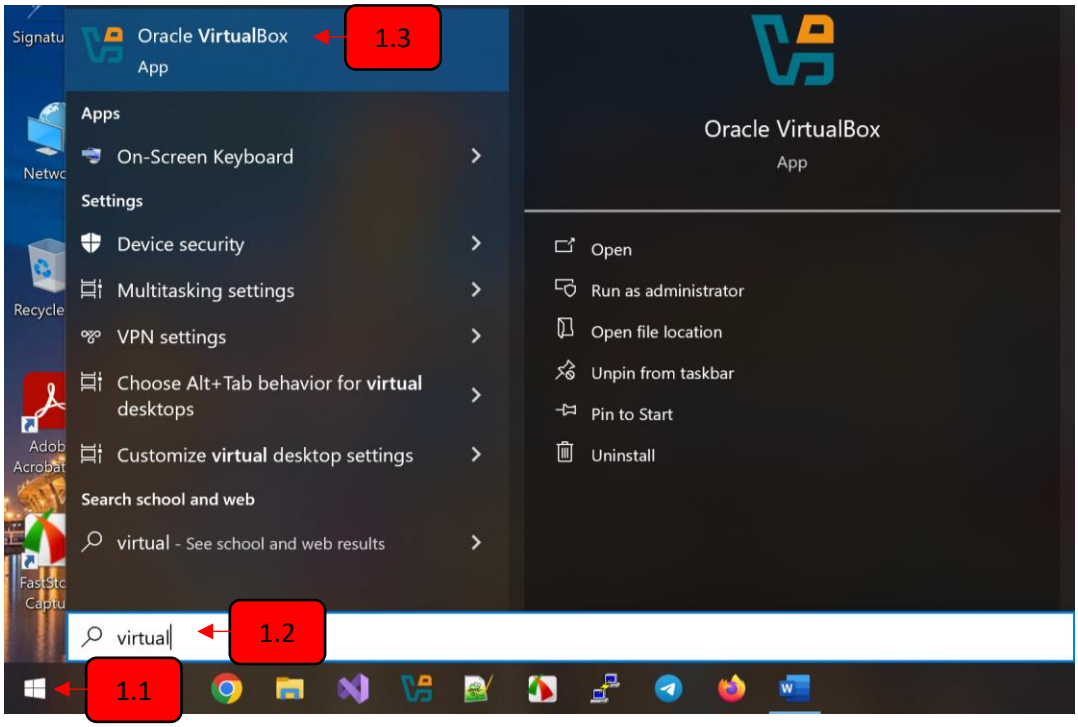
Instruction 2 - Big Data Analytics Via Jupyter in Ubuntu Linux

Part 1/7: Start VirtualBox and Virtual Machine

Step 1.1: Click the **Windows** icon.

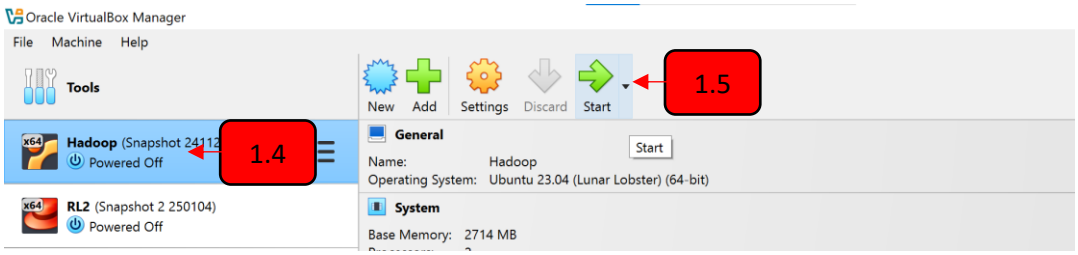
Step 1.2: Type **VirtualBox**.

Step 1.3: Click on the **Oracle VirtualBox** app.



Step 1.4: Select the **Hadoop Virtual Machine**.

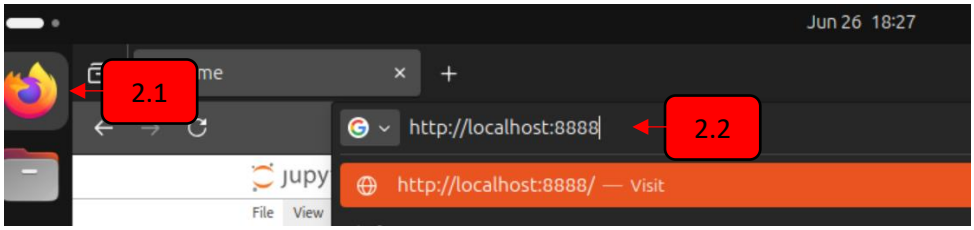
Step 1.5: Click the **Start** icon.



Part 2/7: Start Jupyter Application

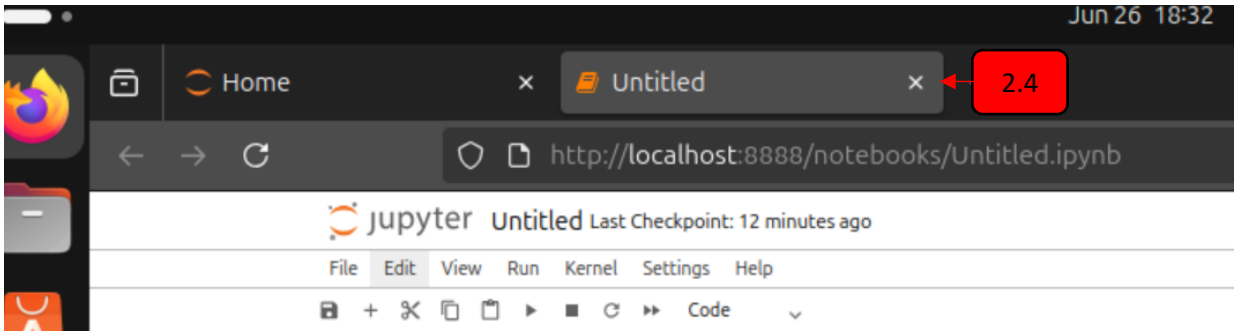
Step 2.1: Click the **Firefox** icon (or any web browser).

Step 2.2: Type **http://localhost:8888** and press **Enter**.



Step 2.3: Click the **New** menu**,** then select **Python3**.

Step 2.4: Jupyter will open a webpage named *Untitled* in a new tab.



Part 3/7: Start Hadoop and Sqoop Services

Step 3.1: Type the following code and press **Shift + Enter**:

```
import os
os.environ["PATH"] += ":/usr/local/hadoop/bin:/usr/local/hadoop/sbin:/home/mony/Downloads/sqoop/bin"
!start-all.sh
!jps
!hadoop version
!sqoop version
```

Step 3.2: Results will display below.

```
4080 DataNode
4464 ResourceManager
3874 NameNode
5573 Jps
4278 SecondaryNameNode
4587 NodeManager
Hadoop 2.7.3

Sqoop 1.4.7
git commit id 2328971411f57f0cb683dfb79d19d4d19d185dd8
Compiled by maugli on Thu Dec 21 15:59:58 STD 2017
```

3.2

Part 4/7: Import Remote Database to Hadoop Using Sqoop

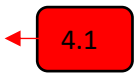
Step 4.1: Type the following and press **Shift + Enter**:

```
!sqoop import --connect jdbc:mysql://127.0.0.1/dbtest --username usertest
```

```
--password Admin1111 --table staff_data --target-dir /staff_data2
--delete-target-dir
!hadoop fs -ls /staff_data2
```

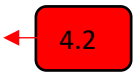
```
!sqoop import --connect jdbc:mysql://127.0.0.1/dbtest --username usertest --password Admin1111 --table staff_data -
!hadoop fs -ls /staff_data2

Warning: /home/mony/Downloads/sqoop/bin/../../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /home/mony/Downloads/sqoop/bin/../../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /home/mony/Downloads/sqoop/bin/../../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /home/mony/Downloads/sqoop/bin/../../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
25/06/28 17:42:43 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
```



Step 4.2: The result will appear below.

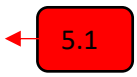
```
25/06/28 17:43:53 INFO mapreduce.ImportJobBase: Transferred 324 bytes in 62.794 seconds
25/06/28 17:43:53 INFO mapreduce.ImportJobBase: Retrieved 3 records.
Found 6 items
-rw-r--r--   1 root supergroup          0 2025-06-28 17:43 /staff_data2/_SUCCESS
-rw-r--r--   1 root supergroup        107 2025-06-28 17:43 /staff_data2/part-m-00000
-rw-r--r--   1 root supergroup          0 2025-06-28 17:43 /staff_data2/part-m-00001
-rw-r--r--   1 root supergroup          0 2025-06-28 17:43 /staff_data2/part-m-00002
-rw-r--r--   1 root supergroup        111 2025-06-28 17:43 /staff_data2/part-m-00003
-rw-r--r--   1 root supergroup        106 2025-06-28 17:43 /staff_data2/part-m-00004
```



Part 5/7: Export HDFS Data to Local Ubuntu

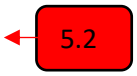
Step 5.1: Type the following and press **Shift + Enter**:

```
%%bash
mkdir /dataset2
hadoop fs -get /staff_data2/part-m-0000* /dataset2
cd /dataset2 && pwd && ls
```



Step 5.2: Output will be shown below.

```
/dataset2
part-m-00000
part-m-00001
part-m-00002
part-m-00003
part-m-00004
```



Part 6/7: Create a Python Script to Clean Data

Step 6.1: Type this and press **Shift + Enter**:

```
%%writefile /dataset2/clean_data.py
import glob

# Match all part files downloaded from Hadoop
file_list = glob.glob("/dataset2/part-m-0000*")
cleaned_data = []
```

```
for file_name in file_list:
    with open(file_name, "r", encoding="utf-8") as file:
        for line in file:
            parts = line.strip().split(",")
            if len(parts) >= 13:
                # Extract selected fields: staff_id, staff_card_number,
staff_full_name, sex, salary
                selected = parts[0:3] + [parts[5], parts[10]]
                cleaned_data.append(selected)


# Write cleaned data to a single output file
with open("/dataset2/staff_filtered.csv", "w", encoding="utf-8") as out:
    for row in cleaned_data:
        out.write(",".join(row) + "\n")
```

6.1

Step 6.2: Output confirms that the file is written.

```
# Write cleaned data to a single output file
with open("/dataset2/staff_filtered.csv", "w", encoding="utf-8") as out:
    for row in cleaned_data:
        out.write(",".join(row) + "\n")
```

```
Writing /dataset2/clean_data.py
```

6.2

Part 7/7: Execute the Clean Data Script and Analyze Data

Step 7.1: Type and run:

```
!python3 /dataset2/clean_data.py
!cd /dataset2 && ls
!cat /dataset2/staff_filtered.csv
```

7.1

Step 7.2: Output will be shown below.

```
clean_data.py  part-m-00001  part-m-00003  staff_filtered.csv
part-m-00000   part-m-00002  part-m-00004
15,C 004,SAK SEREY,Male,200.0
18,C 006,CHANMEAN TORN,Male,210.0
19,C 007,YI BUNLY,Male,205.0
```

7.2

Step 7.3: Type this code and press **Shift + Enter**:

```
with open("/dataset2/staff_filtered.csv", "r", encoding="utf-8") as file:
    data = file.readlines()

male_count = sum(1 for line in data if "Male" in line)
female_count = sum(1 for line in data if "Female" in line)

print(f"Male count: {male_count}")
print(f"Female count: {female_count}")
```