

## Spark Lab 3

# Submit a Spark job to your cluster

### Basic instructions

This is a mandatory lab.

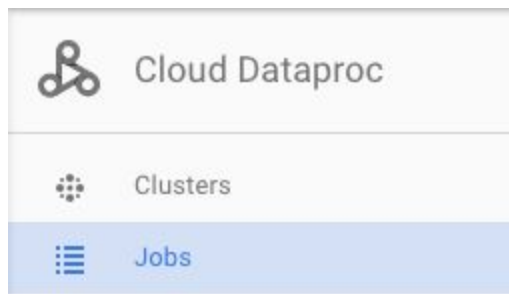
This lab is to be done by each group, that means from Group A to Group H, so we will have 7-8 Members on each group. Due date would be 5th of March 10 am. Any coursework submitted after this time will be discarded.

How to submit?

Generate a complete new email to me with a subject saying: GroupX-Spark-Lab3.pdf and the file attached with the same name format. The email must be sent by one member of the team and must content all the members of the team on cc. The names must be also displayed in the document.

As we did in previous labs create a cluster. You don't need to add any notebook for this scenario.

Select Jobs in the left nav to switch to Dataproc's jobs view.



Click Submit job.

Cloud Dataproc  
Jobs

Cloud Dataproc jobs lets you submit and manage any Hadoop, Hive, Spark, or Pig job that runs in a Cloud Dataproc cluster.

To get started, create and submit your first job.

Submit Job

Select the region where you created your cluster. Region drop-down menu.

Select your new cluster X from the Cluster drop-down menu.


Select Spark from the Job type drop-down menu.

Enter file:///usr/lib/spark/examples/jars/spark-examples.jar in the Jar files field.

Enter org.apache.spark.examples.SparkPi in the Main class or jar field.

Enter 1000 in the Arguments field to set the number of tasks.

It should look like something like this. Please have a look at all the different types of Jobs you can submit.

**Region** 


us-central1

**Cluster**

gcelab


**Job type**

Spark


**Jar files** (Optional) 

file:///usr/lib/spark/examples/jars/spark-examples.jar

Enter file path, for example, hdfs://example/example.jar


**Main class or jar** 

org.apache.spark.examples.SparkPi


**Arguments** (Optional) 

1000

Press <Return> to add more arguments

**Properties** (Optional) 

+ Add item

**Labels** (Optional) 

+ Add item

**Submit** **Cancel**

Equivalent [REST](#)

Click Submit.

Your job should appear in the Jobs list, which shows all your project's jobs with their cluster, type, and current status. The new job displays as "Running", and then "Succeeded" once it completes. To see your completed job's output:

Click the job ID in the Jobs list.

Jobs [+ SUBMIT JOB](#) [REFRESH](#) [STOP](#) [DELETE](#) [REGIONS ▼](#)

---

Search jobs, press Enter ?

<input type="checkbox"/> Job ID	Region	Type	Cluster	Start time	Elapsed time	Status
<input checked="" type="checkbox"/> <b>c99b6097-204a-4130-9e63-c3b7ca0ea1b3</b>	us-central1	Spark	gcelab	Aug 29, 2017, 1:07:38 PM	43 sec	Succeeded

Select Line Wrapping to avoid scrolling.

✓ **8f48de73-9e56-45f0-98a3-971fe3102e8f**  
Start time: Jun 14, 2016, 10:06:48 AM Elapsed time: 48 sec Status: Succeeded

**Output** Configuration

☒ Line wrapping

```
16/06/14 17:06:53 INFO akka.event.slf4j.Slf4jLogger: Slf4jLogger start
16/06/14 17:06:53 INFO Remoting: Starting remoting
16/06/14 17:06:53 INFO Remoting: Remoting started; listening on address
16/06/14 17:06:53 INFO org.spark-project.jetty.server.Server: jetty-8
16/06/14 17:06:53 INFO org.spark-project.jetty.server.AbstractConnector:
16/06/14 17:06:53 INFO org.spark-project.jetty.server.Server: jetty-8
16/06/14 17:06:53 INFO org.spark-project.jetty.server.AbstractConnector:
16/06/14 17:06:54 INFO org.apache.hadoop.yarn.client.RMPProxy: Connecti
16/06/14 17:06:57 INFO org.apache.hadoop.yarn.client.api.impl.YarnCli
```

You should see that your job has successfully calculated a rough value for pi!

<Paste here screenshot of the output>

Now use the python file you create in Lab1 and submit the job from the cluster command line(so you have to ssh into the cluster) in your spark cluster by using spark-submit:  
<https://spark.apache.org/docs/latest/submitting-applications.html>

<Paste here screenshot of the output>

Happy sparking!