Course syllabus
Perform data science with Azure Databricks - Course Syllabus

In this course, you will learn how to harness the power of Apache Spark and powerful clusters running on the Azure Databricks platform to run data science workloads in the cloud.

Module 1 - Introduction to Azure Databricks

In this module, you will discover the capabilities of Azure Databricks and the Apache Spark notebook for processing huge files. You will come to understand the Azure Databricks platform and identify the types of tasks well-suited for Apache Spark. You will also be introduced to the architecture of an Azure Databricks Spark Cluster and Spark Jobs.

Module 2 - Working with data in Azure Databricks

Azure Databricks supports day-to-day data-handling functions, such as reads, writes, and queries. In this module, you will work with large amounts of data from multiple sources in different raw formats. You will also learn to use the DataFrame Column Class Azure Databricks to apply column-level transformations, such as sorts, filters, and aggregations. You will also use advanced DataFrame functions operations to manipulate data, apply aggregates, and perform date and time operations in Azure Databricks.

Module 3 - Processing data in Azure Databricks

Azure Databricks supports a range of built-in SQL functions, however, sometimes you have to write a custom function, known as User-Defined Function (UDF). In this module, you will learn how to register and invoke UDFs. You will also learn how to use Delta Lake to create, append, and upsert data to Apache Spark tables, taking advantage of built-in reliability and optimizations.

Module 4 - Get started with Databricks and machine learning

In this module, you will learn how to use PySpark’s machine learning package to build key components of the machine learning workflows that include exploratory data analysis, model training, and model evaluation. You will also learn how to build pipelines for common data featurization tasks.

Module 5 - Manage machine learning lifecycles and fine-tune models

In this module, you will learn how to use MLflow to track machine learning experiments and how to use modules from the Spark’s machine learning library for hyperparameter tuning and model selection.

Module 6 - Train a distributed neural network and serve models with Azure Machine Learning

In this module, you will learn how to use the Uber’s Horovod framework along with the Petastorm library to run distributed, deep learning training jobs on Spark using training datasets in the Apache Parquet format. You will also learn how to use MLflow and Azure Machine Learning service register, package, and deploy a trained model to both Azure Container Instance, and Azure Kubernetes Service as a scoring web service.

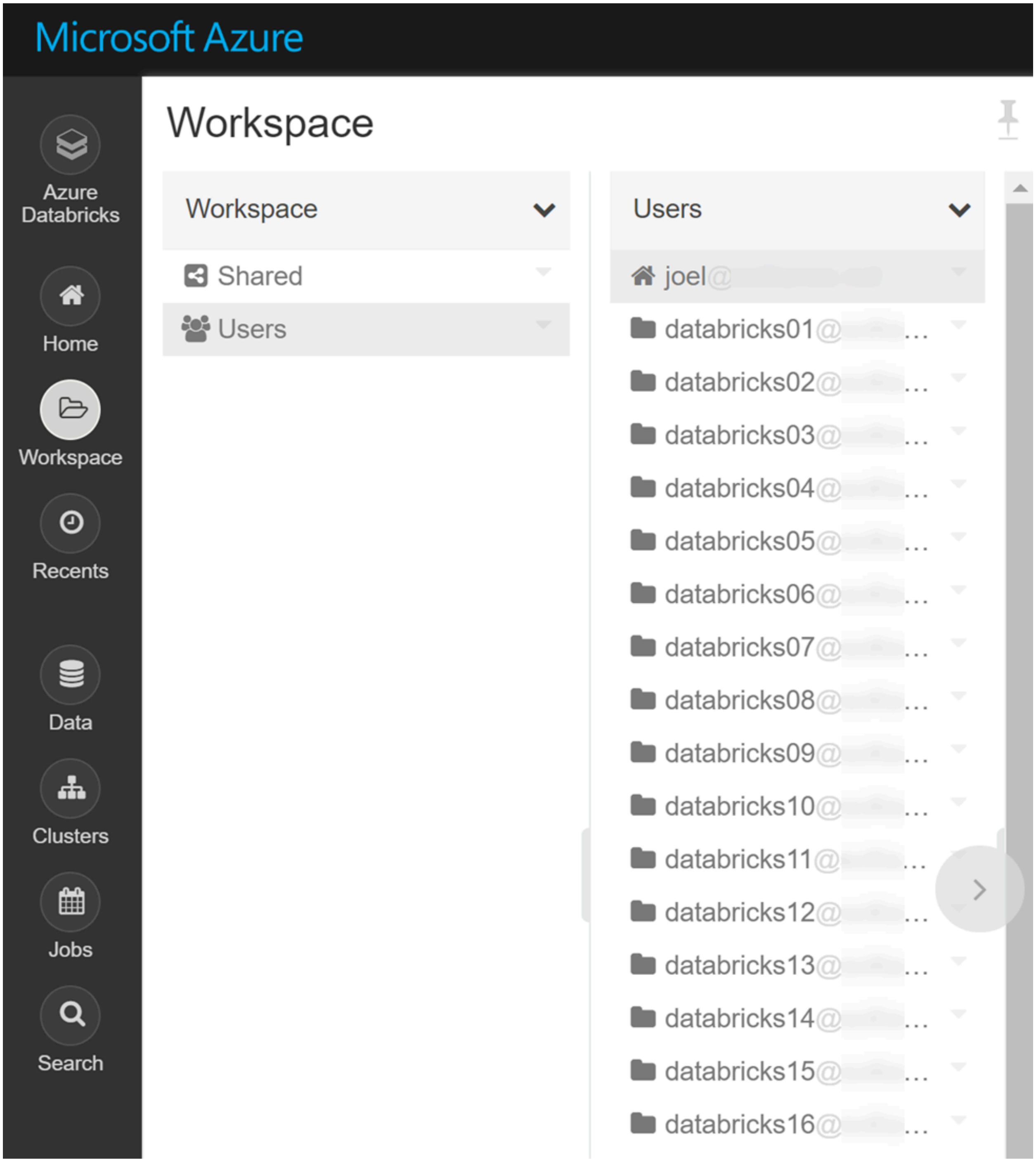
Create an Azure Databricks workspace and cluster

Note: *In this reading you can see the steps involved in the process of creating an Azure Databricks workspace and cluster.*

When talking about the Azure Databricks workspace, we refer to two different things.

The first reference is the logical Azure Databricks environment in which clusters are created, data is stored (via DBFS), and in which the server resources are housed.

The second reference is the more common one used within the context of Azure Databricks. That is the special root folder for all of your organization's Databricks assets, including notebooks, libraries, and dashboards, as shown below:



The Databricks workspace folder is shown.
The first step to using Azure Databricks is to create and deploy a Databricks workspace, which is the logical environment. You can do this in the Azure portal.

Deploy an Azure Databricks workspace

1. Open the Azure portal
2. Click **Create a Resource** in the top left
3. Search for "Databricks"

4. Select *Azure Databricks*
5. On the Azure Databricks page select *Create*
6. Provide the required values to create your Azure Databricks workspace:
   - **Subscription**: Choose the Azure subscription in which to deploy the workspace.
   - **Resource Group**: Use **Create new** and provide a name for the new resource group.
   - **Location**: Select a location near you for deployment. For the list of regions that are supported by Azure Databricks, see Azure services available by region.
   - **Workspace Name**: Provide a unique name for your workspace.
   - **Pricing Tier**: **Trial (Premium - 14 days Free DBUs)**. You must select this option when creating your workspace or you will be charged. The workspace will suspend automatically after 14 days. When the trial is over you can convert the workspace to **Premium** but then you will be charged for your usage.

7. Select **Review + Create**.
8. Select **Create**.

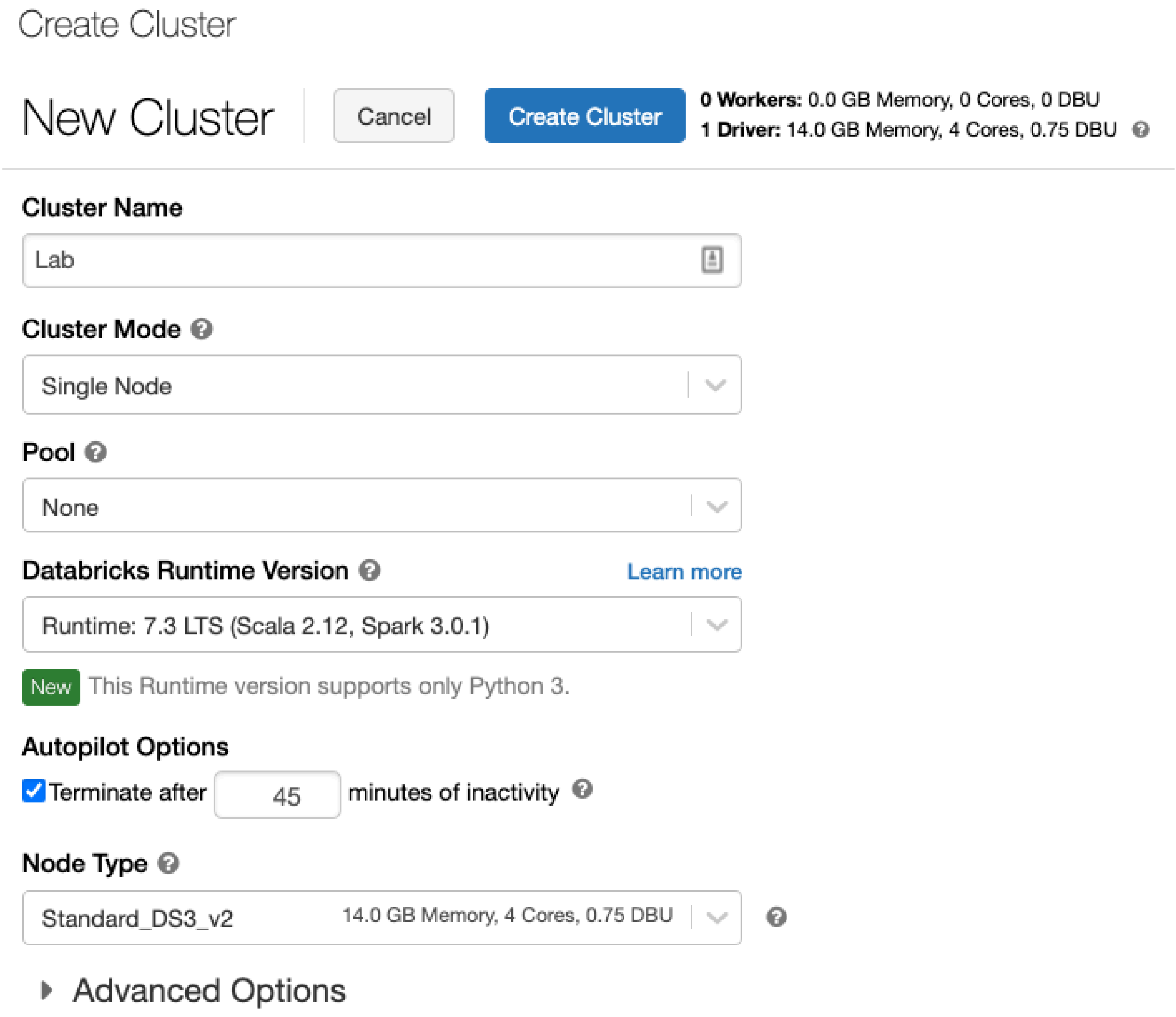
The workspace creation takes a few minutes. During workspace creation, the **Submitting deployment for Azure Databricks** tile appears on the right side of the portal. You might need to scroll right on your dashboard to view the tile. There's also a progress bar displayed near the top of the screen. You can watch either area for progress.

What is a cluster?

The notebooks are backed by clusters, or networked computers, that work together to process your data. The first step is to create a cluster.

Create a cluster

1. When your Azure Databricks workspace creation is complete, select the link to go to the resource.
2. Select **Launch Workspace** to open your Databricks workspace in a new tab.
3. In the left-hand menu of your Databricks workspace, select **Clusters**.
4. Select **Create Cluster** to add a new cluster.



0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU

Cluster Name
Lab

Cluster Mode
Single Node

Pool
None

Databricks Runtime Version Learn more
Runtime: 7.3 LTS (Scala 2.12, Spark 3.0.1)
New This Runtime version supports only Python 3.

Autopilot Options
☑ Terminate after 45 minutes of inactivity

Node Type
Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU

The create cluster page.

5. Enter a name for your cluster. Use your name or initials to easily differentiate your cluster from those of your co-workers.
6. Select the **Cluster Mode**: **Single Node**.
7. Select the **Databricks RuntimeVersion**: **Runtime: 7.3 LTS (Scala 2.12, Spark 3.0.1)**.
8. Under **Autopilot Options**, leave the box **checked** and in the text box enter `45`.
9. Select the **Node Type**: **Standard_DS3_v2**.
10. Select **Create Cluster**.

Create and execute a notebook

Note: *In this reading you can see the steps involved in the process of creating and executing a notebook.*

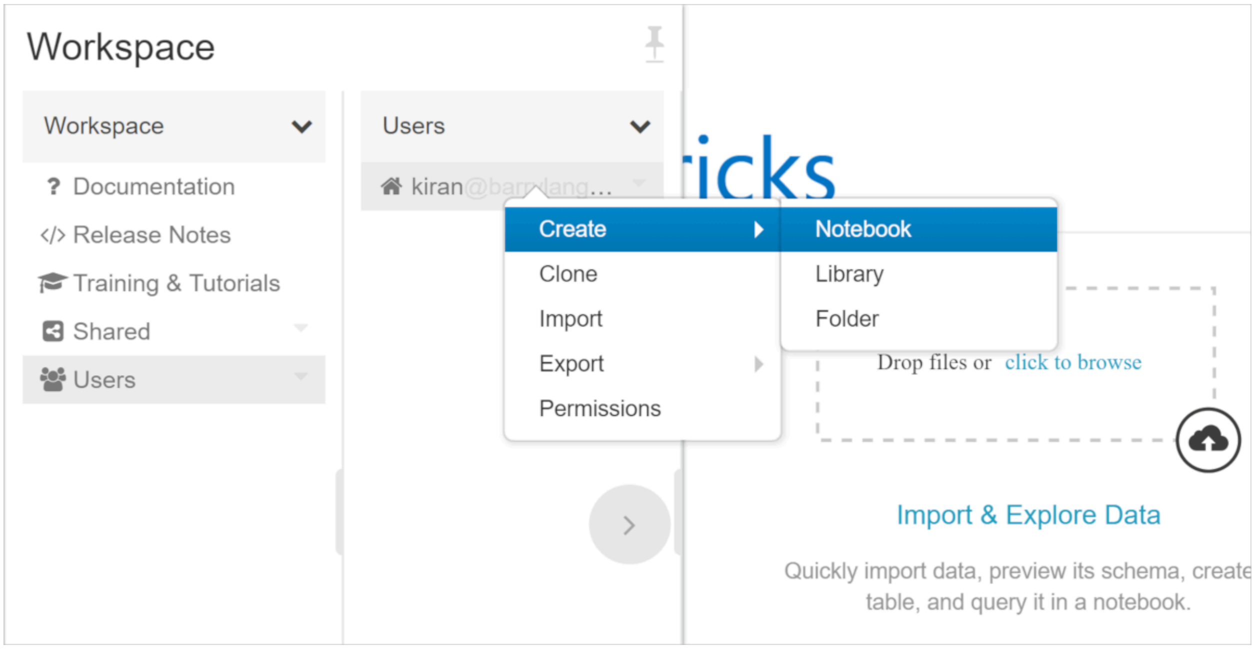
After creating your Databricks workspace, it's time to create your first notebook. To execute your notebook, you will attach the cluster you created in the previous exercise.

What is Apache Spark notebook?

A notebook is a collection of cells. These cells are run to execute code, to render formatted text, or to display graphical visualizations.

Create a notebook

1. In the Azure portal, click **All resources** menu on the left side navigation and select the Databricks workspace you created in the last unit.
2. Select **Launch Workspace** to open your Databricks workspace in a new tab.
3. On the left-hand menu of your Databricks workspace, select **Home**.
4. Right-click on your home folder.
5. Select **Create**.
6. Select **Notebook**.



The menu option to create a new notebook.

7. Name your notebook **First Notebook**.
8. Set the **Language** to **Python**.
9. Select the cluster to which to attach this notebook.

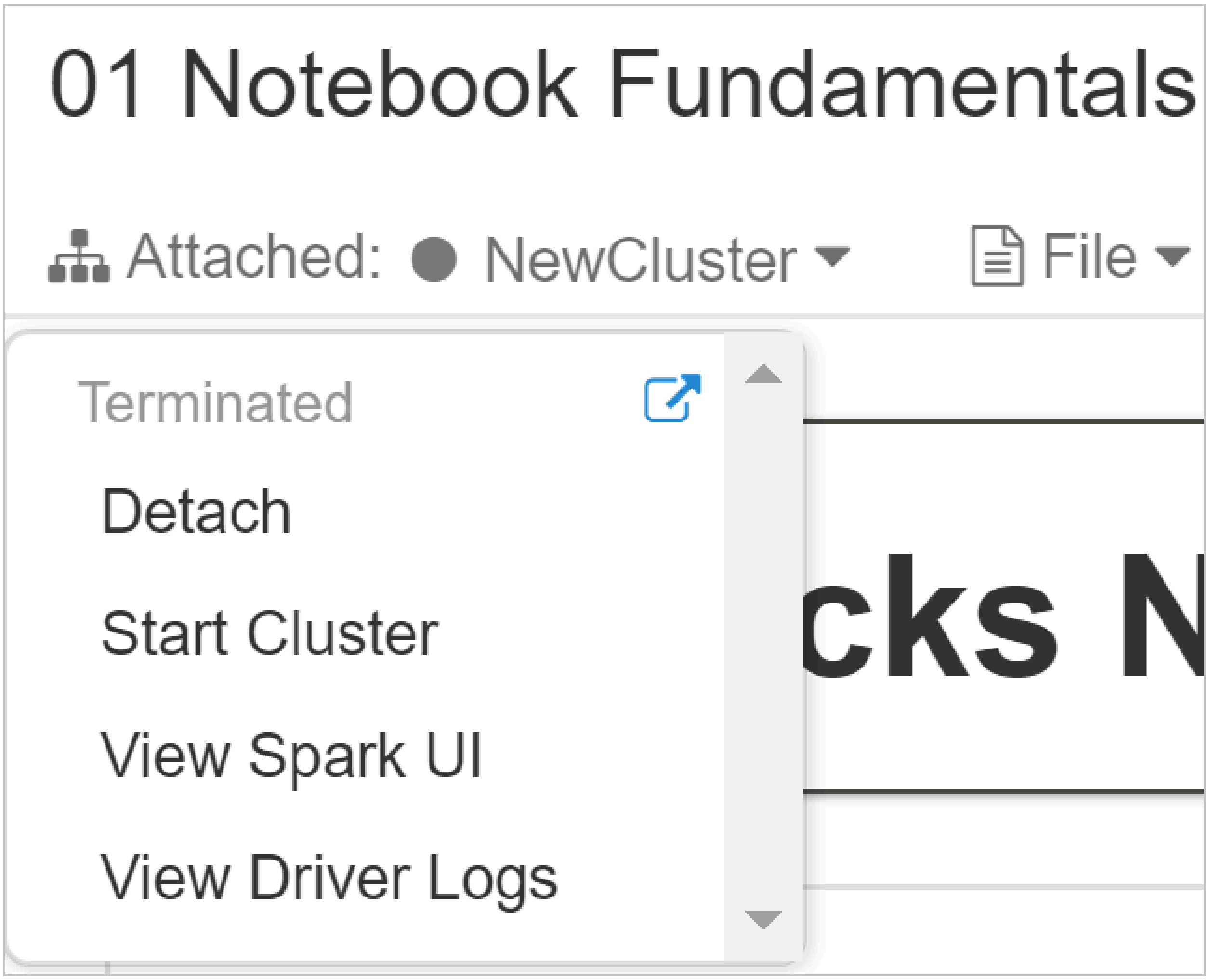
Note: This option displays only when a cluster is currently running. You can still create your notebook and attach it to a cluster later.

10. Select **Create**.

Now that you've created your notebook, let's use it to run some code.

Attach and detach your notebook

To use your notebook to run a code, you must attach it to a cluster. You can also detach your notebook from a cluster and attach it to another depending upon your organization's requirements.



The options that are available when a notebook is attached to a cluster.

If your notebook is attached to a cluster, you can:

- Detach your notebook from the cluster
- Restart the cluster
- Attach to another cluster
- Open the Spark UI
- View the log files of the driver

Exercise: Work with Notebooks

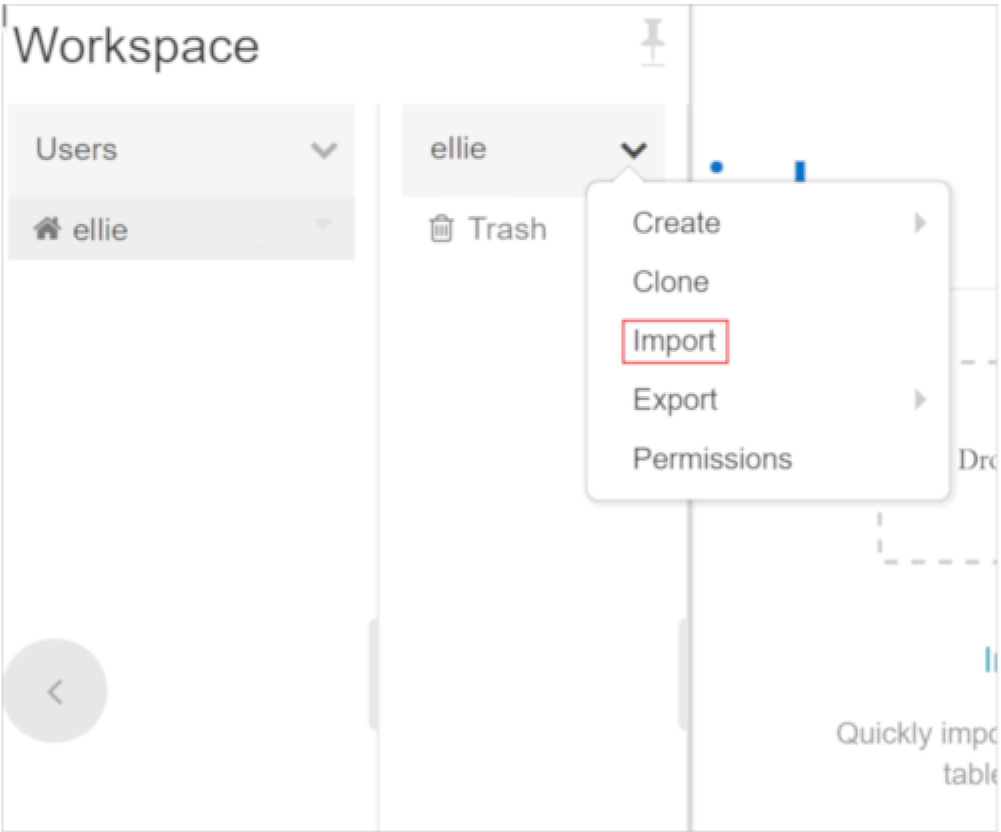
Note: To execute your notebook, you will attach the cluster you created in the previous exercise.
You can use Apache Spark notebooks to:

- Read and process huge files and data sets
- Query, explore, and visualize data sets
- Join disparate data sets found in data lakes
- Train and evaluate machine learning models
- Process live streams of data
- Perform analysis on large graph data sets and social networks

To learn more about using notebooks, clone the labs archive where sample notebooks are provided. These notebooks will help you understand how to use notebooks for your day-to-day tasks.

Clone the Databricks archive

1. In the Azure portal, navigate to your deployed Azure Databricks workspace and select **Launch Workspace**.
2. In the left pane, select **Workspace** > **Users**, and select your username (the entry with the house icon).
3. In the pane that appears, select the arrow next to your name, and select **Import**.



The menu option to import the archive.
4. In the **Import Notebooks** dialog box, select the URL and paste in the following URL:

https://github.com/solliancenet/microsoft-learning-paths-databricks-notebooks/blob/master/data-engineering/DBC/01-Introduction-to-Azure-Databricks.dbc?raw=true

5. Select **Import**.
6. Select the **01-Introduction-to-Azure-Databricks** folder that appears.
7. Use the set of notebooks in this folder to complete this lab.

Complete the following notebook

- **01-The-Databricks-Environment** - This notebook illustrates the fundamentals of a Databricks notebook.

1. Apache Spark is a unified processing engine that can analyze big data with which of the following features?

1 / 1 point

Select all that apply.

☑ SQL

Correct
Feedback: Spark is a unified processing engine that can analyze big data using SQL.

□ Support for multiple Drivers running in parallel on a cluster

☑ Real-time stream analysis

Correct
Spark is a unified processing engine that can analyze big data using real-time stream analysis.

☑ Machine Learning

Correct
Spark is a unified processing engine that can analyze big data using machine learning.

☑ Graph Processing

Correct
Spark is a unified processing engine that can analyze big data using graph processing.

2. Which of the following Databricks features are **not** Open-Source Spark?

0.75 / 1 point

Select all that apply.

□ MLFlow

□ Databricks Runtime

☑ Databricks Workflows

Correct
Databricks Workflows is not open-source Spark.

☑ Databricks Workspace

Correct
Databricks Workspace is not open-source Spark.

You didn't select all the correct answers

3. Apache Spark notebooks allow which of the following?

1 / 1 point

Select all that apply.

☑ Rendering of formatted text

✓ **Correct**
A notebook is a collection of cells. These cells can be run to render formatted text.

☑ Execution of code

✓ **Correct**
A notebook is a collection of cells. These cells are run to execute code.

□ Create new Workspace

☑ Display graphical visualizations

✓ **Correct**
A notebook is a collection of cells. These cells can display graphical visualizations.

4. In Azure Databricks when creating a new Notebook, the default languages available to select from are?

0.8 / 1 point

Select all that apply.

☑ Java

✗ **This should not be selected**
In Azure Databricks when creating a new Notebook, one of the default languages available to select from is not Java.

☑ Scala

✓ **Correct**
In Azure Databricks when creating a new Notebook, one of the default languages available to select from is Scala.

☑ Python

✓ **Correct**
In Azure Databricks when creating a new Notebook, one of the default languages available to select from is Python.

☑ SQL

✓ **Correct**
In Azure Databricks when creating a new Notebook, one of the default languages available to select from is SQL.

5. If your notebook is attached to a cluster, you can carry out which of the following from within the notebook?

1 / 1 point

Select all that apply.

☑ Attach to another cluster

Correct
If your notebook is attached to a cluster, you can attach to another cluster.

☑ Restart the cluster

Correct
If your notebook is attached to a cluster, you can restart the cluster.

☑ Detach your notebook from the cluster

Correct
If your notebook is attached to a cluster, you can detach your notebook from the cluster.

□ Delete the cluster

1. Azure Databricks Runtime adds several key capabilities to Apache Spark workloads that can increase performance and reduce costs. Which of the following are features of Azure Databricks?

0.8 / 1 point

Select all that apply.

☐ Parallel Cluster Drivers

☑ Caching

> **Correct**
> Azure Databricks Runtime adds several key capabilities to Apache Spark workloads that can increase performance and reduce costs including Caching.

☑ High-speed connectors to Azure storage services

> **Correct**
> Azure Databricks Runtime adds several key capabilities to Apache Spark workloads that can increase performance and reduce costs including High-speed connectors to Azure storage services.

☐ Indexing

☑ Auto-scaling and auto-termination

> **Correct**
> Try going back and reviewing the Describe Azure Databricks lesson.

> You didn't select all the correct answers

2. Apache Spark supports which of the following languages?

1 / 1 point

Select all that apply.

☑ Scala

> **Correct**
> Apache Spark supports Scala.

☑ Java

> **Correct**
> Apache Spark supports Java.

☑ Python

> **Correct**
> Apache Spark supports Python.

☐ ORC

3. Which of the following statements are True

1 / 1 point

Select all that apply.

☑ You can detach a notebook from a cluster and attach it to another cluster.

> ✓ **Correct**
> detach your notebook from a cluster and attach it to another depending upon your organization's requirements.

□ Once created a notebook can only be connected to the original cluster.

☑ To use your Azure Databricks notebook to run code, you must attach it to a cluster

> ✓ **Correct**
> To use your notebook to run a code, you must attach it to a cluster.

□ To use your Azure Databricks notebook to run code you do not require a cluster

4. Which of the following Databricks features are not Open-Source Spark?

1 / 1 point

☑ Databricks Runtime

> ✓ **Correct**
> Databricks Runtime is not open-source Spark

☑ Databricks Workspace

> ✓ **Correct**
> Databricks Workspace is not open-source Spark

□ MLFlow

☑ Databricks Workflows

> ✓ **Correct**
> Databricks Workflows is not open-source Spark

5. How many drivers does a Cluster have?

1 / 1 point

○ Configurable between one and eight

○ Two, running in parallel

● Only one

> ✓ **Correct**
> Feedback: A Cluster has one and only one driver.

6. What type of process are the driver and the executors?

1 / 1 point

○ Python processes

○ C++ processes

● Java processes

✓ **Correct**
The driver and the executors are Java processes.

7. You work with Big Data as a data engineer, and you must process real-time data. This is referred to as having which of the following characteristics?

1 / 1 point

○ High volume

○ Variety

● High velocity

✓ **Correct**
This characteristic relates to the requirement for streaming and real-time processing capabilities.

8. Spark's performance is based on parallelism. Which of the following Scalability methods is limited to a finite amount of RAM, Threads and CPU speeds?

1 / 1 point

● Vertical Scaling

○ Horizontal Scaling

○ Diagonal Scaling

✓ **Correct**
Scaling vertically is limited to a finite amount of RAM, Threads and CPU speeds.

9. Spark Cluster use two levels of parallelization. Which of the following are levels of parallelization?

0.5 / 1 point

☐ Slot

☐ Job

☑ Executor

✓ **Correct**
The first level of parallelization is the Executor - a Java virtual machine running on a node, typically, one instance per node.

☑ Partition

⊗ **This should not be selected**
Try going back and reviewing the Spark architecture fundamentals lesson.

10. In an Apache Spark Cluster jobs are divided into which of the following?

1 / 1 point

○ Drivers

○ Executors

● Tasks

○ Slots

✓ **Correct**
Jobs are subdivided into tasks. The input to a job is partitioned into one or more partitions. These partitions are the unit of work for each slot.

Read data in CSV format

Note: *In this reading you can see the steps involved in the process of reading data in CSV format.*

In this unit, you need to complete the exercises within a Databricks Notebook.

To begin, you need to have access to an Azure Databricks workspace. If you do not have a workspace available, follow the instructions below. Otherwise, you can skip to the bottom of the page to Clone the Databricks archive.

Unit Pre-requisites

Microsoft Azure Account: You will need a valid and active Azure account for the Azure labs. If you do not have one, you can sign up for a free trial

- If you are a Visual Studio Active Subscriber, you are entitled to Azure credits per month. You can refer to this link to find out more, including how to activate and start using your monthly Azure credit.
- If you are not a Visual Studio Subscriber, you can sign up for the FREE Visual Studio Dev Essentials program to create Azure free account.

Create the required resources

To complete this lab, you will need to deploy an Azure Databricks workspace in your Azure subscription.

Deploy an Azure Databricks workspace

1. Click the following button to open the Azure Resource Manager Template (ARM) template in the Azure portal. Deploy Databricks from the ARM Template

2. Provide the required values to create your Azure Databricks workspace:

- **Subscription**: Choose the Azure Subscription in which to deploy the workspace.
- **Resource Group**: Leave at Create new and provide a name for the new resource group.
- **Location**: Select a location near you for deployment. For the list of regions supported by Azure Databricks, see Azure services available by region.
- **Workspace Name**: Provide a name for your workspace.
- **Pricing Tier**: Ensure `premium` is selected.

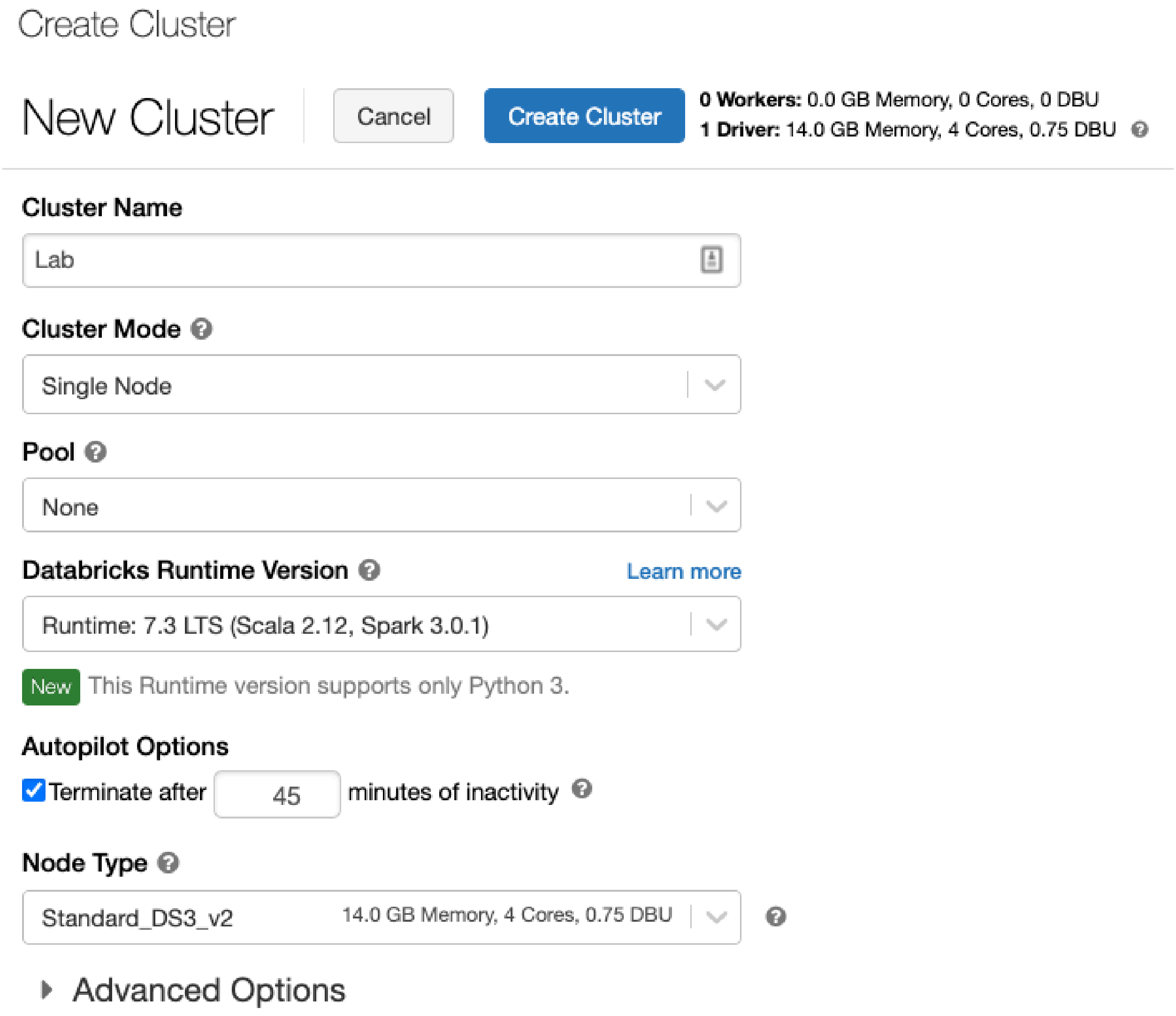
3. Accept the terms and conditions.

4. Select Purchase.

5. The workspace creation takes a few minutes. During workspace creation, the portal displays the Submitting deployment for Azure Databricks tile on the right side. You may need to scroll right on your dashboard to see the tile. There is also a progress bar displayed near the top of the screen. You can watch either area for progress.

Create a cluster

1. When your Azure Databricks workspace creation is complete, select the link to go to the resource.
2. Select **Launch Workspace** to open your Databricks workspace in a new tab.
3. In the left-hand menu of your Databricks workspace, select **Clusters**.
4. Select **Create Cluster** to add a new cluster.



Create Cluster

New Cluster

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU

Cluster Name
Lab

Cluster Mode
Single Node

Pool
None

Databricks Runtime Version
Runtime: 7.3 LTS (Scala 2.12, Spark 3.0.1)

New This Runtime version supports only Python 3.

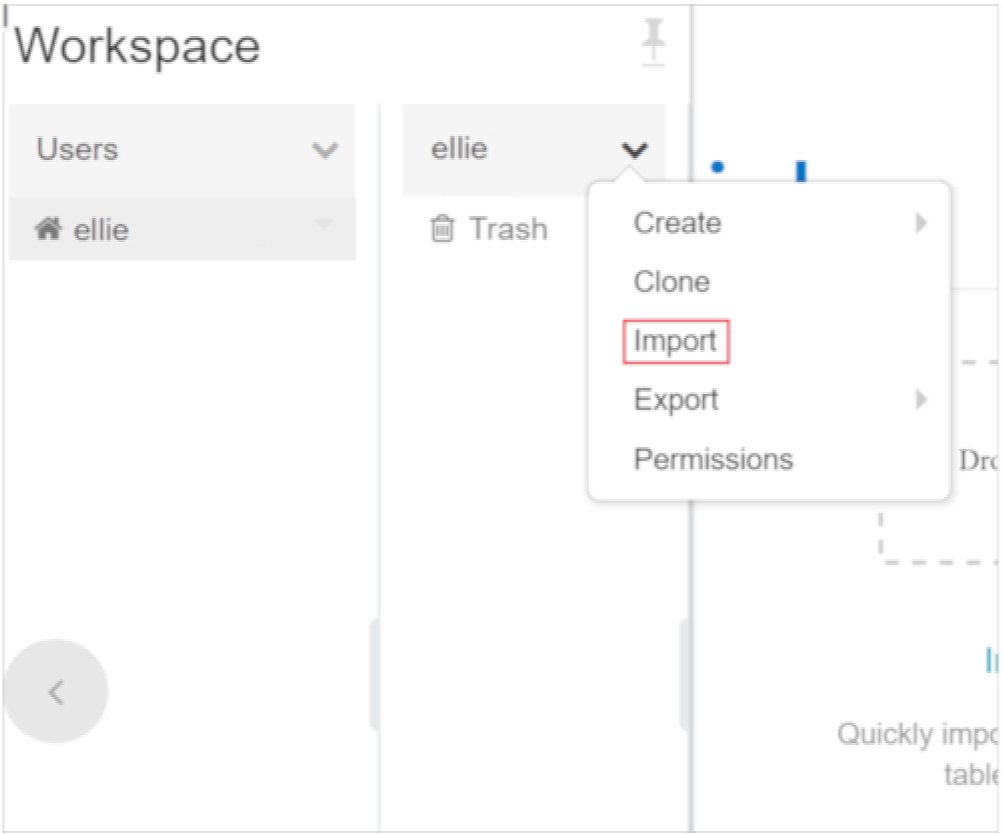
Autopilot Options
☑ Terminate after 45 minutes of inactivity

Node Type
Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU

The create cluster page.
5. Enter a name for your cluster. Use your name or initials to easily differentiate your cluster from your coworkers.
6. Select the **Cluster Mode**: **Single Node**.
7. Select the **Databricks RuntimeVersion**: **Runtime: 7.3 LTS (Scala 2.12, Spark 3.0.1)**.
8. Under **Autopilot Options**, leave the box **checked** and in the text box enter `45`.
9. Select the **Node Type**: **Standard_DS3_v2**.
10. Select **Create Cluster**.

Clone the Databricks archive

1. If you do not currently have your Azure Databricks workspace open: in the Azure portal, navigate to your deployed Azure Databricks workspace and select **Launch Workspace**.
2. In the left pane, select **Workspace** > **Users**, and select your username (the entry with the house icon).
3. In the pane that appears, select the arrow next to your name, and select **Import**.



The menu option to import the archive.
4. In the **Import Notebooks** dialog box, select the URL and paste in the following URL:

```
1
https://github.com/solliancenet/microsoft-learning-paths-databricks-notebooks/blob/master/data-engineering/DBC/03-Reading-and-writing-data-in-Azure-Databricks.dbc?raw=true
```

5. Select **Import**.
6. Select the **03-Reading-and-writing-data-in-Azure-Databricks** folder that appears.

Complete the following notebook

Open the **1.Reading Data - CSV** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

Within the notebook, you will:

- Start working with the API documentation
- Introduce the class **`SparkSession`** and other entry points
- Introduce the class **`DataFrameReader`**

Read data from:
CSV without a Schema

- CSV with a Schema

After you've completed the notebook, return to this screen, and continue to the next step.
Go to next item

Read data in JSON format

Note: *In this reading you can see the steps involved in the process of reading data in JSON format.*
In your Azure Databricks workspace, open the **03-Reading-and-writing-data-in-Azure-Databricks** folder that you imported within your user folder.
Open the **2.Reading Data - JSON** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.
Within the notebook, you will read data from:

- JSON without a Schema
- JSON with a Schema

After you've completed the notebook, return to this screen, and continue to the next step.

Read data in Parquet format

Note: *In this reading you can see the steps involved in the process of reading data in Parquet format.*
In your Azure Databricks workspace, open the **03-Reading-and-writing-data-in-Azure-Databricks** folder that you imported within your user folder.
Open the **3.Reading Data - Parquet** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.
Within the notebook, you will:

- Introduce the Parquet file format

Read data from:
Parquet files without a schema

- Parquet files with a schema

After you've completed the notebook, return to this screen, and continue to the next step.

Read data stored in tables and views

Note: *In this reading you can see the steps involved in the process of reading data stored in tables and views.*
In your Azure Databricks workspace, open the **03-Reading-and-writing-data-in-Azure-Databricks** folder that you imported within your user folder.
Open the **4.Reading Data - Tables and Views** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.
Within the notebook, you will:

- Demonstrate how to pre-register data sources in Azure Databricks
- Introduce temporary views over files
- Read data from tables/views

After you've completed the notebook, return to this screen, and continue to the next step.

Write data

Note: *In this reading you can see the steps involved in the process of writing data.*

In your Azure Databricks workspace, open the **03-Reading-and-writing-data-in-Azure-Databricks** folder that you imported within your user folder.

Open the **5.Writing Data** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

Within the notebook, you will:

- Write data to a Parquet file
- Read the Parquet file back and display the results

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

Exercises: Read and write data

In your Azure Databricks workspace, open the **03-Reading-and-writing-data-in-Azure-Databricks** folder that you imported within your user folder.

Open the **6. Reading Data - Lab** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

The goal of this exercise is to put into practice some of what you have learned about reading data with Apache Spark. The instructions are provided within the notebook, along with empty cells for you to do your work. At the bottom of the notebook are additional cells that will help verify that your work is accurate.

Note: You will find a corresponding notebook within the **`Solutions`** subfolder. This contains completed cells for the exercise. Refer to the notebook if you get stuck or simply want to see the solution.

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

Use common DataFrame methods

Note: *In this reading you can see the steps involved in the process of using common DataFrame methods.*

In your Azure Databricks workspace, open the **04-Working-With-Dataframes** folder that you imported within your user folder.

Open the **2.Use-common-dataframe-methods** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

Within the notebook, you will:

- Develop familiarity with the `DataFrame` APIs
- Use common DataFrame methods for performance
- Explore the Spark API documentation

After you've completed the notebook, return to this screen, and continue to the next step.

Use the display function

Note: *In this reading you can see the steps involved in the process of using the display function.*

In your Azure Databricks workspace, open the **04-Working-With-Dataframes** folder that you imported within your user folder.

Open the **3.Display-function** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

Within the notebook, you will learn the transformations:

- **`limit(..)`**
- **`select(..)`**
- **`drop(..)`**
- **`distinct()`**
- **`dropDuplicates(..)`**

and learn the actions:

- **`show(..)`**
- **`display(..)`**

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

Exercise: Distinct articles

In your Azure Databricks workspace, open the **04-Working-With-Dataframes** folder that you imported within your user folder.

Open the **4.Exercise: Distinct Articles** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

In this exercise, you read Parquet files, apply necessary transformations, perform a total count of records, then verify that all the data was correctly loaded.

As a bonus, try defining a schema that matches the data and update the read operation to use the schema.

Note: You will find a corresponding notebook within the `Solutions` subfolder. This contains completed cells for the exercise. Refer to the notebook if you get stuck or simply want to see the solution.

After you've completed the notebook, return to this screen, and continue to the next step.

1. How do you list files in DBFS within a notebook?

1 / 1 point

● %fs ls /my-file-path

○ ls /my-file-path

○ %fs dir /my-file-path

✓ Correct
Feedback: Correct. You added the file system magic to the cell before executing the ls command.

2. How do you infer the data types and column names when you read a JSON file?

1 / 1 point

● spark.read.option("inferSchema", "true").json(jsonFile)

○ spark.read.option("inferData", "true").json(jsonFile)

○ spark.read.inferSchema("true").json(jsonFile)

✓ Correct
This approach is the correct way to infer the file's schema.

3.

1 / 1 point

Which of the following SparkSession functions returns a DataFrameReader?

● read(..)

○ emptyDataFrame(..)

○ readStream(..)

○ createDataFrame(..)

✓ Correct
The function SparkSession.read() returns a DataFrameReader.

4. When using a notebook and a spark session. We can read a CSV file. Which of the following can be used to view the first couple thousand characters of a file?

1 / 1 point

○ %fs dir /mnt/training/wikipedia/pageviews/

● %fs head /mnt/training/wikipedia/pageviews/pageviews_by_second.tsv

○ %fs ls /mnt/training/wikipedia/pageviews/

✓ Correct
We can use %fs head ... to view the first couple thousand characters of a file.

5. You have created an Azure Databricks cluster, and you have access to a source file.

1 / 1 point

```
fileName = "dbfs:/mnt/training/wikipedia/clickstream/2015_02_clickstream.tsv"
```

You need to determine the structure of the file. Which of the following commands will assist with determining what the column and data types are?

○ .option("header", "false")

● .option("inferSchema", "true")

○ .option("inferSchema", "false")

○ .option("header", "true")

> ✓ Correct
> using .option("inferSchema", "true") Spark will automatically go through the file and infer the schema of each column.

6. In an Azure Databricks workspace you run the following command:

1 / 1 point

```
%fs head /mnt/training/wikipedia/pageviews/pageviews_by_second.tsv
```

The partial output from this command is as follows:

```
[Truncated to first 65536 bytes]

"timestamp"   "site"   "requests"

"2015-03-16T00:09:55"   "mobile"   1595

"2015-03-16T00:10:39"   "mobile"   1544

"2015-03-16T00:19:39"   "desktop"   2460

"2015-03-16T00:38:11"   "desktop"   2237

"2015-03-16T00:42:40"   "mobile"   1656

"2015-03-16T00:52:24"   "desktop"   2452
```

Which of the following pieces of information can be inferred from the command and the output?

Select all that apply.

□ The file has no header

☑ Two columns are strings, and one column is a number

> ✓ Correct
> The strings are enclosed in double quotes while the number column is not

□ All columns are strings

☑ The column is Tab separated

> ✓ Correct
> Feedback: The file is tab separated. This can be inferred from the file extension and the lack of other characters between each "column".

□ the file is a comma separated or CSV file

7. In an Azure Databricks you wish to create a temporary view that will be accessible to multiple notebooks. Which of the following commands will provide this feature?

1 / 1 point

○ createOrReplaceTempView(..)

○ createOrReplaceTempView(set_scope "Global")

● createOrReplaceGlobalTempView(..)

✓ Correct
Feedback: The spark method createOrReplaceGlobalTempView(..) is bound to the spark application allowing it to be read in this notebook and from another.

8. Which of the following is true in respect of Parquet Files?

1 / 1 point

Select all that apply.

☑ Is a Column-Oriented data store

✓ Correct
Parquet files are Column-Oriented.

□ Is a Row-Oriented data store

□ Designed for performance on small data sets

☑ Efficient data compression

✓ Correct
Parquet files provide efficient data compression.

☑ Open Source

✓ Correct
Parquet files are free Open Source.

☑ Is a splittable "file format".

✓ Correct
Parquet files are splittable.

Get started with Delta using Spark APIs

Note: *In this reading you can see the steps involved in the process of setting up Delta using Spark APIs.*

Delta Lake is included with Azure Databricks. You can start using it today. To quickly get started with Delta Lake, do the following:

Instead of **`parquet`**...

```
CREATE TABLE ...
USING parquet
...

dataframe
    .write
    .format("parquet")
    .save("/data")
```

simply say **`delta`**

```
CREATE TABLE ...
USING delta
...
```

```
dataframe
    .write
    .format("delta")
    .save("/data")
```

Using Delta with your existing Parquet tables

Step 1: Convert **`Parquet`** to **`Delta`** tables:

```
CONVERT TO DELTA parquet.`path/to/table` [NO STATISTICS]
[PARTITIONED BY (col_name1 col_type1, col_name2 col_type2, ...)]
```

Step 2: Optimize layout for fast queries:

```
OPTIMIZE events

WHERE date >= current_timestamp() - INTERVAL 1 day

ZORDER BY (eventType)
```

Basic syntax

Two of the core features of Delta Lake are performing upserts (insert/updates) and Time Travel operations. We will explore these concepts more within the notebooks in this module.

To UPSERT means to "UPdate" and "inSERT". In other words, UPSERT is literally TWO operations. It is not supported in traditional data lakes, as running an UPDATE could invalidate data that is accessed by the subsequent INSERT operation.

Using Delta Lake, however, we can do UPSERTS. Delta Lake combines these operations to guarantee atomicity to

- INSERT a row
- if the row already exists, UPDATE the row.

Upsert syntax

Upserting, or merging, in Delta Lake provides fine-grained updates of your data. The following syntax shows how to perform an Upsert:

```
MERGE INTO customers -- Delta table
```

```
USING updates
ON customers.customerId = source.customerId
WHEN MATCHED THEN
    UPDATE SET address = updates.address
WHEN NOT MATCHED
    THEN INSERT (customerId, address) VALUES (updates.customerId, updates.address)
```

See update table data syntax documentation.

Time Travel syntax

Because Delta Lake is version controlled, you have the option to query past versions of the data. Using a single file storage system, you now have access to several versions your historical data, ensuring that your data analysts will be able to replicate their reports (and compare aggregate changes over time) and your data scientists will be able to replicate their experiments.
Other time travel use cases are:

- Re-creating analyses, reports, or outputs (for example, the output of a machine learning model). This could be useful for debugging or auditing, especially in regulated industries.
- Writing complex temporal queries.
- Fixing mistakes in your data.
- Providing snapshot isolation for a set of queries for fast changing tables.

Example of using time travel to reproduce experiments and reports:

```
SELECT count(*) FROM events
TIMESTAMP AS OF timestamp
```

```
SELECT count(*) FROM events

VERSION AS OF version
```

```
spark.read.format("delta").option("timestampAsOf",
timestamp_string).load("/events/")
```

If you need to rollback accidental or bad writes:

```
INSERT INTO my_table

    SELECT * FROM my_table TIMESTAMP AS OF

    date_sub( current_date(), 1)
```

See time travel syntax documentation.

Exercise: Work with basic Delta Lake functionality

In your Azure Databricks workspace, open the **09-Building-And-Querying-A-Delta-Lake** folder that you imported within your user folder.

Open the **2.Delta-Lake-Basics-Lab-1** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

In this exercise, you:

- Create a new Delta Lake from aggregate data of an existing Delta Lake.
- UPSERT records into a Delta lake.
- Append new data to an existing Delta Lake.

The instructions are provided within the notebook, along with empty cells for you to do your work. At the bottom of the notebook are additional cells that will help verify that your work is accurate.

Note: You will find a corresponding notebook within the `Solutions` subfolder. This contains completed cells for the exercise. Refer to the notebook if you get stuck or simply want to see the solution.

After you've completed the notebook, return to this screen, and continue to the next step.

Describe how Azure Databricks manages Delta Lake

Note: *In this reading you can see the steps involved in the process of using Azure Databricks to manage Data Lake.*

In your Azure Databricks workspace, open the **09-Building-And-Querying-A-Delta-Lake** folder that you imported within your user folder.

Open the **3.Managed-Delta-Lake** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

Within the notebook, you will discover Delta Lake's key features that allow for query optimization and garbage collection, resulting in improved performance.

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

1. Delta Lake provides snapshots of data enabling developers to access and revert to earlier versions of data for audits, rollbacks or to reproduce experiments. This functionality is referred to as?

1 / 1 point

○ Schema Enforcement

○ ACID Transactions

● Time Travel

○ Schema Evolution

✓ Correct
Delta Lake provides snapshots of data enabling developers to access and revert to earlier versions of data for audits, rollbacks or to reproduce experiments.

2. One of the core features of Delta Lake is performing upserts. Which of the following statements is true in regard to Upsert?

1 / 1 point

● UpSert is literally TWO operations. Update / Insert

○ Upsert is supported in traditional data lakes

○ Upsert is a new DML statement for SQL syntax

✓ Correct
To UPSERT means to "UPdate" and "inSERT". In other words, UPSERT is literally TWO operations. It is not supported in traditional data lakes.

3. When discussing Delta Lake, there is often a reference to the concept of Bronze, Silver and Gold tables. These levels refer to the state of data refinement as data flows through a processing pipeline and are conceptual guidelines. Based on these table concepts the refinements in Silver tables generally relate to **which of the following?**

1 / 1 point

○ Raw data (or very little processing)

● Data that is directly queryable and ready for insights

○ Highly refined views of the data

✓ Correct
Silver tables generally relate to data that is directly queryable and ready for insights.

4. What is the Databricks Delta command to display metadata?

1 / 1 point

○ MSCK DETAIL tablename

● DESCRIBE DETAIL tableName

○ SHOW SCHEMA tablename

✓ Correct
You display metadata by using DESCRIBE DETAIL tableName.

5. How do you perform UPSERT in a Delta dataset?

0 / 1 point

● Use UPSERT INTO my-table

○ Use UPSERT INTO my-table /MERGE

○ Use MERGE INTO my-table USING data-to-upsert

ⓧ Incorrect
The syntax isn't correct for this operation.

6. What optimization does the following command perform: OPTIMIZE Students ZORDER BY Grade?

1 / 1 point

○ Creates an order-based index on the Grade field to improve filters against that field.

● Ensures that all data backing, for example, Grade=8 is colocated, then rewrites the sorted data into new Parquet files.

○ Ensures that all data backing, for example, Grade=8 is colocated, then updates a graph that routes requests to the appropriate files.

✓ Correct
ZOrdering colocates related information in the same set of files.

Write user defined functions

In this unit, you need to complete the exercises within a Databricks Notebook. To begin, you need to have access to an Azure Databricks workspace with an interactive cluster. If you do not have a workspace and/or the required cluster, follow the instructions below. Otherwise, you can skip to the bottom of the page to Clone the Databricks archive.

Unit Pre-requisites

Microsoft Azure Account: You will need a valid and active Azure account for the Azure labs. If you do not have one, you can sign up for a free trial

- If you are a Visual Studio Active Subscriber, you are entitled to Azure credits per month. You can refer to this link to find out more including how to activate and start using your monthly Azure credit.
- If you are not a Visual Studio Subscriber, you can sign up for the FREE Visual Studio Dev Essentials program to create Azure free account.

Create the required resources

To complete this lab, you will need to deploy an Azure Databricks workspace in your Azure subscription.

Deploy an Azure Databricks workspace

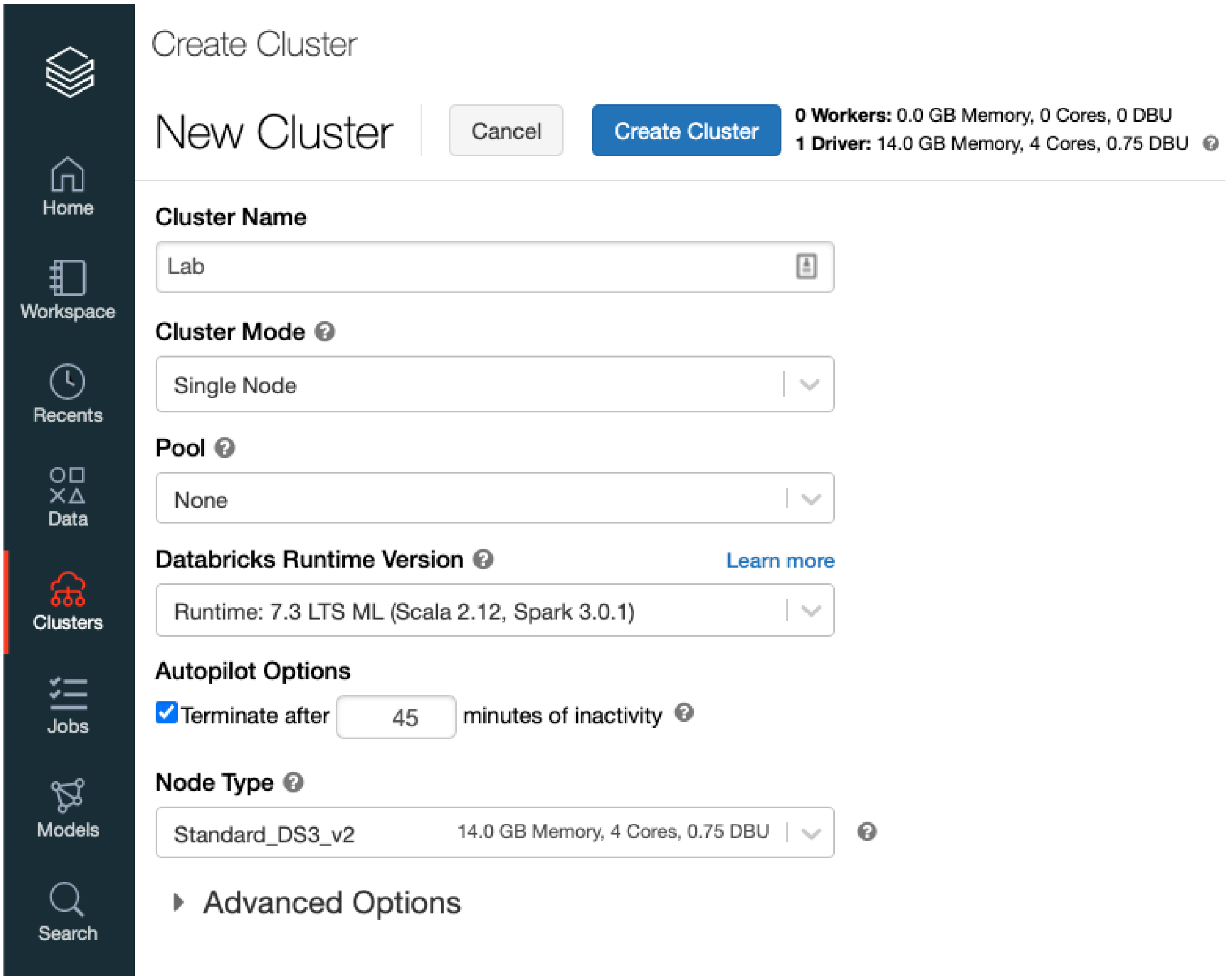
1. Click the following button to open the Azure Resource Manager template in the Azure portal.
Deploy Databricks from the Azure Resource Manager Template
2. Provide the required values to create your Azure Databricks workspace:
    - **Subscription**: Choose the Azure Subscription in which to deploy the workspace.
    - **Resource Group**: Leave at Create new and provide a name for the new resource group.
    - **Location**: Select a location near you for deployment. For the list of regions supported by Azure Databricks, see Azure services available by region.
    - **Workspace Name**: Provide a name for your workspace.
    - **Pricing Tier**: Ensure `premium` is selected.

3. Accept the terms and conditions.
4. Select Purchase.
5. The workspace creation takes a few minutes. During workspace creation, the portal displays the Submitting deployment for Azure Databricks tile on the right side. You may need to scroll right on your dashboard to see the tile. There is also a progress bar displayed near the top of the screen. You can watch either area for progress.

Create a cluster

1. When your Azure Databricks workspace creation is complete, select the link to go to the resource.

2. Select **Launch Workspace** to open your Databricks workspace in a new tab.
3. In the left-hand menu of your Databricks workspace, select **Clusters**.
4. Select **Create Cluster** to add a new cluster.



Create Cluster

New Cluster

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU

Cluster Name: Lab

Cluster Mode: Single Node

Pool: None

Databricks Runtime Version: Runtime: 7.3 LTS ML (Scala 2.12, Spark 3.0.1)

Autopilot Options

☑ Terminate after 45 minutes of inactivity

Node Type: Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU

Advanced Options

The create cluster page.

5. Enter a name for your cluster. Use your name or initials to easily differentiate your cluster from your coworkers.

6. Select the **Cluster Mode**: **Single Node**

7. Select the **Databricks RuntimeVersion**: **Runtime: 7.3 LTS ML (Scala 2.12, Spark 3.0.1)** (remember to select the **ML** version).

8. Under **Autopilot Options**, leave the box **checked** and in the text box enter `45`.

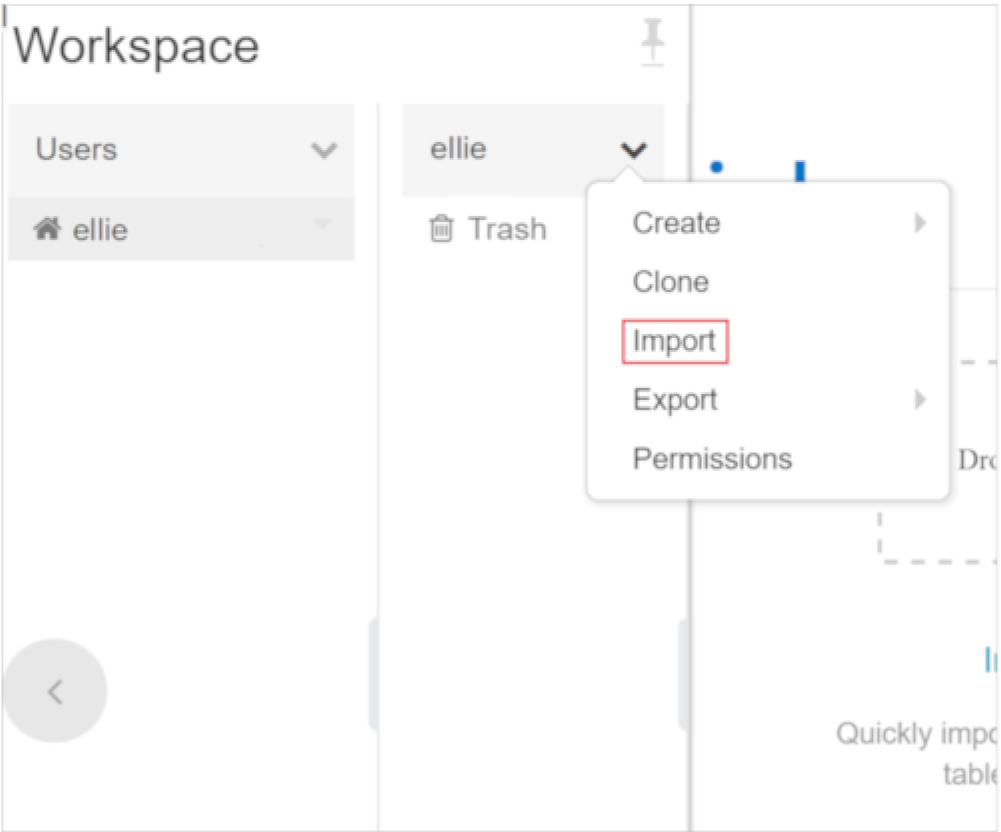
9. Select the **Node Type**: **Standard_DS3_v2**

10. Select **Create Cluster**.

Clone the Databricks archive

1. If you do not currently have your Azure Databricks workspace open: in the Azure portal, navigate to your deployed Azure Databricks workspace and select **Launch Workspace**.

2. In the left pane, select **Workspace** > **Users**, and select your username (the entry with the house icon).
3. In the pane that appears, select the arrow next to your name, and select **Import**.



The menu option to import the archive.

4. In the **Import Notebooks** dialog box, select the URL and paste in the following URL: **`https://github.com/MicrosoftDocs/mslearn_databricks/blob/main/udf/1.1.0/Labs.dbc`**

5. Select **Import**.

6. Select the **udf** folder that appears.

Complete the following notebook

Open the **1. User Defined Functions** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

Within the notebook, you will:

- Create, register, and invoke UDFs
- Create, register, and invoke vectorized UDF

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

Exercise: Perform Extract, Transform, Load (ETL) operations using user-defined functions

In your Azure Databricks workspace, open the **udf** folder that you imported within your user folder.

Open the **2. Exercise User Defined Functions** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

In this exercise, you will be create UDFs to do ETL. The dataset has duplicate records and the format for the social security numbers is inconsistent. The ETL's job is to remove duplicate records and standardize the format for the social security numbers.

Note

You will find a corresponding notebook within the `Solutions` subfolder. This contains completed cells for the exercise. Refer to the notebook if you get stuck or simply want to see the solution.

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

1. You have a dataframe which you preprocessed and filtered down to only the relevant columns.

   1 / 1 point

   The columns are: id, host_name, bedrooms, neighbourhood_cleansed, price.

   You want to retrieve the first initial from the host_name field.

   How would you write that function in local Python/Scala?

   ● def firstInitialFunction(name):

   return name[0]

   firstInitialFunction("Steven")

   ○ new firstInitialFunction(name):

   return name[1]

   firstInitialFunction("Steven")

   ○ new firstInitialFunction(name):

   extract name[]

   firstInitialFunction("Steven")

   ○ def firstInitialFunction(name):

   get name[0]

   firstInitialFunction("Steven")

   ✓ Correct
   This is the correct code that will get the first initial from the host_name column.

2. You've written a function that retrieves the first initial letter from the host_name column.

   1 / 1 point

   You now want to define it as a user-defined named firstInitialUDF.

   How you define that using Python/Scala?

   ○ firstInitial = udf(firstInitial)

   ○ firstInitialUDF = firstInitialFunction()

   ○ firstInitial = udf(firstInitialFunction)

   ● firstInitialUDF = udf(firstInitialFunction)

   ✓ Correct
   This is the correct code that will create your UDF.

3. If you want to create the UDF in the SQL namespace, what class do you need to use?

   0 / 1 point

   ○ spark.sql.register

   ○ spark.sql.read

   ○ spark.sql.create

   ● spark.udf.register

   ✕ Incorrect
   This class will register your functions as a UDF.

4. Which is another syntax that you can use to define a UDF in Python?

   1 / 1 point

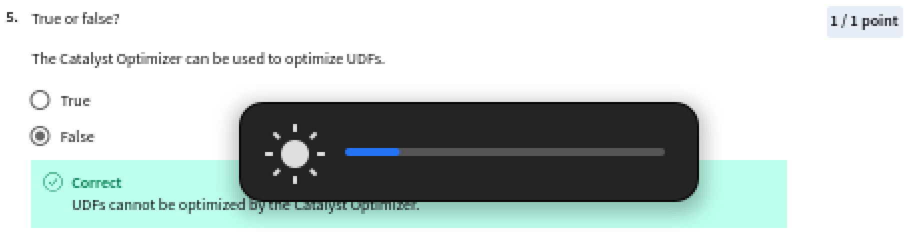
   ○ Designer

   ○ Capsulator

   ○ Wrapper

   ● Decorator

   ✓ Correct
   Alternatively, you can define a UDF using decorator syntax in Python with the dataType the function will



5. True or false?

1 / 1 point

The Catalyst Optimizer can be used to optimize UDFs.

○ True

● False

✓ Correct
UDFs cannot be optimized by the Catalyst Optimizer.

1. Delta Lake enables you to make changes to a table schema that can be applied automatically, without the need for DDL modifications. This functionality is referred to as?

1 / 1 point

○ ACID Transactions

○ Time Travel

● Schema Evolution

○ Schema Enforcement

✓ Correct
Delta Lake enables you to make changes to a table schema that can be applied automatically, without the need for DDL modifications.

2. One of the core features of Delta Lake is performing upserts. Which of the following statements is true regarding Upsert?

1 / 1 point

○ Upsert is a new DML statement for SQL syntax

○ Upsert is supported in traditional data lakes

● Upsert is literally TWO operations. Update / Insert

✓ Correct
To UPSERT means to "UPdate" and "inSERT". In other words, UPSERT is literally TWO operations. It is not supported in traditional data lakes.

3. What is the Databricks Delta command to display metadata?

1 / 1 point

○ MSCK DETAIL tablename

● DESCRIBE DETAIL tableName

○ SHOW SCHEMA tablename

✓ Correct
You display metadata by using DESCRIBE DETAIL tableName.

4. What optimization does the following command perform: OPTIMIZE Customers ZORDER BY City?

1 / 1 point

○ Creates an order-based index on the City field to improve filters against that field

○ Ensures that all data backing, for example, City="London" is colocated, then updates a graph that routes requests to the appropriate files.

● Ensures that all data backing, for example, City='London' is colocated, then rewrites the sorted data into new Parquet files.

✓ Correct
ZOrdering colocates related information in the same set of files.

5. You are planning on registering a user-defined function, g, as g_function in a SQL namespace. How would you achieve this programmatically?

1 / 1 point

● spark.udf.register("g_function", g)

○ spark.udf.register(g, "g_function")

○ spark.register_udf("g_function", g)

✓ Correct
This is the correct syntax to register the UDF in the SQL namespace.

6. True or False?

1 / 1 point

User-defined Functions cannot operate on DataFrames.

● No

○ Yes

✓ Correct
UDF can operate on DataFrames.

7. Suppose you already have a dataframe which only contains relevant columns.

1 / 1 point

The columns are: id, employee_name, age, gender.

You want to retrieve the first initial from the employee_name field by creating a local function in Python/Scala. Which of the following code can be used to get the first initial from the host_name column?

○ new firstInitialFunction(name):

```
 extract name[]

firstInitialFunction("Steven")
```

○ def firstInitialFunction(name):

```
 get name[0]

firstInitialFunction("Steven")
```

● def firstInitialFunction(name):

```
 return name[0]

firstInitialFunction("Steven")
```

✓ Correct
This is the correct code that will get the first initial from the host_name column.

In this unit, you need to complete the exercises within a Databricks Notebook. To begin, you need to have access to an Azure Databricks workspace with an interactive cluster. If you do not have a workspace and/or the required cluster, follow the instructions below. Otherwise, you can skip to the bottom of the page to Clone the Databricks archive.

Unit prerequisites

Microsoft Azure Account: You will need a valid and active Azure account for the Azure labs. If you do not have one, you can sign up for a free trial

- If you are a Visual Studio Active Subscriber, you are entitled to Azure credits per month. You can refer to this link to find out more including how to activate and start using your monthly Azure credit.
- If you are not a Visual Studio Subscriber, you can sign up for the FREE Visual Studio Dev Essentials program to create Azure free account.

Create the required resources

To complete this lab, you will need to deploy an Azure Databricks workspace in your Azure subscription.

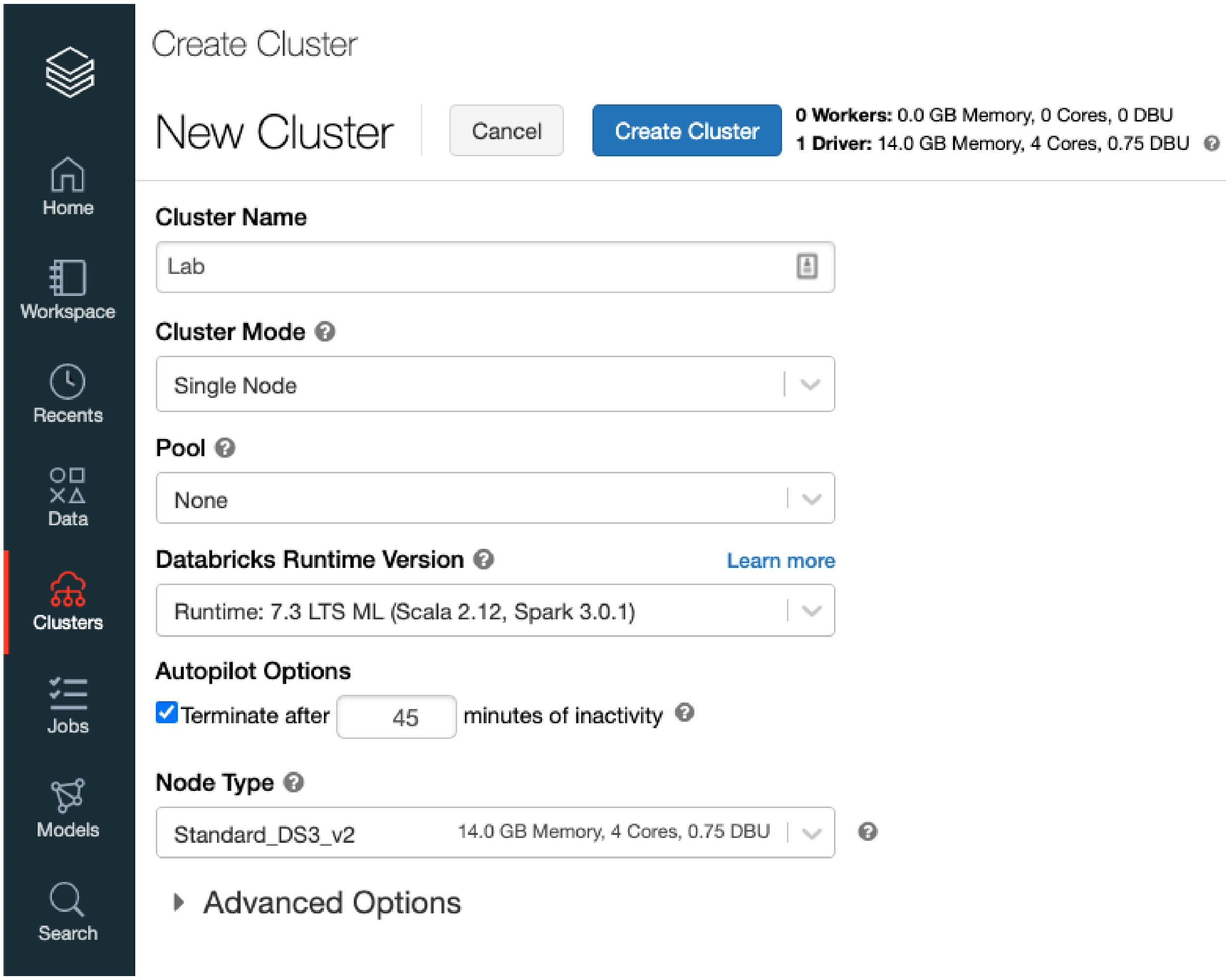
Deploy an Azure Databricks workspace

1. Select the following link to open the Azure Resource Manager template in the Azure portal.
Deploy Databricks from the Azure Resource Manager Template
2. Provide the required values to create your Azure Databricks workspace:
    - **Subscription**: Choose the Azure Subscription in which to deploy the workspace.
    - **Resource Group**: Leave at Create new and provide a name for the new resource group.
    - **Location**: Select a location near you for deployment. For the list of regions supported by Azure Databricks, see Azure services available by region.
    - **Workspace Name**: Provide a name for your workspace.
    - **Pricing Tier**: Ensure `premium` is selected.

3. Accept the terms and conditions.
4. Select Purchase.
5. The workspace creation takes a few minutes. During workspace creation, the portal displays the Submitting deployment for Azure Databricks tile on the right side. You may need to scroll right on your dashboard to see the tile. There is also a progress bar displayed near the top of the screen. You can watch either area for progress.

Create a cluster

1. When your Azure Databricks workspace creation is complete, select the link to go to the resource.
2. Select **Launch Workspace** to open your Databricks workspace in a new tab.
3. In the left-hand menu of your Databricks workspace, select **Clusters**.
4. Select **Create Cluster** to add a new cluster.



Create Cluster

New Cluster

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU

Cluster Name: Lab

Cluster Mode: Single Node

Pool: None

Databricks Runtime Version: Runtime: 7.3 LTS ML (Scala 2.12, Spark 3.0.1)

Autopilot Options

☑ Terminate after 45 minutes of inactivity

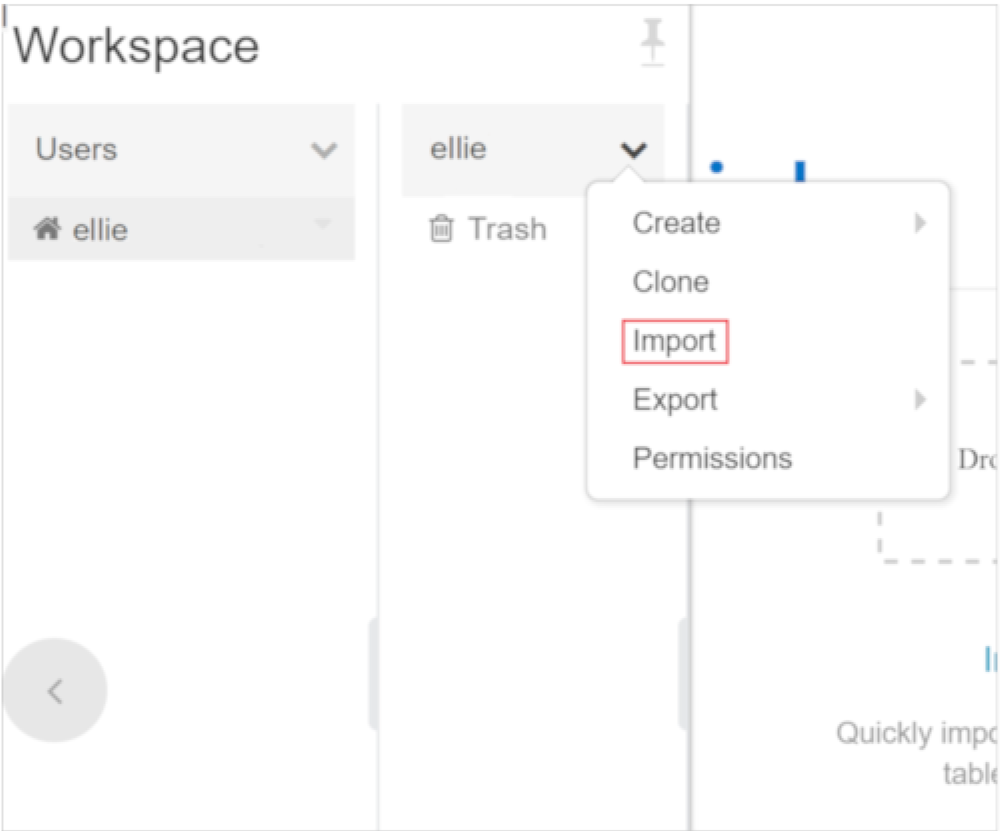
Node Type: Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU

Advanced Options

The create cluster page.
5. Enter a name for your cluster. Use your name or initials to easily differentiate your cluster from your coworkers.
6. Select the **Cluster Mode**: **Single Node**
7. Select the **Databricks RuntimeVersion**: **Runtime: 7.3 LTS ML (Scala 2.12, Spark 3.0.1)** (remember to select the **ML** version).
8. Under **Autopilot Options**, leave the box **checked** and in the text box enter `45`.
9. Select the **Node Type**: **Standard_DS3_v2**
10. Select **Create Cluster**.

Clone the Databricks archive

1. If you do not currently have your Azure Databricks workspace open: in the Azure portal, navigate to your deployed Azure Databricks workspace and select **Launch Workspace**.
2. In the left pane, select **Workspace** > **Users**, and select your username (the entry with the house icon).
3. In the pane that appears, select the arrow next to your name, and select **Import**.



The menu option to import the archive.

4. In the **Import Notebooks** dialog box, select the URL and paste in the following URL: `https://github.com/MicrosoftDocs/mslearn_databricks/blob/main/ml/1.1.0/Labs.dbc`
5. Select **Import**.
6. Select the **ml** folder that appears.

Complete the following notebook

Open the **1. What is Machine Learning** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

Within the notebook, you will:

- Define machine learning
- Differentiate supervised and unsupervised tasks
- Identify regression and classification tasks
- Train a model, interpret the results, and create predictions

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

Exercise: Train a model and create predictions

In your Azure Databricks workspace, open the **ml** folder that you imported within your user folder.

Open the **2. Exercise Train a Model and Create Predictions** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

In this exercise, you will train a model using the Boston housing dataset to predict price of a home. You will use the trained model to predict house prices on test data.

Note

You will find a corresponding notebook within the `Solutions` subfolder. This contains completed cells for the exercise. Refer to the notebook if you get stuck or simply want to see the solution.

After you've completed the notebook, return to this screen, and continue to the next step.

Understand data using exploratory data analysis

In your Azure Databricks workspace, open the **ml** folder that you imported within your user folder.

Open the **3. Exploratory Analysis** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

Within the notebook, you will:

- Identify the main objectives of exploratory analysis
- Calculate statistical moments to determine the center and spread of data
- Create plots of data including histograms and scatterplots
- Calculate correlations between variables
- Explore more advanced plots to visualize the relation between variables

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

Exercise: Perform exploratory data analysis

In your Azure Databricks workspace, open the **ml** folder that you imported within your user folder.

Open the **4. Exercise Exploratory Analysis** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

In this exercise, you will perform exploratory analysis on the bike sharing dataset by calculating and interpreting summary statistics, creating basic plots, and calculating correlations.

Note

You will find a corresponding notebook within the `Solutions` subfolder. This contains completed cells for the exercise. Refer to the notebook if you get stuck or simply want to see the solution.

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

Describe machine learning workflows

In your Azure Databricks workspace, open the **ml** folder that you imported within your user folder.

Open the **5. ML Workflows** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

Within the notebook, you will:

- Define the data analytics development cycle
- Motivate and perform a split between training and test data
- Train a baseline model
- Evaluate a baseline model's performance and improve it

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

Exercise: Build and evaluate a baseline machine learning model

In your Azure Databricks workspace, open the **ml** folder that you imported within your user folder.

Open the **6. Exercise ML Workflows** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

In this exercise, you will enact parts of the Machine Learning workflow, such as do train-test split on a dataset, and then build and evaluate a baseline model. Optionally, you can try to beat the baseline model by training a linear regression model.

Note

You will find a corresponding notebook within the `Solutions` subfolder. This contains completed cells for the exercise. Refer to the notebook if you get stuck or simply want to see the solution.

After you've completed the notebook, return to this screen, and continue to the next step.

1. Which are the two main types of Machine Learning problems?

   1 / 1 point

   Select all that apply.

   ☐ Classification

   ☑ Unsupervised learning

   > ✓ Correct
   > In unsupervised learning, the data points aren't labeled—the algorithm labels them for you by organizing the data or describing its structure. This technique is useful when you don't know what the outcome should look like.

   ☑ Supervised learning

   > ✓ Correct
   > In supervised learning, algorithms make predictions based on a set of labeled examples that you provide. This technique is useful when you know what the outcome should look like.

   ☐ Regression

2. You are tasked with using Machine Learning to develop an intelligent app that can predict real estate prices.

   1 / 1 point

   The dataset you're using contains input features and the output variable.

   Which type of Machine learning problem is this?

   ● Supervised

   ○ Unsupervised

   ○ Semi-supervised

   > ✓ Correct
   > Supervised learning looks to predict the value of some outcome based on one or more input measures. You would use a regression algorithm to predict the label, or the price, in this scenario.

3. Which type of operation does VectorAssembler perform on the features of the model?

   1 / 1 point

   ● Transform

   ○ Load

   ○ Extract

   ○ Estimate

   > ✓ Correct
   > VectorAssembler is a transformer, which implements a .transform() method.

4. Which are the types of variables that can be found in Machine Learning? Select all that apply.

1 / 1 point

□ Underestimated

□ Overestimated

☑ Qualitative

✓ Correct
Qualitative values take on a set number of classes or categories.

☑ Quantitative

✓ Correct
Quantitative values are numeric and generally unbounded, taking any positive or negative value.

5. Which are some examples of quantitative variables? Select all that apply.

0 / 1 point

☑ Gender

☒ This should not be selected
This is an example of a qualitative variable.

☑ Salary

✓ Correct
This is an example of a quantitative variable.

□ State of residence

☑ Age

✓ Correct
This is an example of a quantitative variable.

Perform featurization of the dataset

In this unit, you need to complete the exercises within a Databricks Notebook. To begin, you need to have access to an Azure Databricks workspace with an interactive cluster. If you do not have a workspace and/or the required cluster, follow the instructions below. Otherwise, you can skip to the bottom of the page to Clone the Databricks archive.

Unit Pre-requisites

Microsoft Azure Account: You will need a valid and active Azure account for the Azure labs. If you do not have one, you can sign up for a free trial

- If you are a Visual Studio Active Subscriber, you are entitled to Azure credits per month. You can refer to this link to find out more including how to activate and start using your monthly Azure credit.
- If you are not a Visual Studio Subscriber, you can sign up for the FREE Visual Studio Dev Essentials program to create Azure free account.

Create the required resources

To complete this lab, you will need to deploy an Azure Databricks workspace in your Azure subscription.

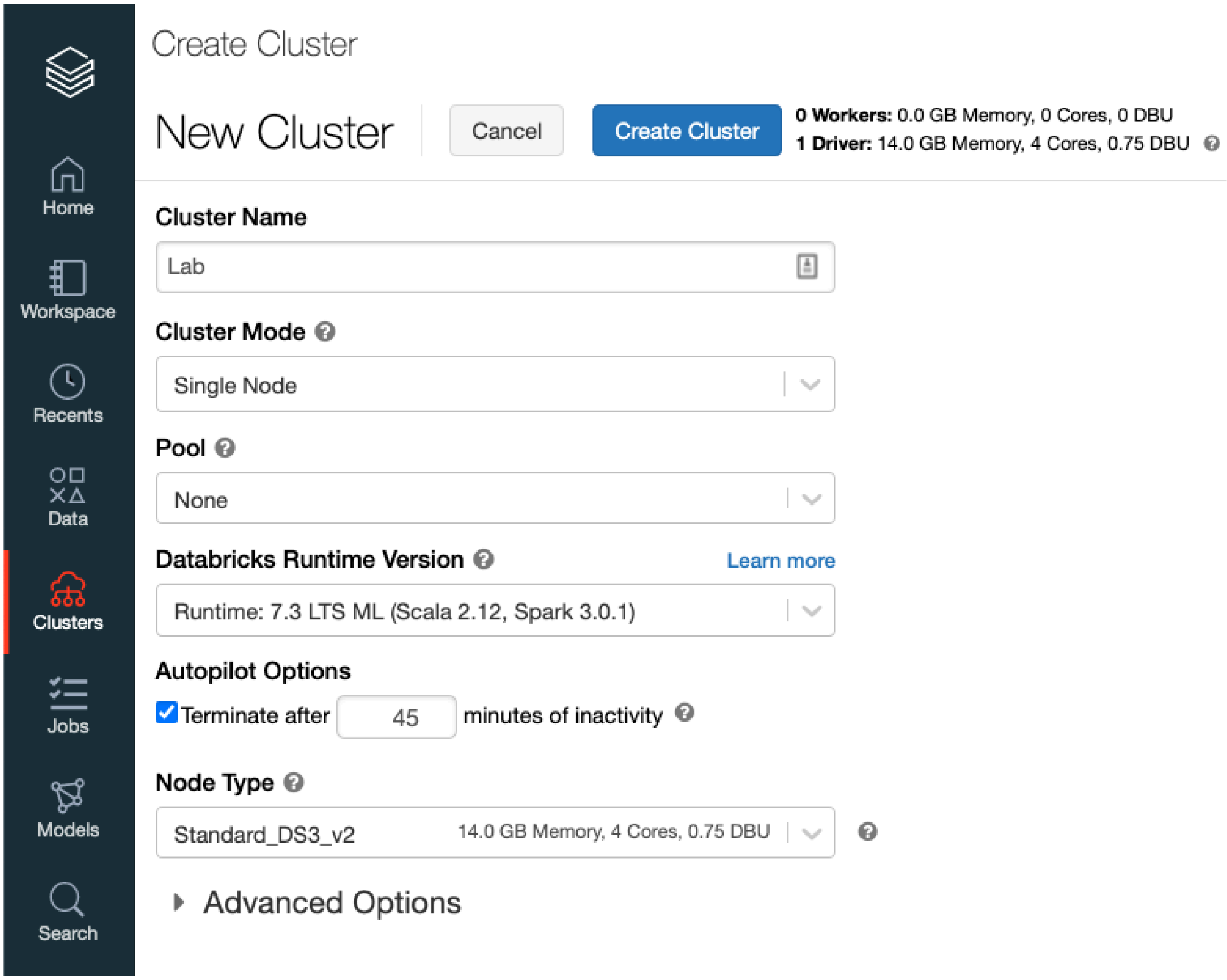
Deploy an Azure Databricks workspace

1. Click the following button to open the Azure Resource Manager template in the Azure portal.
Deploy Databricks from the Azure Resource Manager Template
2. Provide the required values to create your Azure Databricks workspace:
   - **Subscription**: Choose the Azure Subscription in which to deploy the workspace.
   - **Resource Group**: Leave at Create new and provide a name for the new resource group.
   - **Location**: Select a location near you for deployment. For the list of regions supported by Azure Databricks, see Azure services available by region.
   - **Workspace Name**: Provide a name for your workspace.
   - **Pricing Tier**: Ensure `premium` is selected.

3. Accept the terms and conditions.
4. Select Purchase.
5. The workspace creation takes a few minutes. During workspace creation, the portal displays the Submitting deployment for Azure Databricks tile on the right side. You may need to scroll right on your dashboard to see the tile. There is also a progress bar displayed near the top of the screen. You can watch either area for progress.

Create a cluster

1. When your Azure Databricks workspace creation is complete, select the link to go to the resource.
2. Select **Launch Workspace** to open your Databricks workspace in a new tab.
3. In the left-hand menu of your Databricks workspace, select **Clusters**.
4. Select **Create Cluster** to add a new cluster.



Create Cluster

New Cluster

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU

Cluster Name: Lab

Cluster Mode: Single Node

Pool: None

Databricks Runtime Version: Runtime: 7.3 LTS ML (Scala 2.12, Spark 3.0.1)

Autopilot Options

☑ Terminate after 45 minutes of inactivity

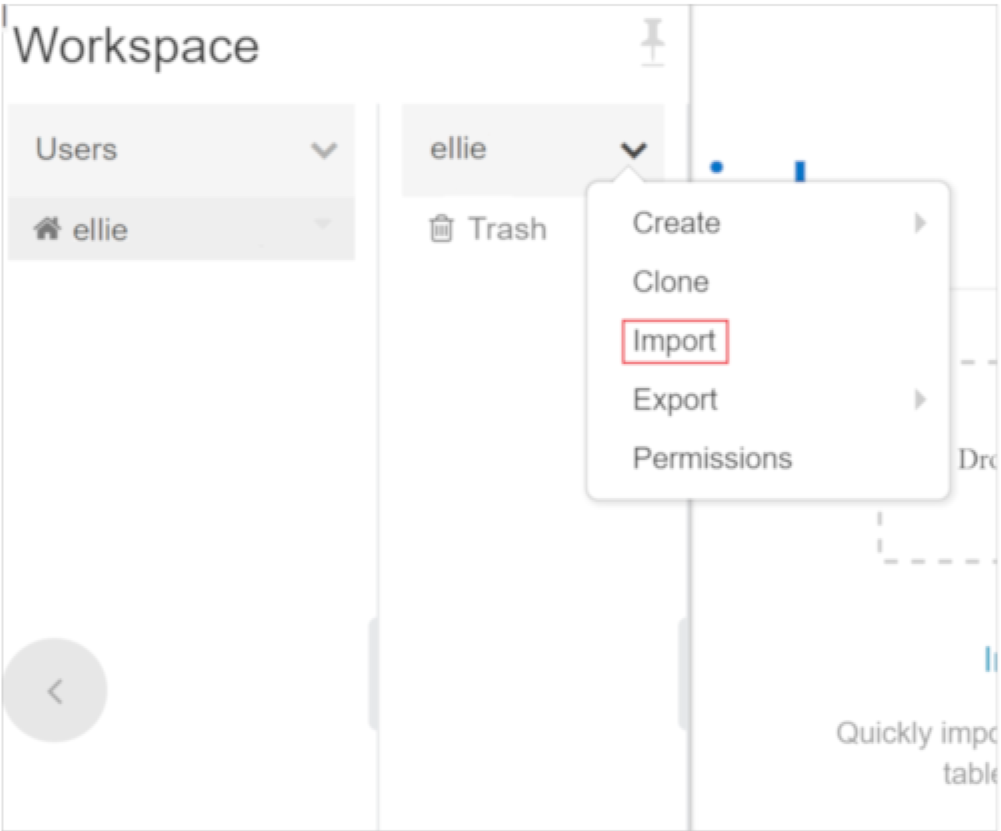
Node Type: Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU

The create cluster page.

5. Enter a name for your cluster. Use your name or initials to easily differentiate your cluster from your coworkers.
6. Select the **Cluster Mode**: **Single Node**
7. Select the **Databricks RuntimeVersion**: **Runtime: 7.3 LTS ML (Scala 2.12, Spark 3.0.1)** (remember to select the **ML** version).
8. Under **Autopilot Options**, leave the box **checked** and in the text box enter `45`.
9. Select the **Node Type**: **Standard_DS3_v2**
10. Select **Create Cluster**.

Clone the Databricks archive

1. If you do not currently have your Azure Databricks workspace open: in the Azure portal, navigate to your deployed Azure Databricks workspace and select **Launch Workspace**.
2. In the left pane, select **Workspace** > **Users**, and select your username (the entry with the house icon).
3. In the pane that appears, select the arrow next to your name, and select **Import**.



The menu option to import the archive.

4. In the **Import Notebooks** dialog box, select the URL and paste in the following URL: `https://github.com/MicrosoftDocs/mslearn_databricks/blob/main/ml-model/1.1.0/Labs.dbc`
5. Select **Import**.
6. Select the **ml-model** folder that appears.

Complete the following notebook

Open the **1. Featurization** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

Within the notebook, you will:

- Differentiate Spark transformers, estimators, and pipelines
- One-hot encode categorical features
- Impute missing data
- Combine different featurization stages into a pipeline

After you've completed the notebook, return to this screen, and continue to the next step.

Exercise: Finish featurization of the dataset

In your Azure Databricks workspace, open the **ml-model** folder that you imported within your user folder.

Open the **2. Exercise Featurization** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

In this exercise you are going featurize categorical data by dividing it into bins. You will also remove records with incorrect data values, and standardize the label column.

Note

You will find a corresponding notebook within the `Solutions` subfolder. This contains completed cells for the exercise. Refer to the notebook if you get stuck or simply want to see the solution.

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

Understand regression modeling

In your Azure Databricks workspace, open the **ml-model** folder that you imported within your user folder.

Open the **3. Regression Modeling** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

Within the notebook, you will:

- Motivate the use of linear regression
- Train a simple regression model
- Interpret regression models
- Train a multivariate regression model

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

Exercise: Build and interpret a regression model

In your Azure Databricks workspace, open the **ml-model** folder that you imported within your user folder.

Open the **4. Exercise Regression Modeling** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

In this exercise, you will train a regression model using the Boston housing dataset to predict price of a home and interpret the statistical significance of the trained model coefficients.

Note

You will find a corresponding notebook within the **`Solutions`** subfolder. This contains completed cells for the exercise. Refer to the notebook if you get stuck or simply want to see the solution. After you've completed the notebook, return to this screen, and continue to the next step.

1. Which are the three main building blocks that form the machine learning process in Spark from featurization to model training and deployment? Select all that apply.

1 / 1 point

☐ Extractor

☑ Estimator

> Correct
> This is one of the main abstractions used in Spark.

☑ Pipelines

> Correct
> This is one of the main abstractions used in Spark.

☐ Loader

☑ Transformer

> Correct
> This is one of the main abstractions used in Spark.

2. From the Spark's machine learning library MLlib, which one of the following abstractions takes a dataframe as an input and returns a new dataframe with one or more columns appended to it?

1 / 1 point

● Transformer

○ Pipeline

○ Estimator

> Correct
> Transformers achieve this by implementing a .transform() method.

3. True or false?

1 / 1 point

Random forest models also need one-hot encoding.

○ True

● False

> Correct
> Certain models, such as random forest, do not need one-hot encoding (and can actually be negatively affected by the process).

4. When dealing with null values, which strategy can you implement if you want to see missing data later on without violating the schema?

1 / 1 point

○ Dropping the records

○ Advanced inputting

● Adding a placeholder

○ Basic inputting

✓ Correct

This will allow you to see missing data later without violating a schema.

5. When working with regression models, if the p-value of your model coefficient is <0.5 between the input feature and the predicted output, what does that mean? Select all that apply.

0 / 1 point

● There is a 5% probability of seeing the correlation by chance.

○ There is a 95% probability of seeing the correlation by chance.

○ There is less than 5% chance of seeing the correlation by chance.

○ There is more than 95% probability of seeing the correlation by chance.

⊗ Incorrect

This interpretation of p-value 0.5 is incorrect.

1. How are qualitative variables also known as?

1 / 1 point

Select all that apply.

☑ Discrete

Correct
This is one of the ways qualitative variables are also known.

☑ Categorical

Correct
This is one of the ways qualitative variables are also known.

□ Numerical

□ Continuous

2. Which type of supervised learning problem tends to output quantitative values?

1 / 1 point

○ Clustering

○ Classification

● Regression

Correct
This would be the algorithm used because you would predict a label based on numerical values.

3. In the process of explanatory data analysis, when we want to calculate the number of observations in the data set, which of the following will tell us if there are missing values in the dataset?

1 / 1 point

○ Standard deviation

● Count

○ Mean

Correct
Count gives us the number of observed values, indicating the size of the dataset and whether there are missing values.

4. In terms of correlations, what does a negative correlation of -1 means?

1 / 1 point

○ There is no association between the variables.

○ For each unit increase in one variable, the same increase is seen in the other..

● For each unit increase in one variable, the same decrease is seen in the other

Correct
This is what a negative correlation of -1 indicates.

5. Regarding visualization tools, which of the following can help you visualize quantiles and outliers?

1 / 1 point

○ Q-Q plots

● Box plots

○ t-SNE

○ Heat maps

Correct
Boxplot is a chart that is used to visualize how a given data (variable) is distributed using quartiles. It shows the minimum, maximum, median, first quartile and third quartile in the data set.

6. You have an AirBnB dataset where one categorical variable is room type.

1 / 1 point

There are three types of rooms: private room, entire home/apt, and shared room.

You must first encode each unique string into a number so that the machine learning model knows how to handle these room types.

How should you code that?

●
```
from pyspark.ml.feature import StringIndexer

uniqueTypesDF = airbnbDF.select("room_type").distinct()

indexer = StringIndexer(inputCol="room_type", outputCol="room_type_index")

indexerModel = indexer.fit(uniqueTypesDF)

indexedDF = indexerModel.transform(uniqueTypesDF)

display(indexedDF)
```

○
```
from pyspark.ml.feature import StringIndexer

uniqueTypesDF = airbnbDF.select("room_type").distinct()

indexer = StringIndexer(inputCol="room_type")

indexerModel = indexer.fit(uniqueTypesDF)

indexedDF = indexerModel.transform(uniqueTypesDF)

display(indexedDF)
```

○
```
from pyspark.ml.feature import Indexer

uniqueTypesDF = airbnbDF.select("room_type").distinct()

indexer = StringIndexer(inputCol="room_type", outputCol="room_type_index")

indexerModel = indexer.fit(uniqueTypesDF)

indexedDF = indexerModel.transform(uniqueTypesDF)

display(indexedDF)
```

○
```
from pyspark.ml.feature import StringIndexer

uniqueTypesDF = airbnbDF.select("room_type").distinct()
```

7. You have an AirBnB dataset where one categorical variable is room type.

1 / 1 point

There are three types of rooms: private room, entire home/apt, and shared room.

After you've encoded each unique string into a number, each room has a unique numerical value assigned.

Now you must one-hot encode each of those values to a location in an array, so that the machine learning algorithm can effect each category.

How should you code that?

○

```
from pyspark.ml.feature import OneHotEncoder

encoder = OneHotEncoder(inputCols=["room_type_index"], outputCols=["encoded_room_type"])

encoderModel = encoder.fit(indexedDF)

encodedDF = encoderModel_transform()

display(encodedDF)
```

●

```
from pyspark.ml.feature import OneHotEncoder

encoder = OneHotEncoder(inputCols=["room_type_index"], outputCols=["encoded_room_type"])

encoderModel = encoder.fit(indexedDF)

encodedDF = encoderModel.transform(indexedDF)

display(encodedDF)
```

○

```
from pyspark.ml.feature import OneHotEncoder

encoder = OneHotEncoder(inputCols=["room_type_index"], outputCols=["encoded_room_type"])

encoderModel = encoder.fit(indexedDF)

encodedDF = encoderModel(indexedDF)

display(encodedDF)
```

○

```
from pyspark.ml.feature import OneHotEncoder

encoder = OneHotEncoder(inputCols=["room_type_index"], outputCols=["encoded_room_type"])

encoderModel = encoder.fit(indexedDF)

encodedDF = encoderModel.fit (indexedDF)

display(encodedDF)
```

✓ Correct

This is the correct code. You need to change these values to a binary yes/no value if a listing is for a shared room, entire home, or private room.

Do this by training and fitting the OneHotEncoderEstimator, which only operates on numerical values (this is why we needed to use StringIndexer first).

Use MLflow to track experiments, log metrics, and compare runs

In this unit, you need to complete the exercises within a Databricks Notebook. To begin, you need to have access to an Azure Databricks workspace with an interactive cluster. If you do not have a workspace and/or the required cluster, follow the instructions below. Otherwise, you can skip to the bottom of the page to Install required libraries.

Unit Pre-requisites

Microsoft Azure Account: You will need a valid and active Azure account for the Azure labs. If you do not have one, you can sign up for a free trial

- If you are a Visual Studio Active Subscriber, you are entitled to Azure credits per month. You can refer to this link to find out more including how to activate and start using your monthly Azure credit.
- If you are not a Visual Studio Subscriber, you can sign up for the FREE Visual Studio Dev Essentials program to create Azure free account.

Create the required resources

To complete this lab, you will need to deploy an Azure Databricks workspace in your Azure subscription.

Deploy an Azure Databricks workspace

1. Click the following button to open the Azure Resource Manager template in the Azure portal. Deploy Databricks from the Azure Resource Manager Template
2. Provide the required values to create your Azure Databricks workspace:

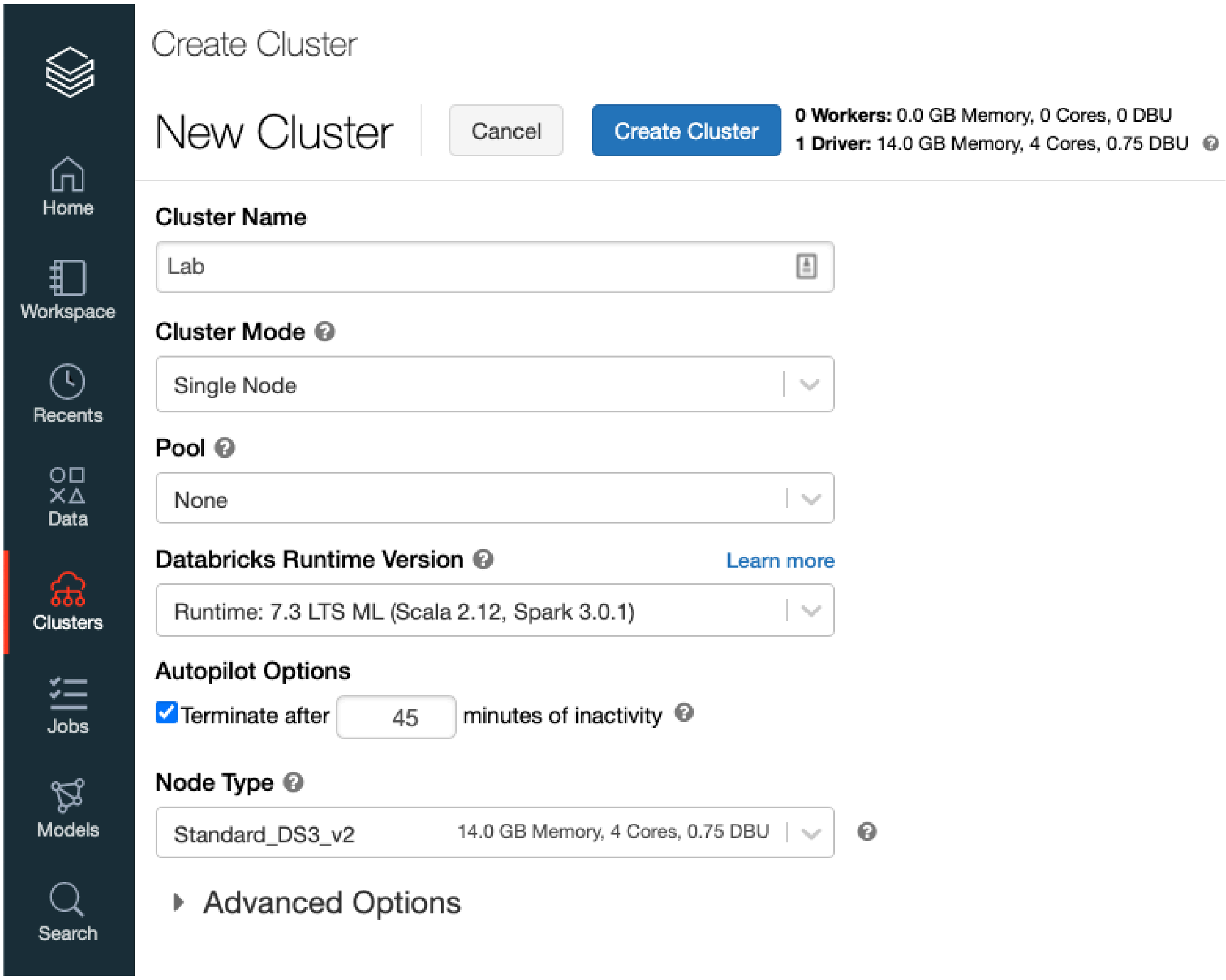
- **Subscription**: Choose the Azure Subscription in which to deploy the workspace.
- **Resource Group**: Leave at Create new and provide a name for the new resource group.
- **Location**: Select a location near you for deployment. For the list of regions supported by Azure Databricks, see Azure services available by region.
- **Workspace Name**: Provide a name for your workspace.
- **Pricing Tier**: Ensure premium is selected.

3. Accept the terms and conditions.
4. Select Purchase.
5. The workspace creation takes a few minutes. During workspace creation, the portal displays the Submitting deployment for Azure Databricks tile on the right side. You may need to scroll right on your dashboard to see the tile. There is also a progress bar displayed near the top of the screen. You can watch either area for progress.

Create a cluster

1. When your Azure Databricks workspace creation is complete, select the link to go to the resource.

2. Select **Launch Workspace** to open your Databricks workspace in a new tab.
3. In the left-hand menu of your Databricks workspace, select **Clusters**.
4. Select **Create Cluster** to add a new cluster.



Create Cluster

New Cluster

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU

Cluster Name: Lab

Cluster Mode: Single Node

Pool: None

Databricks Runtime Version: Runtime: 7.3 LTS ML (Scala 2.12, Spark 3.0.1)

Autopilot Options
☑ Terminate after 45 minutes of inactivity

Node Type: Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU

Advanced Options

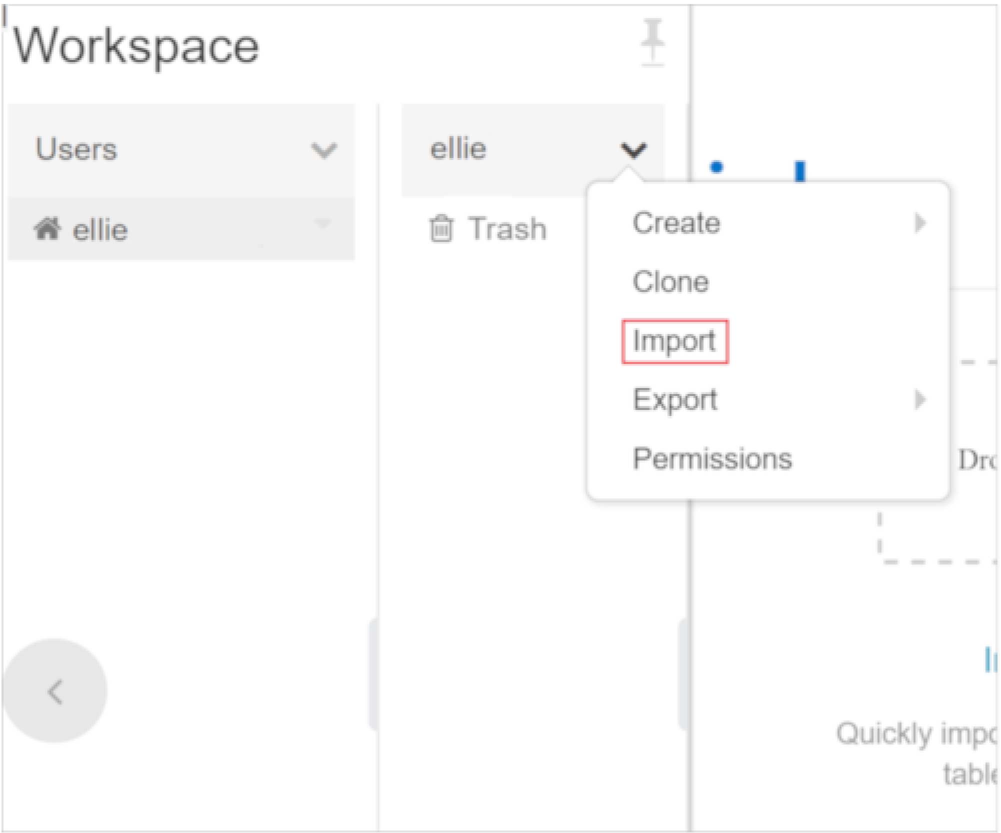
5. Enter a name for your cluster. Use your name or initials to easily differentiate your cluster from your coworkers.
6. Select the **Cluster Mode**: **Single Node**
7. Select the **Databricks RuntimeVersion**: **Runtime: 7.3 LTS ML (Scala 2.12, Spark 3.0.1)** (remember to select the **ML** version).
8. Under **Autopilot Options**, leave the box **checked** and in the text box enter `45`.
9. Select the **Node Type**: **Standard_DS3_v2**
10. Select **Create Cluster**.

Clone the Databricks archive

1. If you do not currently have your Azure Databricks workspace open: in the Azure portal, navigate to your deployed Azure Databricks workspace and select **Launch Workspace**.
2. In the left pane, select **Workspace** > **Users**, and select your username (the entry with the house icon).

In the pane that appears, select the arrow next to your name, and select **Import**.

3.



4. In the **Import Notebooks** dialog box, select the URL and paste in the following URL: **`https://github.com/MicrosoftDocs/mslearn_databricks/blob/main/mlflow/1.1.0/Labs.dbc`**

5. Select **Import**.

6. Select the **mlflow** folder that appears.

Complete the following notebook

Open the **1. MLflow** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

Within the notebook, you will:

- Use MLflow to track experiments, log metrics, and compare runs

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

Exercise: Work with MLflow to track experiment metrics, parameters, artifacts and modelss

In your Azure Databricks workspace, open the **mlflow** folder that you imported within your user folder.

Open the **2. Exercise MLflow** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

In this exercise, you will use the diabetes dataset in scikit-learn and predict the progression metric (a quantitative measure of disease progression after one year after) based on BMI, and blood pressure. You will use the scikit-learn ElasticNet linear regression model, where you vary the alpha and l1_ratio parameters for tuning. You will use MLflow to log metrics, parameters, artifacts, and model.

> **Note**
>
> You will find a corresponding notebook within the solutions subfolder. This contains completed cells for the exercise. Refer to the notebook if you get stuck or simply want to see the solution.

After you've completed the notebook, return to this screen, and continue to the next step.

1. What are the three core issues MLflow seeks to address?

1 / 1 point

Select all that apply.

☑ Code reproducing

Correct
This is one of the issues MLflow is addressing.

□ Keeping track of identity

☑ The standardization of model packaging and deployment

Correct
This is one of the issues MLflow is addressing, but there is no standard way to package and deploy models.

☑ Keeping track of experiments

Correct
This is one of the issues MLflow is addressing.

2. What is the MLflow Tracking tool?

1 / 1 point

○ An environment

● A logging API

○ A library

○ A class

Correct
MLflow Tracking is a logging API specific for machine learning and agnostic to libraries and environments that do the training.

3. MLflow Tracking is organized around the concept of runs, which are basically executions of data science code.

1 / 1 point

Runs are aggregated into which of the following?

○ Dataframe

○ Workflows

● Experiments

○ Datasets

Correct
Runs are aggregated into experiments where many runs can be a part of a given experiment and an MLflow server can host many experiments.

4. What information can be recorded for each run? Select all that apply.

0 / 1 point

☑ Variables

⊗ **This should not be selected**
Variables are not recorded.

☑ Artifacts

✓ **Correct**
Arbitrary output files in any format. This can include images, pickled models, and data files.

□ Source

☑ Parameters

✓ **Correct**
Key-value pairs of input parameters such as the number of trees in a random forest model.

☑ Metrics

✓ **Correct**
Evaluation metrics such as RMSE or Area Under the ROC Curve.

5. Which of the following objects can be used to query past runs programmatically?

1 / 1 point

● MlflowClient

○ MlflowQuery

○ MlflowTracker

○ MlflowFetcher

✓ **Correct**
The MlflowClient object is the pathway to querying past runs programmatically in order to use the data back in Python.

Describe model selection and hyperparameter tuning

In this unit, you need to complete the exercises within a Databricks Notebook. To begin, you need to have access to an Azure Databricks workspace with an interactive cluster. If you do not have a workspace and/or the required cluster, follow the instructions below. Otherwise, you can skip to the bottom of the page to Clone the Databricks archive.

Unit Pre-requisites

Microsoft Azure Account: You will need a valid and active Azure account for the Azure labs. If you do not have one, you can sign up for a free trial

- If you are a Visual Studio Active Subscriber, you are entitled to Azure credits per month. You can refer to this link to find out more including how to activate and start using your monthly Azure credit.
- If you are not a Visual Studio Subscriber, you can sign up for the FREE Visual Studio Dev Essentials program to create Azure free account.

Create the required resources

To complete this lab, you will need to deploy an Azure Databricks workspace in your Azure subscription.

Deploy an Azure Databricks workspace

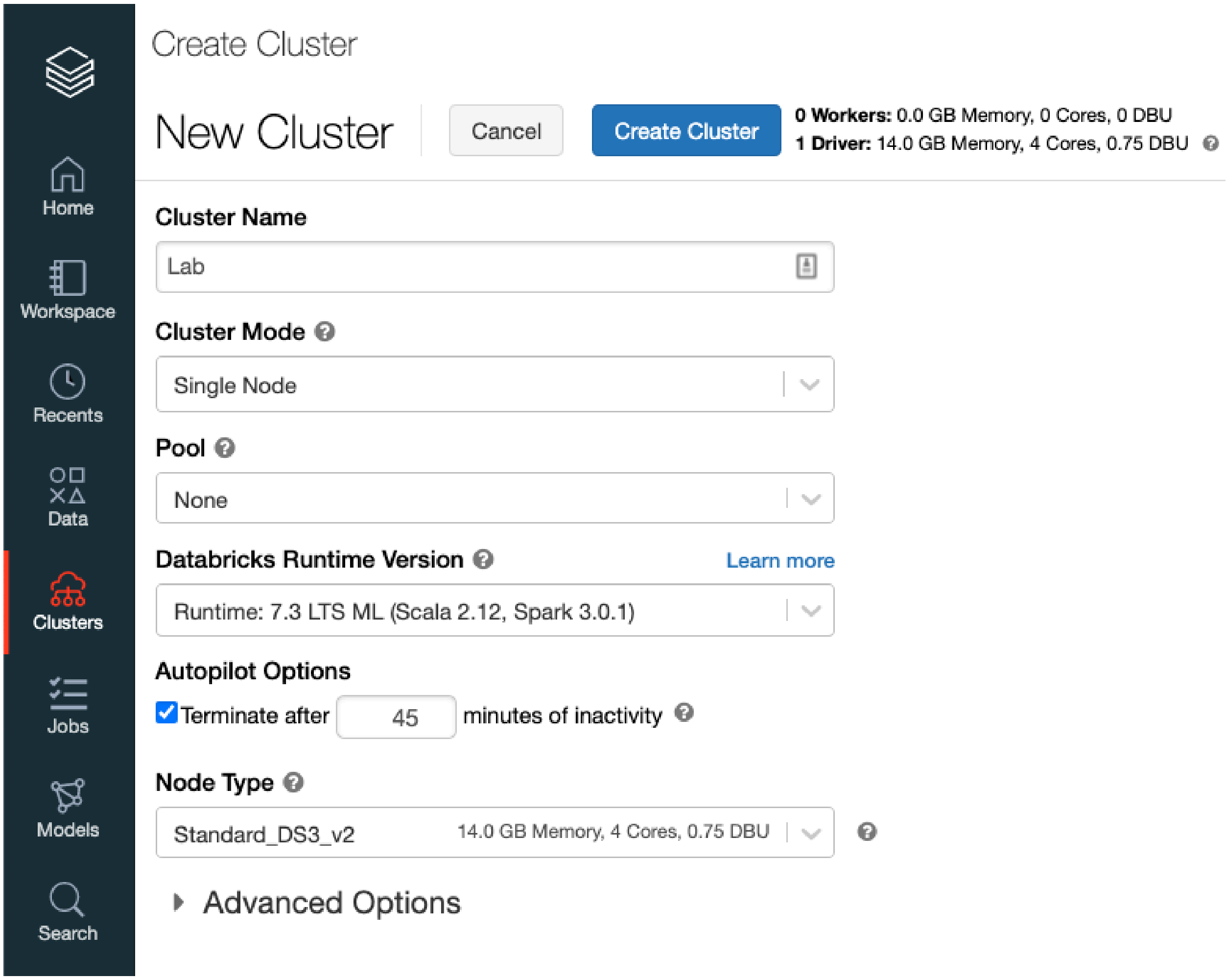
1. Click the following button to open the Azure Resource Manager template in the Azure portal. Deploy Databricks from the Azure Resource Manager Template
2. Provide the required values to create your Azure Databricks workspace:

- **Subscription**: Choose the Azure Subscription in which to deploy the workspace.
- **Resource Group**: Leave at Create new and provide a name for the new resource group.
- **Location**: Select a location near you for deployment. For the list of regions supported by Azure Databricks, see Azure services available by region.
- **Workspace Name**: Provide a name for your workspace.
- **Pricing Tier**: Ensure `premium` is selected.

3. Accept the terms and conditions.
4. Select Purchase.
5. The workspace creation takes a few minutes. During workspace creation, the portal displays the Submitting deployment for Azure Databricks tile on the right side. You may need to scroll right on your dashboard to see the tile. There is also a progress bar displayed near the top of the screen. You can watch either area for progress.

Create a cluster

1. When your Azure Databricks workspace creation is complete, select the link to go to the resource.
2. Select **Launch Workspace** to open your Databricks workspace in a new tab.
3. In the left-hand menu of your Databricks workspace, select **Clusters**.
4. Select **Create Cluster** to add a new cluster.



Create Cluster

New Cluster

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU

Cluster Name
Lab

Cluster Mode
Single Node

Pool
None

Databricks Runtime Version
Runtime: 7.3 LTS ML (Scala 2.12, Spark 3.0.1)

Autopilot Options
☑ Terminate after 45 minutes of inactivity

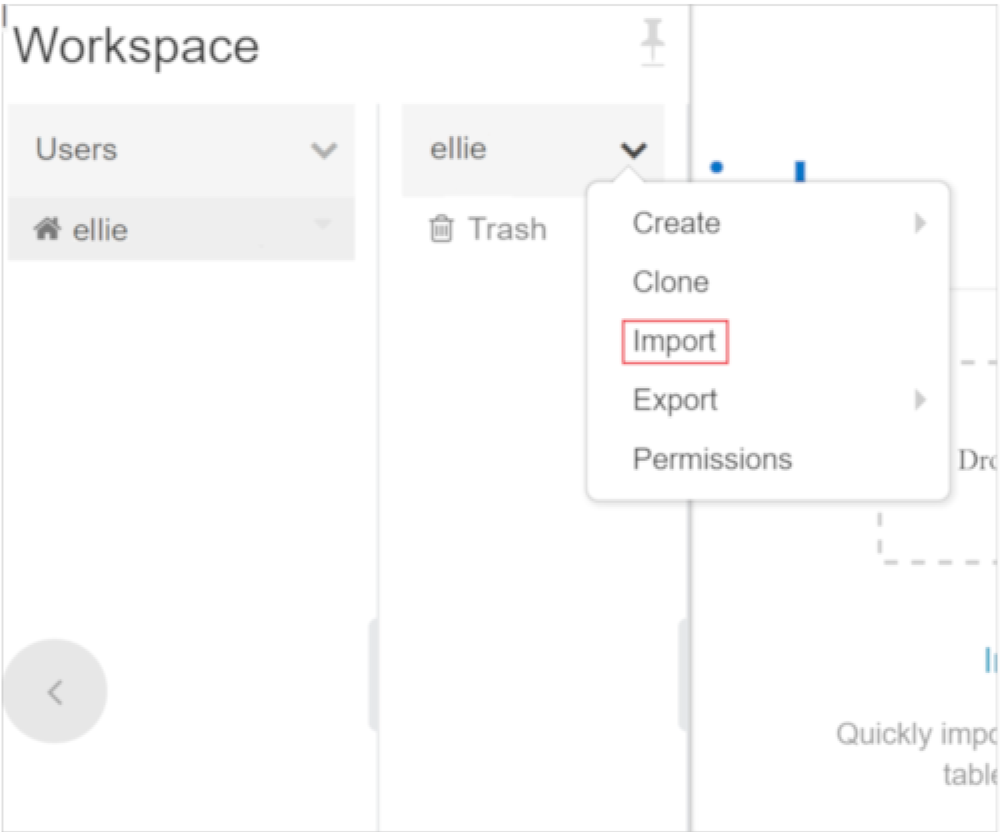
Node Type
Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU

5. Enter a name for your cluster. Use your name or initials to easily differentiate your cluster from your coworkers.
6. Select the **Cluster Mode**: **Single Node**
7. Select the **Databricks RuntimeVersion**: **Runtime: 7.3 LTS ML (Scala 2.12, Spark 3.0.1)** (remember to select the **ML** version).
8. Under **Autopilot Options**, leave the box **checked** and in the text box enter `45`.
9. Select the **Node Type**: **Standard_DS3_v2**
10. Select **Create Cluster**.

Clone the Databricks archive

1. If you do not currently have your Azure Databricks workspace open: in the Azure portal, navigate to your deployed Azure Databricks workspace and select **Launch Workspace**.

2. In the left pane, select **Workspace** > **Users**, and select your username (the entry with the house icon).
3. In the pane that appears, select the arrow next to your name, and select **Import**.



4. In the **Import Notebooks** dialog box, select the URL and paste in the following URL:
`https://github.com/MicrosoftDocs/mslearn_databricks/blob/main/hyperparameter/1.1.0/Labs.dbc`
5. Select **Import**.
6. Select the **hyperparameter** folder that appears.

Complete the following notebook

Open the **1. Model Selection** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.
Within the notebook, you will:

- Define hyperparameters and motivate their role in machine learning
- Tune hyperparameters using grid search
- Validate model performance using cross-validation
- Save a trained model and its predictions

After you've completed the notebook, return to this screen, and continue to the next step.
Go to next item

Exercise: Select optimal model by tuning hyperparameters

In your Azure Databricks workspace, open the **hyperparameter** folder that you imported within your user folder.

Open the **2. Exercise Model Selection** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

In this exercise, you will use grid search and cross-validation to tune the hyperparameters from a logistic regression model.

> **Note**
>
> You will find a corresponding notebook within the solutions subfolder. This contains completed cells for the exercise. Refer to the notebook if you get stuck or simply want to see the solution.

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

1. What will happen to a model that has been trained and evaluated on the same data?

1 / 1 point

○ Underfitting

● Overfitting

○ Well generalized

✓ Correct
Overfitting occurs when the model performs well on data it has already seen but fails to predict anything useful on data it has not already seen. This is the case here.

2. True or false?

1 / 1 point

A machine learning algorithm can learn hyperparameters from the data itself.

○ True

● False

✓ Correct
A hyperparameter is a parameter used in a machine learning algorithm that is set before the learning process begins. In other words, a machine learning algorithm cannot learn hyperparameters from the data itself.

3. Which of the following best describes the process of Hyperparameter tuning?

0 / 1 point

○ The process of dropping the hyperparameters that do not perform well on the loss function of the model.

● The process of modifying the hyperparameter until we get the best result on the loss function of the model.

○ The process of choosing the hyperparameter that performs the best on the loss function of the model.

⊗ Incorrect
This is close, but not the actual process.

4. When training different models with different hyperparameters and evaluating their performance, there is a risk of overfitting by choosing the hyperparameter that happens to perform best on the data found in the dataset.

1 / 1 point

Which cross-validation technique would be the best fit for solving this problem?

● K-fold cross-validation

○ Repeated random subsampling validation

○ Time Series cross-validation

○ Holdout cross-validation

✓ Correct
This would be the best choice because in k-fold cross-validation the original dataset is equally partitioned into k subparts or folds. Out of the k-folds or groups, for each iteration, one group is selected as validation data, and the remaining (k-1) groups are selected as training data.

5. Which of the following hyperparameter optimization technique is the process of exhaustively trying every combination of hyperparameters?

1 / 1 point

● Grid Search

○ Random Search

○ Bayesian Search

✓ Correct

Grid Search is a method wherein we try all possible combination of the set of Hyperparameters. Each combination of the Hyperparameters represent a machine learning model. Hence, N combinations represent N machine learning models. Through Grid search, we identify the model which shows the best performance.

1. You can query previous runs programmatically by using the MlflowClient object as the pathway.

1 / 1 point

How would you code that in Python?

○ from mlflow.tracking import MlflowClient

client = MlflowClient()

list.client_experiments()

○ from mlflow.pipelines import MlflowClient

client = MlflowClient()

client.list_experiments()

○ from mlflow.pipelines import MlflowClient

client = MlflowClient()

list.experiments()

● from mlflow.tracking import MlflowClient

client = MlflowClient()

client.list_experiments()

✓ **Correct**
This is the correct code syntax for this job.

2. You can also use the search_runs method to find all runs for a given experiment.

1 / 1 point

How would you code that in Python?

○ experiment_id = run.experiment_id

runs_df = mlflow.search_runs(experiment_id)

display(runs_df)

● experiment_id = run.info.experiment_id

runs_df = mlflow.search_runs(experiment_id)

display(runs_df)

○ experiment = run.experiment_id

runs_df = mlflow.search_runs(experiment_id)

display(runs_df)

○ experiment_id = info.experiment_id

runs_df = mlflow.search_runs(experiment_id)

display(runs_df)

✓ **Correct**
This is the correct code syntax.

3. You need to retrieve the last run from the list of experiments.

1 / 1 point

How would you code that in Python?

○ runs = client.search_runs(experiment_id, order_by=["attributes.start_time asce"], max_results=1)

runs[0].data.metrics

● runs = client.search_runs(experiment_id, order_by=["attributes.start_time desc"], max_results=1)

runs[0].data.metrics

○ runs = client.search_runs(experiment_id, order_by=["attributes.start_time desc"], max_results=3)

runs[0].data.metrics

○ runs = client.search_runs(experiment_id, order_by=["attributes.start_time"], max_results=1)

runs[0].data.metrics

✓ Correct
This is the correct code syntax.

4. Knowing that each algorithm has different hyperparameter available for tuning, which method can you use to explore the hyperparameters on a model?

1 / 1 point

○ getParams()

● explainParams()

○ showParams()

○ exploreParams()

✓ Correct
You can explore these hyperparameters by using the .explainParams() method on a model.

5. Which method from the PySpark class can you use to string together all the different possible hyperparameters you want to test?

1 / 1 point

○ ParamBuilder()

○ ParamGridSearch()

● ParamGridBuilder()

○ ParamSearch()

✓ Correct
ParamGridBuilder() allows you to string together all of the different possible hyperparameters you would like to test. In this case, you can test the maximum number of iterations, whether you want to use an intercept with the y axis, and whether you want to standardize our features.

6. Which of the following belong to the exhaustive type of cross-validation techniques?

0 / 1 point

☑ K-fold cross-validation

ⓧ This should not be selected

Try going back and reviewing Describe model selection and hyperparameter tuning.

☑ Leave-p-out cross-validation

✓ Correct

Leave-p-out cross-validation (LpO CV) is an exhaustive type of cross-validation technique. It involves using p observations as the validation set and the remaining observations as the training set. This is repeated on all ways to cut the original sample on a validation set of p observations and a training set.

□ Holdout cross-validation

☑ Leave-one-out cross-validation

✓ Correct

Leave-one-out cross-validation (LOOCV) is a particular case of leave-p-out cross-validation with p = 1, which makes it an exhaustive type of cross-validation.

7. In which of the following non-exhaustive cross validation techniques do you randomly assign data points to the training set and the test set?

1 / 1 point

○ K-fold cross-validation

● Holdout cross-validation

○ Repeated random sub-sampling validation

✓ Correct

In the holdout method, you randomly assign data points to two sets d0 and d1, usually called the training set and the test set, respectively. The size of each of the sets is arbitrary although typically the test set is smaller than the training set. You then train (build a model) on d0 and test (evaluate its performance) on d1.

Use Horovod to train a deep learning model

In this unit, you need to complete the exercises within a Databricks Notebook. To begin, you need to have access to an Azure Databricks workspace with an interactive cluster. If you do not have a workspace and/or the required cluster, follow the instructions below. Otherwise, you can skip to the bottom of the page to Install required libraries.

Unit Pre-requisites

Microsoft Azure Account: You will need a valid and active Azure account for the Azure labs. If you do not have one, you can sign up for a free trial

- If you are a Visual Studio Active Subscriber, you are entitled to Azure credits per month. You can refer to this link to find out more including how to activate and start using your monthly Azure credit.
- If you are not a Visual Studio Subscriber, you can sign up for the FREE Visual Studio Dev Essentials program to create Azure free account.

Create the required resources

To complete this lab, you will need to deploy an Azure Databricks workspace in your Azure subscription.

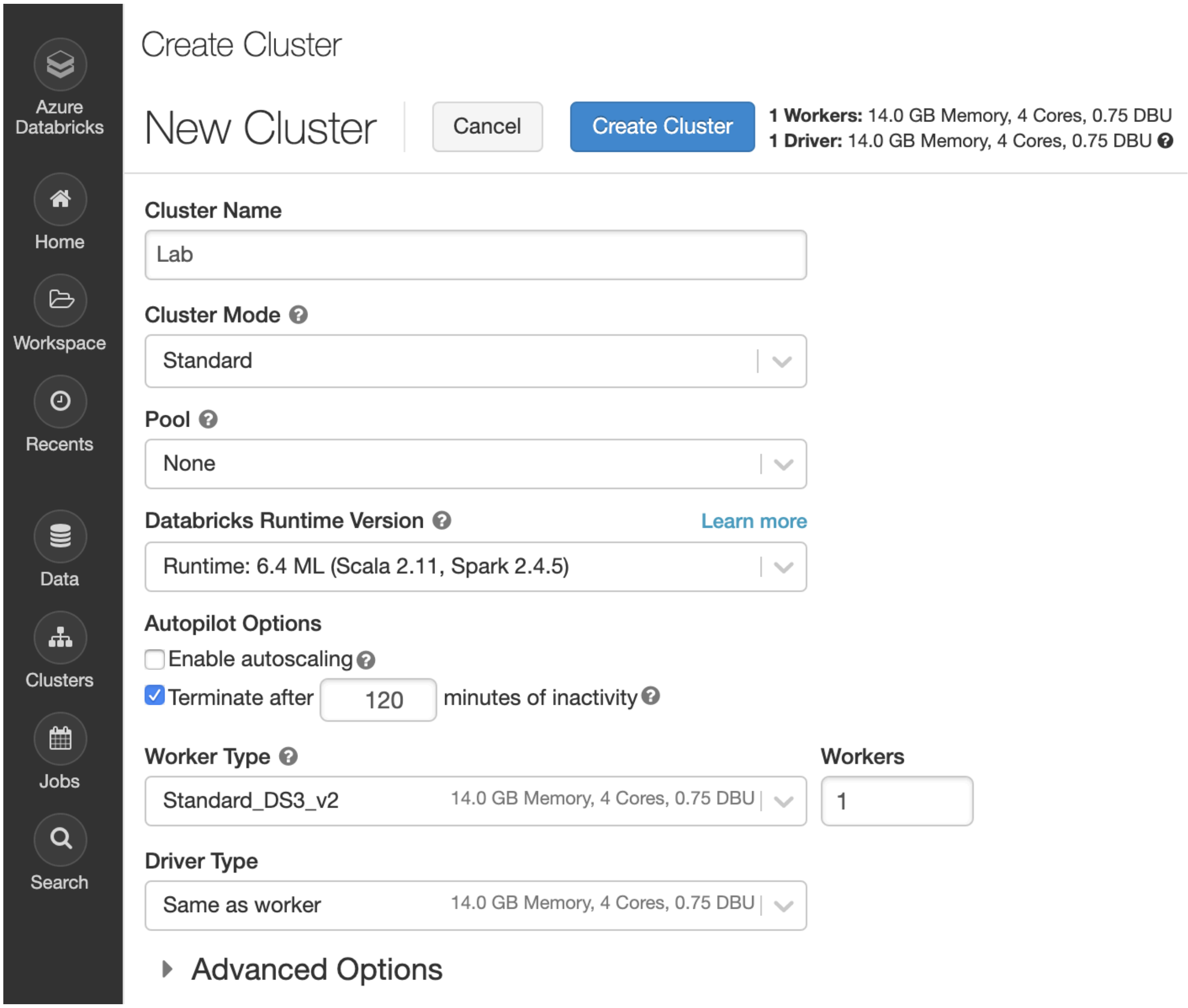
Deploy an Azure Databricks workspace

1. Click the following button to open the Azure Resource Manager template in the Azure portal.
Deploy Databricks from the Azure Resource Manager Template
2. Provide the required values to create your Azure Databricks workspace:
   - **Subscription**: Choose the Azure Subscription in which to deploy the workspace.
   - **Resource Group**: Leave at Create new and provide a name for the new resource group.
   - **Location**: Select a location near you for deployment. For the list of regions supported by Azure Databricks, see Azure services available by region.
   - **Workspace Name**: Provide a name for your workspace.
   - **Pricing Tier**: Ensure `premium` is selected.

3. Accept the terms and conditions.
4. Select Purchase.
5. The workspace creation takes a few minutes. During workspace creation, the portal displays the Submitting deployment for Azure Databricks tile on the right side. You may need to scroll right on your dashboard to see the tile. There is also a progress bar displayed near the top of the screen. You can watch either area for progress.

Create a cluster

1. When your Azure Databricks workspace creation is complete, select the link to go to the resource.
2. Select **Launch Workspace** to open your Databricks workspace in a new tab.
3. In the left-hand menu of your Databricks workspace, select **Clusters**.
4. Select **Create Cluster** to add a new cluster.



Create Cluster

New Cluster

1 Workers: 14.0 GB Memory, 4 Cores, 0.75 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU

Cluster Name: Lab

Cluster Mode: Standard

Pool: None

Databricks Runtime Version: Runtime: 6.4 ML (Scala 2.11, Spark 2.4.5)

Autopilot Options

☐ Enable autoscaling

☑ Terminate after 120 minutes of inactivity

Worker Type: Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU

Workers: 1

Driver Type: Same as worker 14.0 GB Memory, 4 Cores, 0.75 DBU

The create cluster page.

5. Enter a name for your cluster. Use your name or initials to easily differentiate your cluster from your coworkers.

6. Select the **Databricks RuntimeVersion**: **Runtime: 6.4 ML (Scala 2.11, Spark 2.4.5)** (remember to select the **ML** version).

7. Select the values for the cluster configuration.

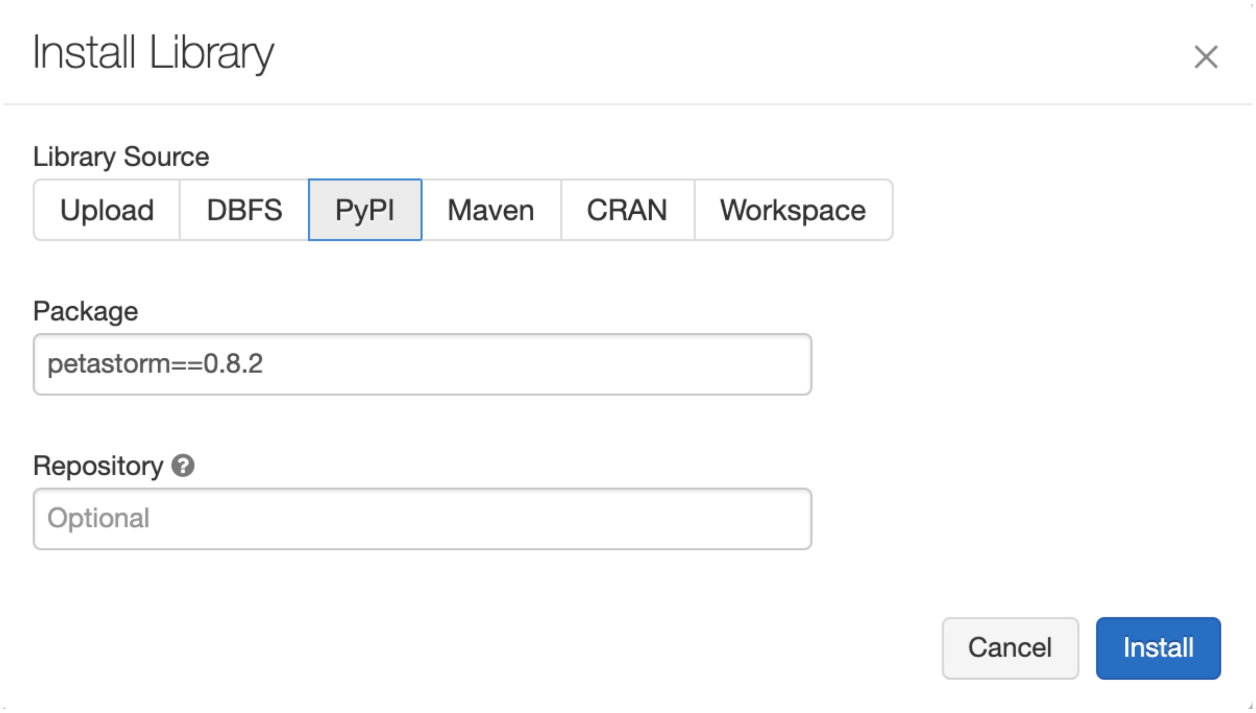
- **Enable autoscaling**: **Uncheck** this option.
- **Auto Termination**: Leave **checked** and in the text box enter `120`.
- **Worker Type**: **Standard_DS3_v2**
- **Workers**: `1`
- **Driver Type**: **Same as worker**

8. Select **Create Cluster**.

Install required libraries

1. From the cluster configuration page, select the **Libraries** link and then select **Install New**.

2. In the `Library Source`, select **PyPi** and in the `Package` text box type `petastorm==0.8.2` and select **Install**.


Install Library

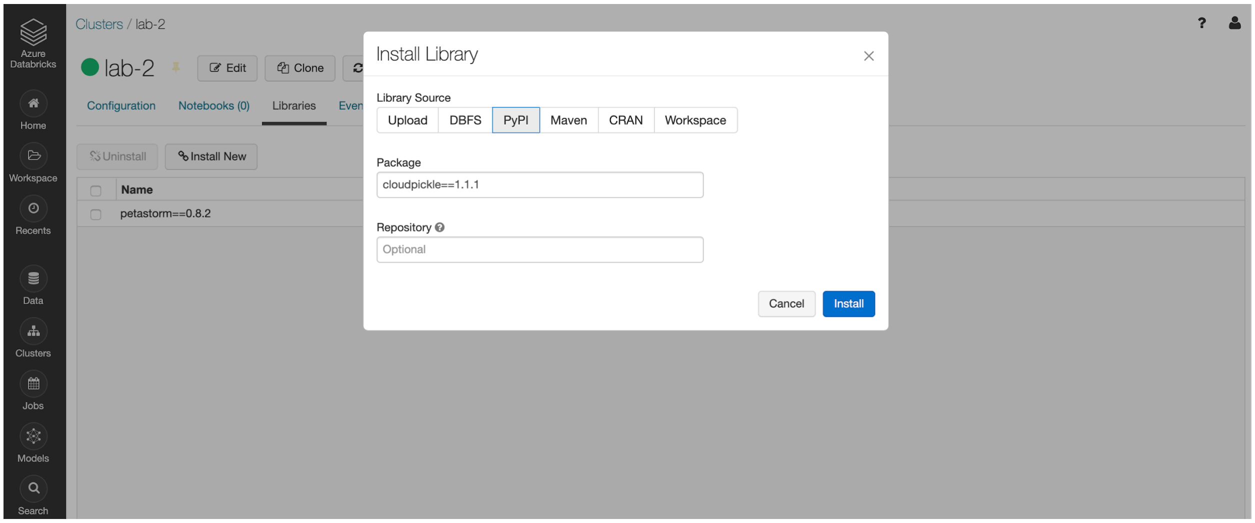
Library Source: PyPI

Package: petastorm==0.8.2

Repository: Optional

Install petastorm.

3. In the `Library Source`, select **PyPi** and in the `Package` text box type `cloudpickle==1.1.1` and select **Install**.


Clusters / lab-2

lab-2

	Name
□	petastorm==0.8.2

Install Library

Library Source: PyPI

Package: cloudpickle==1.1.1

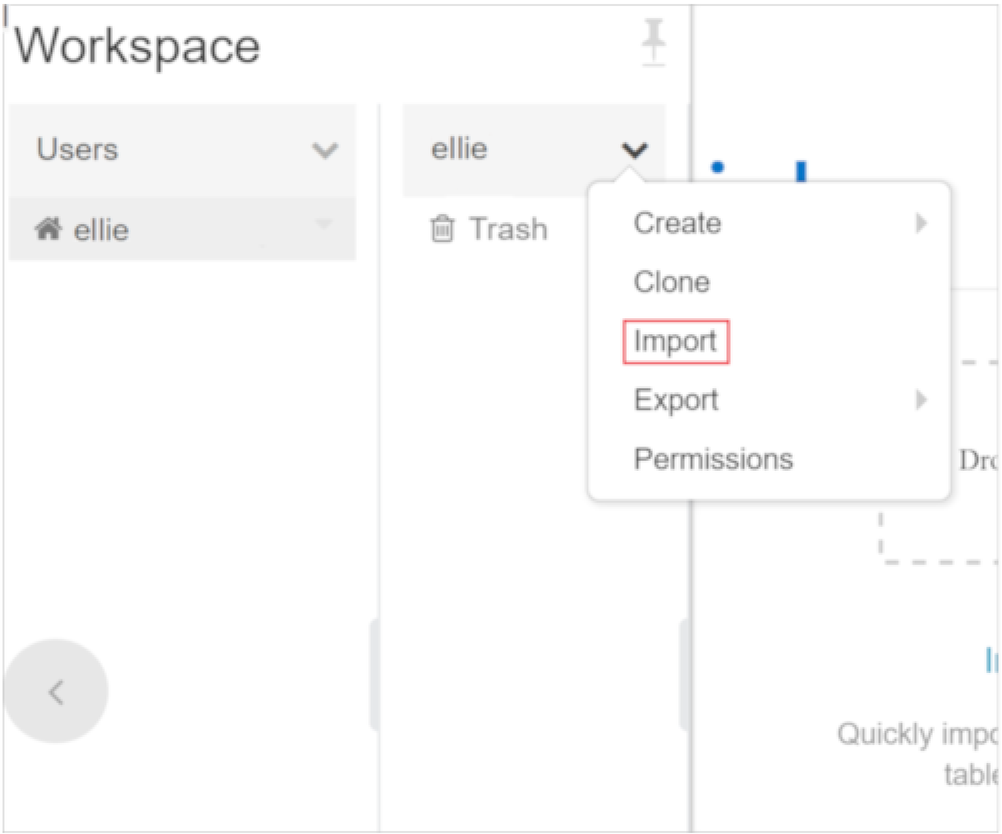
Repository: Optional

Install cloudpickle.

Clone the Databricks archive

1. If you do not currently have your Azure Databricks workspace open: in the Azure portal, navigate to your deployed Azure Databricks workspace and select **Launch Workspace**.

2. In the left pane, select **Workspace** > **Users**, and select your username (the entry with the house icon).
3. In the pane that appears, select the arrow next to your name, and select **Import**.



The menu option to import the archive.
4. In the **Import Notebooks** dialog box, select the URL and paste in the following URL:
`https://github.com/solliancenet/microsoft-learning-paths-databricks-notebooks/blob/master/data-science/11-Deep-Learning.dbc?raw=true`
5. Select **Import**.
6. Select the **11-Deep-Learning** folder that appears.

Complete the following notebook

Open the **1. Horovod** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.
Within the notebook, you will:

- Use Horovod to train a distributed neural network

After you've completed the notebook, return to this screen, and continue to the next step.

Use Petastorm to read in Apache Parquet format with Horovod for distributed model training

In your Azure Databricks workspace, open the **11-Deep-Learning** folder that you imported within your user folder.

Open the **2. Horovod Petastorm** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

Within the notebook, you will:

- Use Horovod to train a distributed neural network using Parquet files + Petastorm

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

Completed

Exercise: Work with Horovod and Petastorm for training a deep learning model

In your Azure Databricks workspace, open the **11-Deep-Learning** folder that you imported within your user folder.

Open the **3. Exercise Horovod Petastorm** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.

In this exercise you will build a model on the Boston housing dataset and distribute the deep learning training process using both HorovodRunner and Petastorm.

Note

You will find a corresponding notebook within the `Solutions` subfolder. This contains completed cells for the exercise. Refer to the notebook if you get stuck or simply want to see the solution.

After you've completed the notebook, return to this screen, and continue to the next step.

Go to next item

1. What is HorovodRunner?

1 / 1 point

○ A logging API

● A general API

○ A framework

○ A Python class

Correct
HorovodRunner is a general API for running distributed DL workloads on Databricks using Uber's Horovod framework.

2. What does HorovodRunner use to take a Python method that contains deep learning training code?

1 / 1 point

○ URL

○ Paths

○ URI

● Hooks

Correct
HorovodRunner takes a Python method that contains deep learning training code with Horovod hooks. HorovodRunner pickles the method on the driver and distributes it to Spark workers.

3. Which are two methods supported by the HorovodRunner API?

1 / 1 point

☑ run(self, main, **kwargs)

Correct
This method is supported by the HorovodRunner API and it runs a Horovod training job invoking main(**kwargs). The main function and the keyword arguments are serialized using cloudpickle and distributed to cluster workers.

☑ init(self, np)

Correct
This method is supported by the HorovodRunner API and it creates an instance of HorovodRunner.

□ run(self, main, np, **kwargs)

□ init(self, main)

4. Regarding the MPI concepts on which the Horovod core principles are based on, which MPI concept would be the unique process ID?

1 / 1 point

○ Local Rank

○ Size

● Rank

○ Density

✓ Correct
Rank would be the unique process ID.

5. True or false?

1 / 1 point

TensorFlow objects cannot be found or pickled using the HorovodRunner API.

○ True

● False

✓ Correct
A common error is that TensorFlow objects cannot be found or pickled. This happens when the library import statements are not distributed to other executors. To avoid this issue, include all import statements (for example, import tensorflow as tf) both at the top of the Horovod training method and inside any other user-defined functions called in the Horovod training method.

Use Azure Machine Learning to deploy serving models

In this unit, you need to complete the exercises within a Databricks Notebook. To begin, you need to have access to an Azure Databricks workspace with an interactive cluster. If you do not have a workspace and/or the required cluster, follow the instructions below. Otherwise, you can skip to the bottom of the page to **Install required libraries**.

Unit prerequisites

Microsoft Azure Account: You will need a valid and active Azure account for the Azure labs. If you do not have one, you can sign up for a free trial

- If you are a Visual Studio Active Subscriber, you are entitled to Azure credits per month. You can refer to this link to find out more including how to activate and start using your monthly Azure credit.
- If you are not a Visual Studio Subscriber, you can sign up for the FREE Visual Studio Dev Essentials program to create Azure free account.

Create the required resources

To complete this lab, you will need to deploy an Azure Databricks workspace in your Azure subscription.

Deploy an Azure Databricks workspace

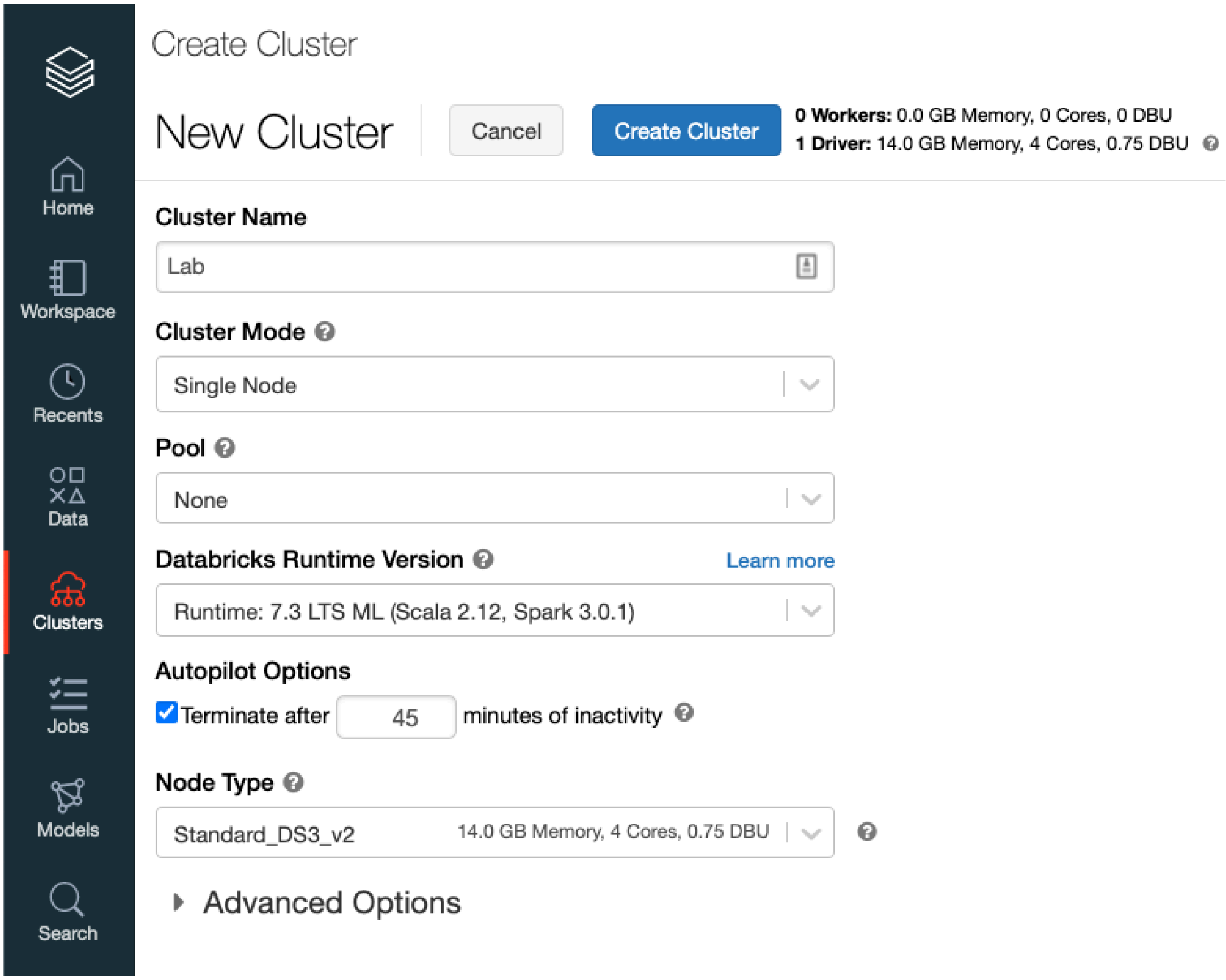
1. Click the following button to open the Azure Resource Manager template in the Azure portal.
Deploy Databricks from the Azure Resource Manager Template
2. Provide the required values to create your Azure Databricks workspace:
   - **Subscription**: Choose the Azure Subscription in which to deploy the workspace.
   - **Resource Group**: Leave at Create new and provide a name for the new resource group.
   - **Location**: Select a location near you for deployment. For the list of regions supported by Azure Databricks, see Azure services available by region.
   - **Workspace Name**: Provide a name for your workspace.
   - **Pricing Tier**: Ensure `premium` is selected.

3. Accept the terms and conditions.
4. Select Purchase.
5. The workspace creation takes a few minutes. During workspace creation, the portal displays the Submitting deployment for Azure Databricks tile on the right side. You may need to scroll right on your dashboard to see the tile. There is also a progress bar displayed near the top of the screen. You can watch either area for progress.

Create a cluster

1. When your Azure Databricks workspace creation is complete, select the link to go to the resource.
2. Select **Launch Workspace** to open your Databricks workspace in a new tab.

3. In the left-hand menu of your Databricks workspace, select **Clusters**.
4. Select **Create Cluster** to add a new cluster.



Create Cluster

New Cluster

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU

Cluster Name
Lab

Cluster Mode
Single Node

Pool
None

Databricks Runtime Version
Runtime: 7.3 LTS ML (Scala 2.12, Spark 3.0.1)

Autopilot Options
☑ Terminate after 45 minutes of inactivity

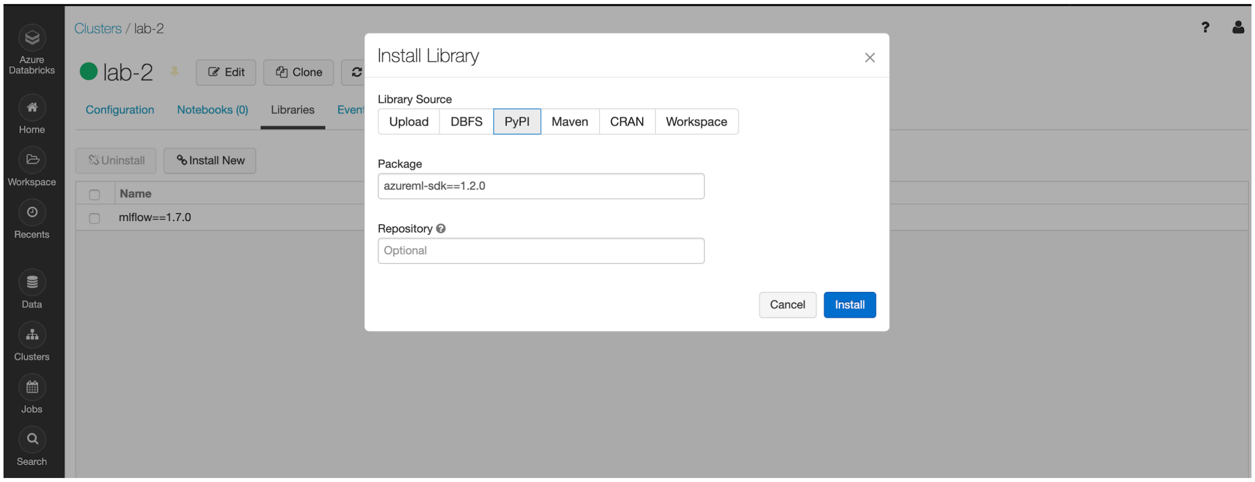
Node Type
Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU

The create cluster page.

5. Enter a name for your cluster. Use your name or initials to easily differentiate your cluster from your coworkers.
6. Select the **Cluster Mode**: **Single Node**
7. Select the **Databricks RuntimeVersion**: **Runtime: 7.3 LTS ML (Scala 2.12, Spark 3.0.1)** (remember to select the **ML** version).
8. Under **Autopilot Options**, leave the box **checked** and in the text box enter `45`.
9. Select the **Node Type**: **Standard_DS3_v2**
10. Select **Create Cluster**.

Install required libraries

1. From the cluster configuration page, select the **Libraries** link and then select **Install New**.
2. In the `Library Source`, select **PyPi** and in the `Package` text box type `azureml-sdk==1.2.0` and select **Install**.



mlflow==1.7.0

Install Library

Library Source: Upload | DBFS | PyPI | Maven | CRAN | Workspace

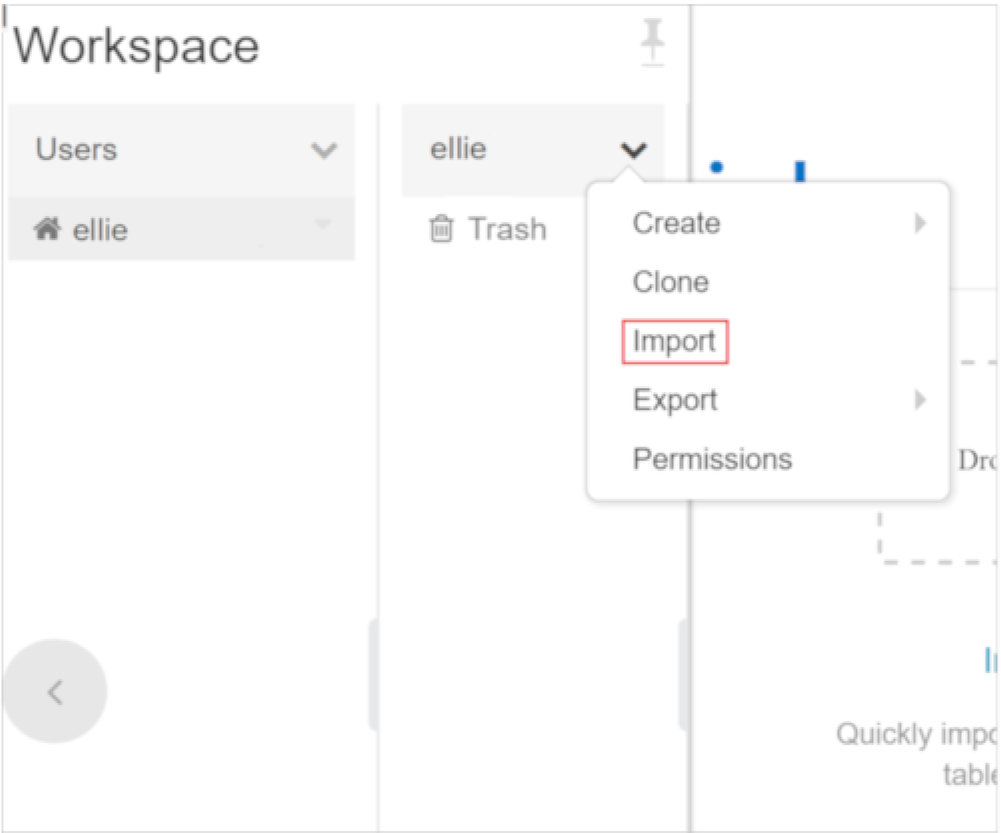
Package: azureml-sdk==1.2.0

Repository: Optional

Install required library.

Clone the Databricks archive

1. If you do not currently have your Azure Databricks workspace open: in the Azure portal, navigate to your deployed Azure Databricks workspace and select **Launch Workspace**.
2. In the left pane, select **Workspace** > **Users**, and select your username (the entry with the house icon).
3. In the pane that appears, select the arrow next to your name, and select **Import**.



The menu option to import the archive.

4. In the **Import Notebooks** dialog box, select the URL and paste in the following URL:
`https://github.com/MicrosoftDocs/mslearn_databricks/blob/main/azure-ml/1.1.0/Labs.dbc`
5. Select **Import**.
6. Select the **azure-ml** folder that appears.

Complete the following notebook

Open the **1. Serving Models with Microsoft Azure ML** notebook. Make sure you attach your cluster to the notebook before following the instructions and running the cells within.
Within the notebook, you will:

- Create or load an Azure ML Workspace
- Build an Azure Container Image for model deployment
- Deploy the model to "dev" using ACI
- Query the deployed model in "dev"
- Deploy the model to production using AKS
- Query the deployed model in production
- Update the production deployment
- Clean up the deployments

After you've completed the notebook, return to this screen, and continue to the next step.

1. To deploy a model to Azure ML, you must create or obtain an Azure ML Workspace.

   1 / 1 point

   You can do that programmatically by using a function.

   Which of the following functions can you use to create the workspace?

   ○ azureml.core.environment.create()

   ● azureml.core.workspace.create()

   ○ azureml.core.model.create()

   ○ azureml.core.dataset.workspace()

   ✓ Correct
   The azureml.core.Workspace.create() function will load a workspace of a specified name or create one if it does not already exist.

2. You want to use Azure ML to train a Diabetes Model and build a container image for the trained model.

   1 / 1 point

   You will use the scikit-learn ElasticNet linear regression model.

   You need to load the diabetes datasets. How should you code that?

   ○ datasets = diabetes.load()

   X = diabetes.data

   y = diabetes.target

   ○ diabetes.tf = datasets.load_diabetes()

   X = diabetes.data

   y = diabetes.target

   ● diabetes = datasets.load_diabetes()

   X = diabetes.data

   y = diabetes.target

   ○ diabetes = datasets_load_diabetes()

   X = diabetes.data

   y = diabetes.target

   ✓ Correct
   This is the correct code for the task.

3. When working with Azure ML, you can use MLflow to build a container image for the trained model.

0 / 1 point

Which MLflow function can you use for that task?

○ mlflow.build_image()

● azureml.mlflow.build_image()

○ mlflow.azureml.build.image()

○ mlflow.azureml.build_image()

⊗ Incorrect
This function is incorrect.

4. Which kind of HTTP request can you send to the AKS webservice's scoring endpoint to evaluate the sample data?

1 / 1 point

○ PATCH

● POST

○ PUT

○ GET

✓ Correct
POST is used to send data to a server to create/update a resource. Query the AKS webservice's scoring endpoint by sending an HTTP POST request that includes the input vector.

5. Which Azure ML function can you use to replace the deployment's existing model image with the new model image?

1 / 1 point

○ azureml.core.webservice.AksWebservice.deploy_configuration()

○ azureml.core.webservice.AksWebservice.serialize()

○ azureml.core.webservice.AksWebservice.add_properties()

● azureml.core.webservice.AksWebservice.update()

✓ Correct
This function will update the webservice with the provided properties, which you can use to replace the current model image with the new model image.

1. When developing a distributed training program using HorovodRunner you would generally follow these steps:

1 / 1 point

1. Create a HorovodRunner instance initialized with the number of nodes.

2. Define a Horovod training method according to the methods described in Horovod usage, making sure to add any import statements inside the method.

3. Pass the training method to the HorovodRunner instance.

How would you code that in Python?

○ hr = HorovodRunner(tf)

def train():

 import tensorflow as np

 hvd.init(2)

hr.run(train)

● hr = HorovodRunner(np=2)

def train():

 import tensorflow as tf

 hvd.init()

hr.run(train)

○ hr = HorovodRunner(np)

def train():

 import tensorflow as tf

 hvd.init()

hr.run(train)

○ hr = HorovodRunner()

def train():

 import tensorflow as tf

 hvd.init(np)

hr.run(train)

✓ **Correct**
This would be the correct code syntax.

2. You're using Horovod to train a distributed neural network using Parquet files and Petastorm.

1 / 1 point

You have a dataset of housing prices in California named cal_housing.

After loading the data, you want to create a Spark DataFrame from the Pandas DataFrame so that you can concatenate the features and labels of the model.

How would you code that in Python?

● data = pd.concat([pd.DataFrame(X_train, columns=cal_housing.feature_names), pd.DataFrame(y_train, columns=["label"])], axis=1)

trainDF = spark.createDataFrame(data)

display(trainDF)

○ data = pd.concat([pd.DataFrame(X_train, columns=cal_housing.feature_names), pd.DataFrame(y_train, columns=["label"])])

trainDF = spark.createDataFrame(data)

display(trainDF)

○ data = pd.concat([pd.DataFrame(X_train, columns=cal_housing.feature_names), pd.DataFrame(y_train, columns=["label"])], axis=1)

trainDF = spark.createDataFrame()

display(trainDF)

○ data = pd.concat([pd.DataFrame(X_train, columns=cal_housing.feature_names), pd.DataFrame(y_train, columns=["label"])], axis=1)

trainDF = spark.DataFrame(data)

display(trainDF)

✓ **Correct**
This is the correct code for the job.

3. You're using Horovod to train a distributed neural network using Parquet files and Petastorm.

1 / 1 point

You have a dataset of housing prices in California named cal_housing.

After loading the data, you created a Spark DataFrame from the Pandas DataFrame so that you can concatenate the features and labels of the model.

Now you need to create Dense Vectors for the features.

How would you code that in Python?

○

```
from pyspark.ml.feature import VectorAssembler



vecAssembler = VectorAssembler(inputCols=cal_housing.feature_names, outputCol="features")

vecTrainDF = vecAssembler.transform(trainDF).hook("features", "label")

display(vecTrainDF)
```

○

```
from pyspark.ml.feature import VectorAssembler

vecAssembler = VectorAssembler(inputCols=cal_housing.feature_names, outputCol="features")

vecTrainDF = vecAssembler.transform(trainDF).call("features", "label")

display(vecTrainDF)
```

●

```
from pyspark.ml.feature import VectorAssembler

vecAssembler = VectorAssembler(inputCols=cal_housing.feature_names, outputCol="features")

vecTrainDF = vecAssembler.transform(trainDF).select("features", "label")

display(vecTrainDF)
```

○

```
from pyspark.ml.feature import VectorAssembler

vecAssembler = VectorAssembler(inputCols=cal_housing.feature_names, outputCol="labels ")

vecTrainDF = vecAssembler.transform(trainDF).select("features", "label")

display(vecTrainDF)
```

✓ **Correct**
This is the correct code for the task.

4. True or false?

0 / 1 point

Petastorm requires a Vector as an input, not an Array.

● True

○ False

⊗ **Incorrect**
Try going back and reviewing Use Petastorm to read datasets in Apache Parquet format with Horovod for distributed model training.

5. You're working with Azure Machine Learning and you want to train a Diabetes Model and build a container image for the trained model.

1 / 1 point

You will use the scikit-learn ElasticNet linear regression model.

You want to deploy the model to production using Azure Kubernetes Service (AKS).

You don't have an active AKS cluster, so you need to create one using the Azure ML SDK.

You'll be using the default configuration.

How would you code that?

○ aks_target = ComputeTarget.create(workspace = workspace,

name = aks_cluster_name,)

○ aks_target = ComputeTarget.workspace = workspace

(name = aks_cluster_name,

provisioning_configuration = prov_config)

○ aks_target = ComputeTarget.deploy(workspace = workspace,

name = aks_cluster_name,

provisioning_configuration = prov_config)

● aks_target = ComputeTarget.create(workspace = workspace,

name = aks_cluster_name,

provisioning_configuration = prov_config)

✓ **Correct**
This is the correct code for this task.

6. You're working with Azure Machine Learning and you want to train a Diabetes Model and build a container image for the trained model.

1 / 1 point

You will use the scikit-learn ElasticNet linear regression model.

You want to deploy the model to production using Azure Kubernetes Service (AKS).

You've created a AKS cluster for model deployment.

You've deployed the model's image to the specified AKS cluster.

After you've trained a new model with different hyperparameters, you need to deploy the new model's image to the AKS cluster.

How would you code that?

○ prod_webservice.create (image=model_image_updated)

prod_webservice.wait_for_deployment(show_output = True)

○ prod_webservice.deploy (image=model_image_updated)

prod_webservice.wait_for_deployment(show_output = True)

● prod_webservice.update(image=model_image_updated)

prod_webservice.wait_for_deployment(show_output = True)

○ prod_webservice.delete (image=model_image_updated)

prod_webservice.wait_for_deployment(show_output = True)

✓ Correct
This is the correct code for this task.

7. After working with Azure Machine Learning, you want to clean up the deployments and terminate the "dev" ACI webservice using the Azure ML SDK.

1 / 1 point

Which method should do the job?

○ dev_webservice.terminate()

● dev_webservice.delete()

○ dev_webservice.remove()

○ dev_webservice.flush()

✓ Correct
Because ACI manages compute resources on your behalf, deleting the "dev" ACI webservice will remove all resources associated with the "dev" model deployment