
Microsoft
Databricks Development:
Databricks CLI
Microsoft Services

Lesson 2: Working with Azure Databricks CLI

- After completing this lesson, you will be able to:
  - Learn an easy way to interact with Azure Databricks through the Command Line Interface
  - Learn common commands used to interact with Azure Databricks via the CLI

What is the Databricks CLI

- Like many Big Data or Cloud platforms, there exists a Command Line Interface or CLI to interact with your Azure Databricks workspace
- The Databricks CLI is organized into command groups based on the Workspace API, DBFS API, Jobs API, Clusters API, Libraries API, and Secrets API (*workspace*, *fs*, *jobs*, *runs*, *clusters*, *libraries*, and *secrets*)
- The CLI allows users to have a fast and easy way of interacting with their Databricks assets and workspace

https://docs.azuredatabricks.net/user-guide/dev-tools/databricks-cli.html#id1

The Databricks command-line interface (CLI) provides an easy-to-use interface to the Azure Databricks platform. The open source project is hosted on GitHub. The CLI is built on top of the Databricks REST API and is organized into command groups based on the Workspace API, Clusters API, DBFS API, Groups API, Jobs API, Libraries API, and Secrets API: workspace, clusters, fs, groups, jobs, runs, libraries, and secrets.

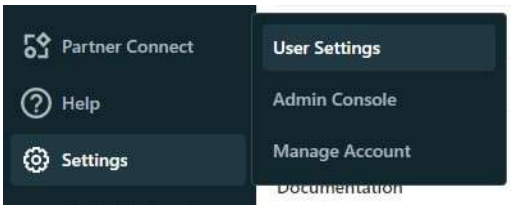
Using the Azure Databricks CLI

Prerequisites of running Azure Databricks CLI

- Install Python
  - Download Python | Python.org
- Install the Azure Databricks CLI
  - Execute "pip install databricks-cli" in a command prompt.

Get a Databricks Access Token

1. Go to your Databricks Workspace, click **Settings** in the bottom right, and choose **User Settings**

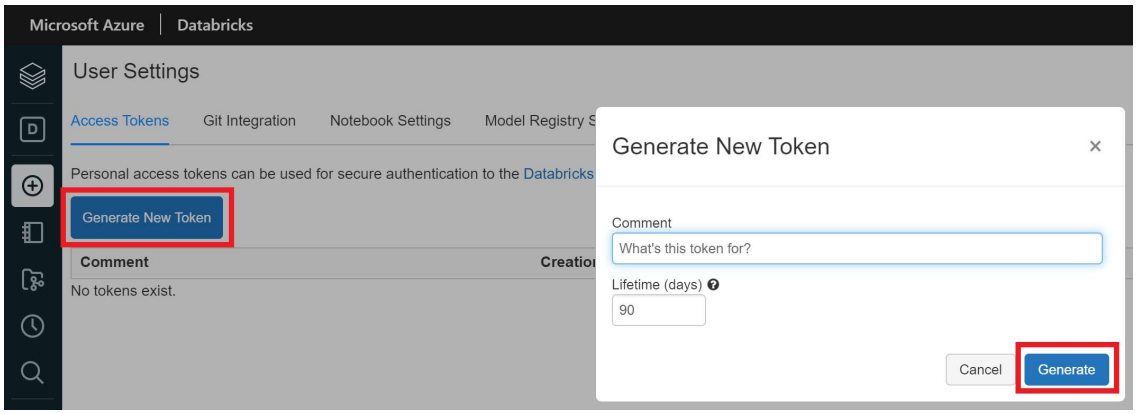


If using Windows OS, download python compiler: Download Python | Python.org

Python compiler will contain Pip within the scripts folder. Add this location of pip to the environment variables path. Open cmd or powershell and type "pip install databricks-cli" to start installation of Databricks CLI

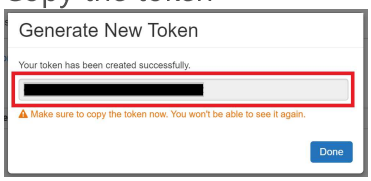
Using the Azure Databricks CLI

2. Click the "Generate New Token" button
3. Specify the details of the token. If the Lifetime is blank, the token will live indefinitely.



Using the Azure Databricks CLI

4. Copy the token



5. Go to the terminal and run the following

```
PS C:\> databricks configure --token
Databricks Host (should begin with https://): https://adb-3239123986190774.14.azuredatabricks.net/
Token:
PS C:\>
```

6. Run a sample command like

```
PS C:\> databricks fs ls
FileStore
databricks-results
tmp
PS C:\>
```

Store your token in a secure location. You won't be able to access it again after initial generation

Example Commands using Databricks CLI

- Workspace Commands
  - These are commands that help you interact with the workspace itself
  - An example command would be to list all the files in your workspace

    ```
    databricks workspace ls /Users/example@databricks.com
    ```
- DBFS Commands
  - These are commands that allow you to work and interact with the Databricks File System or DBFS and the various files that you have created or uploaded
  - An example command would be to copy a file to DBFS

    ```
    dbfs cp test.txt dbfs:/test.txt
    ```
- Jobs Commands
  - These are commands that allow you to run or monitor jobs within Azure Databricks
  - An example would be to list the jobs that have been submitted to Azure Databricks

    ```
    databricks jobs list
    ```
- Cluster Commands
  - These are commands that allow you to create, restart, and monitor your clusters in Azure Databricks
  - An example command would be to list the node types of your clusters

    ```
    databricks clusters list-node-types
    ```

Example Commands using Databricks CLI

- Library Commands
  - This family of commands allows you to attach libraries to clusters or see what has been already attached, among other functions
  - An example command would be to install a library via a JAR file to a cluster running in your Azure Databricks workspace

```
databricks libraries install --cluster-id $CLUSTER_ID --jar dbfs:/test-dir/test.jar
```

- Secrets Commands
  - These commands allow you to create, update, and maintain the secrets you have created for your Databricks workspace
  - A common example is to create a scope for a secret via the CLI

```
databricks secrets create-scope --scope my-scope
```

Demo:

Using the Databricks CLI

How to manage Databricks through command line (CLI)

Instructions
Follow the instructions below

Let's Start a Cluster

Note: Have a dummy cluster that is separate from your main demo cluster that you can use to start, attach a library to and delete. Use any jar you want and import it into your workspace.

1. List the clusters in your workspace

Command: **databricks clusters list**

Grab the cluster id that is listed in the first column of the table

2. Start the cluster with the correct cluster-id

Command: **databricks clusters start --cluster-id [cluster-id]**

3. Re-run the command from step one, and make sure the cluster does not say TERMINATED or PENDING but rather RUNNING

****

Let's Attach Some Libraries

1. We are going to first see what libraries are attached to our cluster

Command: **databricks libraries list --cluster-id [cluster-id]**

2. Find and write down the path of the JAR in your workspace
3. Using the path you wrote down or saved and the cluster-id from the first part of this demo, run the following command to attach it

Command: **databricks libraries install –cluster-id [cluster-id] –jar [dbfs-path-to-jar]**

****

Let's Delete the Cluster

1. To completely tear down a cluster you can use the clusters delete command, run that now to delete your cluster

Command: **databricks clusters delete --cluster-id [cluster-id]**

Knowledge Checks

1. Name two families of commands mentioned in this lesson.
2. Name the command that will list the assets in a Databricks workspace.

1. Name two families of commands mentioned in this lesson.
    1. Libraries, Jobs, Clusters, Sercrets, Workspace, DBFS
        1. Two of the above list
2. Name the command that will list the assets in a Databricks workspace.
    1. Databricks workspace list