

[Before you start](#)

[Provision Azure resources](#)

[Import a notebook](#)

[Enable Azure Databricks integration with Azure Data Factory.](#)

[Use a pipeline to run the Azure Databricks notebook](#)

[Delete Azure Databricks resources](#)

Automate an Azure Databricks Notebook with Azure Data Factory

You can use notebooks in Azure Databricks to perform data engineering tasks, such as processing data files and loading data into tables. When you need to orchestrate these tasks as part of a data engineering pipeline, you can use Azure Data Factory.

This lab will take approximately **40** minutes to complete.

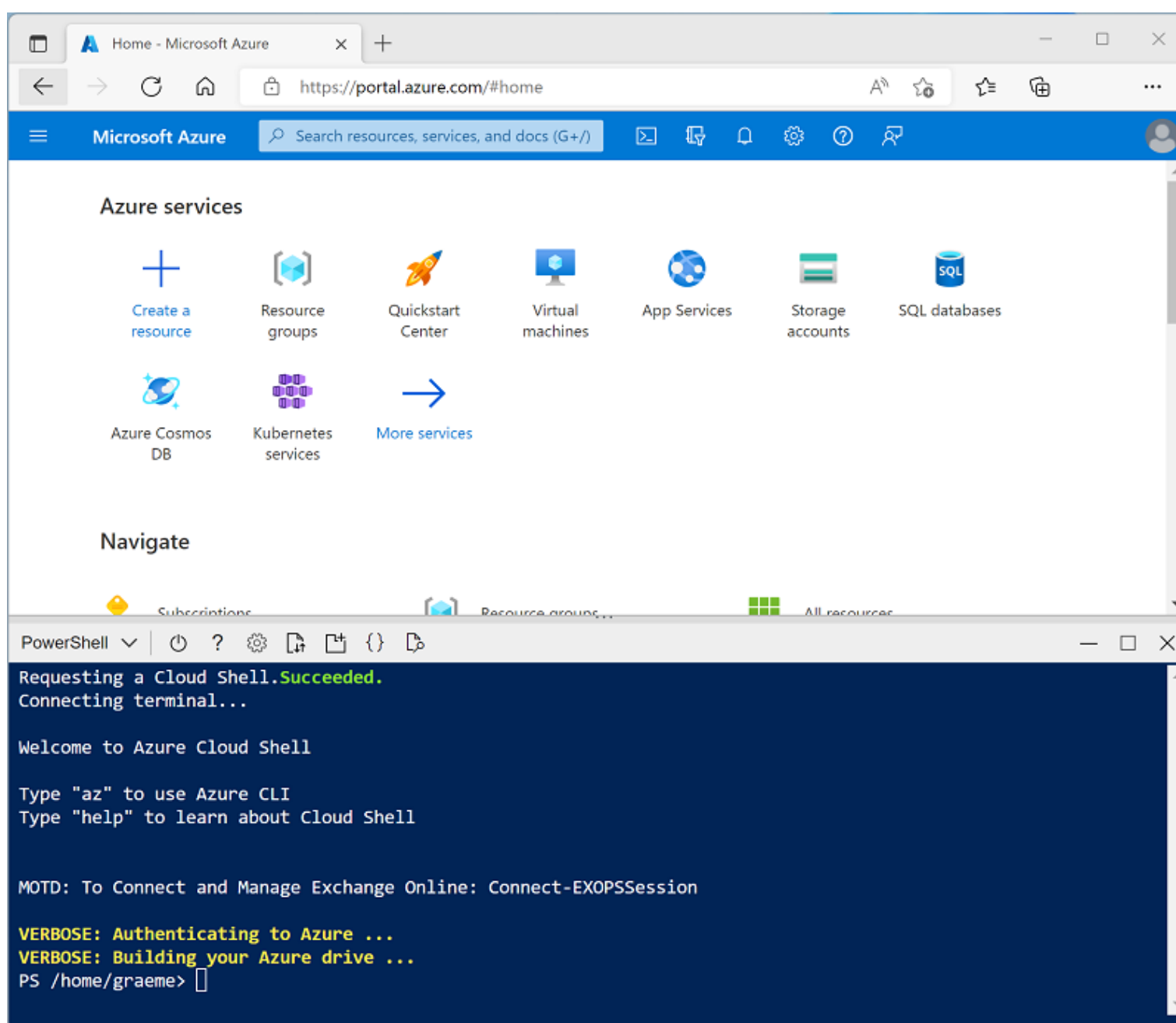
Before you start

You'll need an [Azure subscription](#) in which you have administrative-level access.




Provision Azure resources


In this exercise, you'll use a script to provision a new Azure Databricks workspace and an Azure Data Factory resource in your Azure subscription.

1. In a web browser, sign into the [Azure portal](#) at `https://portal.azure.com`.
2. Use the `[>]` button to the right of the search bar at the top of the page to create a new Cloud Shell in the Azure portal, selecting a **PowerShell** environment and creating storage if prompted. The cloud shell provides a command line interface in a pane at the bottom of the Azure portal, as shown here:




! **Note:** If you have previously created a cloud shell that uses a *Bash* environment, use the the drop-down menu at the top left of the cloud shell pane to change it to **PowerShell**.

3. Note that you can resize the cloud shell by dragging the separator bar at the top of the pane, or by using the , , and  icons at the top right of the pane to minimize, maximize, and close the pane. For more information about using the Azure Cloud Shell, see the [Azure Cloud Shell documentation](#).
4. In the PowerShell pane, enter the following commands to clone this repo:

Code  Copy

```
rm -r dp-000 -f
git clone https://github.com/MicrosoftLearning/mslearn-databricks dp-000
```

5. After the repo has been cloned, enter the following commands to change to the folder for this lab and run the **setup.ps1** script it contains:



Code  Copy

```
cd dp-000/Allfiles/Labs/05
./setup.ps1
```

6. If prompted, choose which subscription you want to use (this will only happen if you have access to multiple Azure subscriptions).
7. Wait for the script to complete - this typically takes around 5 minutes, but in some cases may take longer. While you are waiting, review [What is Azure Data Factory?](#).
8. When the script has completed, close the cloud shell pane and browse to the **dp000-xxxxxxx** resource group that was created by the script to verify that it contains an Azure Databricks workspace and an Azure Data Factory (V2) resource (you may need to refresh the resource group view).

Import a notebook

You can create notebooks in your Azure Databricks workspace to run code written in a range of programming languages. In this exercise, you'll import an existing notebook that contains some Python code.


1. In the Azure portal, in the **dp000-xxxxxxx** resource group, select the **databricksxxxxxxx** Azure Databricks Service resource.
2. In the **Overview** page for **databricksxxxxxxx**, use the **Launch Workspace** button to open your Azure Databricks workspace in a new browser tab; signing in if prompted.
3. If a **What's your current data project?** message is displayed, select **Finish** to close it. Then view the Azure Databricks workspace portal and note that the sidebar on the left side contains icons for the various tasks you can perform. The sidebar expands to show the names of the task categories.
4. Expand the sidebar and select the **Workspace** tab. Then select the **Users** folder and in the  menu for the  **your_user_name** folder, select **Import**.
5. In the **Import Notebooks** dialog box, select **URL** and import the notebook from

```
https://github.com/MicrosoftLearning/mslearn-
databricks/raw/main/Allfiles/Labs/05/Process-Data.dbc
```

6. Select  **Home** and then open the **Process Data** notebook you just imported.

Note: If a tip is displayed, use the **Got it** button to close it. This applies to any future tips that may be displayed as you navigate the workspace interface for the first time.

7. Review the contents of the notebook, which include some Python code cells to:
- Retrieve a parameter named **folder** if it is has been passed (otherwise use a default value of *data*).
 - Download data from GitHub and save it in the specified folder in the Databricks File System (DBFS).
 - Exit the notebook, returning the path where the data was saved as an output

 **Tip:** The notebook could contain practically any data processing logic you need. This simple example is designed to show the key principles.

Enable Azure Databricks integration with Azure Data Factory

To use Azure Databricks from an Azure Data Factory pipeline, you need to create a linked service in Azure Data Factory that enables access to your Azure Databricks workspace.

Generate an access token

1. In the Azure Databricks portal, at the bottom of the sidebar, select **Settings** and then select **User Settings**.
2. In the **User Settings** page, on the **Access tokens** tab, select **Generate new token** and generate a new token with the comment *Data Factory* and a blank lifetime (so the token doesn't expire). Be careful to *copy the token when it is displayed before selecting **Done***.
3. Paste the copied token to a text file so you have it handy for later in this exercise.

Create a linked service in Azure Data Factory

1. Return to the Azure portal, and in the **dp000-xxxxxxx** resource group, select the **adfxxxxxxx** Azure Data Factory resource.
2. On the **Overview** page, select the link to **Open Azure Data Factory Studio**. Sign in if prompted.
3. In Azure Data Factory Studio, use the » icon to expand the navigation pane on the left. Then select the **Manage** page.
4. On the **Manage** page, in the **Linked services** tab, select **+ New** to add a new linked service.
5. In the **New linked service** pane, select the **Compute** tab at the top. Then select **Azure Databricks**.
6. Continue, and create the linked service with the following settings:
 - **Name:** AzureDatabricks
 - **Description:** Azure Databricks workspace
 - **Connect via integration runtime:** AutoResolveIntegrationRuntime
 - **Account selection method:** From Azure subscription
 - **Azure subscription:** *Select your subscription*
 - **Databricks workspace:** *Select your **databricksxxxxxxx** workspace*
 - **Select cluster:** New job cluster
 - **Databrick Workspace URL:** *Automatically set to your Databricks workspace URL*
 - **Authentication type:** Access token
 - **Access token:** *Paste your access token*
 - **Cluster version:** 10.4 LTS (Scala 2.12, Spark 3.2.1)
 - **Python version:** 3
 - **Cluster node type:** Standard_DS3_v2
 - **Python version:** 3
 - **Worker options:** Fixed
 - **Workers:** 1

Use a pipeline to run the Azure Databricks notebook

Now that you have created a linked service, you can use it in a pipeline to run the notebook you viewed previously.

Create a pipeline

1. In Azure Data Factory Studio, in the navigation pane, select **Author**.
2. On the **Author** page, in the **Factory Resources** pane, use the + icon to add a **Pipeline**.
3. In the **Properties** pane for the new pipeline, change its name to **Process Data with Databricks**. Then use the **Properties** button (which looks similar to *) on the right end of the toolbar to hide the **Properties** pane.
4. In the **Activities** pane, expand **Databricks** and drag a **Notebook** activity to the pipeline designer surface.
5. With the new **Notebook1** activity selected, set the following properties in the bottom pane:
 - **General:**

- **Name:** Process Data
 - **Azure Databricks:**
 - **Databricks linked service:** Select the **AzureDatabricks** linked service you created previously
 - **Settings:**
 - **Notebook path:** Browse to the **Users/your_user_name** folder and select the **Process Data** notebook
 - **Base parameters:** Add a new parameter named **folder** with the value **product_data**
6. Use the **Validate** button above the pipeline designer surface to validate the pipeline. Then use the **Publish all** button to publish (save) it.

Run the pipeline

1. Above the pipeline designer surface, select **Add trigger**, and then select **Trigger now**.
2. In the **Pipeline run** pane, select **OK** to run the pipeline.
3. In the navigation pane on the left, select **Monitor** and observe the **Process Data with Databricks** pipeline on the **Pipeline runs** tab. It may take a while to run as it dynamically creates a Spark cluster and runs the notebook. You can use the **Refresh** button on the **Pipeline runs** page to refresh the status.

Note: If your pipeline fails, your subscription may have insufficient quota in the region where your Azure Databricks workspace is provisioned to create a job cluster. See [CPU core limit prevents cluster creation](#) for details. If this happens, you can try deleting your workspace and creating a new one in a different region. You can specify a region as a parameter for the setup script like this: `./setup.ps1 eastus`

4. When the run succeeds, select its name to view the run details. Then, on the **Process Data with Databricks** page, in the **Activity Runs** section, select the **Process Data** activity and use its **output** icon to view the output JSON from the activity, which should resemble this:

Code Copy

```
{
  "runPageUrl": "https://adb-....../run/...",
  "runOutput": "dbfs:/product_data/products.csv",
  "effectiveIntegrationRuntime": "AutoResolveIntegrationRuntime (East US)",
  "executionDuration": 61,
  "durationInQueue": {
    "integrationRuntimeQueue": 0
  },
  "billingReference": {
    "activityType": "ExternalActivity",
    "billableDuration": [
      {
        "meterType": "AzureIR",
        "duration": 0.03333333333333333,
        "unit": "Hours"
      }
    ]
  }
}
```

5. Note the **runOutput** value, which is the *path* variable to which the notebook saved the data.

Delete Azure Databricks resources

Now you've finished exploring Azure Data Factory integration with Azure Databricks, you must delete the resources you've created to avoid unnecessary Azure costs and free up capacity in your subscription.

1. Close the Azure Databricks workspace and Azure Data Factory studio browser tabs and return to the Azure portal.
2. On the Azure portal, on the **Home** page, select **Resource groups**.
3. Select the **dp000-xxxxxxx** resource group containing your Azure Databricks and Azure Data Factory workspace (not the managed resource group).
4. At the top of the **Overview** page for your resource group, select **Delete resource group**.
5. Enter the resource group name to confirm you want to delete it, and select **Delete**.

After a few minutes, your resource group and the managed workspace resource group associated with it will be deleted.