ScholarNest

Spark Azure Databricks

Databricks Spark Certification and beyond

Instructor: Prashant Kumar Pandey

Absolute Beginner to Specialization in Apache Spark and Azure Databricks

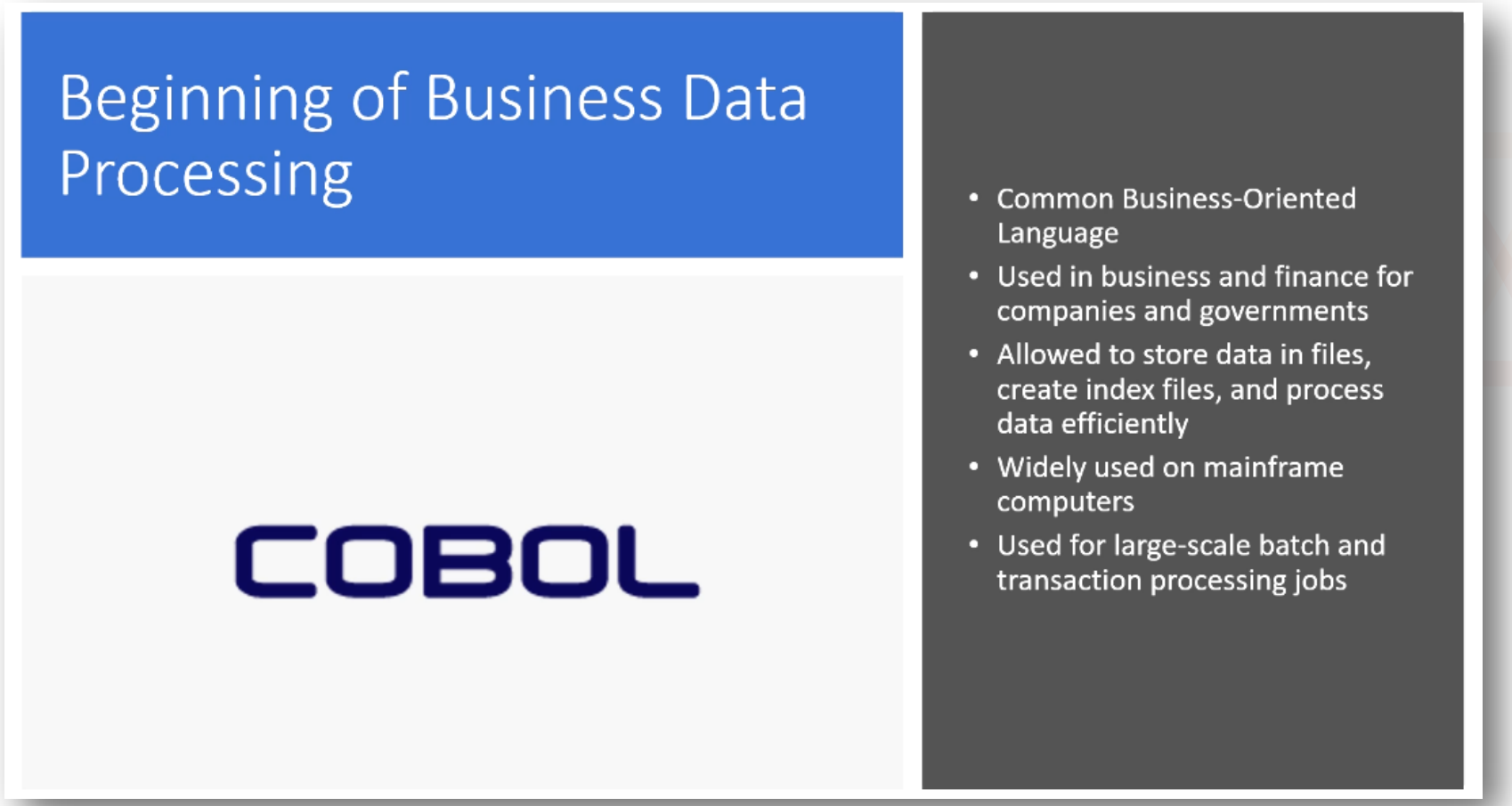
Emergence of Big Data

Categorizing software systems accurately and crafting a complete list of all possible categories is almost impossible.
However, we can still start with the following categories listed below.

Software Engineering Specializations

1. System Software
3. Programming Tools
3. Desktop Applications
4. Data Processing Application
5. Web & Mobile Applications
6. Software Platforms
7. Cloud Computing
8. Machine Learning and AI

We saw the evolution of the COBOL programming language specifically designed for business data processing.
The COBOL, also known as Common Business-Oriented Language, was the first of its kind.
COBOL allowed us to store data in files, create index files, and process data efficiently.
However, we saw data processing shift from COBOL to relational databases such as Oracle and Microsoft SQL Server.



Out of all the software specializations we have listed earlier, I wanted to attract your attention towards Data processing applications.
You can think of COBOL as the first serious attempt towards enabling data processing.
And COBOL was designed in 1959.
The Oracle database achieved the subsequent major success in enabling data processing.
And Oracle was founded in 1977.
So data processing has always been at the centre of the Software industry. Everything else will come and go, but data will only grow.

Data processing is an evergreen field for software engineers.
Technologies may evolve, but the requirement for processing more and more data at a faster speed will keep becoming more critical.

We have used RDBMS technology for many decades.
Some popular RDBMS systems are Oracle, SQL Server, PostgreSQL, MySQL, Teradata, and Exadata.
These RDBMS systems offered us three main features to help us develop Data Processing applications.

1. SQL - An easy Data Query Language
2. Scripting Languages such as PL/SQL and Transact SQL
3. Interface for other programming languages such as JDBC and ODBC

So we used SQL for querying data and PL/SQL for doing things that we couldn't do using SQL.
They also offered interfaces such as ODBC/JDBC so we could interact with data using the programming languages.
We could create data processing applications using these technologies.

We had the following requirements in the Big Data problem.

There was two approaches to the big data problem:

1. The monolithic approach designs one large and robust system that handles all the requirements. Teradata and Exadata are examples. These two systems mainly support only structured data. So we cannot call them big data systems, but they are designed using a monolithic approach.
2. The distributed approach takes many smaller systems and bring them together to solve a bigger problem.

Here are the two approaches to Big Data solution.

Approaches of Big Data Solution

Criteria

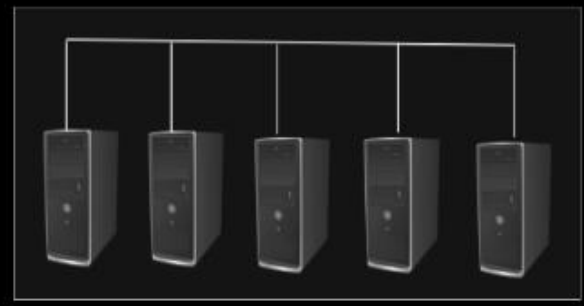
1. Scalability
2. Fault Tolerance and HA
3. Cost-Effectiveness

Monolithic Approach

1. Vertical
2. Primary/Secondary
3. Expensive

Distributed Approach

1. Horizontal
2. Multifold
3. Economical



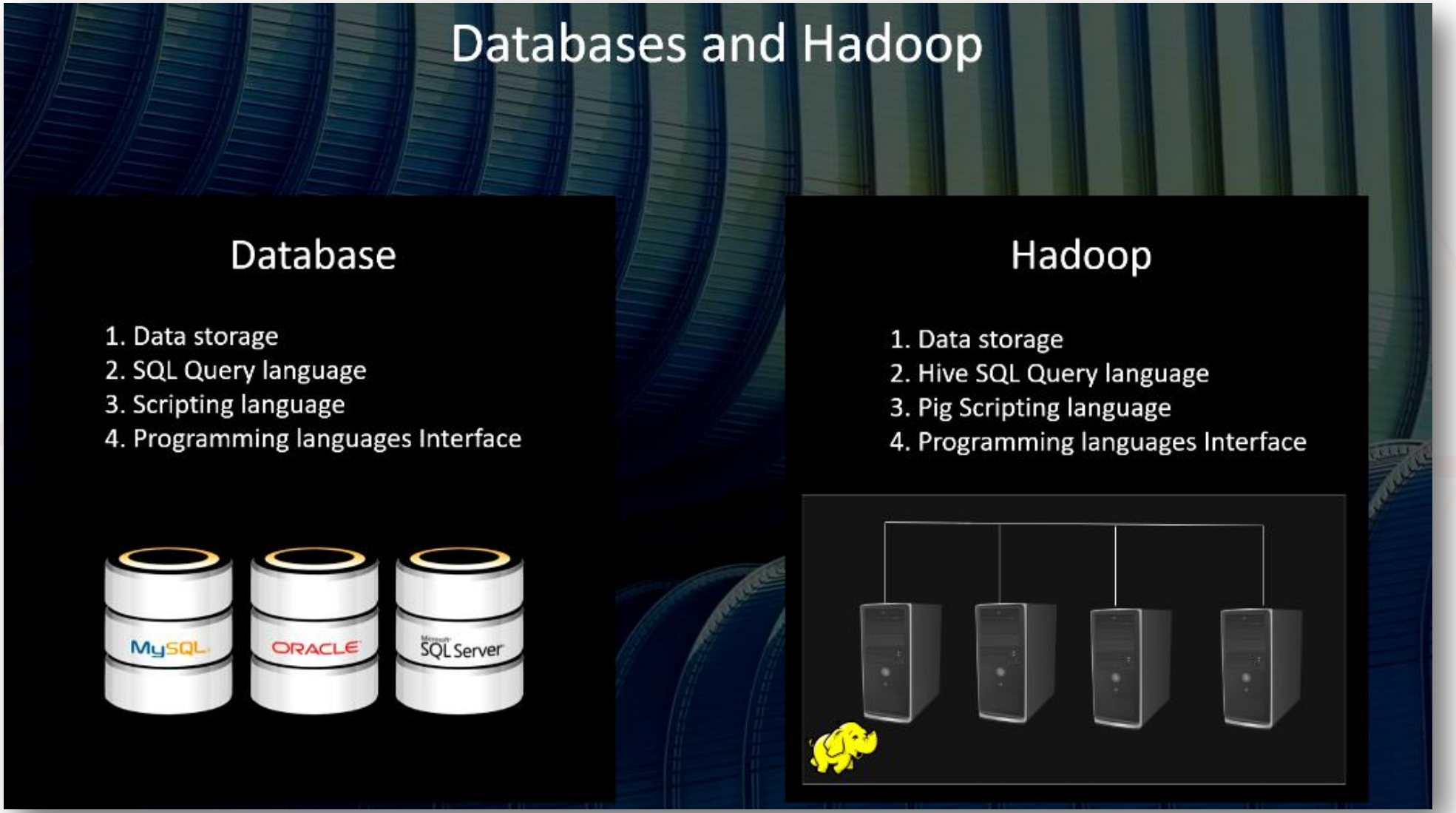
Hadoop came up as a new data processing platform to solve Big Data problems.

The Hadoop platform was designed and developed in layers.

The core platform layer offered three capabilities:

1. Distributed cluster formation or Cluster Operating System
2. Data storage and retrieval on the distributed cluster or Distributed Storage
3. Distributed data processing using Java programming language or Map-Reduce Framework

Here is a comparison between Database and Hadoop.



Thank You

ScholarNest Technologies Pvt Ltd.

www.scholarnest.com

ScholarNest

Spark Azure Databricks

Databricks Spark Certification and beyond

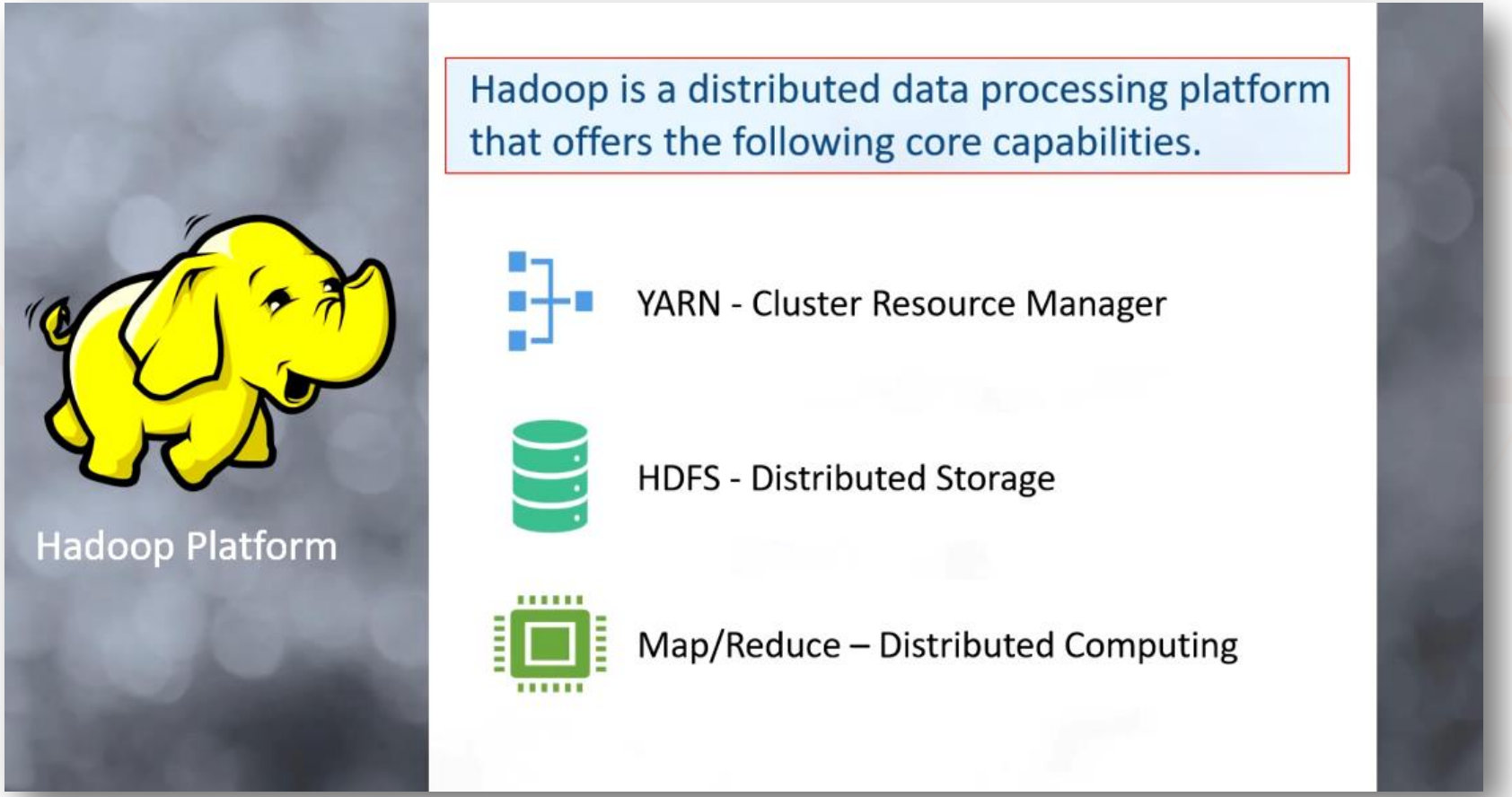
Instructor: Prashant Kumar Pandey

Absolute Beginner to Specialization in Apache Spark and Azure Databricks

Hadoop Architecture – History and Evolution

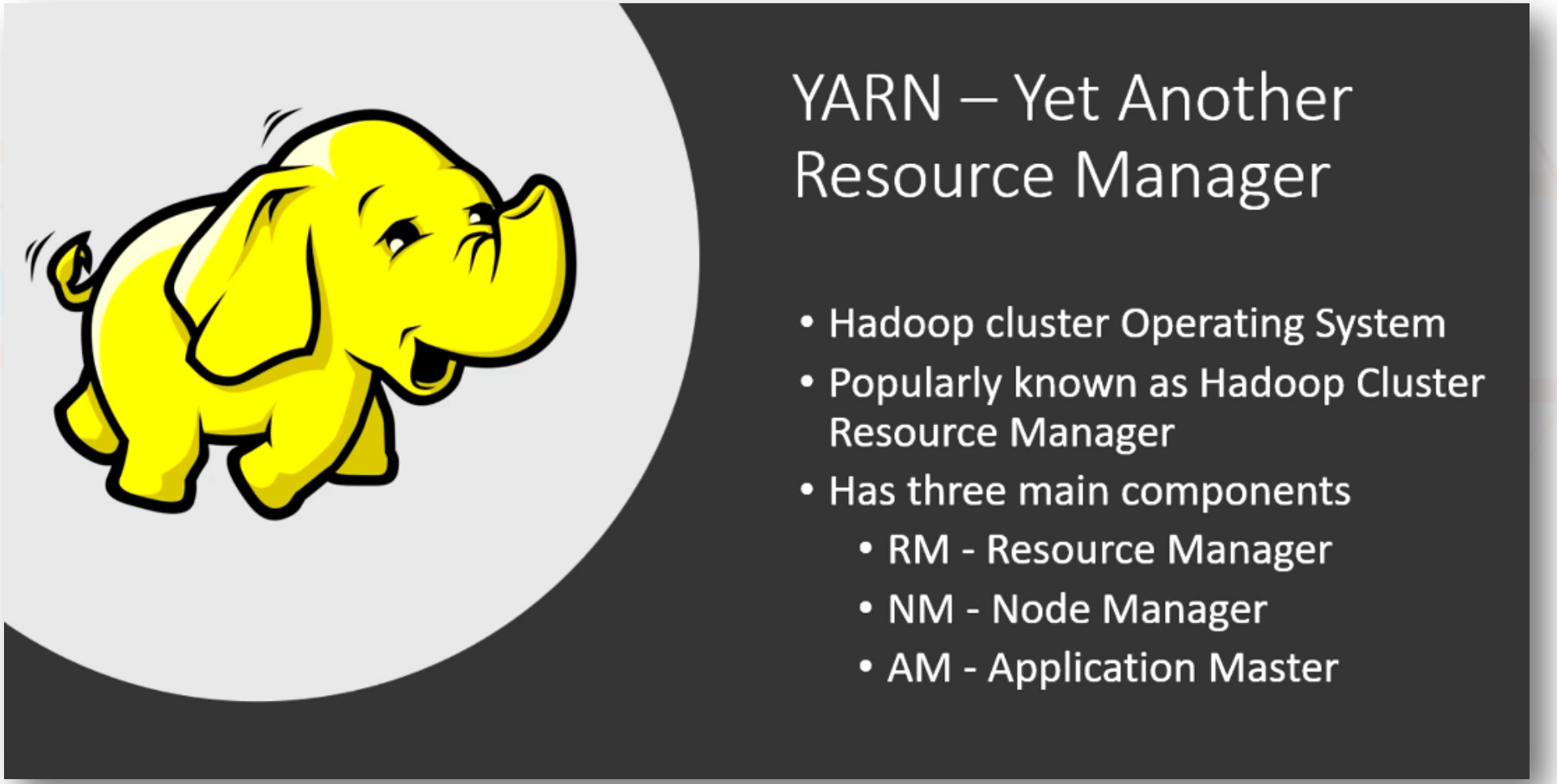
Hadoop is a distributed data processing platform that offers three core capabilities listed below:

1. YARN
2. HDFS
3. Map/Reduce



YARN:

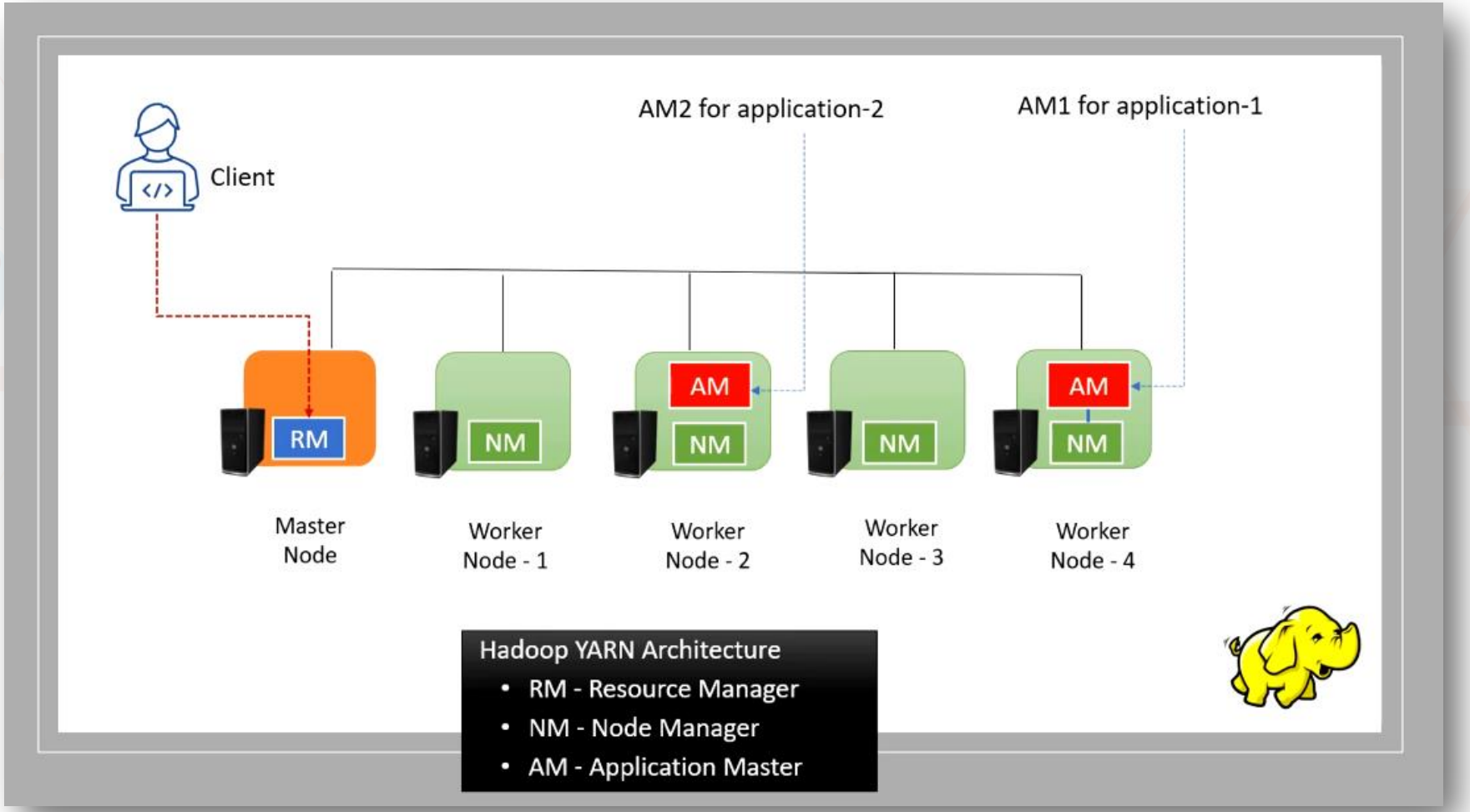
YARN is the Hadoop cluster resource manager. It allows multiple applications to run on the Hadoop Cluster and share resources amongst the applications.



YARN:

Assume we installed Hadoop, and now these five computers form a Hadoop cluster. Hadoop uses a master-slave architecture.

So one of these machines will become the master, and the remaining will act as the worker node.



YARN:

We have a five-node cluster that I showed you in the earlier slide. One node acts as a master and runs the YARN resource manager service. The other four nodes act as a Worker and run a node manager service.

The node manager will regularly send the node status report to the resource manager. We created a Hadoop cluster so we can run big data applications. For running an application on Hadoop, you must submit the application to the YARN resource manager. Assume you submitted a Java application to the YARN using a command line submit tool. Now the resource manager should run this application on the cluster.

So, the resource manager will request one of the node managers to start a resource container and run an AM (application master) in the container. And your application starts running inside the Application Master container.

So, we submit our application to the Resource Manager.

The resource manager requests the node manager for allocating an application master container and starting your application inside the AM container. Each application on YARN runs inside a different AM container.

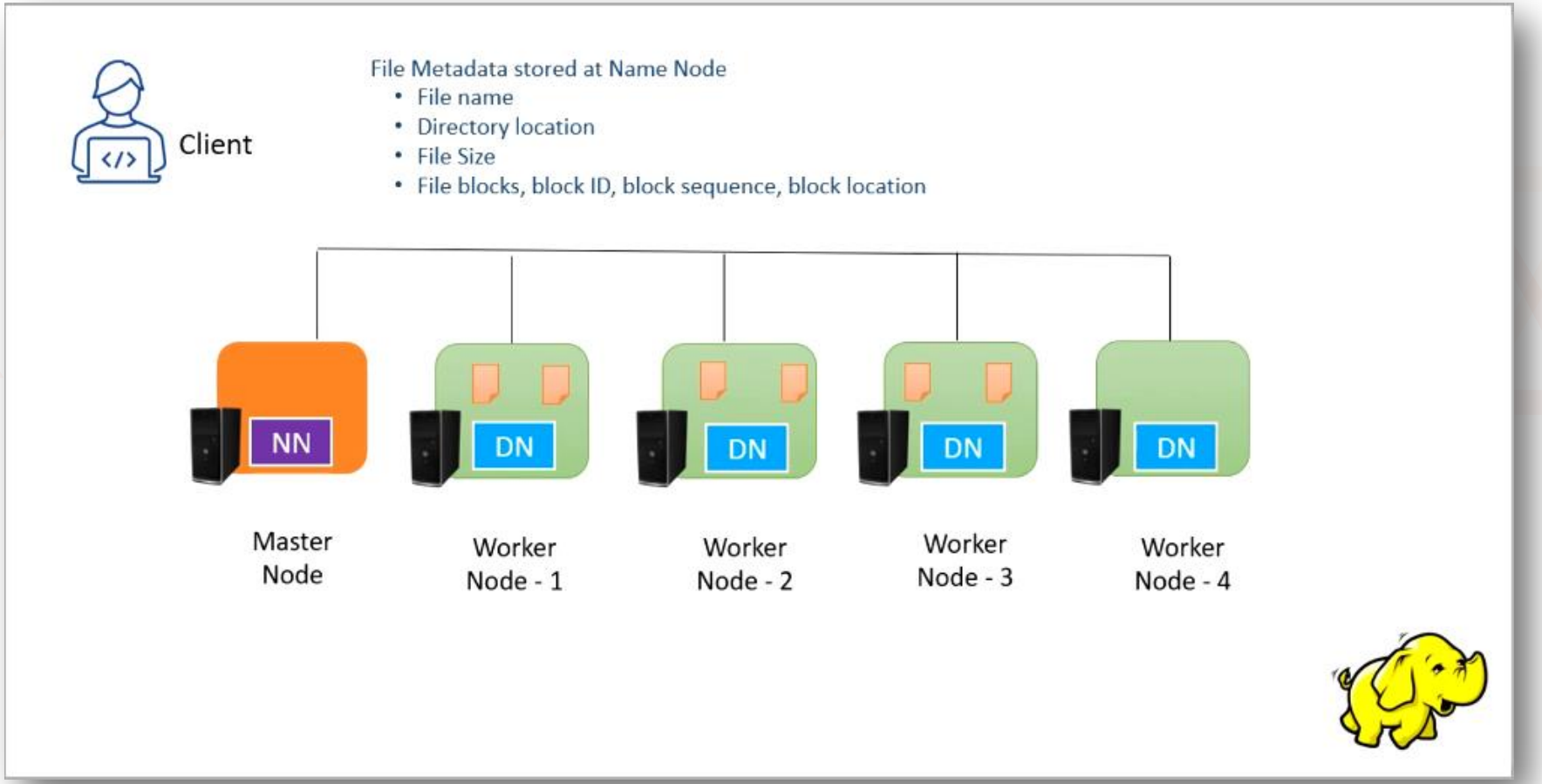
If you have ten applications running in parallel, you will see 10 AM containers on your Hadoop cluster.

HDFS:

The HDFS stands for Hadoop Distributed File system, and it allows you to save and retrieve data files in the Hadoop Cluster.

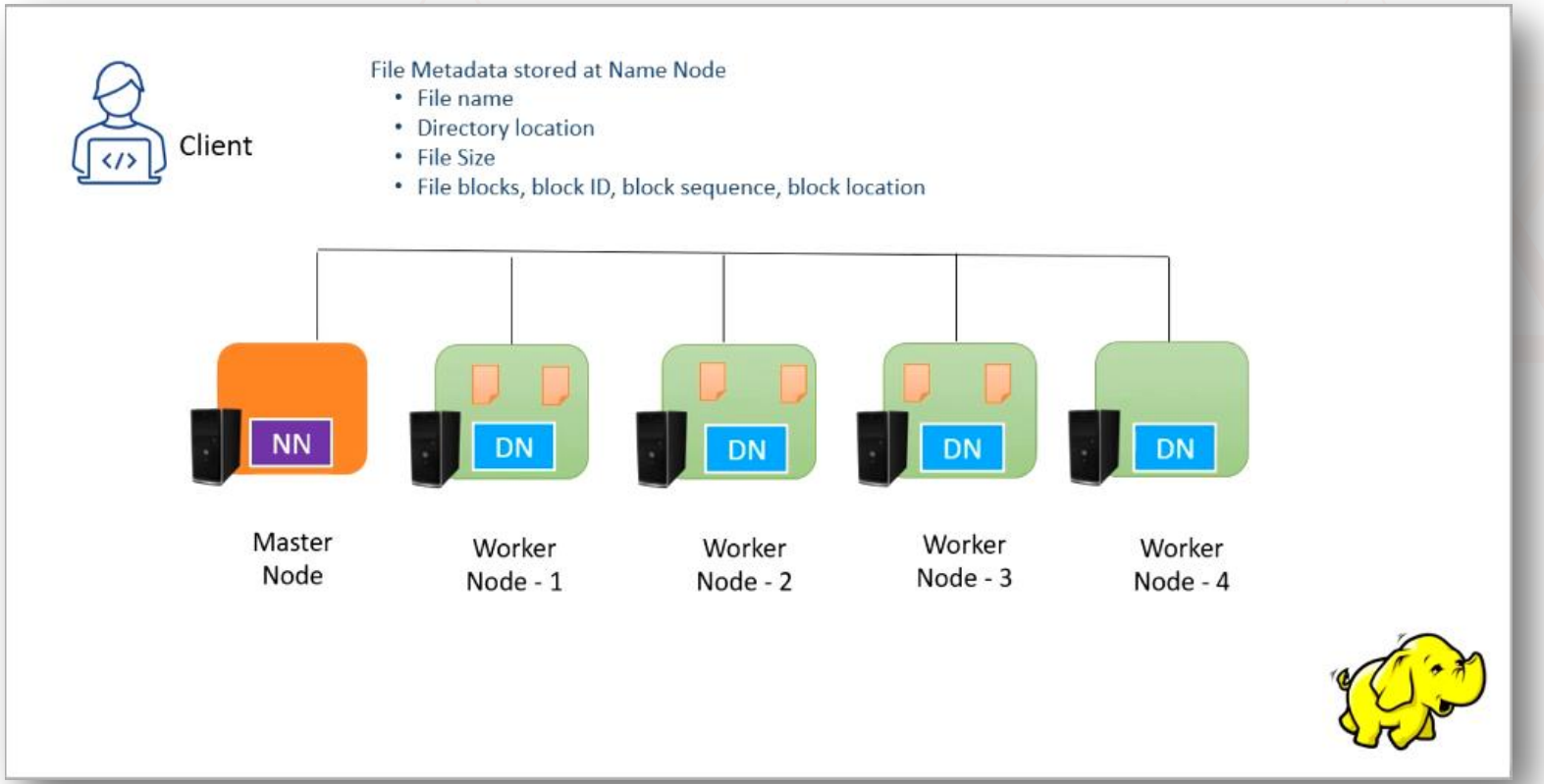
The HDFS has the following components.

1. Name Node (NN)
2. Data Node (DN)



HDFS:

Assume we have five computers shown below. We already installed Hadoop on these computers and created a Hadoop cluster. Hadoop will install the Name Node service on the master.
And each worker node runs a data node service. The name node with the data node service forms the HDFS. The primary purpose of the HDFS is to allow us to save files on this Hadoop cluster and read them whenever required.



Map/Reduce:

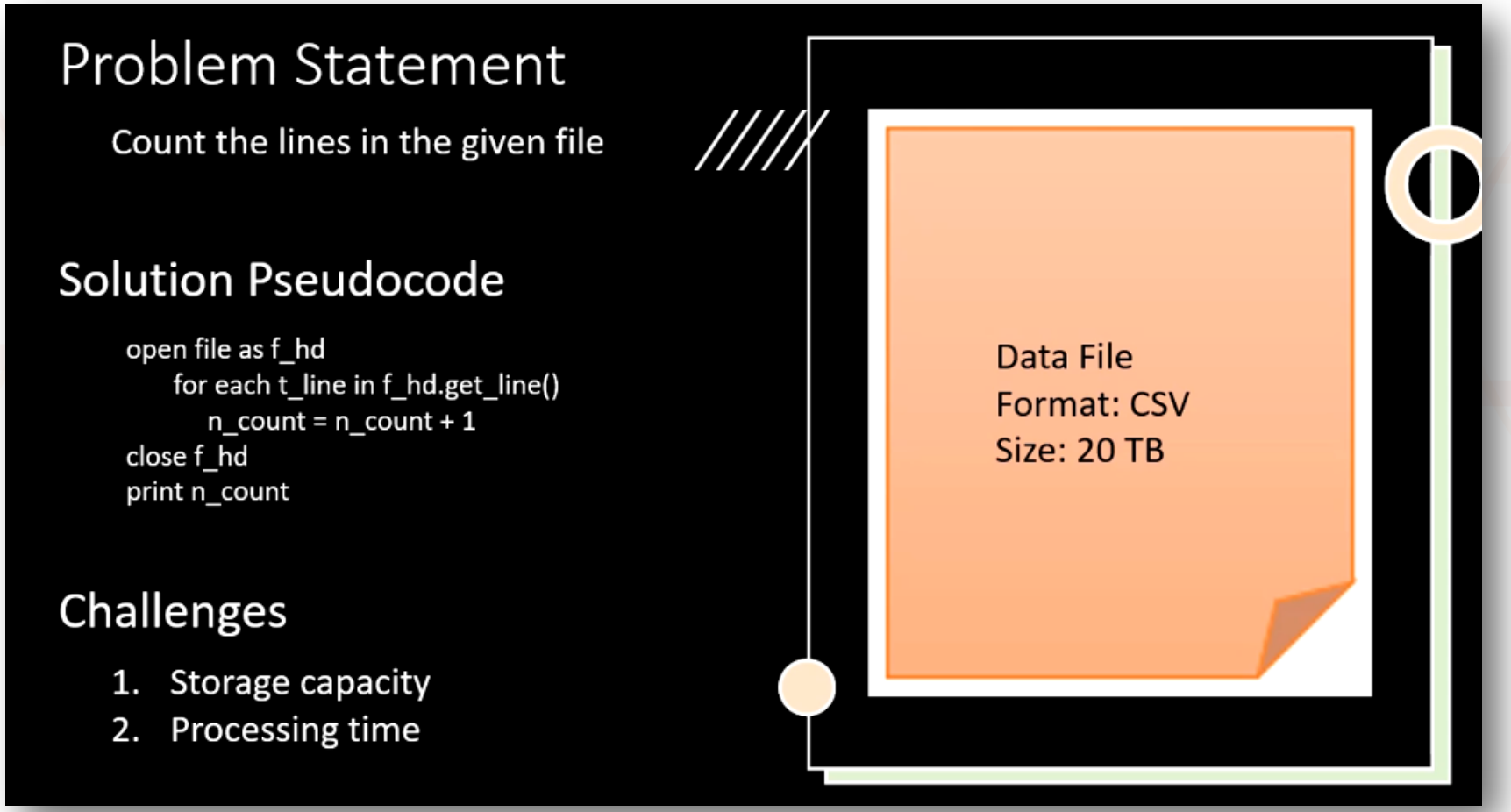
Map-reduce is a programming model and a framework. A programming model is a technique or a way of solving problems. The M/R framework is a set of APIs and services that allow you to apply the map-reduce programming model.

Hadoop taught us the map-reduce programming modal and also offered a Map-Reduce programming framework to implement it.

Map/Reduce:

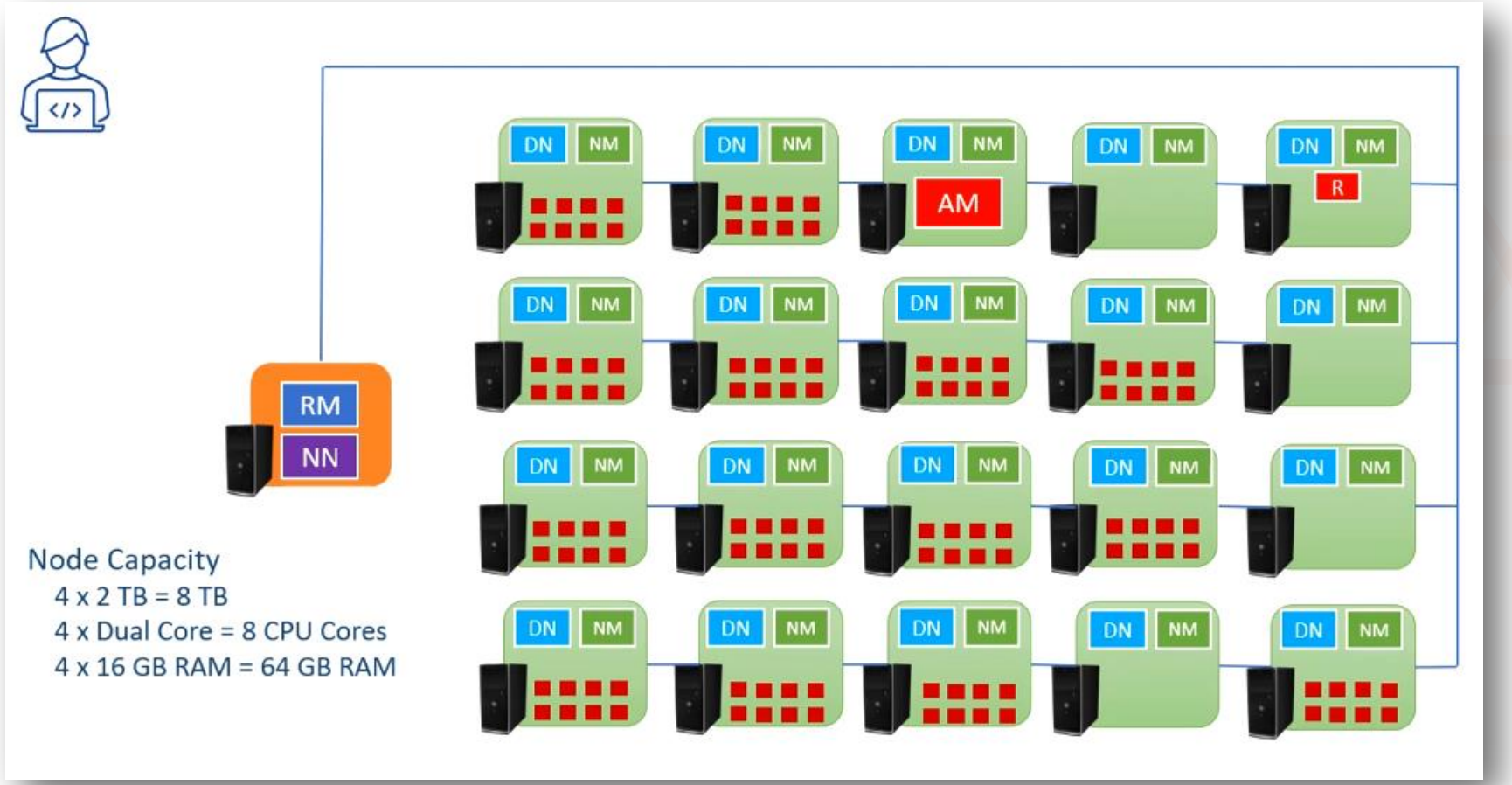
You have to count the lines in a 20 TB CSV file. There are two challenges in this problem statement:

1. Huge file size, It is hard to find machines to store 20 TB of data. And this problem becomes more complex if we grow the size in petabytes.
2. We also have a processing time problem. A simple line count on a 20 TB file takes hours or days.



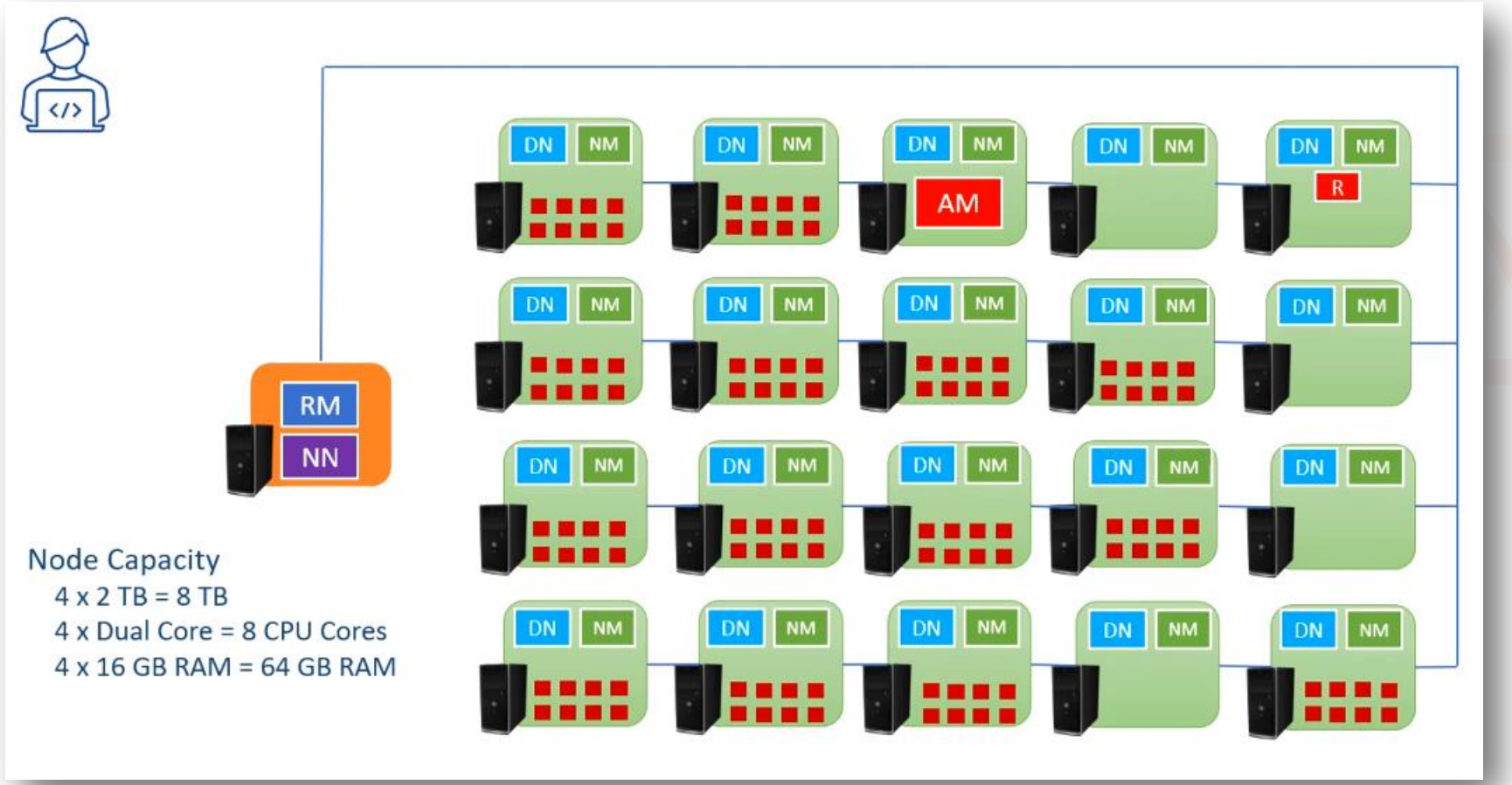
Map/Reduce:

Hadoop offered to solution to both problems we discussed in the previous slide. You can use the Hadoop cluster to store the file. Let's assume you have a 21 node Hadoop cluster. One node becomes the master, and the other 20 nodes are the workers. HDFS runs a name node in the master and a data node on the other workers. YARN runs a Resource Manager on the master, and Node Manager runs on the workers. So we have those services running on the cluster.



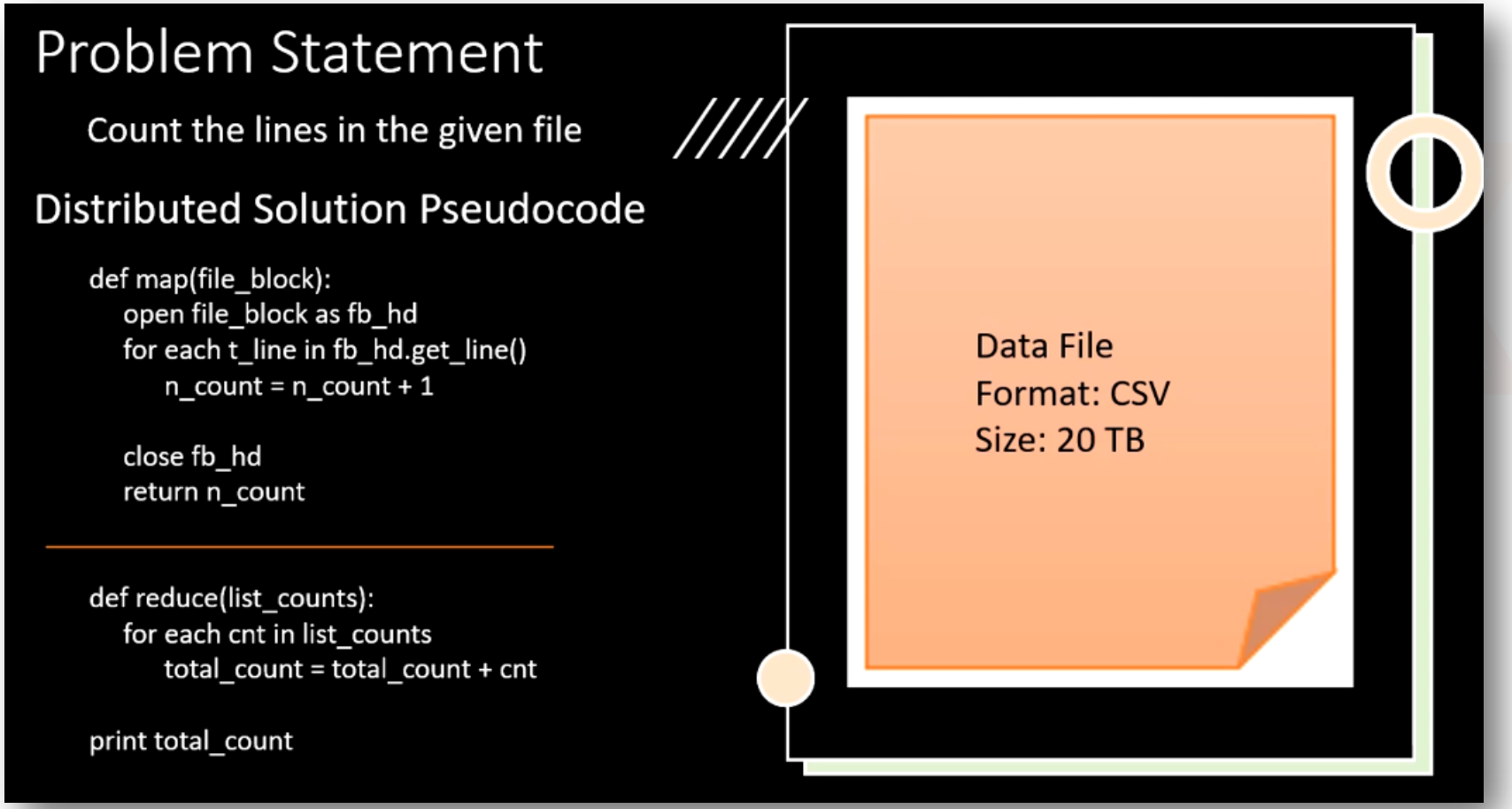
Map/Reduce:

You can use HDFS to copy your 20 TB file on this cluster. HDFS will break the file into small 128 MB blocks and spread them across the cluster. So some data nodes will sore data blocks, and altogether they can easily store your 20 TB file. Your storage problem is taken care of by the Hadoop cluster. If you need more storage, you can increase the cluster size and add more computers.



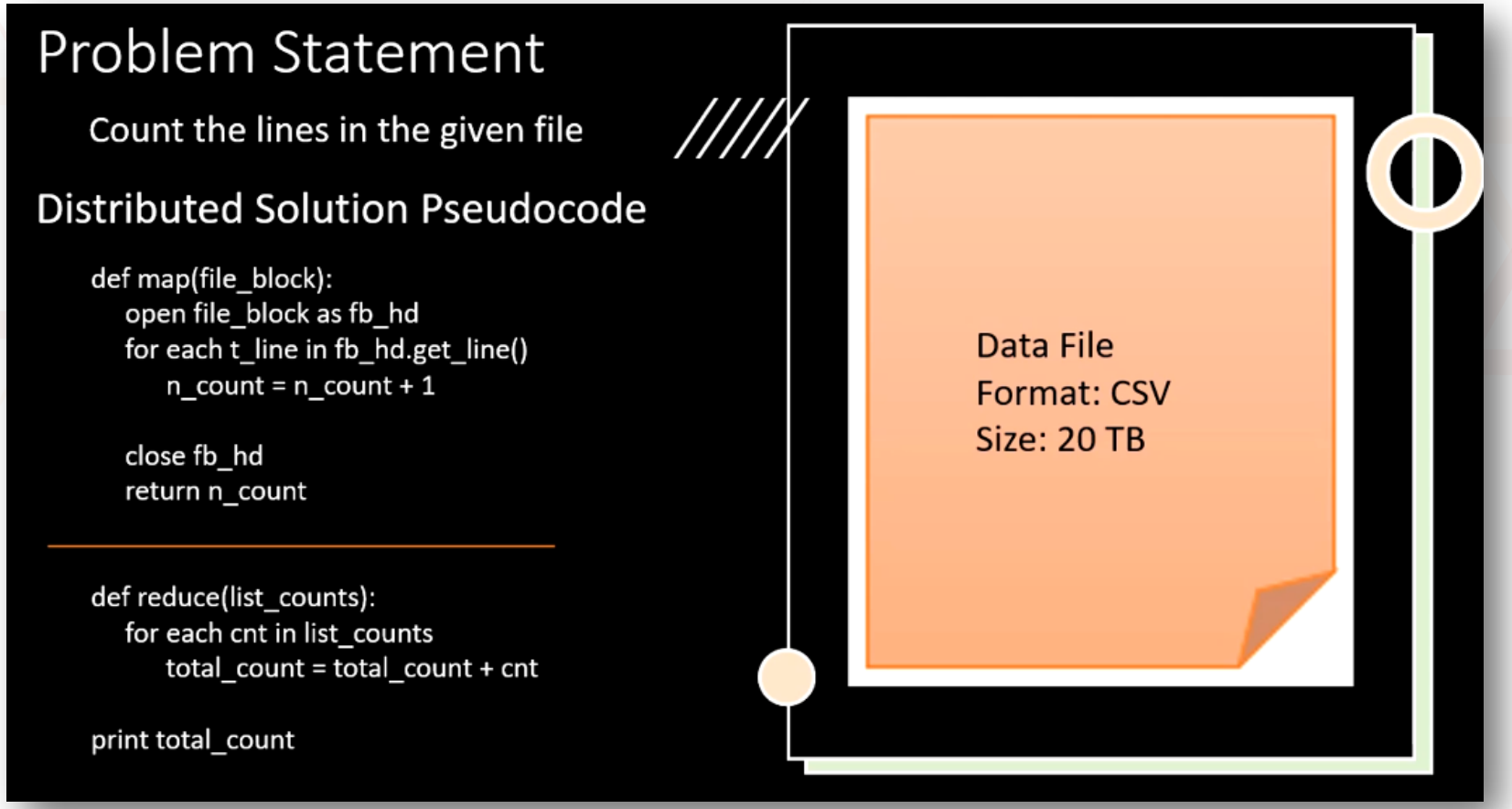
Map/Reduce:

Now let us come to the processing time problem. I have broken my logic into two part which you can see in the image below. The first part is known as the Map function. The second part is known as the Reduce function. The old logic was to open the file and count the lines. And the new logic is almost the same as old logic. But the map function opens the file block and counts the lines. And the old logic opens the file and counts the lines.



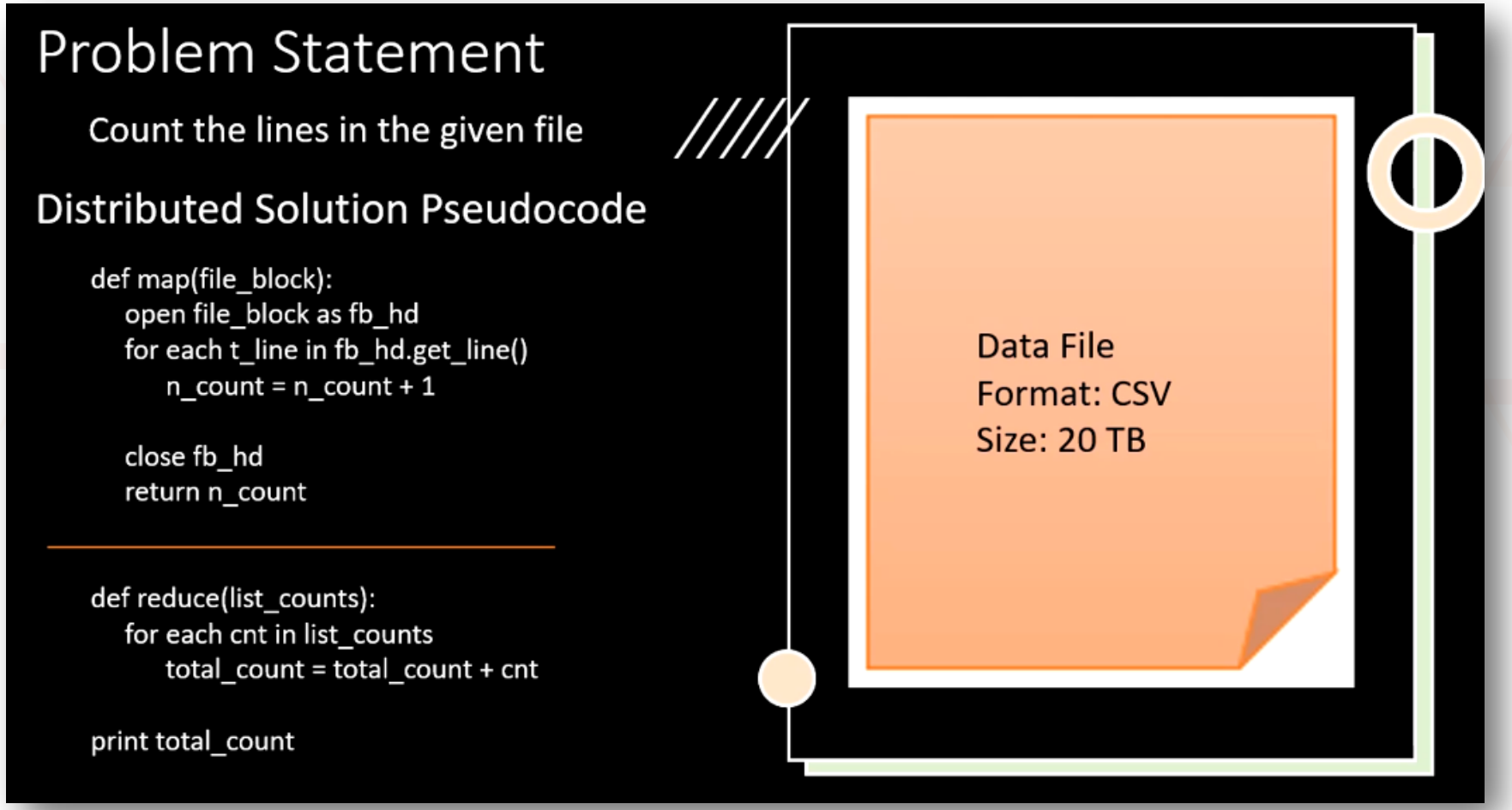
Map/Reduce:

I can run the map function on all the data nodes in parallel. This map() function will open each block on the data node and count the lines. End of the execution, I will have the number of lines in the blocks at the given data node. I am counting lines on 14 data nodes in parallel. Everything runs at the same time. And I will get the line counts in 1/14th of the time compared to doing it on a single machine. However, I will have 14 line counts. Each count represents the number of lines on their respective data node.



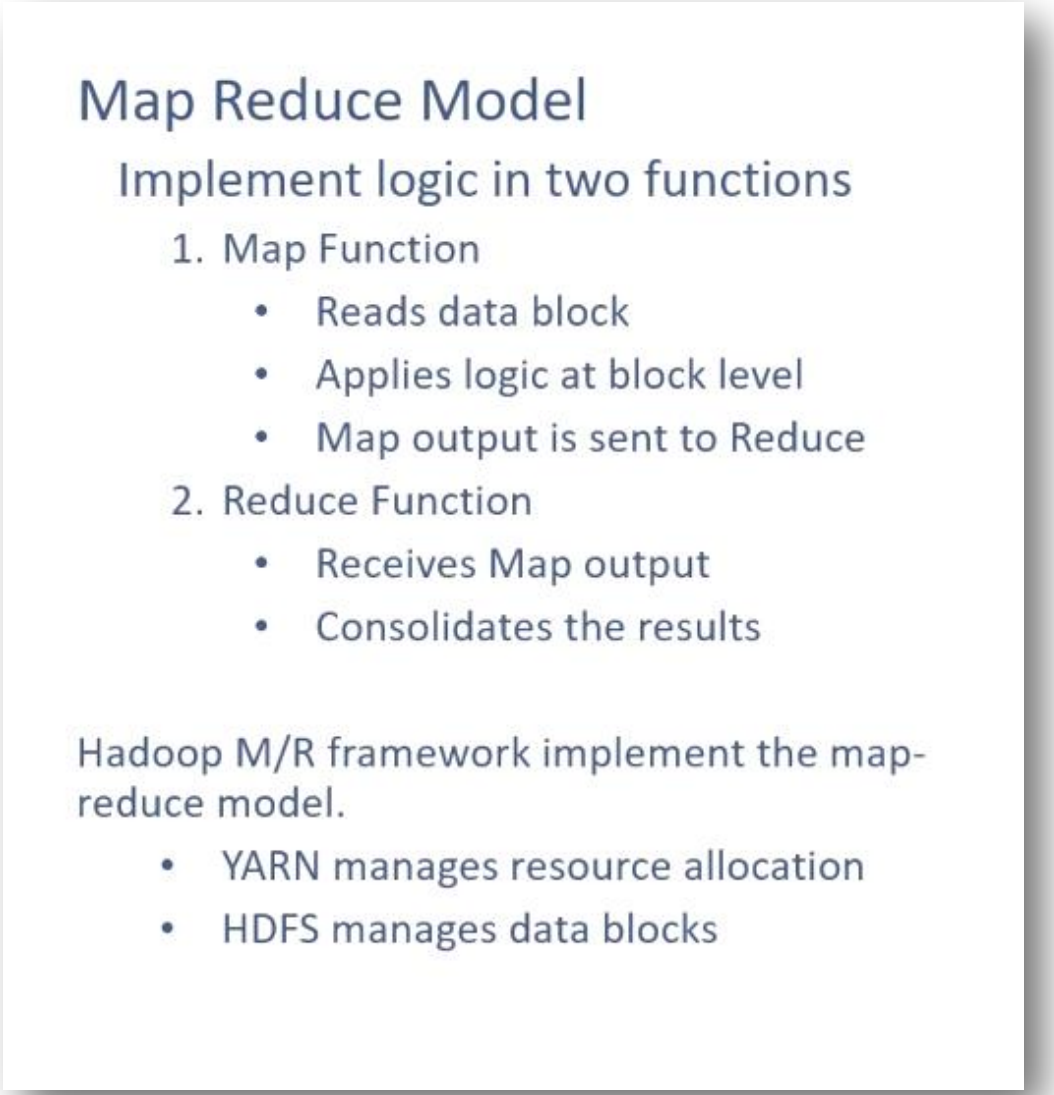
Map/Reduce:

Then, I will start a Reduce function at one node. All the data node will send their counts to the reduce function. The reduce function will receive 14 line counts in an array. So I will look through the array and sum up all the line counts. The reduce function will loop through the list of counts and sum it up. And the sum is the number of lines in the file.



Map/Reduce:

Here is the summarized context of Map Reduce.



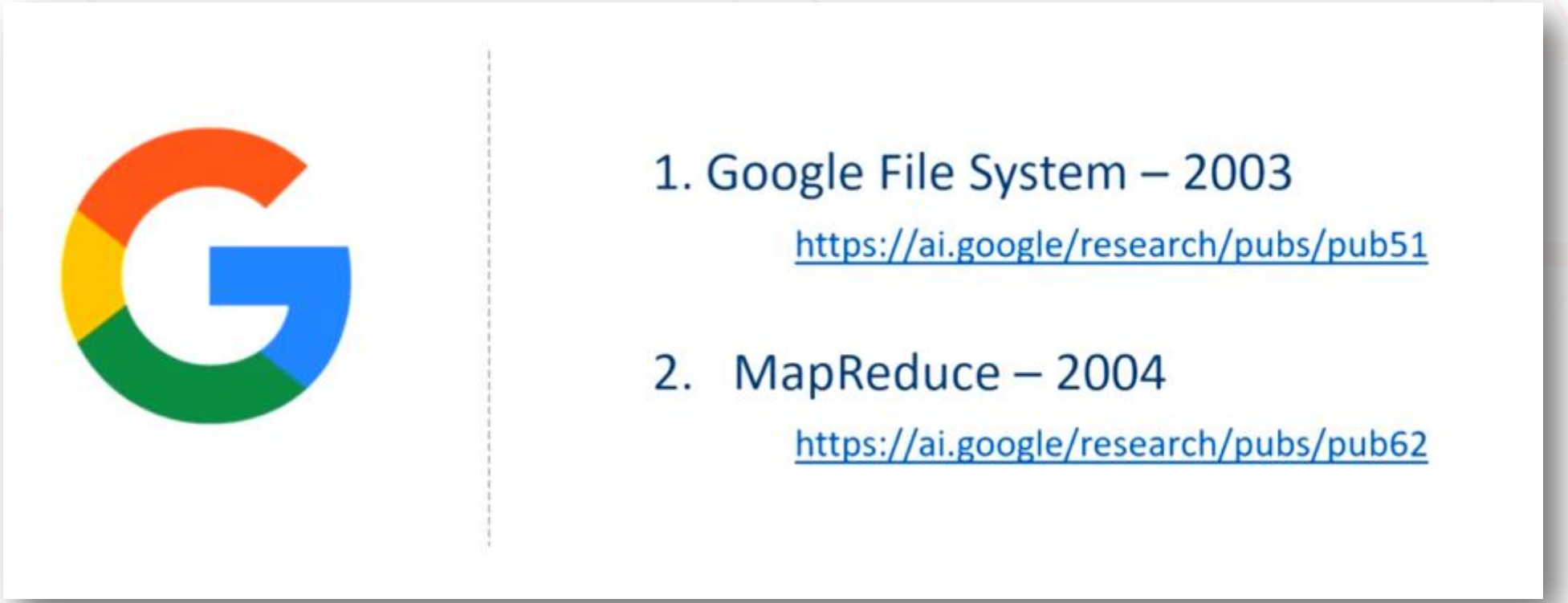
The three Big Data problems:

1. High Data Volume
2. Variety of Data
3. High speed

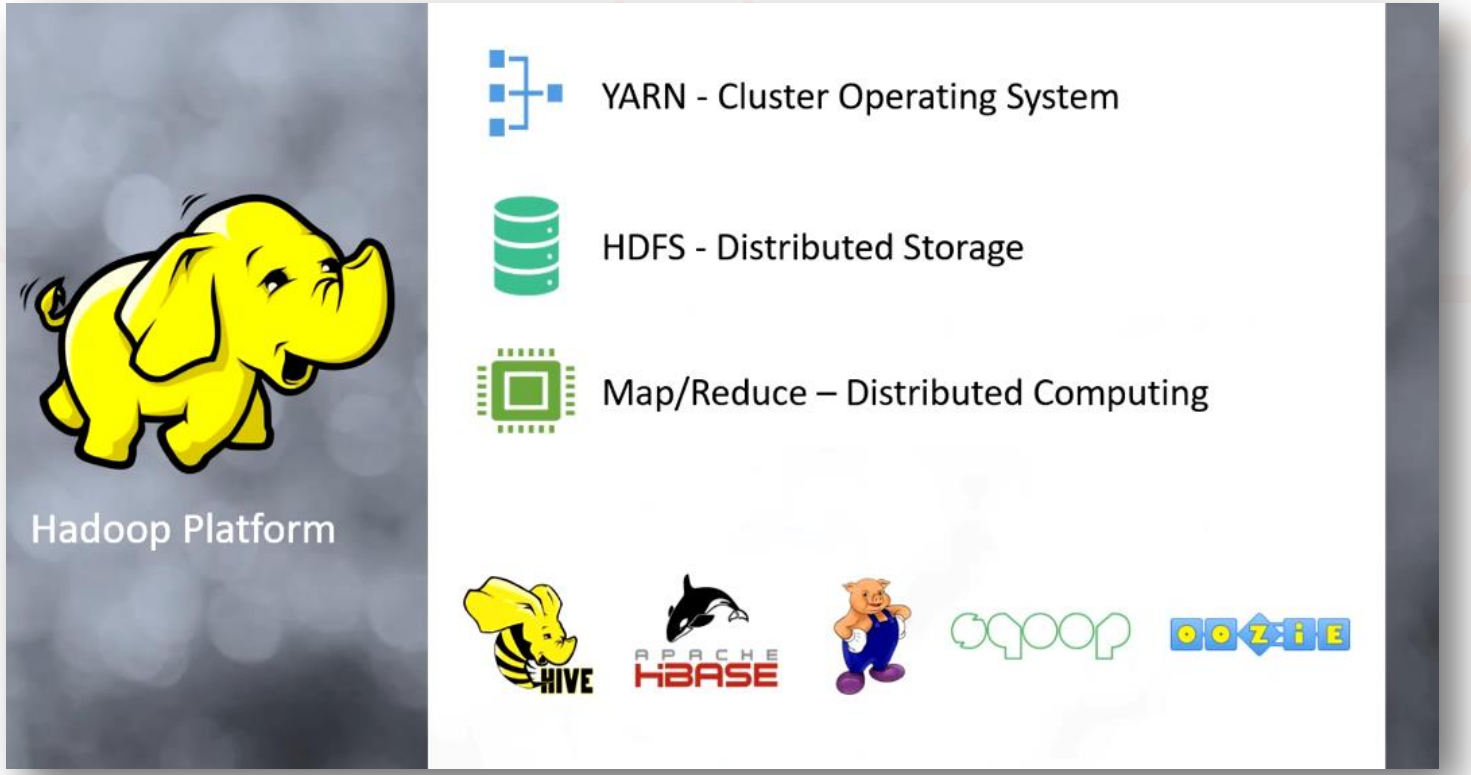
Google was the first company to realize the big data problem. And they were also the first company to develop a viable solution and establish a commercially successful business around it.

Google had four main problems to solve which are listed below.
They were creating a search engine, and the first thing they wanted to do was to discover and crawl the web pages over the Internet and capture the content and metadata. We now categorize these problems as data collection and data ingestion.
Once the webpage-related data is collected, they want to store and manage it. So the next problem was to store and manage hundreds of petabytes of data.
The next problem was to get the computation power for processing those massive volumes.
They wanted to apply the PageRank algorithm to the received data and create an index. So, the third problem was to get the required processing power.
Finally, the last problem was organizing and storing the outcome of the processing. In Google's case, the result was the index.
They wanted to keep it in a random-access database to support high-speed queries by the Google Search Engine application.

Google successfully solved all the four problems, and they were generous enough to reveal their solution to the rest of the world in a series of white papers. Google published the first whitepaper in 2003, and it talked about solving the Data Storage and Management problem. They termed the solution as Google File System (GFS).
The second whitepaper was published by Google in 2004 and talked about the Data Processing and Transformation problem. They termed it as MapReduce (MR) programming model.



The open-source community well appreciated these whitepapers, and they formed the basis for the design and the development of a similar open-source implementation – The Hadoop. The open-source community implemented the GFS as a Hadoop Distributed File System – HDFS. They applied Google MR as the Hadoop MapReduce programming framework.
Hadoop grabbed immense attention and popularity among organizations and professionals.
Since the development of Hadoop, there have been many other solutions developed over the Hadoop platform and open-sourced by various organizations.
Some of the most widely adopted systems were Pig, Hive, and HBase.



Apache Hive was one of the most popular components of Hadoop. Hive offered the following core capabilities on the Hadoop Platform:
1. Create Databases, Tables, and Views
2. Run SQL Queries on the Hive Tables

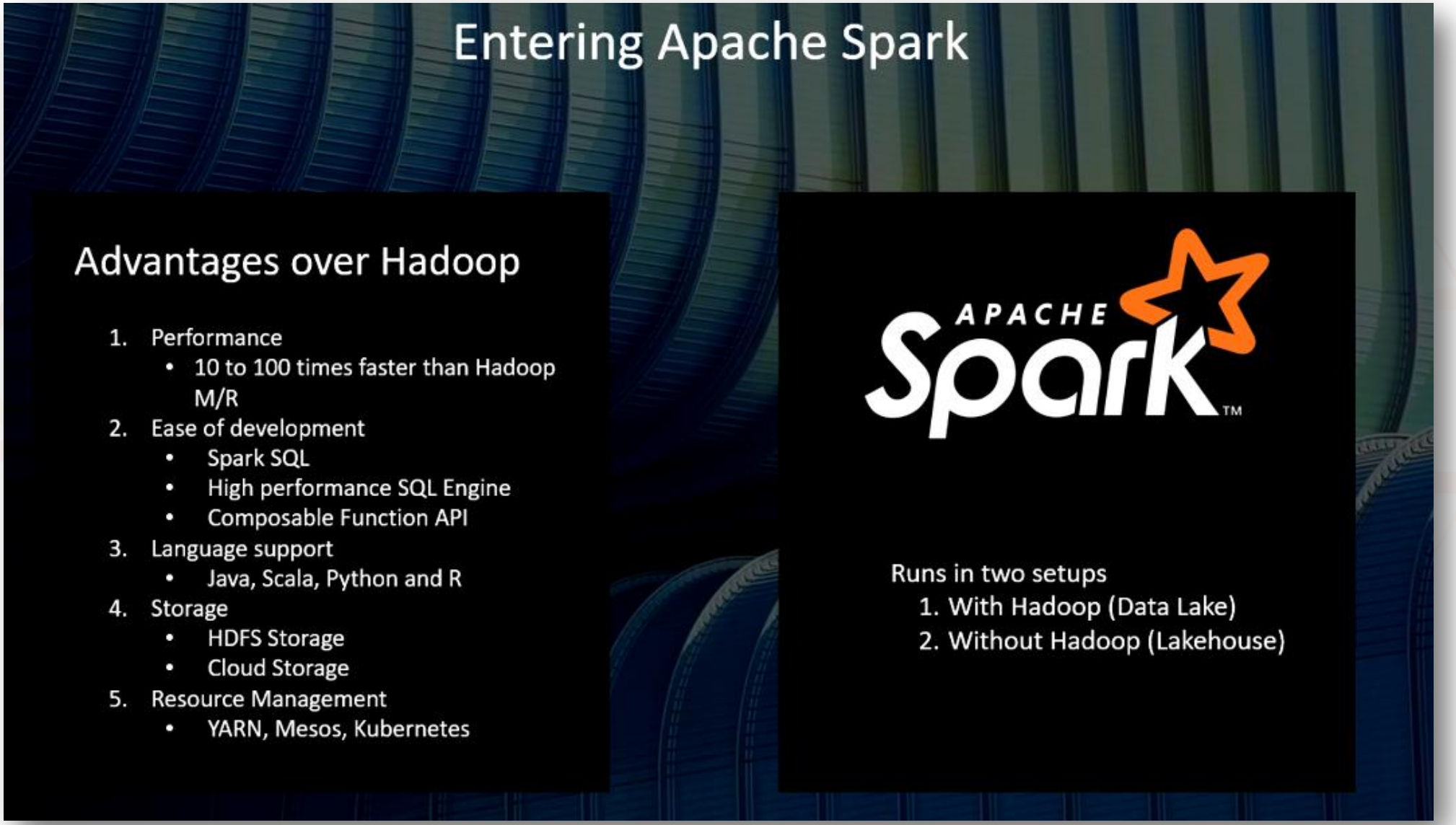
So Hive simplified using Hadoop. Application developers struggled to solve data processing problems using Map Reduce. Hive came to the rescue. It allowed us to create databases and tables using DDL Statement.
Then they also allowed us to use SQL queries on the table.
The majority of the development workforce was familiar with the RDBMS, and they already knew SQL.
So using SQL was easy to adopt.

Hive SQL engine internally translated SQL queries into M/R programs.
But application developers were saved from writing Map Reduce code in Java.

Hadoop as a platform and Hive as a Hadoop database became very popular.
But we still had the following problems which needed improvements:

1. Performance
2. Ease of development
3. Language support
4. Storage
5. Resource Management

Apache Spark comes to rescue.



Spark exists on two kinds of platforms:

1. On Hadoop Platform - Data Lake
2. On Cloud Platform - Lakehouse

We use the Hadoop platform as the Data Lake platform, and the primary developer technology on Hadoop Data Lake is now Apache Spark.
Map/Reduce Framework is gone away forever, and Hive is also losing its place for Spark SQL.

The Cloud platforms are more popular these days.
So the idea of Hadoop Data Lake is now advanced and modernized with a new name of Lakehouse on the cloud platforms.
The driving force behind Cloud Lakehouse is the Databricks Spark platform.
So Spark is again the primary developer technology for Lakehouse.

Thank You

ScholarNest Technologies Pvt Ltd.

www.scholarnest.com

ScholarNest

Spark Azure Databricks

Databricks Spark Certification and beyond

Instructor: Prashant Kumar Pandey

Absolute Beginner to Specialization in Apache Spark and Azure Databricks

Data Lake – Emergence and Use

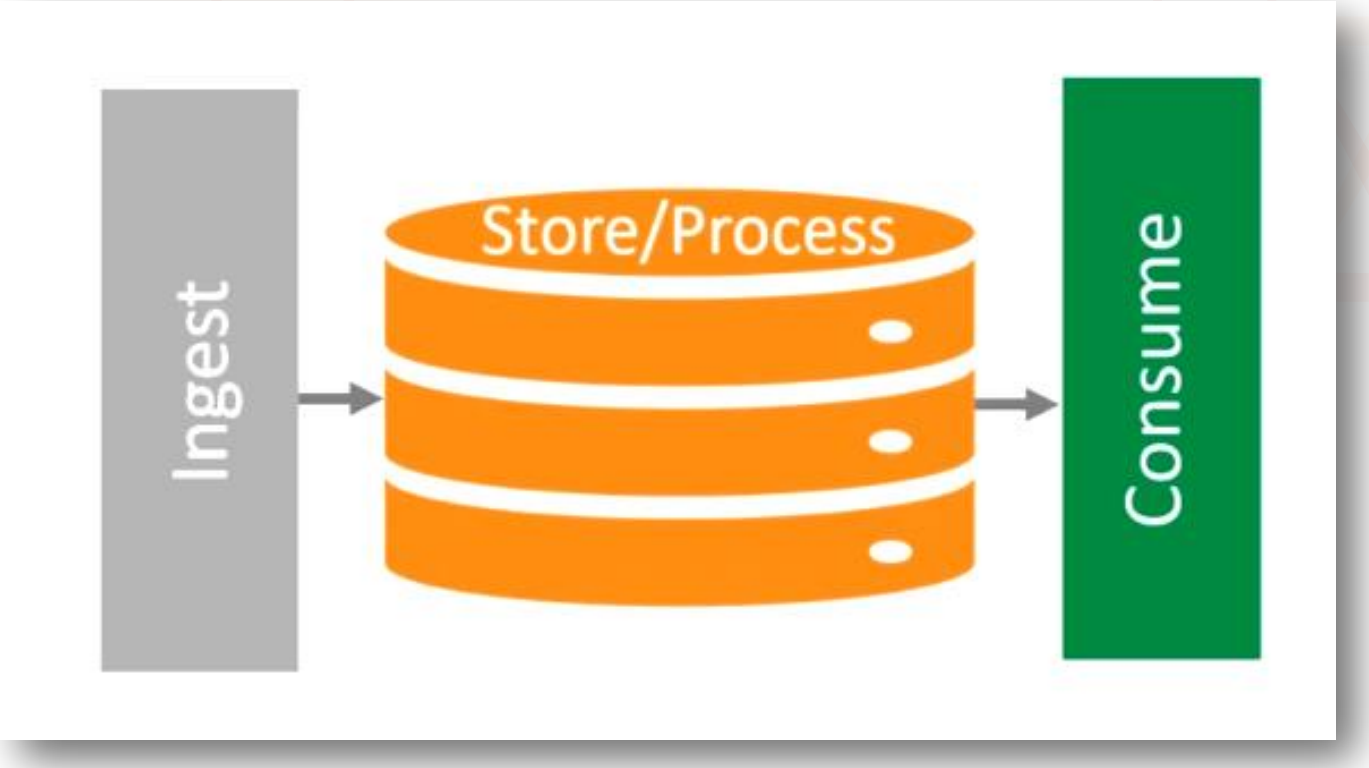
We know that the distributed computing started with Google finding a solution for their storage requirements using GFS. The open-source community created a similar solution called HDFS that allowed us to form a cluster of computers and use the combined capacity for storing our data. Then we also got the MapReduce framework which allowed us to use the combined computing power of the cluster and use it to process the enormous data volumes that we stored in HDFS.

Before the HDFS and Map/Reduce came into existence we had Data Warehouses - Like Teradata and Exadata. We created pipelines to collect data from many OLTP systems and brought them into the Data Warehouse. Then we processed all that data to extract business insights and used it to make the correct business decision.

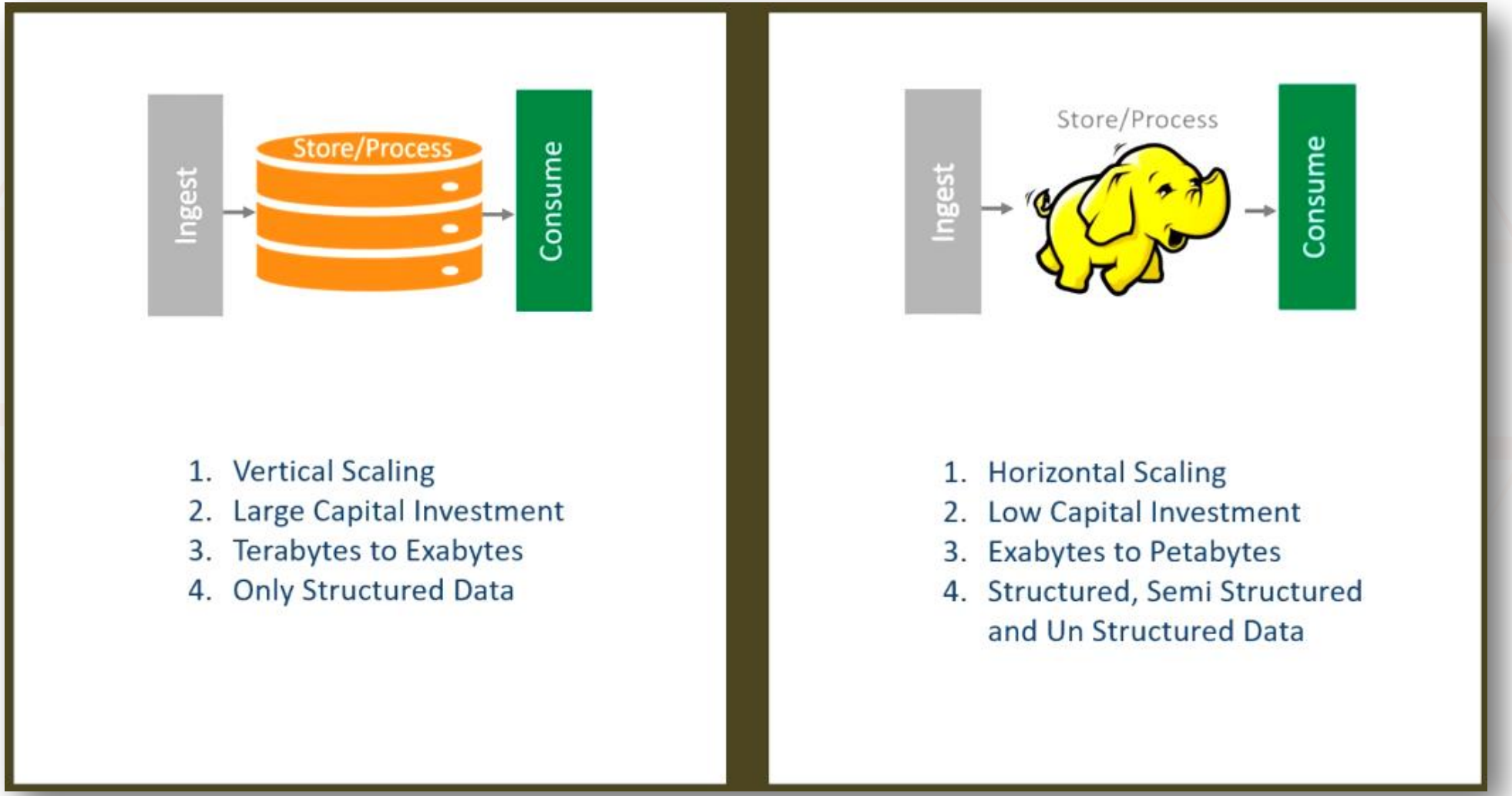
Hadoop also offered to collect data and process it to extract business insights.

So the advent of HDFS and MR started challenging these Data Warehouses in three critical respects:

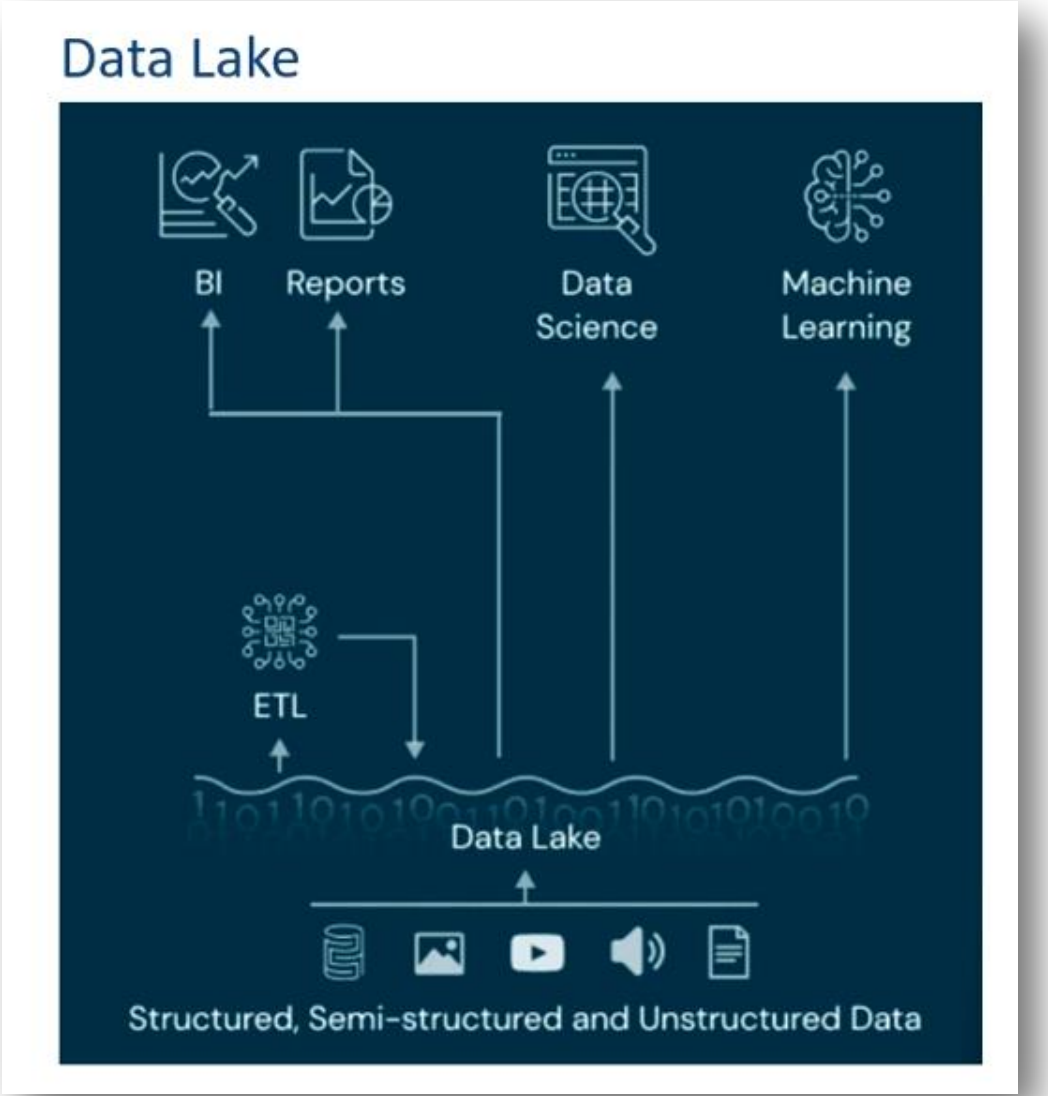
1. Ease of Horizontal Scalability
2. Capital Cost
3. Volume and Variety of Data



Here is a comparison between Data Warehouse and Hadoop offerings.



Data Warehouses were challenged, and they needed a new name for the Hadoop approach. And it is when the Data Lake came into existence.

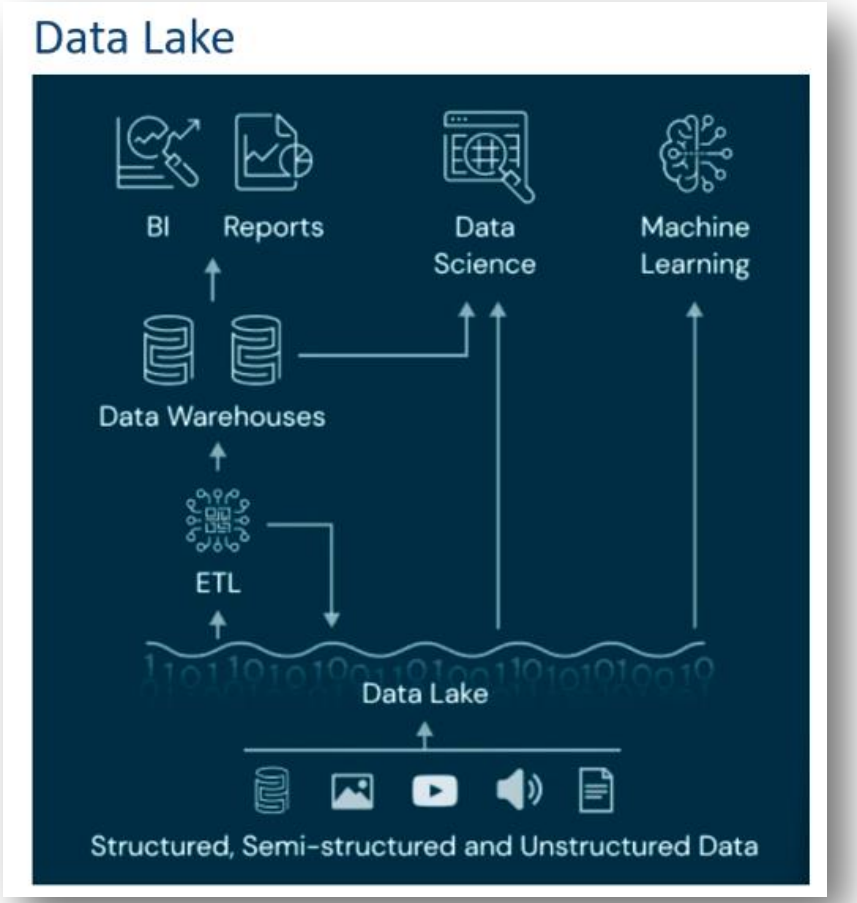


Same as Data warehouses, we collected data from different data sources and stored them in HDFS.
Then we used Map Reduce and Spark to process this data and prepare new data models for generating reports and business insights. Spark took over the Map/Reduce and other tools over a period of time, so it is safe to consider that we used Spark to process and prepare data for reporting.
This processed data was also stored in Data Lake storage for business intelligence and reporting.
Most of the popular BI and reporting tools offered a connector to access data from the data lake.
So, Data Lake also allowed us to collect and process huge volumes of semi-structured and unstructured data.

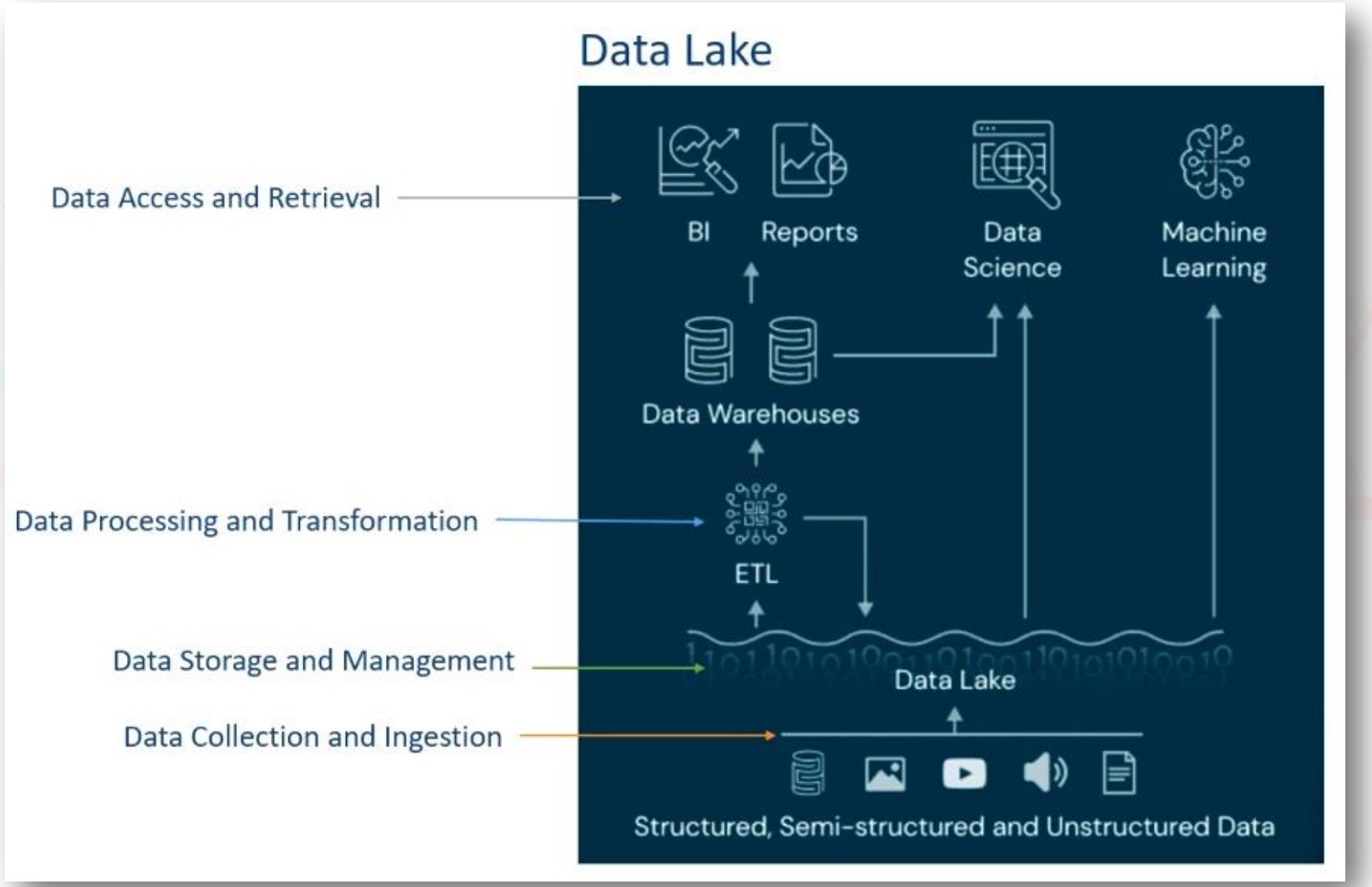
However, the Data Lake technology missed two supercritical features that Data warehouses offered:

1. Transaction and Consistency
2. Reporting Performance

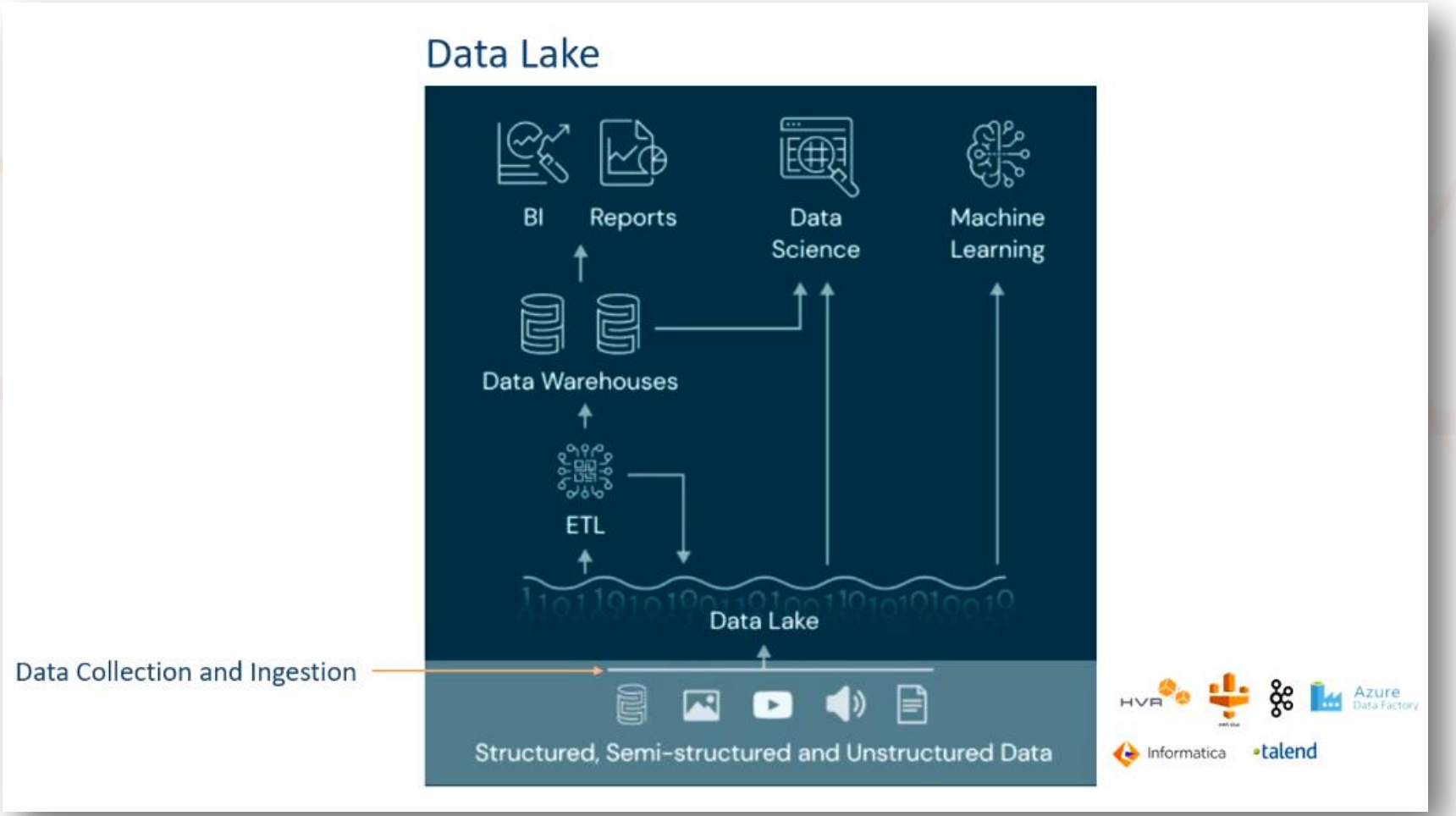
So, we adopted a different architecture for implementing Data Lakes. We collected data in Data lake storage, proceeded it using Apache Spark, and stored the result in a Data Warehouse. And finally, we connected the BI and Reporting with the Data Warehouse.



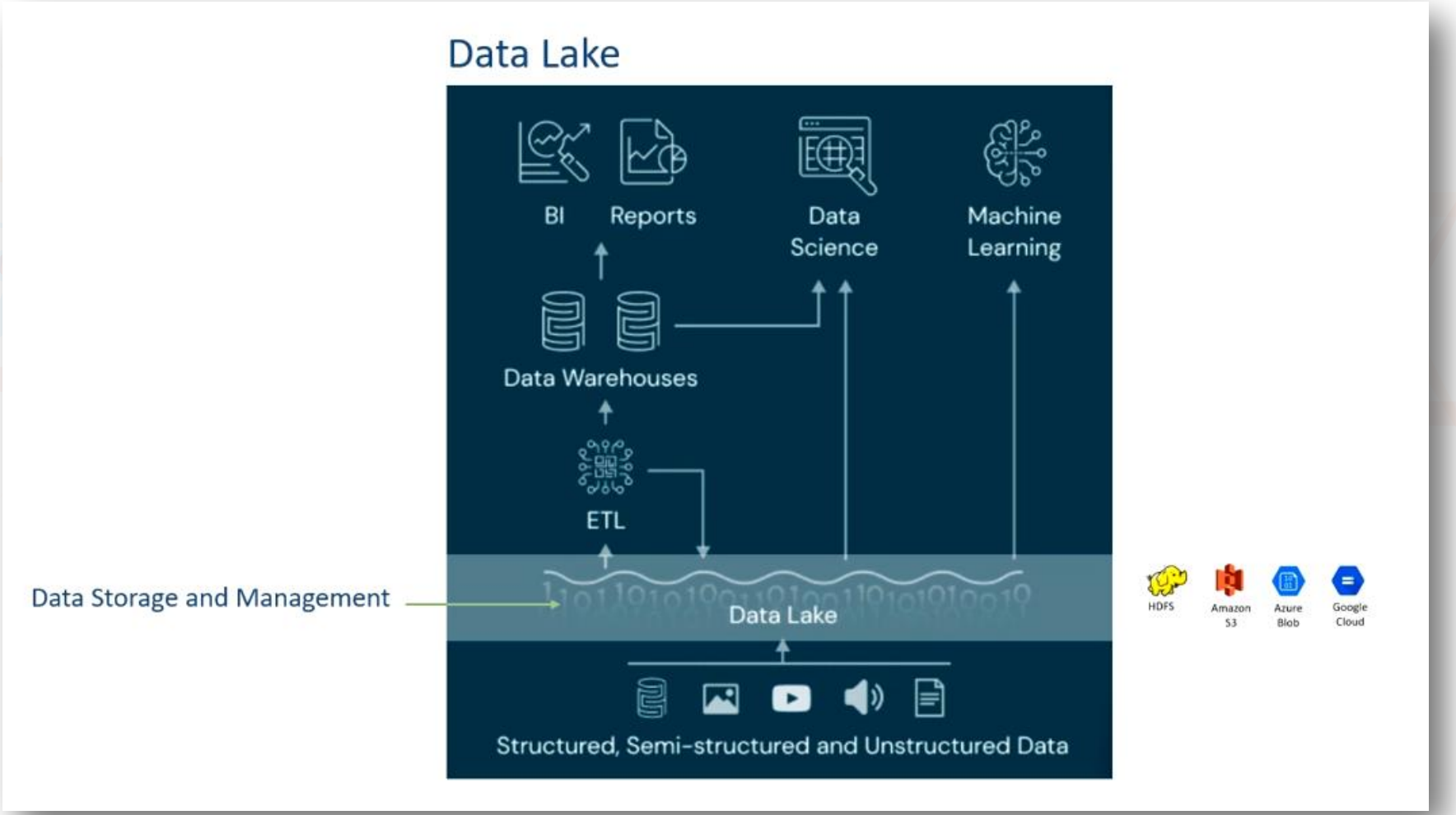
This new architecture also matured as a platform with four key capabilities. And these are nothing but the same problems that Google attacked and solved.



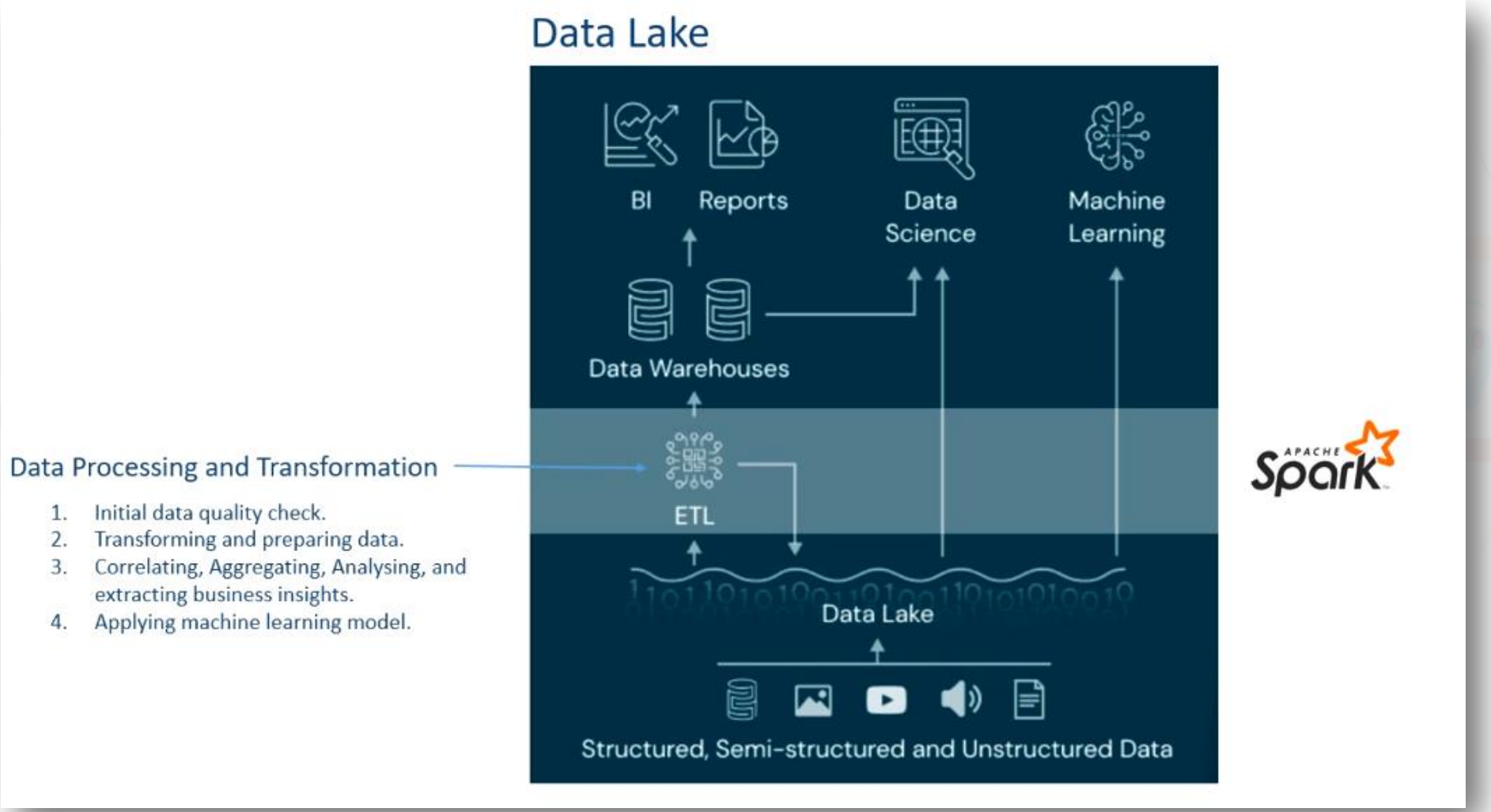
The ingest block of the data lake is all about identifying, implementing, and managing the right tools to bring data from the source systems to the data lake. And we have many vendors competing for a place in this box.



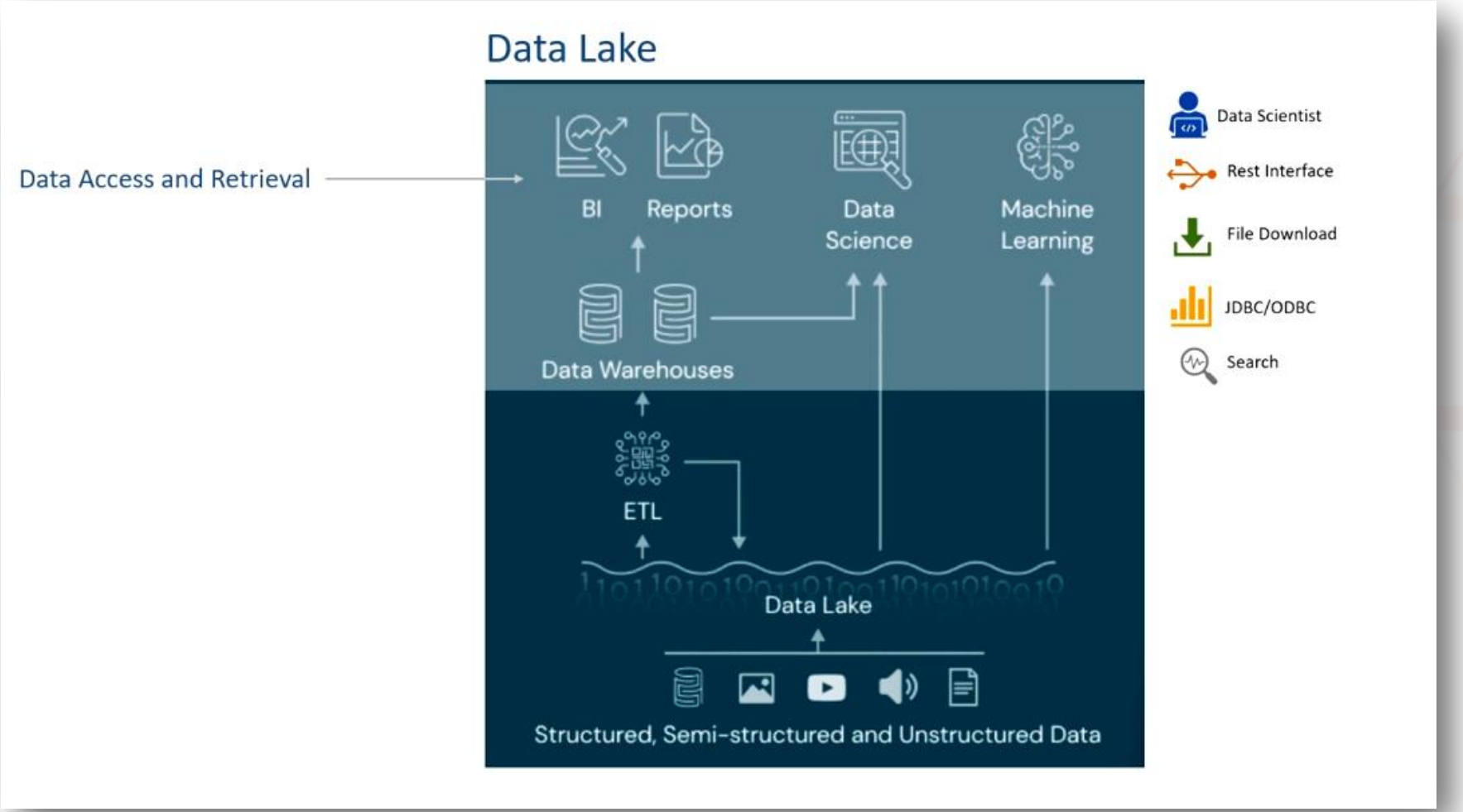
The core of the data lake platform is the storage infrastructure. In today's data lake, this could be an on-premise HDFS or Cloud Object Stores such as Amazon S3, Azure Blob, Azure Data Lake Storage, or Google Cloud Storage. Cloud storage is leading because they offer scalable and high availability access at an extremely low cost in almost no time to procure.



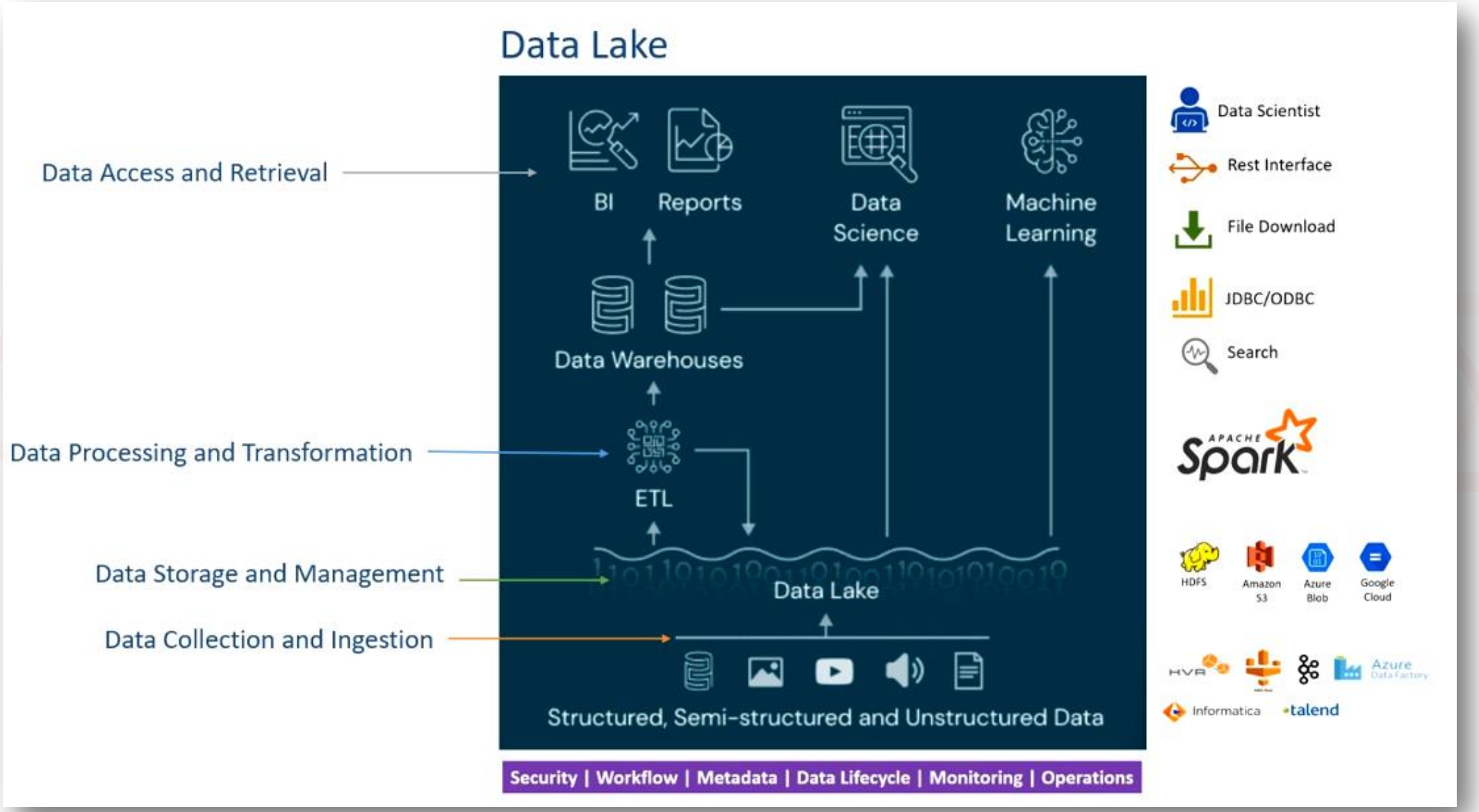
The next layer is the processing layer. This is the place where all the computation is going to happen. When I say computation, it means the following kind of work listed in the image below.



The last and most critical capability of the data lake is to allow you to consume the data from the data lake. You can think of your data lake as a repository of raw and processed data. Now the consumption is all about putting that data for real-life usage, which can be of various types.



This is the complete Data Lake architecture.



Thank You

ScholarNest Technologies Pvt Ltd.

www.scholarnest.com

ScholarNest

Spark Azure Databricks

Databricks Spark Certification and beyond

Instructor: Prashant Kumar Pandey

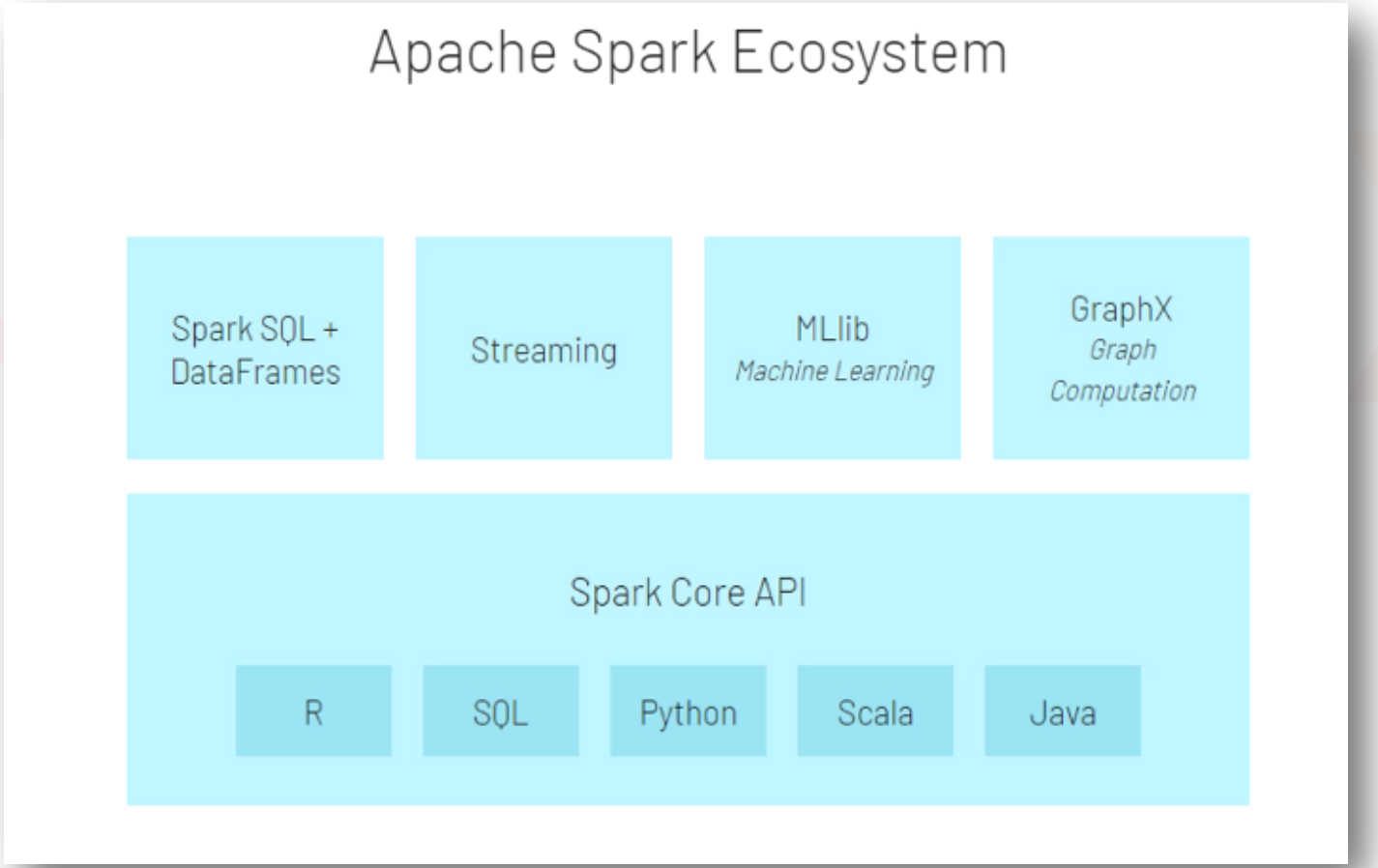
Absolute Beginner to Specialization in Apache Spark and Azure Databricks

Apache Spark and Databricks Cloud - Introduction

Apache Spark is a distributed data processing framework, and this diagram represents the Spark Ecosystem.

The Spark ecosystem is designed in two layers:

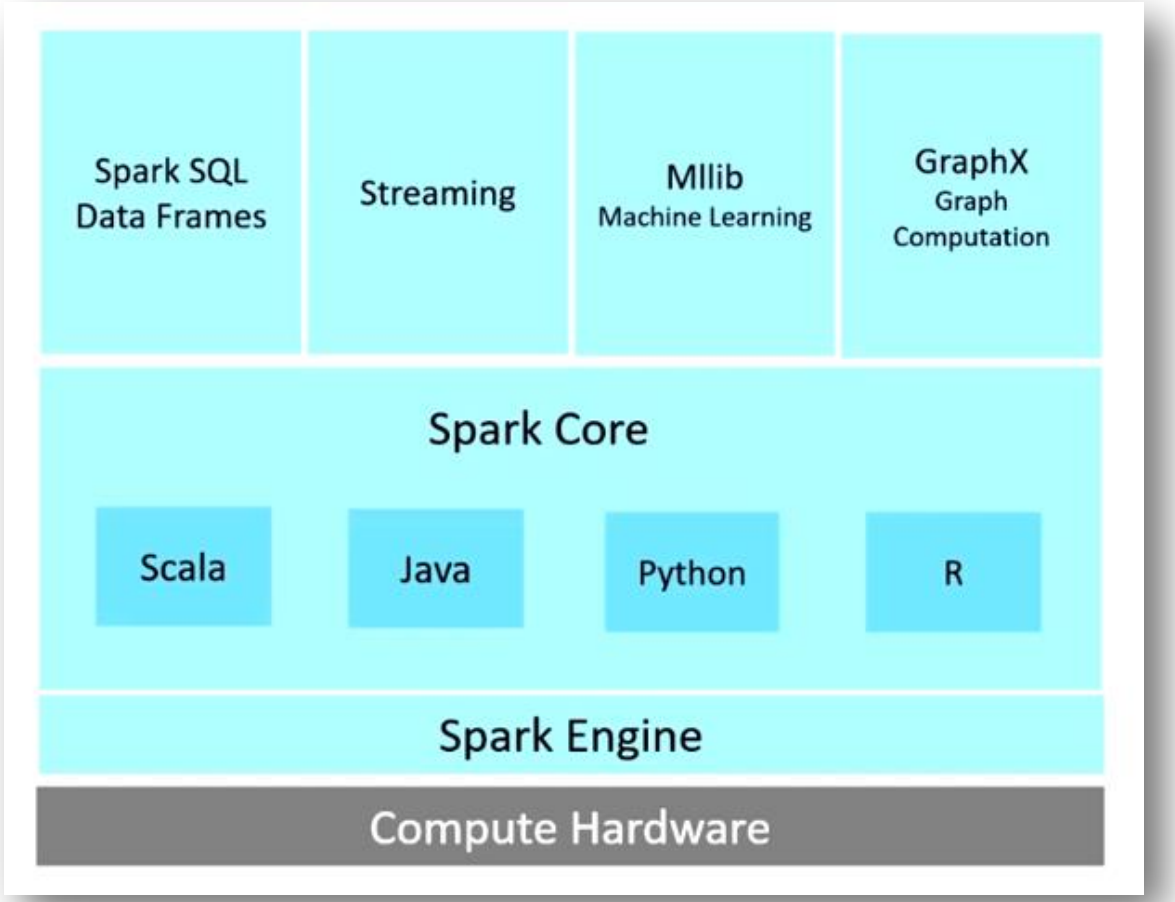
1. The top layer is a set of DSLs, libraries, and APIs.
2. The bottom layer is the Spark Core Layer.



The Spark core layer itself has got two parts:

1. A distributed Computing Engine
2. A set of Core APIs

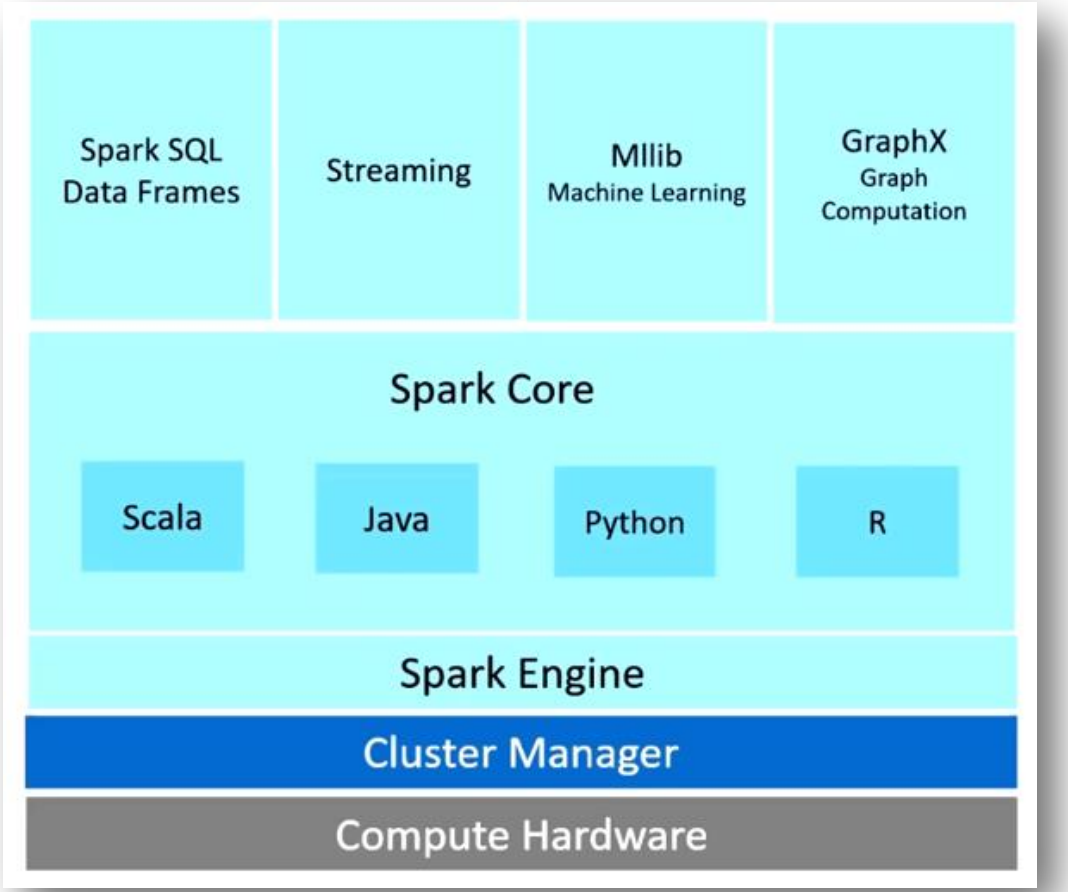
And this whole thing runs on a cluster of computers to offer you distributed data processing. However, Spark does not manage the cluster. It only gives you a data processing framework.



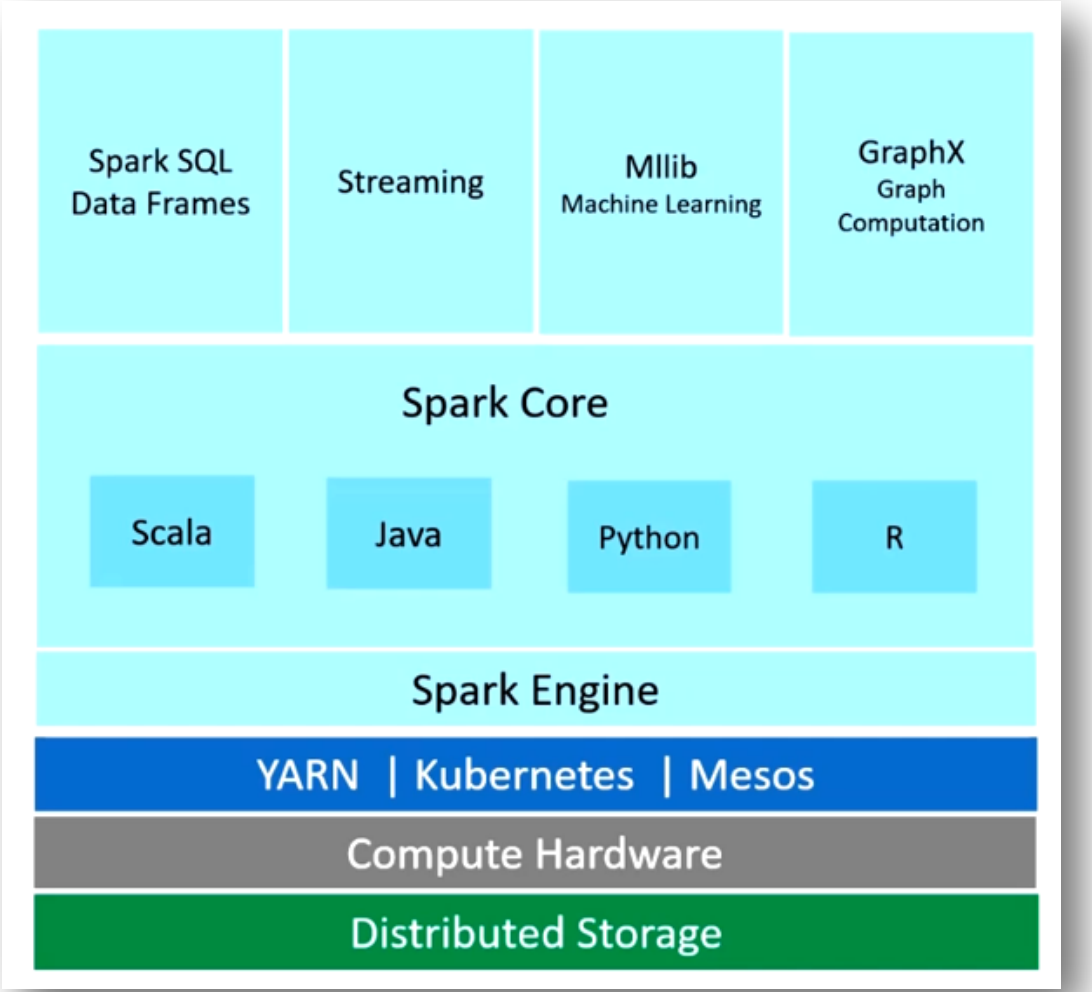
The Spark core layer itself has got two parts:

1. A distributed Computing Engine
2. A set of Core APIs

And this whole thing runs on a cluster of computers to offer you distributed data processing. However, Spark does not manage the cluster. It only gives you a data processing framework. Some examples of Cluster Manager are: YARN, Mesos, Kubernetes

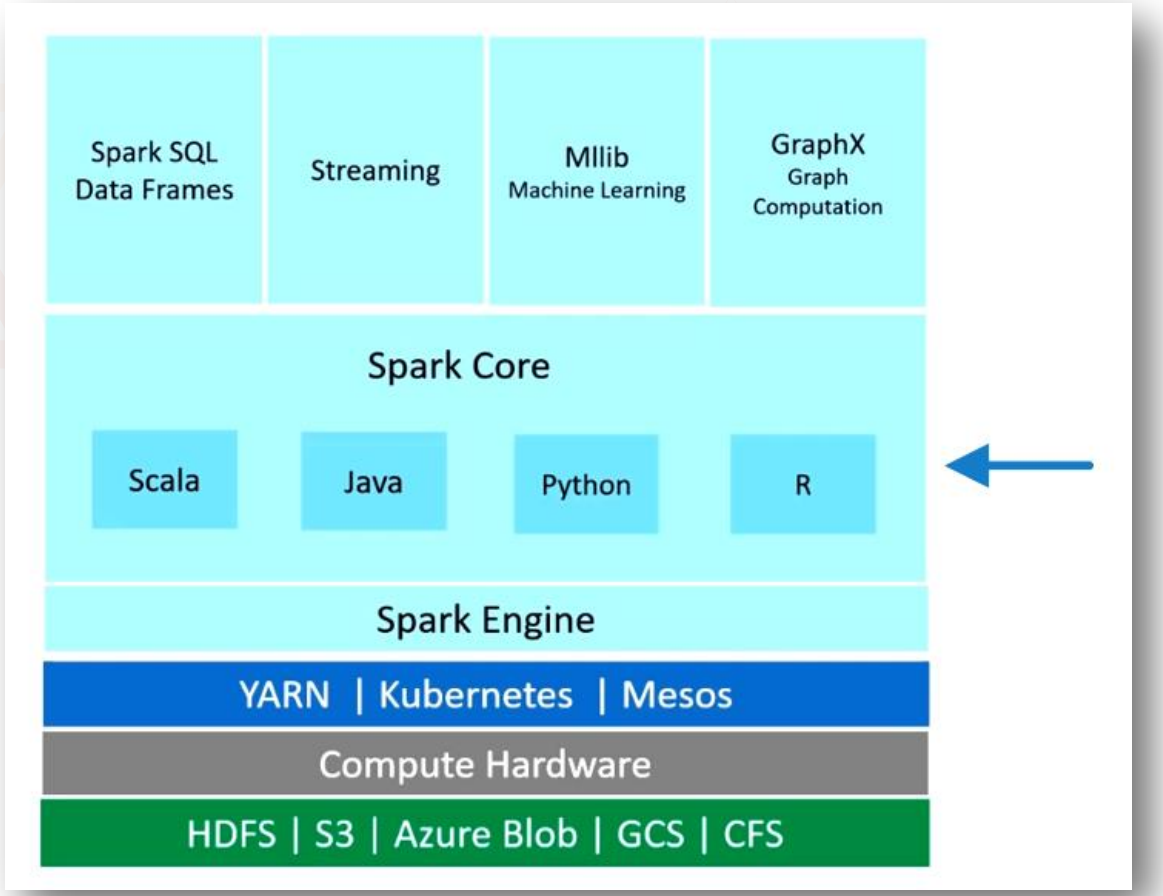


Spark also doesn't come with an in-built storage system.
And it allows you to process the data, which is stored in a variety of storage systems.
The most popular and commonly used storage systems are HDFS, Amazon S3, Azure Data Lase Storage, Google Cloud Storage, and the Cassandra file system.

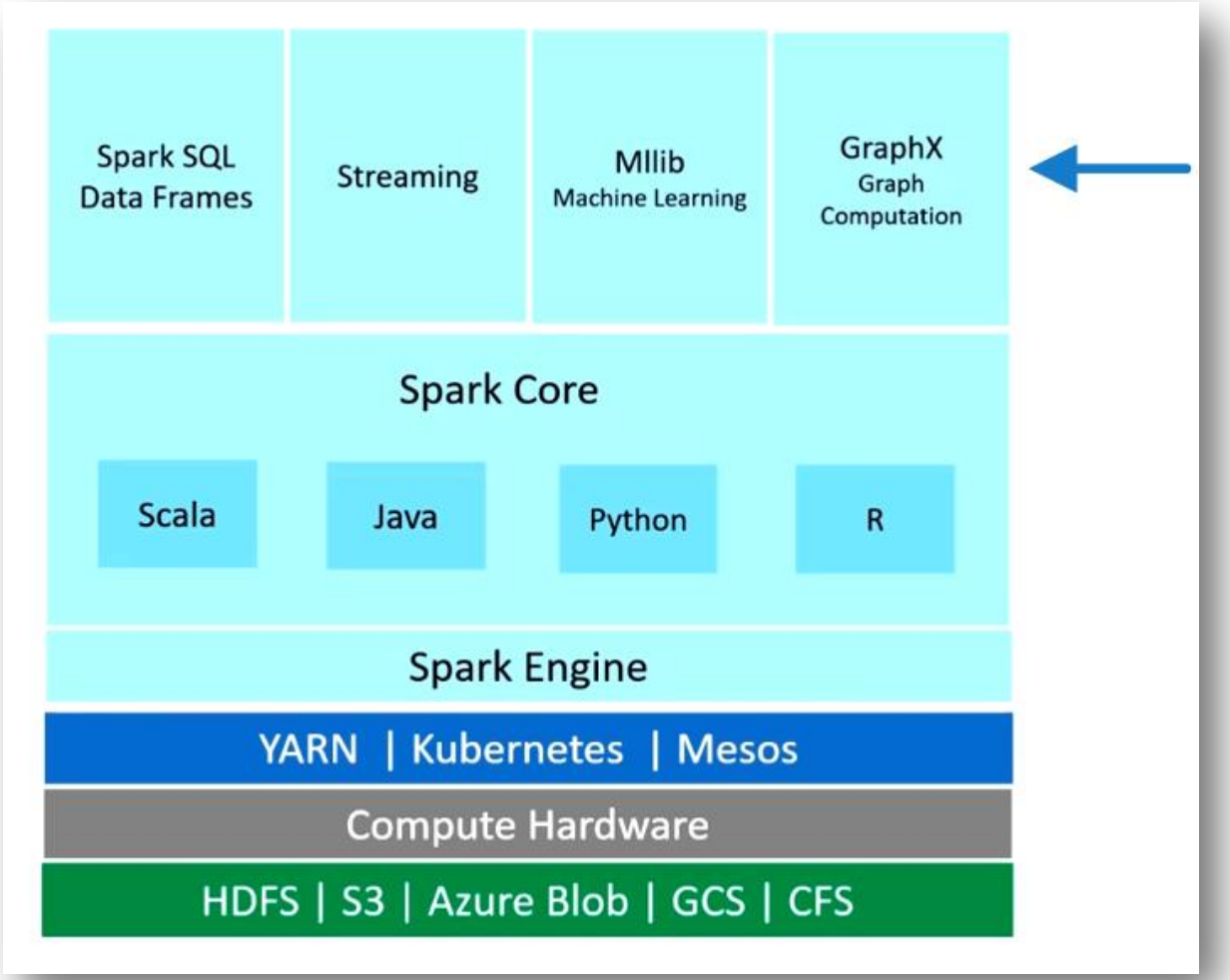


Apache Spark does not offer Cluster Management and Storage Management.
All you can do with Apache Spark is to run your data processing workload.
And that part is managed by the Spark Compute Engine.
So the Spark compute engine is the core that runs and manages your data processing work and provides you with a seamless experience.
All you need to do is submit your data processing jobs to Spark, and the Spark core will take care of everything else.

The Core API layer is the programming interface layer that offers you the core APIs in four major languages: Scala, Java, Python, and R. These are the APIs that we used to write data processing logic during the initial days of Apache Spark. However, these APIs were based on resilient distributed datasets (RDD). These APIs are the most complicated way to work with Apache Spark and they also lack some performance optimization features. However, these APIs can offer you the highest level of flexibility to solve some complex data processing problems.



The topmost layer is the prime area of interest for most Spark developers and data scientists. It is a set of libraries, packages, APIs, and DSL. These are developed by the Spark community over and above the Core APIs. The topmost API layer is grouped into four categories to support four different data processing requirements. However, this is just a logical grouping, and there is no rigid boundary.



Let us look at the four categories of the top most API layer:

1. Spark SQL and DataFrame/Dataset APIs - Spark SQL allows you to use SQL queries to process your data. Both of these together can help you resolve most of the structured and semi-structured data crunching problems.
2. Spark Streaming libraries - They allow you to process a continuous and unbounded stream of data.
3. A set of libraries specifically designed to meet your machine learning, deep learning, and AI requirements.
4. Graph Processing libraries, and they allow you to implement Graph Processing Algorithms using Apache Spark.

So the topmost layer is nothing but a set of libraries and DSLs to help you solve your data crunching problems.
And all these are available in multiple languages such as Java, Scala, Python, and R.

There are three reasons why Spark Ecosystem is so popular:

1. Abstract - Spark will abstract away that you are coding to execute your program on a cluster of computers. So, all the complexities of distributed storage, computation, and parallel programming, is abstracted away by the Spark core.
2. Unified Platform - Spark combines the capability of SQL queries, Batch Processing, Stream Processing, Structured and semi-structured data handling, Graph processing, machine learning, and deep learning. All of this in a single framework using your favourite programming language. You can mix and match them to solve many sophisticated requirements.
3. Ease of Use - Comparing it with old Hadoop and MapReduce code, Spark code is much shorter, simpler, and easy to read and understand.

Now let’s come to Databricks. Apache Spark is an open-source project.
The original creators of Apache Spark donated it to Apache Foundation and made it an Open Source project. However, that same team formed a company and a commercial product around Apache Spark. The company and the product are both named Databricks.
And Databricks offers the following things over and above Apache Spark:

1. Spark on Cloud Platform
2. Spark Cluster Management
3. Notebooks and Workspace
4. Administration Controls
5. Optimized Spark
6. Databases/Tables and Catalog
7. Databricks SQL Analytics
8. Delta Lake Integration
9. ML Flow
10. Industry vertical solutions and accelerators

Thank You

ScholarNest Technologies Pvt Ltd.

www.scholarnest.com

ScholarNest

Spark Azure Databricks

Databricks Spark Certification and beyond

Instructor: Prashant Kumar Pandey

Absolute Beginner to Specialization in Apache Spark and Azure Databricks

Spark Development Environments

Spark projects are developed and deployed in two kinds of environments.

Cloud Platforms

On-premise Platform

We have two standard methods to set up your Spark development environment

Notebook

Python IDE

If your project is going in Cloud, you should prefer Notebook. And an on-premise project prefers to use Python IDE. However, it is not mandatory. Cloud platforms also allow you to develop on your local machine and deploy it in Cloud.

Notebook

Python IDE

The first and the easiest method to get Spark is to use Spark Notebooks in the Databricks Cloud environment. However, you should also learn to set up a Local Spark development Environment.

Notebook

Python IDE

Thank You

ScholarNest Technologies Pvt Ltd.

www.scholarnest.com

ScholarNest

Spark Azure Databricks

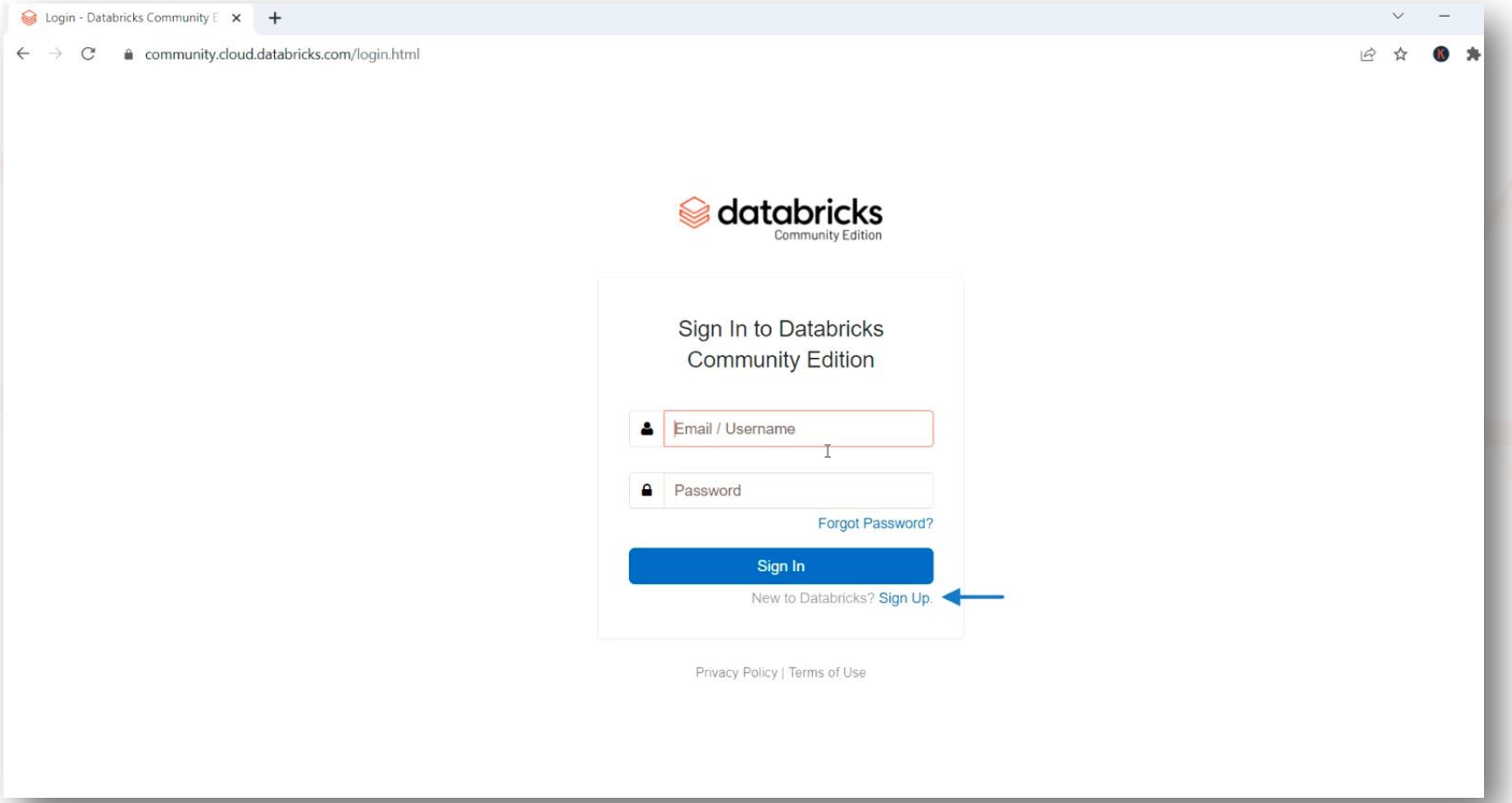
Databricks Spark Certification and beyond

Instructor: Prashant Kumar Pandey

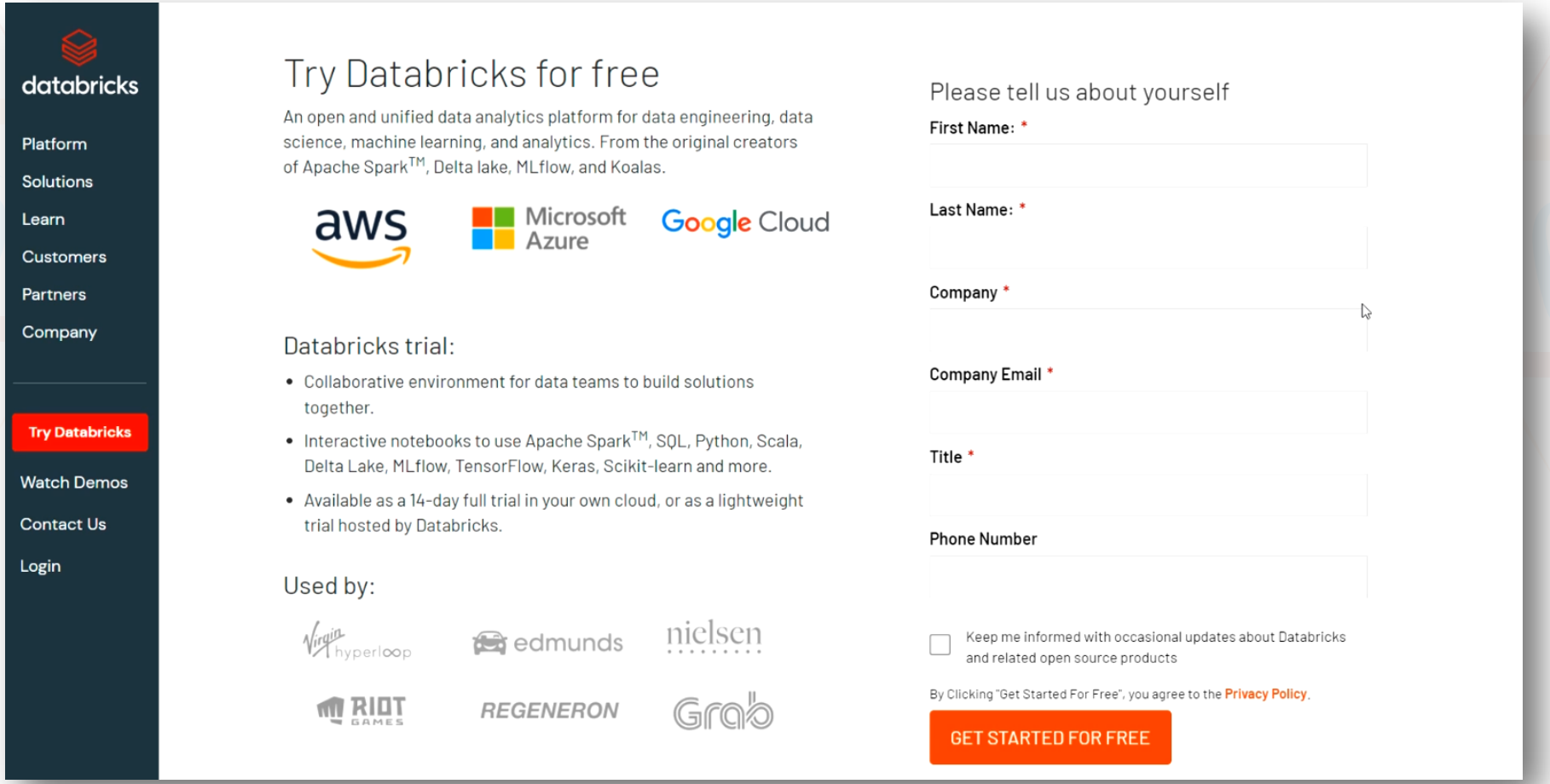
Absolute Beginner to Specialization in Apache Spark and Azure Databricks

Databricks Community Account

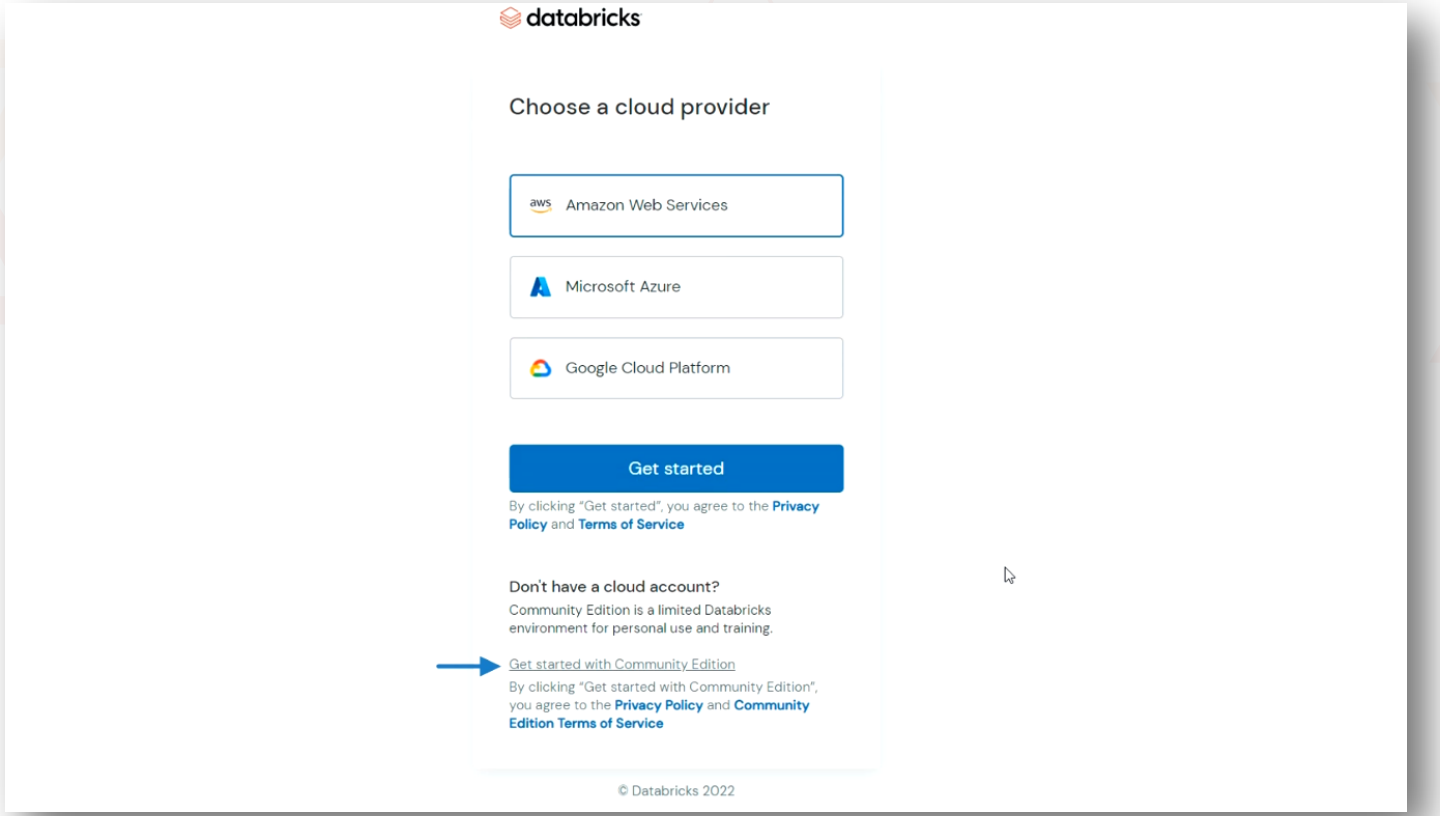
Visit https://community.cloud.databricks.com/login.html
Click the Sign-Up Button



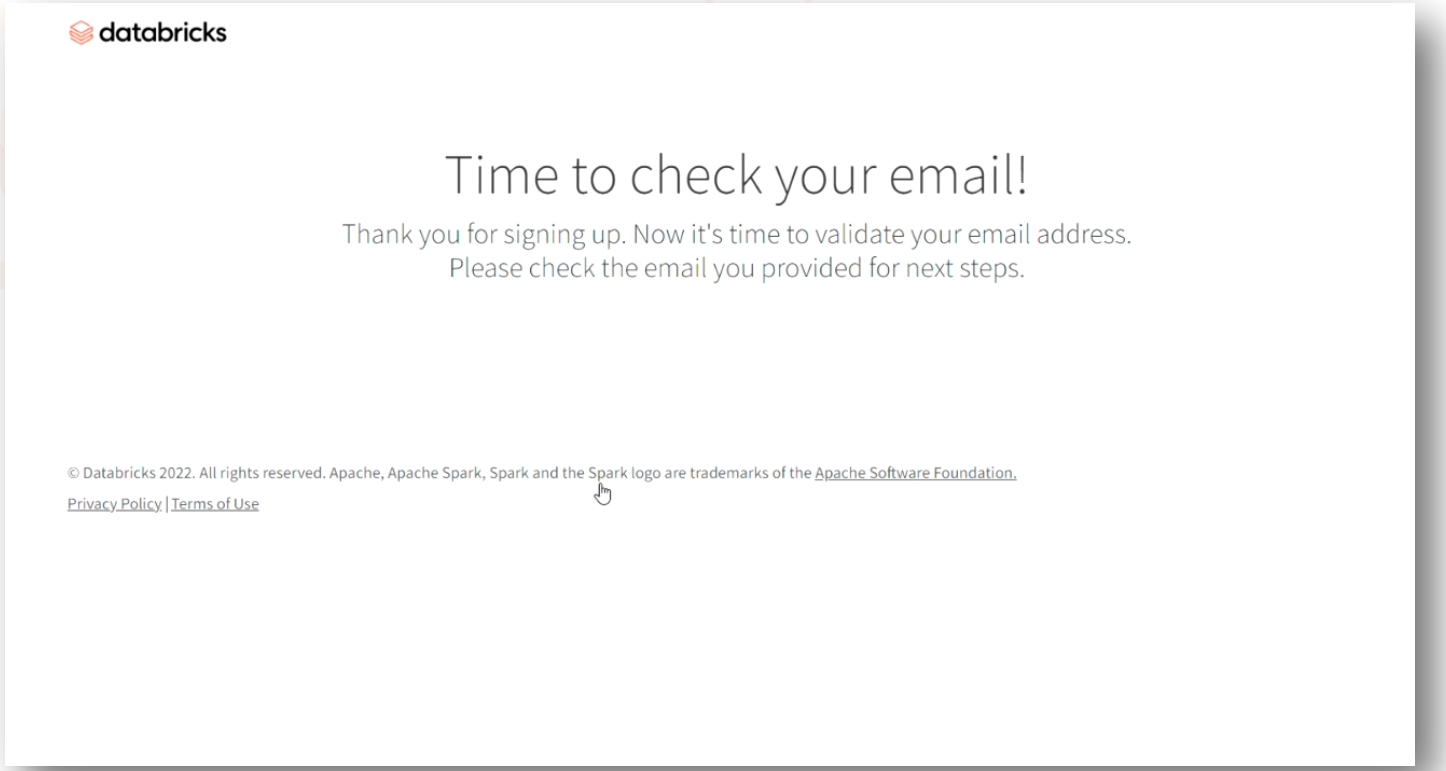
Databricks Cloud is available in AWS, Microsoft Azure, and Google cloud platforms. If you already have an account in any of these cloud platforms, you can get 14 day free trial. However, we will be using a completely free lightweight community version of the Databricks Cloud. But you should register for the account creation. Fill out the details below and click the get started for free button.



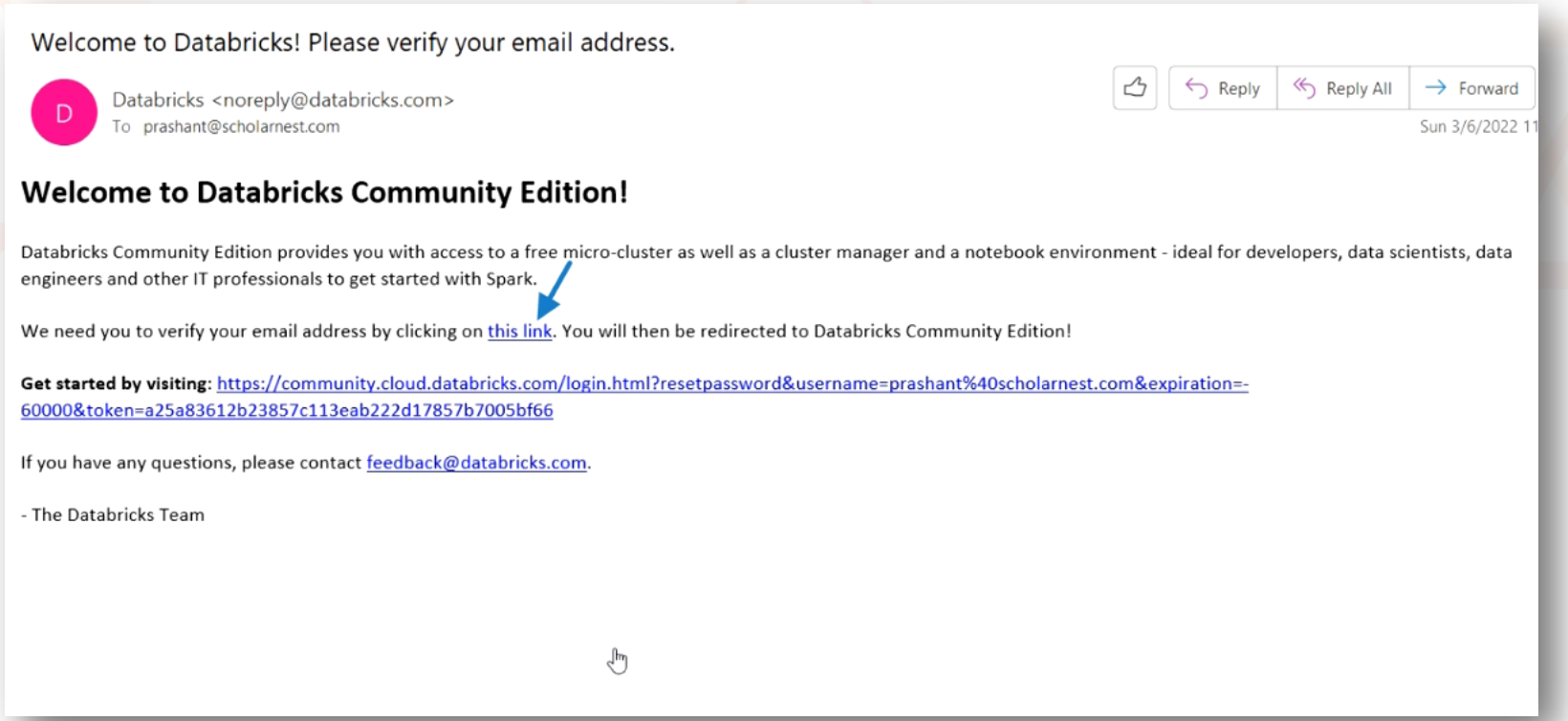
They are asking to choose the cloud environment. We will not choose any of these three cloud environments because that option is for availing of 14 day free trial. We wanted to sign up for a free community edition. So you should go down and click the community edition link.



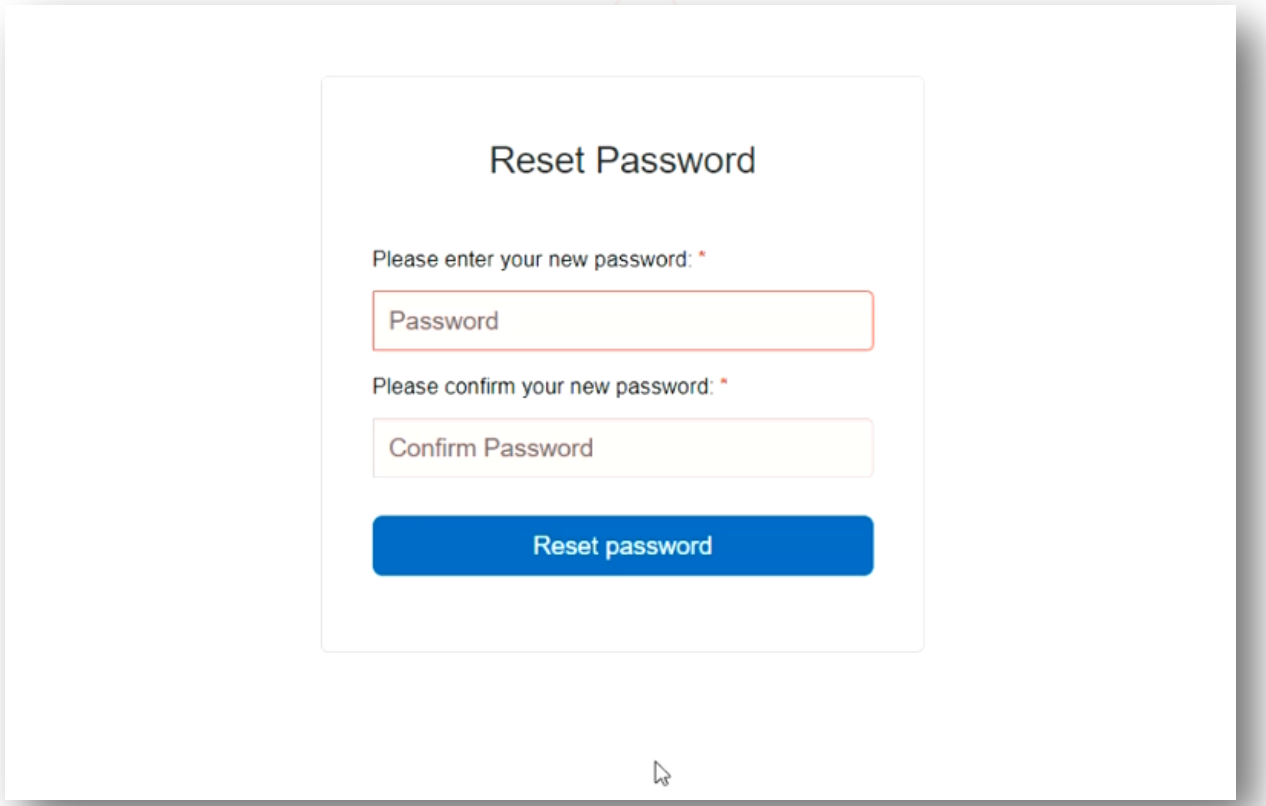
They might ask for captcha verification, so go ahead and verify that you are a human. Once your captcha verification is complete, they will send you an email.



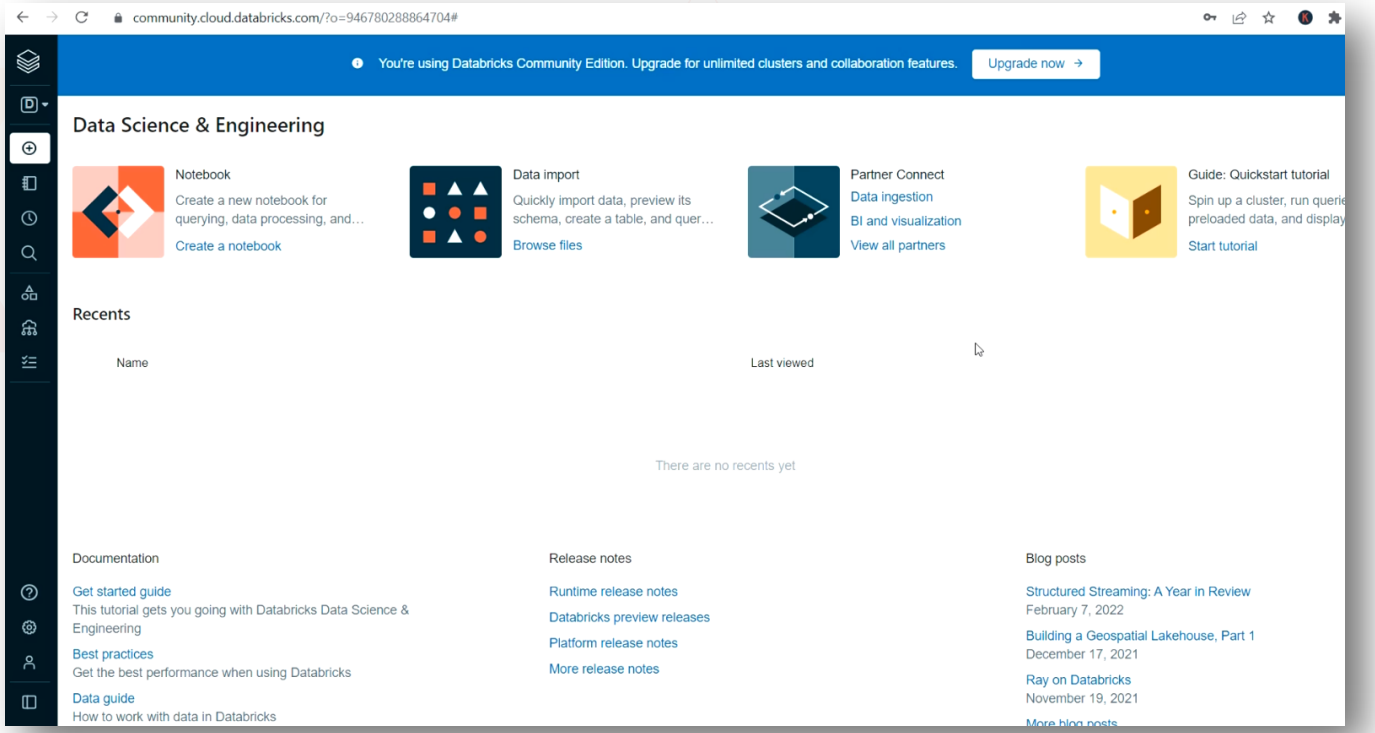
Login to your email box and check the email from Databricks. Below is an example email. You should expect a similar email from Databricks. Make sure you are checking your spam folder and found this email. Once you find the email, click the link provided in the email.



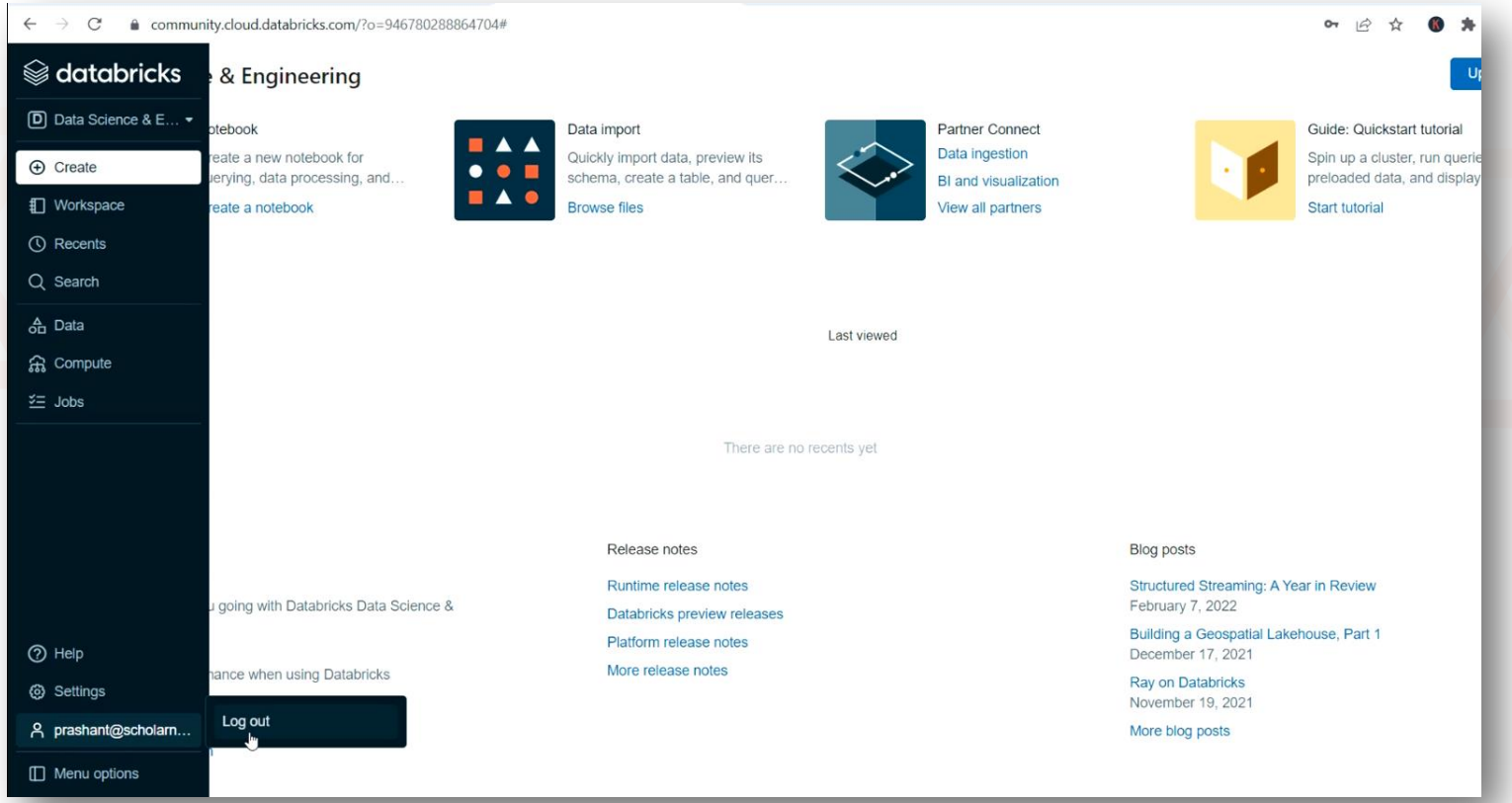
The confirmation link takes you to the password reset page. Set your password and confirm it.



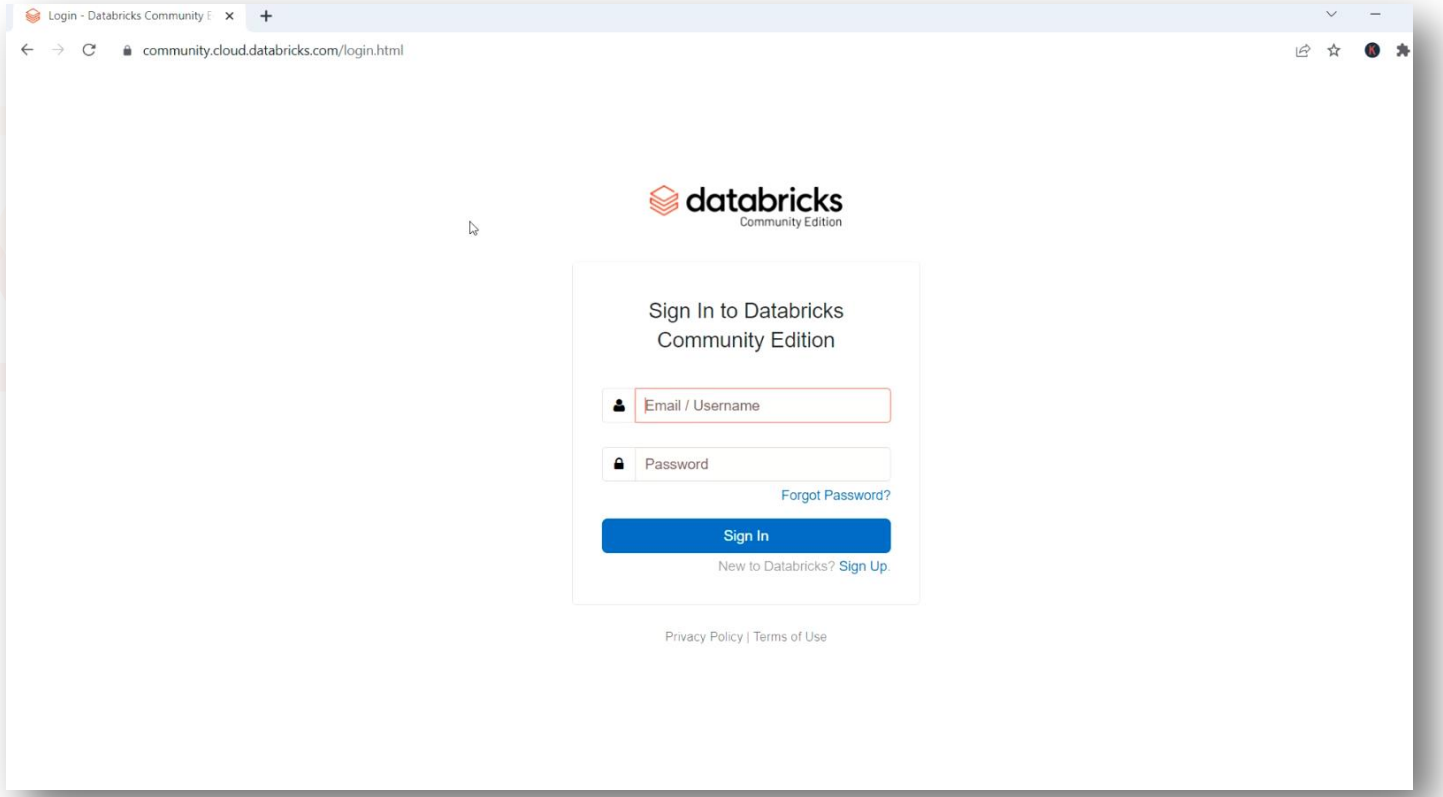
The password reset will take you inside the Databricks Cloud Workspace. Below page is the Databricks Cloud Workspace home page.



You can log out from the workspace and close everything.



You can visit https://community.cloud.databricks.com/login.html
And login using your credentials. I recommend that you bookmark this page.



Thank You

ScholarNest Technologies Pvt Ltd.

www.scholarnest.com

ScholarNest

Spark Azure Databricks

Databricks Spark Certification and beyond

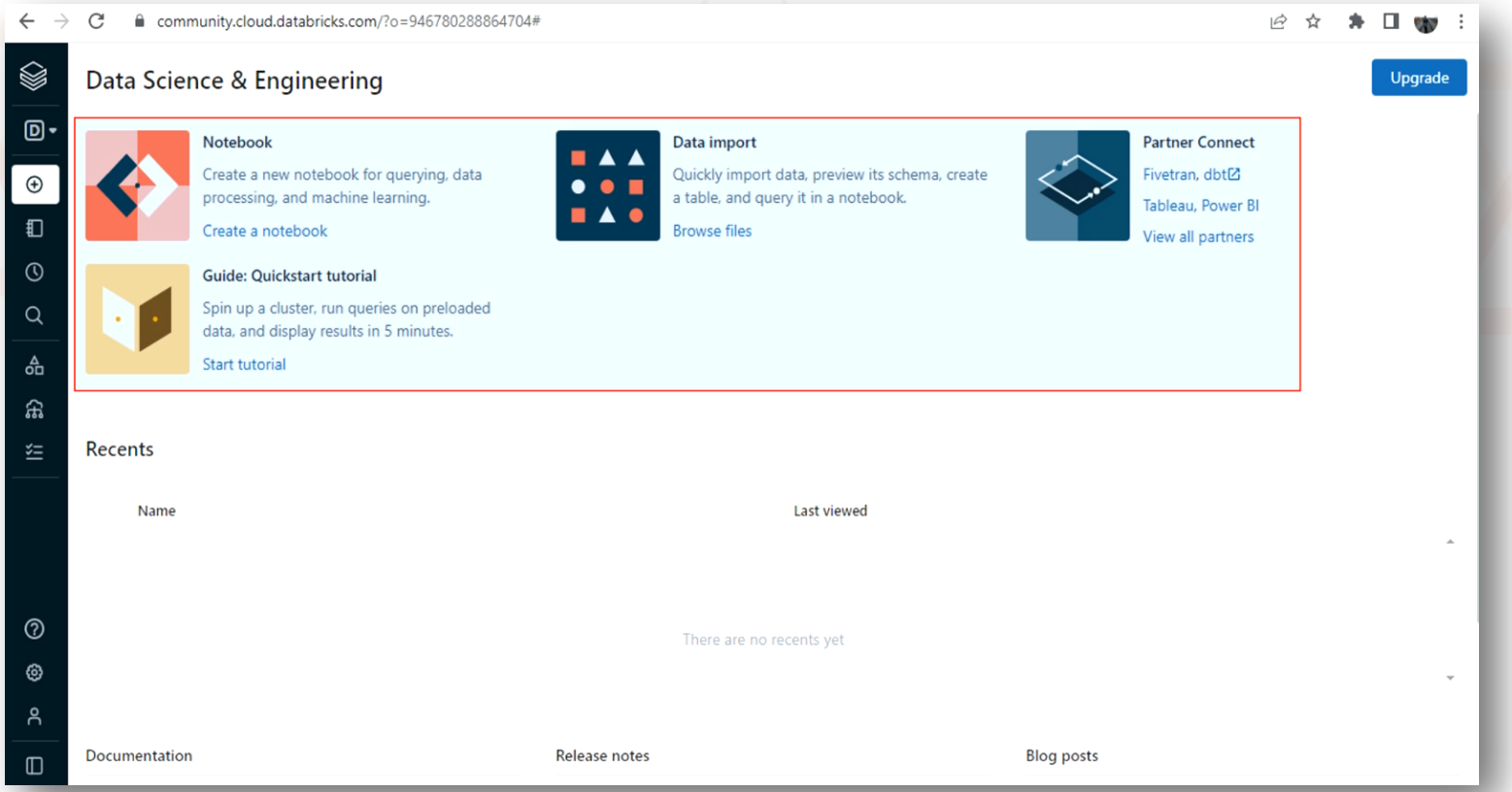
Instructor: Prashant Kumar Pandey

Absolute Beginner to Specialization in Apache Spark and Azure Databricks

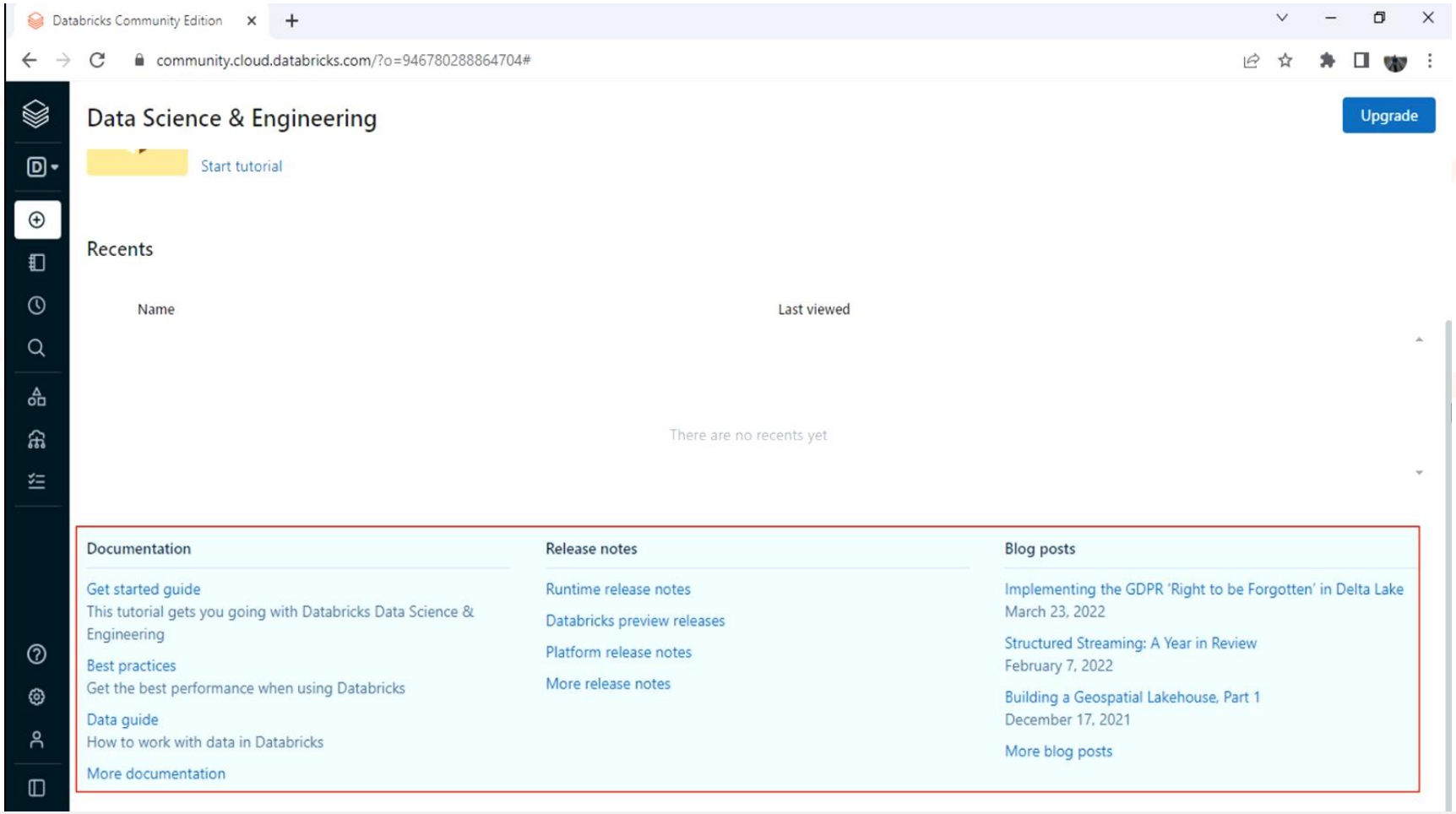
Databricks Workspace Introduction

The below screenshot shows Databricks Workspace home page. In the center of the page, you have some shortcuts for the most common tasks.

1. The first link is to create a new notebook and start writing some code.
2. The second link is to import data files.
3. The next one allows you to use some partner tool.
4. The last link is for a quick start tutorial.

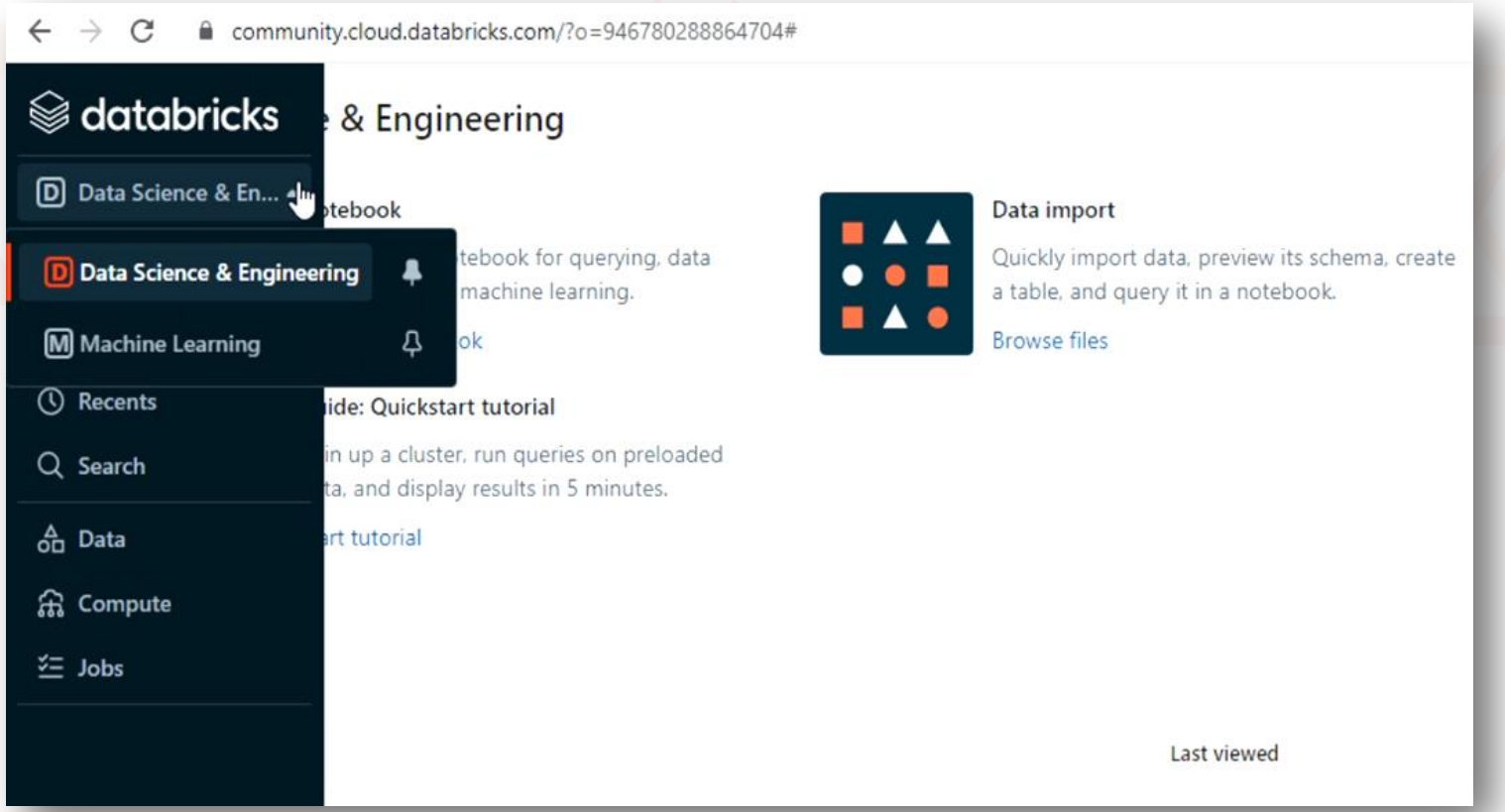


You also have some links at the bottom. These are mainly Databricks documentation and blog posts.

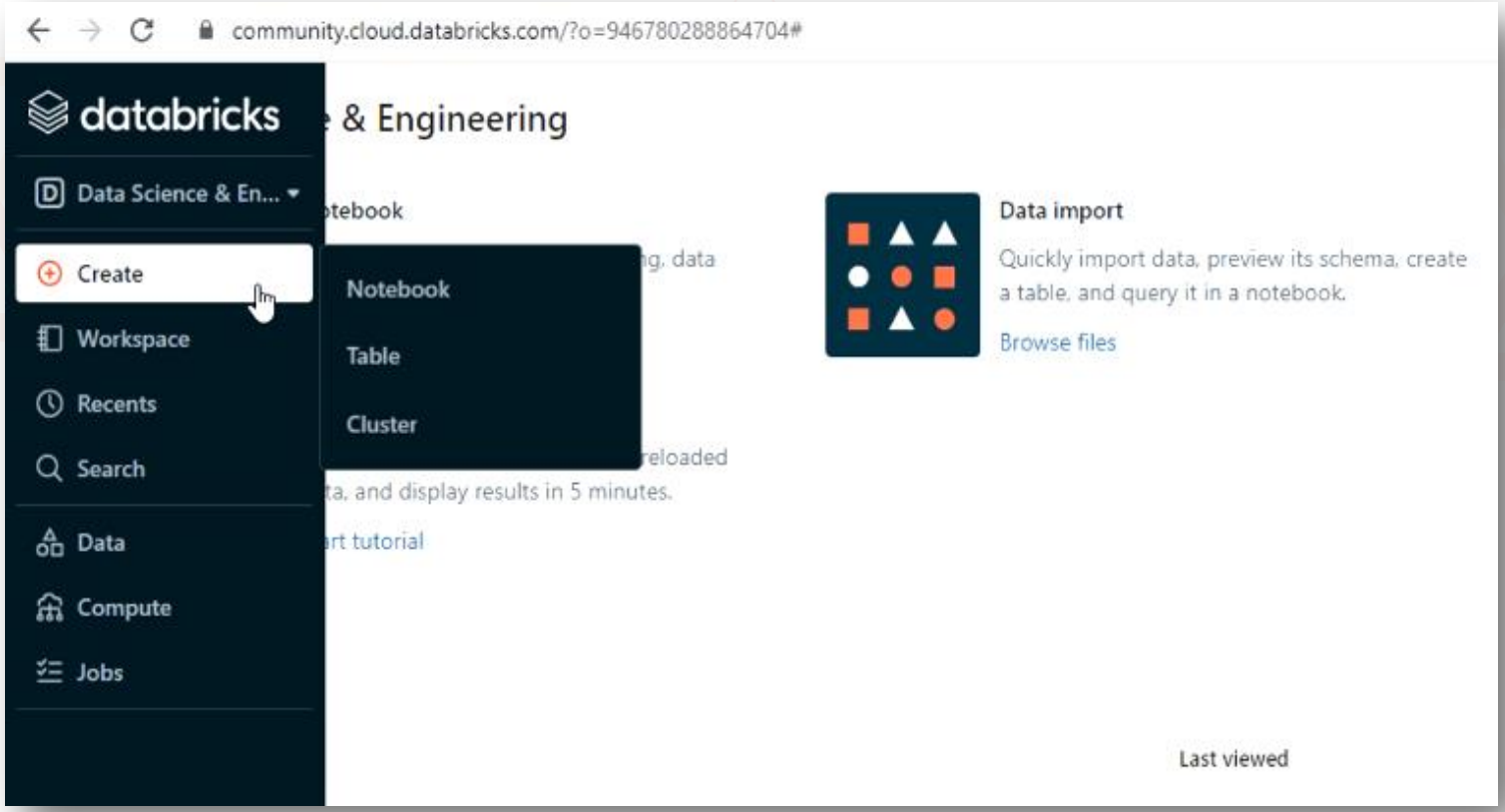


The main functionality is hidden behind the left side menu.
The first item in the navigation menu allows you to choose workspace type. You can see two workspace types.

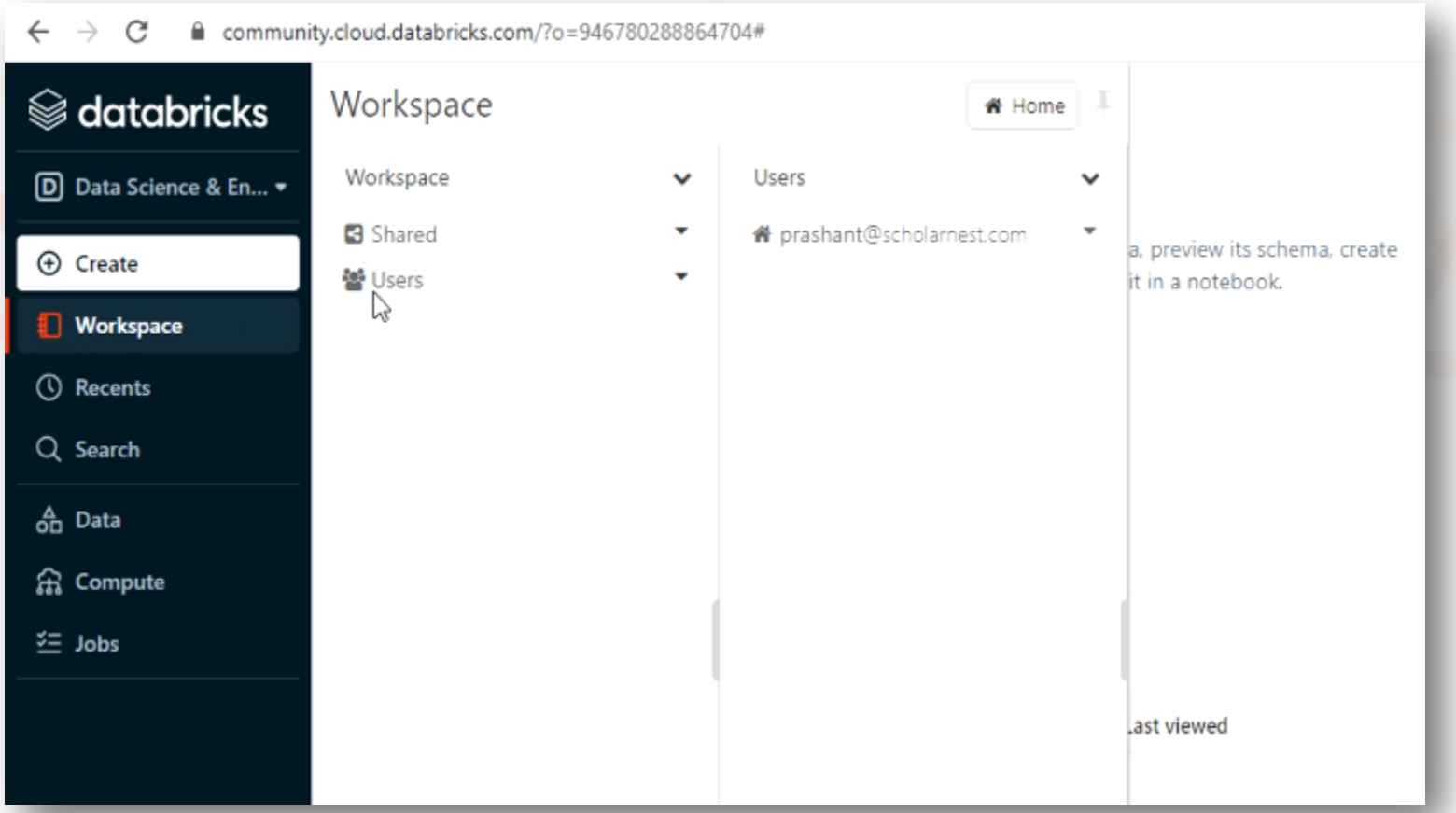
1. The first one is Data Science and engineering
2. The second workspace type is for machine learning professionals.



Up next, you will see the create button on the menu.
When you click on this button, it will show you options to create a Notebook, Table, and Cluster. We create and use these three main things while working with the Databricks community.

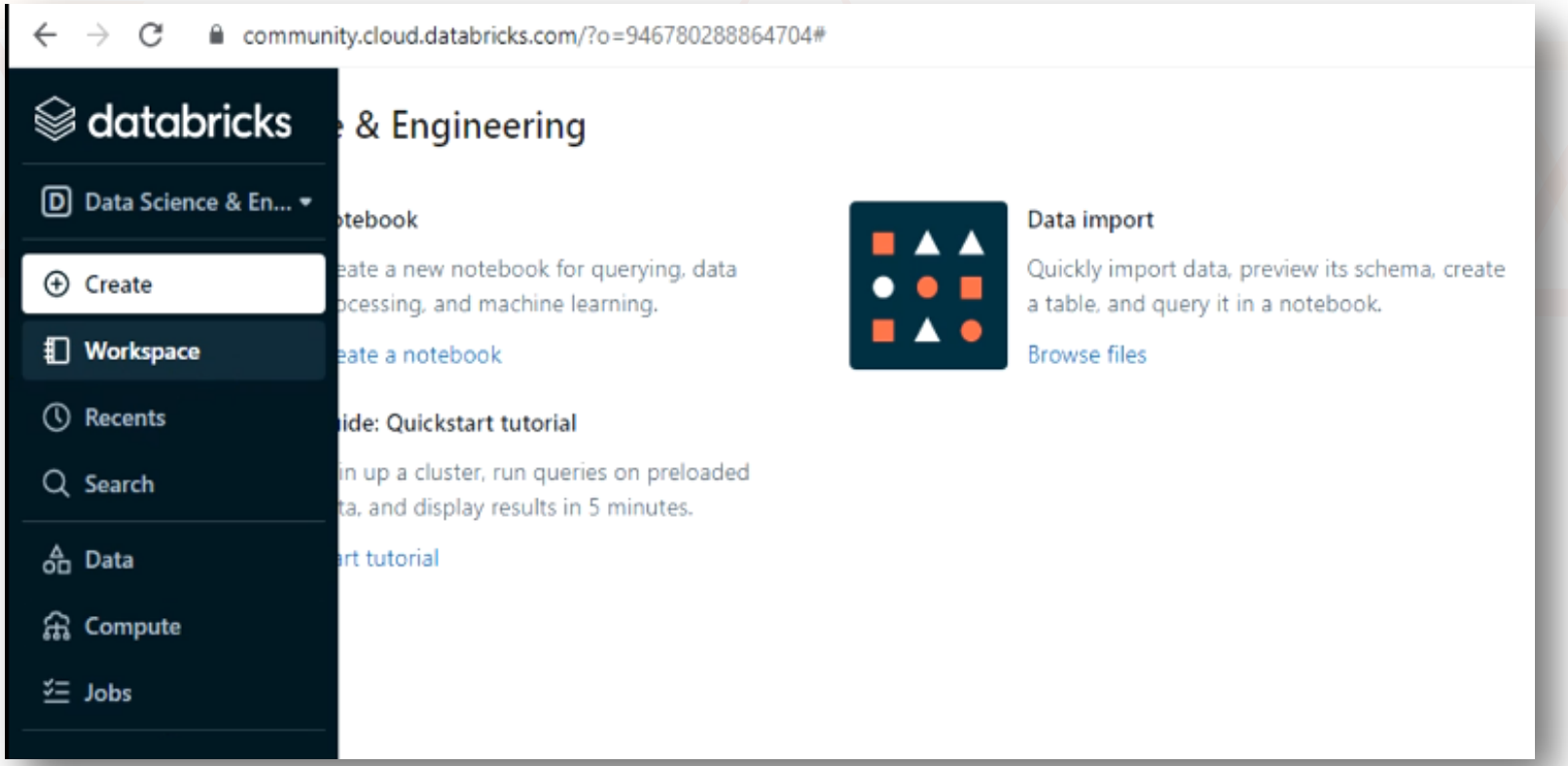


The following item in your menu option will show your workspace.
So, this workspace here is a set of folders and files assigned to each user.
We only have one user here, so the workspace shown below is my workspace.

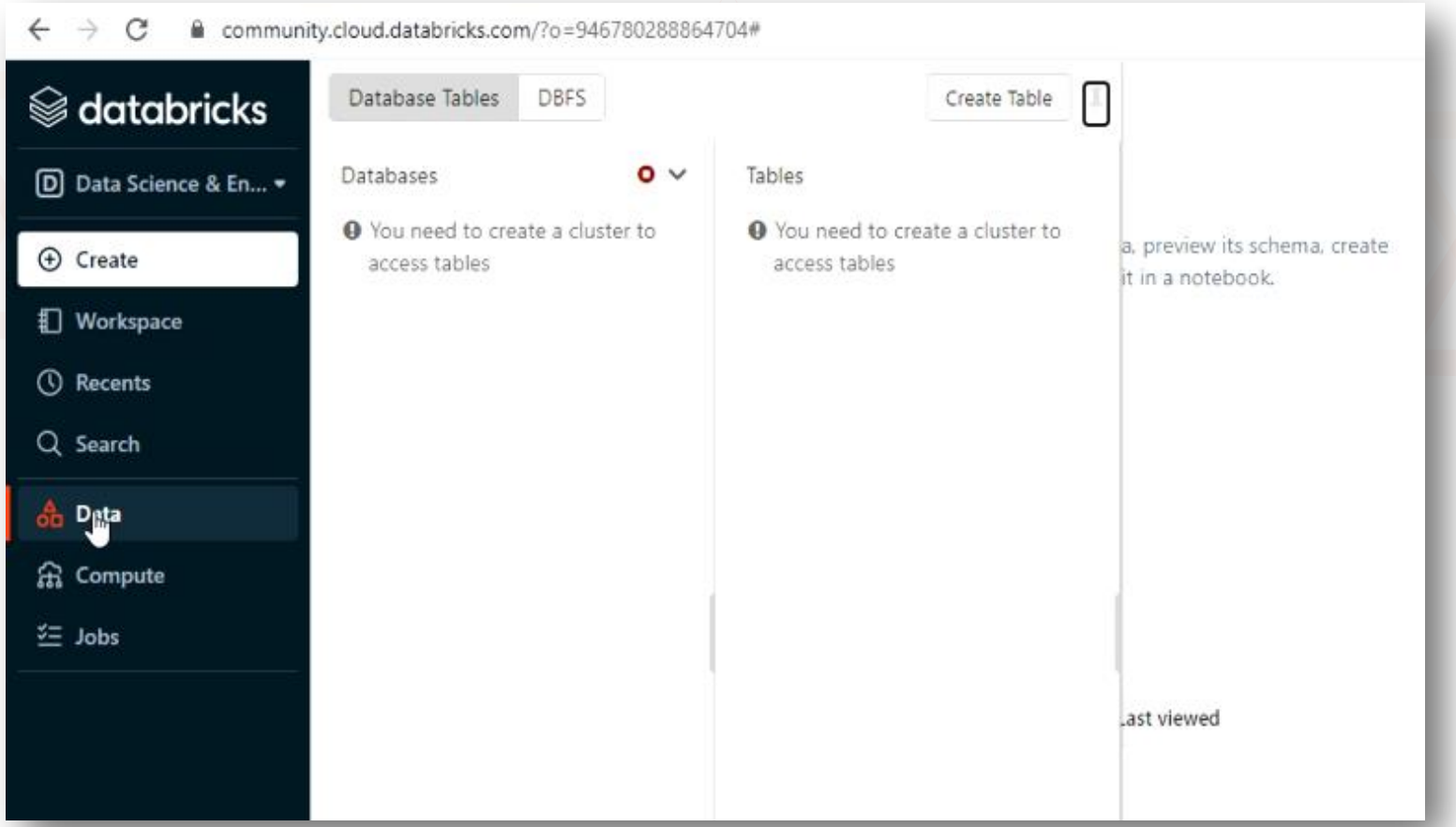


The next two items in your menu options are:

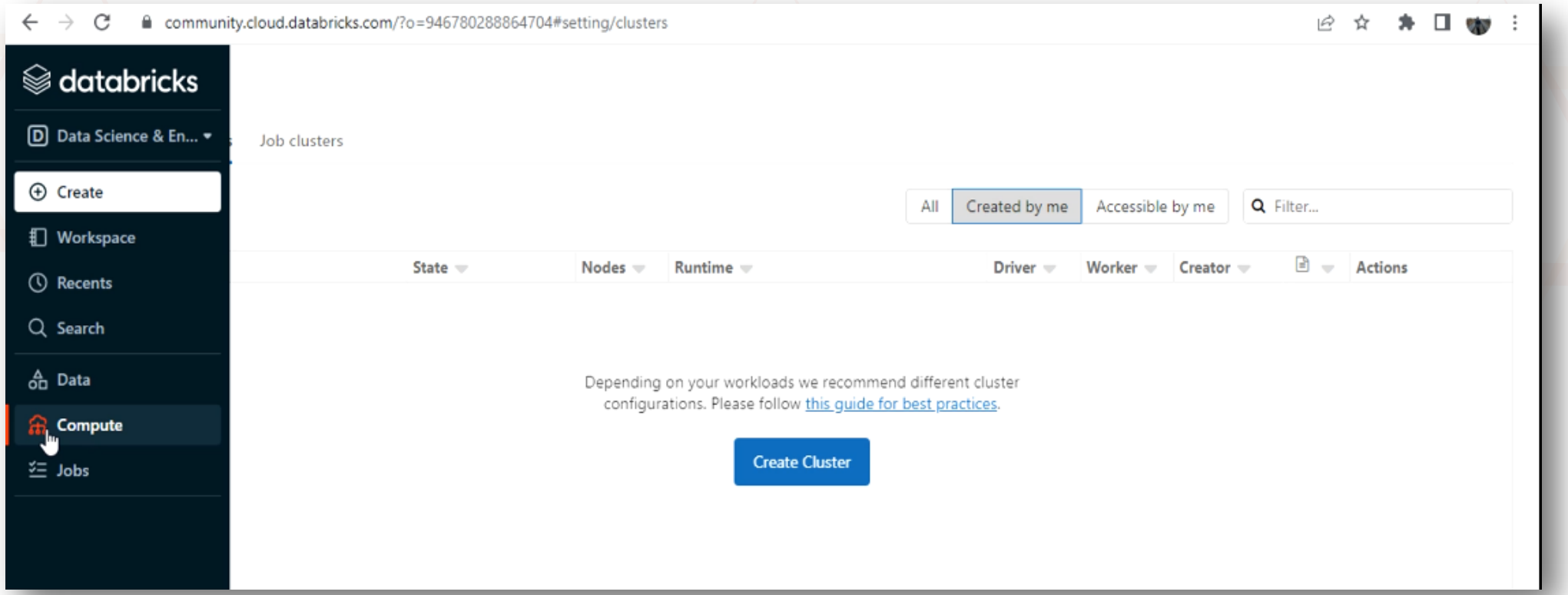
1. Recents – It shows a list of recent items used in the Databricks Workspace.
2. Search - It allows you to do a full-text search in your Databricks workspace.



The next item is for data. It allows you to create Spark Databases and Tables. Even if you create databases and tables using your program, they will appear here.

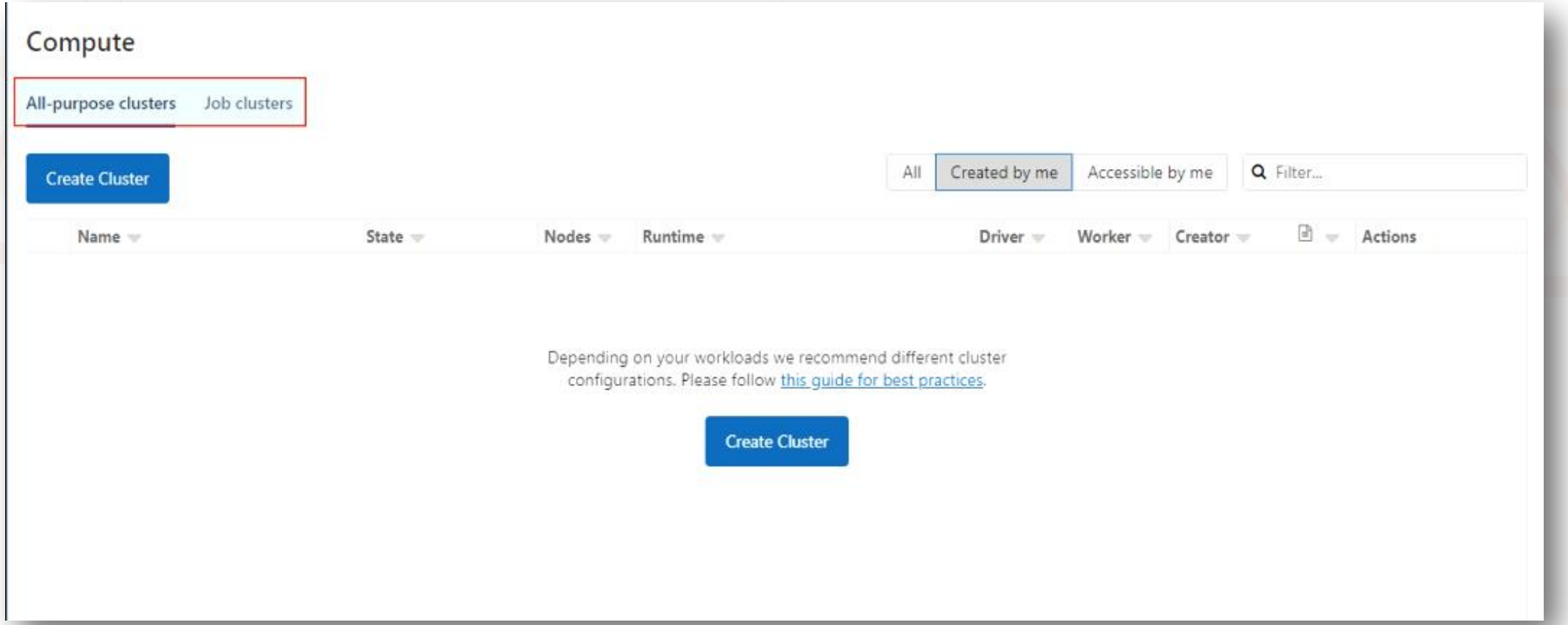


The following item is the Compute menu. This page allows you to create, launch and manage spark clusters. Creating a database and table requires some computation power. So you can create them only when you have a Spark cluster.

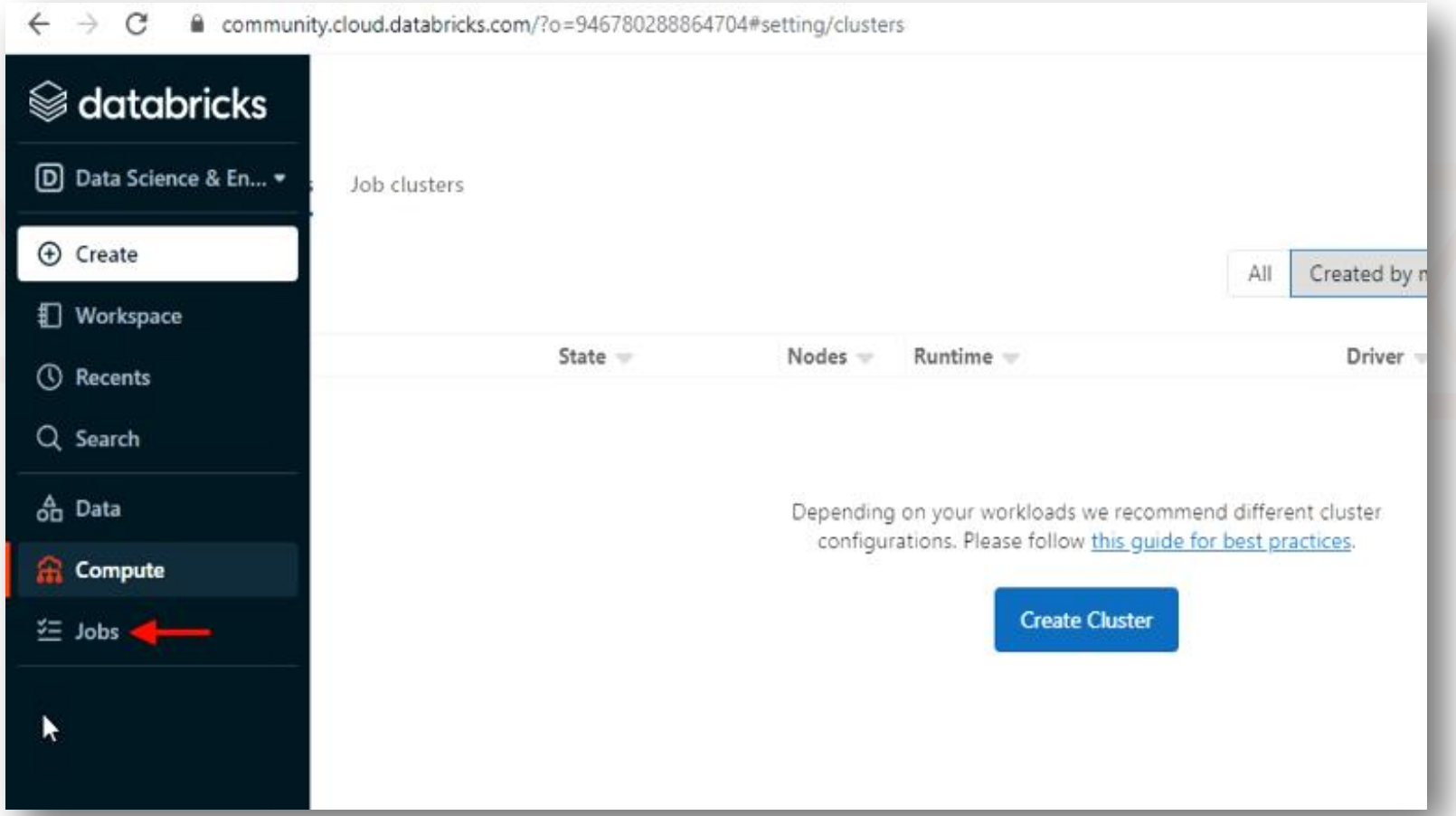


You can create two types of clusters in compute menu:

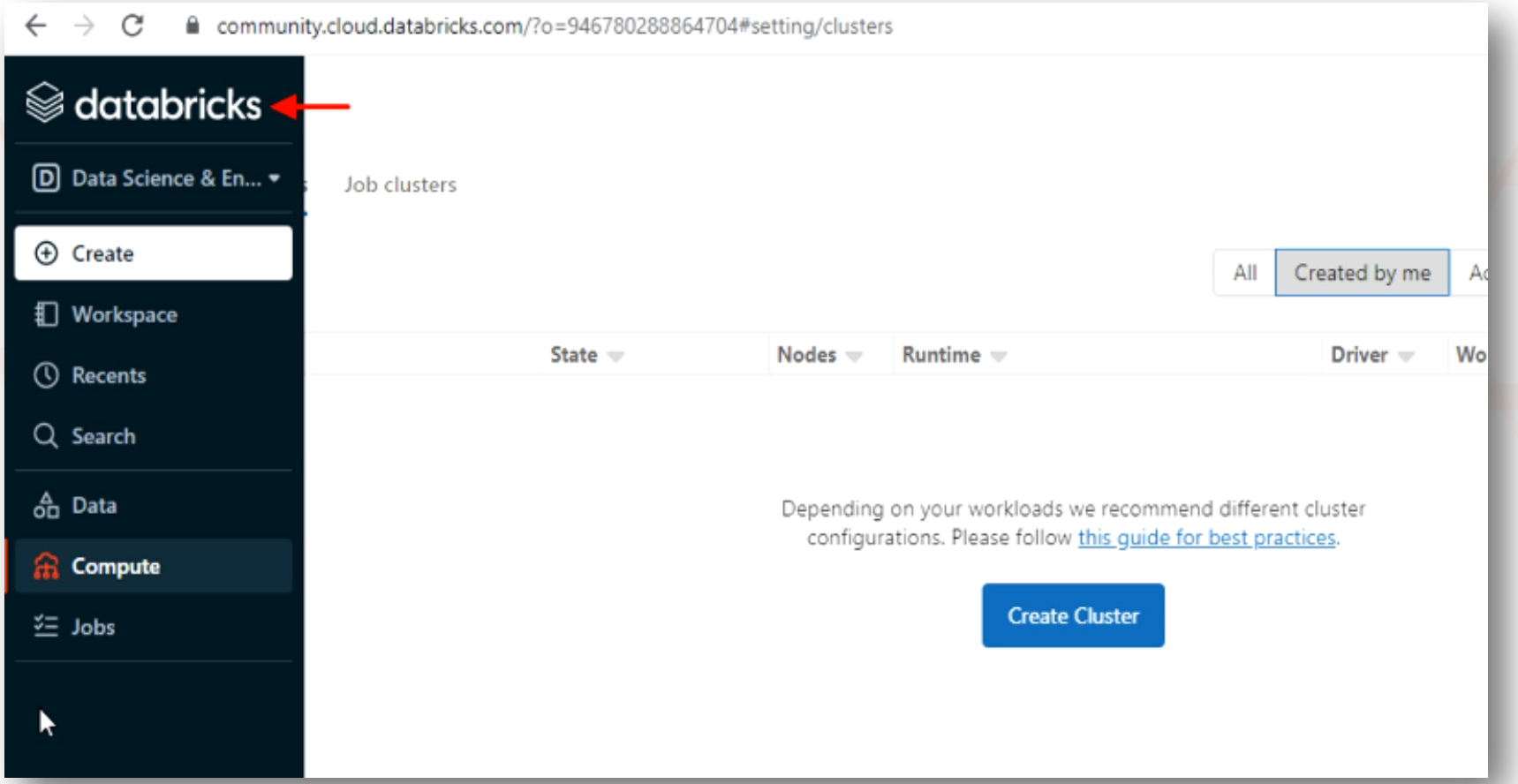
1. All purpose clusters – They are suitable for development and testing activities.
2. Job Clusters - These are short-term on-demand clusters, which launch a Spark application, run it, and automatically shuts down once your application finishes. It is not available for the community edition.



The following menu item is for creating Spark Jobs.
However, the Spark Jobs feature is not available in the community edition.



You can click on this “databricks” logo shown below to return to the workspace homepage.



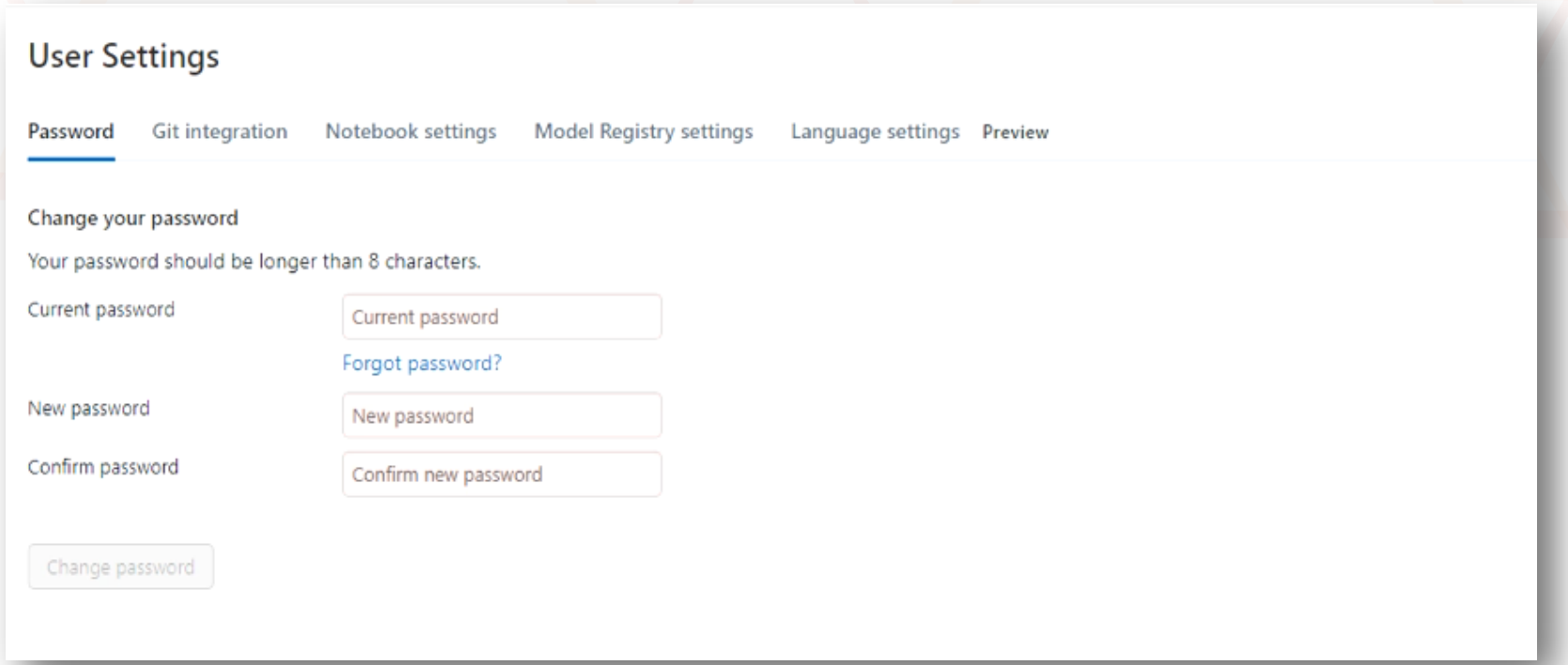
You also have a few more things in the menu down below, such as Help, Settings, Logout Link, and Menu Options.

If you go to the settings option, you can see 3 links:

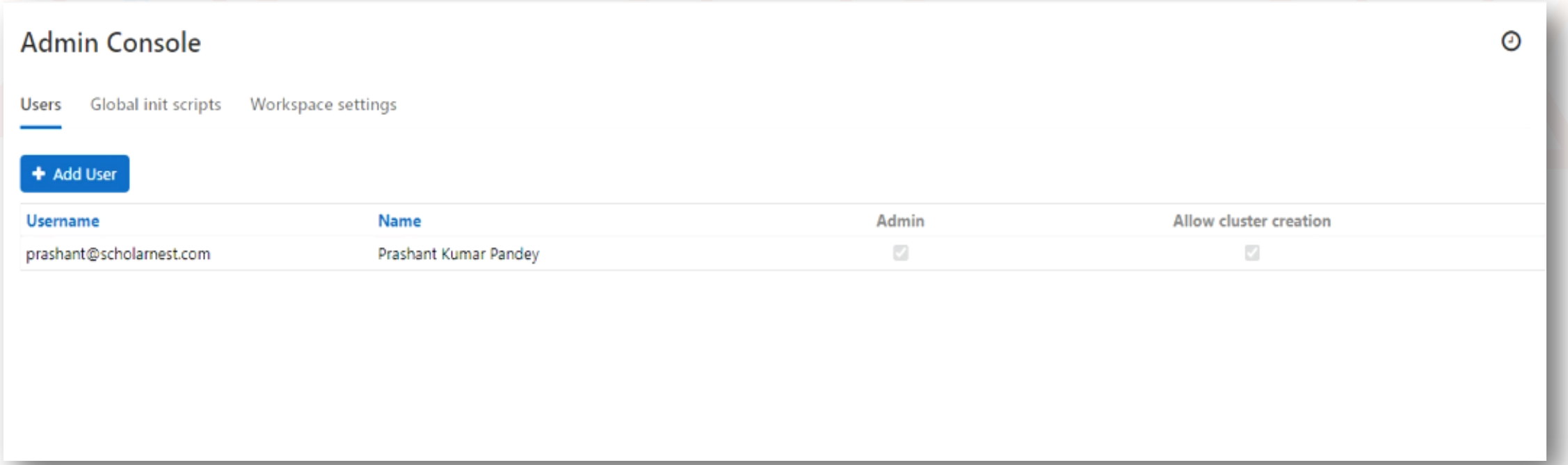
1. User Settings
2. Admin Console
3. Manage Account

If you go inside user settings from the main settings option, you will see the following alternatives.

1. You can change your password from here.
2. You can manage your Git integration.
3. You have some notebook settings
4. There is some model registry setting
5. There are some language settings as well

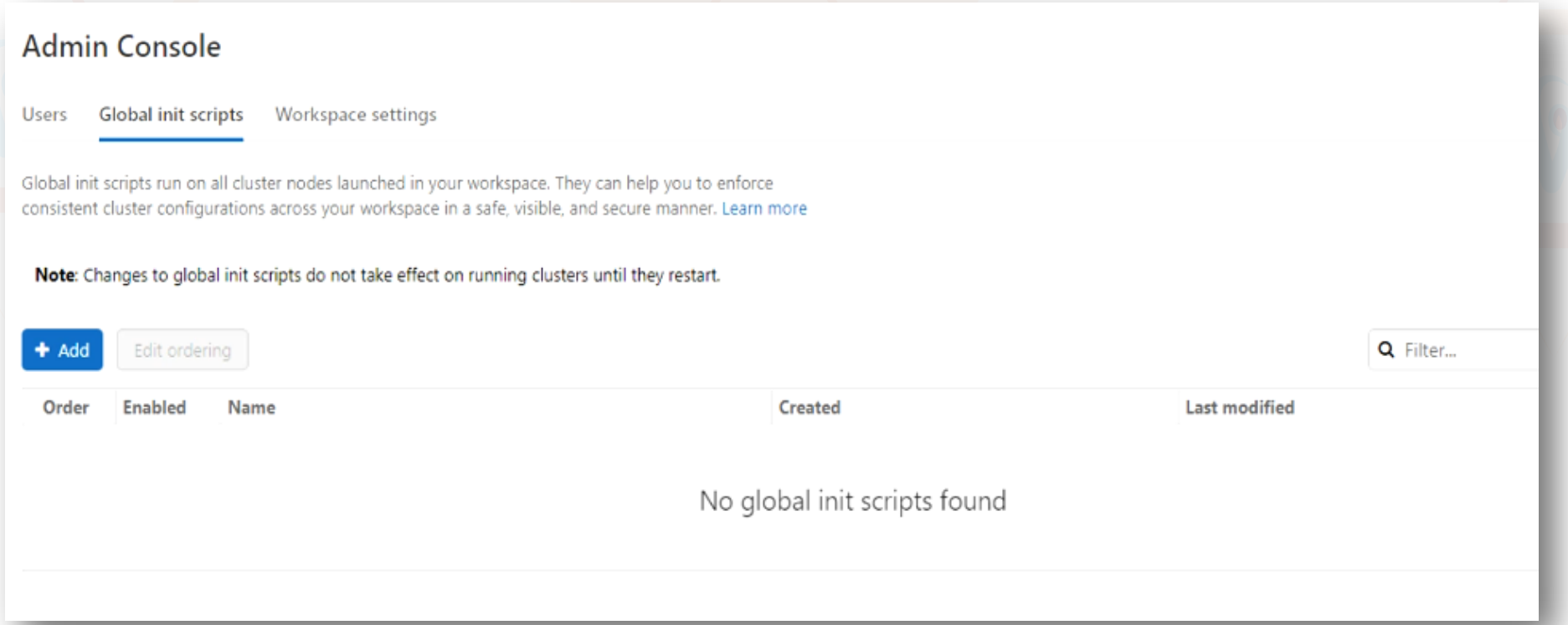


If you go inside admin console from the main settings option, you will see a screen as shown below.
I created this workspace, so I became the admin for this workspace. However, you can invite other users by clicking the add user button shown below.



There are a couple of more options available inside the admin console.

1. Global Init Scripts – It is known as cluster node initialization scripts, which is basically a shell script that runs during the start-up of each cluster node before the Apache Spark driver or worker JVM starts. You can use these scripts to configure your cluster and install other things not included in the Databricks runtime.
2. Workspace Settings – You can use this option to customize your workspace.



Thank You

ScholarNest Technologies Pvt Ltd.

www.scholarnest.com

ScholarNest

Spark Azure Databricks

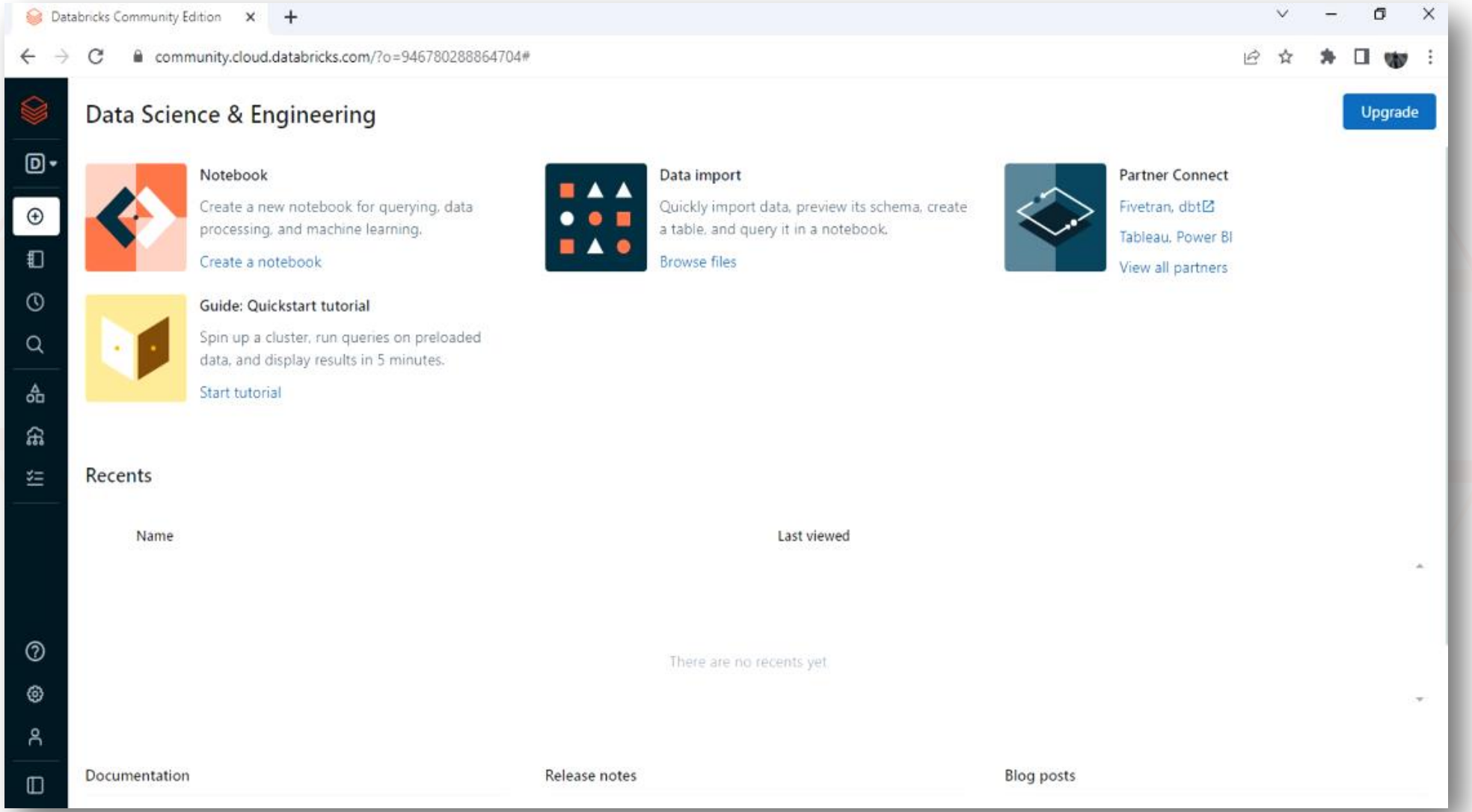
Databricks Spark Certification and beyond

Instructor: Prashant Kumar Pandey

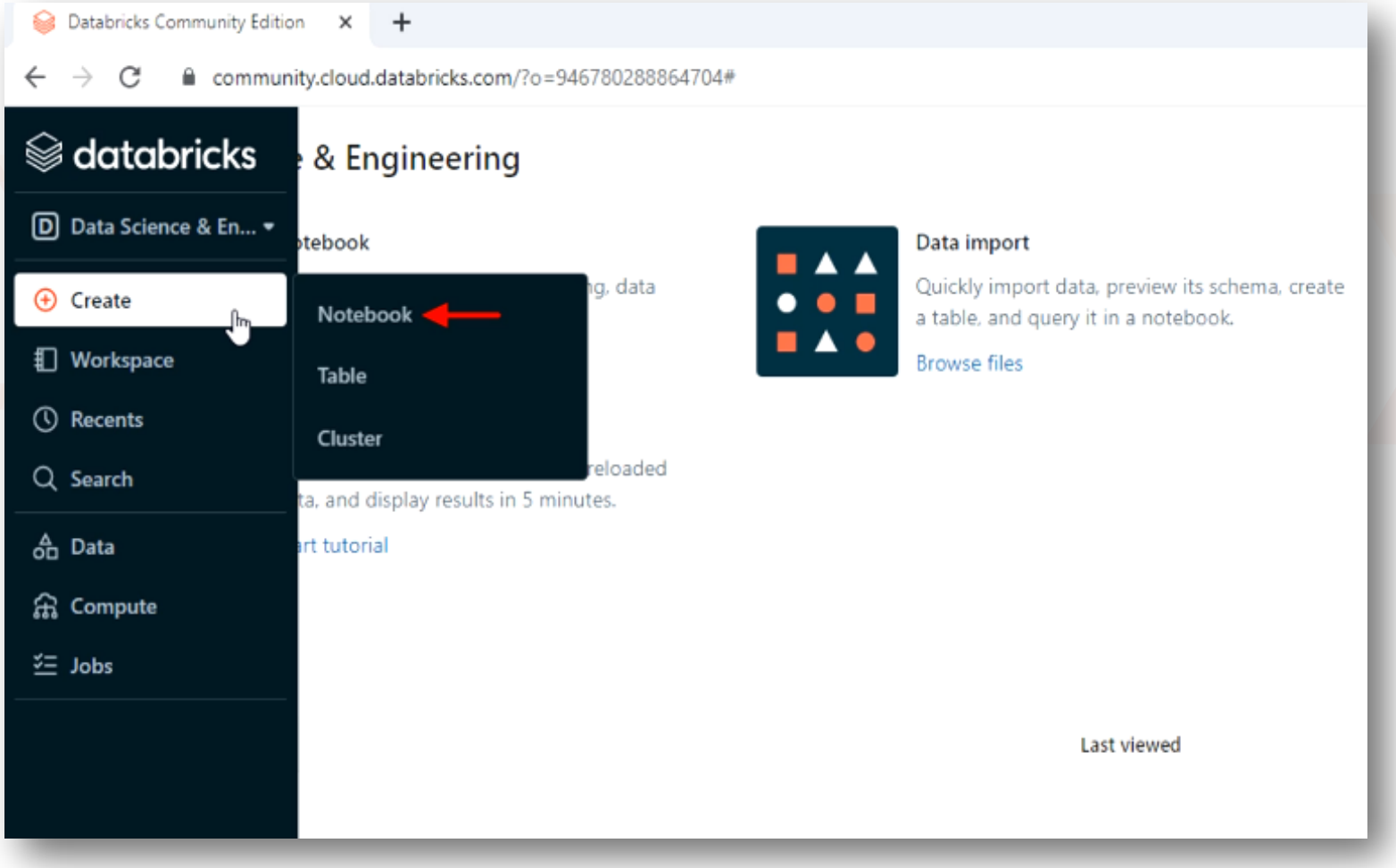
Absolute Beginner to Specialization in Apache Spark and Azure Databricks

First Spark Application in Databricks Cloud

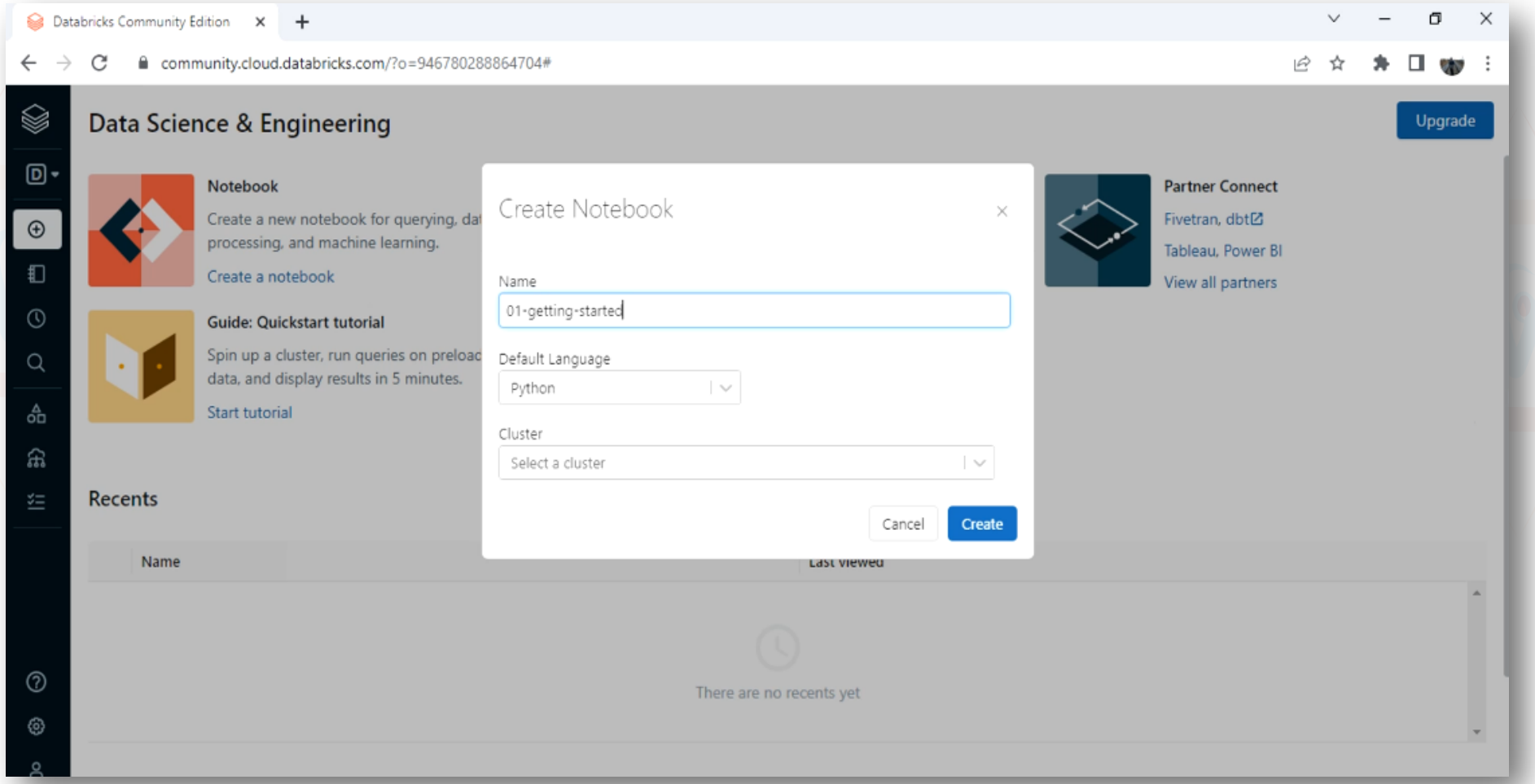
Login to your Databricks Workspace.



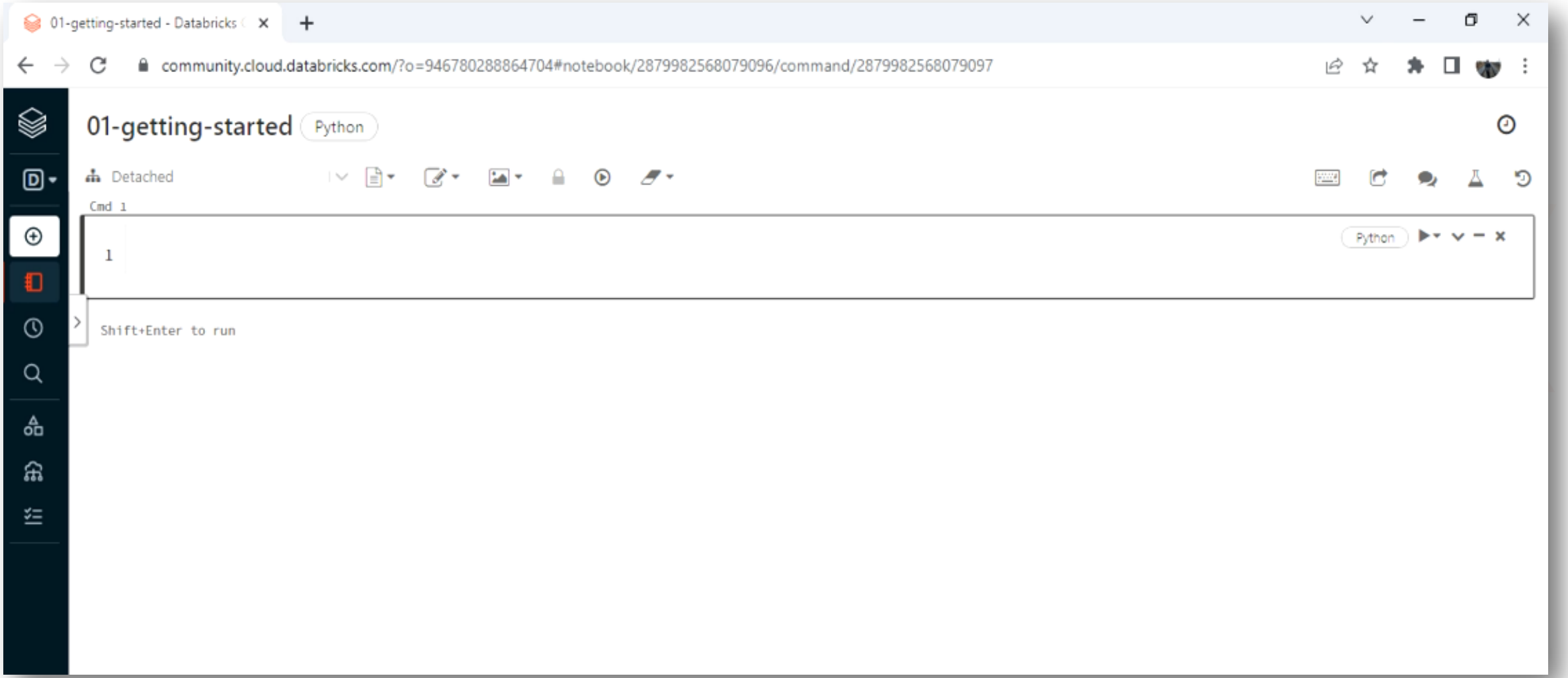
Come to the navigation menu and hit the create button. You will see three options there: Notebook, Table and Cluster.
Hit the notebook option and start creating a notebook.



Give a name to your notebook and hit the create button to get started.



Here is your notebook.



Requirements:

We have some data available in the cloud environment at the location given below. It is a CSV format data file, which contains some information about the diamonds.

Requirement

Given data file

/databricks-datasets/Rdatasets/data-001/csv/ggplot2/diamonds.csv

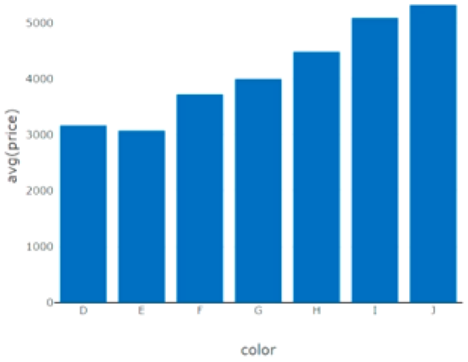
1. Read show

_c0	carat	cut	color	clarity	depth	table	price	x	y	z
1	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
2	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
3	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
4	0.29	Premium	I	VS2	62.4	58.0	334	4.2	4.23	2.63
5	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

2. Average price by colour.

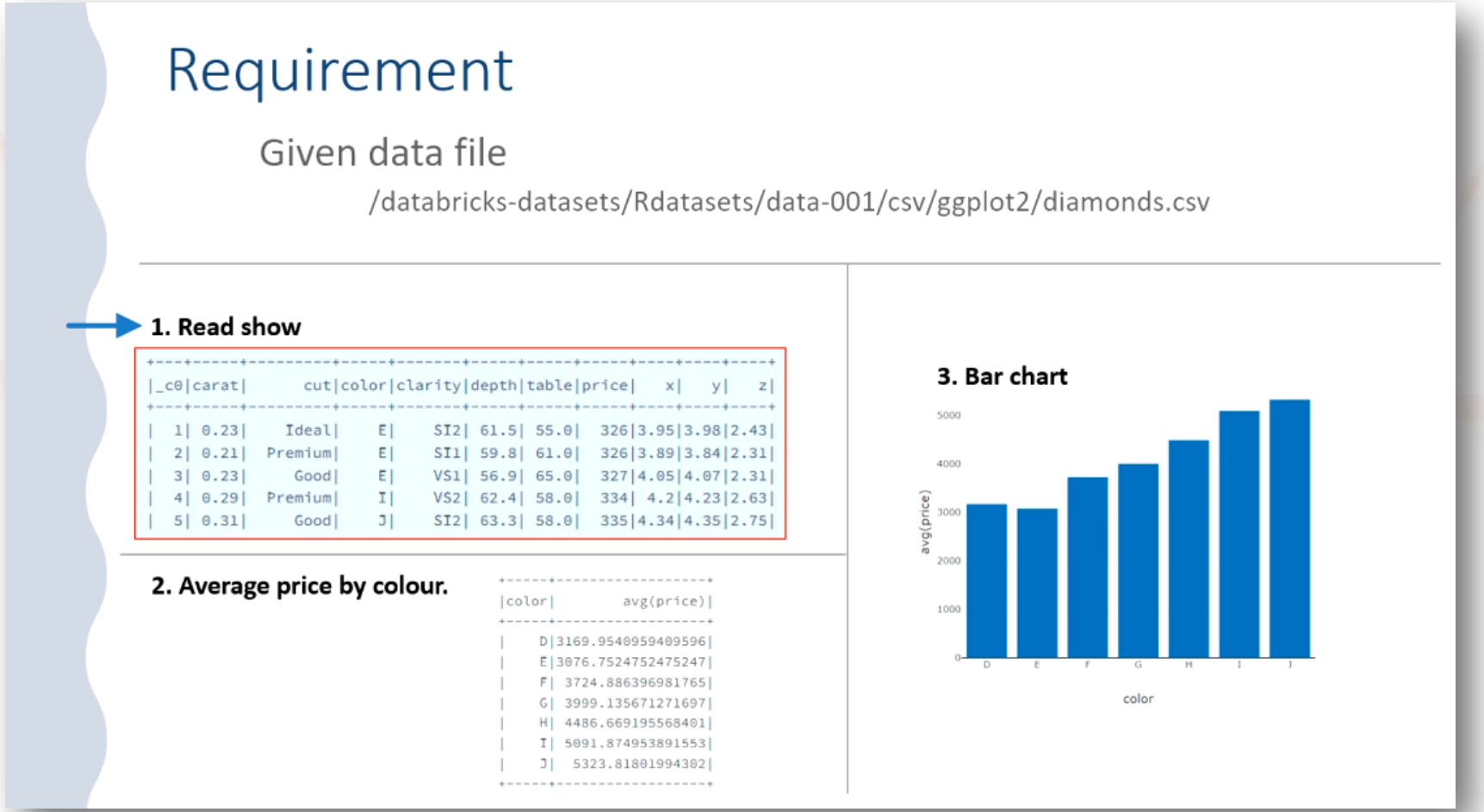
color	avg(price)
D	3169.9540959409596
E	3076.7524752475247
F	3724.886396981765
G	3999.135671271697
H	4486.669195568401
I	5091.874953891553
J	5323.81801994302

3. Bar chart



Requirements:

1. Read the data file and show some records in tabular format. And the output should look like the image highlighted below.



Requirements:

2. Calculate the Average price of the diamonds by color. And the result might look like the image highlighted below.

Requirement

Given data file

/databricks-datasets/Rdatasets/data-001/csv/ggplot2/diamonds.csv

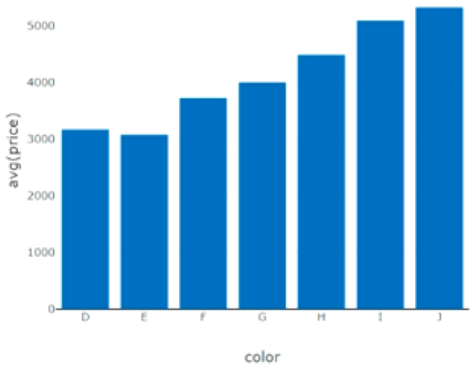
1. Read show

_c0	carat	cut	color	clarity	depth	table	price	x	y	z
1	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
2	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
3	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
4	0.29	Premium	I	VS2	62.4	58.0	334	4.2	4.23	2.63
5	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

2. Average price by colour.

color	avg(price)
D	3169.9540959409596
E	3076.7524752475247
F	3724.886396981765
G	3999.135671271697
H	4486.669195568401
I	5091.874953891553
J	5323.81801994302

3. Bar chart



Requirements:

3. Draw a bar chart using this table to visualize the results. And the bar chart might look like the image highlighted below.

Requirement

Given data file

/databricks-datasets/Rdatasets/data-001/csv/ggplot2/diamonds.csv

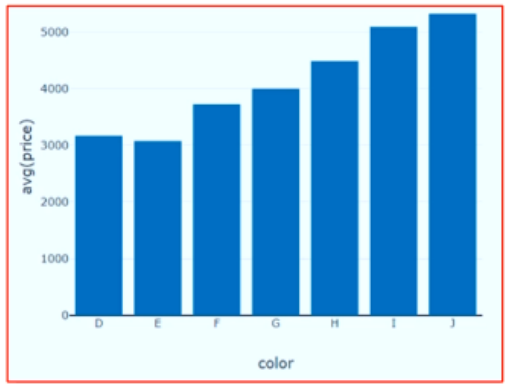
1. Read show

_c0	carat	cut	color	clarity	depth	table	price	x	y	z
1	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
2	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
3	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
4	0.29	Premium	I	VS2	62.4	58.0	334	4.2	4.23	2.63
5	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

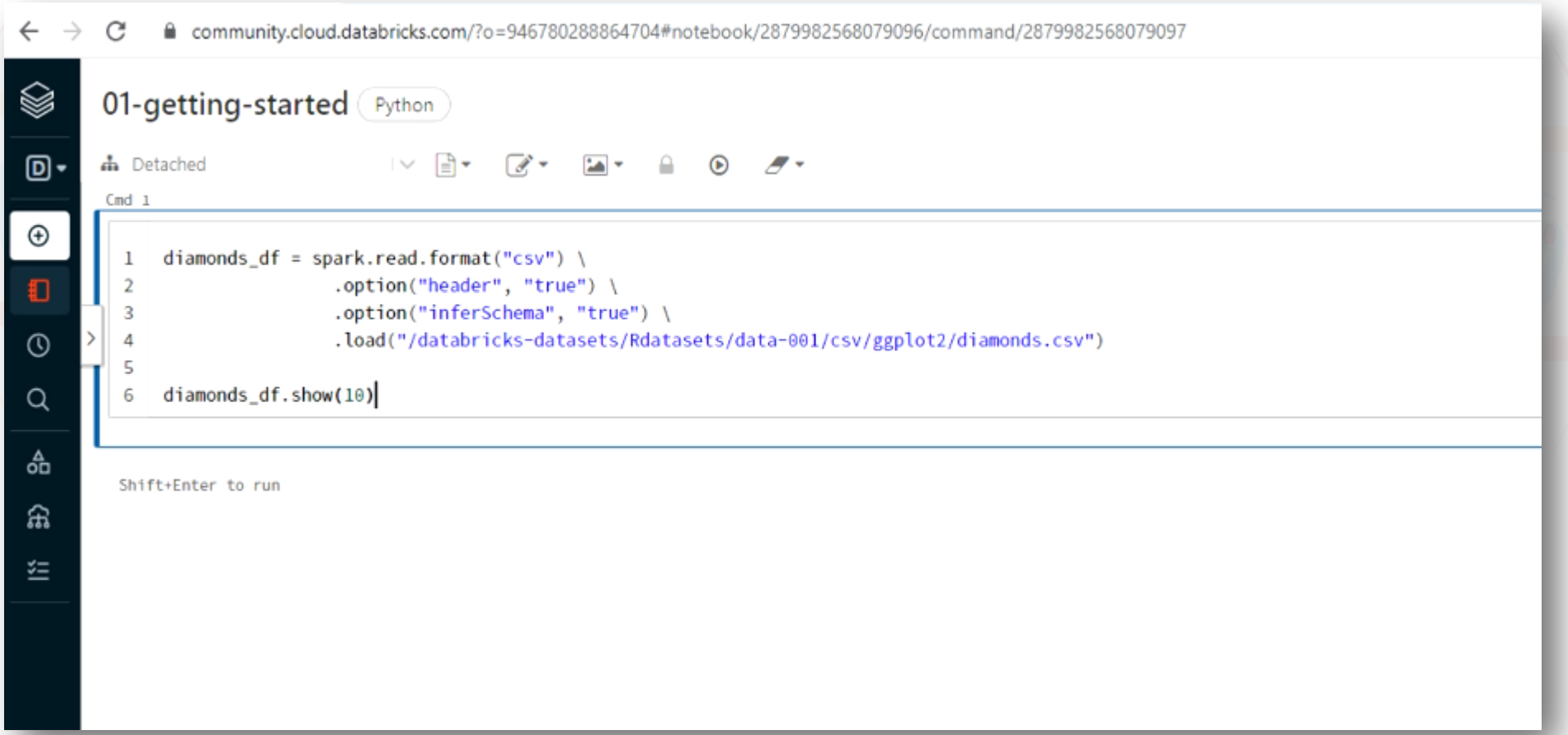
2. Average price by colour.

color	avg(price)
D	3169.9540959409596
E	3076.7524752475247
F	3724.886396981765
G	3999.135671271697
H	4486.669195568401
I	5091.874953891553
J	5323.81801994302

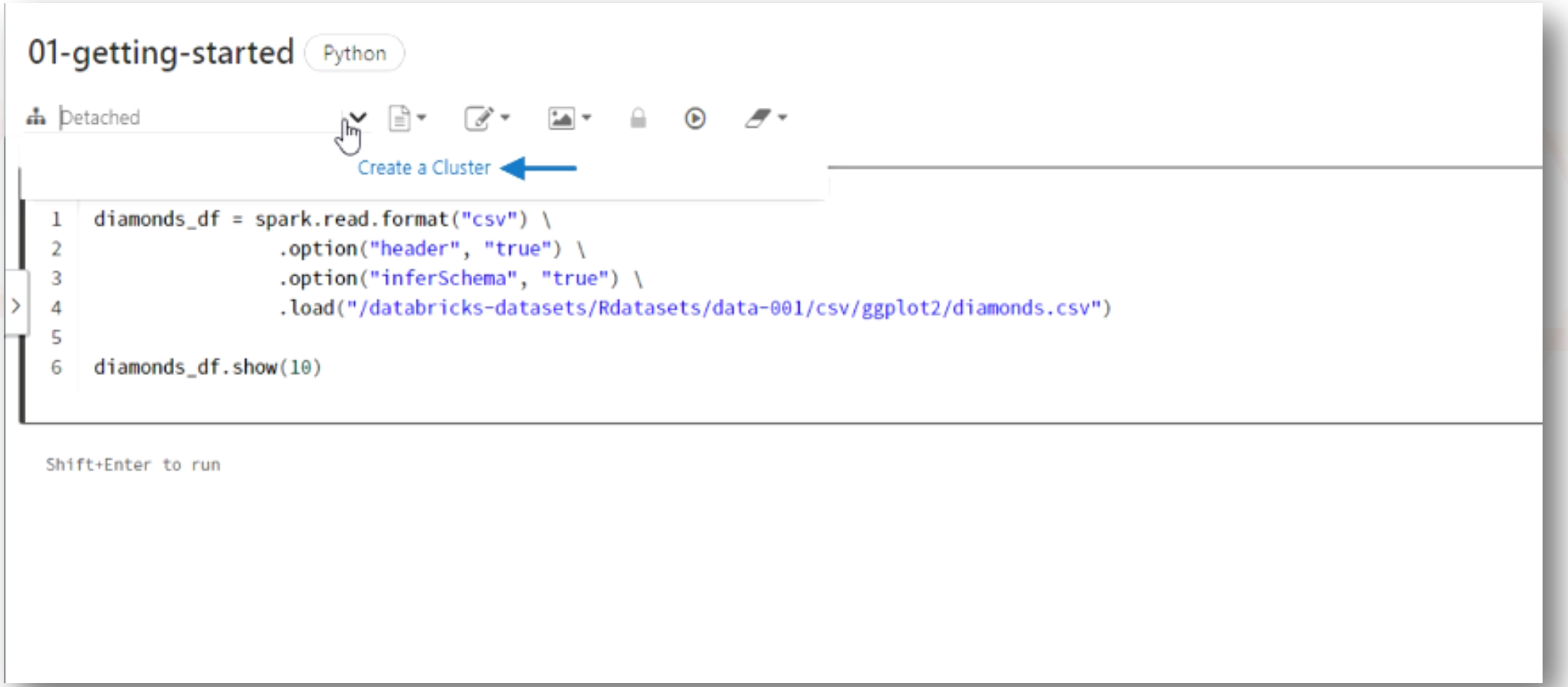
3. Bar chart



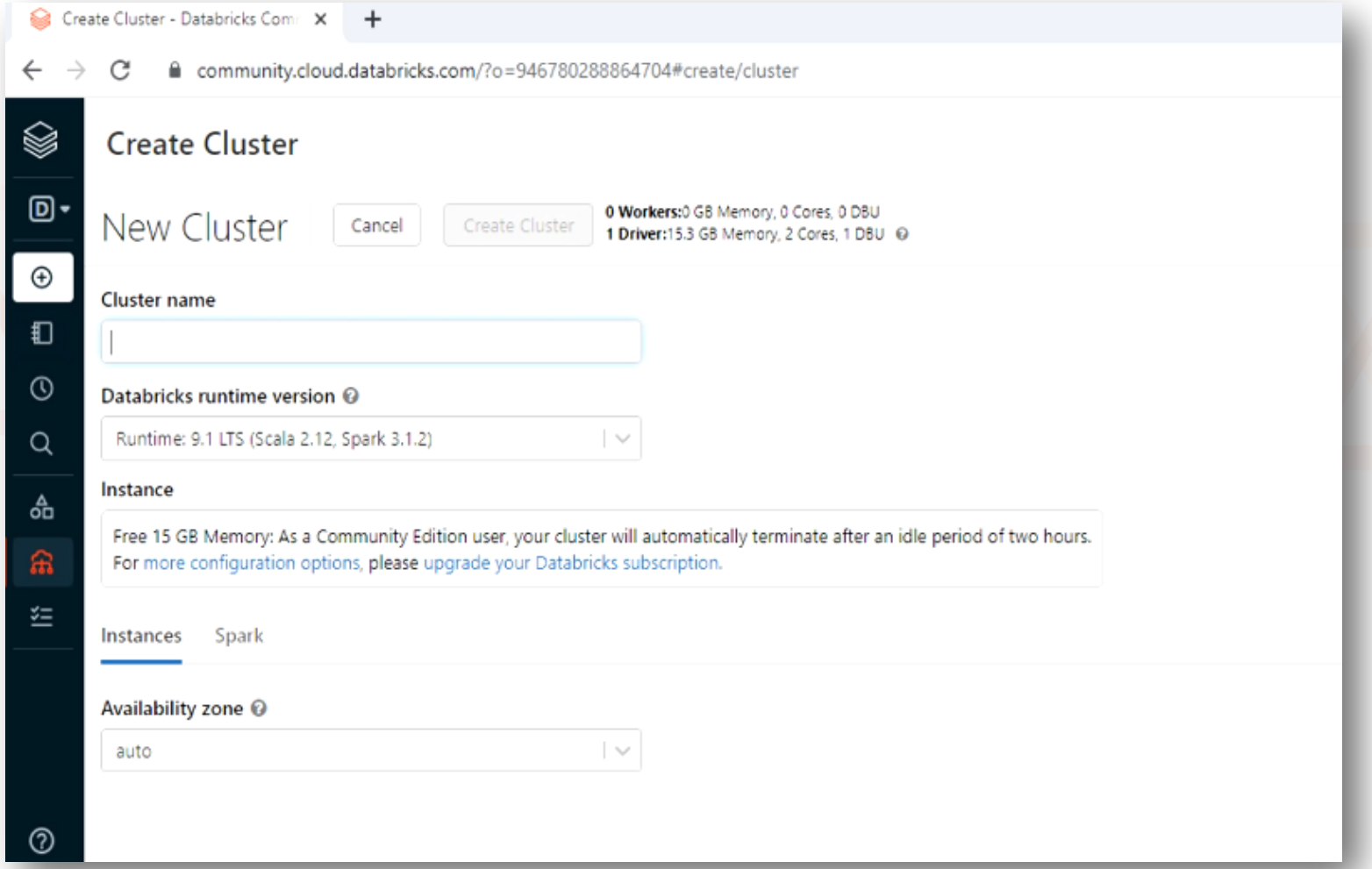
Here is the code shown below which meets your first requirement.
(Reference: 01-getting-started.ipynb)
It will read the data file and show some records in tabular format.



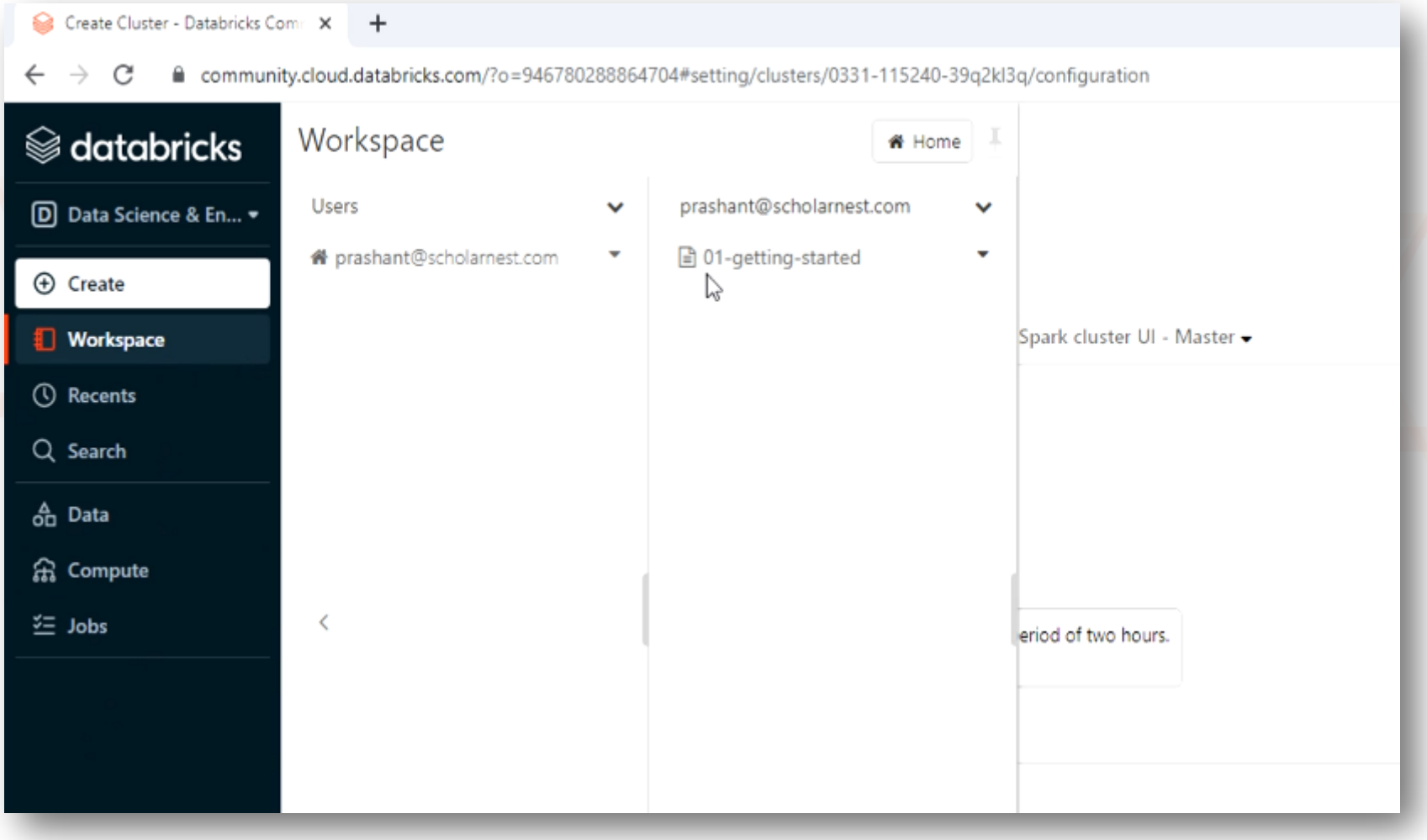
Now, we want to run this code and see the results. But running a Spark code requires a Spark cluster. As shown in the image below, click the dropdown menu, and you will see a link to create a cluster. Hit the link, and it will take you to the cluster creation page.



Give a name to the cluster and hit the create button, it might take a few minutes to wait for the cluster to start.

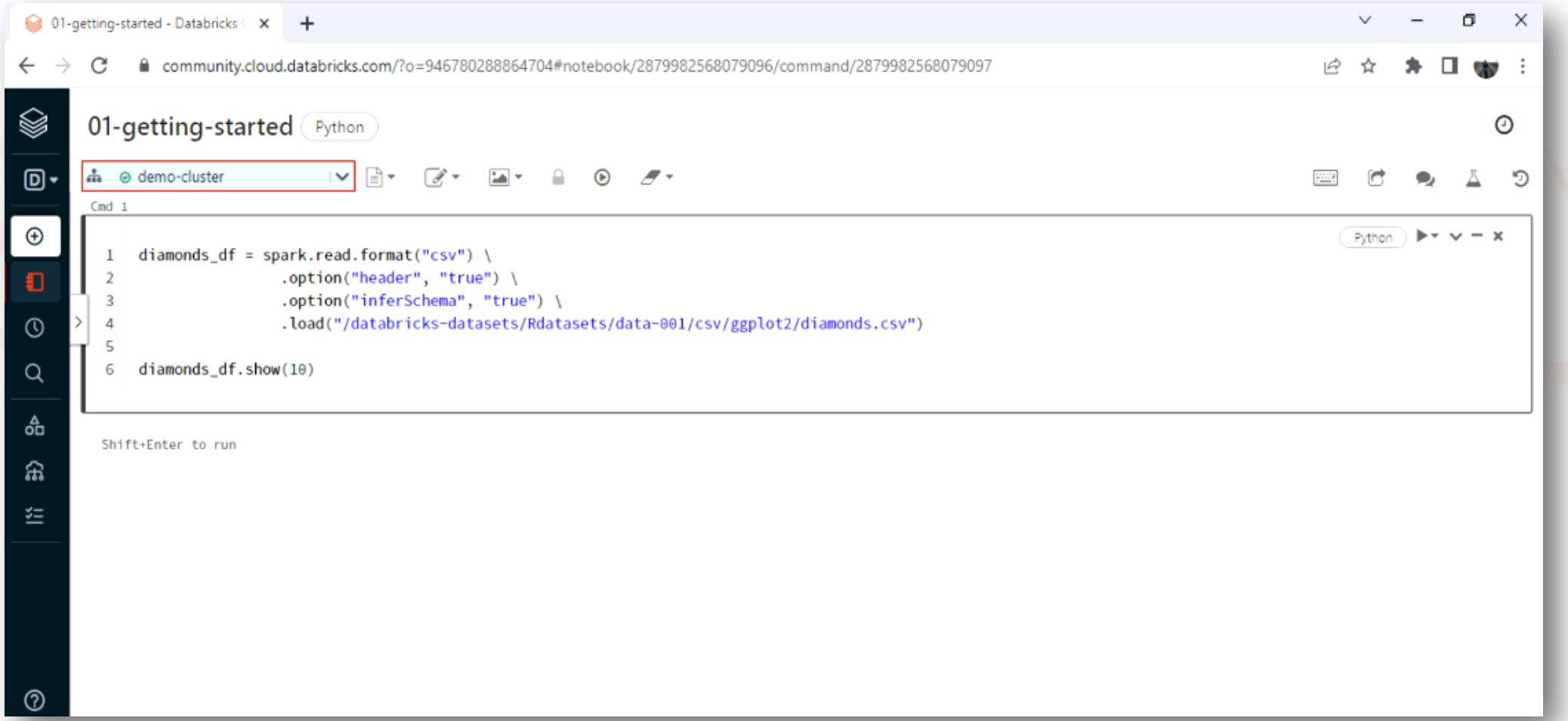


Now, go to the workspace menu option and go back to your notebook.

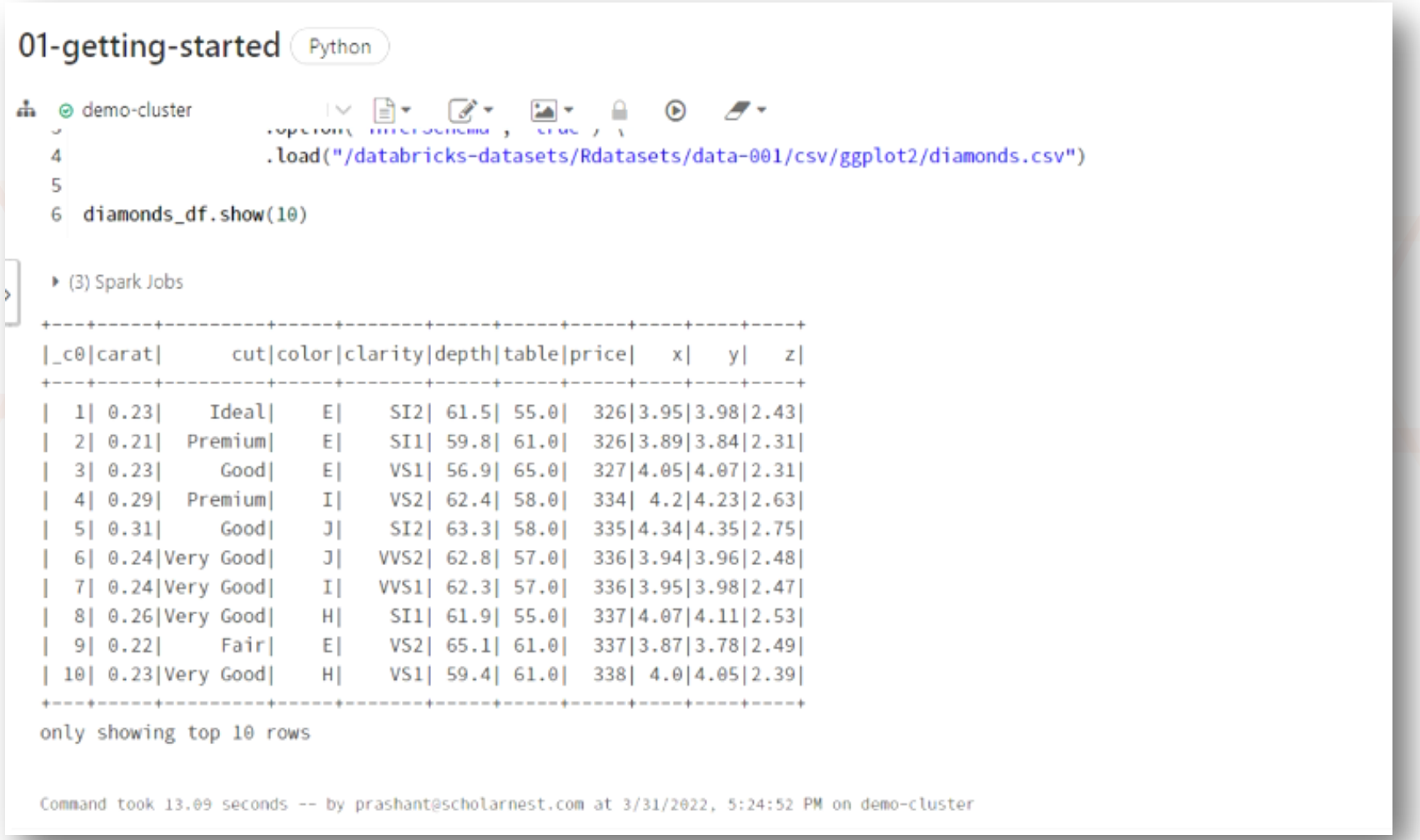


Make sure your cluster is attached to the notebook, and your code will run on this cluster.

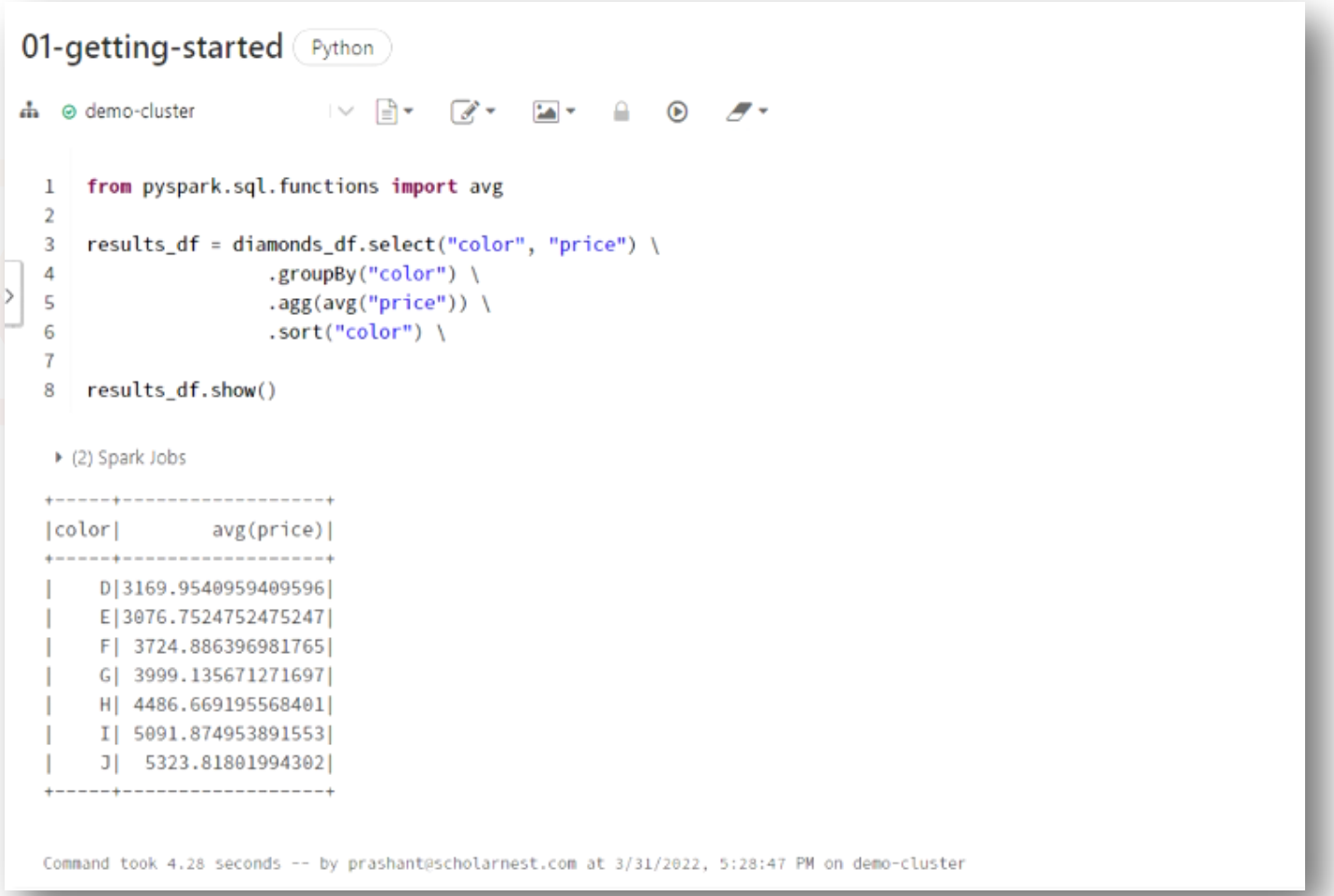
Now, click inside the code cell and press shift+enter to run the code.



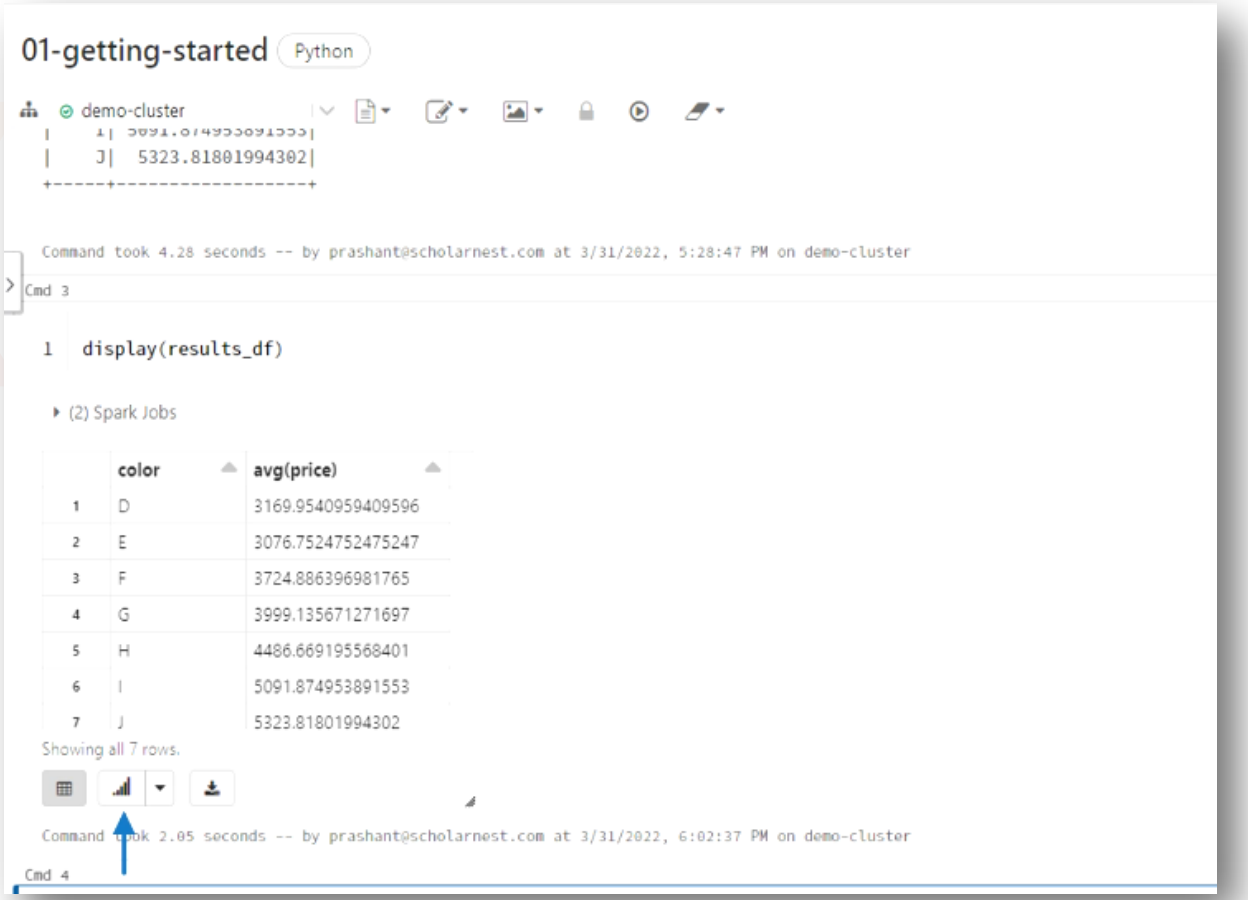
You can see the results here, and it matches our expected results. So we loaded the dimonds.csv file into Spark and showed ten records from the data file.



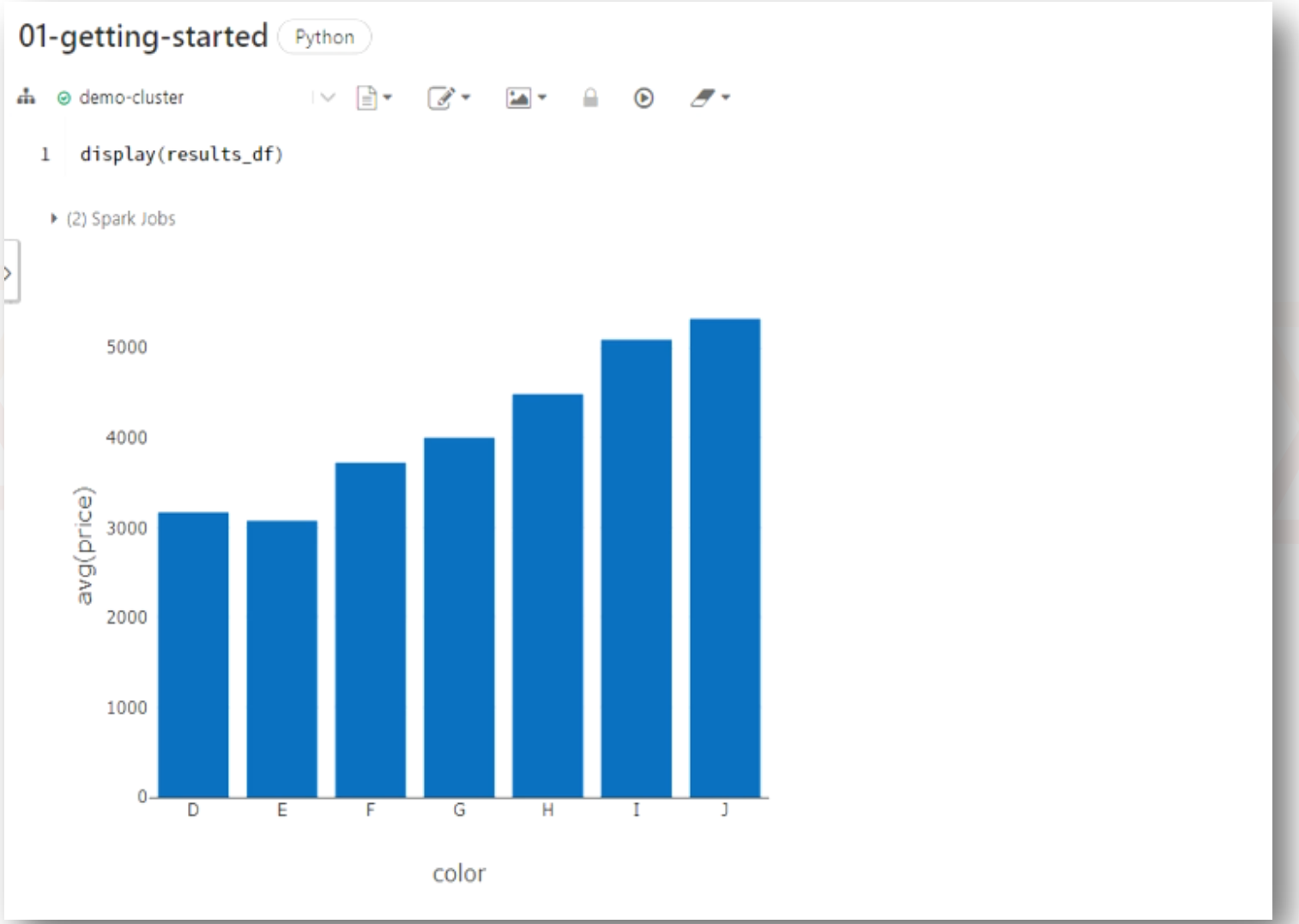
The following requirement is to process this diamond's data frame and calculate the average price by color. And here is the code for the same.
So this code takes color and price columns, groups them by color, and calculates the average price. Finally, I show the results. You can also see the output shown below.



In the last requirement, we wanted to present the result obtained using a bar chart. Here is the code for the same. And you can see the output below.
The display function allows you to format the results. The display function depicts the result in the tabular format but you can switch to the chart format by clicking chart button below the results.



And here is the bar chart we wanted.



Thank You

ScholarNest Technologies Pvt Ltd.

www.scholarnest.com

ScholarNest

Spark Azure Databricks

Databricks Spark Certification and beyond

Instructor: Prashant Kumar Pandey

Absolute Beginner to Specialization in Apache Spark and Azure Databricks

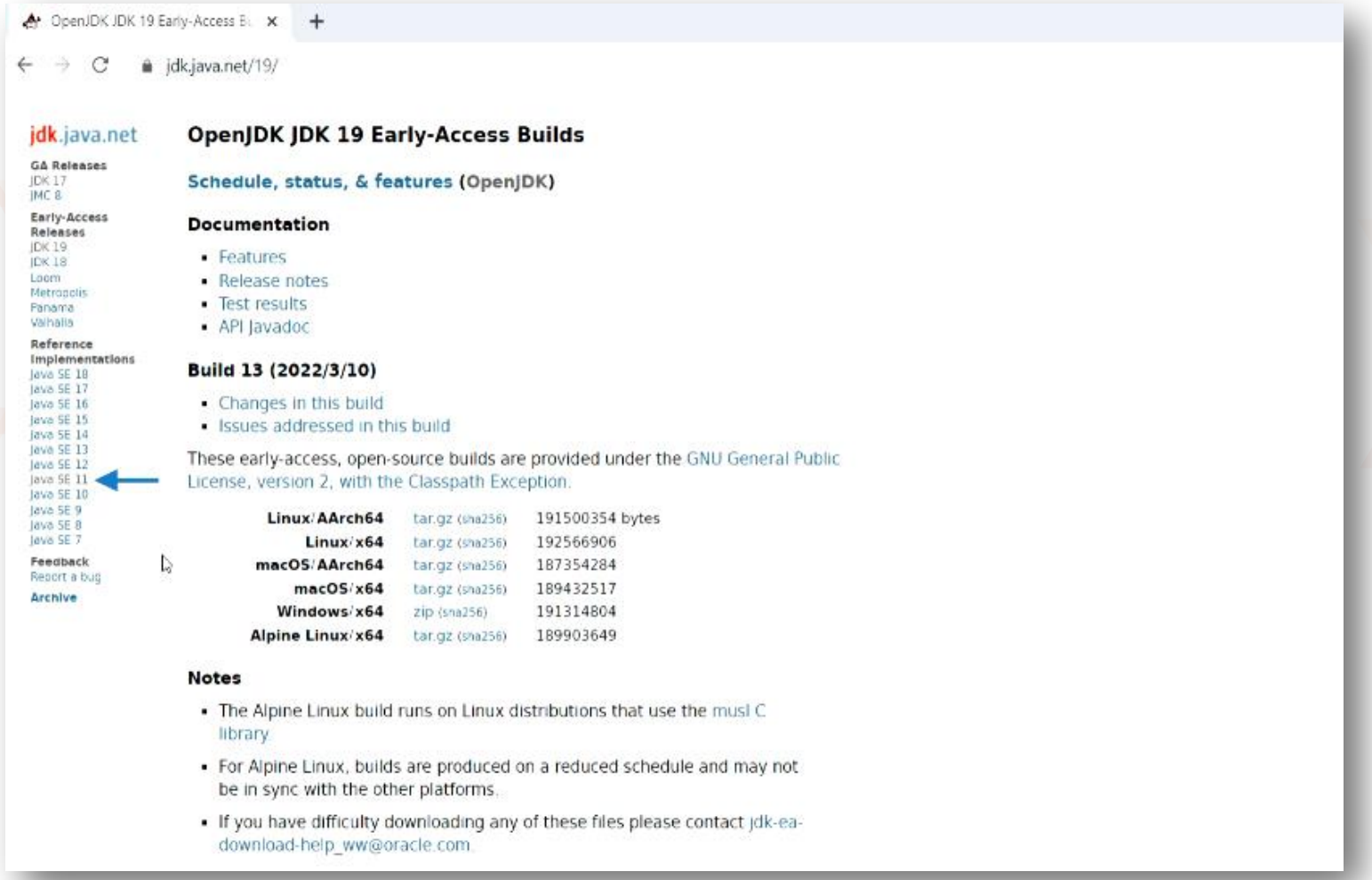
Setup your Local Development IDE

You need the following set of requirements for Spark setup on your machine. Let us go through this setup one by one.

Let's start with the JDK installation.
Spark is a JVM application. So you need JDK on your local machine for installing and running your Spark application.
The current most recent Java is at JDK 19. However, as of today, Spark supports JDK 8 or JDK 11.
Do not use any other version of JDK because Spark is not well tested on other versions.
You might face problems running Spark on other JDK versions.

Start your browser and visit http://jdk.java.net/
You will a link for the most recent version of the JDK. Click the link to go to the downloads page.

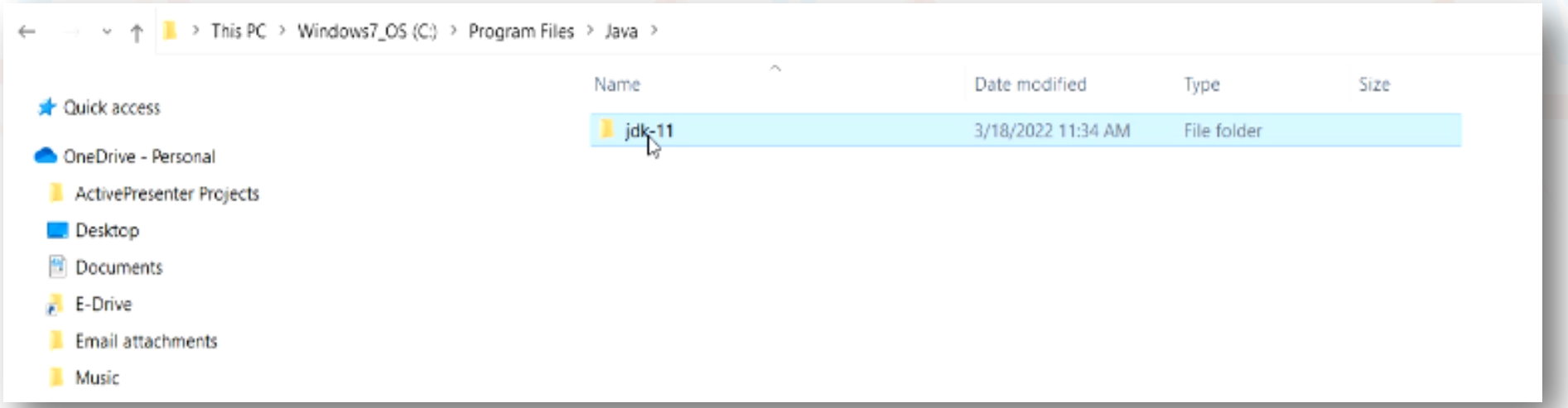
Then choose Java SE 11 from the list of all available versions.
And download the JDK 11 for your platform (Windows/Linux) in the next step.



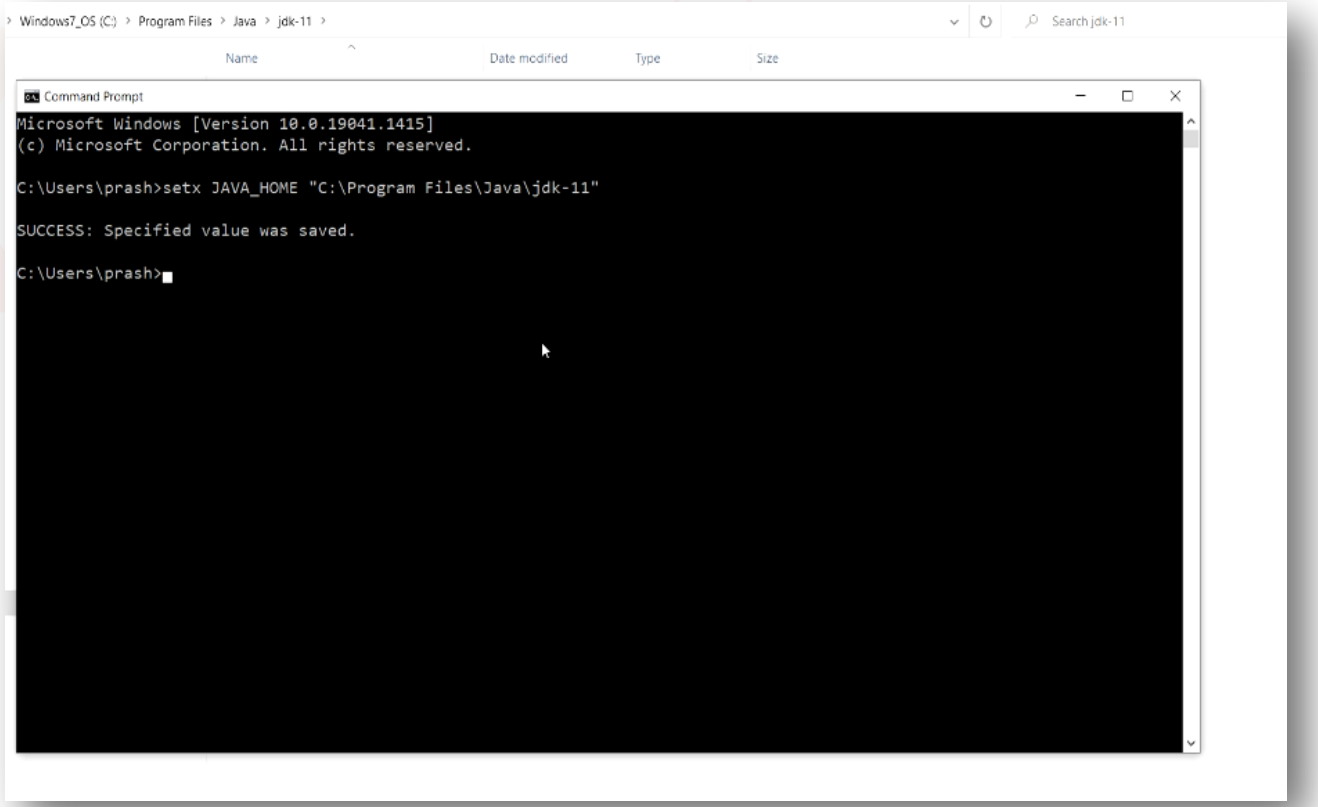
Go to the downloaded file and extract it. You will see the JDK-11 folder.

Inside the JDK-11, you will see some more folders. Copy the JDK-11 folder and paste it at some permanent place.

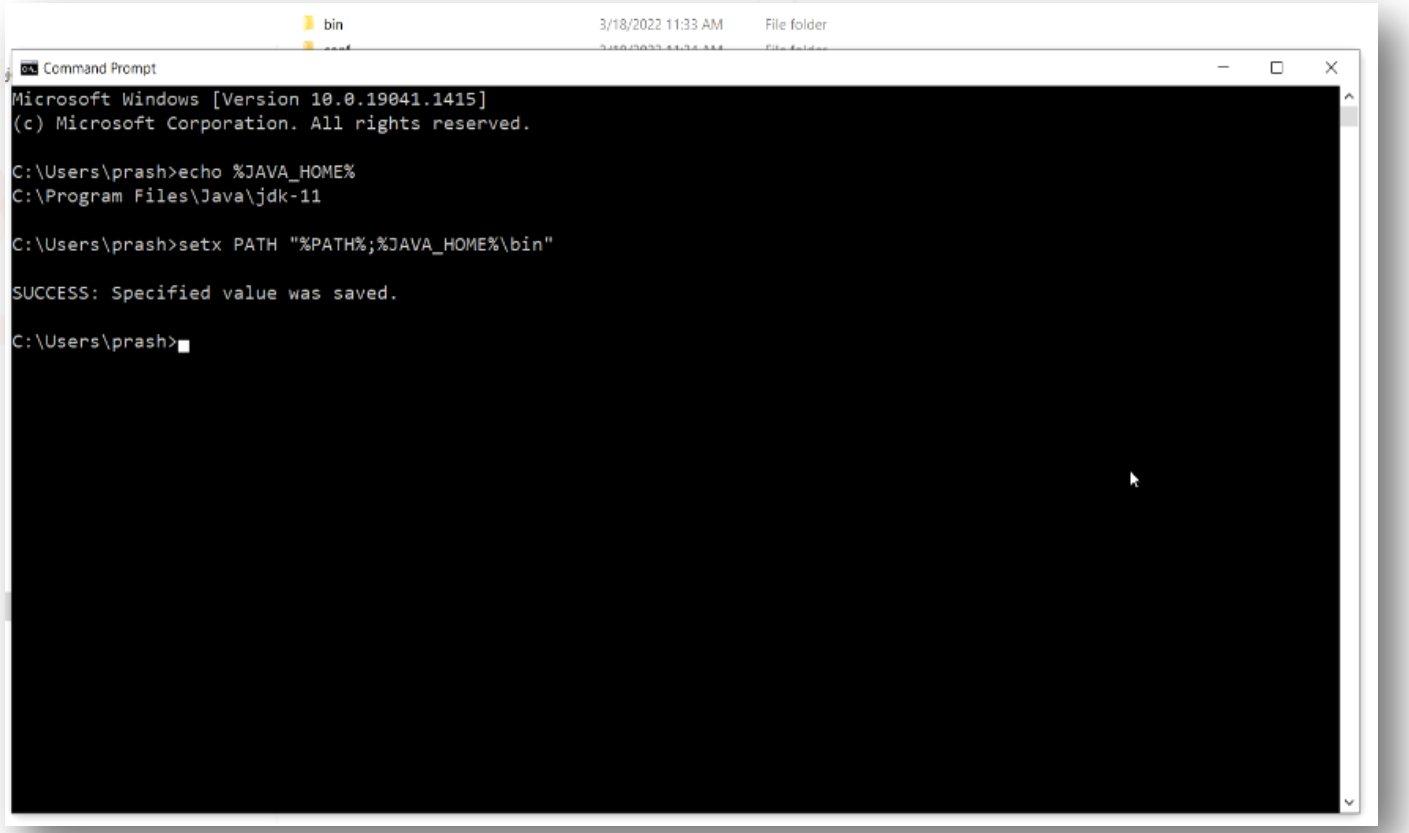
It is recommended to keep it in *C:\Program Files\Java* folder.



You have JDK 11 on your system. But it won't work as we need to set up two environment variables to make it work.
Start windows command prompt.
Now you can use the setx command as shown below to set the JAVA_HOME environment variable.
The JAVA_HOME variable must point to your JDK-11 directory.
Make sure you see a success message.

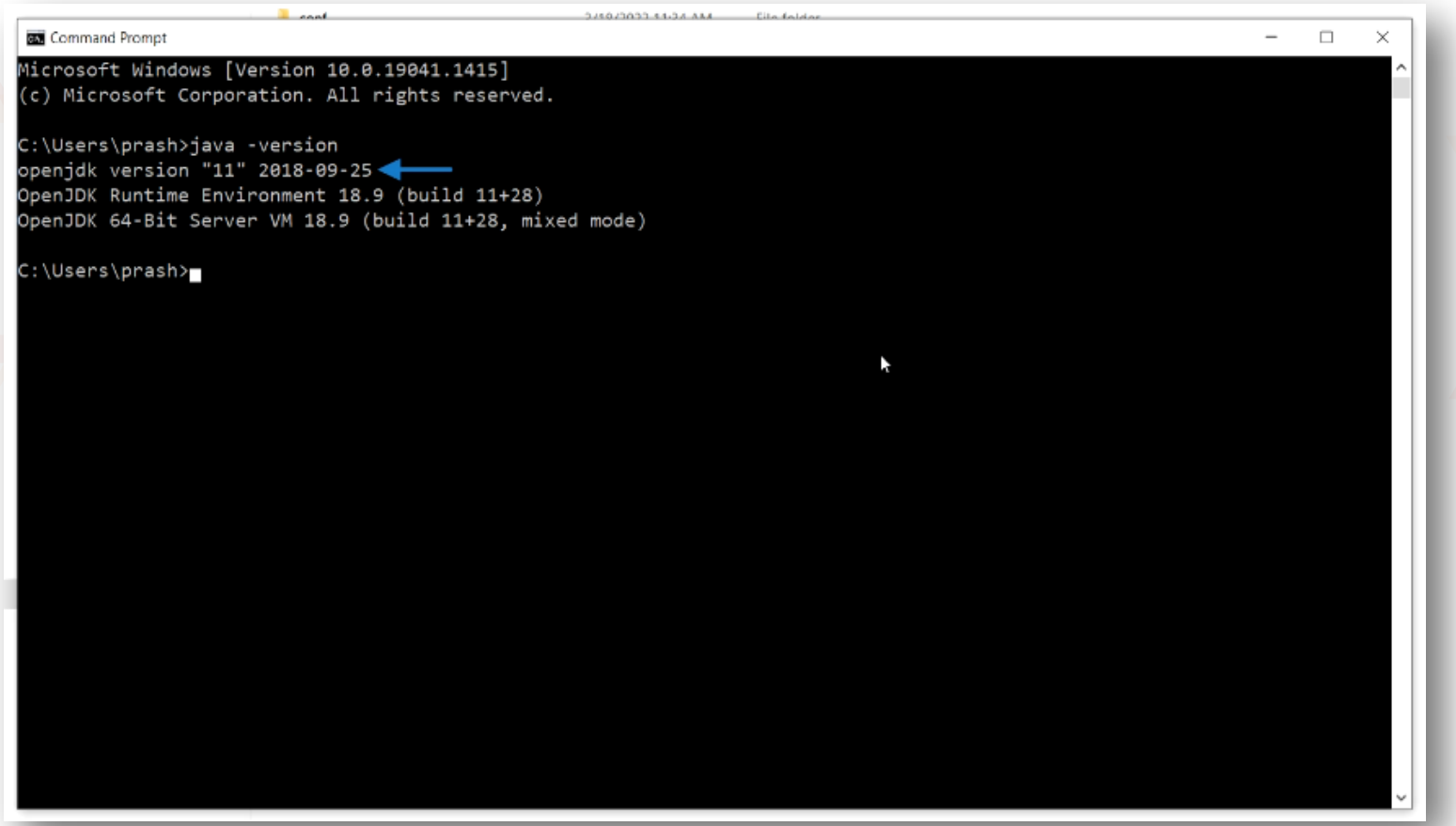


If you want to check that your environment variable is set properly, you can restart the command prompt and check it using the echo command as shown below.
Then the second requirement is to add the JAVA_HOME\bin to your PATH environment variable. You can use the setx command as shown in the image below. And make sure you get a success message. Ensure you include the current value of your PATH environment variable and add JAVA_MOME\bin to the same.



Finally, execute the Java -version command, and you should see the current Java version. Make sure you see Java version 11.

If you already have a Java 8 or Java 11 pre-installed on your machine, you can skip the Java installation steps.

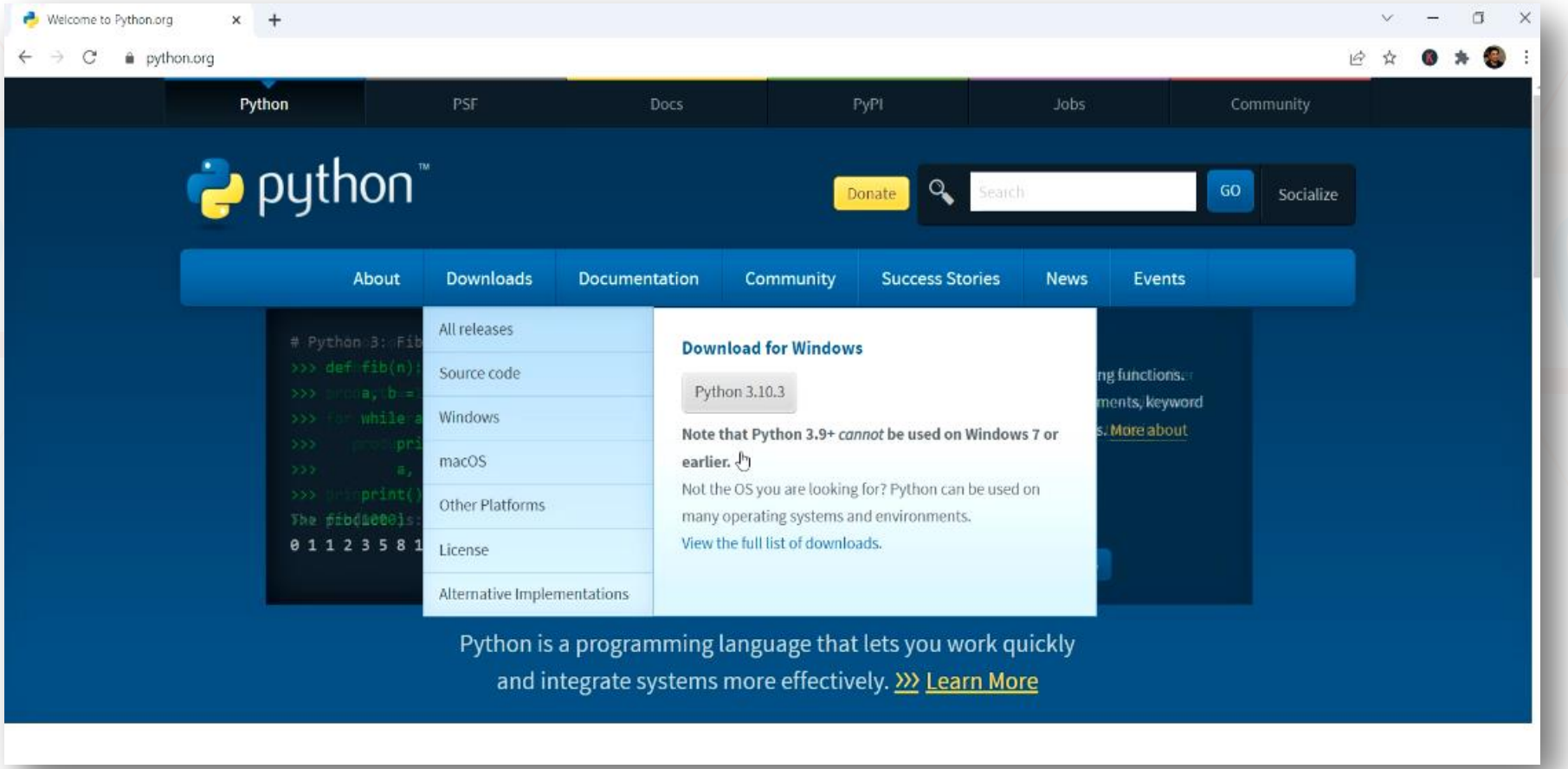


So, basically there are three steps you need to check in order to complete your JDK installation.

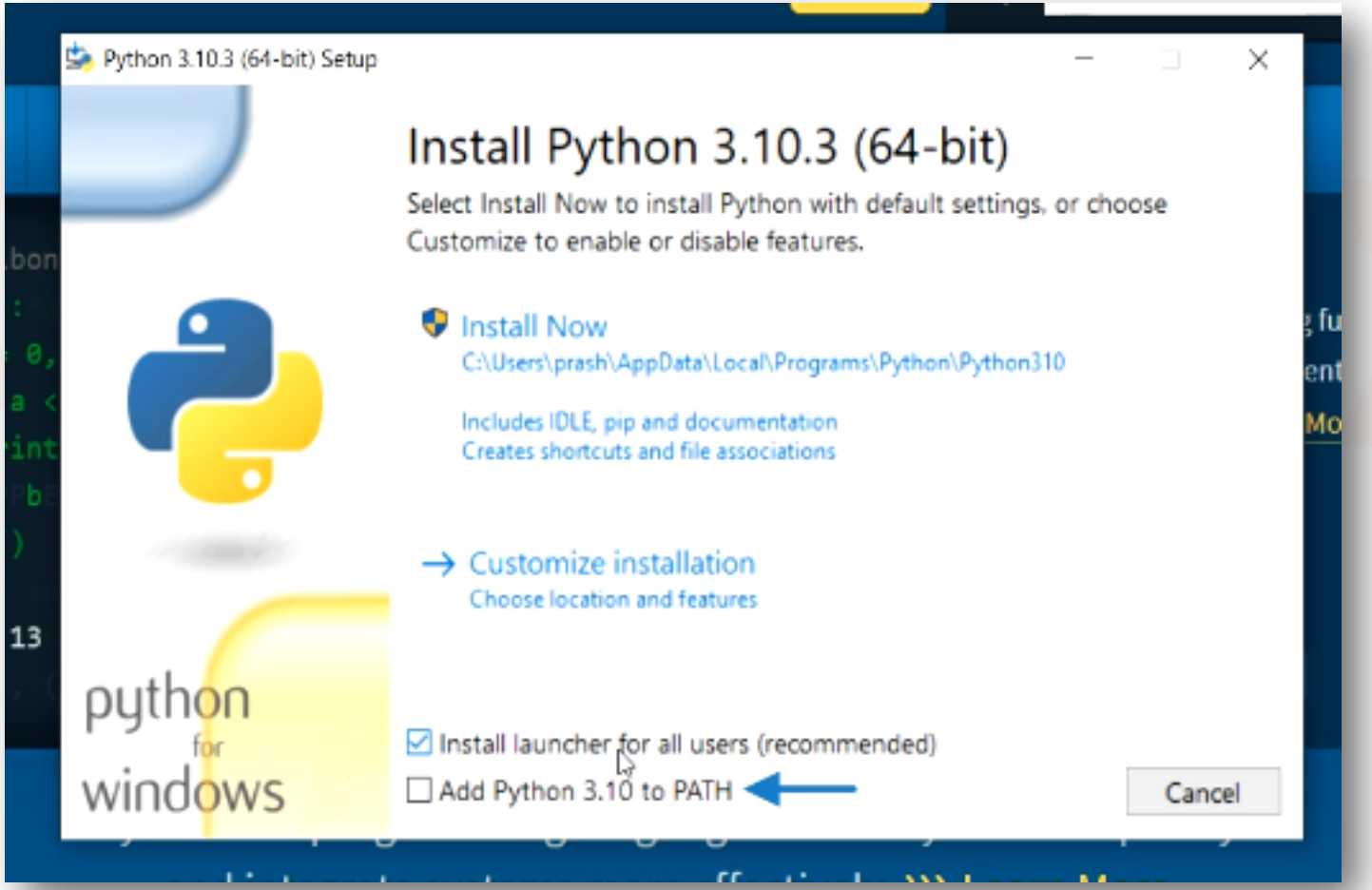
1. JAVA_HOME environment variable is set, and it is pointing to Java 8 or java 11
2. JAVA_HOME\bin is included in your PATH environment variable.
3. java -version command is showing Java 8 or Java 11

If all these three are there in place, you are done.

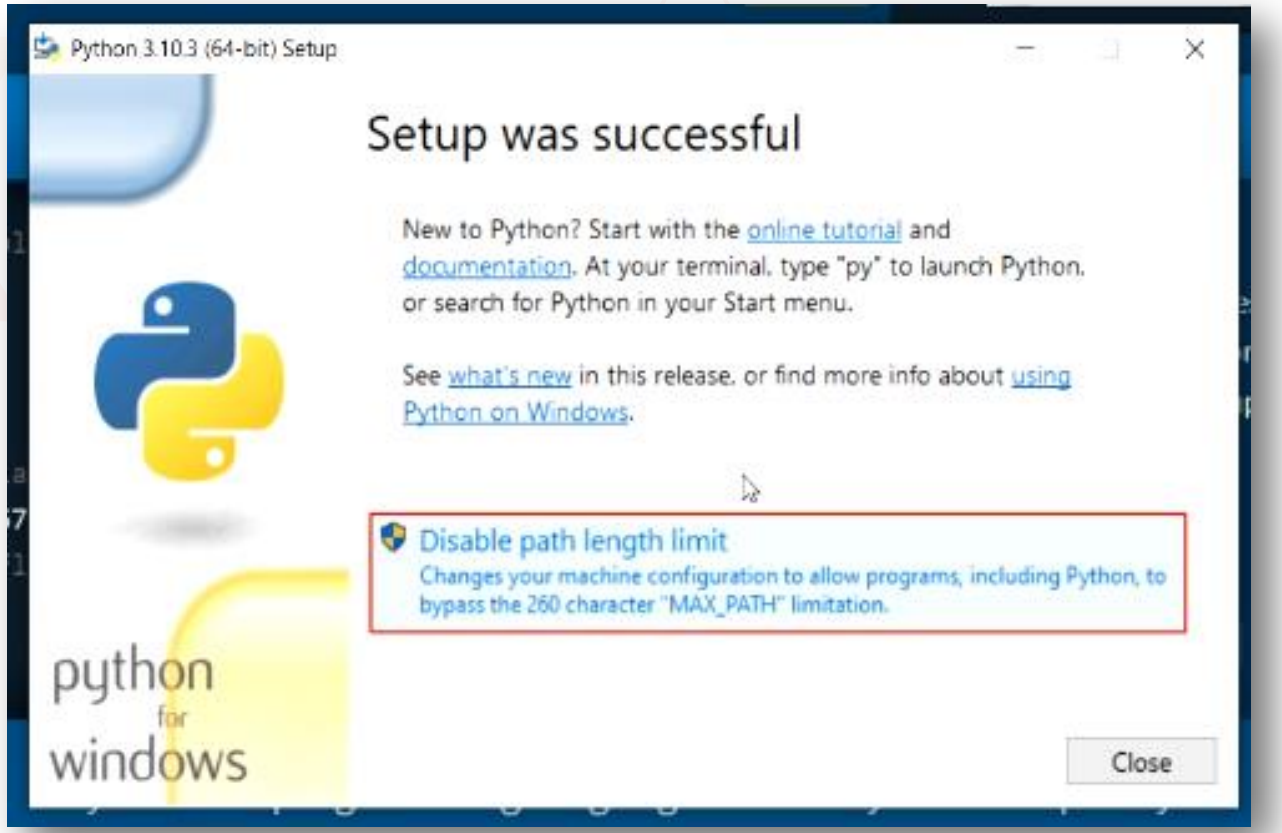
The next step to setup your Local Development IDE is to install Python. Visit https://www.python.org/ and download the latest Python version for your machine.



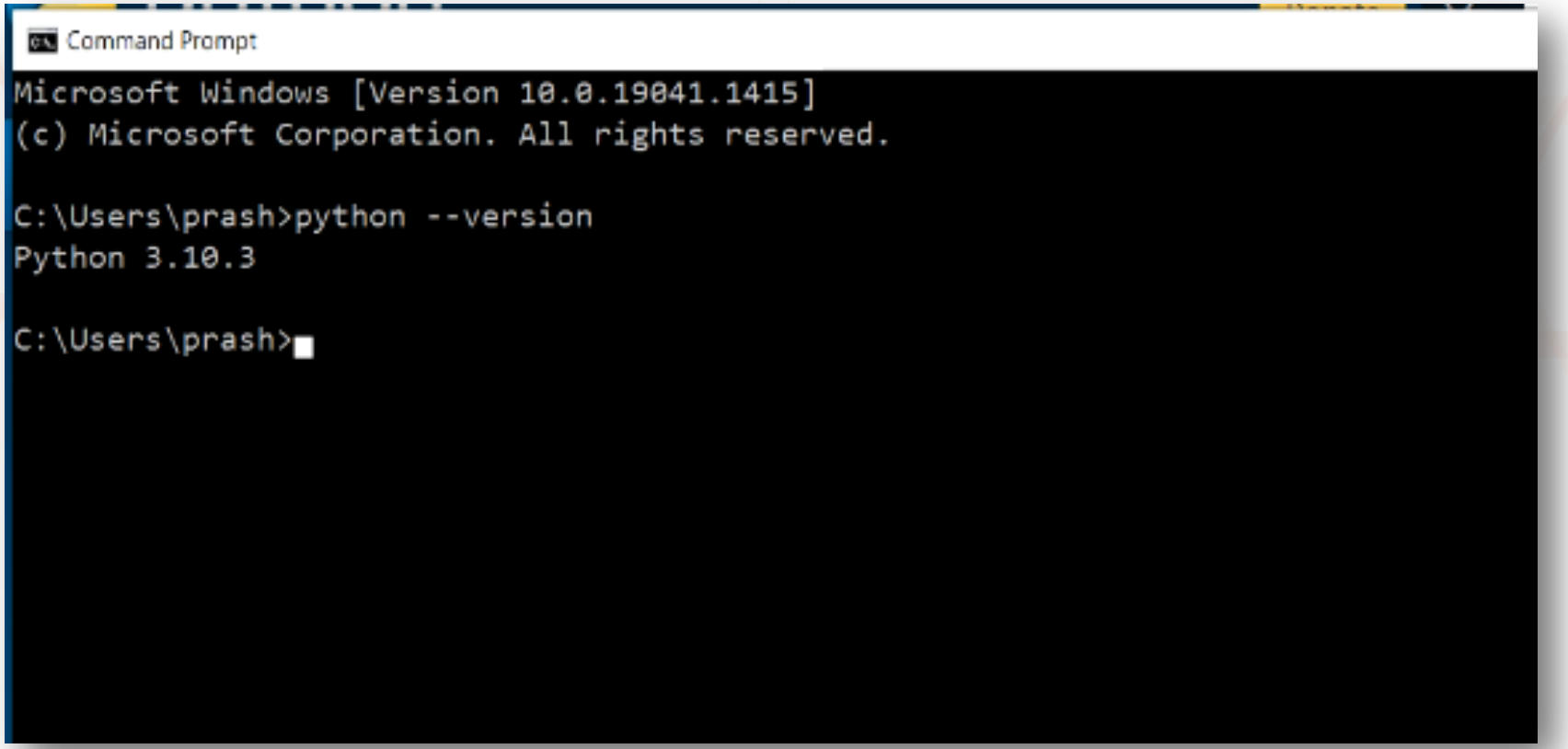
Once downloaded, click the file to run it. You should see an option to add Python to your PATH environment variable. Enable that option, so we do not have to do it manually. Then click the install now button.



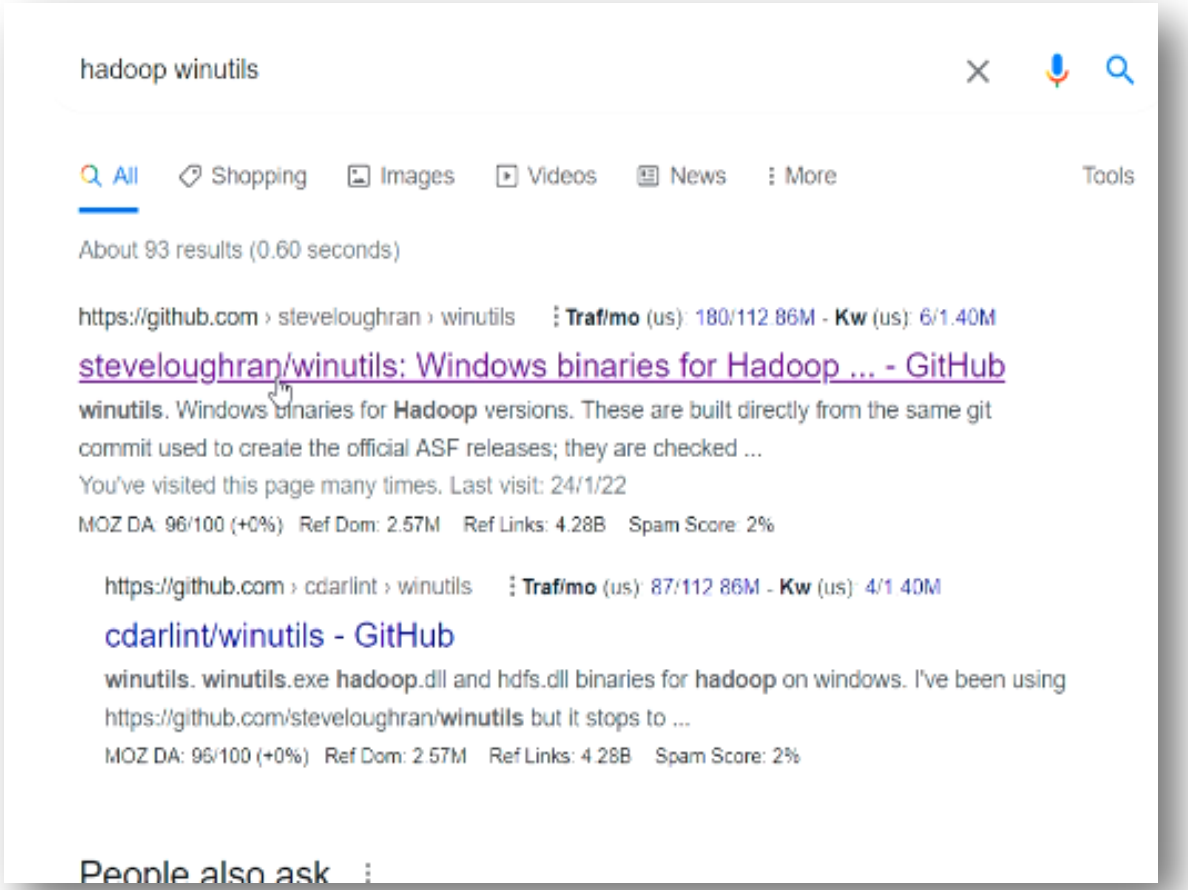
It might take a couple of minutes to complete the installation.
You might see a notification to disable the path length limit. An older windows machine allows only 255 characters for any path. The max path length limit is a problem, and this limitation is removed from the newer systems. If you are using an older system, you might see this message. I recommend that you choose this option and remove the max path length.



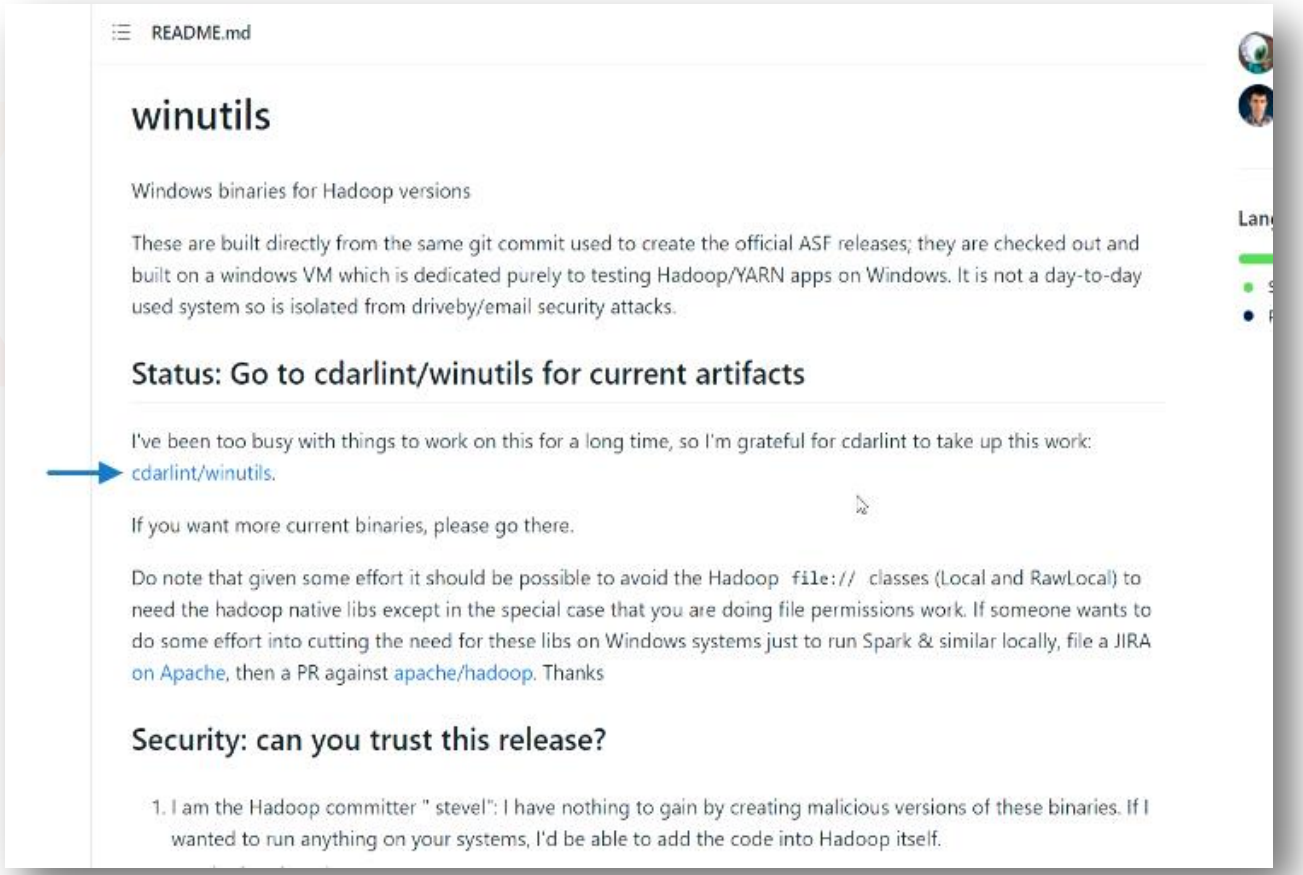
Once done with the installation, you should check it once.
Start a command prompt and run *python --version* command and make sure you see the latest version that you recently installed.



The following requirement is to setup Hadoop Winutils. Google for Hadoop Winutils, and you should see a GitHub page link. This is an old repository and is not being actively managed.

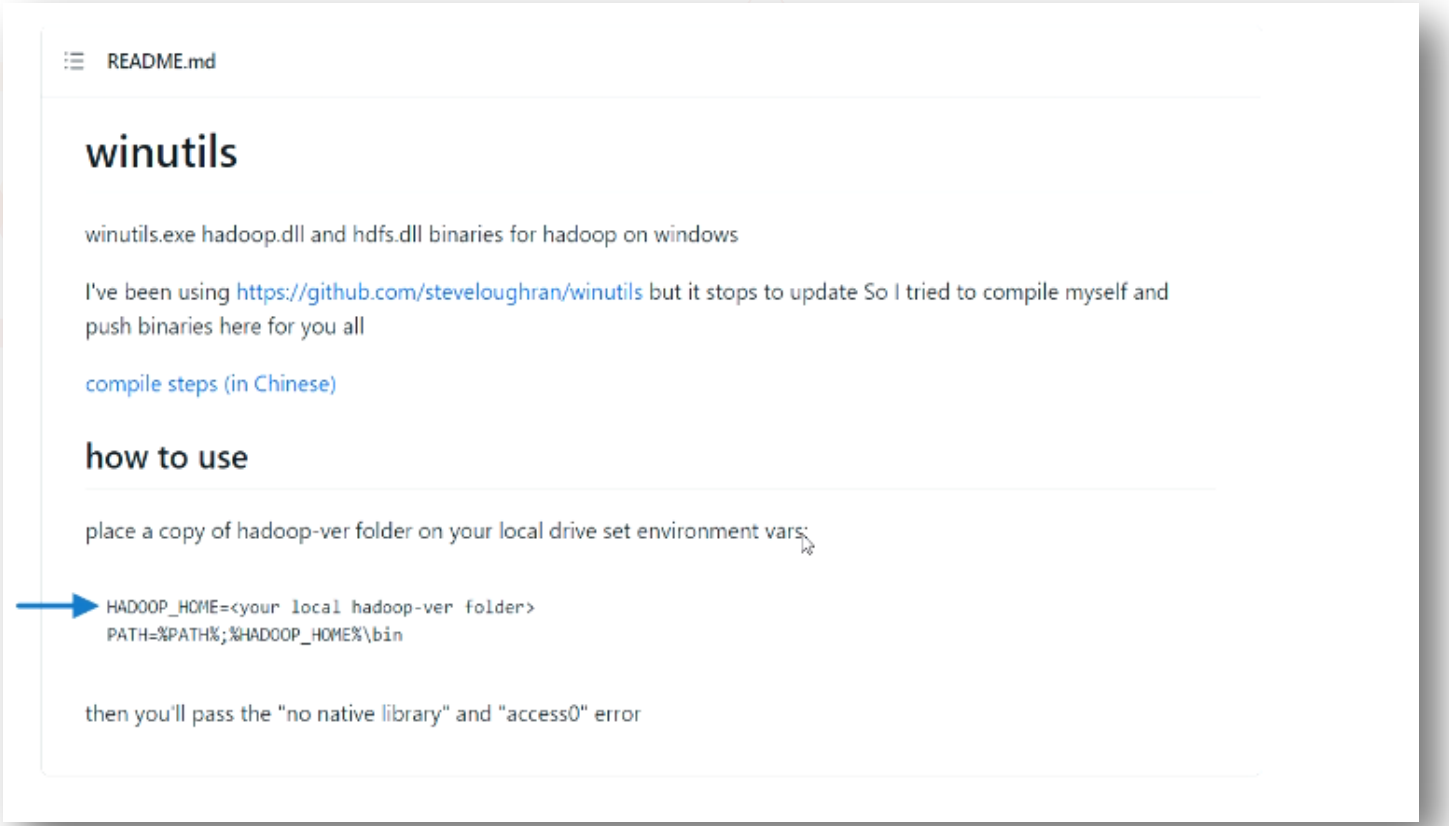


After following the GitHub link, scroll down, and you will see a link for the new repository as shown in the image below.
Follow the link to go to the current repository.

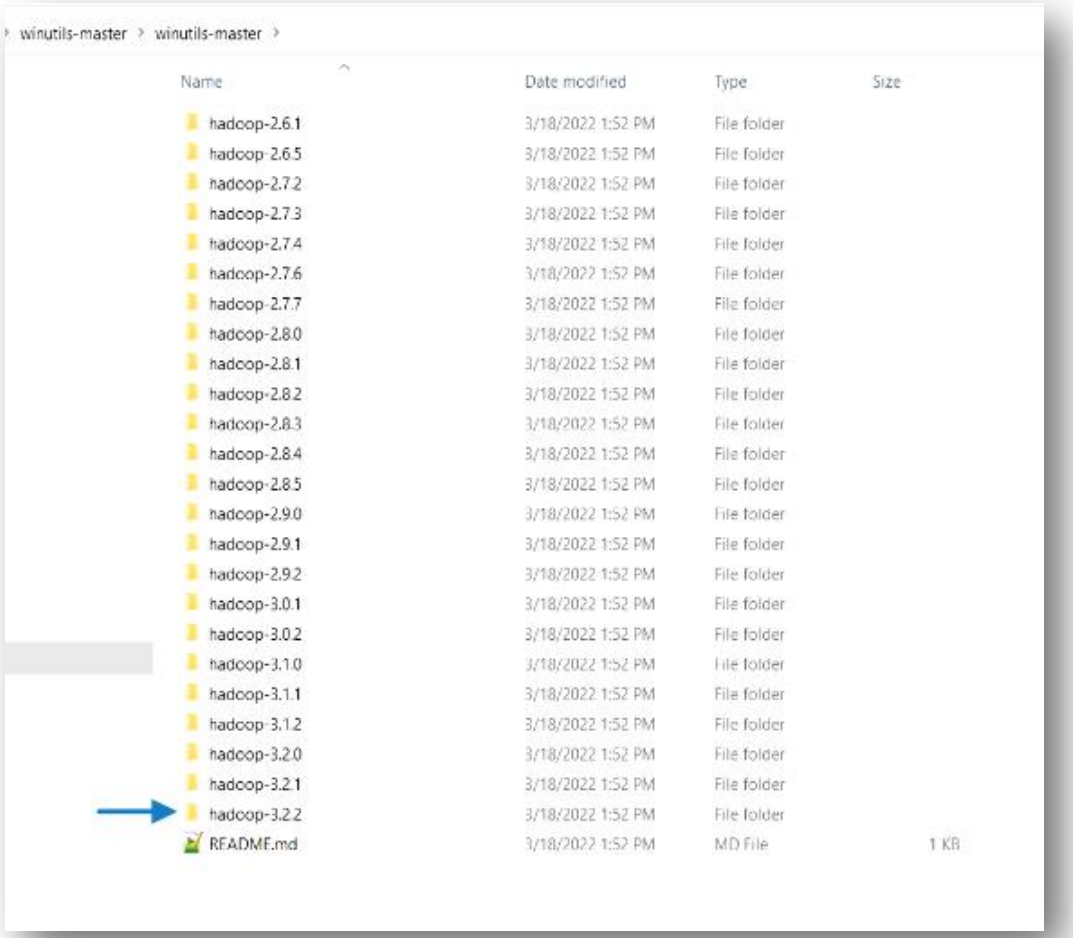


Download the current repository, and if you scroll down you will see 2 steps of using this:

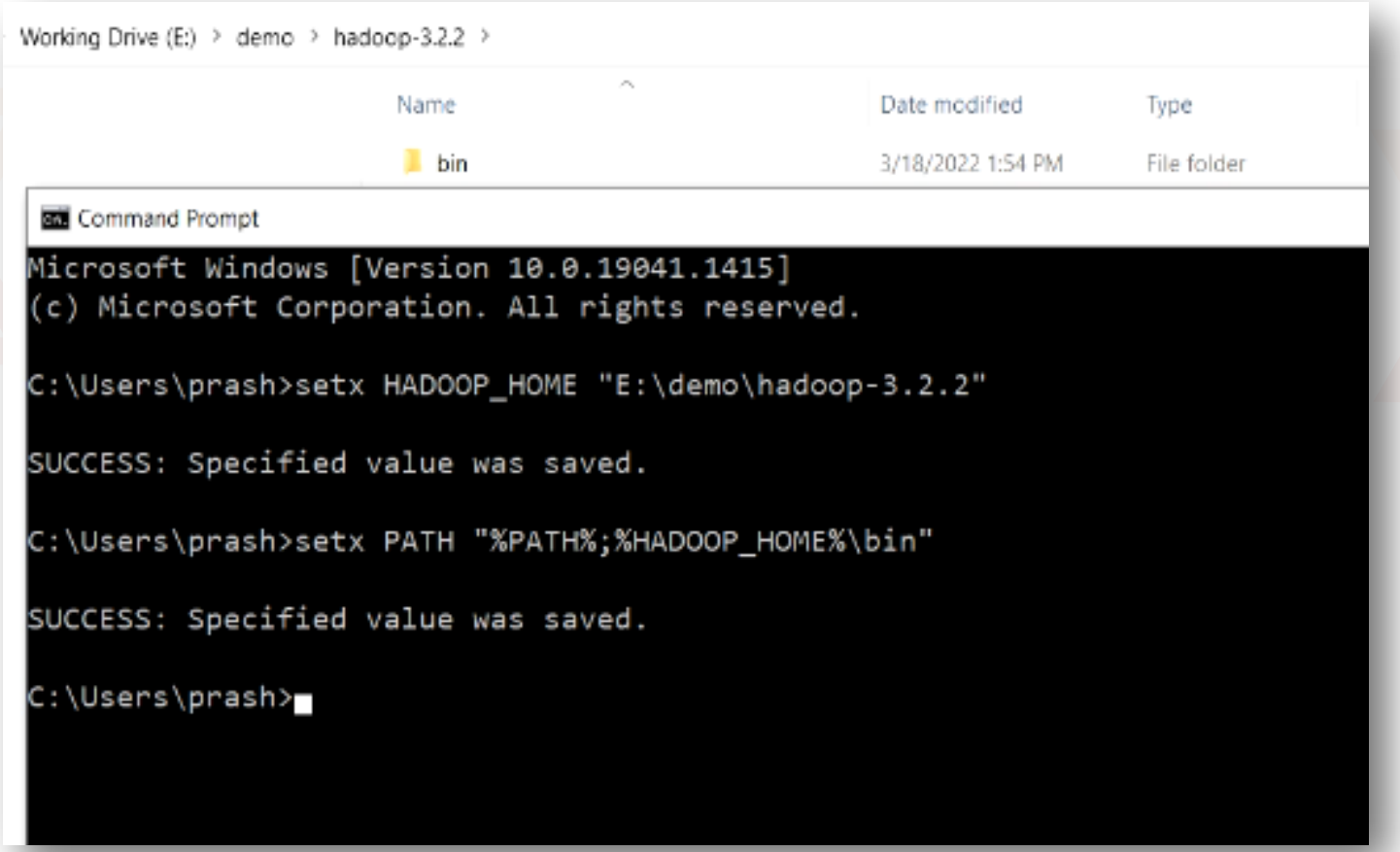
1. Setup your HADOOP_HOME environment variable
2. Include HADOOP_HOME\bin to your PATH environment variable.



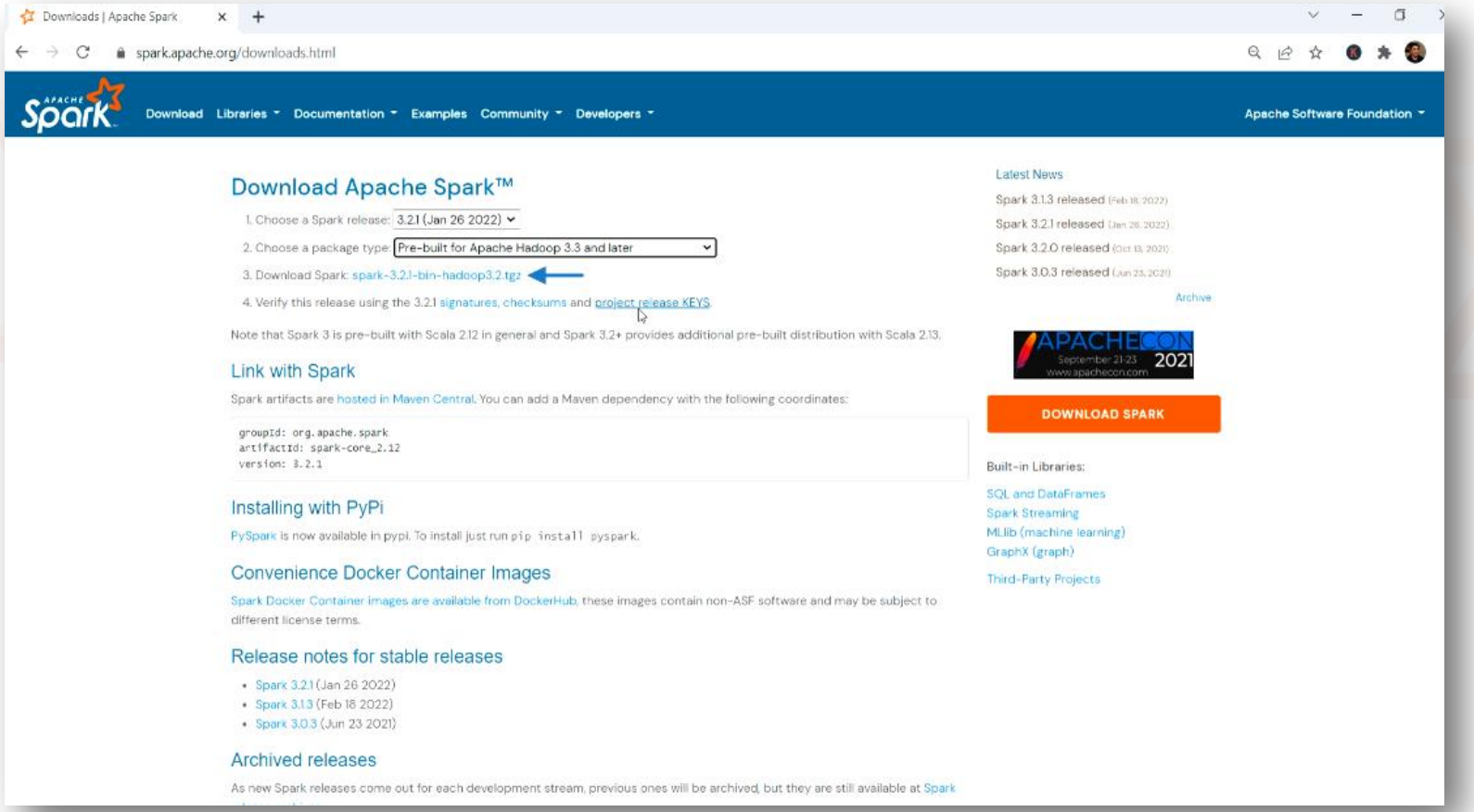
Once downloaded, go to the zip file and un-compress it. You will see the winutils-master directory. Go inside, and you will see many other folders.
So I recommend copying the latest version folder and pasting it at another permanent location. If you go inside, and you will see a bin directory.



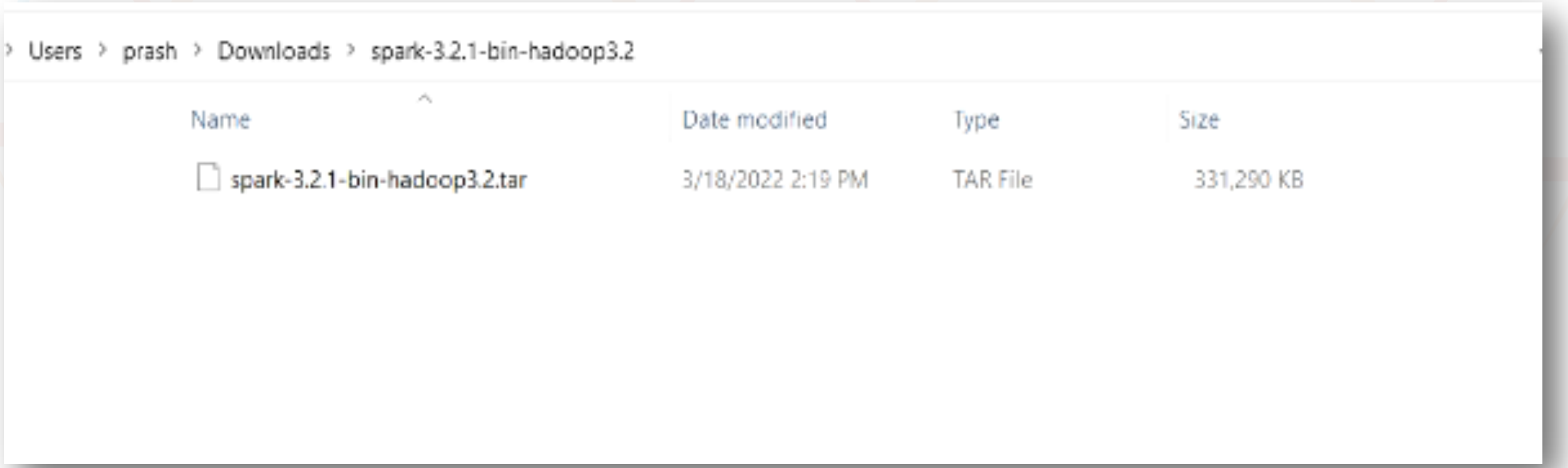
The next step is to setup your environment variable. Go the command prompt and use the setx command to set the environment variable, as shown in the image below. Make sure you see the success message.
Now add it to the PATH environment variable, as shown in the image below. Make sure you see the success message again.



The next requirement is Spark Binaries.
Visit https://spark.apache.org/ and click the download link. Then choose the latest Spark version and the Hadoop version. Now you can click the download link and start the download.

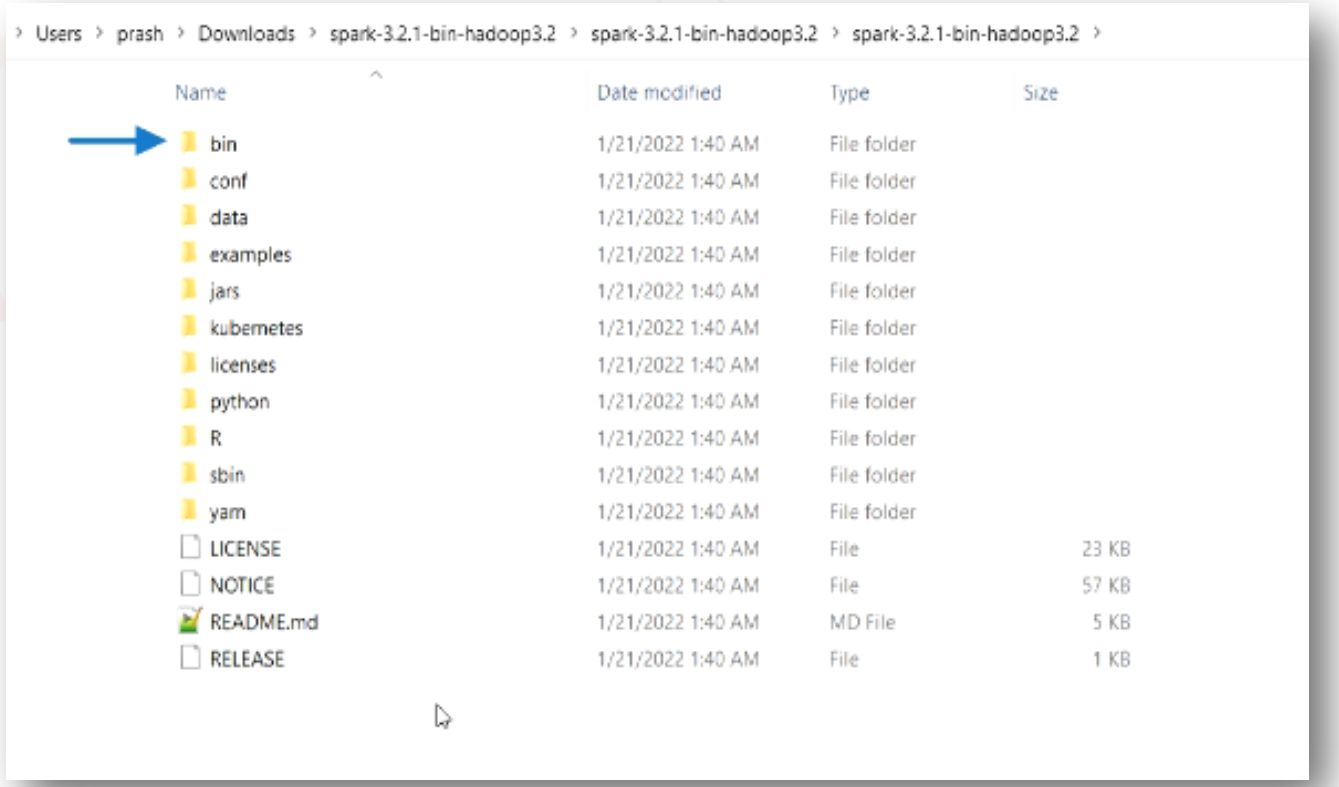


Once downloaded, go to the file and uncompress it. The Spark binary is not a zip file. It is a tgz file.
So you might need 7zip for uncompressing this tgz on the windows platform.
After compressing the file, go inside, and you will see a tar file. This one is also a compressed file. You can untar this file using the 7zip tool.

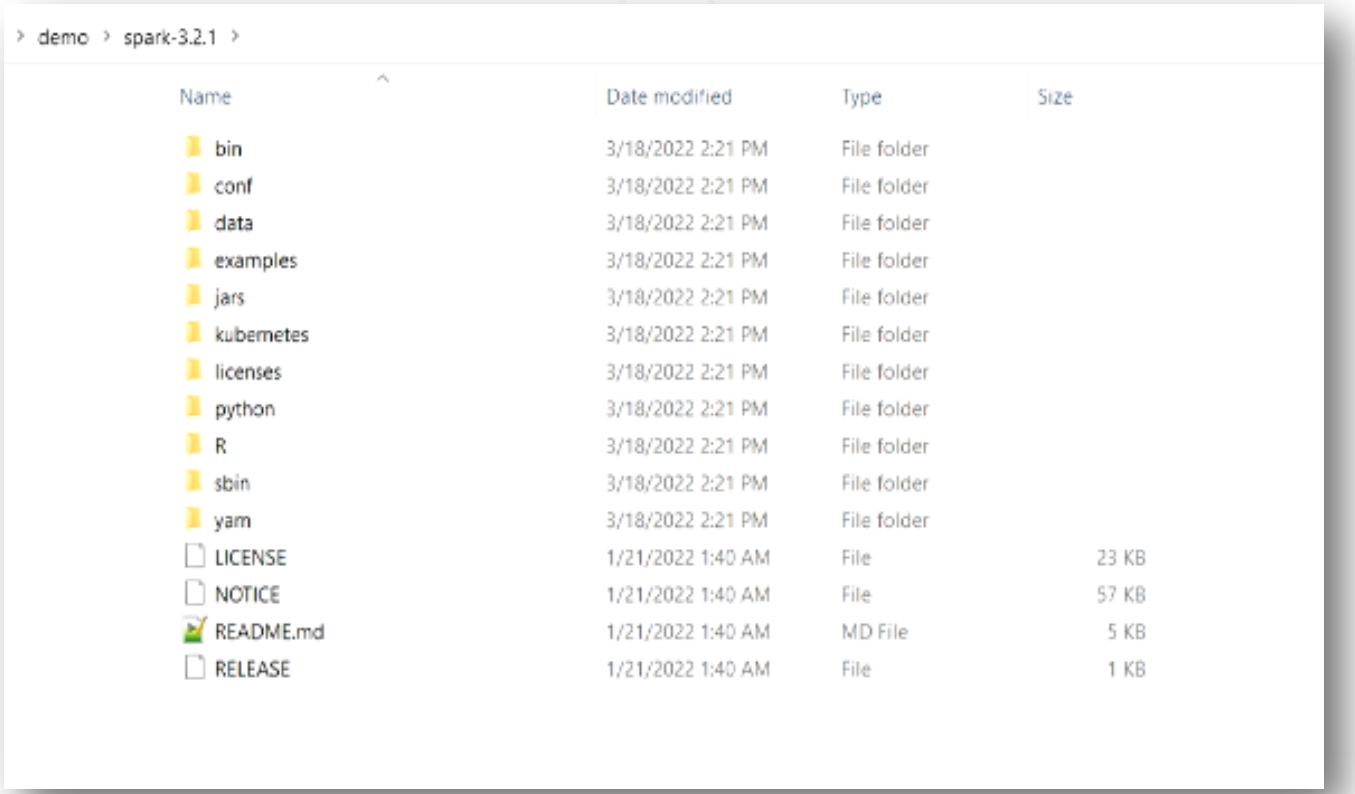


After compressing the tar file, you will see another uncompressed directory.

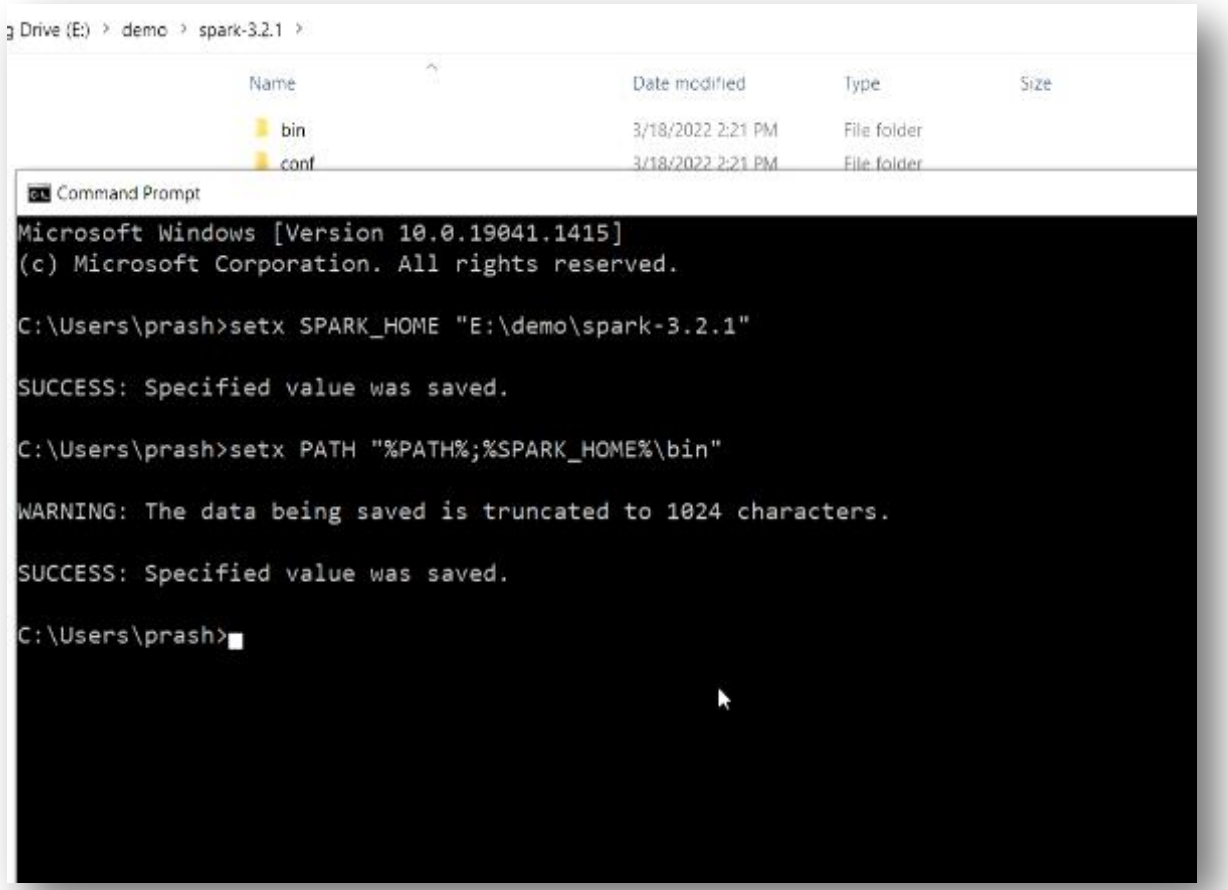
If you go inside the folder a couple of times you will the see the bin directory shown in the image below.



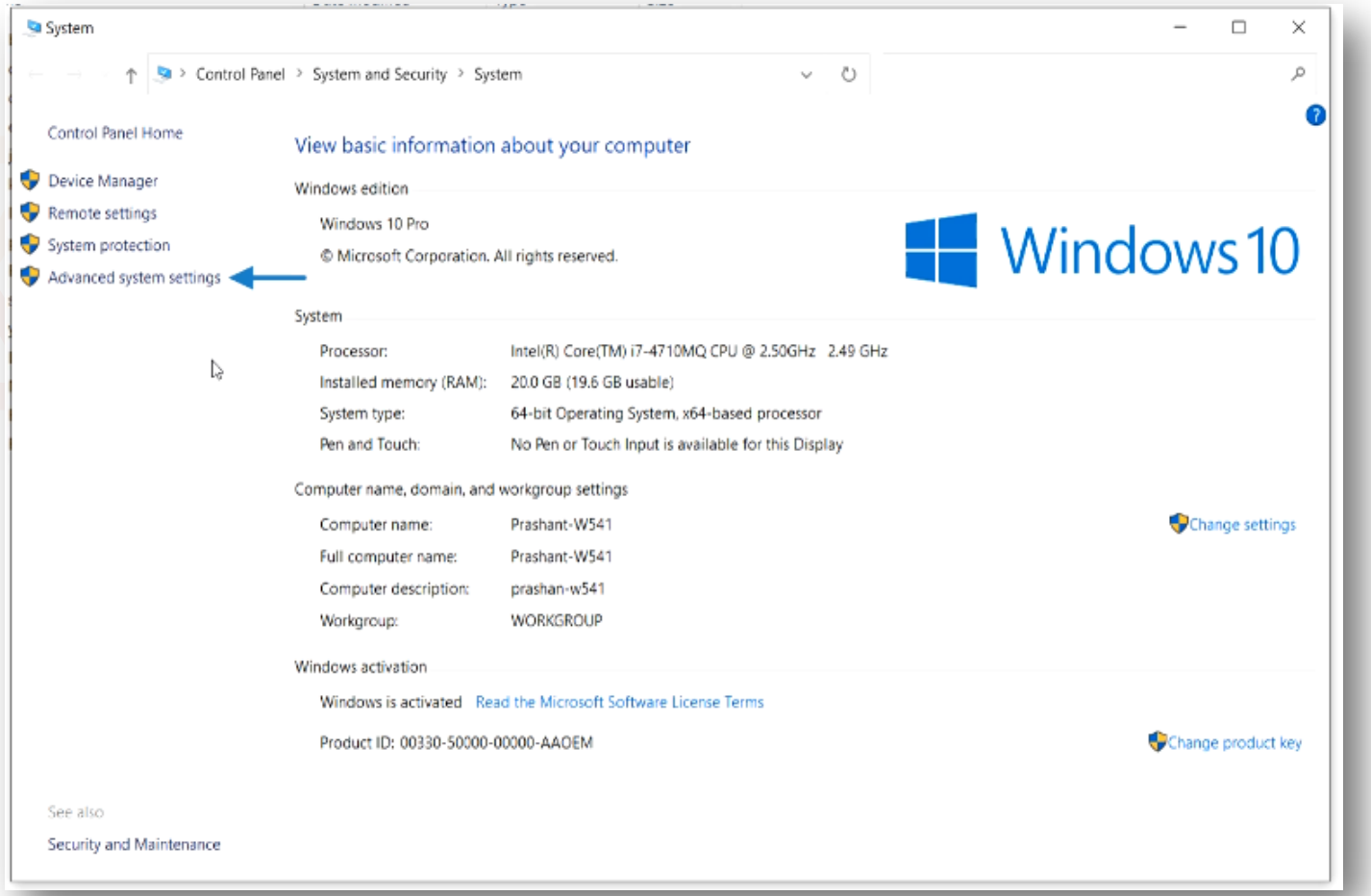
There are too much folders inside folders, so you can copy the parent folder and paste it to a permanent location. So the image shown below is your spark home. But we must set the environment variables.



Start your command prompt and use the setx command to set the SAPRK_HOME environment variable. Make sure you see the success message. Then you should also add the SPARK_HOME\bin to our PATH environment variable. You might also see a warning that data being saved is truncated. If you see it, your PATH environment variable is not set correctly.

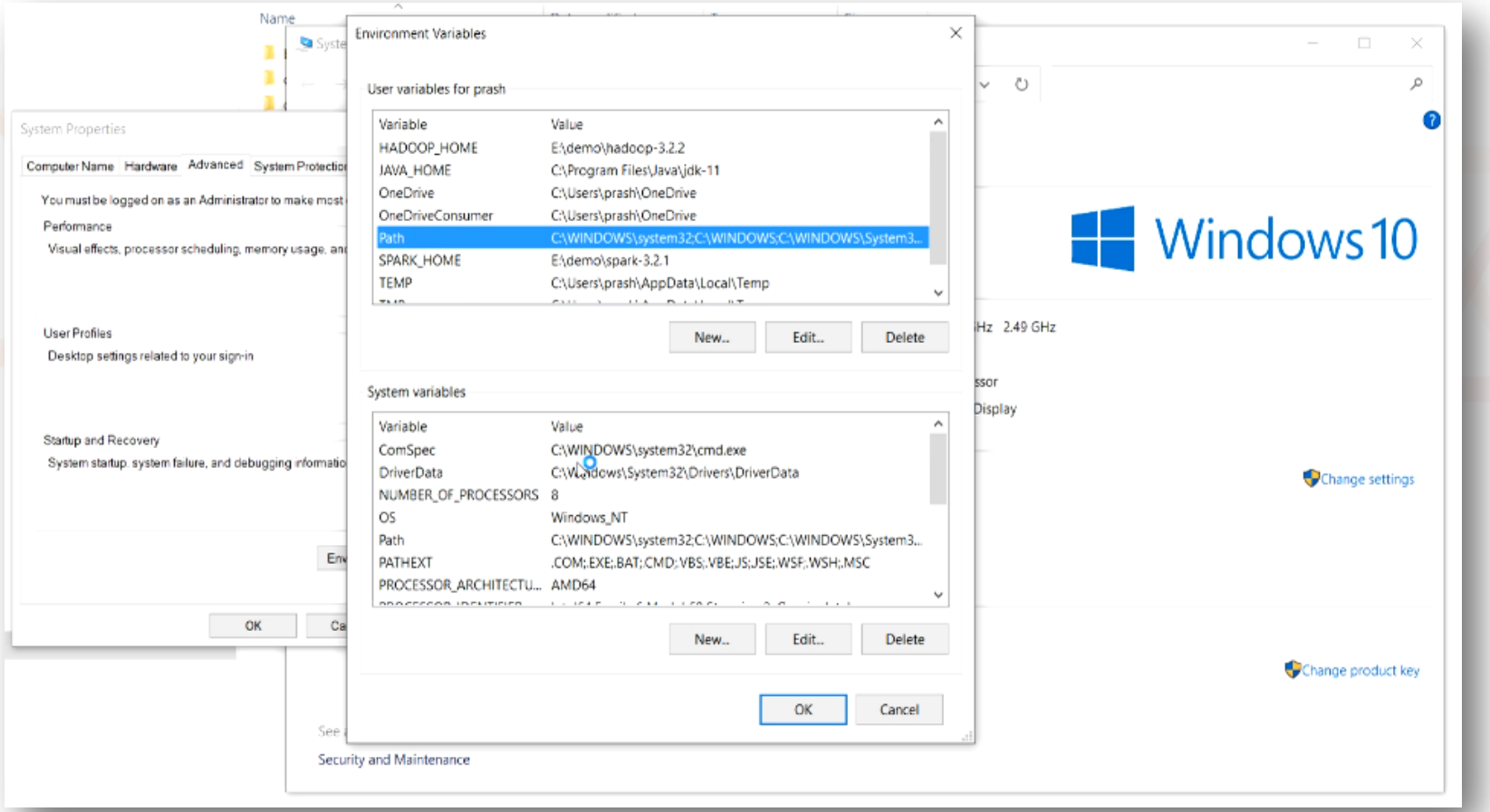


You can add a more extended PATH environment variable using the Windows UI. Right-click on This PC and choose Properties. You will see a window as shown in the image below. Then, go to advanced system settings and then follow the Environment Variable button.

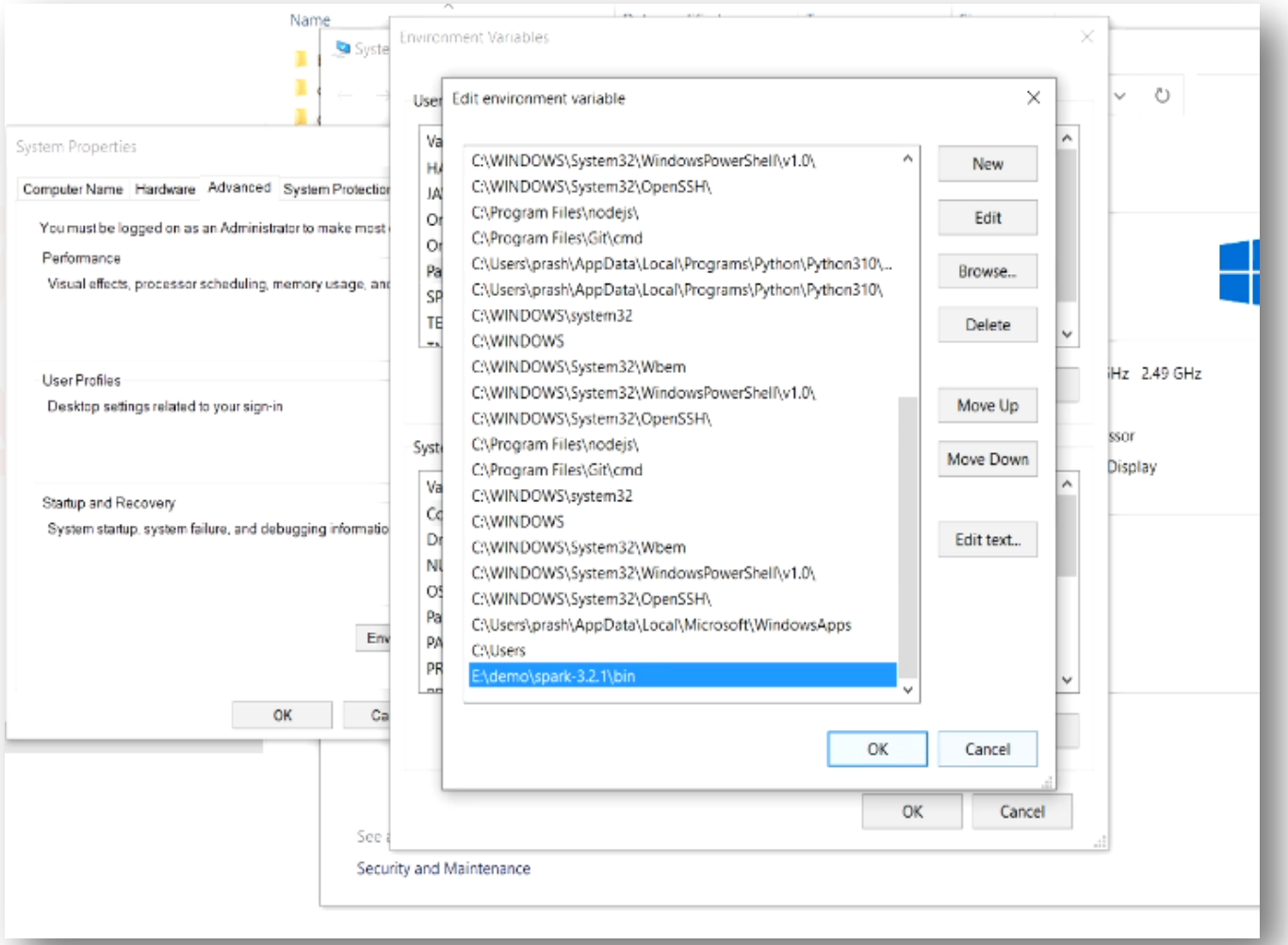


You will land on the window shown below. You will see the Path environment variable.

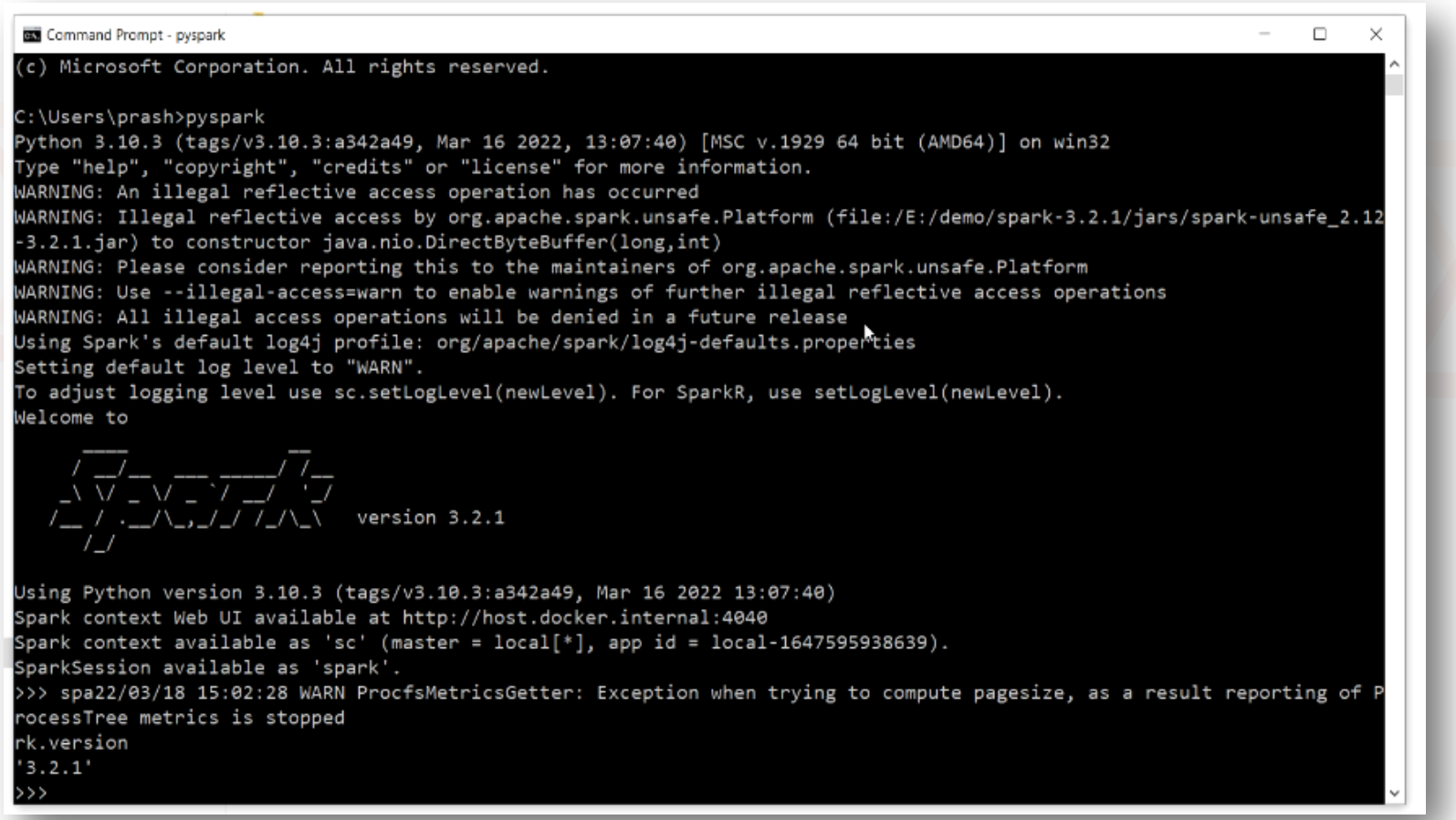
Edit it. You can see a long list of things included in the path ENVIRONMENT Variable.



Click the new button, and paste the Spark Home directory location including the “\bin” at the end. That's all. Click Ok and close everything.

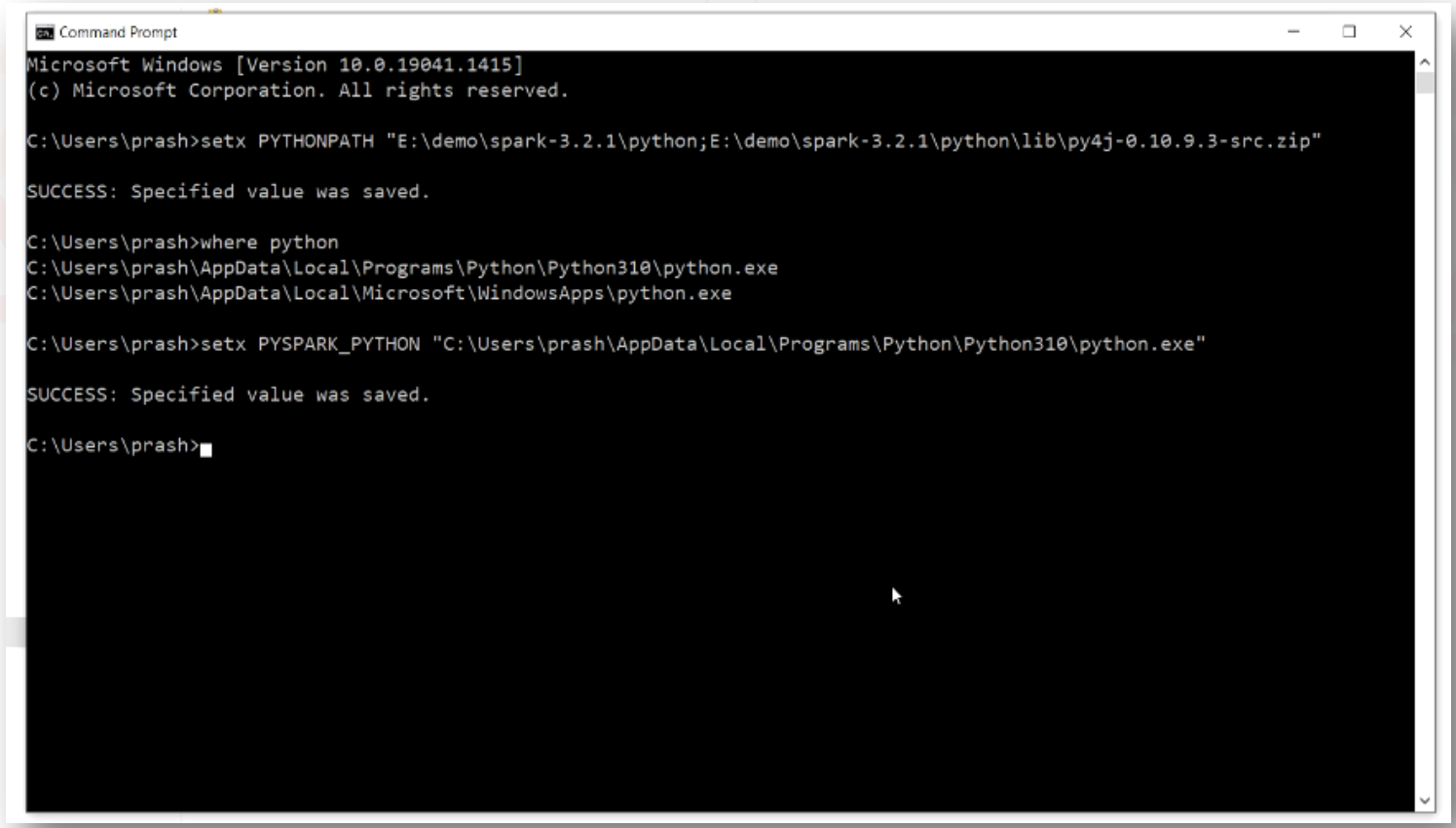


To test if Spark setup is successful, go to command prompt and type “pyspark”
You should see some messages and finally a command prompt. If you see a command prompt and no error messages, you are good to go. You have Apache Spark running on your local machine. You might see a warning stating Exception when trying to compute the page side.



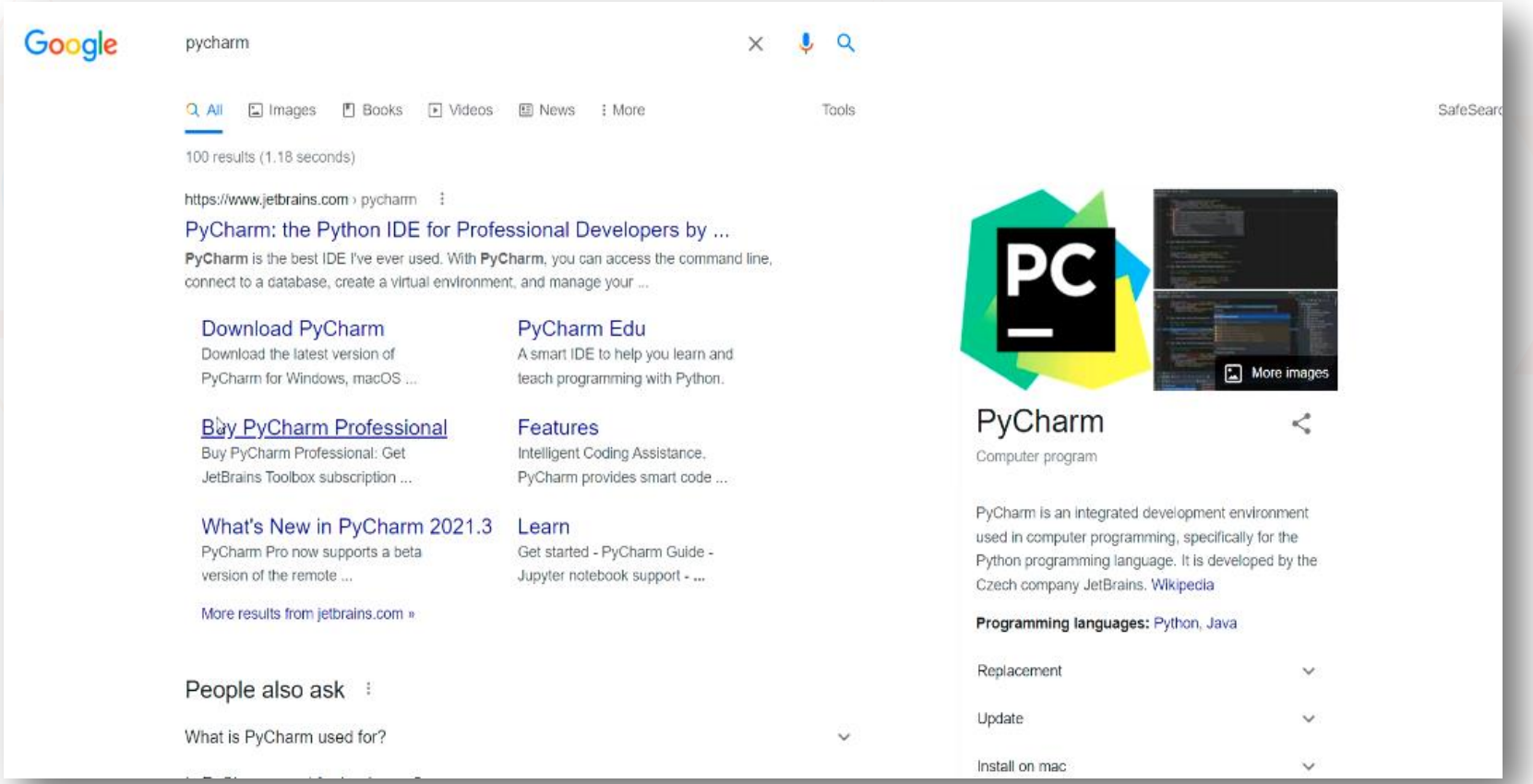
The next step is to set up the following two environment variables that can save you from seeing unnecessary errors while working with Spark on your local machine.

1. PYTHONPATH – If you go back to your Spark home directory, you will see a Python directory there. Go inside that python directory and copy that path. Then, Paste it into the PYTHONPATH variable value. Make sure to take the absolute path to the py4j zip file and paste it to PYTHONPATH.
2. PYSPARK_PYTHON - You can use the where-command to find your Python installation location. And use the location to set PYSPARK_PYTHON variable.

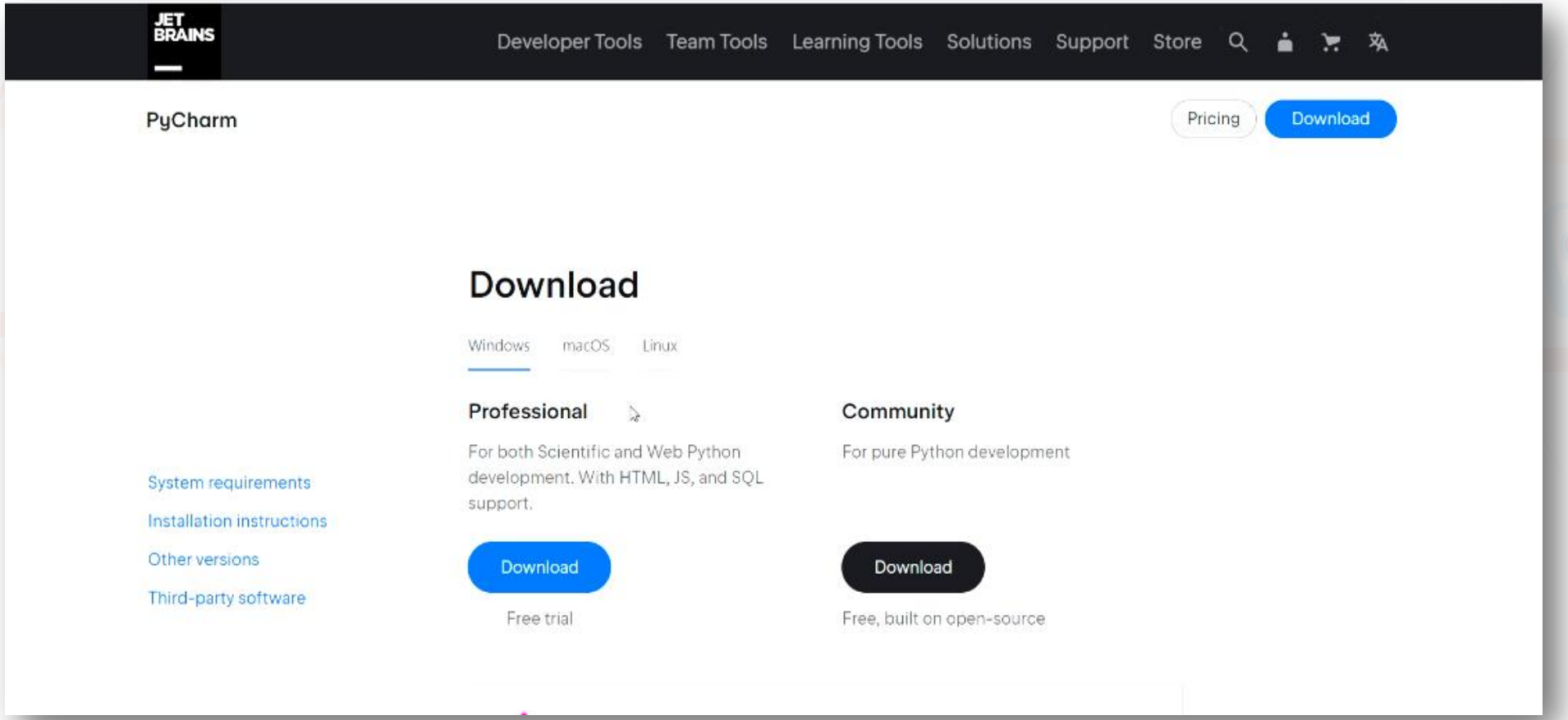


But writing code on command prompt if not a very good approach. We do not want to develop large Spark applications working on the command prompt. I need an IDE for being productive and managing large project development. PyCharm is the most popular IDE for Spark development.

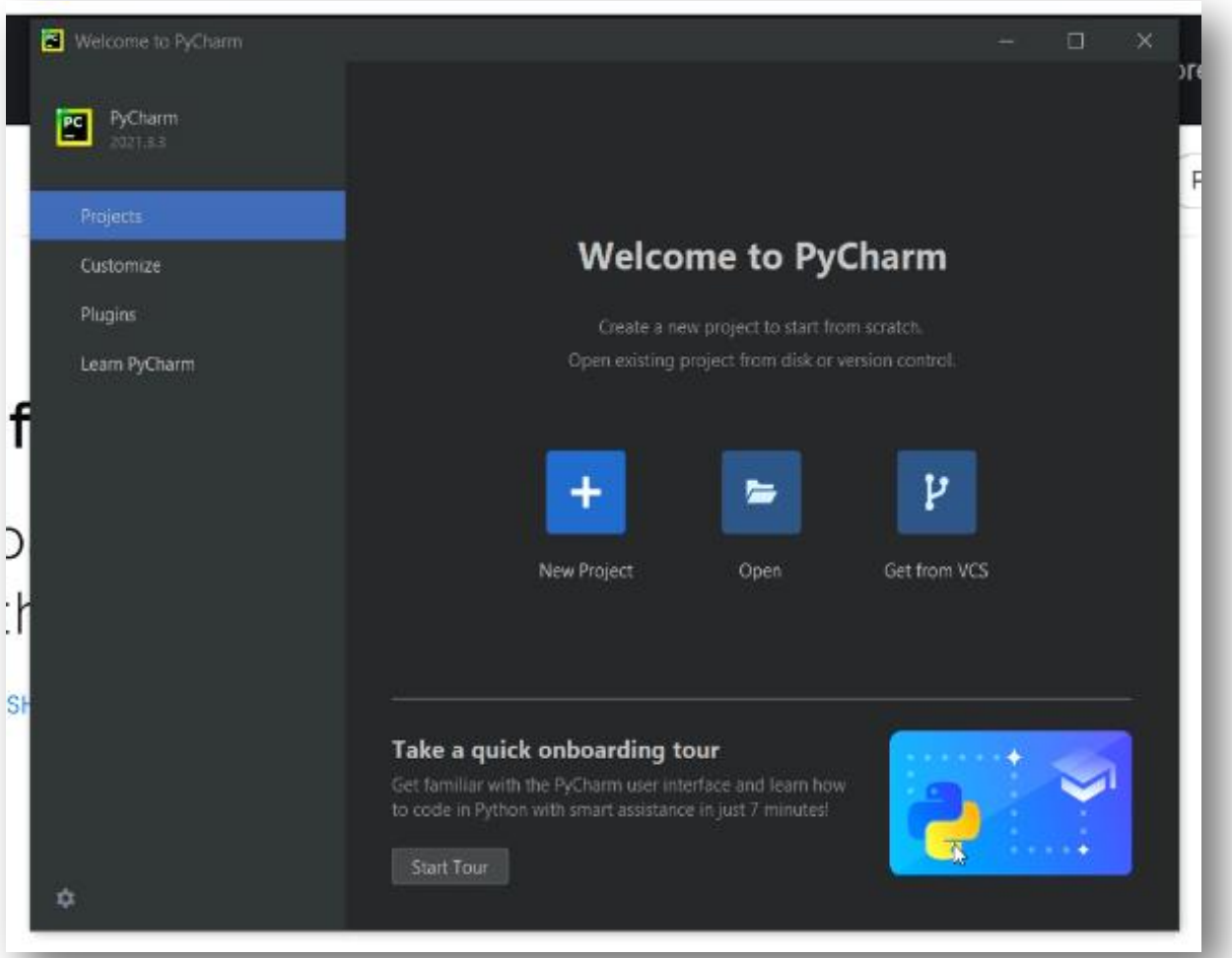
Start your browser and search for PyCharm. And click the link shown jetbrains.com.



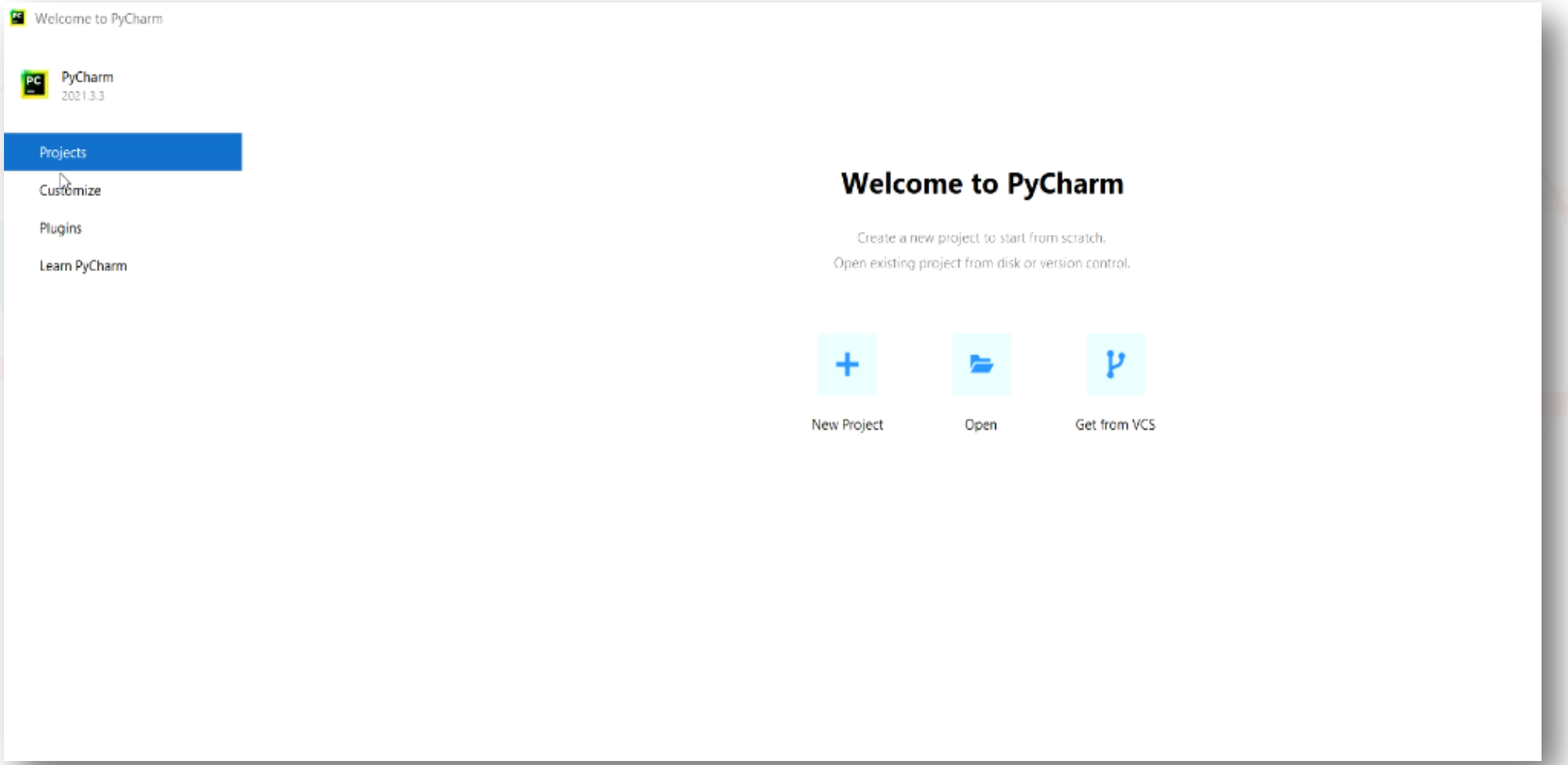
Click on the downloads button, and select the community edition. The community edition is a free opensource version of PyCharm. And the community edition is more than enough for our purpose.



Execute the downloaded installer and follow the on-screen default instructions. Once installed, you can start your PyCharm IDE. And you will see the welcome screen as shown below. You can go to customize option to set a lighter theme.



You can come to the Project menu, and you are ready to create your first Spark project.



Thank You

ScholarNest Technologies Pvt Ltd.

www.scholarnest.com

ScholarNest

Spark Azure Databricks

Databricks Spark Certification and beyond

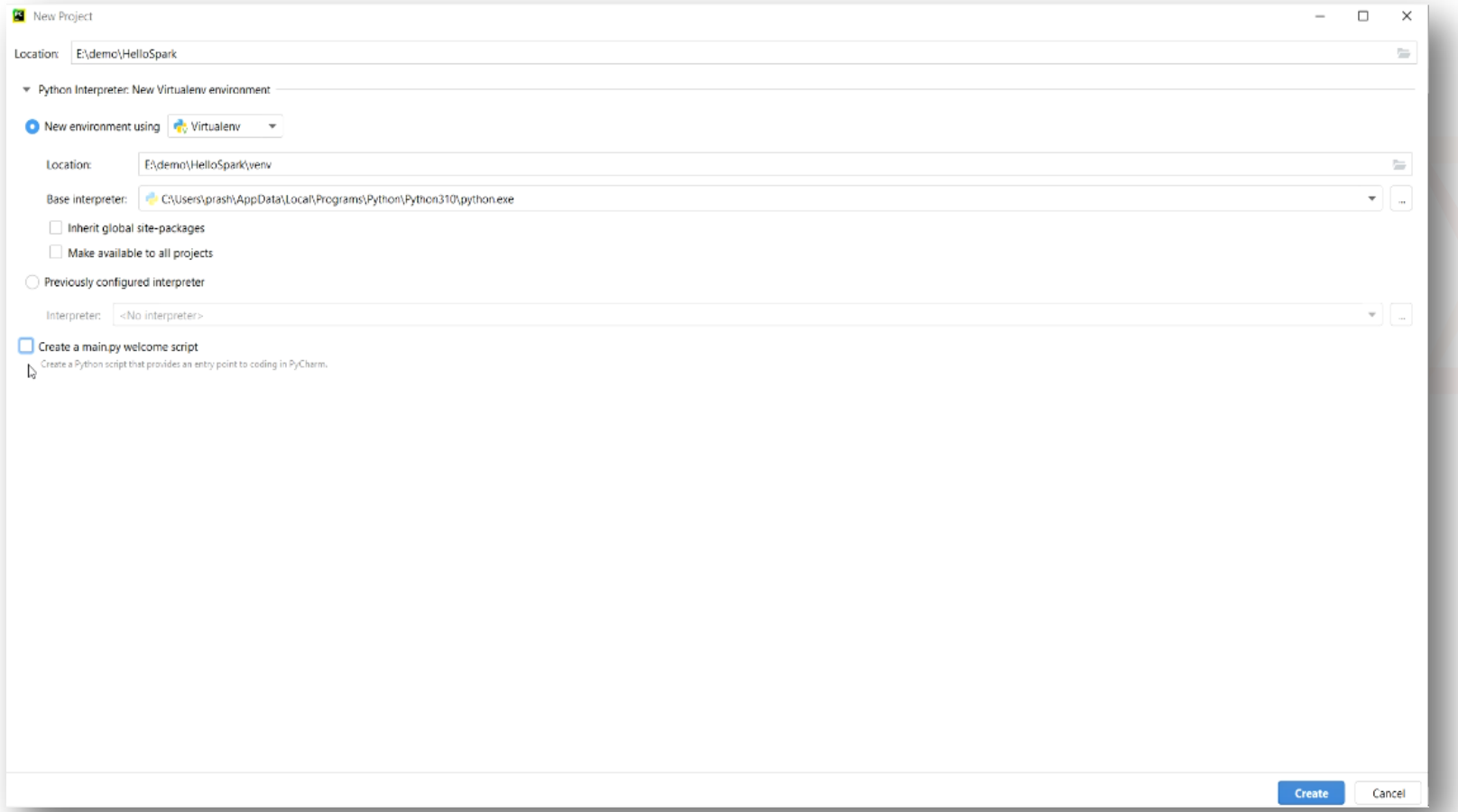
Instructor: Prashant Kumar Pandey

Absolute Beginner to Specialization in Apache Spark and Azure Databricks

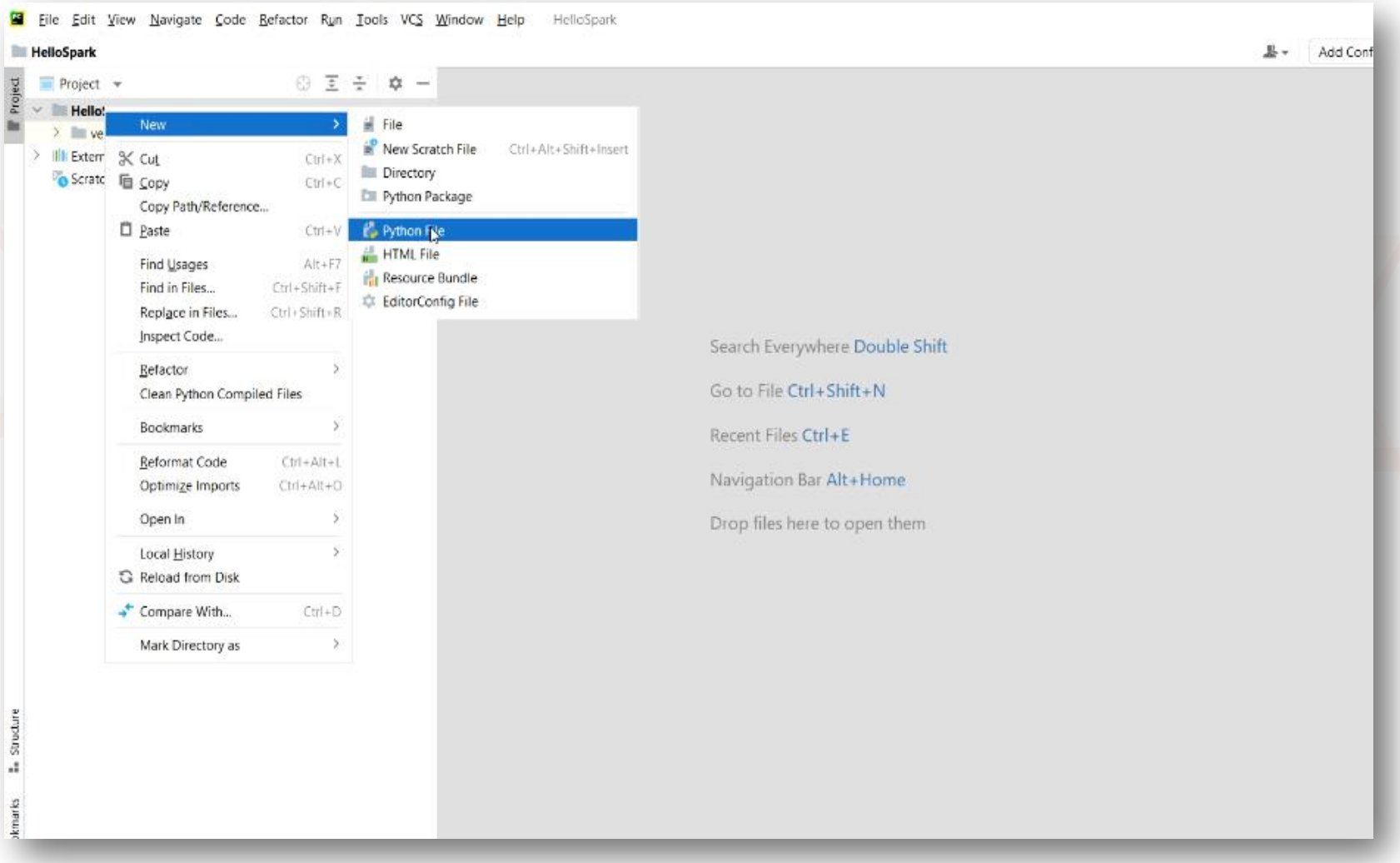
First Spark Application using IDE

Start your PyCharm IDE and create a new project.

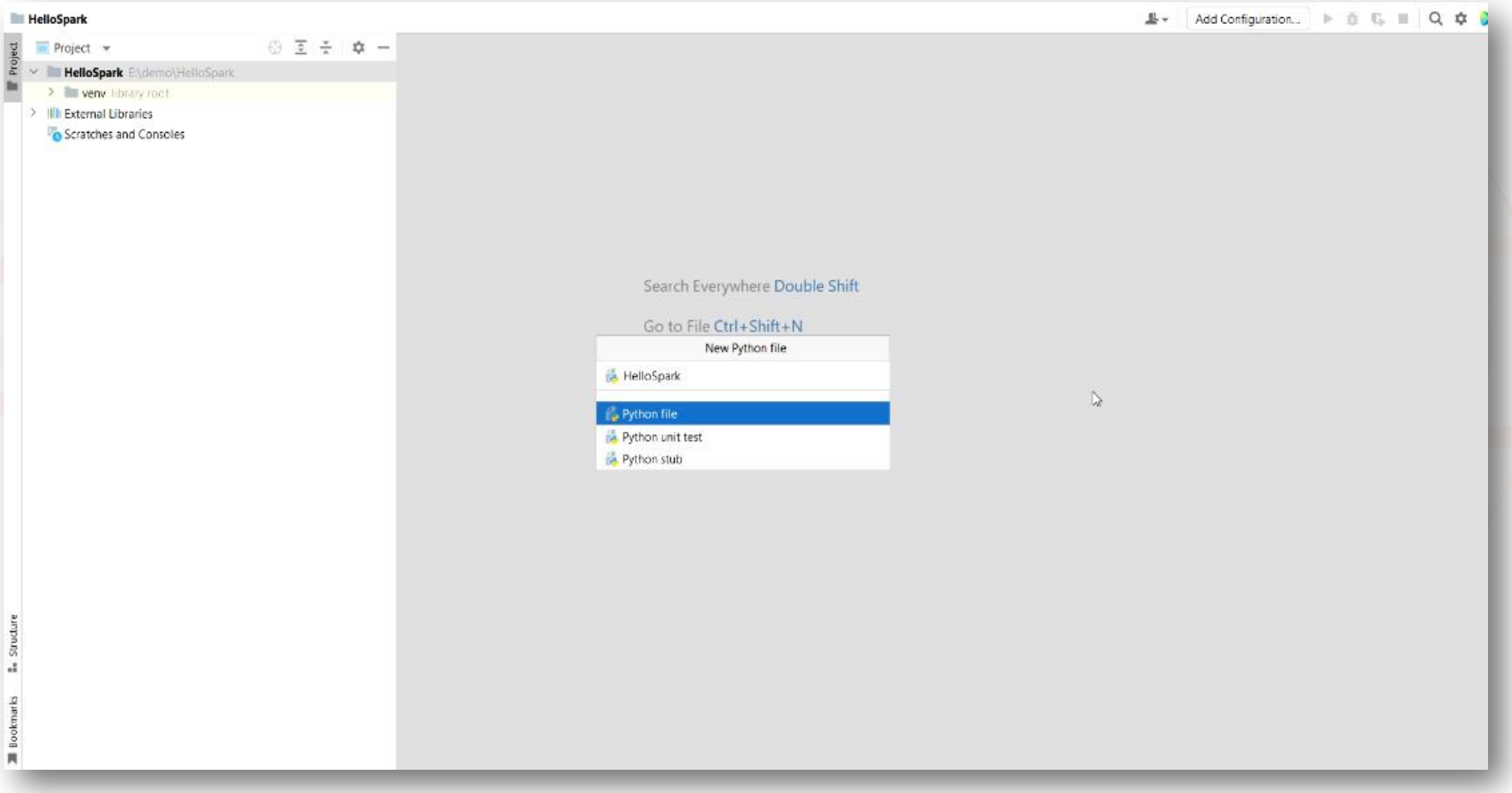
1. Set your new project location by hitting the browse button and choosing your working directory. Also type a new project directory location.
2. Ensure you have the latest version of Python selected in the Python Interpreter dropdown.
3. Uncheck all other options. We do not need anything else.
4. Use VirtualEnv for setting up our project.
5. Hit the create button.



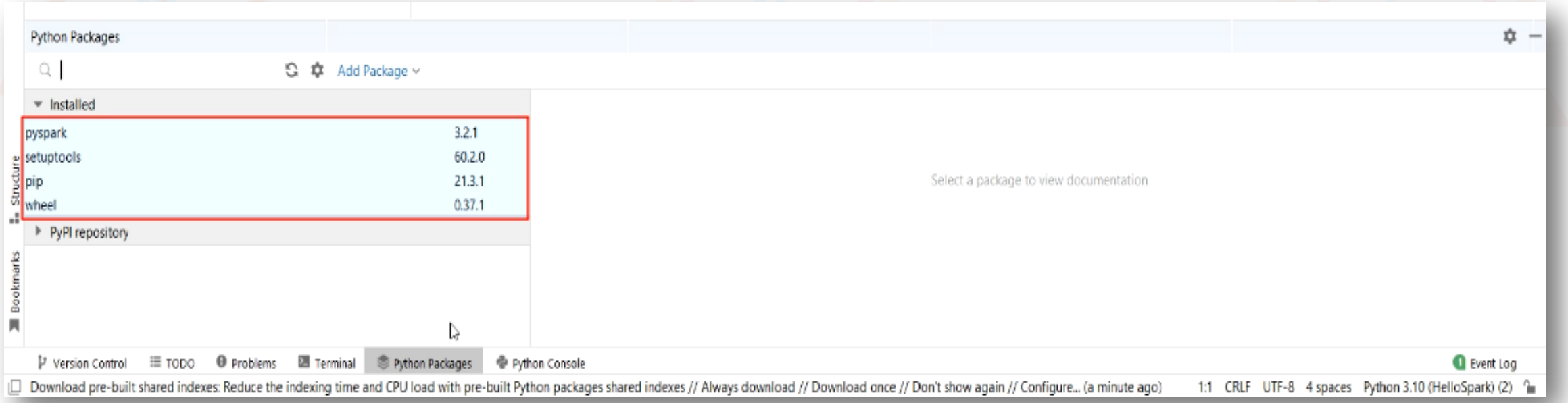
Once your blank project setup is done. You can create a new Python file in your project directory as shown in the image below.



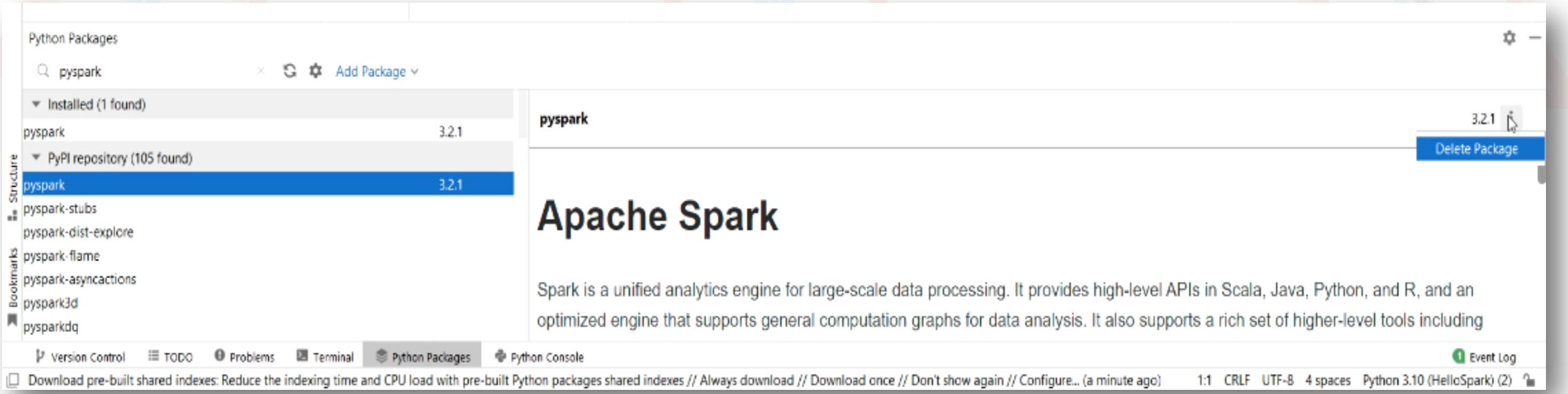
Give a name to your Python file.



Make sure you have the PySpark package installed in your project's VirtualEnv. You check that by clicking the python packages window at the bottom of your IDE. And you should see your installed packages there as shown in the image below. I already have the pyspark package installed.



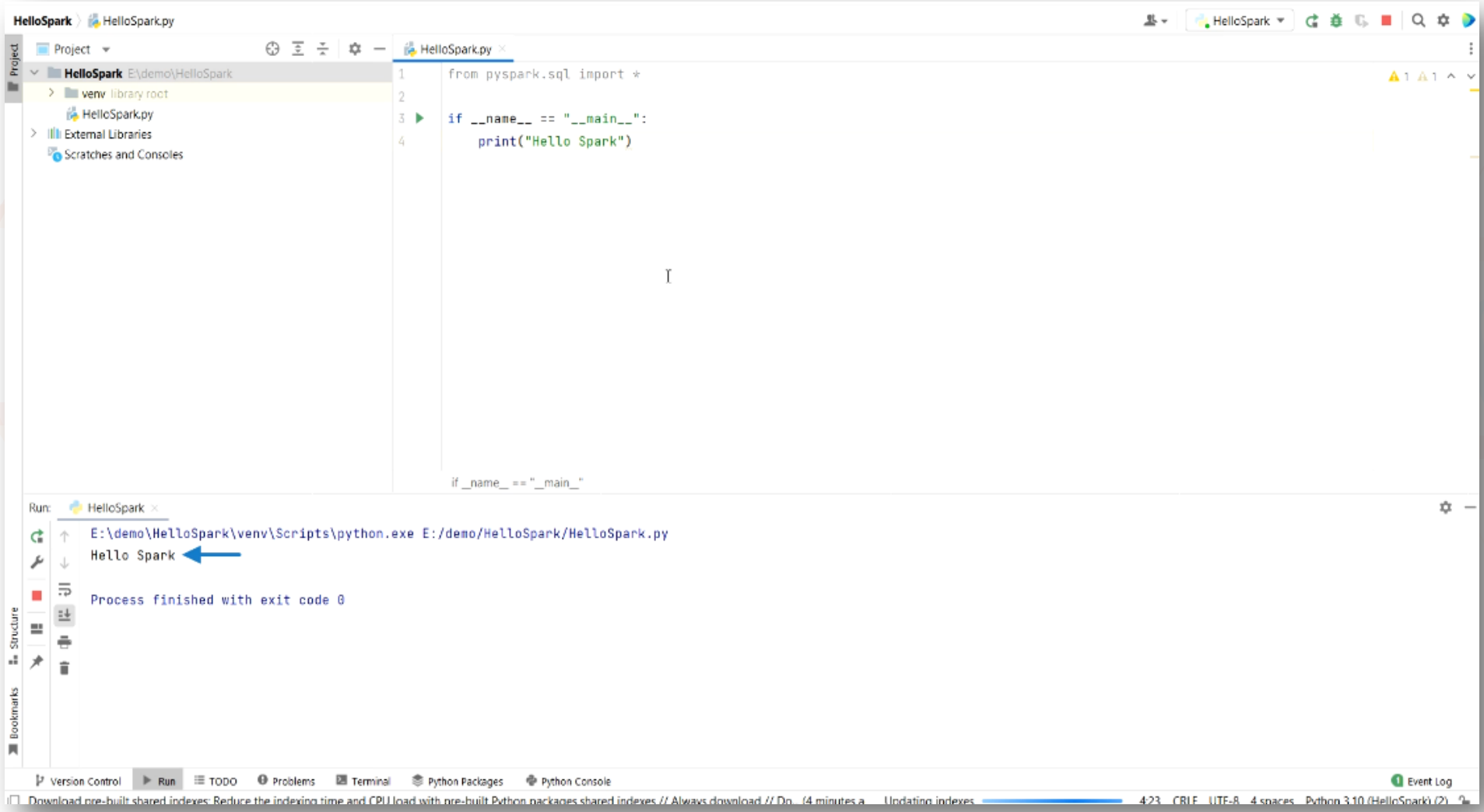
However, if you do not see PySpark in the list of your installed packages, search for it. Select the pyspark package under the PyPI repository. Come to the right side, three dots, and you should see an option to install.
I see a delete option here because I have installed it already.
But you may see an install button to install the selected package.
Install it if you do not have it already installed.



We haven't started learning Spark programming yet. So I am not expecting you to understand the code. However, I want to create a super simple Spark application and run it from the IDE. In the screenshot shown below, we have a simple Hello Spark Python code, it is not a Spark program.

Run your program as shown below.

It worked! I can see the output in the console below.

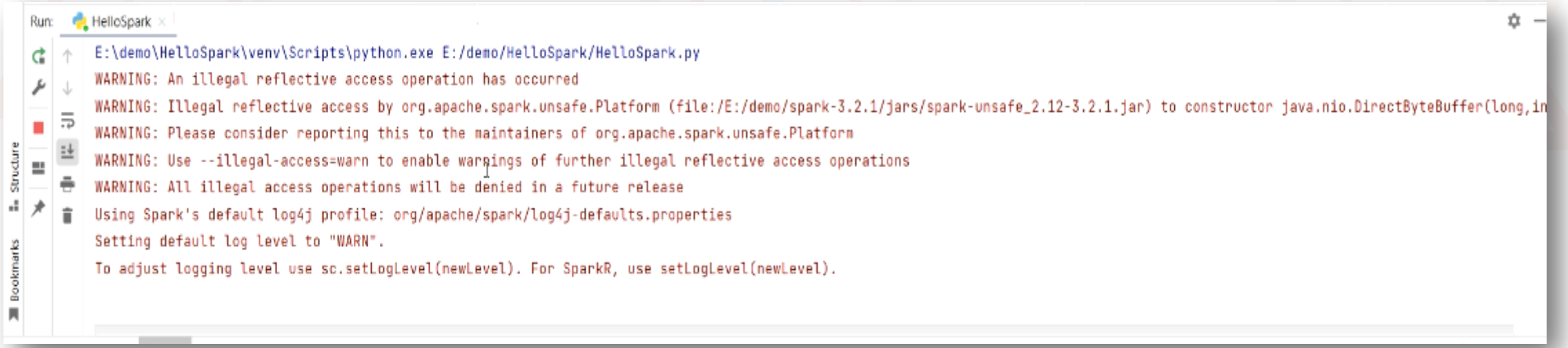


Now, let us look at a Spark code.
Here is my super simple Spark application. I am creating a Spark Dataframe using a data list. I am giving some column names to list items and executing the show() method on my Dataframe.
Do not stress yourself to understand the code.

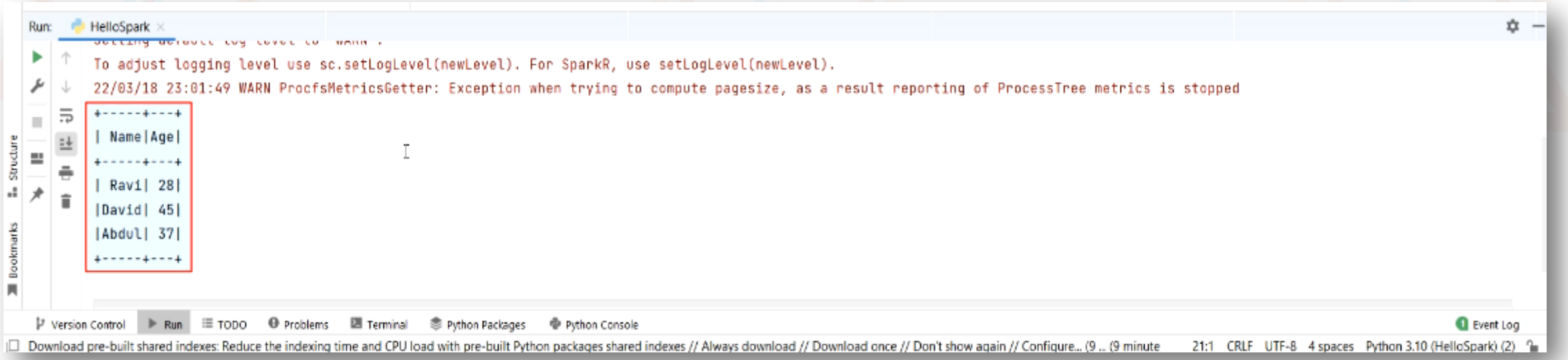
HelloSpark.py

```
from pyspark.sql import *

if __name__ == "__main__":

    spark = SparkSession.builder \
            .appName("Hello Spark") \
            .master("local[2]") \
            .getOrCreate()

    data_list = [("Ravi", 28),
                 ("David", 45),
                 ("Abdul", 37)]

    df = spark.createDataFrame(data_list).toDF("Name", "Age")
    df.show()
```

You will see some warnings when you run the code. But you can ignore those warnings.



Finally, you will see Dataframe output.
We managed to run a super simple Spark application on our local computer.



Thank You

ScholarNest Technologies Pvt Ltd.

www.scholarnest.com