Hands on Lab: Getting Started with Apache Spark on Watson Studio (20 mins)

Objectives

After completing this lab you will be able to:

- Use your IBM Cloud account to explore and create resources.
- Create a Watson Studio Service instance.
- Create a Jupyter Notebook on Watson Studio with a Apache Spark + Python kernel
- Run the notebook and inspect the outputs

> **Note:** If you already have an IBM Cloud account, please skip Exercise 1. Additionally if you also have a Watson Studio service created, skip Exercise 2 as well.

Exercise 1: Create an IBM Cloud Account

Follow the steps in Hands on Lab: IBM Cloud Service Creation to create an IBM cloud account.

Exercise 2: Create an instance of Watson Studio service

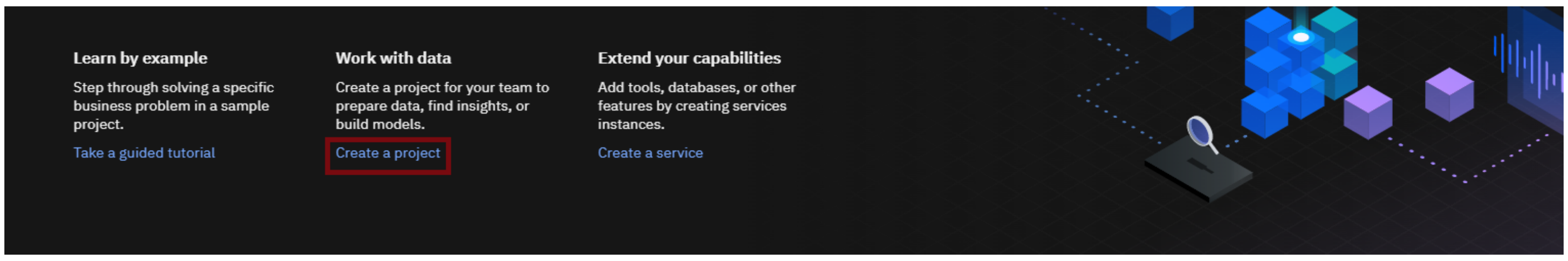
Follow the steps in Hands on Lab: IBM Watson Setup to create a Watson Studio service and launch it.

Exercise 3: Create a Spark + Python Jupyter notebook on Watson Studio

Once the Watson Studio service has been created and Watson Studio has been launched via the Cloud Pak for Data dashboard.

Step 1: Creating a Watson Studio Project:

Click on **Create a project:**



Learn by example

Step through solving a specific business problem in a sample project.

Take a guided tutorial

Work with data

Create a project for your team to prepare data, find insights, or build models.

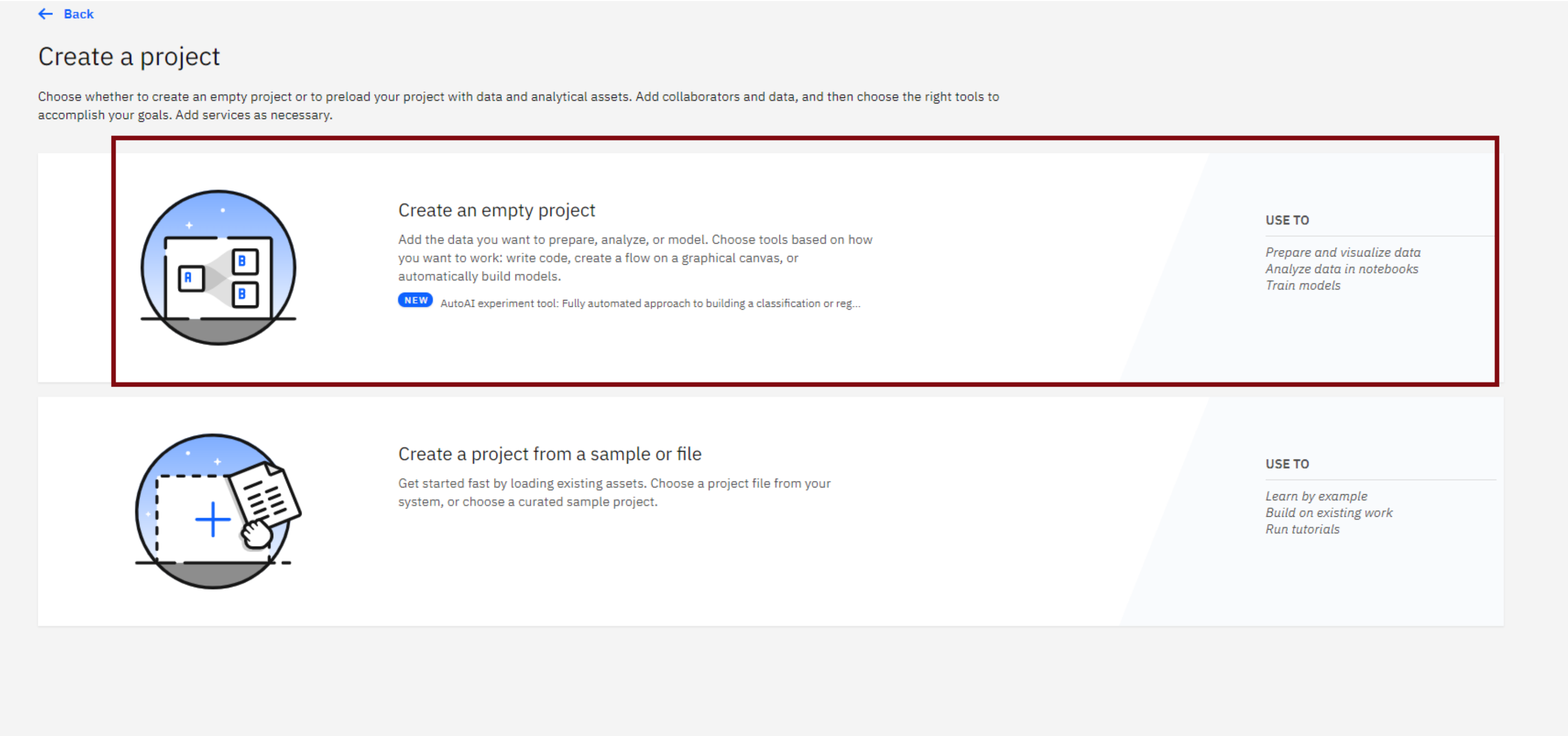
Create a project

Extend your capabilities

Add tools, databases, or other features by creating services instances.

Create a service

On the Create a project page, click **Create an empty project**



Create a project

Choose whether to create an empty project or to preload your project with data and analytical assets. Add collaborators and data, and then choose the right tools to accomplish your goals. Add services as necessary.

Create an empty project

Add the data you want to prepare, analyze, or model. Choose tools based on how you want to work: write code, create a flow on a graphical canvas, or automatically build models.

NEW AutoAI experiment tool: Fully automated approach to building a classification or reg...

USE TO

Prepare and visualize data
Analyze data in notebooks
Train models

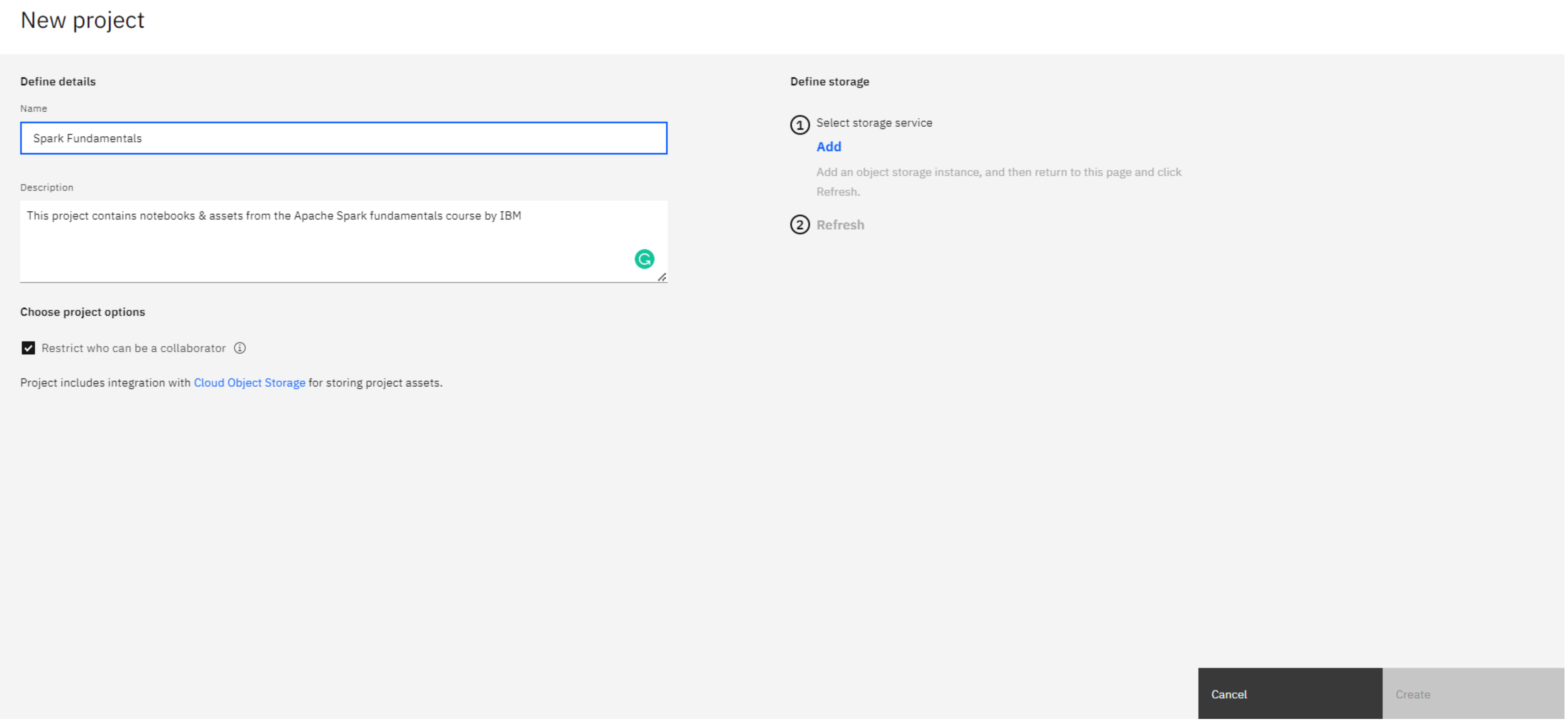
Create a project from a sample or file

Get started fast by loading existing assets. Choose a project file from your system, or choose a curated sample project.

USE TO

Learn by example
Build on existing work
Run tutorials

Provide a **Project Name** and **Description**, as shown below:



New project

Define details

Name: Spark Fundamentals

Description: This project contains notebooks & assets from the Apache Spark fundamentals course by IBM

Choose project options

☑ Restrict who can be a collaborator ⓘ

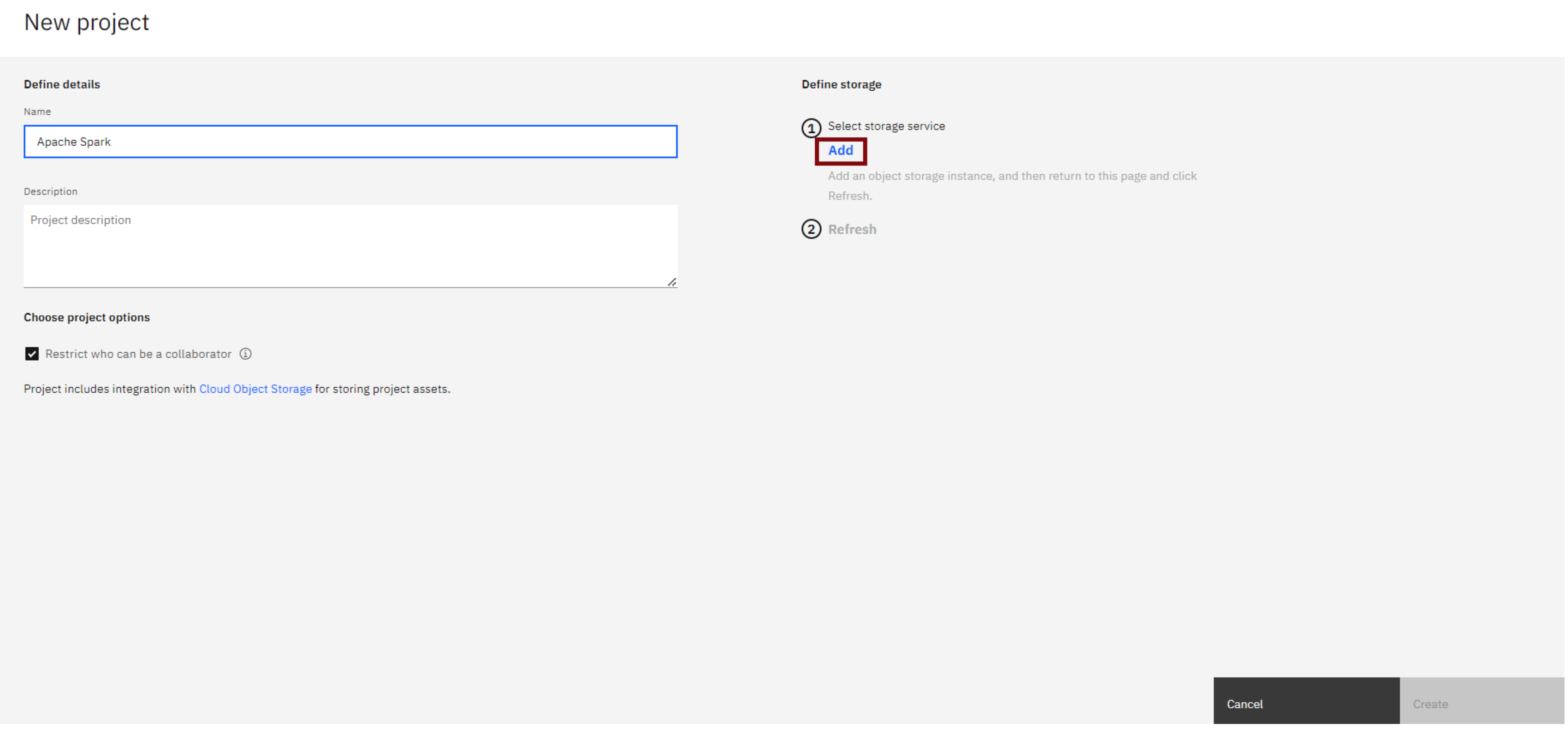
Project includes integration with Cloud Object Storage for storing project assets.

Define storage

1. Select storage service
   Add
   Add an object storage instance, and then return to this page and click Refresh.
2. Refresh

You must also create storage for the project.

Click **Add**



New project

Define details

Name: Apache Spark

Description: Project description

Choose project options

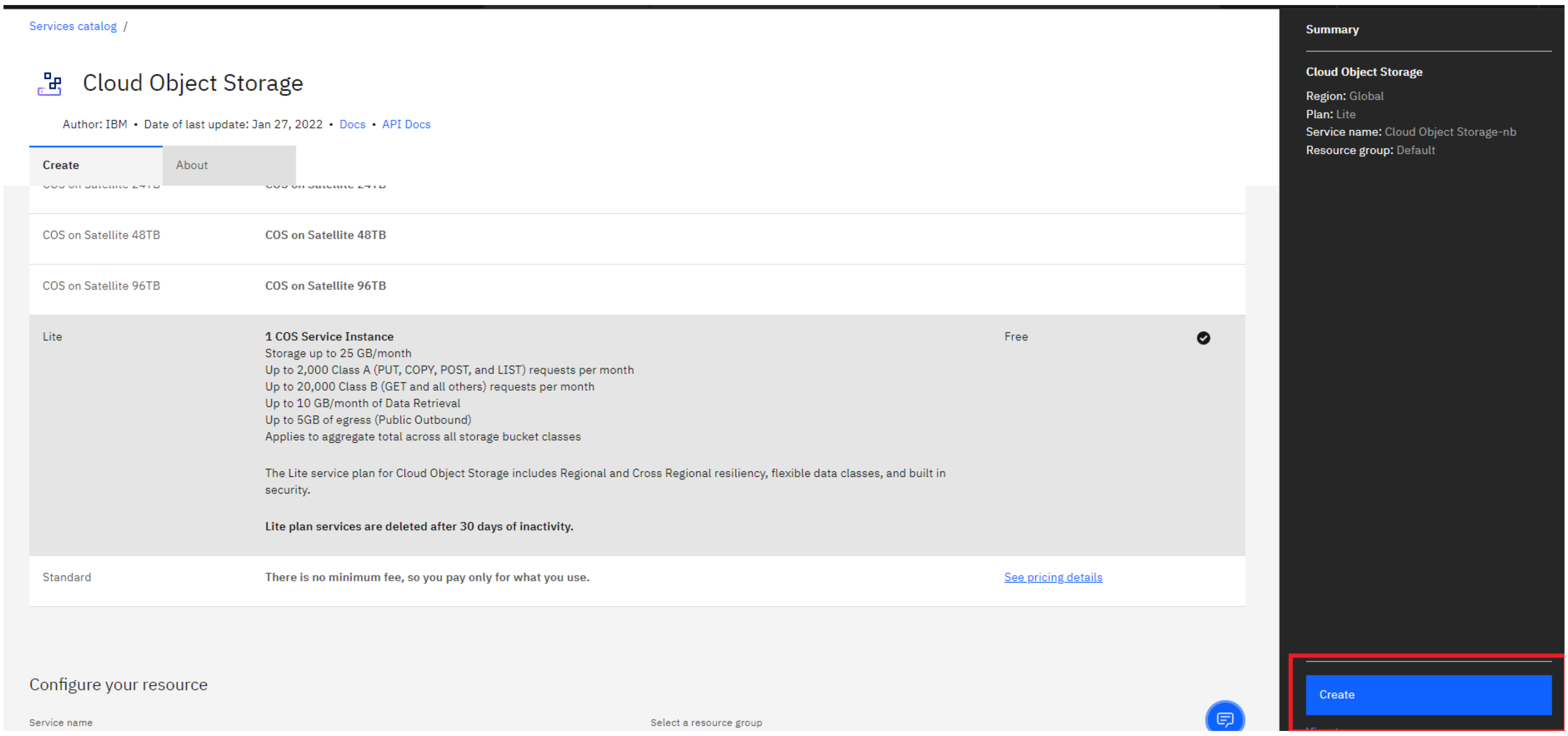
☑ Restrict who can be a collaborator ⓘ

Project includes integration with Cloud Object Storage for storing project assets.

Define storage

1. Select storage service
   Add
   Add an object storage instance, and then return to this page and click Refresh.
2. Refresh

On the Cloud Object Storage page, Select the 'Lite' plan and then click on **Create.** at the bottom.



Services catalog /

Cloud Object Storage

Author: IBM • Date of last update: Jan 27, 2022 • Docs • API Docs

COS on Satellite 48TB	COS on Satellite 48TB		
COS on Satellite 96TB	COS on Satellite 96TB		
Lite	**1 COS Service Instance** Storage up to 25 GB/month Up to 2,000 Class A (PUT, COPY, POST, and LIST) requests per month Up to 20,000 Class B (GET and all others) requests per month Up to 10 GB/month of Data Retrieval Up to 5GB of egress (Public Outbound) Applies to aggregate total across all storage bucket classes The Lite service plan for Cloud Object Storage includes Regional and Cross Regional resiliency, flexible data classes, and built in security. **Lite plan services are deleted after 30 days of inactivity.**	Free	✓
Standard	There is no minimum fee, so you pay only for what you use.	See pricing details	

Configure your resource

Service name

Select a resource group

Summary

Cloud Object Storage

Region: Global
Plan: Lite
Service name: Cloud Object Storage-nb
Resource group: Default

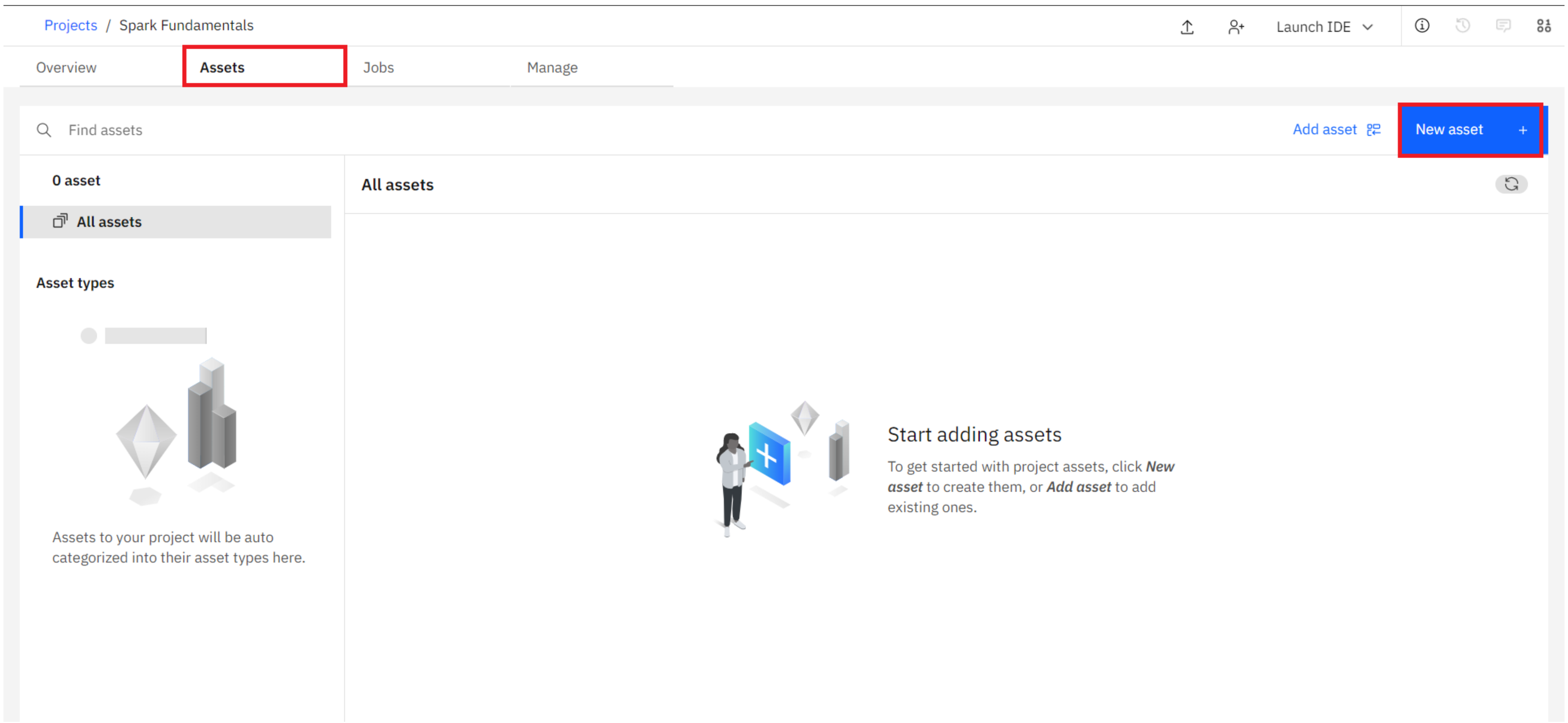
You will be redirected to the Object storage page. If you do not see your instance active, please click on **Refresh** as below:

On the New project page, note that the storage has been added, and then click **Create.**

After creating the project you will need to add a Jupyter notebook to your project.

Step 2: Adding a Notebook to the Project:

You need to add a Notebook to your project. Go to the **Assets** tab & Click on **New asset.**



Projects / Spark Fundamentals

0 asset

All assets

Asset types

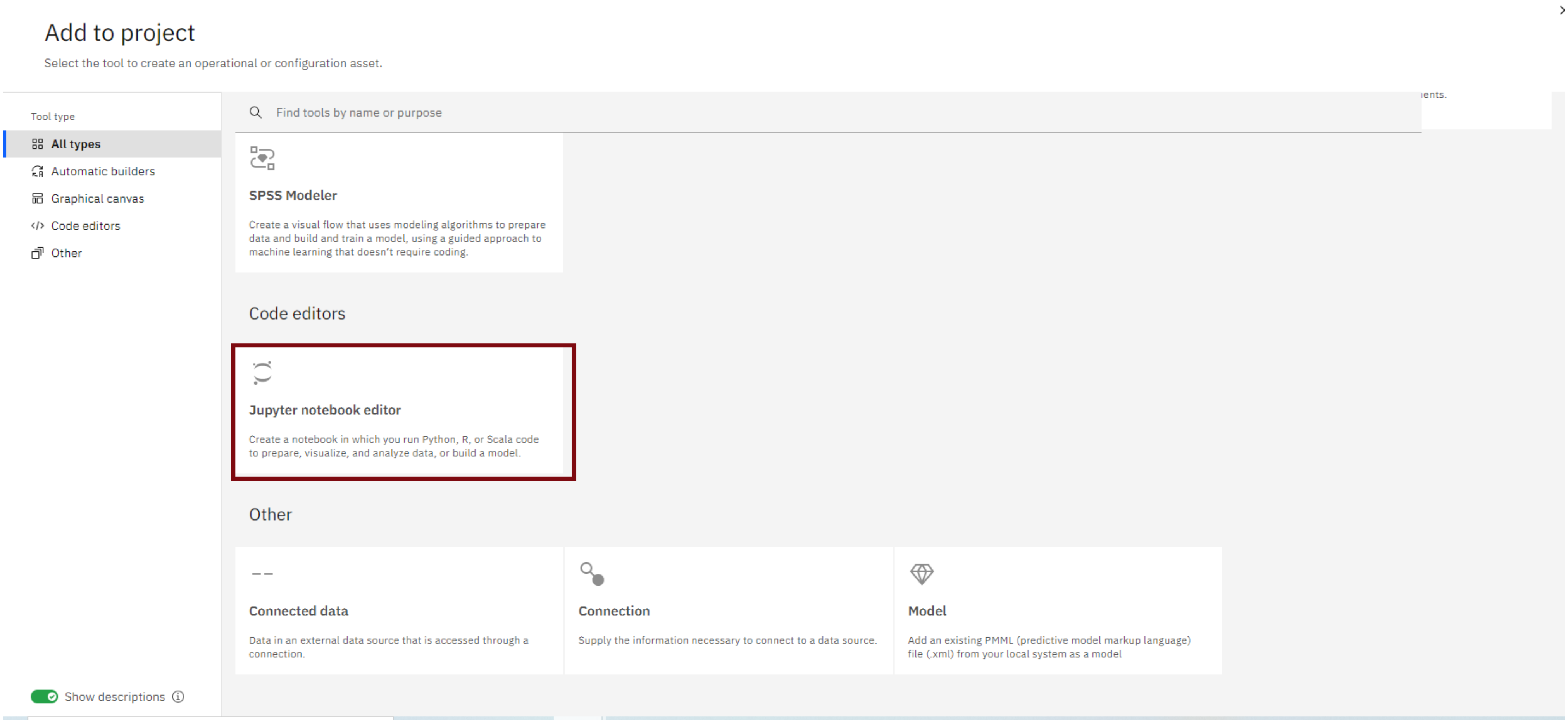
Assets to your project will be auto categorized into their asset types here.

All assets

Start adding assets

To get started with project assets, click ***New asset*** to create them, or ***Add asset*** to add existing ones.

Under **All types** select **Jupyter Notebook Editor**



Add to project

Select the tool to create an operational or configuration asset.

SPSS Modeler

Create a visual flow that uses modeling algorithms to prepare data and build and train a model, using a guided approach to machine learning that doesn't require coding.

Code editors

Jupyter notebook editor

Create a notebook in which you run Python, R, or Scala code to prepare, visualize, and analyze data, or build a model.

Other

Connected data

Data in an external data source that is accessed through a connection.

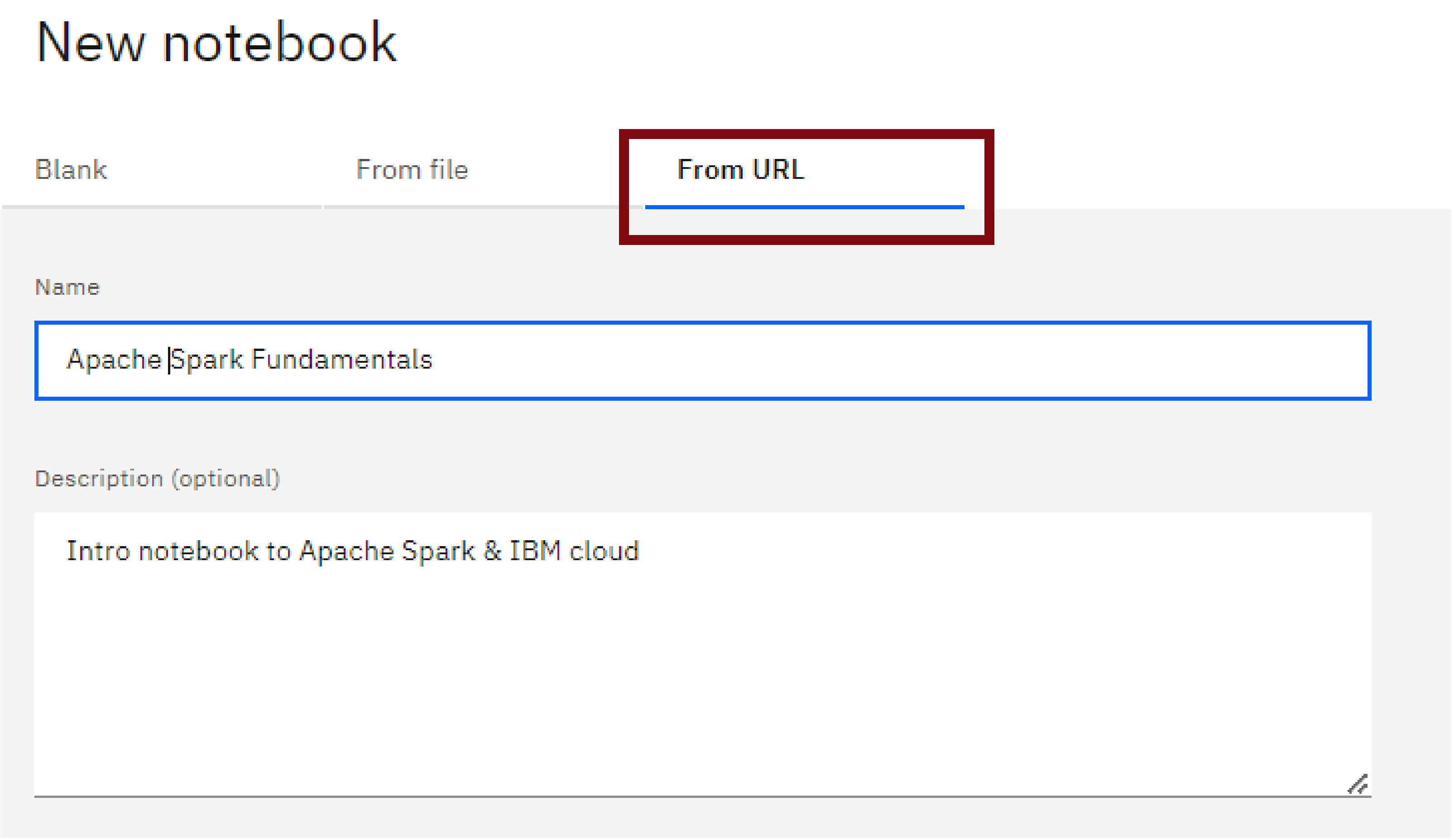
Connection

Supply the information necessary to connect to a data source.

Model

Add an existing PMML (predictive model markup language) file (.xml) from your local system as a model

On the New Notebook page, enter a name and description for the notebook, and then click From URL as shown below.



New notebook

Name

Apache Spark Fundamentals

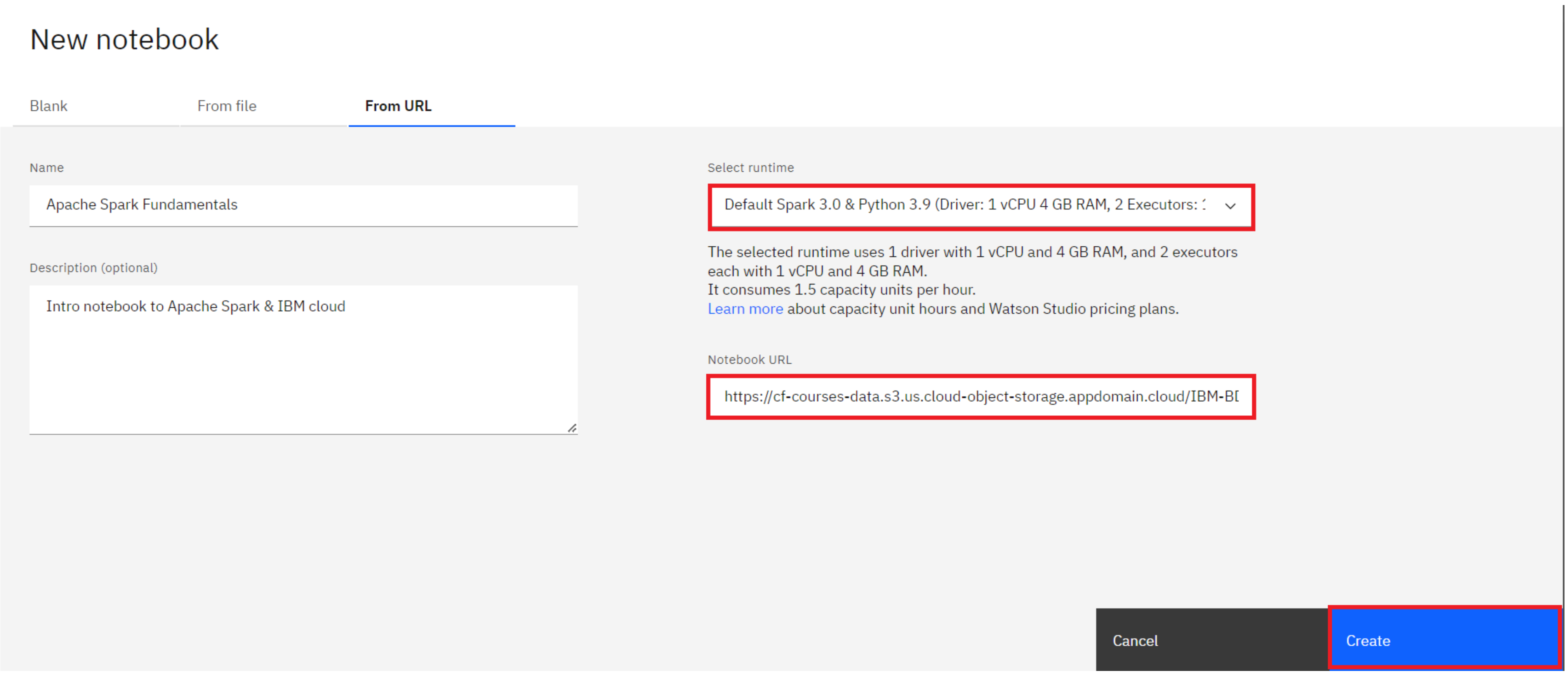
Description (optional)

Intro notebook to Apache Spark & IBM cloud

Important: **Select `"Default Spark 3.0 & Python 3.9"` as the runtime.**

This will initiate a kernel with Spark installed. Copy and paste the notebook URL - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-BD0225EN-SkillsNetwork/labs/SparkIntro.ipynb for the **Apache Spark Python Intro** from this course into the **Notebook URL** box, and then click **Create Notebook**.

Note: For future Watson Studio labs that involve Jupyter notebooks, please replace the above notebook link with the relevant link or upload the notebook manually if needed.



New notebook

Name: Apache Spark Fundamentals

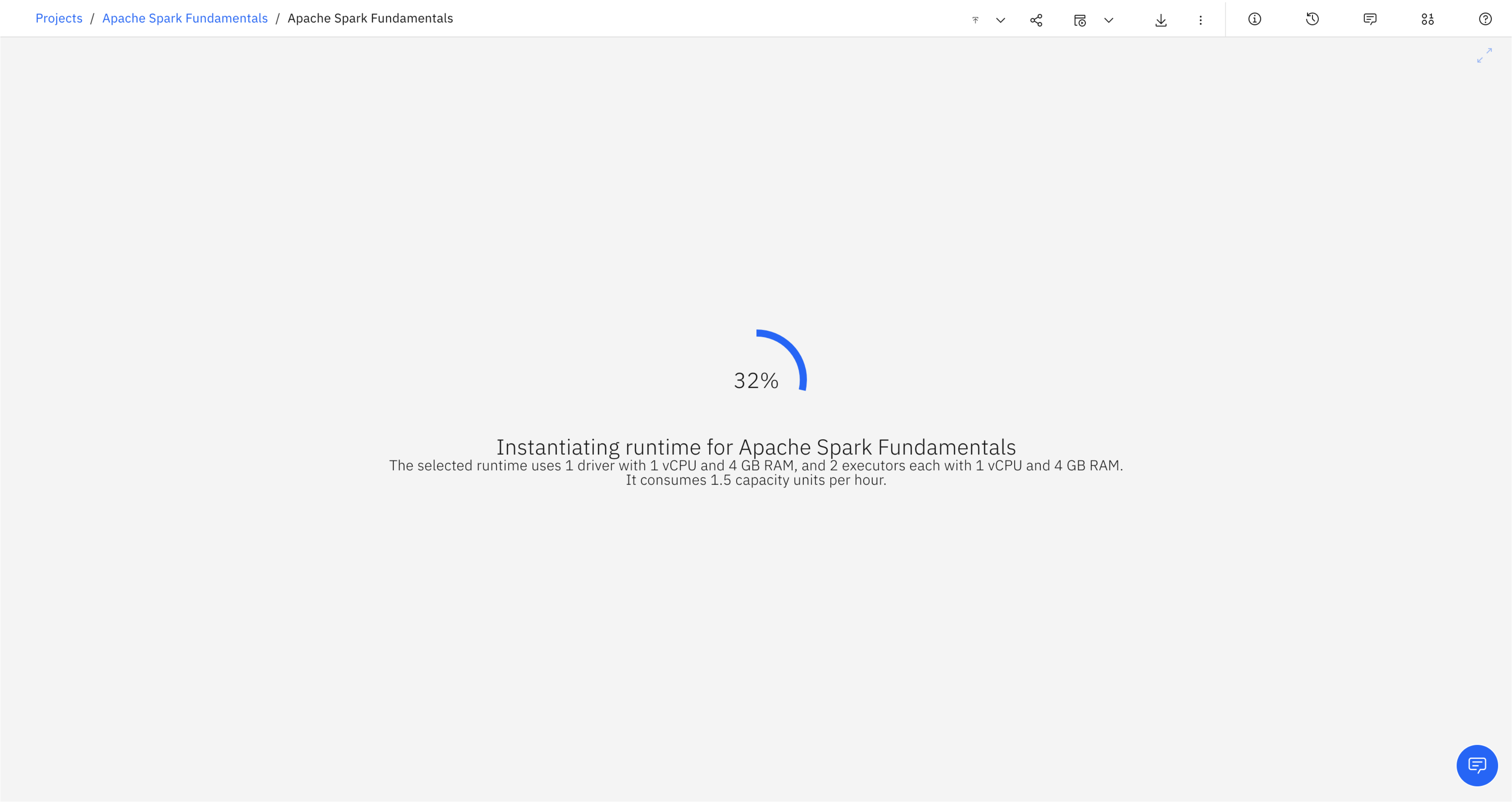
Description (optional): Intro notebook to Apache Spark & IBM cloud

Select runtime: Default Spark 3.0 & Python 3.9 (Driver: 1 vCPU 4 GB RAM, 2 Executors: 1

The selected runtime uses 1 driver with 1 vCPU and 4 GB RAM, and 2 executors each with 1 vCPU and 4 GB RAM.
It consumes 1.5 capacity units per hour.
Learn more about capacity unit hours and Watson Studio pricing plans.

Notebook URL: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-BI

You should see a loading screen like this:

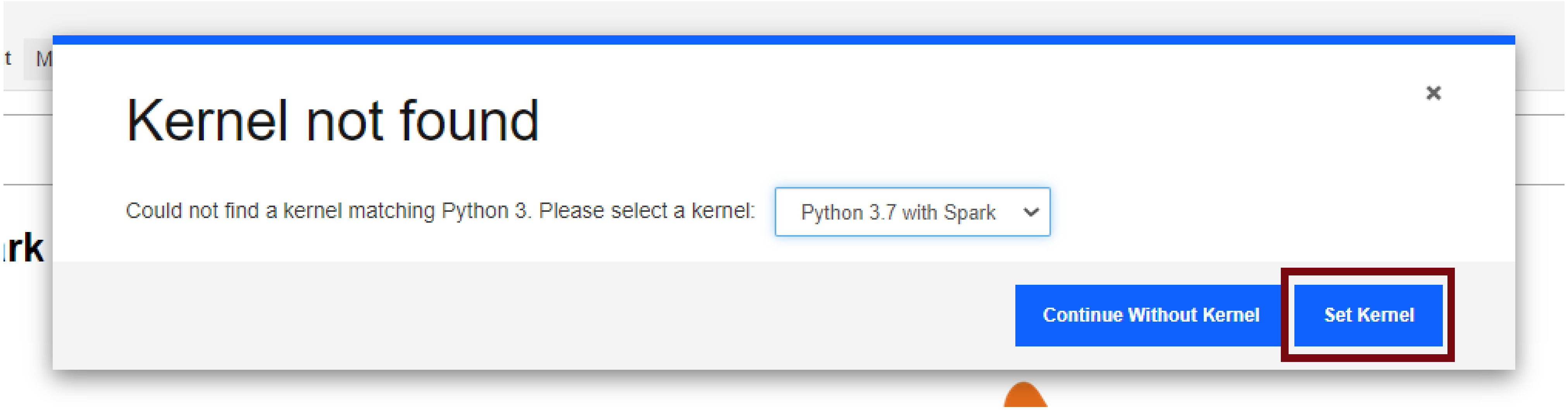


Projects / Apache Spark Fundamentals / Apache Spark Fundamentals

32%

Instantiating runtime for Apache Spark Fundamentals
The selected runtime uses 1 driver with 1 vCPU and 4 GB RAM, and 2 executors each with 1 vCPU and 4 GB RAM.
It consumes 1.5 capacity units per hour.

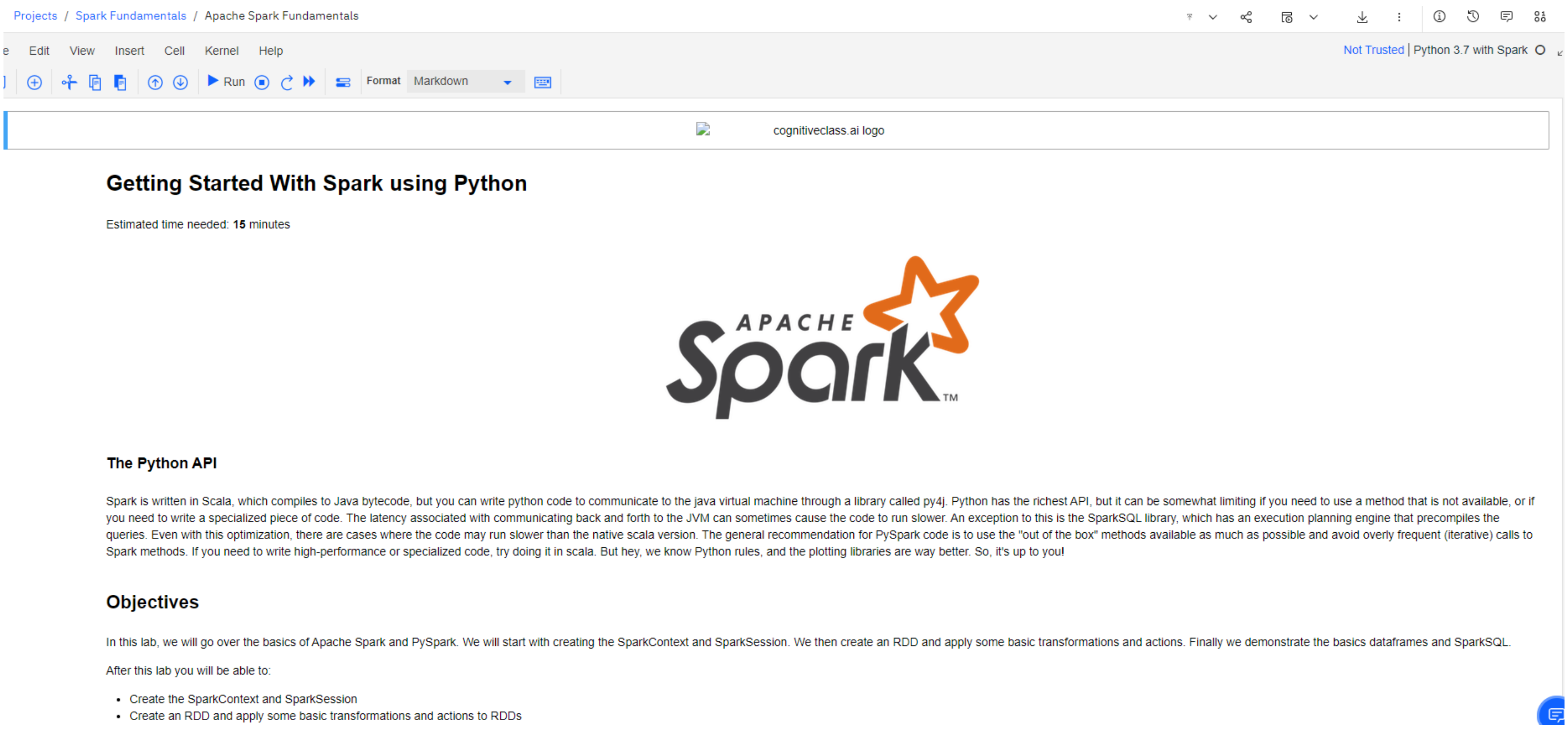
Click on **Set Kernel**



Kernel not found

Could not find a kernel matching Python 3. Please select a kernel: Python 3.7 with Spark

Once the kernel has been initiated you will see the notebook like this. Please run all the cells to complete the lab.



Projects / Spark Fundamentals / Apache Spark Fundamentals
e Edit View Insert Cell Kernel Help
Not Trusted | Python 3.7 with Spark
Run
Format Markdown
cognitiveclass.ai logo

Getting Started With Spark using Python

Estimated time needed: **15** minutes

The Python API

Spark is written in Scala, which compiles to Java bytecode, but you can write python code to communicate to the java virtual machine through a library called py4j. Python has the richest API, but it can be somewhat limiting if you need to use a method that is not available, or if you need to write a specialized piece of code. The latency associated with communicating back and forth to the JVM can sometimes cause the code to run slower. An exception to this is the SparkSQL library, which has an execution planning engine that precompiles the queries. Even with this optimization, there are cases where the code may run slower than the native scala version. The general recommendation for PySpark code is to use the "out of the box" methods available as much as possible and avoid overly frequent (iterative) calls to Spark methods. If you need to write high-performance or specialized code, try doing it in scala. But hey, we know Python rules, and the plotting libraries are way better. So, it's up to you!

Objectives

In this lab, we will go over the basics of Apache Spark and PySpark. We will start with creating the SparkContext and SparkSession. We then create an RDD and apply some basic transformations and actions. Finally we demonstrate the basics dataframes and SparkSQL.

After this lab you will be able to:

- Create the SparkContext and SparkSession
- Create an RDD and apply some basic transformations and actions to RDDs

Changelog

Date	Version	Changed by	Change Description
2021-07-15	1.0	Karthik	Initial draft
2021-08-17	1.1	Karthik	Post Beta feedback
2022-02-22	1.2	K Sundararajan	Instructions Updated
2022-04-06	1.3	Sourabh	Images Updated