Practical no.: 07

# **Aim:** Install, configure and run Apache Spark. Create & transform RDDs.

# **Steps:**

### Step 1: Update System Packages

First, update and upgrade the system packages:

**sudo apt update && sudo apt upgrade -y**

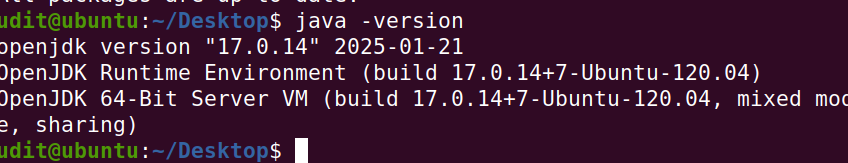
### Step 2: Install Java (OpenJDK 11)

Apache Spark requires Java. Install OpenJDK 11:

**sudo apt install openjdk-11-jdk -y**

Verify installation:

**java -version**

****

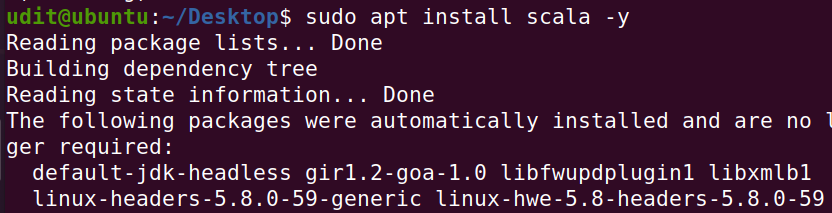
Expected output:

openjdk version "17.0.x" ...

### Step 3: Install Scala

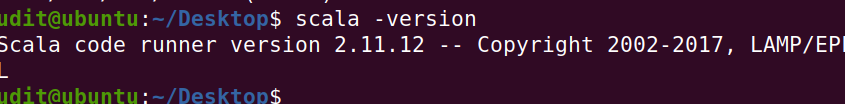
Spark requires Scala for execution:

**sudo apt install scala -y**

****

Verify installation:

**scala -version**

****

### Step 4: Download and Install Apache Spark

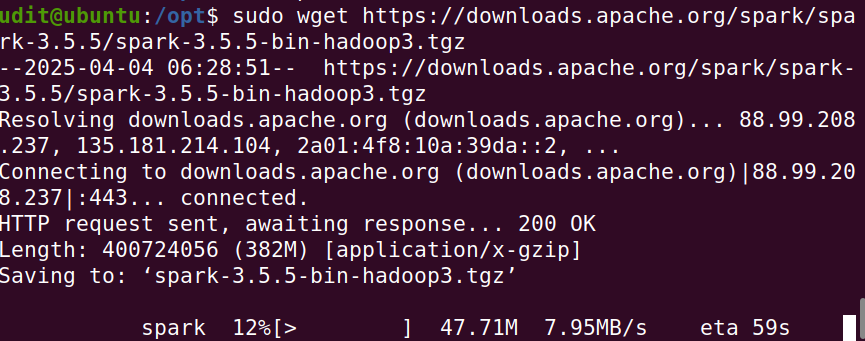
Download the latest **Apache Spark (3.x)** release:

**cd /opt**

****

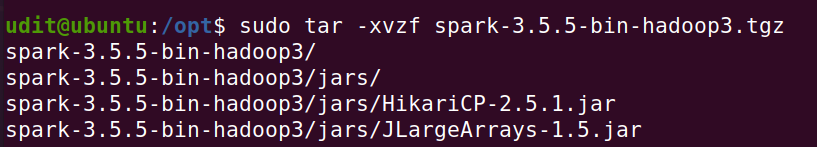
****

**sudo wget https://downloads.apache.org/spark/spark-3.5.5/spark-3.5.5-bin-hadoop3.tgz**

****

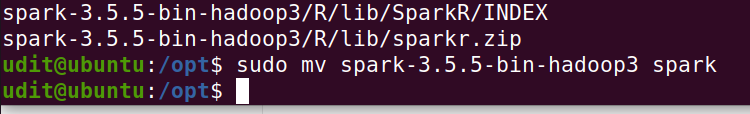
Extract the archive:

**sudo tar -xvzf spark-3.5.5-bin-hadoop3.tgz**

****

Rename the folder:

**sudo mv spark-3.5.5-bin-hadoop3 spark**



### Step 5: Set Up Environment Variables

Edit the. bashrc file:

**nano ~/.bashrc**

****

Add the following lines at the end:

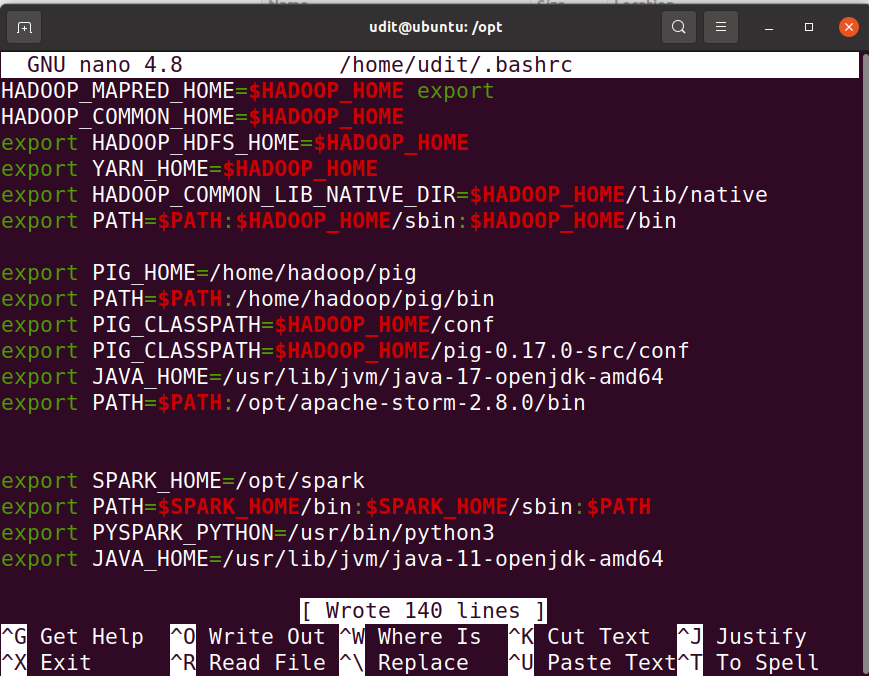
**export SPARK\_HOME=/opt/spark**

**export PATH=$SPARK\_HOME/bin:$SPARK\_HOME/sbin:$PATH**

**export PYSPARK\_PYTHON=/usr/bin/python3**

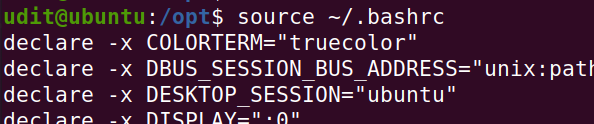
**export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64**

**Save and exit (Ctrl+S & Ctrl+X).**

****

Apply the changes:

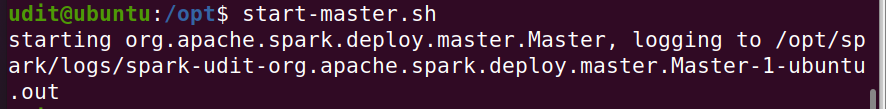
**source ~/.bashrc**

****

### Step 6: Start Spark

**Start Spark Master and Worker**

**start-master.sh**

****

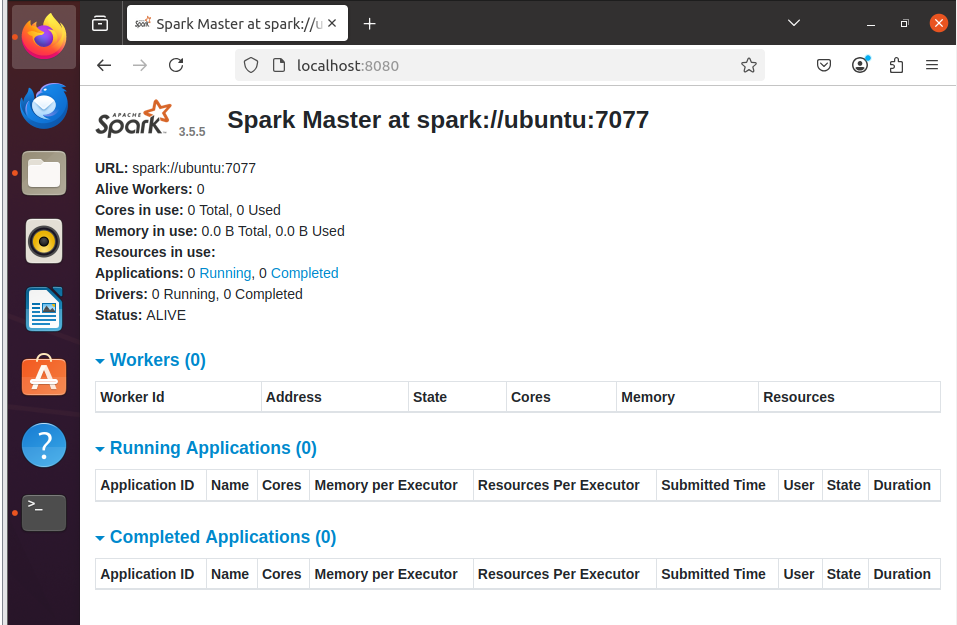
Find the **Spark Master URL** in the output (e.g., spark://yourhostname:7077).

**🔍 Next Steps: Verify Spark Master UI**

You can now access the **Spark Master Web UI** to confirm everything is working:

📍 Open your browser and go to:

[**http://localhost:8080**](http://localhost:8080)

****

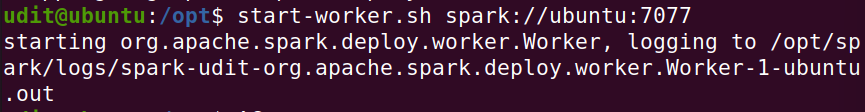
You should see a dashboard showing:

* Spark version
* Master URL (e.g., spark://ubuntu:7077)
* Worker nodes (currently 0 if none are started yet)

Now, start a worker:

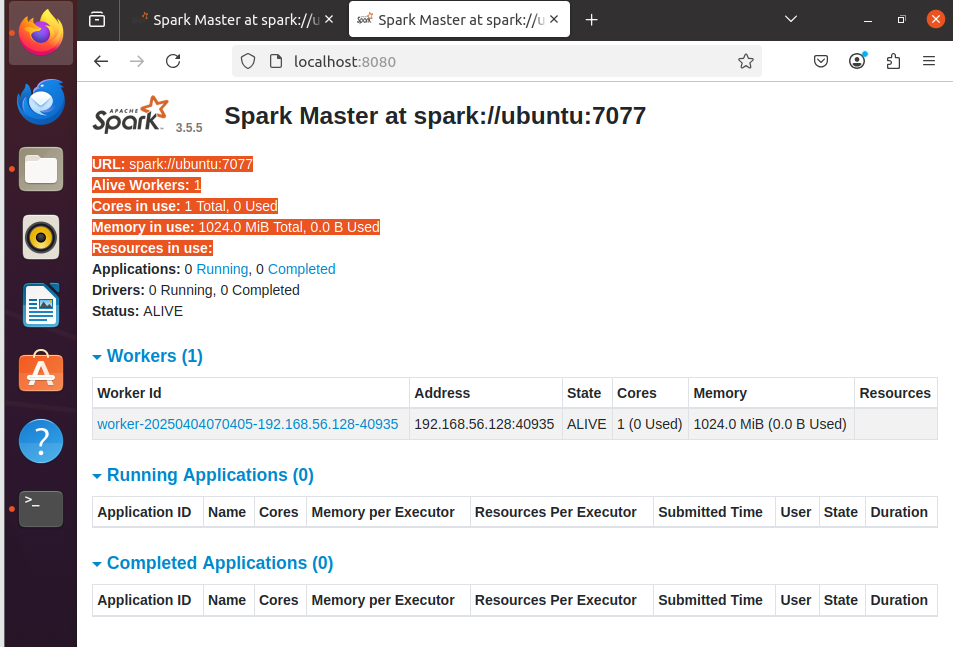
**### start-worker.sh spark://yourhostname:7077**

**start-worker.sh spark://ubuntu:7077**



To check if Spark is running, open a browser and go to:

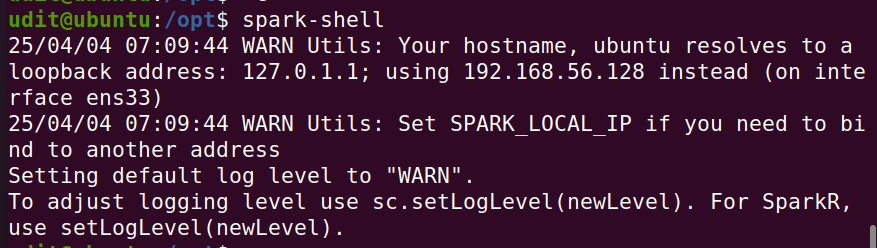
➡ [**http://localhost:8080**](http://localhost:8080)

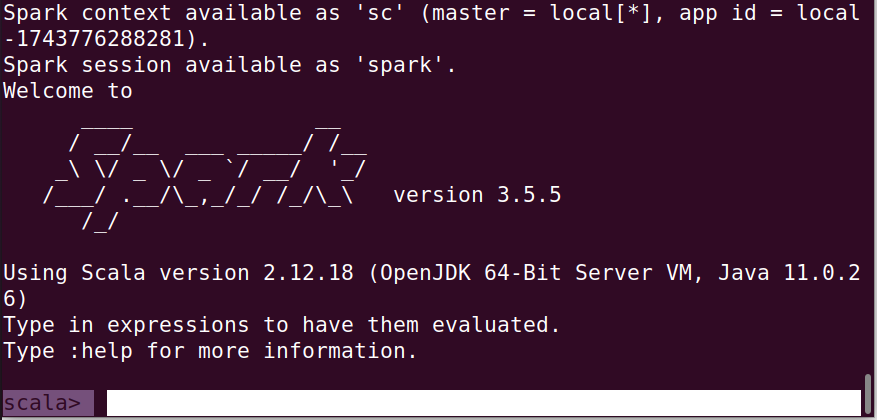
****

### Step 7: Start Spark Shell

Start the interactive Spark shell:

**spark-shell**

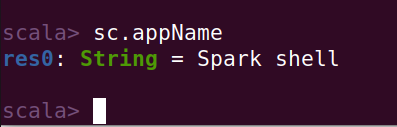
****

****

* ✅ Great — launching spark-shell successfully shows that Spark is working properly!
* The warning:
* Your hostname, ubuntu resolves to a loopback address: 127.0.1.1; using 192.168.56.128 instead
* is normal on virtual machines and doesn’t stop Spark from working. It simply means Spark prefers using your real IP (192.168.56.128) over the loopback.

Verify the installation by running:

**sc.appName**



If Spark is running correctly, it will return something like: **res0: String = Spark shell**

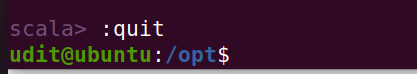
# For Creating and Transforming RDDs

❌ You cannot directly run PySpark (Python commands) in the Scala shell.

Instead, you need to exit and start PySpark separately

✅ Exit the current Spark shell by typing:

**:quit**

****

### Creating and Transforming RDDs

✅ Where to Perform These Operations?

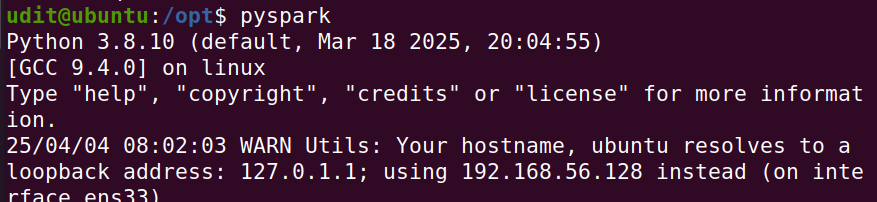
You should run the code inside the PySpark interactive shell.

**🧪 Step-by-Step Guide to Run PySpark RDD Operations**

### Step 1: Launch PySpark Shell

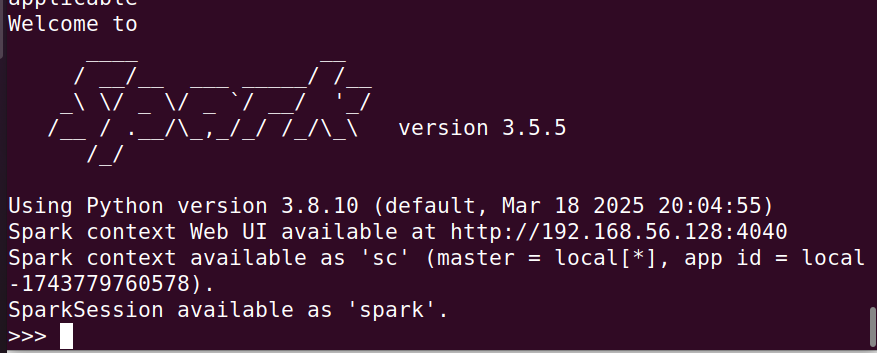
From your terminal (bash, like this: udit@ubuntu:/opt$), run:

Pyspark



If it's correctly installed, your prompt will change to:

>>> #######This is the PySpark (Python) shell.

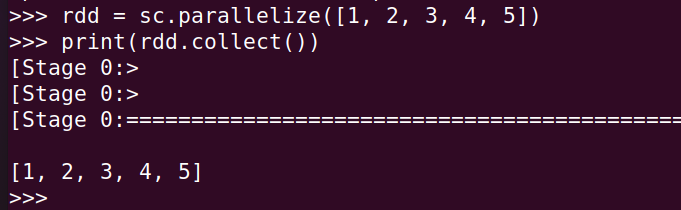


### Step 2: Create an RDD

Create an RDD from a Python list:

**rdd = sc.parallelize([1, 2, 3, 4, 5])**

**print(rdd.collect())**  # Output: [1, 2, 3, 4, 5]

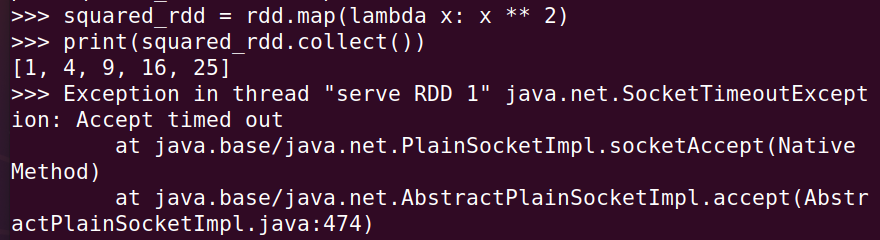


### Step 3: Apply Transformations

Apply **map()** transformation to square each number:

**squared\_rdd = rdd.map(lambda x: x \*\* 2)**

**print(squared\_rdd.collect()) # Output: [1, 4, 9, 16, 25]**

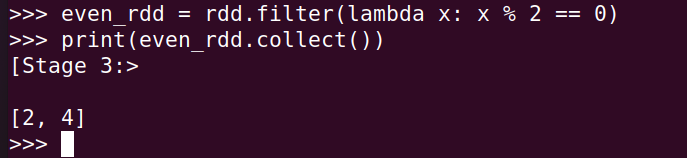


Your PySpark command executed successfully, However, you are seeing a **SocketTimeoutException**, which is a common issue in Spark when running in **local mode**. It does **not** affect the correctness of your computations, but it indicates a network communication timeout in Spark’s internal processes.

Apply **filter()** transformation to keep even numbers:

**even\_rdd = rdd.filter(lambda x: x % 2 == 0)**

**print(even\_rdd.collect())** # Output: [2, 4]

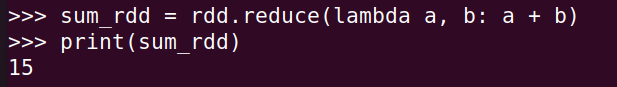
****

### Step 4: Perform Actions

Use the **reduce()** action to sum all elements:

**sum\_rdd = rdd.reduce(lambda a, b: a + b)**

**print(sum\_rdd) # Output: 15**



### Conclusion

* ✅ Installed and configured Spark on Ubuntu VM
* ✅ Created an RDD
* ✅ Performed transformations (map, filter)
* ✅ Executed actions (reduce, collect)