

## Unidad 2 - Tarea 3 - Procesamiento de Datos con Apache Spark

### Nombre del estudiante

Giovanny Alejandro Pardo

### Grupo:

Big Data (202016911\_27)

### Tutora

Sandra Milena Patino Avella

Universidad Nacional Abierta y a Distancia-UNAD

Escuela de ciencias básicas, tecnológicas e ingeniería

Ingeniería de sistemas

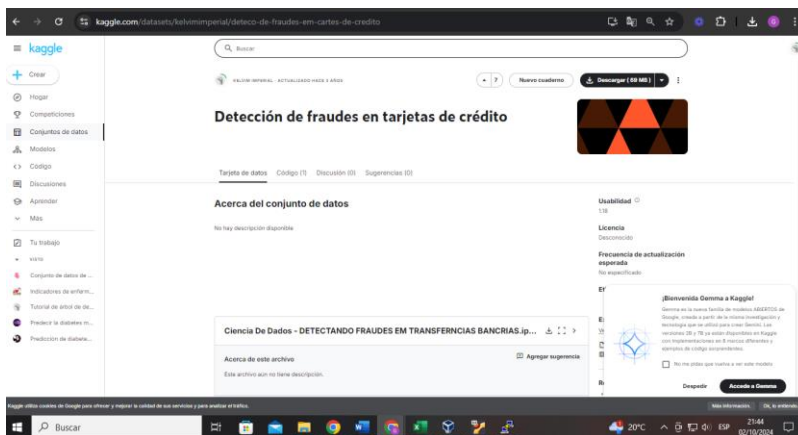
Palmira – octubre 03 del 2024

## Definición del problema y conjunto de datos

Para el desarrollo de la guía de trabajo y el tema “Procesamiento de Datos con Apache Spark” usare el dataframe de kaggle de nombre “**Detección de fraudes en tarjetas de crédito**”

“, el archivo csv tiene como nombre “transferencias” y tiene alrededor de 172792 registros.

Imagen 01



Nota: En la imagen se puede apreciar el sitio kaggle donde se descargó el dataframe.

Se procede a abrir el archivo y organizar para ver su estructura de datos.

Imagen 02

The image shows a Microsoft Excel spreadsheet with a large table of data. The columns are labeled: 'pais', 'ciudad', 'latitud', 'longitud', 'hora', 'minuto', and 'segundo'. The data is organized in rows, with the first row being a header. The spreadsheet is titled 'transferecias.csv' and is open in the 'Inicio' (Home) tab. The data is organized in columns, with the first column being 'pais' and the last column being 'segundo'. The data is organized in rows, with the first row being a header. The spreadsheet is titled 'transferecias.csv' and is open in the 'Inicio' (Home) tab. The data is organized in columns, with the first column being 'pais' and the last column being 'segundo'. The data is organized in rows, with the first row being a header.

Nota: En esta imagen se aprecia el archivo transferencia.csv organizado.

## Implementación en Spark

Antes de instalar Spark debemos actualizar los paquetes del sistema mediante la siguiente línea de comando:

```
>> sudo apt update
```

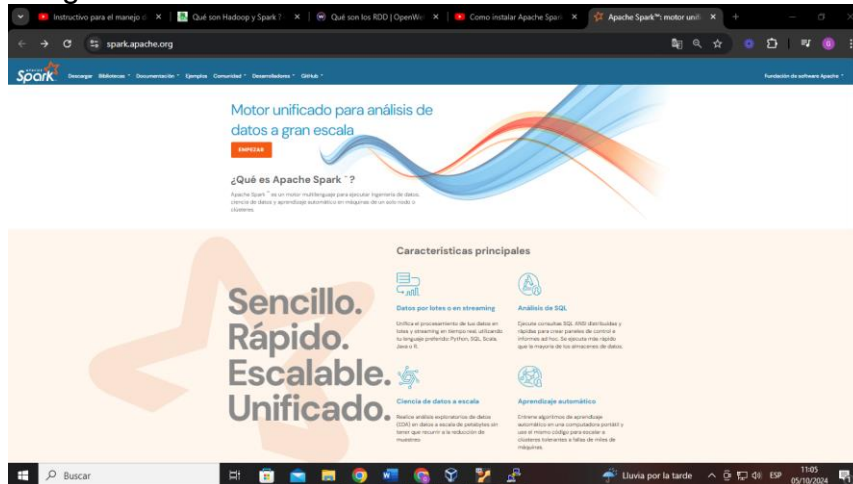
Imagen 03

The image shows a terminal window with the command prompt 'vifono@vifono:~\$'. The user has entered the command 'sudo apt update'. The output shows the system is updating its package lists and installing new packages. The terminal window is titled 'vifono.com/community/tutorial\_ho...'. The command prompt is 'vifono@vifono:~\$'. The user has entered the command 'sudo apt update'. The output shows the system is updating its package lists and installing new packages.

Nota: En esta imagen se evidencia la actualización de los paquetes del sistema.

Luego nos dirigimos a la página oficial de [spark](#) para descargar la versión mas actual que para este caso es la versión 3.5.3 del 24 septiembre del 2024.

## Imagen 04

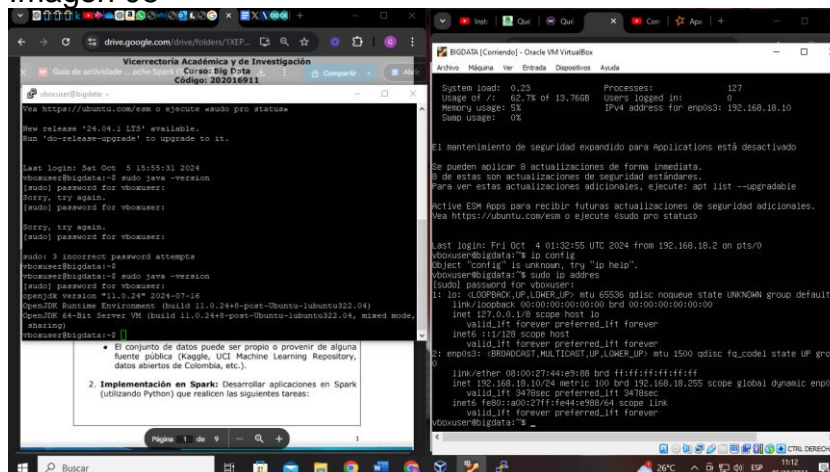


Nota: En esta imagen se evidencia el sitio oficial de Spark.

Dedemos previamente a la instalación tener instalado una dependencia requerida por Spark como es java, pero debemos validar si tenemos java y la versión para ello usamos la siguiente línea de comando.

```
>>$ sudo java -version
```

## Imagen 05



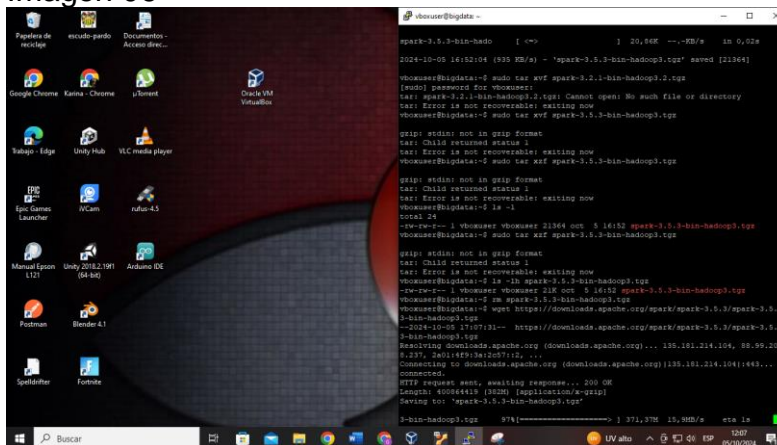
Nota: En esta imagen se evidencia que tenemos java instalado en la versión 11.0.24



Ahora si procedemos a descarga e instalar Apache Spark mediante la siguiente línea de comando.

>>\$ wget <https://www.apache.org/dyn/closer.lua/spark/spark-3.5.3/spark-3.5.3-bin-hadoop3.tgz>

Imagen 06

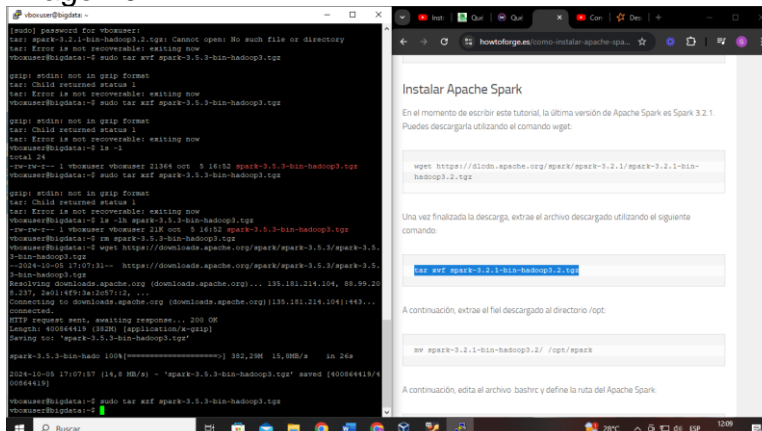


Nota: En esta imagen se evidencia que se descargó el archivo spark-3.5.3-bin-hadoop3.tgz

Ahora procedemos a extraer el archivo descargado mediante línea de comando.

>>\$ sudo tar xvf spark-3.5.3-bin-hadoop3.tgz

Imagen 07

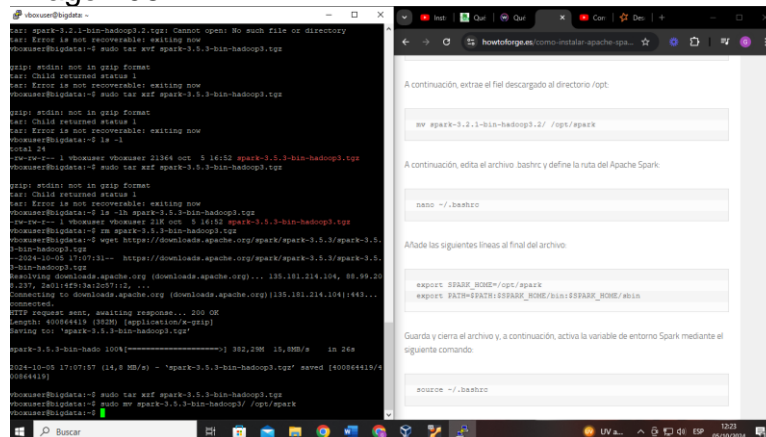


Nota: En esta imagen se evidencia que se descomprimió en archivo.

Ahora se procede a extraer el archivo al directorio /opt/spark mediante la siguiente línea de comando.

```
>>$ sudo mv spark-3.5.2-bin-hadoop3/ /opt/spark
```

Imagen 08



Nota: En esta imagen se evidencia que se movieron los archivos.

Ahora se procede a editar las variables de entorno en el archivo. bashrc para definirla ruta del Apache Spark mediante la siguiente línea de comando.

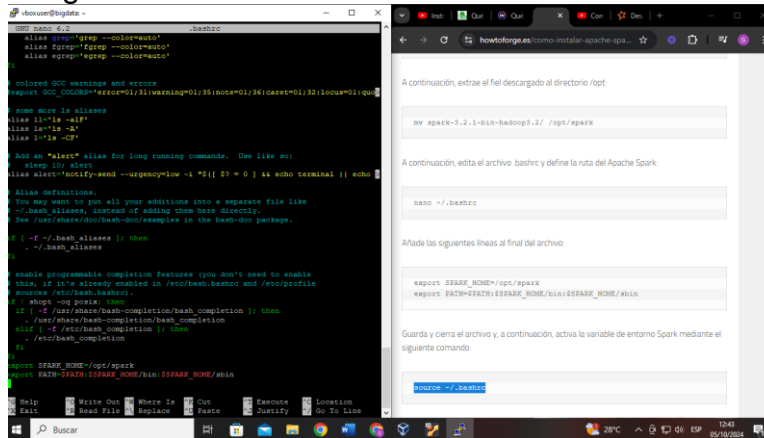
```
>>$ sudo nano ~/.bashrc
```

Luego añadimos las siguientes líneas de comando al final del archivo.

```
>>$ export SPARK_HOME=/opt/spark
```

```
>>$ export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin
```

Imagen 09



Nota: En esta imagen se evidencia la edición del archivo **bashrc**.

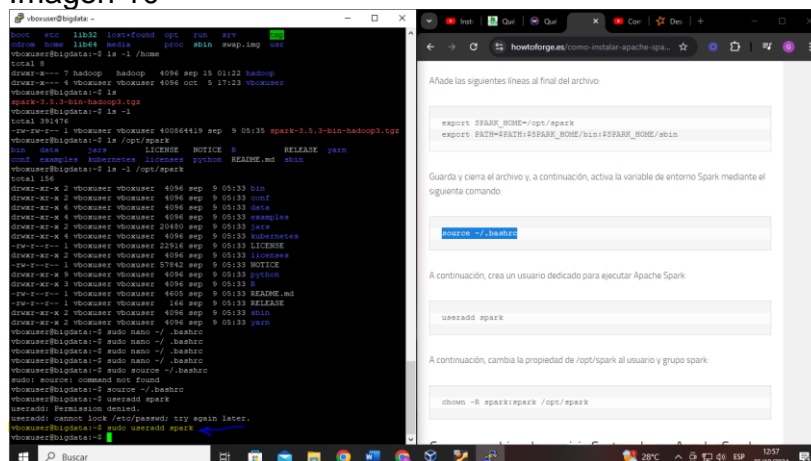
Guradamos y ceramos el archivo y activamos la variable de entorno de spark mediante la siguiente línea de comando

```
>>$ source ~/.bashrc
```

Y creamos un usuario dedicado para que ejecute Apache Spark mediante la siguiente línea de comando.

```
>>$ sudo useradd spark
```

Imagen 10

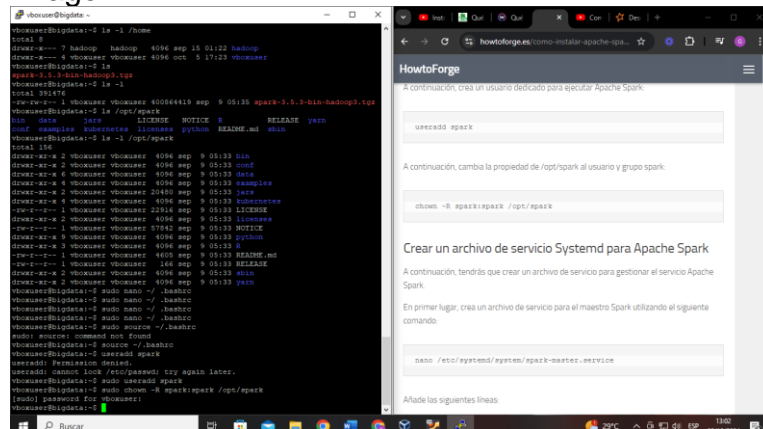


Nota: En esta imagen se evidencia la creación del usuario Spark.

Cambiamos la propiedad de /opt/spark al usuario y grupo spark mediante las siguiente linea de comando.

```
>>$ sudo chown -R spark:spark /opt/spark
```

Imagen 11



Nota: En esta imagen se evidencia que la capeta se asigna al usuario spark.

Se procede a crear un archivo de servicio para gestionar el servicio maestro Spark mediante las siguientes líneas de comando:

```
>>$ nano /etc/systemd/system/spark-master.service
```

Y escribimos estas líneas.

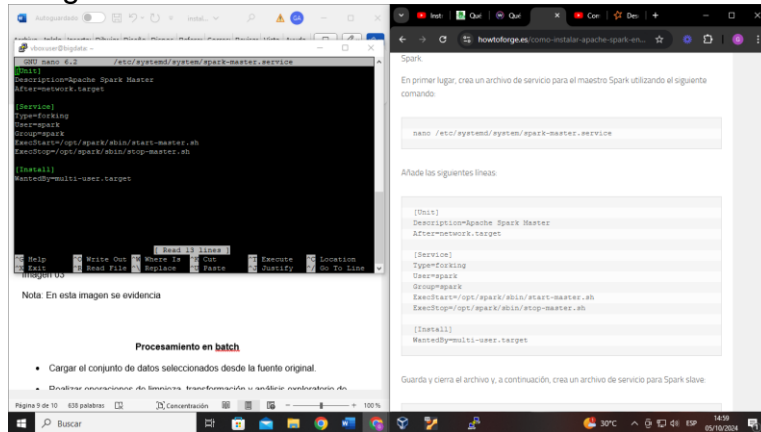
```
>> [Unit]
>> Description=Apache Spark Master
>> After=network.target

>> [Service]
>> Type=forking
>> User=spark
>> Group=spark
>> ExecStart=/opt/spark/sbin/start-master.sh
>> ExecStop=/opt/spark/sbin/stop-master.sh

>> [Install]
>> WantedBy=multi-user.target
```



Imagen 12



Nota: En esta imagen se evidencia que se agregó las líneas de comando.

Ahora creamos el archivo para el servicio para Spark slave mediante las siguientes líneas de comando:

```
>>$ sudo nano /etc/systemd/system/spark-slave.service
```

Y escribimos estas líneas.

```
>> [Unit]
```

```
>> Description=Apache Spark Slave
```

```
>> After=network.target
```

```
>> [Service]
```

```
>> Type=forking
```

```
>> User=spark
```

```
>> Group=spark
```

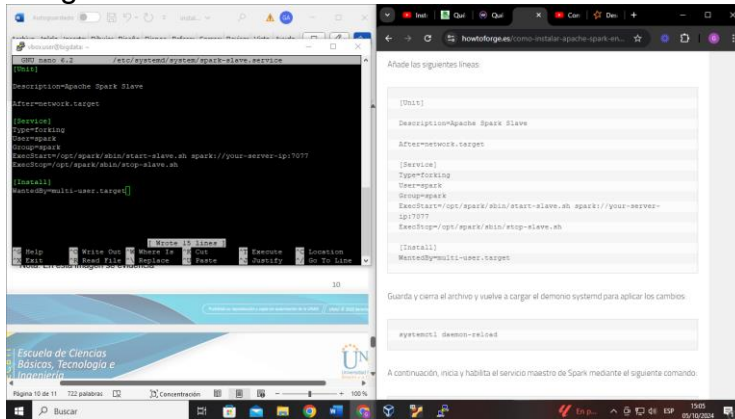
```
>> ExecStart=/opt/spark/sbin/start-slave.sh spark://your-server-ip:7077
```

```
>> ExecStop=/opt/spark/sbin/stop-slave.sh
```

```
>> [Install]
```

```
>> WantedBy=multi-user.target
```

Imagen 13

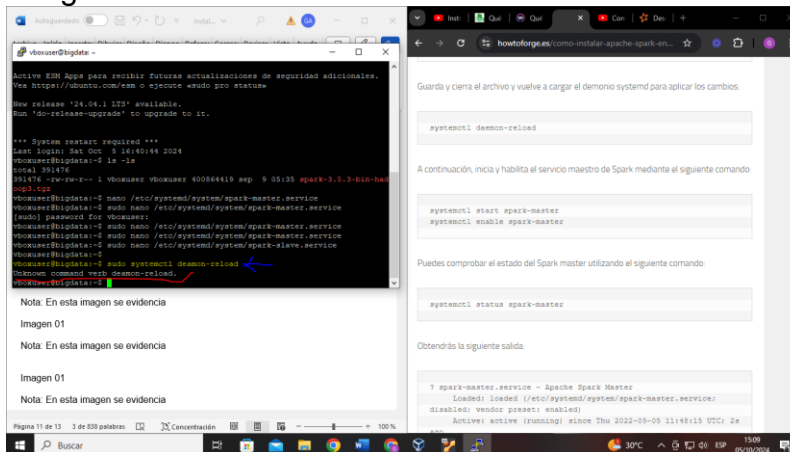


Nota: En esta imagen se evidencia la creación del archivo Spark slave.

Se procede a volver a cargar el demonio systemd para aplicar los cambios mediante la siguiente línea de comando:

```
>>$ sudo systemctl daemon-reload
```

Imagen 14



Nota: En esta imagen se evidencia que se reiniciaron los servicios **systemctl**.

Ahora iniciamos y habilitamos el servicio maestro de Spark mediante la siguiente línea de comando:

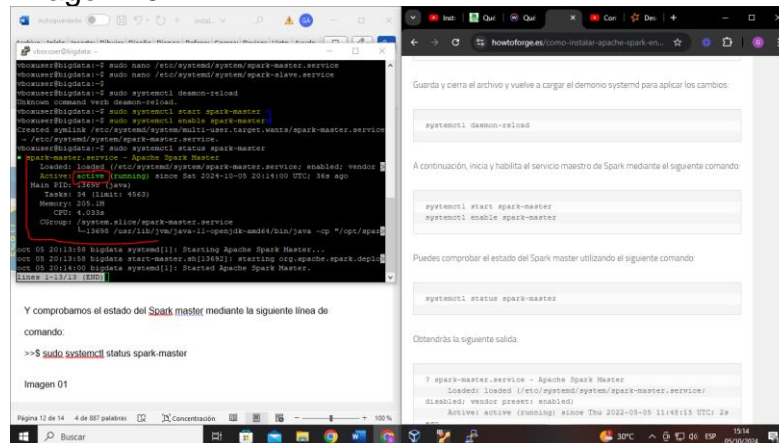
```
>>$ sudo systemctl start spark-master
```

```
>>$ sudo systemctl enable spark-master
```

Y comprobamos el estado del Spark master mediante la siguiente línea de comando:

```
>>$ sudo systemctl status spark-master
```

Imagen 15

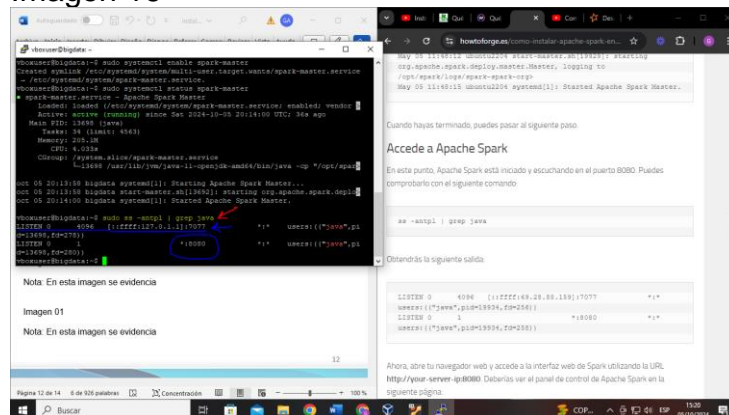


Nota: En esta imagen se evidencia el estado de **spark-master**.

Ahora procedemos a acceder a Apache Spark ya que en este punto ya está inicializado y escuchando por el puerto 8080 lo cual se puede comprobar mediante la siguiente línea de comando:

```
>>$ sudo ss -antpl | grep java
```

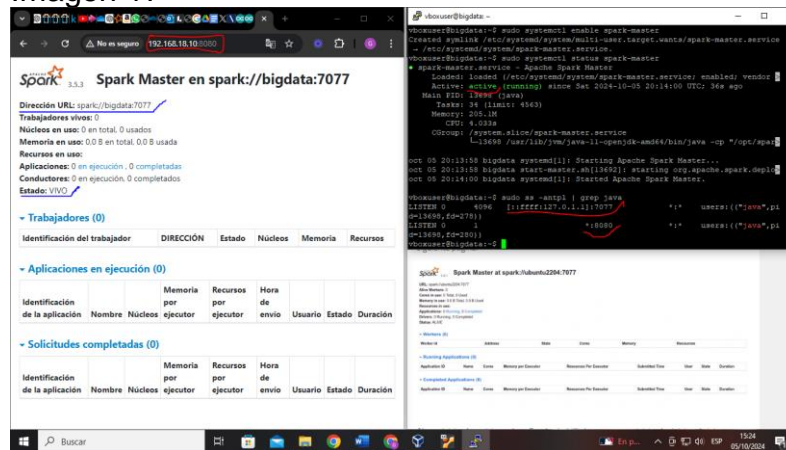
Imagen 16



Nota: En esta imagen se evidencia que Spark master escucha por el puerto 8080.

Ahora en nuestro navegador en una nueva pestaña escribimos la siguiente ruta <http://192.168.18.10:8080> y nos habra el panel de control de Apache Spark.

Imagen 17



Nota: En esta imagen se evidencia que tenemos abierto el panel de control de Apache Spark en nuestro navegador de nuestra PC.

Ahora procederemos a iniciar los servicios esclavos Spark y los ahilitamos para que se inicien al reiniciar el sistema mediante la siguiente liena de comandos:

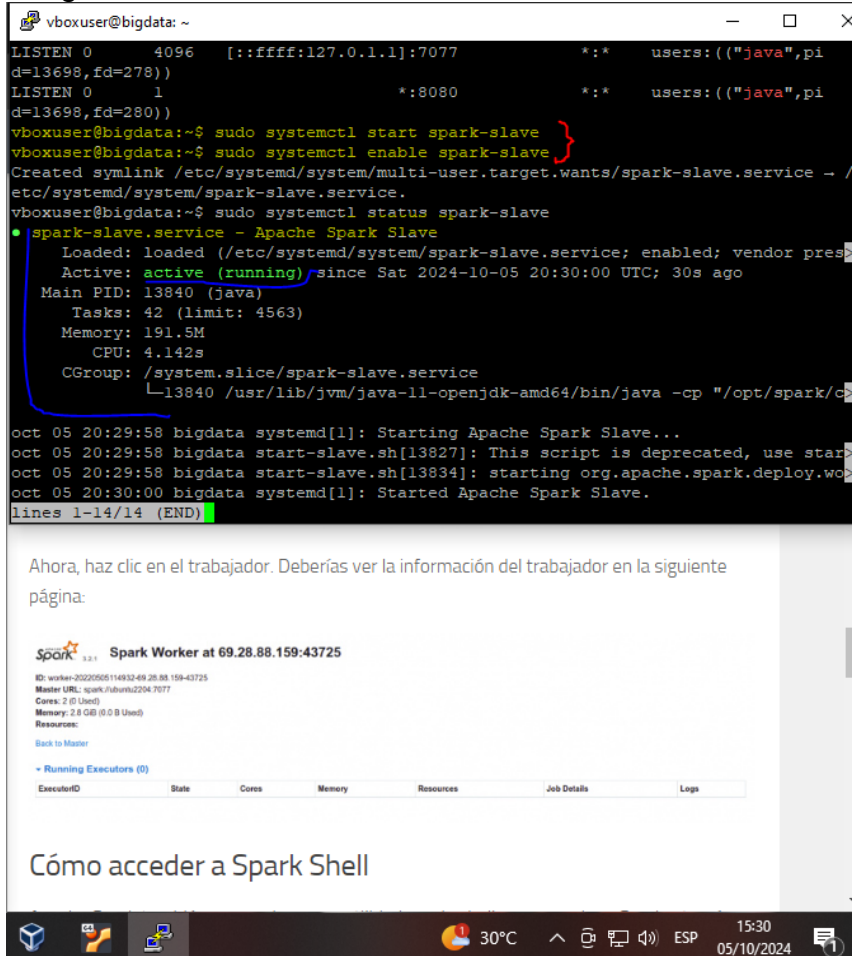
```
>>$ sudo systemctl start spark-slave
```

```
>>$ sudo systemctl enable spark-slave
```

Y comprobamos el estado del Spark master mediante la siguiente línea de comando:

```
>>$ sudo systemctl status spark-slave
```

Imagen 18



```

vboxuser@bigdata: ~
LISTEN 0      4096      [::ffff:127.0.1.1]:7077      *:*    users: (("java",pi
d=13698,fd=278))
LISTEN 0      1          *:8080          *:*    users: (("java",pi
d=13698,fd=280))
vboxuser@bigdata:~$ sudo systemctl start spark-slave
vboxuser@bigdata:~$ sudo systemctl enable spark-slave
Created symlink /etc/systemd/system/multi-user.target.wants/spark-slave.service - /
etc/systemd/system/spark-slave.service.
vboxuser@bigdata:~$ sudo systemctl status spark-slave
● spark-slave.service - Apache Spark Slave
   Loaded: loaded (/etc/systemd/system/spark-slave.service; enabled; vendor prese
   Active: active (running) since Sat 2024-10-05 20:30:00 UTC; 30s ago
   Main PID: 13840 (java)
     Tasks: 42 (limit: 4563)
    Memory: 191.5M
       CPU: 4.142s
   CGroup: /system.slice/spark-slave.service
           └─13840 /usr/lib/jvm/java-11-openjdk-amd64/bin/java -cp "/opt/spark/c

oct 05 20:29:58 bigdata systemd[1]: Starting Apache Spark Slave...
oct 05 20:29:58 bigdata start-slave.sh[13827]: This script is deprecated, use start
oct 05 20:29:58 bigdata start-slave.sh[13834]: starting org.apache.spark.deploy.wo
oct 05 20:30:00 bigdata systemd[1]: Started Apache Spark Slave.
lines 1-14/14 (END)

```

Ahora, haz clic en el trabajador. Deberías ver la información del trabajador en la siguiente página:

**Spark Worker at 69.28.88.159-43725**

ID: worker-20220505114932-49.28.88.159-43725  
 Master URL: spark://burns2204.7077  
 Cores: 2 (0 Used)  
 Memory: 2.0 GB (0.0 B Used)  
 Resources:

[Back to Master](#)

Running Executors (0)

ExecutorID	State	Cores	Memory	Resources	Job Details	Logs
------------	-------	-------	--------	-----------	-------------	------

Cómo acceder a Spark Shell

Nota: En esta imagen se evidencia que en el navegador se abre la consola de spark.

Ahora recargamos nuestra página donde se abrió el panel de control Spark donde se apreciará el trabajo añadido (slave).

Imagen 19

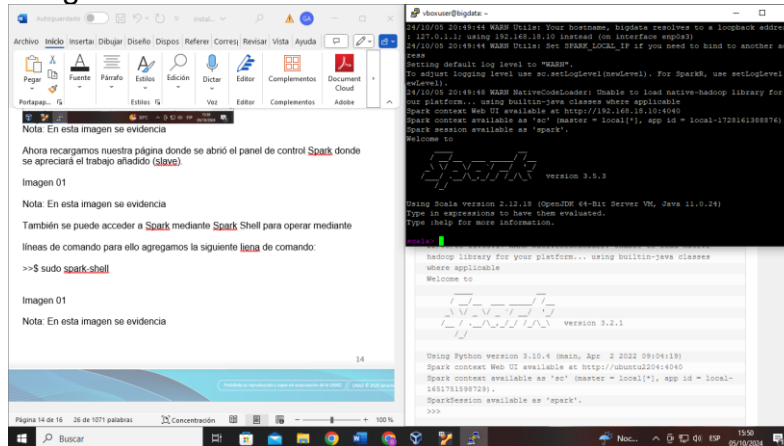
Nota: En esta imagen se evidencia

También se puede acceder a Spark mediante Spark Shell para operar mediante líneas de comando para ello agregamos la siguiente línea de comando:

```
>>$ sudo spark-shell
```



Imagen 20



Nota: En esta imagen se evidencia la consola de comando **spark-shell**.

Para salir solo basta con pulsar CTRL + D