EEL 6761: Cloud Computing

Homework 2

Spring 2022

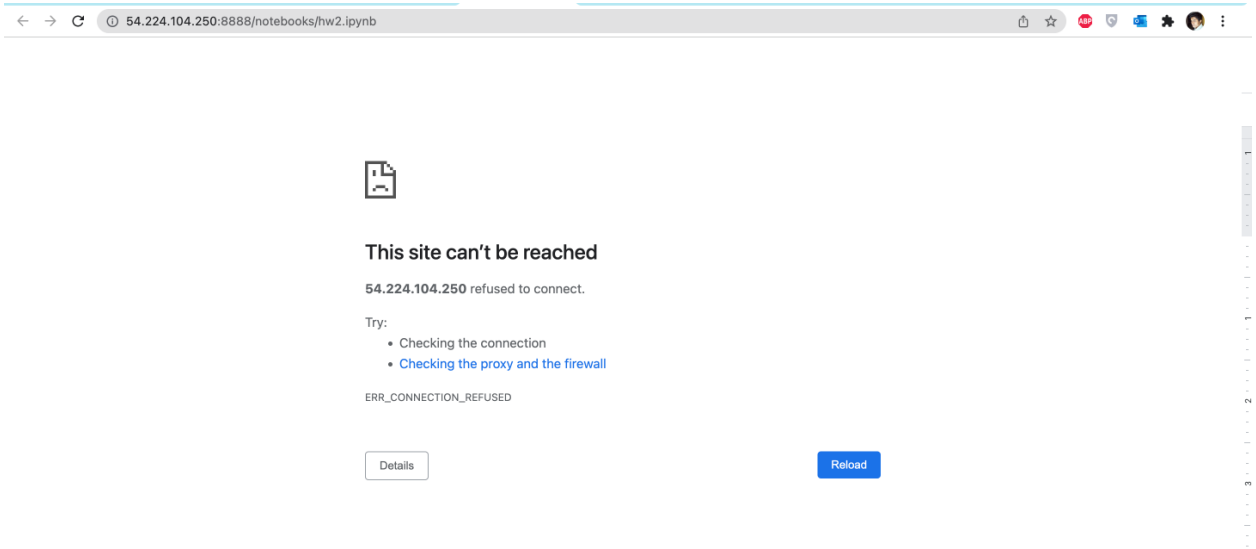
Deploying an Elastic Cloud Service

Anmol Lingan Gouda Patil
UFID: 1967 3150

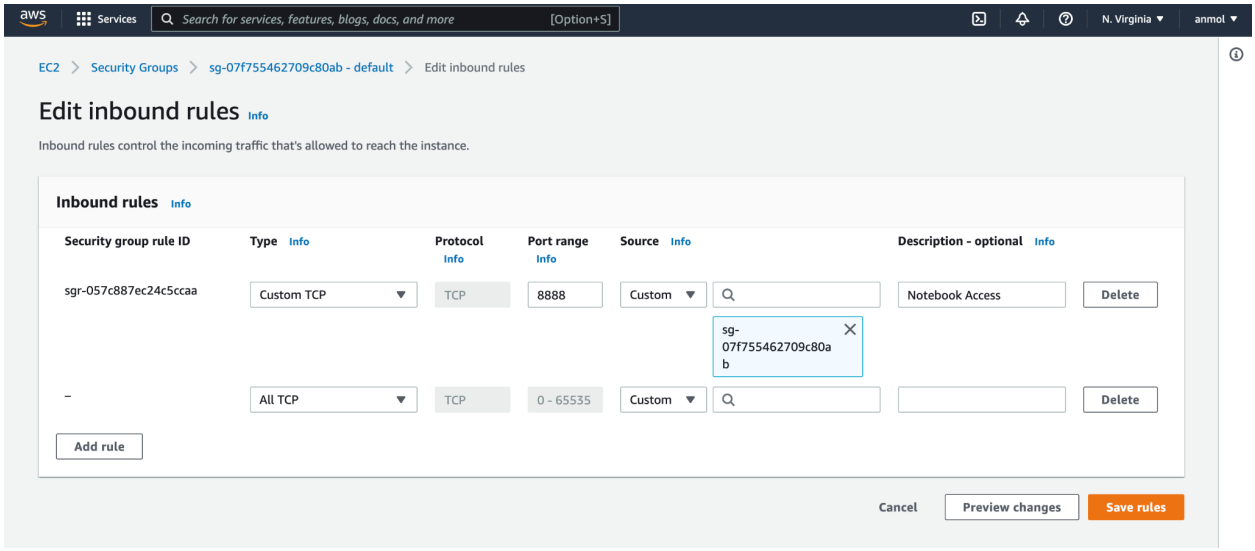
A. When configuring PySpark to use Jupyter as its driver, you were asked to set its execution options using the PYSPARK_DRIVER_PYTHON_OPTS environment variable. Why is the --ip=* option needed? What happens if it is omitted?

Ans: As per the Jupyter Notebook user documentation, by default a notebook server runs locally on the localhost i.e ip = 127.0.0.1 and port = 8888. And the access to the notebook server via a browser is limited only to http://127.0.0.1:8888.
But when we are trying to use the jupyter notebook from an external source outside of the local machine(launched EC2), we need the jupyter server to allow accepting requests from different/external interfaces(ips). Thus the –ip='*' option allows the jupyter server to bind on all interfaces for the public server(launched EC2) thereby allowing us to access the notebook outside of the local instance.

When it is omitted we will not be able to access the jupyter notebook server from outside the EC2 instance, and a connection refused error like the one below is thrown.



B. What is the difference between "Custom TCP" and "All TCP" types in the EC2 Security Groups? How can you protect your jupyter notebook apart from the credentialed access? Provide the steps.
Ans: As seen in the image below, when the **Custom TCP** option is selected we must specify a port or a range of ports to limit the incoming tcp traffic to only such port or range ports. However when the **All TCP** option is chosen, AWS auto populates the port range to all the ports from 0 to 65535, stating that when you choose All TCP, incoming tcp traffic shall bind on any ports from 0 to 65535.



Besides protecting the jupyter notebook using a password here are some more steps to achieve additional security:

1. **Preparing a hashed password**: we can prepare a hashed password manually by using the function: notebook.auth.security.passwd(). This function when called will ask you for your password and verify it by retyping and then outputs a sha1 hash.
2. **Using hashed password in notebook configuration file/ PYSPARK_DRIVER_PYTHON_OPTS:**
   we can then add the hashed password from previous step to jupyter_notebook_config.py file or set the **–NotebookApp.password** = the hashed string in unicode, if we are using the **PYSPARK_DRIVER_PYTHON_OPTS**.
3. **Encrypted communication using secure socket layer(SSL)**: a hashed password needs to be encrypted before it is sent via the browser, thus it is prudent to use SSL with a self signed web certificate. Mention the certificate file path using the **–certfile=** option along with the key using **–keyfile yourkey.key.** A self signed certificate can be generated using the following command using the openssl:

   ```
   openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout your_key.key
   -out your_cert.pem
   ```

C. When configuring the pyspark service used by systemd, you were asked to include the following lines in your pyspark.service file:

After=network.target
After=systemd-user-sessions.service
After=network-online.target

Why are these lines needed? What happens if they are omitted?

Ans:
network.target:
This unit is supposed to indicate when network functionality is available.

systemd-user-sessions.service:
is a service that controls user logins through pam_nologin(Prevent non-root users from login).

network-online.target:
This target unit is intended to pull in a service that delays further execution until the network is sufficiently set up.

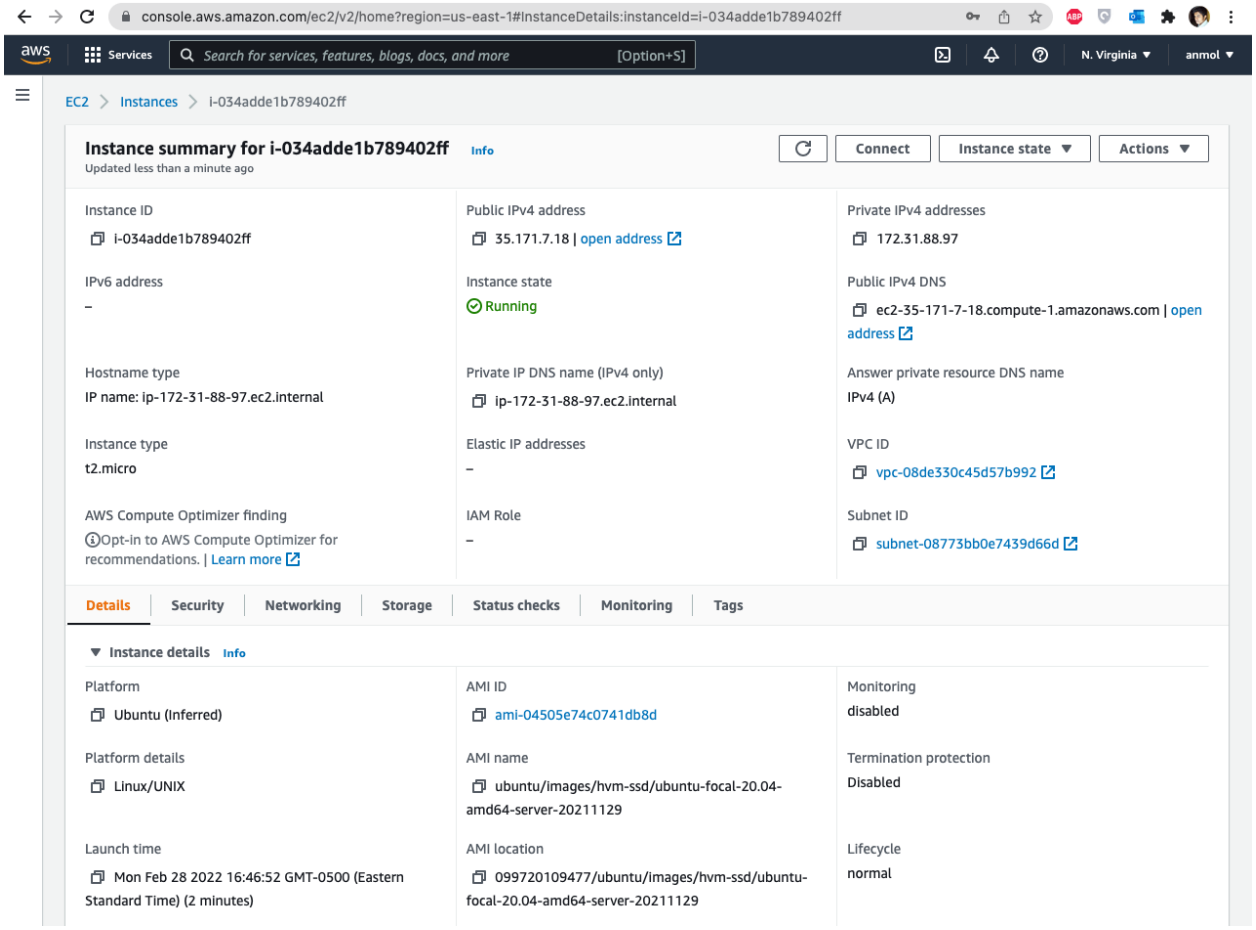
So when we append these options to **After=** , we are essentially stating that we do not start our service until the system on which it is going to run has network functionality up and running and the root user has logged into the system. And if the internet connectivity is not yet established, delay it until it is sufficiently set up.

If these lines are omitted, we will not be able to connect to the jupyter notebook server from outside the local instance in which it is running.

D. Does increasing the number of instances in your cloud decrease the execution time of Spark jobs? Why or why not?
Ans:
In the current scope of the assignment, since the calculation of the value of **Pi** is an infinite process, hence we cannot measure the execution time.
However, since we are running the spark jobs in a **standalone mode** and not **cluster mode** there is no distribution of work across instances. All instances are running their own version of the process. If we would have run the spark job in a cluster mode, then actual distribution of work could be done using parallelization that spark takes care of. Thus increasing the number of instances will not decrease the execution time of the spark jobs

E. What happens if the minimum number of instances in the autoscaling group is set to zero?
Ans:
If the minimum number of instances in the autoscaling group is set to 0 along with the **desired capacity** option = 0. The autoscaling group will not launch any instances. But even if your minimum capacity is zero, the autoscaling group launches the maximum of (min capacity, desired capacity) # of instances. Suppose we set minimum capacity to 0 and desired capacity to 0 there would be no process started/ instance launched.
However if the desired capacity is greater than 0 , say 1 then the autoscaling group will launch an instance and then later follows the target policy set.

SCREENSHOTS



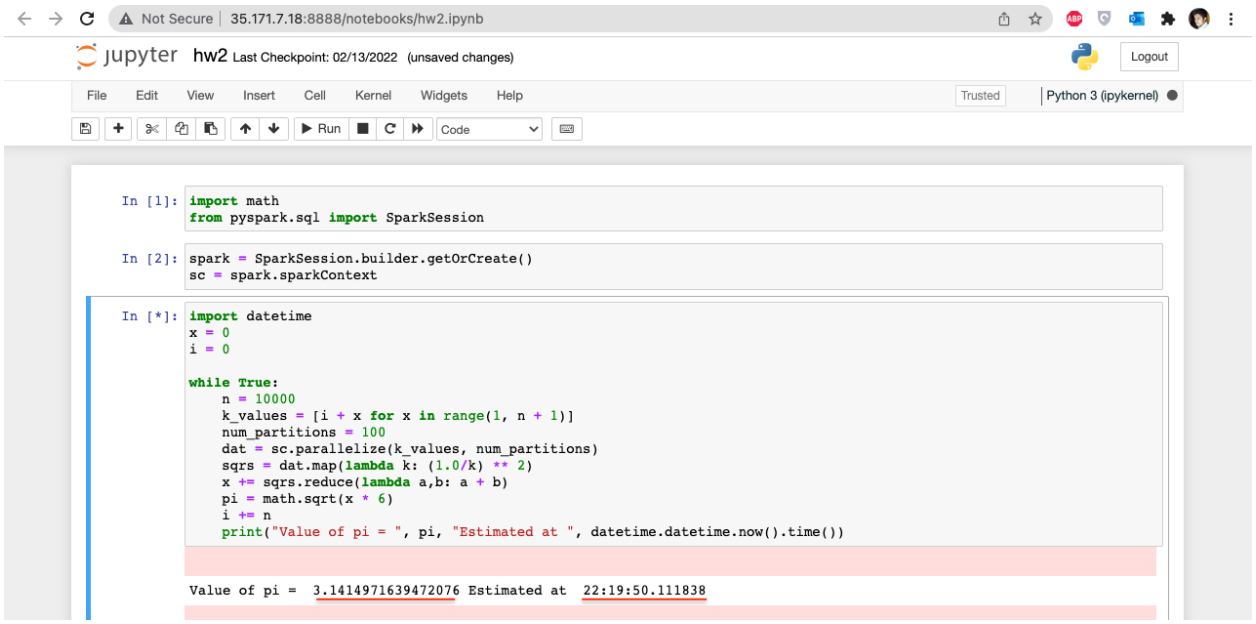
A

```
anmol@Anmols-MacBook-Pro Patil_hw2 $ ssh -i "eel6761.cer" ubuntu@ec2-35-171-7-18.compute-1.amazonaws.com
The authenticity of host 'ec2-35-171-7-18.compute-1.amazonaws.com (35.171.7.18)' can't be established.
ED25519 key fingerprint is SHA256:nwIxPR1WtZVQj0e3IMIKN+xCyhL/kwopkZ6zSussxWU.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:11: ec2-3-83-223-119.compute-1.amazonaws.com
    ~/.ssh/known_hosts:14: ec2-44-202-46-169.compute-1.amazonaws.com
    ~/.ssh/known_hosts:15: ec2-52-90-52-134.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-35-171-7-18.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1028-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Feb 28 21:53:07 UTC 2022

  System load:  0.01              Processes:             105
  Usage of /:   49.8% of 7.69GB   Users logged in:       0
  Memory usage: 26%               IPv4 address for eth0: 172.31.88.97
  Swap usage:   0%


12 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sat Feb 19 01:35:18 2022 from 184.188.101.162
ubuntu@ip-172-31-88-97:~$
```

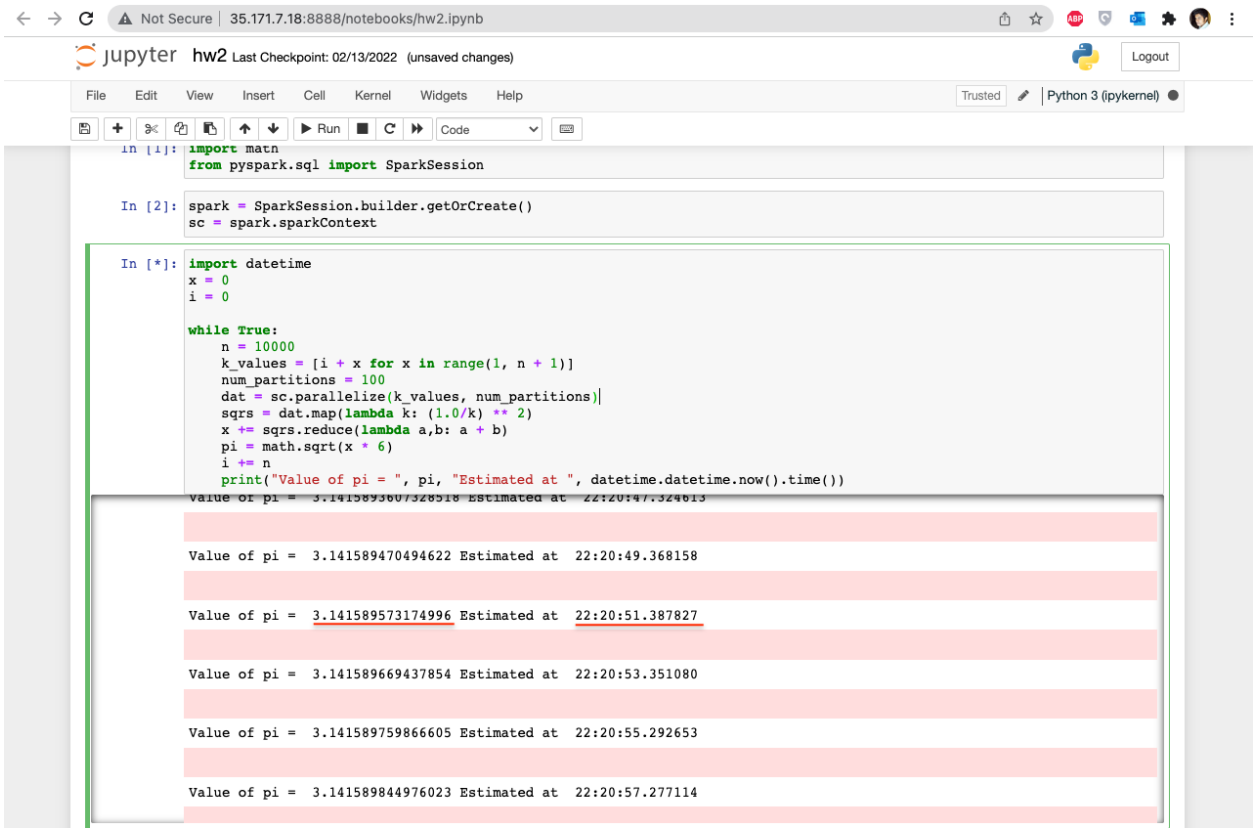
```
ubuntu@ip-172-31-88-97:~$ pyspark
[W 22:10:32.331 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[I 22:10:32.334 NotebookApp] Serving notebooks from local directory: /home/ubuntu
[I 22:10:32.334 NotebookApp] Jupyter Notebook 6.4.8 is running at:
[I 22:10:32.334 NotebookApp] http://ip-172-31-88-97:8888/
[I 22:10:32.334 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[W 22:10:32.338 NotebookApp] No web browser found: could not locate runnable browser.
```

B

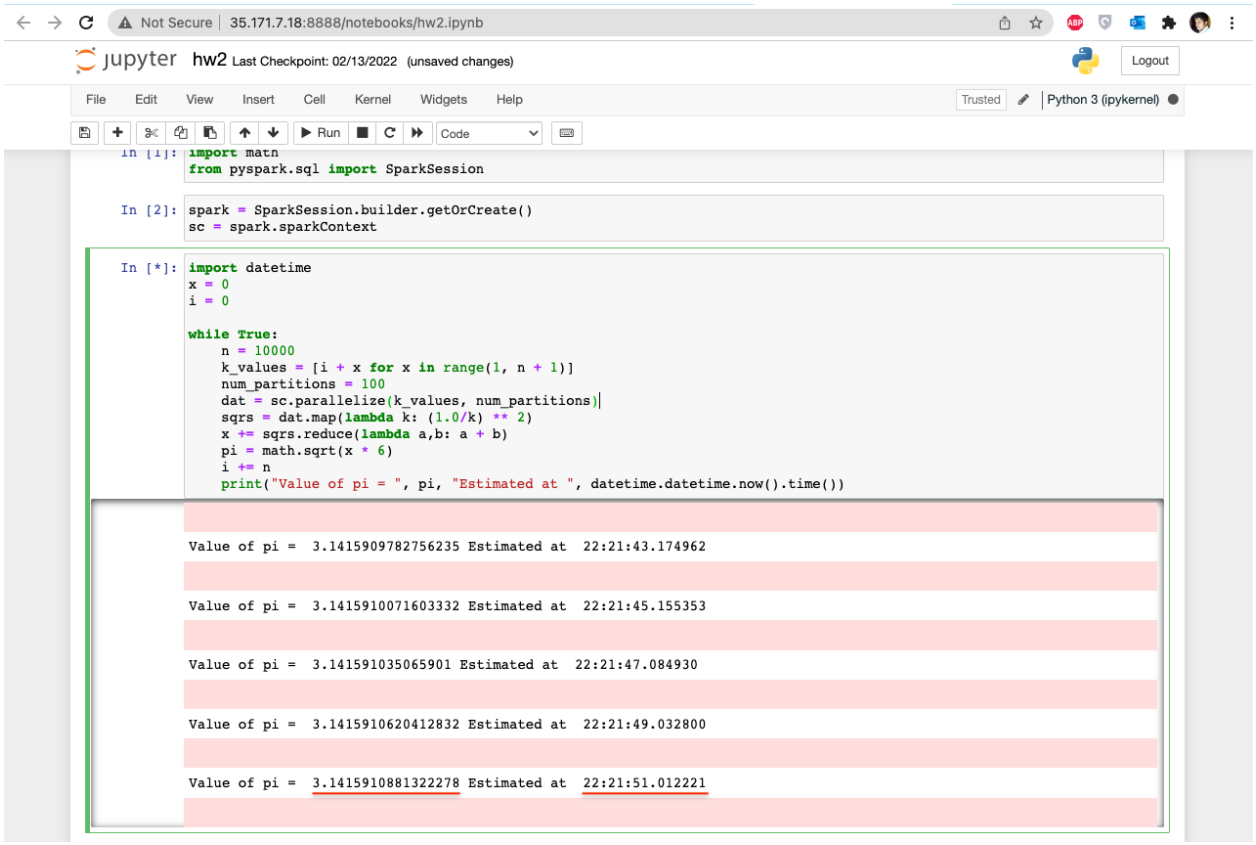
C


Not Secure | 35.171.7.18:8888/notebooks/hw2.ipynb
jupyter hw2 Last Checkpoint: 02/13/2022 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [1]:
import math
from pyspark.sql import SparkSession
In [2]:
spark = SparkSession.builder.getOrCreate()
sc = spark.sparkContext
In [*]:
import datetime
x = 0
i = 0
while True:
n = 10000
k_values = [i + x for x in range(1, n + 1)]
num_partitions = 100
dat = sc.parallelize(k_values, num_partitions)
sqrs = dat.map(lambda k: (1.0/k) ** 2)
x += sqrs.reduce(lambda a,b: a + b)
pi = math.sqrt(x * 6)
i += n
print("Value of pi = ", pi, "Estimated at ", datetime.datetime.now().time())
Value of pi = 3.1414971639472076 Estimated at 22:19:50.111838

D-1


Not Secure | 35.171.7.18:8888/notebooks/hw2.ipynb
jupyter hw2 Last Checkpoint: 02/13/2022 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [1]:
import math
from pyspark.sql import SparkSession
In [2]:
spark = SparkSession.builder.getOrCreate()
sc = spark.sparkContext
In [*]:
import datetime
x = 0
i = 0
while True:
n = 10000
k_values = [i + x for x in range(1, n + 1)]
num_partitions = 100
dat = sc.parallelize(k_values, num_partitions)
sqrs = dat.map(lambda k: (1.0/k) ** 2)
x += sqrs.reduce(lambda a,b: a + b)
pi = math.sqrt(x * 6)
i += n
print("Value of pi = ", pi, "Estimated at ", datetime.datetime.now().time())
Value of pi = 3.1415894704946222 Estimated at 22:20:49.368158
Value of pi = 3.1415895731749960 Estimated at 22:20:51.387827
Value of pi = 3.1415896694378540 Estimated at 22:20:53.351080
Value of pi = 3.1415897598666050 Estimated at 22:20:55.292653
Value of pi = 3.1415898449760230 Estimated at 22:20:57.277114

D-2


Not Secure 35.171.7.18:8888/notebooks/hw2.ipynb
jupyter hw2 Last Checkpoint: 02/13/2022 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3 (ipykernel)
Run Code
In [1]:
import math
from pyspark.sql import SparkSession
In [2]:
spark = SparkSession.builder.getOrCreate()
sc = spark.sparkContext
In [*]:
import datetime
x = 0
i = 0
while True:
n = 10000
k_values = [i + x for x in range(1, n + 1)]
num_partitions = 100
dat = sc.parallelize(k_values, num_partitions)
sqrs = dat.map(lambda k: (1.0/k) ** 2)
x += sqrs.reduce(lambda a,b: a + b)
pi = math.sqrt(x * 6)
i += n
print("Value of pi = ", pi, "Estimated at ", datetime.datetime.now().time())
Value of pi = 3.1415909782756235 Estimated at 22:21:43.174962
Value of pi = 3.1415910071603332 Estimated at 22:21:45.155353
Value of pi = 3.141591035065901 Estimated at 22:21:47.084930
Value of pi = 3.1415910620412832 Estimated at 22:21:49.032800
Value of pi = 3.1415910881322278 Estimated at 22:21:51.012221

D-3