

Introduction to Databricks in Azure (2023 Version)

Azure Databricks is an excellent tool for data science and machine learning. It makes cleaning, preparing, and processing data quick and straightforward. Plus, with its scalable computing resources, you can train and deploy models efficiently. But Databricks isn't just for data science; it's also great for data engineering tasks.

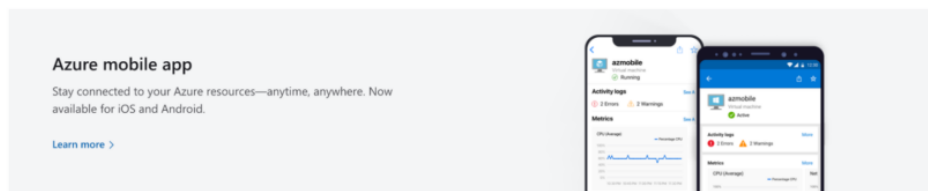
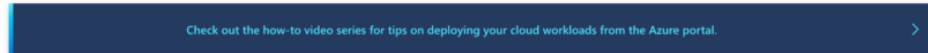
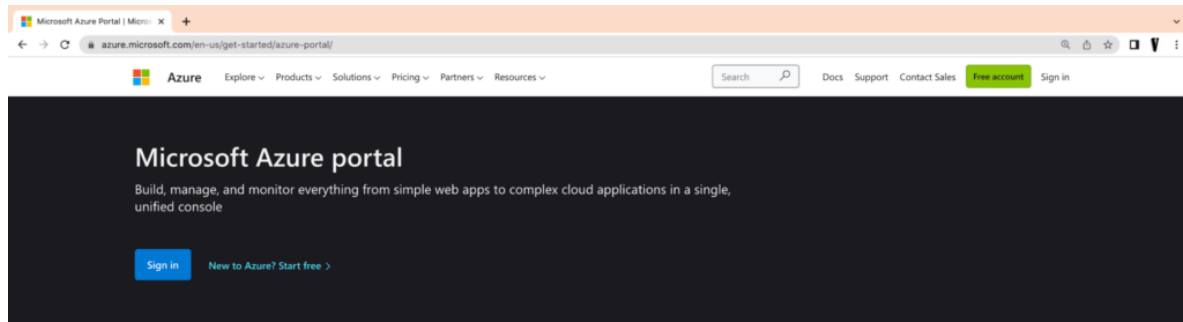
This cloud-based data processing platform has everything you need to build and run data pipelines. It's fully managed and comes with features like Azure Active Directory integration and role-based access control to keep your data secure.

Databricks features an interactive workspace that simplifies collaboration on data projects. It also includes tools to help you optimize your pipelines and boost performance. Overall, Azure Databricks is a fantastic option for building or running data pipelines in the cloud.

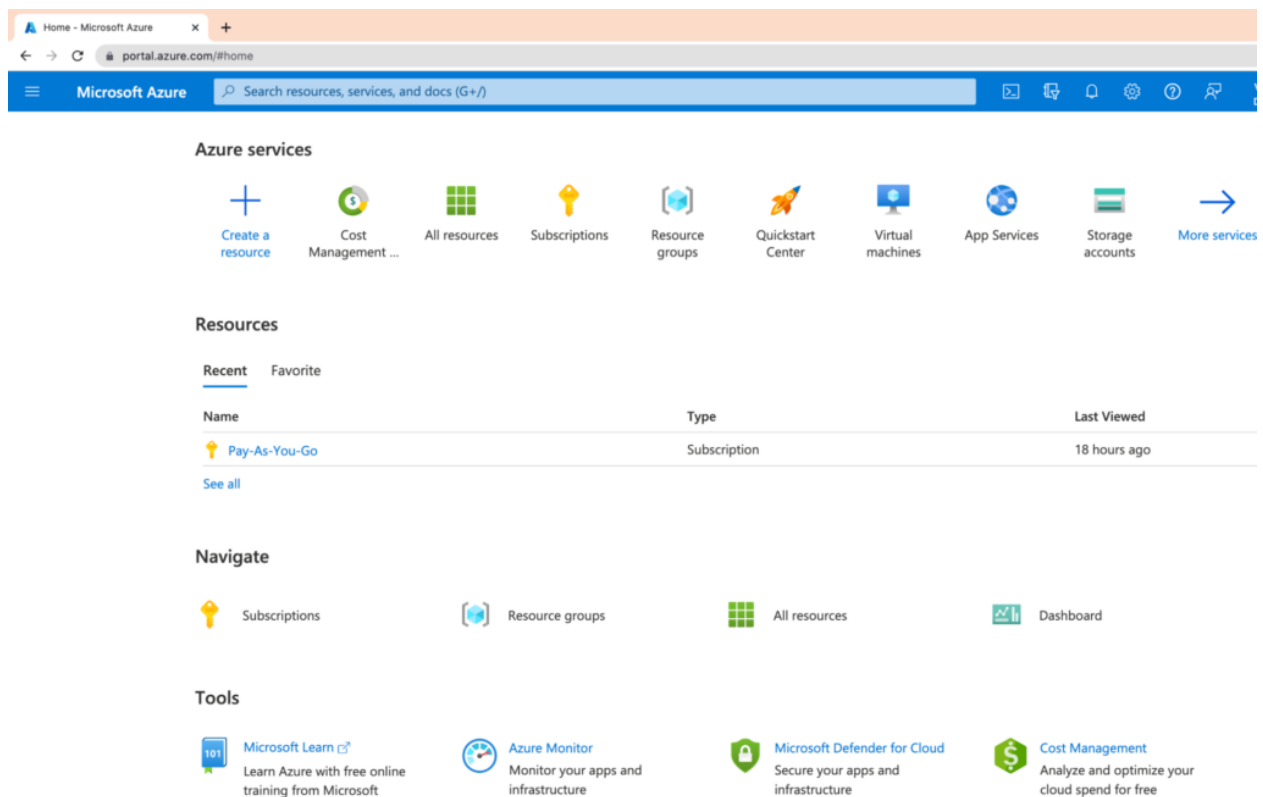
In this reading material, you'll learn how to build, train, and evaluate a machine learning model using Azure Databricks.

Log in to Azure Databricks

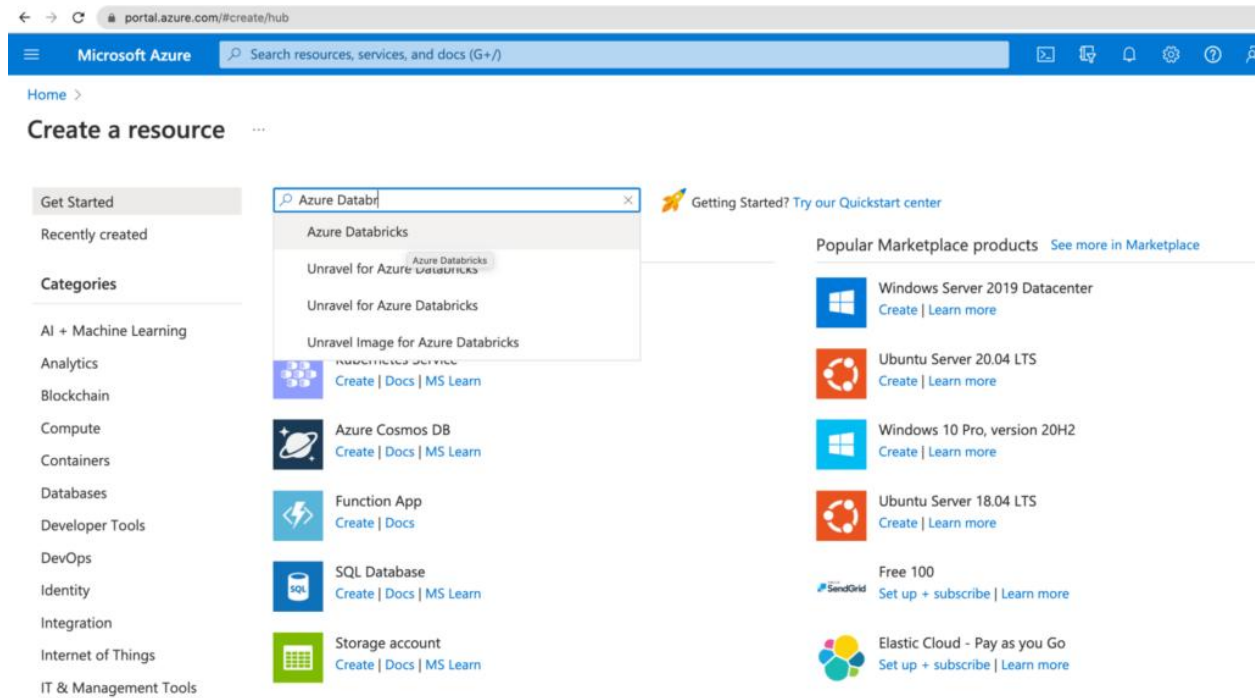
The first step is to create an account with Azure. You can do this by visiting the Azure portal <https://azure.microsoft.com/en-us/get-started/azure-portal/>



After creating an account and logging in, you'll land on this page:



You'll notice your subscription, which is Pay-As-You-Go in this example. You'll also see an option to "Create a resource." Click on it, type "Azure Databricks" in the search bar, and you'll see the following:



Click on the **Create** button for Databricks.


Microsoft Azure

Search resources, services, and docs (G+/I)

Home > Create a resource >

Azure Databricks

Microsoft

 **Azure Databricks** [Add to Favorites](#)

Microsoft

★ 4.3 (158 Marketplace ratings)

Plan

Azure Databricks

Create

Overview

Plans

Usage Information + Support

Reviews

Fast, easy, and collaborative Apache Spark-based analytics platform

Accelerate innovation by enabling data science with a high-performance analytics platform that's optimized for Azure.

Drive innovation and increase productivity

Bring teams together in an interactive workspace. From data gathering to model creation, use Databricks Notebooks to unify the process and instantly deploy to production. Launch your new Spark environment with a single click. Integrate effortlessly with a wide variety of data stores and services such as [Azure SQL Data Warehouse](#), [Azure Cosmos DB](#), [Azure Data Lake Store](#), [Azure Blob storage](#), and [Azure Event Hub](#). Add advanced artificial intelligence (AI) capabilities instantly and share your insights through rich integration with PowerBI.

Fill in the required fields like subscription, region, and resource group. Make sure to give unique names to the resource group and Databricks workspace. Then, click "Review + create."

Create an Azure Databricks workspace ...

Basics Networking Advanced Tags Review + create

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

Pay-As-You-Go

Resource group * ⓘ

(New) Dbricks

[Create new](#)

Instance Details

Workspace name *

DbricksWS

Region *

East US

Pricing Tier * ⓘ

Standard (Apache Spark, Secure with Azure AD)

Standard (Apache Spark, Secure with Azure AD)

Premium (+ Role-based access controls)

Trial (Premium - 14-Days Free DBUs)

Review + create

< Previous

Next : Networking >

Finally, click on "Create" to begin creating your Databricks resource! Keep in mind that this process may take a few minutes to complete.

[Home](#) > [Create a resource](#) > [Azure Databricks](#) >

Create an Azure Databricks workspace ...

Validation Succeeded

[Basics](#) [Networking](#) [Advanced](#) [Tags](#) [Review + create](#)

Summary

Basics

Workspace name	DbricksWS
Subscription	Pay-As-You-Go
Resource group	Dbricks
Region	East US
Pricing Tier	standard

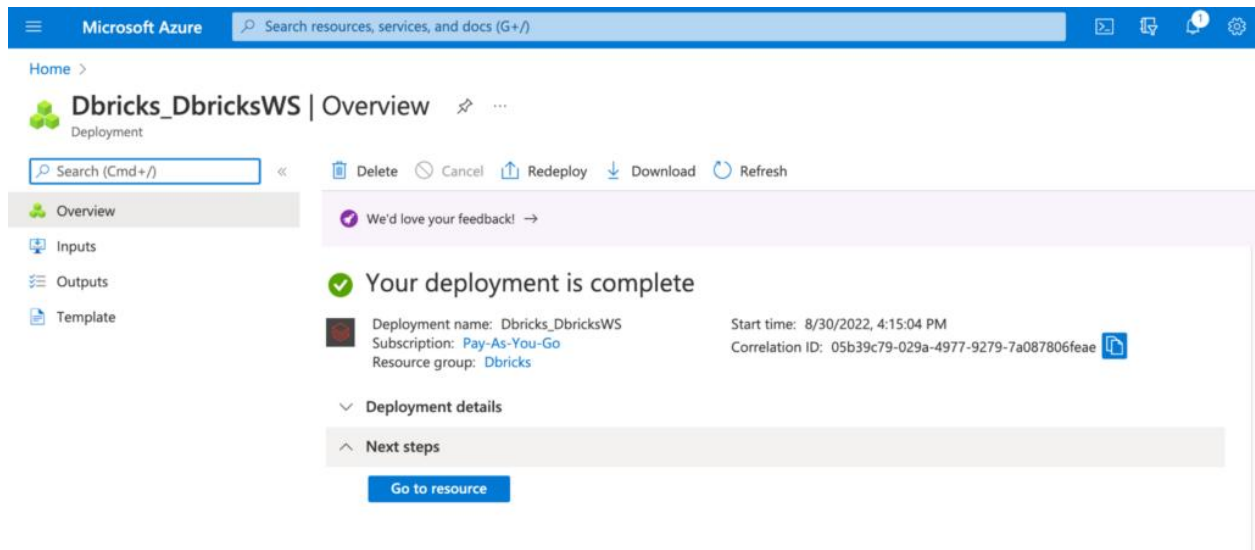
Networking

Deploy Azure Databricks workspace with Secure Cluster Connectivity (No Public IP)	No
Deploy Azure Databricks workspace in your own Virtual Network (VNet)	No

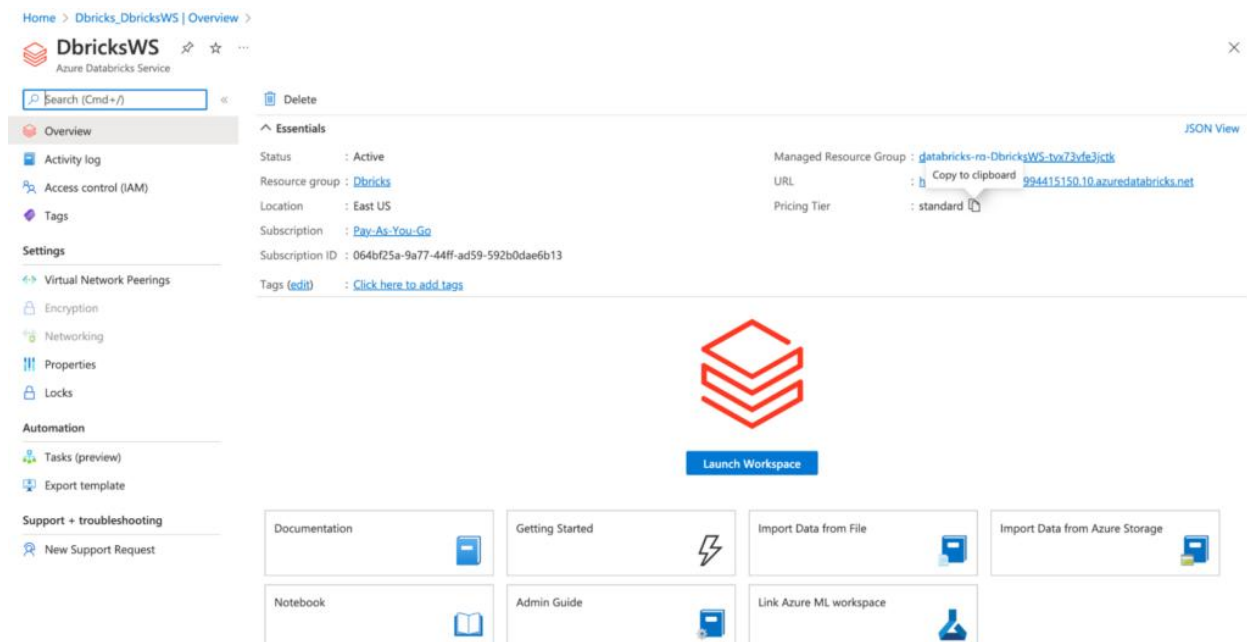
Advanced

Enable Infrastructure Encryption	No
----------------------------------	----

Once the resource is created, you can click on "Go to resource" to access it.



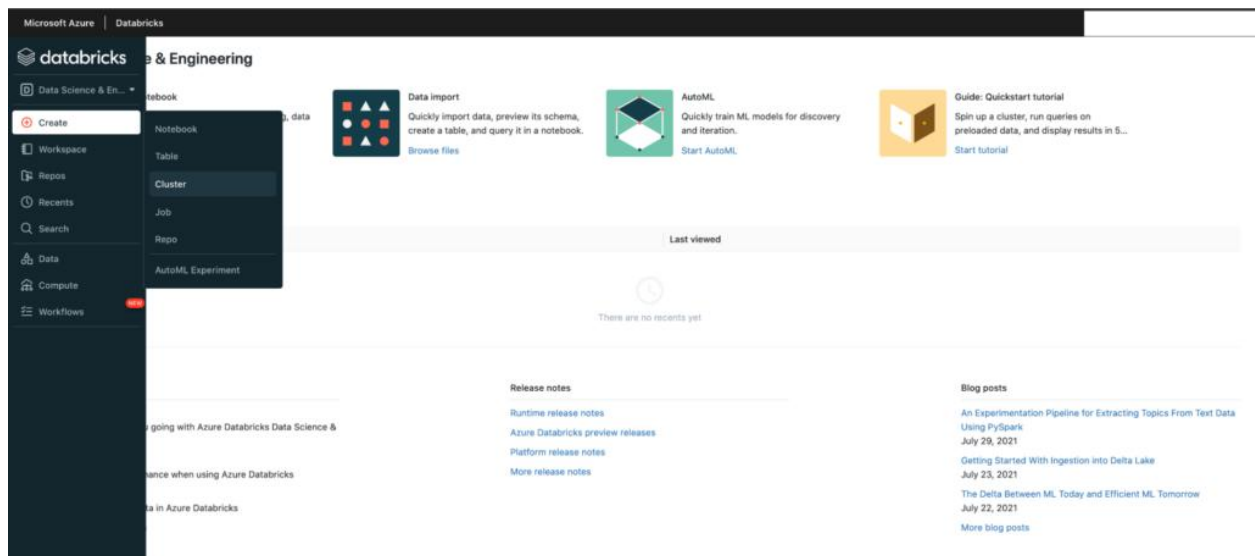
The Databricks page will appear like this:



Click on "Launch Workspace" to access the Azure Databricks environment.

Create Cluster

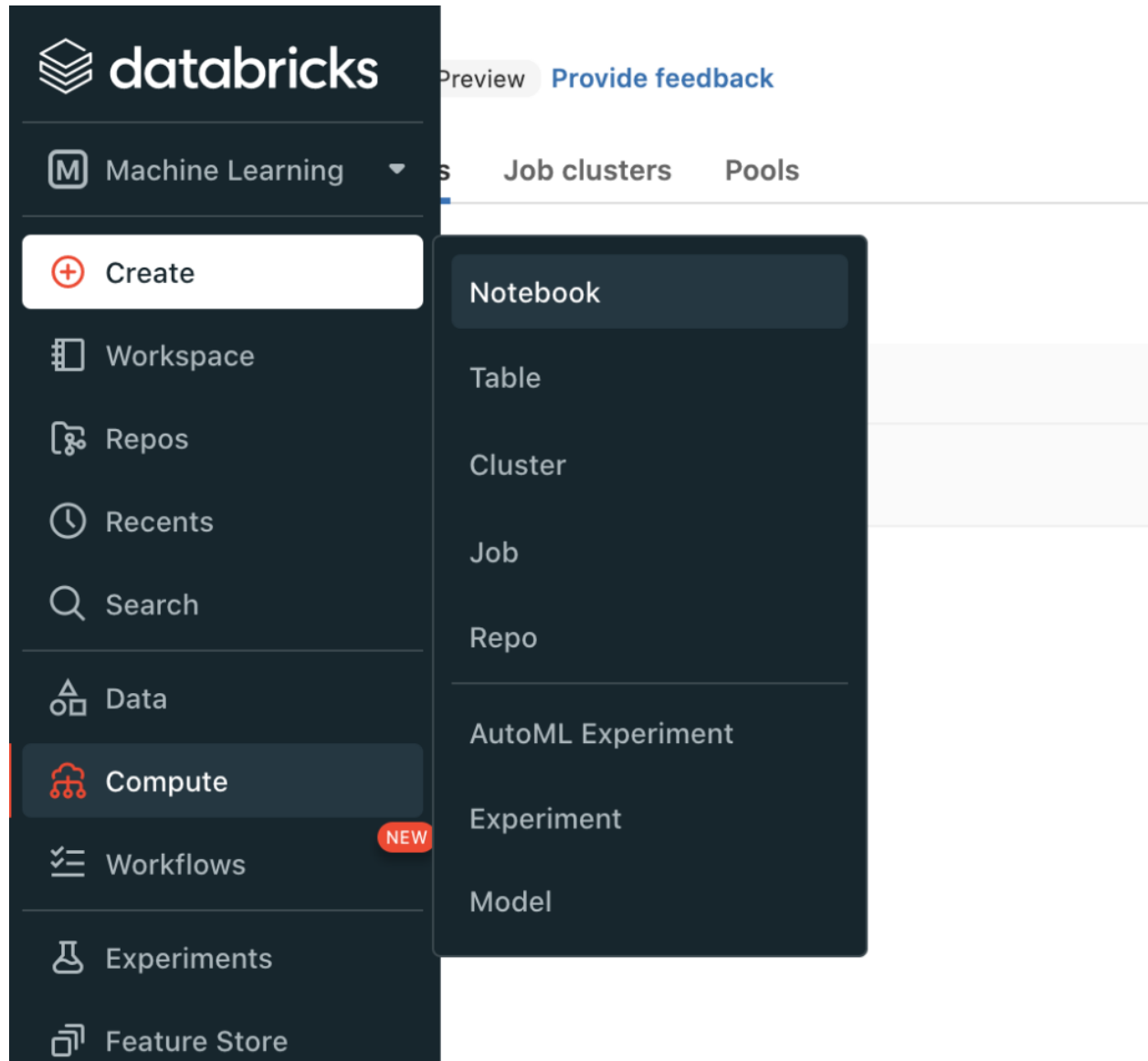
Next, you'll create a new Databricks cluster. To get started, click on "Create" and then select the "Cluster" button from the left panel of the Azure Databricks workspace.



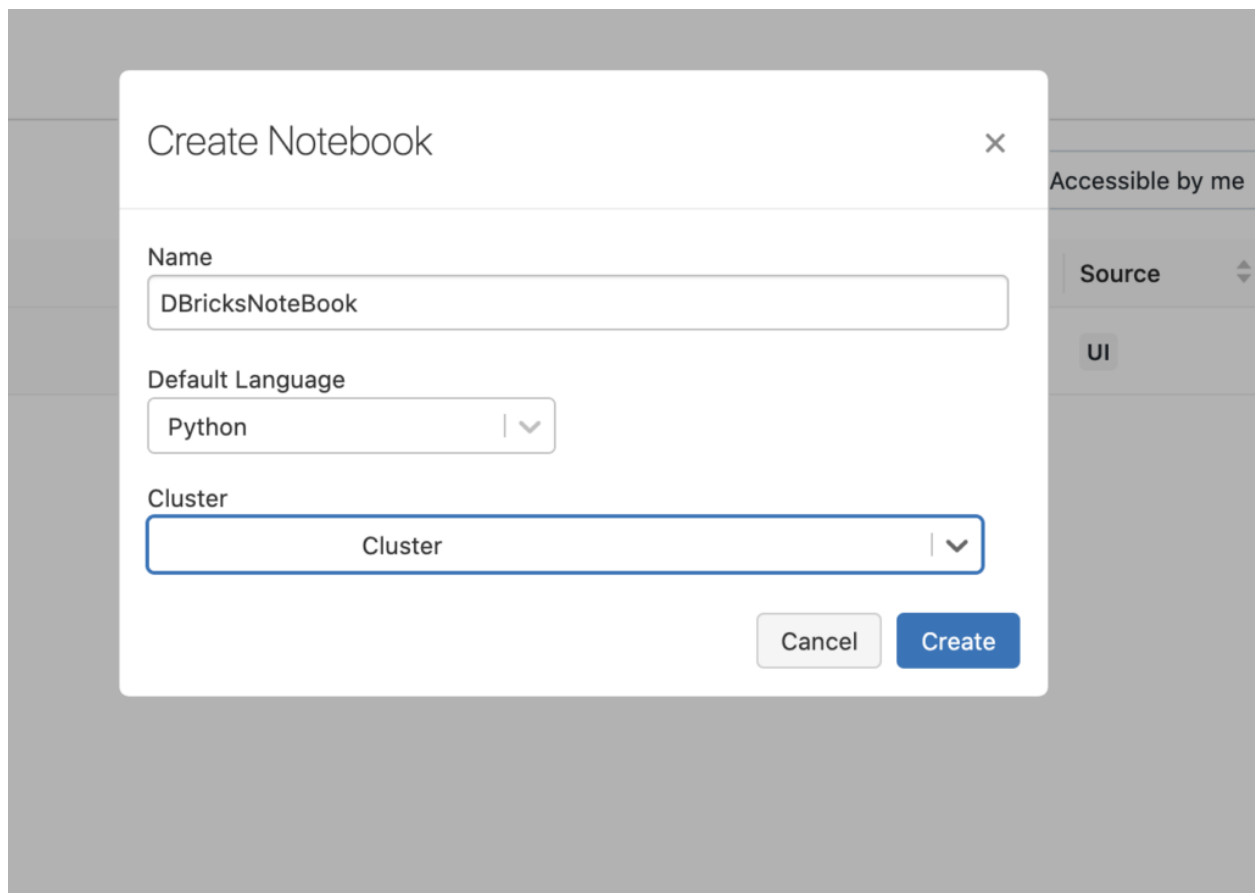
On the next page, you'll specify the configuration for your new cluster. For instance, you can set the number of users in Access mode and define the node type and runtime version. In this tutorial, we've specified a Single node, but feel free to accept the default setting as well.

Launch Notebook

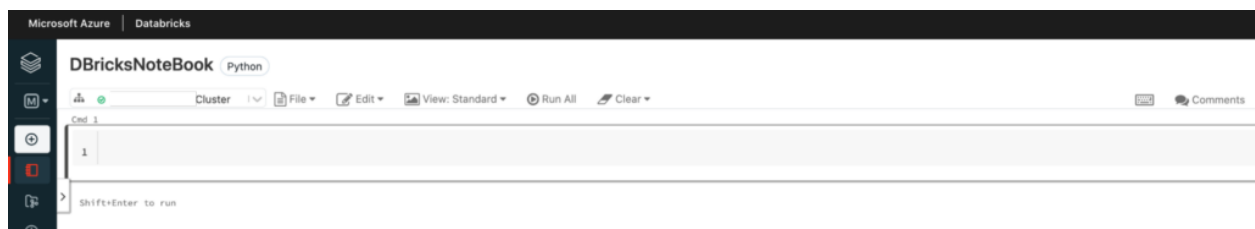
You'll run your machine learning code in a Python notebook within the Databricks workspace. To launch a notebook, click on "Create" and then select the "Notebook" icon in the left-hand sidebar of your Databricks workspace.



Now, you'll be prompted to give your new notebook a name and select its language. For our purposes, give your notebook a unique name and choose Python as the language.



Click on "Create," and you'll see the notebook you just created.



After creating your notebook, you'll be directed to its editor page where you can start writing and running your code! Now, let's dive into some machine learning!

Loading Data

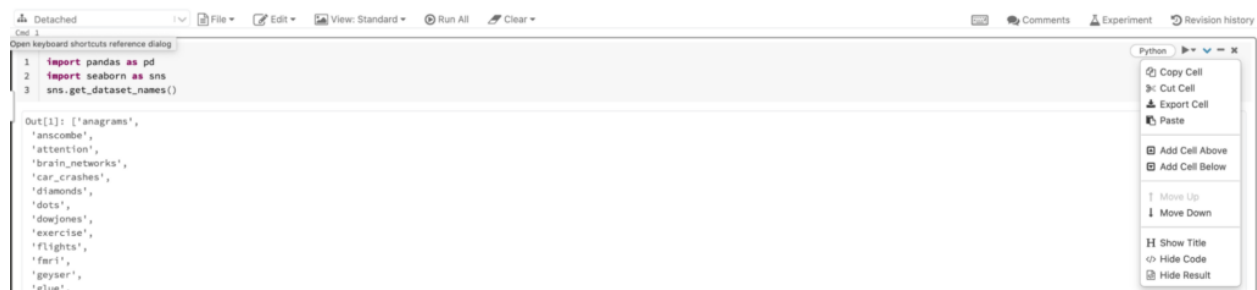
Before building the machine learning model, we need to load the data. We'll use one of the built-in datasets provided by the seaborn library. Seaborn includes several important datasets in its library. When you install seaborn, these datasets are automatically downloaded and ready to use! The following commands import the library and list the datasets available from seaborn.

```
import pandas as pd
import seaborn as sns
sns.get_dataset_names()
```

This will display the list of datasets available in seaborn, as shown below:

```
Out[1]: ['anagrams', 'anscombe', 'attention', 'brain_networks',
'car_crashes', 'diamonds', 'dots', 'dowjones', 'exercise', 'flights', 'fmri',
'geyser', 'glue', 'healthexp', 'iris', 'mpg', 'penguins', 'planets',
'seaice', 'taxi', 'tips',...]
```

To load the dataset, we'll use the `load_dataset()` function. In this tutorial, we'll use the Iris dataset, which contains information about different types of flowers. You can easily create a new code block by clicking on the downward arrow and selecting "Add Cell Below" or "Add Cell Above," depending on your needs.



In the new code block, load the dataset and inspect the first five rows with the following commands:

```
df = sns.load_dataset('iris')
df.head()
```

This will produce the following output:

```
| | sepal_length | sepal_width | petal_length | petal_width | species |
|--:|-----:|-----:|-----:|-----:|-----:|
| 0 | 5.1   | 3.5   | 1.4   | 0.2   | setosa |
| 1 | 4.9   | 3.0   | 1.4   | 0.2   | setosa |
| 2 | 4.7   | 3.2   | 1.3   | 0.2   | setosa |
| 3 | 4.6   | 3.1   | 1.5   | 0.2   | setosa |
| 4 | 5.0   | 3.6   | 1.4   | 0.2   | setosa |
```

```
1 df = sns.load_dataset('iris')
2 df.head()
```

Out[2]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Once we've downloaded the dataset, we need to split it into training and test sets. The training set is used to train our machine learning model, while the test set is used to evaluate its performance. We can

accomplish this using the `train_test_split()` function from `scikit-learn`. This is also a good time to import all the required libraries.

Here's the code to reproduce the above result:

```
# Import other required libraries
import sklearn
import numpy as np

# Import necessary modules
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report

# Create arrays for the features and the response variable
y = df['species'].values
X = df.drop('species', axis=1).values

# Create training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.4,
                                                    random_state=10)
X_train.shape, X_test.shape

Out[3]: ((90, 4), (60, 4))
```

Build and Evaluate Machine Learning Model

Now that we have our training and test sets, we can begin building our machine learning model. We'll utilize the `LogisticRegression` class from `scikit-learn` for this purpose.

To train our model, we will use the `fit()` function on our `LogisticRegression` object and pass in our training set as a parameter. After training our model, we can make predictions using our test set. We'll achieve this by calling the `predict()` function on our `LogisticRegression` object and passing in our test set as a parameter. This function will return a list of predictions for each sample in our test set.

Assessing the performance of your machine learning model is crucial to ensure it's functioning as intended. There are various metrics available for evaluating model performance. In this tutorial, we'll focus on using accuracy as our metric.

To determine the accuracy of our model, we'll need to compare our predictions with the actual labels in our test set. We can accomplish this by using the `confusion_matrix` function to calculate the accuracy of our predictions.

```
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
y_pred = logreg.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Running the above command will generate the following output:

```
precision recall f1-score support
setosa 1.00 1.00 1.00 18
versicolor 1.00 0.96 0.98 24
virginica 0.95 1.00 0.97 18

accuracy 0.98 60
macro avg 0.98 0.99 0.98 60
weighted avg 0.98 0.98 0.98 60
```

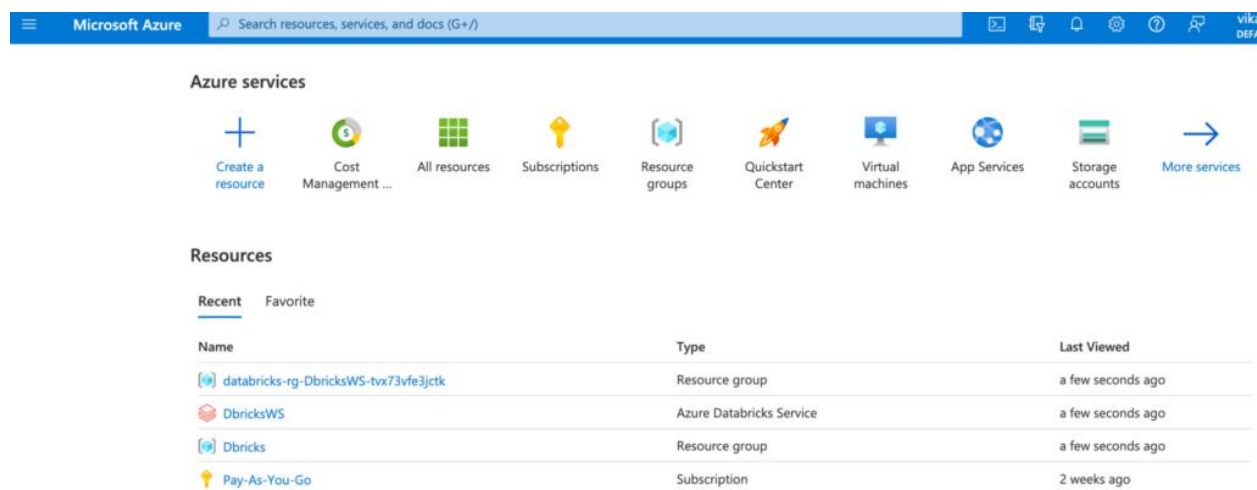
	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	18
versicolor	1.00	0.96	0.98	24
virginica	0.95	1.00	0.97	18
accuracy			0.98	60
macro avg	0.98	0.99	0.98	60
weighted avg	0.98	0.98	0.98	60

The output above indicates that the model accuracy is 98%, which is excellent.

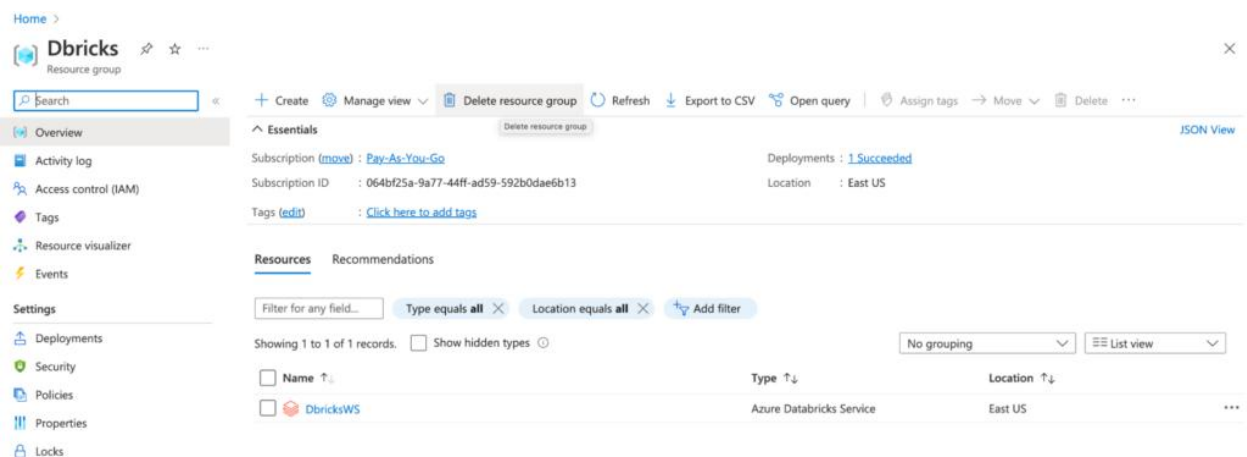
Conclusion

In this tutorial, you explored Azure Databricks, a popular unified analytics platform for data science and machine learning. You learned how to launch the Databricks workspace and build and evaluate a machine learning model using the Databricks notebook. You're now equipped to embark on your machine learning journey with Azure Databricks!

Finally, if you don't plan to use the resources you've created in the future, you should delete them. It's a straightforward process. First, go to the Azure portal where you can view a list of your resource groups.



Find the resource group you want to delete, and click on it.



Click on the "Delete resource group" button at the top of the page.

Lastly, confirm the deletion by entering the name of the resource group and clicking the "Delete" button.