> In this tutorial, you use *Account key* as the authentication type for your source data store, but you can choose other supported authentication methods: *SAS URI*,*Service Principal* and *Managed Identity* if needed. Refer to corresponding sections in this article for details. To store secrets for data stores securely, it's also recommended to use an Azure Key Vault. Refer to this article for detailed illustrations.

1. Go to the **Source** tab. Select **+ New** to create a source dataset.
2. In the **New Dataset** dialog box, select **Azure Blob Storage**, and then select **Continue**. The source data is in Blob storage, so you select **Azure Blob Storage** for the source dataset.
3. In the **Select Format** dialog box, choose the format type of your data, and then select **Continue**.
4. In the **Set Properties** dialog box, enter **SourceBlobDataset** for Name. Select the checkbox for **First row as header**. Under the **Linked service** text box, select **+ New**.
5. In the **New Linked Service (Azure Blob Storage)** dialog box, enter **AzureStorageLinkedService** as name, select your storage account from the **Storage account name** list. Test connection, select **Create** to deploy the linked service.
6. After the linked service is created, it's navigated back to the **Set properties** page. Next to **File path**, select **Browse**.
7. Navigate to the **adftutorial/input** folder, select the **emp.txt** file, and then select **OK**.
8. Select **OK**. It automatically navigates to the pipeline page. In **Source** tab, confirm that **SourceBlobDataset** is selected. To preview data on this page, select **Preview data**.



Configure sink

> **TIP**
> In this tutorial, you use *SQL authentication* as the authentication type for your sink data store, but you can choose other supported authentication methods: *Service Principal* and *Managed Identity* if needed. Refer to corresponding sections in this article for details. To store secrets for data stores securely, it's also recommended to use an Azure Key Vault. Refer to this article for detailed illustrations.

1. Go to the **Sink** tab, and select **+ New** to create a sink dataset.
2. In the **New Dataset** dialog box, input "SQL" in the search box to filter the connectors, select **Azure SQL Database**, and then select **Continue**. In this tutorial, you copy data to a SQL database.
3. In the **Set Properties** dialog box, enter **OutputSqlDataset** for Name. From the **Linked service** dropdown list, select **+ New**. A dataset must be associated with a linked service. The linked service has the connection string that Data Factory uses to connect to SQL Database at runtime. The dataset specifies the container, folder, and the file (optional) to which the data is copied.
4. In the **New Linked Service (Azure SQL Database)** dialog box, take the following steps:

   a. Under **Name**, enter **AzureSqlDatabaseLinkedService**.

   b. Under **Server name**, select your SQL Server instance.

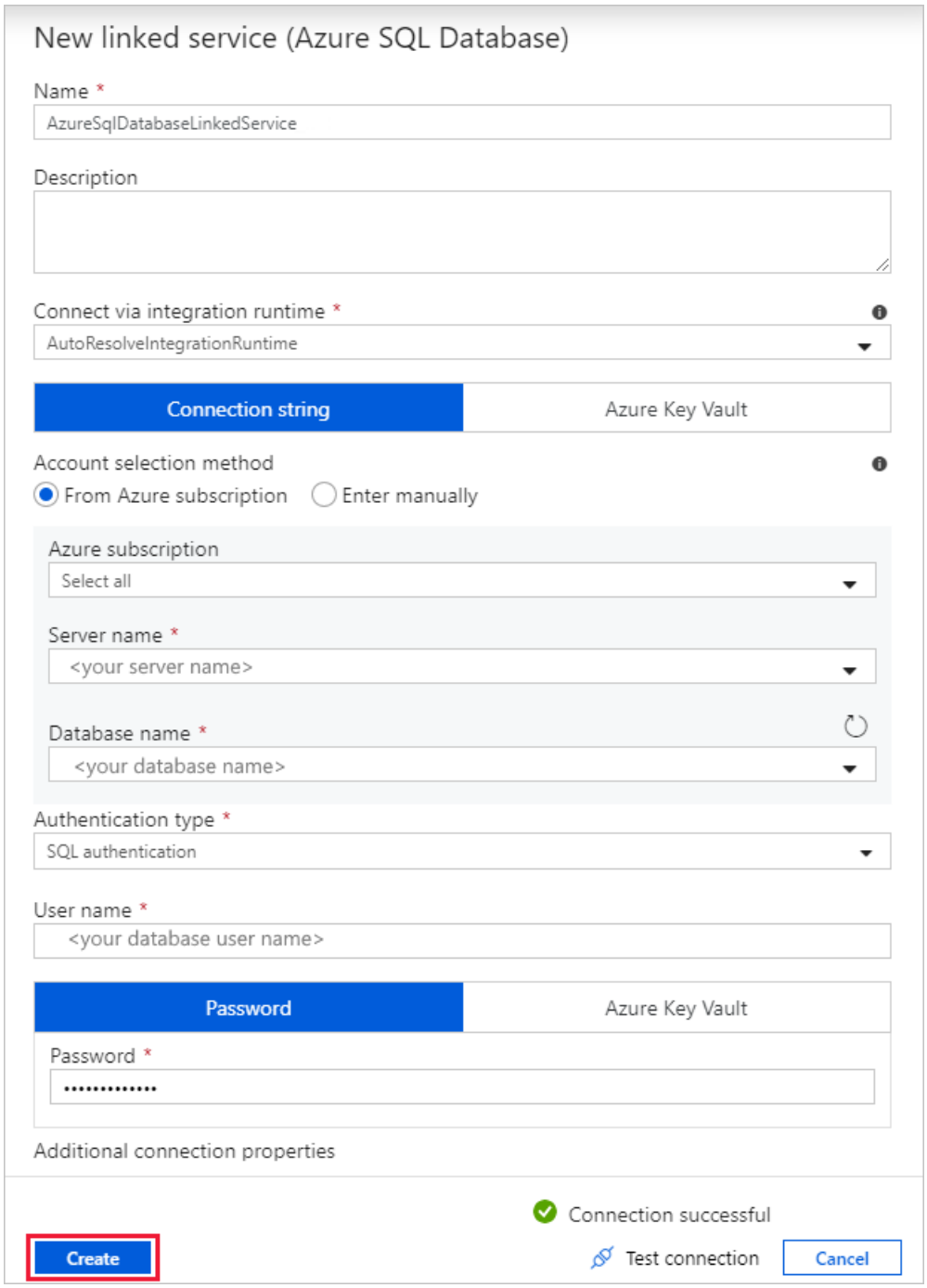
   c. Under **Database name**, select your database.

   d. Under **User name**, enter the name of the user.

   e. Under **Password**, enter the password for the user.

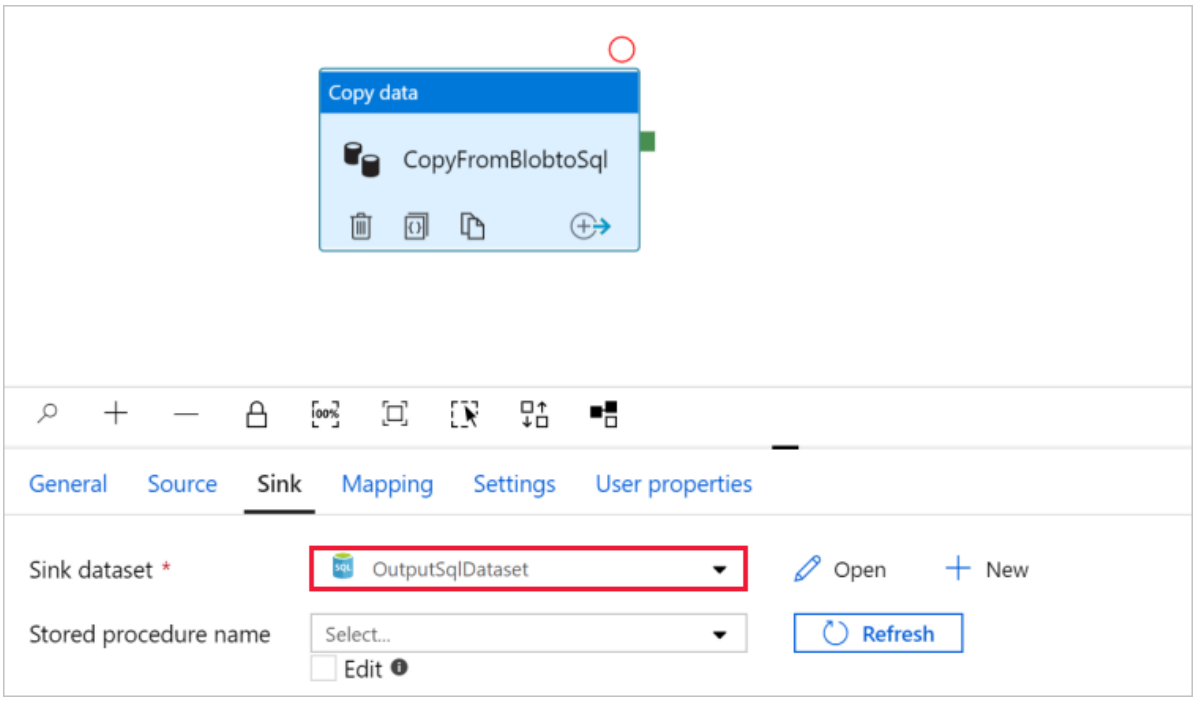
   f. Select **Test connection** to test the connection.

   g. Select **Create** to deploy the linked service.



5. It automatically navigates to the **Set Properties** dialog box. In **Table**, select **[dbo].[emp]**. Then select **OK**.

6. Go to the tab with the pipeline, and in **Sink Dataset**, confirm that **OutputSqlDataset** is selected.



You can optionally map the schema of the source to corresponding schema of destination by following Schema mapping in copy activity.

Validate the pipeline

To validate the pipeline, select **Validate** from the tool bar.

You can see the JSON code associated with the pipeline by clicking **Code** on the upper right.

Debug and publish the pipeline

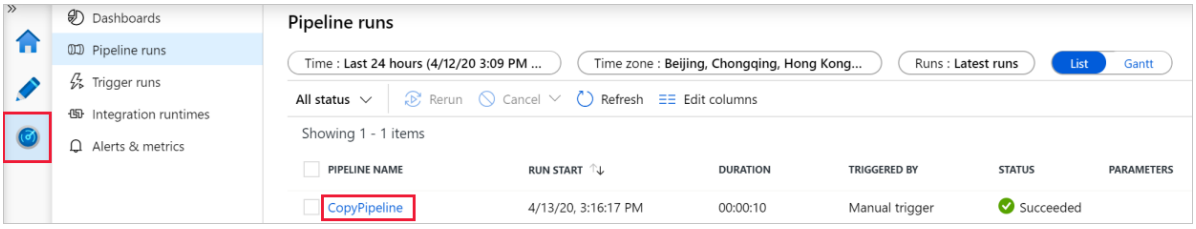
You can debug a pipeline before you publish artifacts (linked services, datasets, and pipeline) to Data Factory or your own Azure Repos Git repository.

1. To debug the pipeline, select **Debug** on the toolbar. You see the status of the pipeline run in the **Output** tab at the bottom of the window.
2. Once the pipeline can run successfully, in the top toolbar, select **Publish all**. This action publishes entities (datasets, and pipelines) you created to Data Factory.
3. Wait until you see the **Successfully published** message. To see notification messages, click the **Show Notifications** on the top-right (bell button).

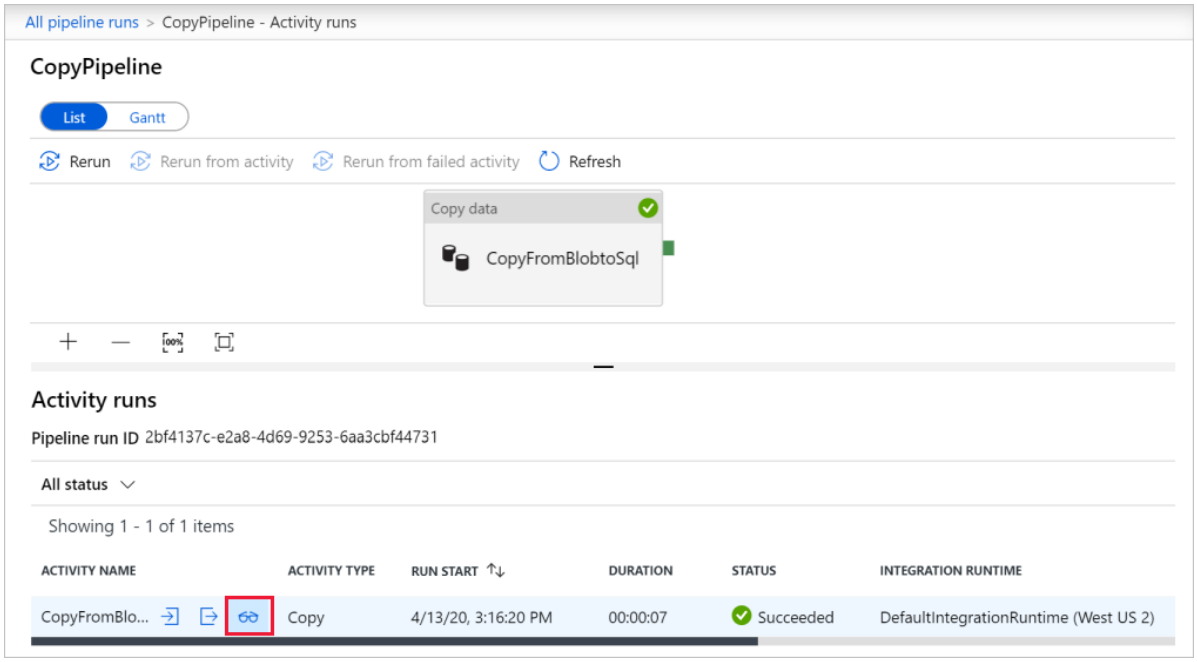
Trigger the pipeline manually

In this step, you manually trigger the pipeline you published in the previous step.

1. Select **Trigger** on the toolbar, and then select **Trigger Now**. On the **Pipeline Run** page, select **OK**.
2. Go to the **Monitor** tab on the left. You see a pipeline run that is triggered by a manual trigger. You can use links under the **PIPELINE NAME** column to view activity details and to rerun the pipeline.



3. To see activity runs associated with the pipeline run, select the **CopyPipeline** link under the **PIPELINE NAME** column. In this example, there's only one activity, so you see only one entry in the list. For details about the copy operation, select the **Details** link (eyeglasses icon) under the **ACTIVITY NAME** column. Select **All pipeline runs** at the top to go back to the Pipeline Runs view. To refresh the view, select **Refresh**.



4. Verify that two more rows are added to the **emp** table in the database.

Trigger the pipeline on a schedule

In this schedule, you create a schedule trigger for the pipeline. The trigger runs the pipeline on the specified schedule, such as hourly or daily. Here you set the trigger to run every minute until the specified end datetime.

1. Go to the **Author** tab on the left above the monitor tab.
2. Go to your pipeline, click **Trigger** on the tool bar, and select **New/Edit**.
3. In the **Add triggers** dialog box, select **+ New** for **Choose trigger** area.
4. In the **New Trigger** window, take the following steps:

   a. Under **Name**, enter **RunEveryMinute**.

   b. Under **End**, select **On Date**.

   c. Under **End On**, select the drop-down list.

   d. Select the **current day** option. By default, the end day is set to the next day.

   e. Update the **End Time** part to be a few minutes past the current datetime. The trigger is activated only after you publish the changes. If you set it to only a couple of minutes apart, and you don't publish it by then, you don't see a trigger run.

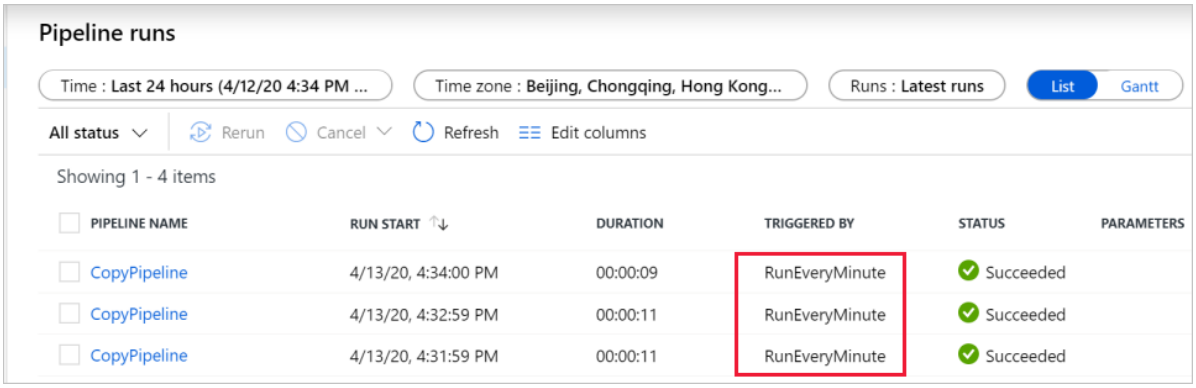
   f. Select **OK**.

   g. For **Activated** option, select **Yes**.

   h. Select **OK**.

> **IMPORTANT**
> A cost is associated with each pipeline run, so set the end date appropriately.

5. On the **Edit trigger** page, review the warning, and then select **Save**. The pipeline in this example doesn't take any parameters.
6. Click **Publish all** to publish the change.
7. Go to the **Monitor** tab on the left to see the triggered pipeline runs.



8. To switch from the **Pipeline Runs** view to the **Trigger Runs** view, select **Trigger Runs** on the left side of the window.
9. You see the trigger runs in a list.
10. Verify that two rows per minute (for each pipeline run) are inserted into the **emp** table until the specified end time.

Next steps

The pipeline in this sample copies data from one location to another location in Blob storage. You learned how to:

- Create a data factory.
- Create a pipeline with a copy activity.
- Test run the pipeline.
- Trigger the pipeline manually.
- Trigger the pipeline on a schedule.
- Monitor the pipeline and activity runs.

Advance to the following tutorial to learn how to copy data from on-premises to the cloud:

Copy data from on-premises to the cloud