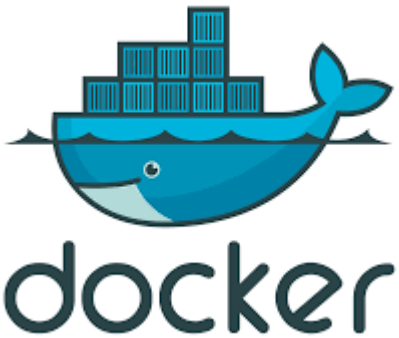
Rapport du projet



Belhouchet Maram
2BD2

Plan

I.Introduction

II. Hadoop sur Docker

Installation de Docker
Installation d'Apache Hadoop
Utilisation de Docker avec Hadoop
Exécution de tâches MapReduce
Captures d'écran du travail réalisé

III. Spark dans Docker

Installation de Spark
Téléchargement de la dataset pour les analyses
Manipulation des données avec Spark

IIII. Erreurs et Problèmes Rencontrés

Gestion des conteneurs Docker Hadoop
Erreurs Courantes
Solution pour l'erreur de safemode

I.Introduction:

Ce projet explore l'utilisation de Docker pour créer des environnements conteneurisés, intégrant Apache Hadoop et Spark pour le traitement distribué de données massives. Nous présentons les étapes d'installation, la configuration des clusters, et la réalisation d'analyses de données à l'aide de MapReduce et de Spark. Des solutions aux erreurs rencontrées sont également fournies, offrant ainsi une initiation pratique au Big Data distribué.

II-Hadoop sur docker:

1/installation de docker :

sudo apt update

sudo apt install apt-transport-https ca-certificates curl software-properties-common

description:Cette commande installe les paquets requis pour permettre à Apt d'utiliser des référentiels HTTPS, de gérer les certificats SSL, de télécharger des fichiers via curl, et de gérer les référentiels de logiciels à l'aide de `add-apt-repository`.

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

description:Cette commande télécharge la clé GPG officielle de Docker à partir de `https://download.docker.com/linux/ubuntu/gpg`, la passe à `apt-key add -` pour l'ajouter à votre trousseau de clés. Cela garantit l'authenticité des paquets Docker téléchargés.

sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu focal stable"

description:Cette commande ajoute le référentiel Docker à la liste des sources de paquets gérées par Apt. Le référentiel est configuré pour prendre en charge l'architecture `amd64` (64 bits) et est spécifique à la version `focal` d'Ubuntu. La branche `stable` est spécifiée pour installer la version stable de Docker.

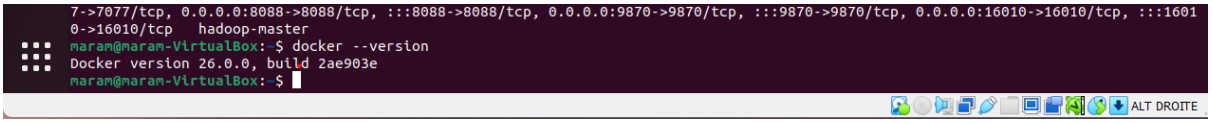
sudo apt update

apt-cache policy docker-ce

description:En exécutant cette commande, vous obtiendrez une sortie qui affiche les détails de la politique de paquet pour Docker CE, y compris les versions disponibles, leurs priorités et les dépôts à partir desquels elles sont obtenues. Cela peut être utile pour vérifier quelle version de Docker CE est installée sur votre système et à partir de quel référentiel elle provient.

sudo apt install docker-ce

description:En exécutant cette commande avec les privilèges appropriés, Apt téléchargera et installera Docker CE sur votre système. Une fois l'installation terminée, Docker sera prêt à être utilisé.



2/j'ai installé Apache Hadoop version3.3.6

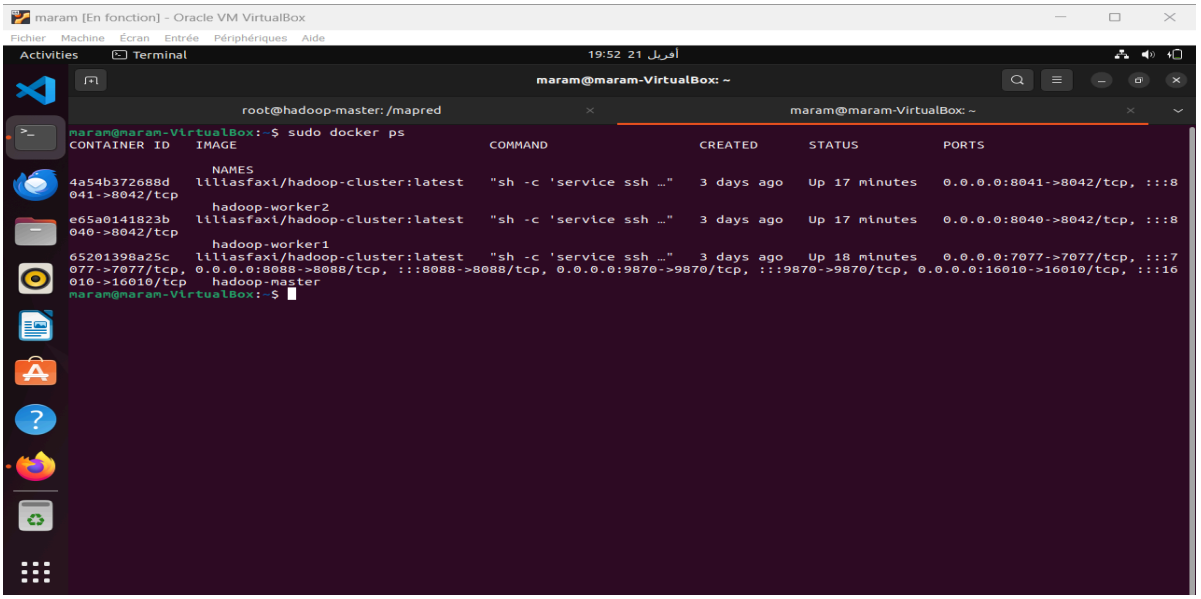
3/avec l'image de docker uplodée sur dockerhub j'ai la téléchargé :

`docker pull liliasfaxi/hadoop-cluster:latest`

`4/ création` un réseau qui permettra de relier les trois contenaires :
docker network create --driver=bridge hadoop
5/ Créer et lancer les trois contenaires (les instructions -p permettent de faire un mapping entre les ports de la machine hôte et ceux du contenaire):
sudo docker run -itd --net=hadoop -p 9870:9870 -p 8088:8088 -p 7077:7077 -p 16010:16010 --name hadoop-master --hostname hadoop-master liliasfaxi/hadoop-cluster:latest

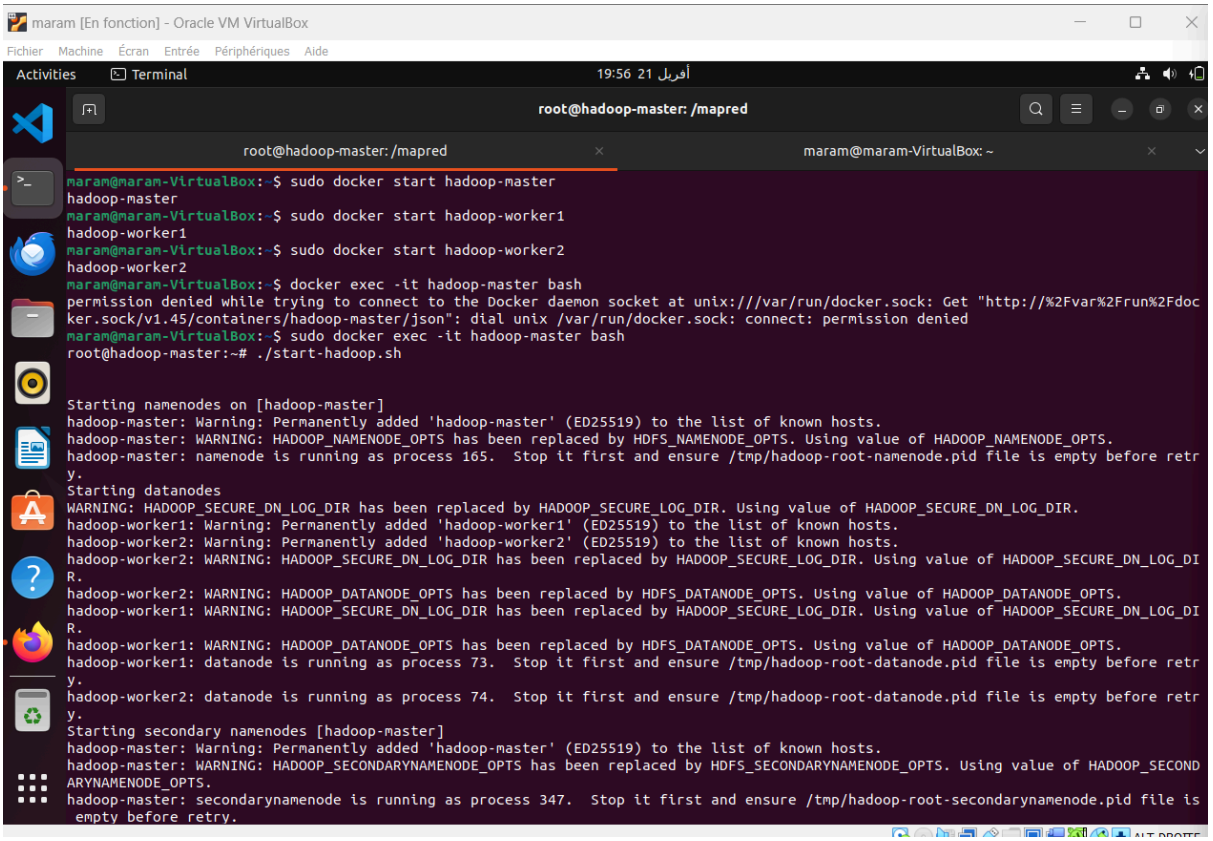
sudo docker run -itd -p 8040:8042 --net=hadoop --name hadoop-worker1 --hostname hadoop-worker1 liliasfaxi/hadoop-cluster:latest

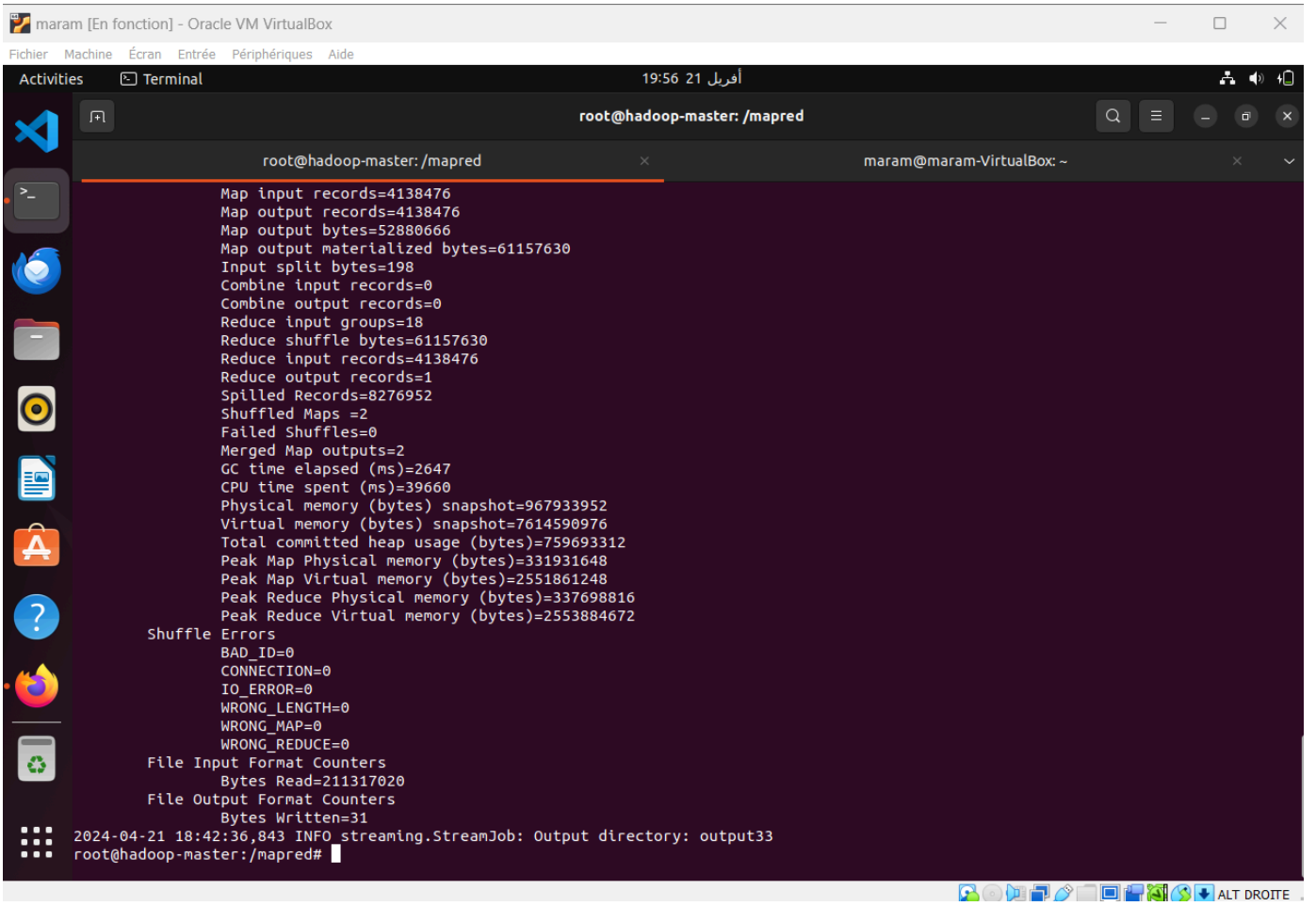
sudo docker run -itd -p 8041:8042 --net=hadoop --name hadoop-worker2 --hostname hadoop-worker2 liliasfaxi/hadoop-cluster:latest
5/vérifier les 3 contenaires avec docker ps :



6/exécuter hadoop master , lancer hadoop lancer le job mapreduce aprés création de mapper et reducer et les mettre dans un répertoire j'ai la nommée mapred et puis le copier vers le master voila les commandes et les captures d'écran sur mon travail:
1/Créer un répertoire dans HDFS, appelé *input:*
hdfs dfs –mkdir -p input
2/Commencer par décompresser le fichier sur mon machine, puis par le charger dans le contenaire `hadoop-master` *avec la commande suivante:*
docker cp purchases.txt hadoop-master:/root/purchases.txt
7/changer directory vers mapred dans lequel j'ai mis les files mapper et reducer:
cd mapred
6/lancer le mapreduce sur docker :
hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar -file mapper.py -mapper "python3 mapper.py" -file reducer.py -reducer "python3 reducer.py" -input /myinput/purchases.txt -output output33

7/captures sur mon travail :





III-Spark dans Docker:

1/installation de spark:

L'image que j'ai utilisé contient spark sans installation mais je peux l'installer par :

docker pull docker-spark

avec docker-spark est l'image docker.

2/télécharger la dataset pour lancer les analyses :

sudo docker cp purchases.txt hadoop-master:/home/maram/Downloads/sal.csv

description:moi j'ai choisit un dataset sur les salaires des software professional

et puis j'ai la copié du local au hadoop-master avec la commande :

sudo docker cp purchases.txt hadoop-master:/home/maram/Downloads/sal.csv

3/hdfs dfs –put sal.csv input

description: La commande `hdfs dfs -put sal.csv input` est utilisée pour copier le fichier `sal.csv` du système de fichiers local vers le système de fichiers Hadoop (HDFS), dans le répertoire spécifié `input`.

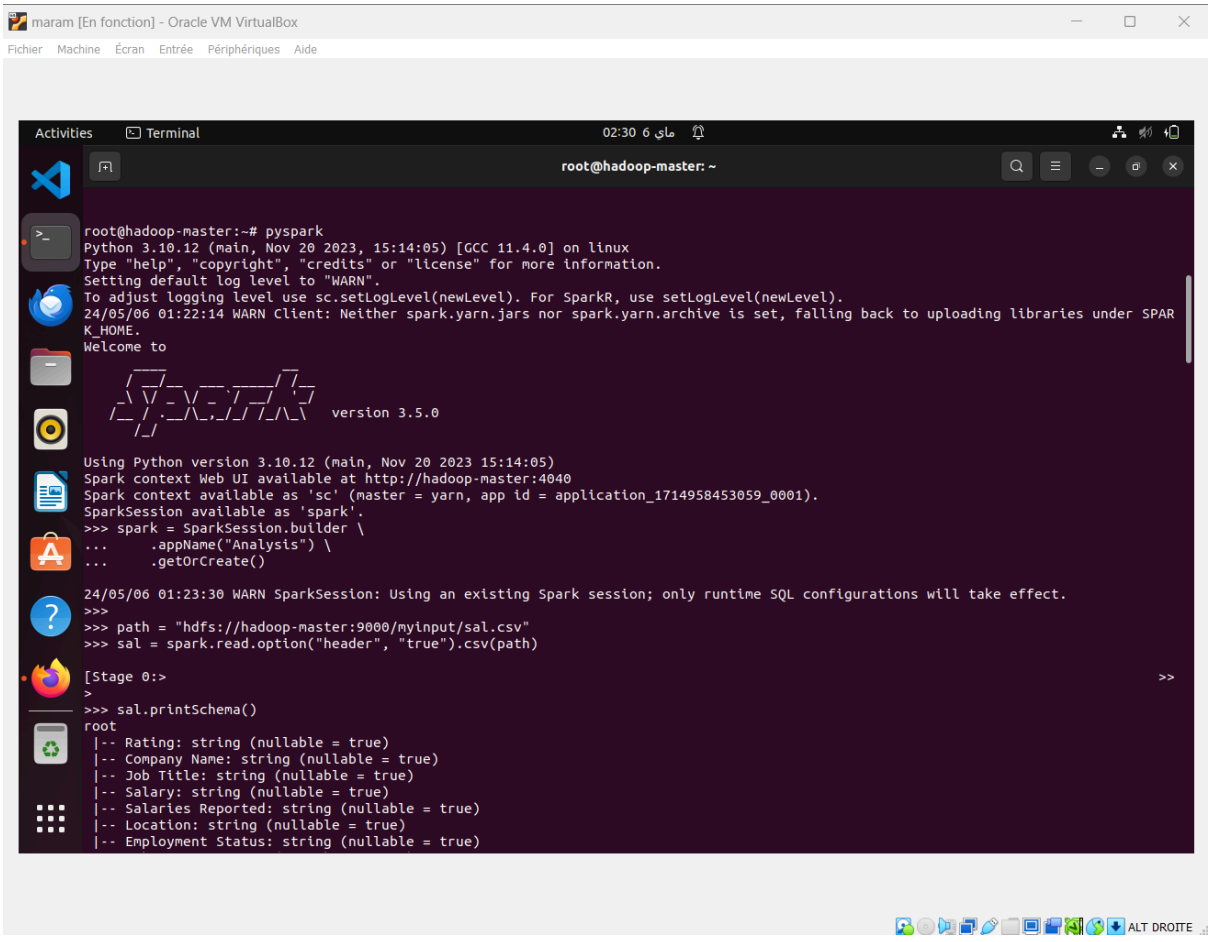
4/hadoop fs -put sal.csv myinput/

description:

La commande `hadoop fs -put sal.csv myinput/` est utilisée pour copier le fichier `sal.csv` du système de fichiers local vers le système de fichiers Hadoop (HDFS), dans le répertoire spécifié `myinput/`.

5/spark-shell

description: exécuter cette commande pour commencer les analyses



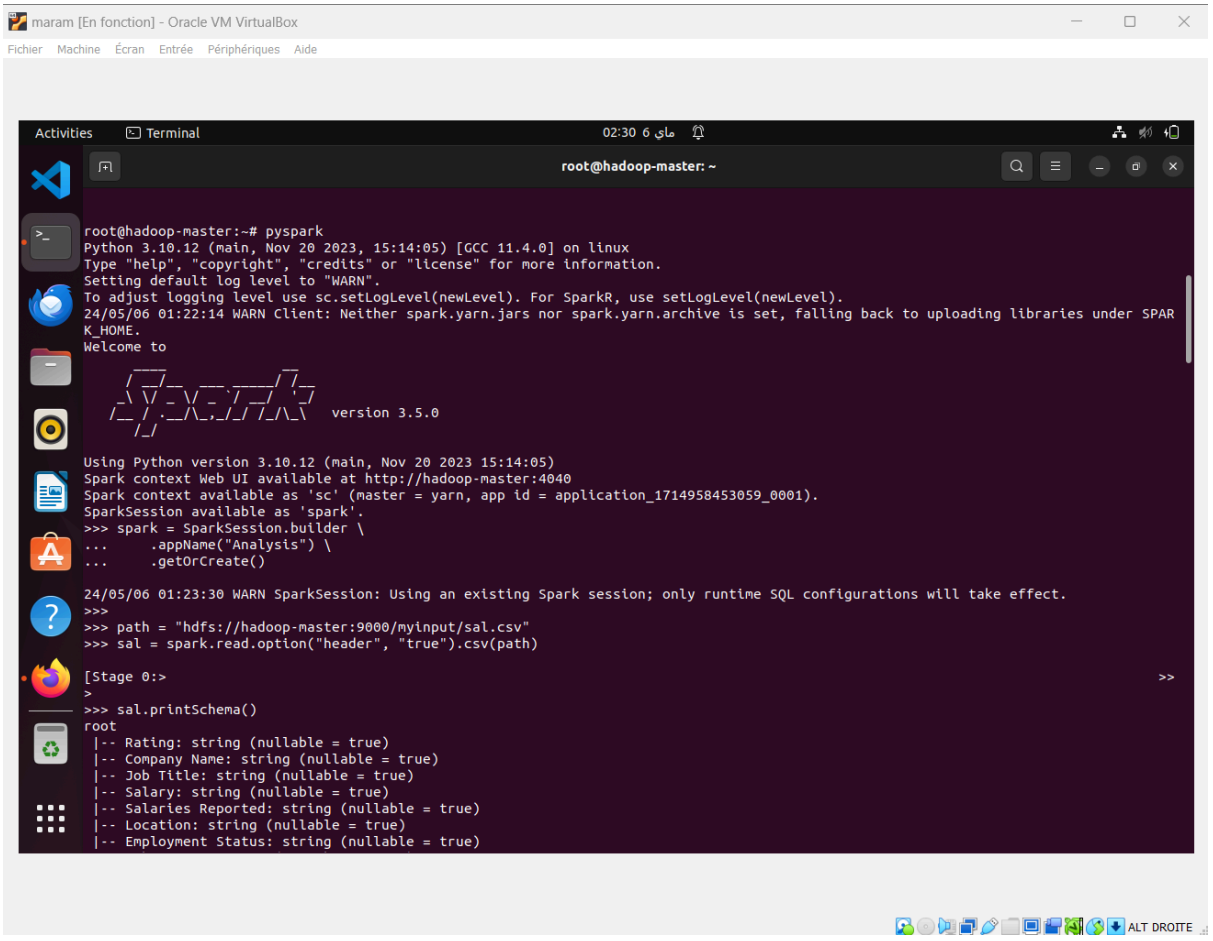
6/val path = "hdfs://hadoop-master:9000/myinput/sal.csv"
val sal = spark.read.option("header", "true").csv(path)
description:Ces lignes de code Scala sont utilisées dans un environnement Spark pour lire un fichier CSV à partir du système de fichiers Hadoop (HDFS) et créer un DataFrame à l'aide de Spark.
7/ lancer les analyses sur scala et pyspark (j'ai mis cela dans le fichier code spark)

Code SPARK

1/ code scala:

1. Charger le fichier CSV et afficher le schéma des données :

val path = "hdfs://hadoop-master:9000/myinput/sal.csv"
val sal = spark.read.option("header", "true").csv(path)
sal.printSchema()

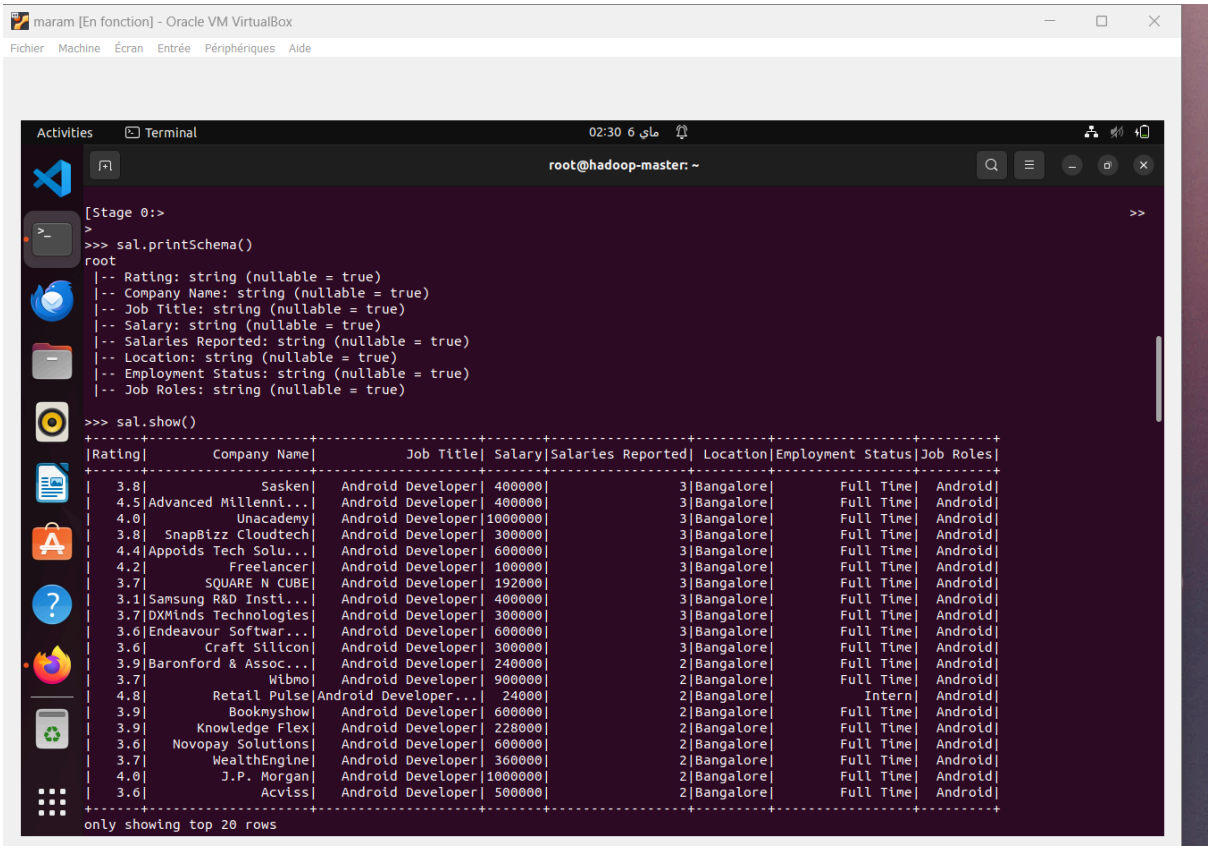


description:Cette commande charge le fichier CSV depuis HDFS dans un DataFrame Spark appelé `sal` et affiche le schéma des données, montrant les noms des colonnes et leurs types.

2. Afficher les premières lignes du DataFrame :

sal.show()

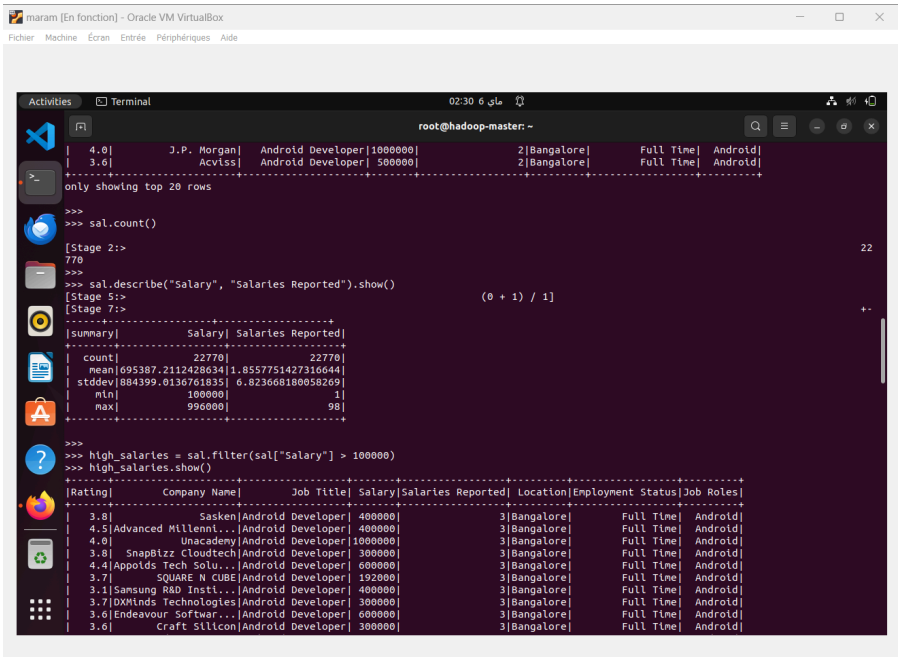
description:Cela affiche les premières lignes du DataFrame `sal`.



3. Compter le nombre total de lignes dans le DataFrame :

sal.count()

description:Cette commande compte le nombre total de lignes dans le DataFrame `sal`.

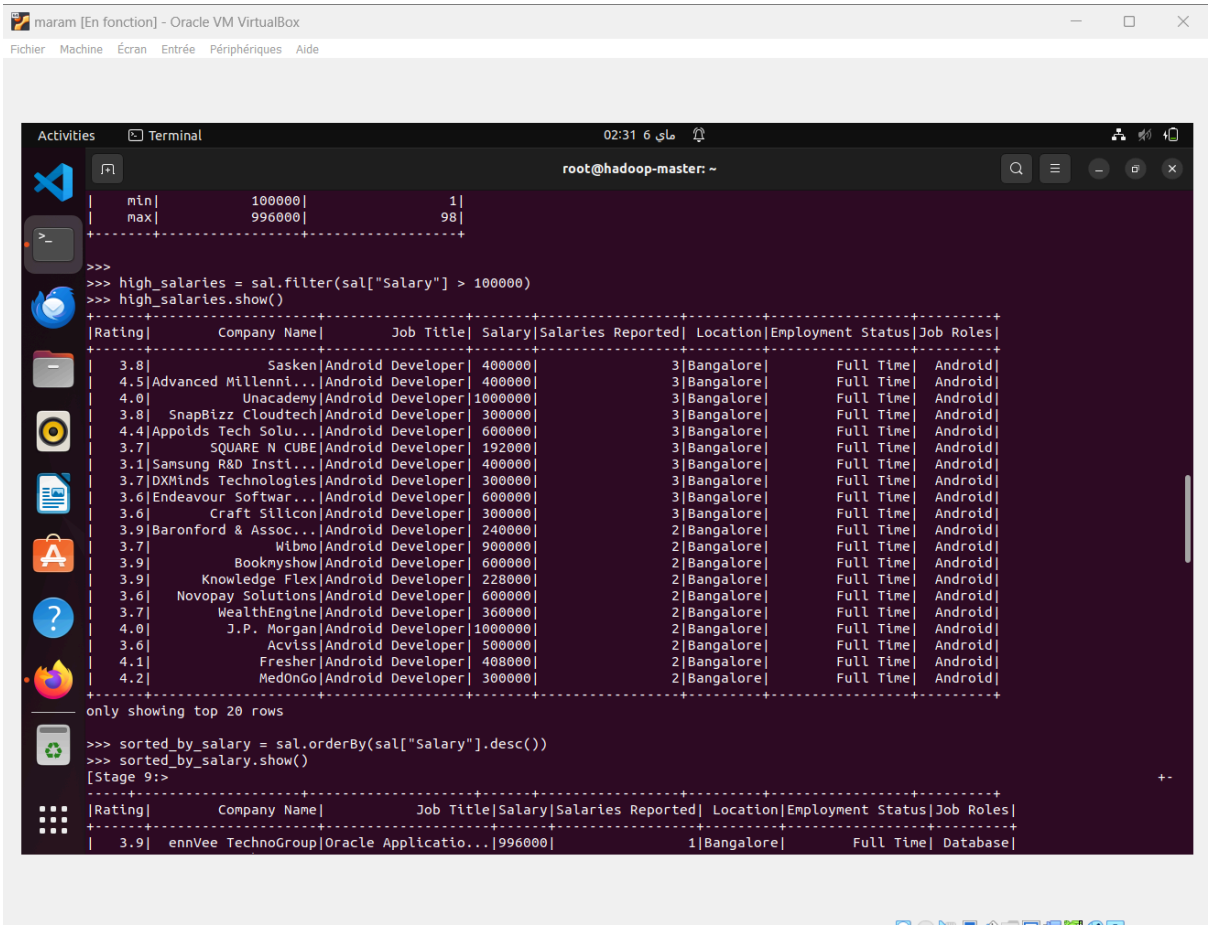


4. Afficher des statistiques descriptives pour les colonnes "Salary" et "Salaries Reported" :

sal.describe("Salary", "Salaries Reported").show()
description:Cela affiche des statistiques descriptives telles que la moyenne, l'écart type, le minimum, le maximum, etc., pour les colonnes spécifiées.

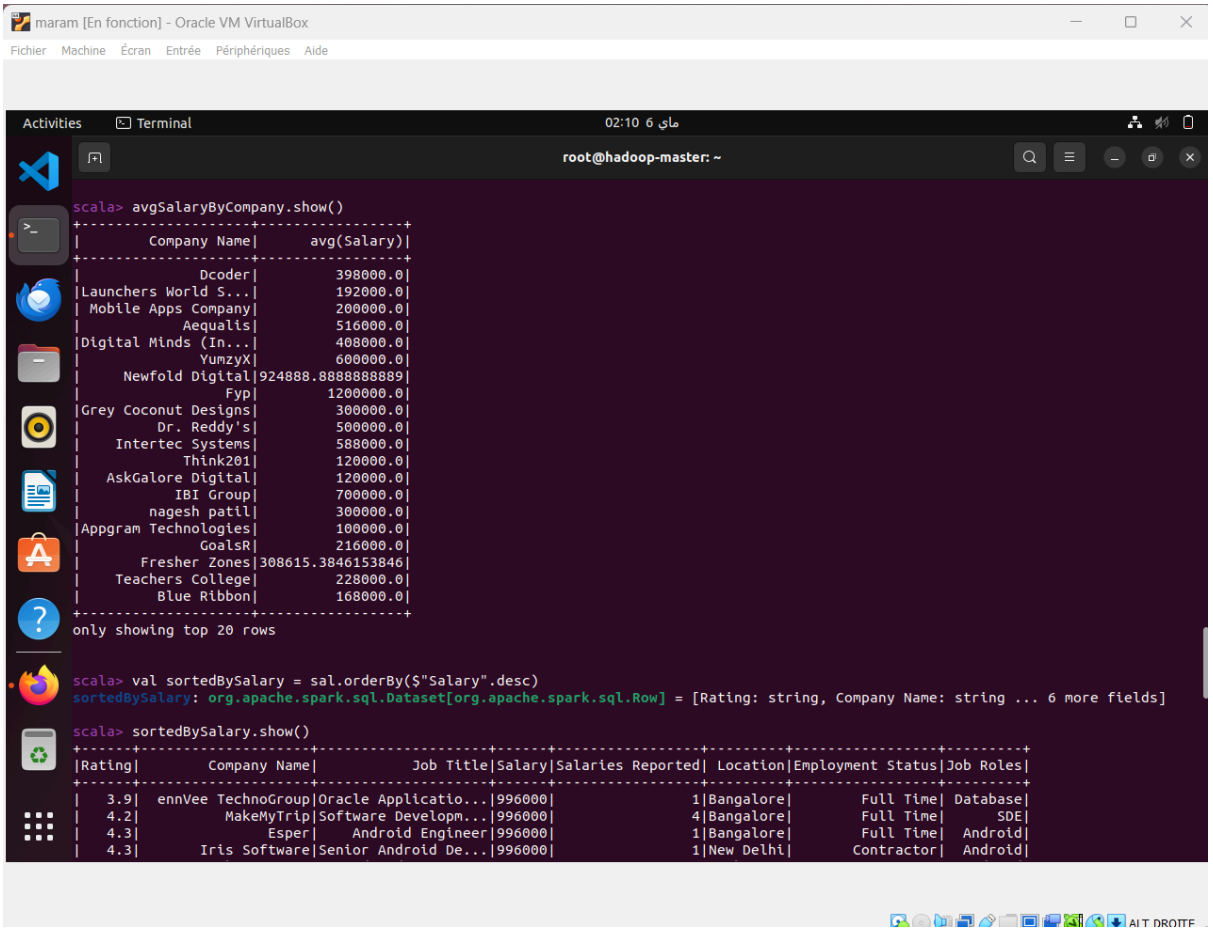
5. Filtrer les lignes avec un salaire supérieur à 100 000 :

val highSalaries = sal.filter($"Salary" > 100000)
highSalaries.show()
description:Cela filtre les lignes du DataFrame où la valeur de la colonne "Salary" est supérieure à 100 000 et affiche le résultat.



6. Calculer le salaire moyen par entreprise :

val avgSalaryByCompany = sal.groupBy("Company Name").avg("Salary")
avgSalaryByCompany.show()

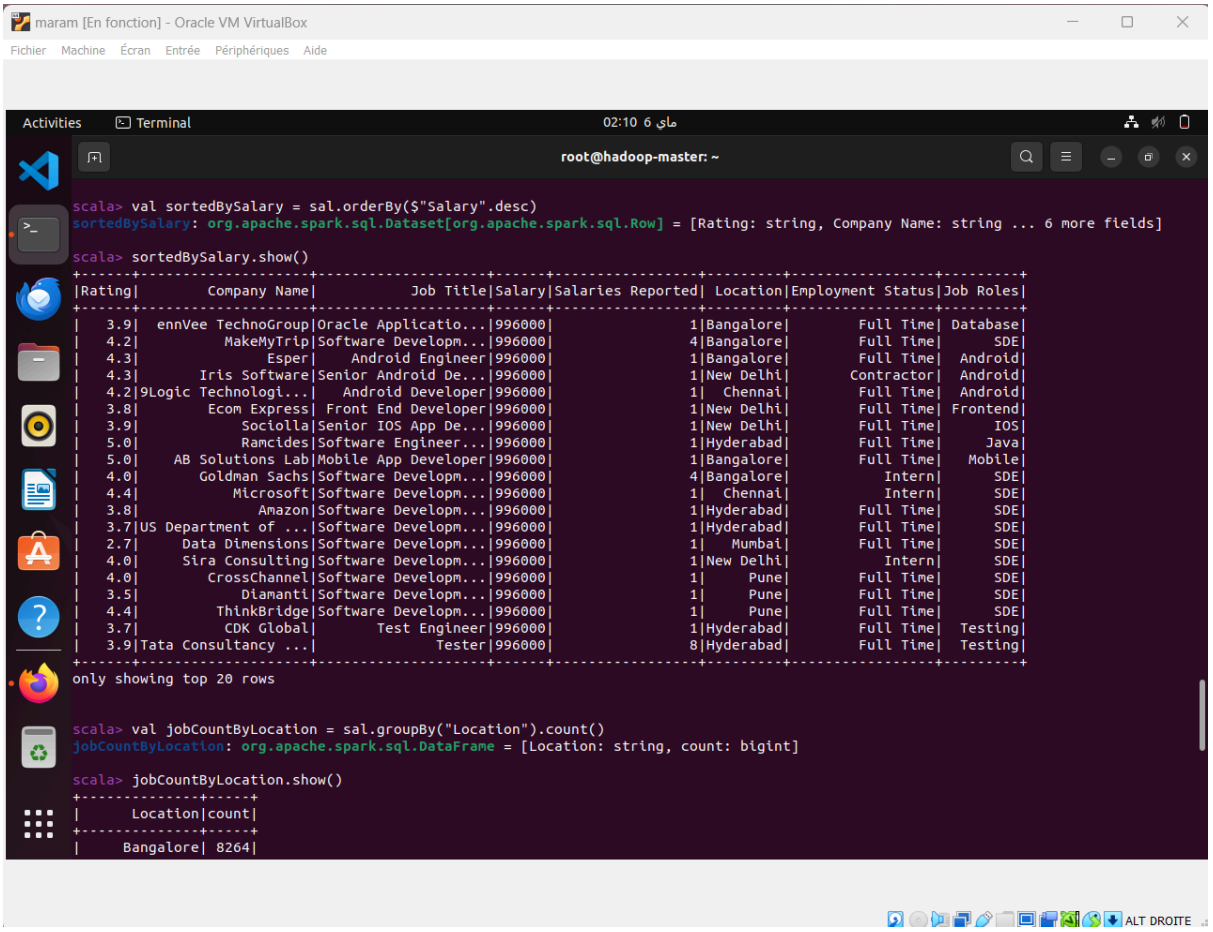


description:Cela regroupe les données par entreprise et calcule la moyenne du salaire pour chaque entreprise, puis affiche le résultat.

7. Trier le DataFrame par salaire décroissant :

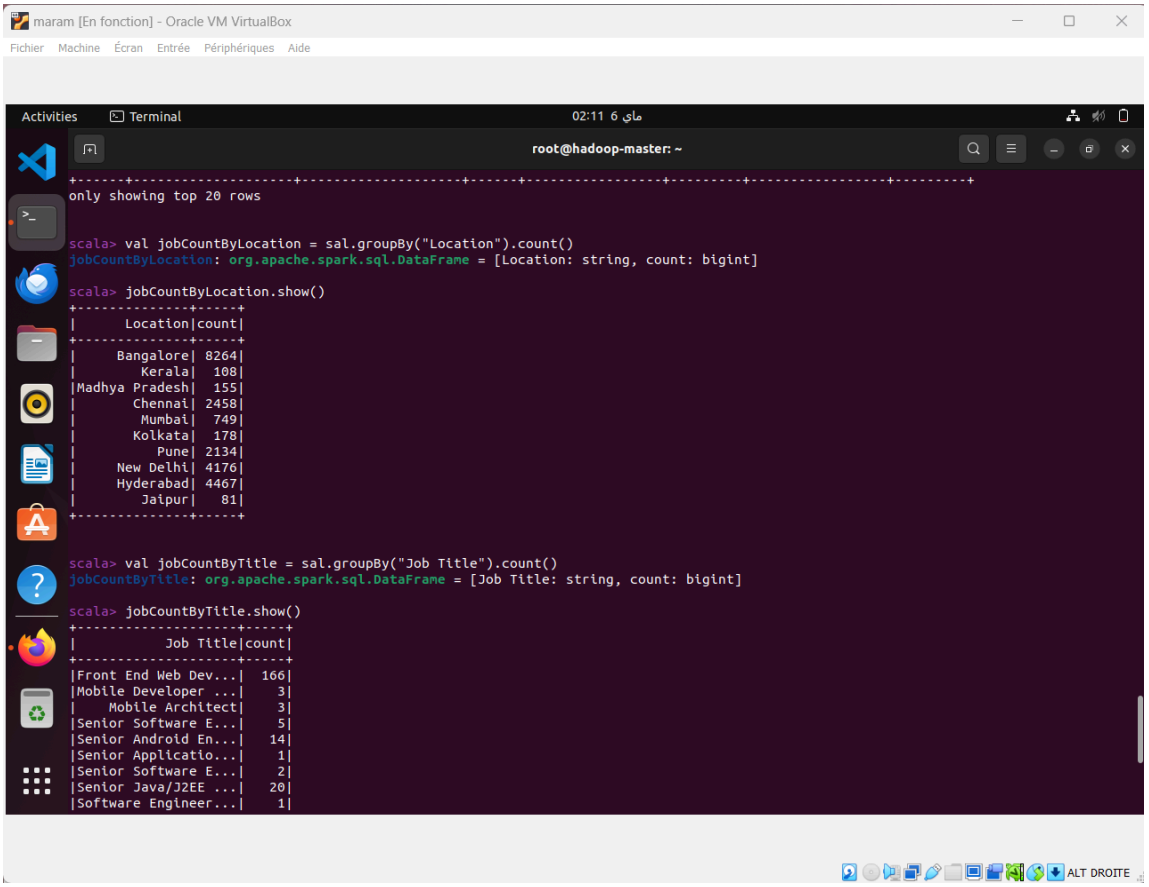
val sortedBySalary = sal.orderBy($"Salary".desc)
sortedBySalary.show()

description:Cela trie le DataFrame en fonction de la colonne "Salary" de manière décroissante et affiche le résultat.



8. Compter le nombre d'emplois par emplacement :

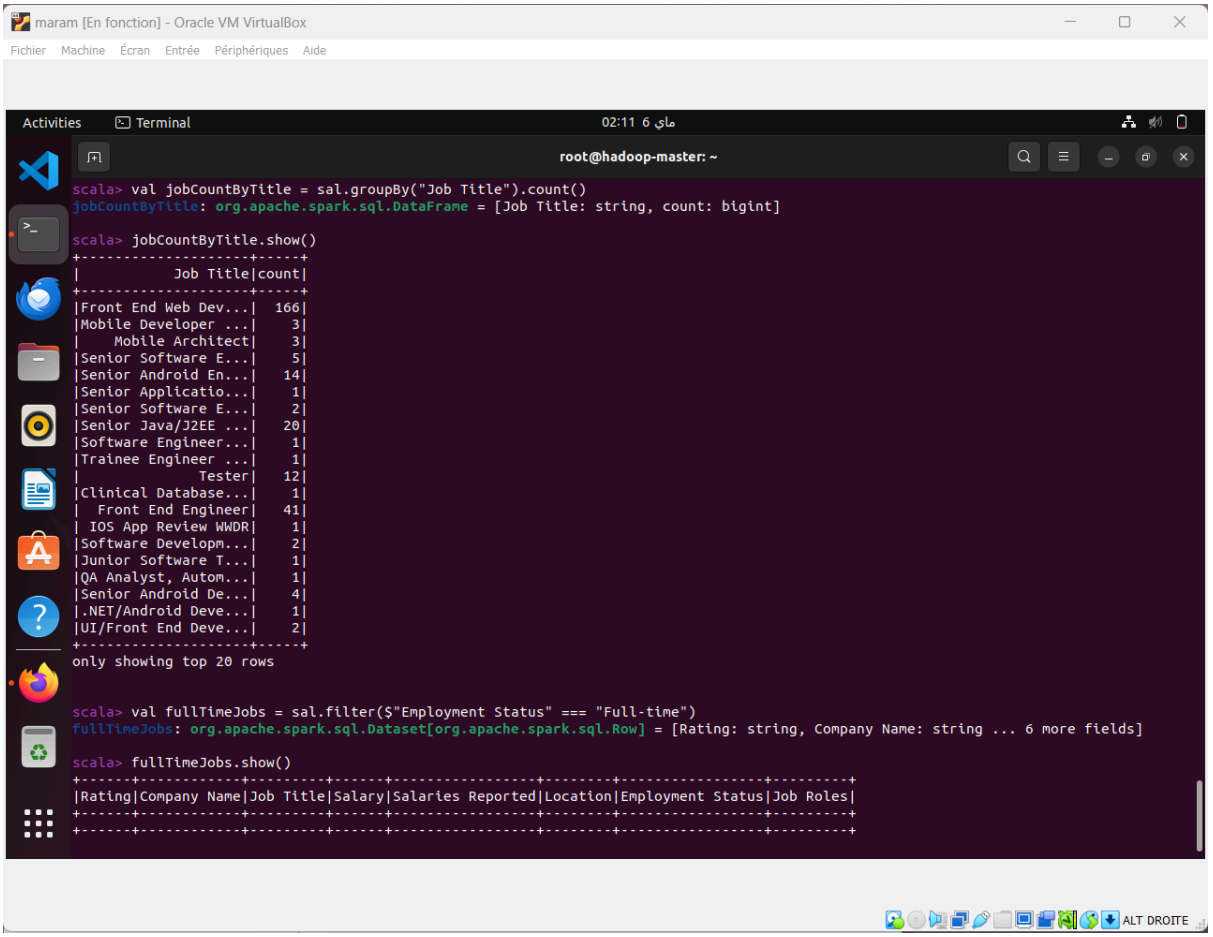
val jobCountByLocation = sal.groupBy("Location").count()
jobCountByLocation.show()
description:Cela regroupe les données par emplacement et compte le nombre d'emplois pour chaque emplacement, puis affiche le résultat.



9. Compter le nombre d'emplois par titre de poste :

val jobCountByTitle = sal.groupBy("Job Title").count()
jobCountByTitle.show()

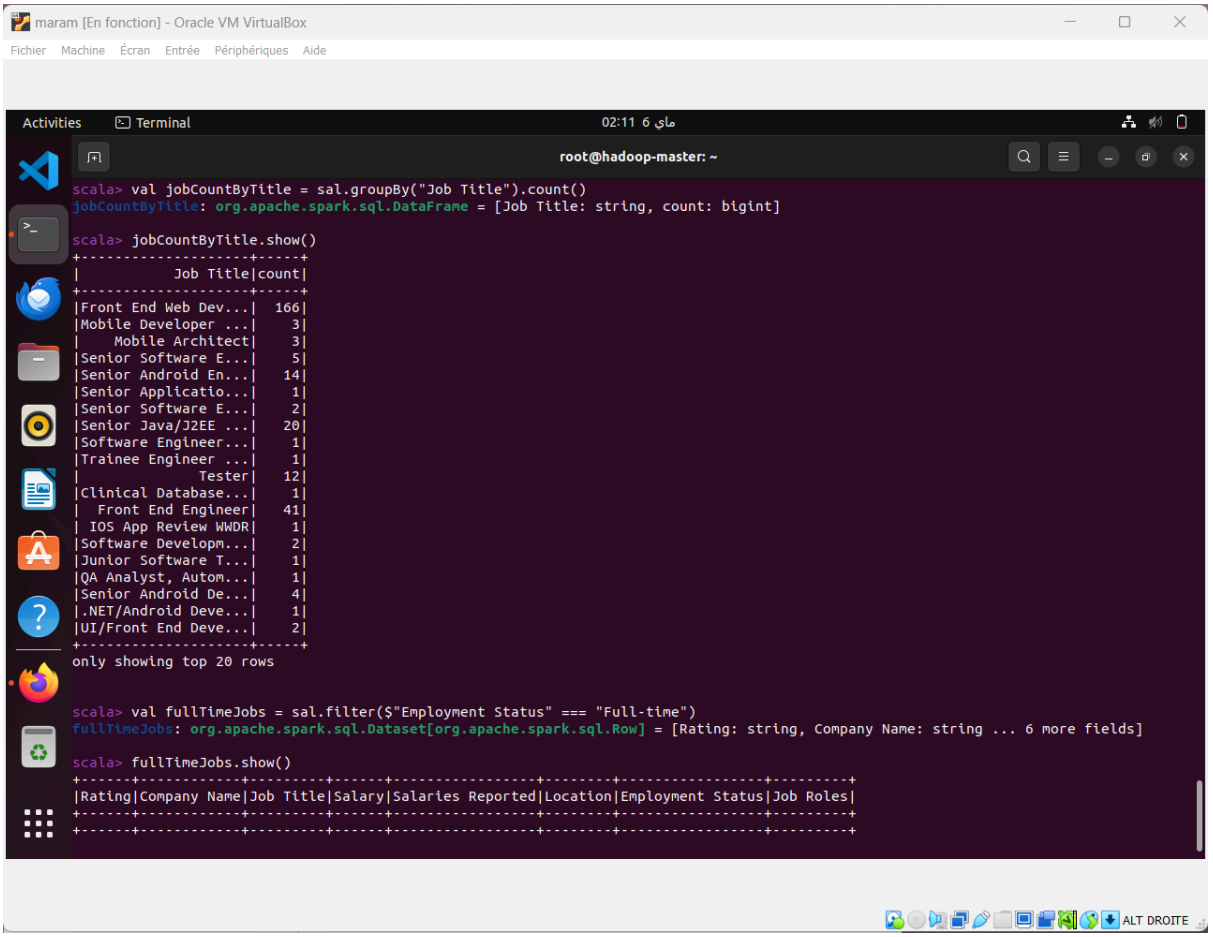
description:Cela regroupe les données par titre de poste et compte le nombre d'emplois pour chaque titre de poste, puis affiche le résultat.



10. Filtrer les emplois à temps plein :

val fullTimeJobs = sal.filter($"Employment Status" === "Full-time")
fullTimeJobs.show()

description:Cela filtre les emplois à temps plein en sélectionnant les lignes où la valeur de la colonne "Employment Status" est "Full-time", puis affiche le résultat.

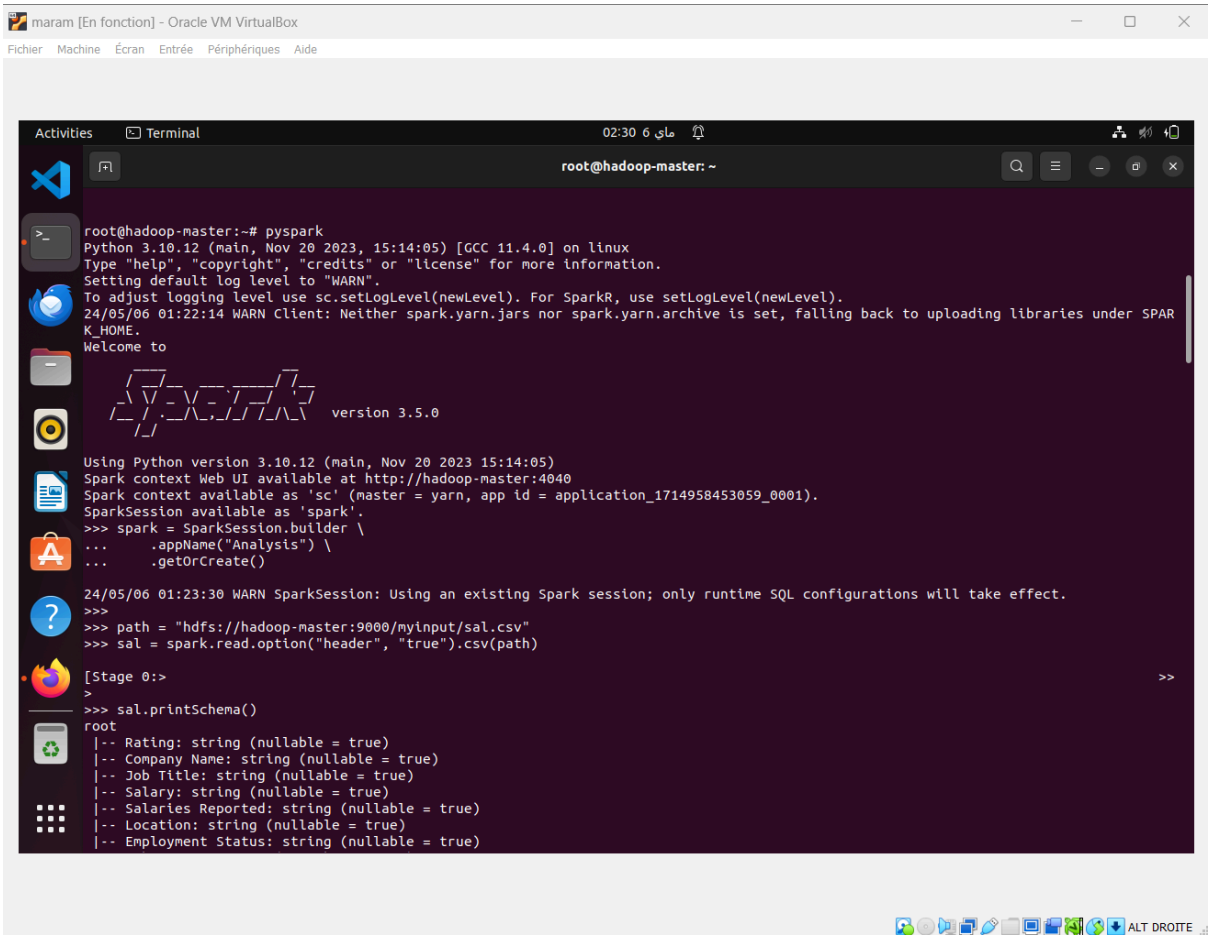


1/ code pyspark:

1. Créer une session Spark et charger le fichier CSV :

```
from pyspark.sql import SparkSession
spark = SparkSession.builder \
    .appName("Analysis") \
    .getOrCreate()
path = "hdfs://hadoop-master:9000/myinput/sal.csv"
sal = spark.read.option("header", "true").csv(path)
```

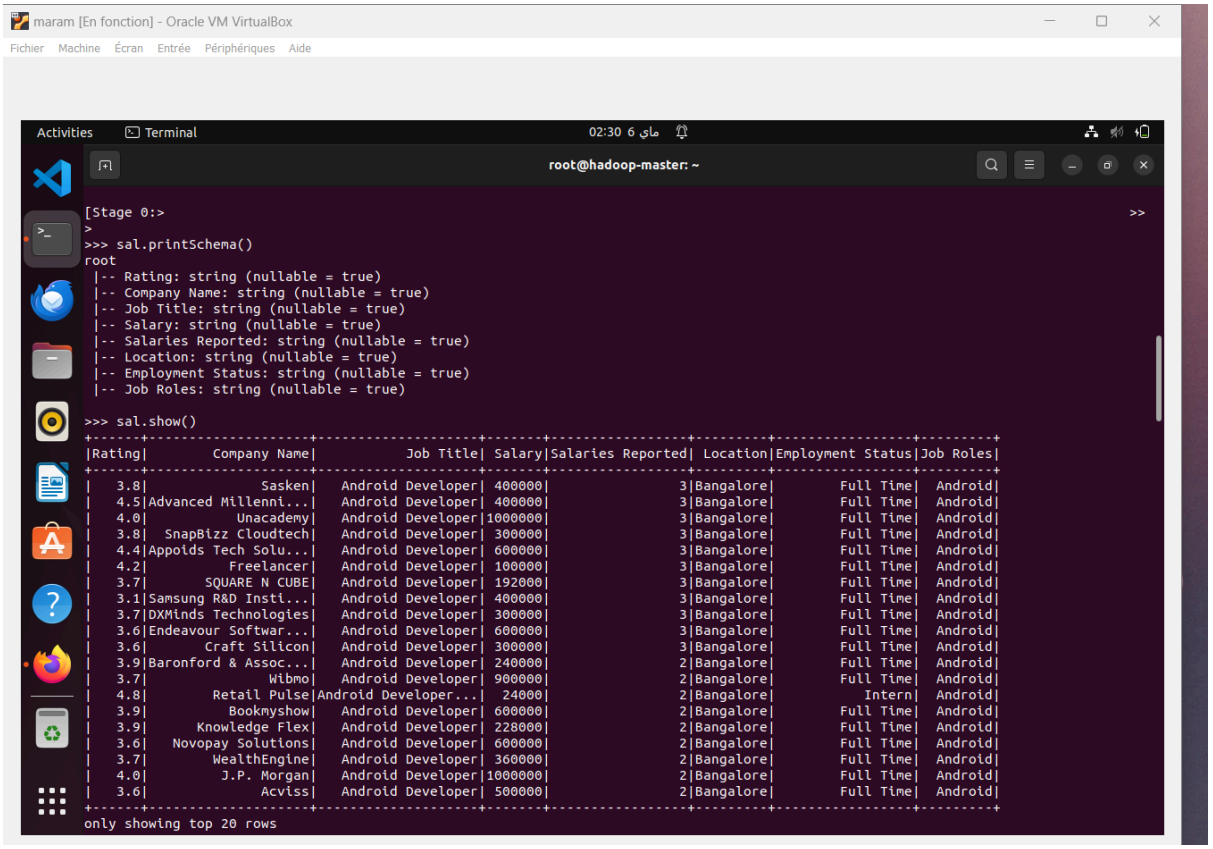
`description` :Cela crée une session Spark et charge le fichier CSV depuis HDFS dans un DataFrame Spark appelé `sal`.



2. Afficher le schéma des données :

```
sal.printSchema()
```

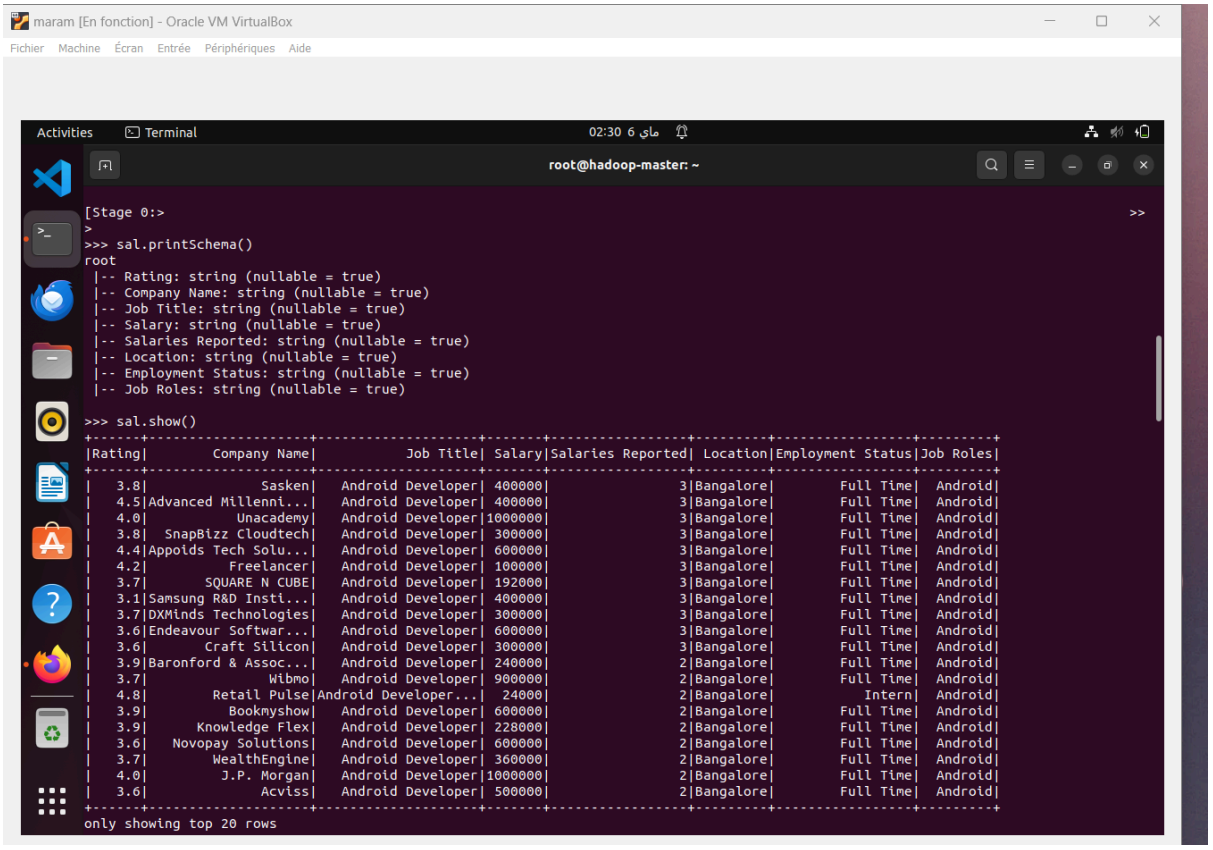
`description`:Cette commande affiche le schéma des données du DataFrame `sal`, montrant les noms des colonnes et leurs types.



3. Afficher les premières lignes du DataFrame :

```
sal.show()
```

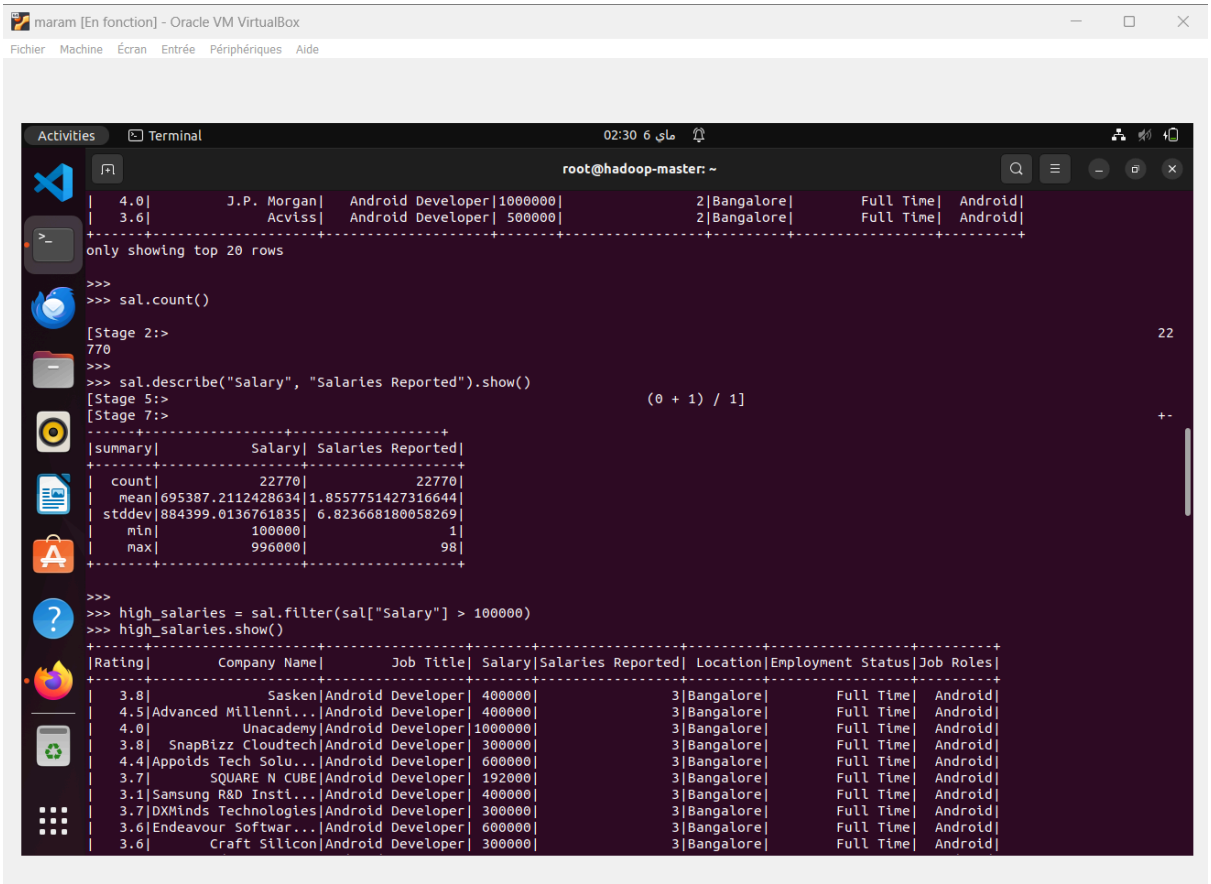
`description`:Cela affiche les premières lignes du DataFrame `sal`.



4. Compter le nombre total de lignes dans le DataFrame :

```
sal.count()
```

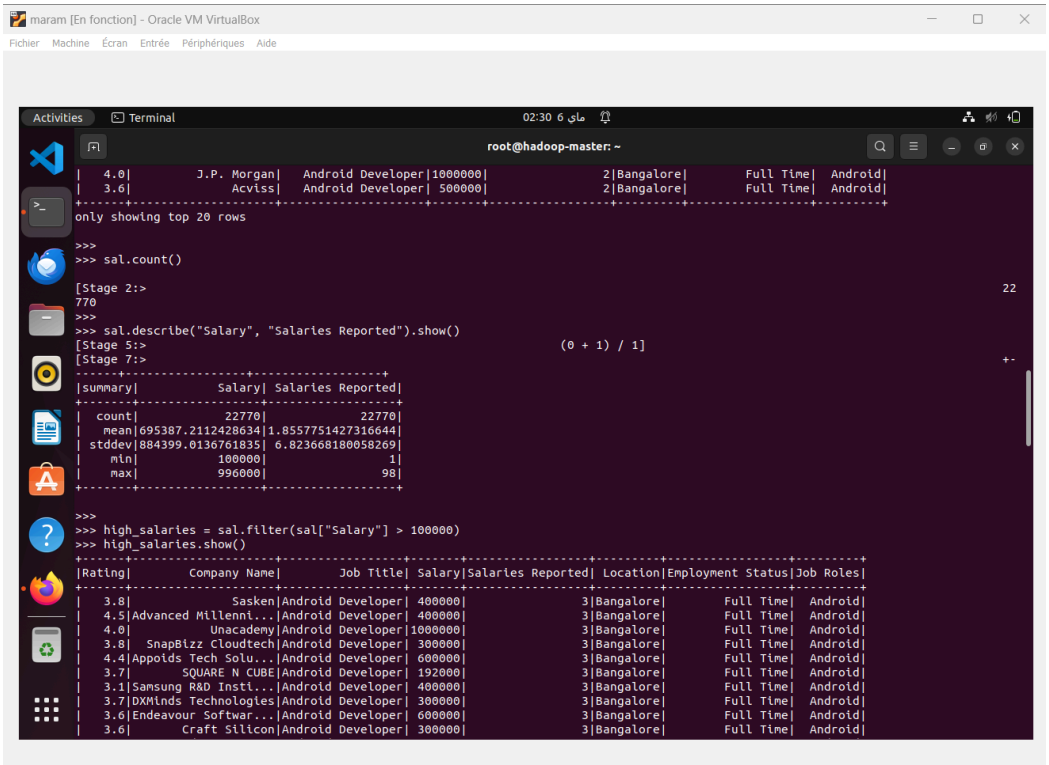
`description`:Cette commande compte le nombre total de lignes dans le DataFrame `sal`.



5. Afficher des statistiques descriptives pour les colonnes "Salary" et "Salaries Reported" :

```
sal.describe("Salary", "Salaries Reported").show()
```

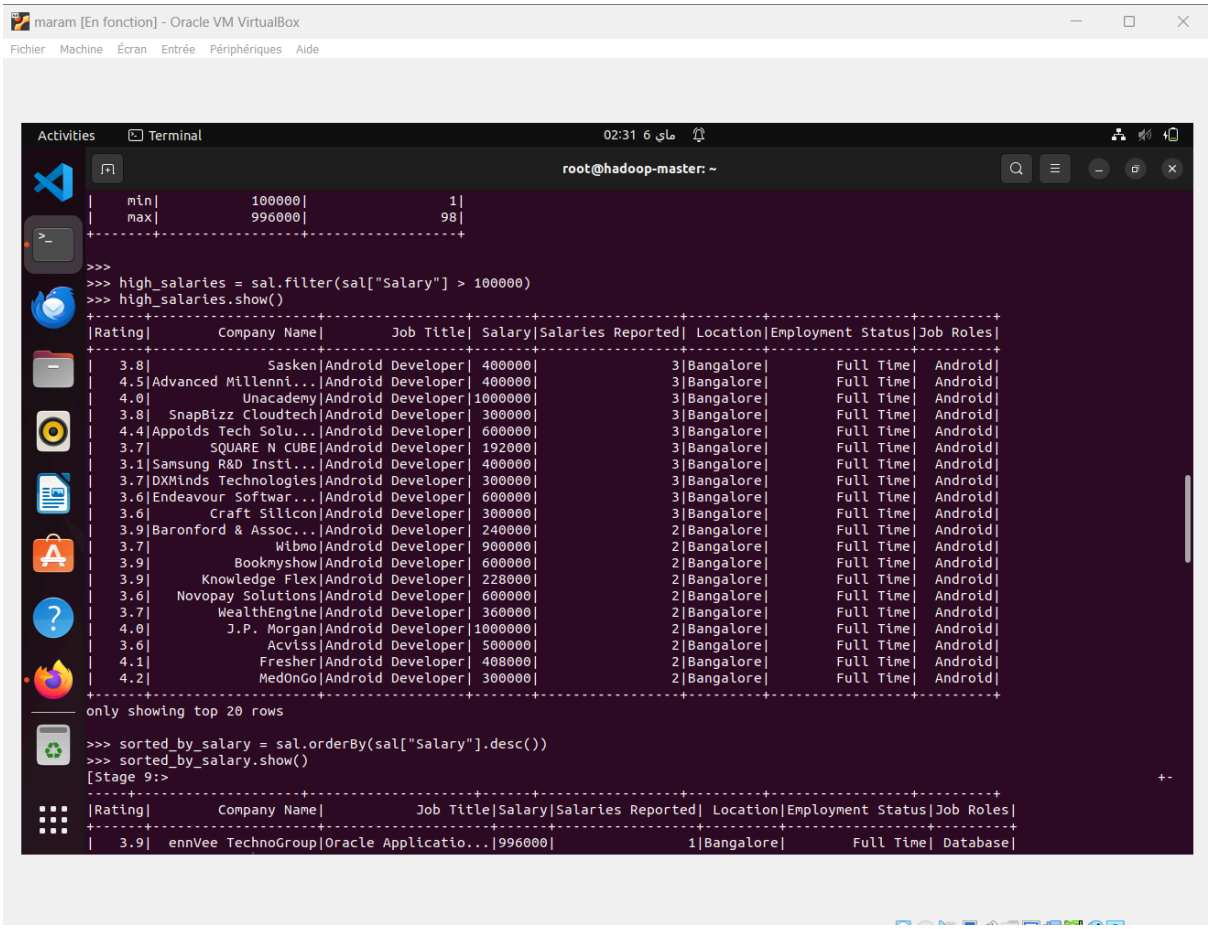
`description` : **Cela affiche des statistiques descriptives telles que la moyenne, l'écart type, le minimum, le maximum, etc., pour les colonnes spécifiées.**



6. Filtrer les lignes avec un salaire supérieur à 100 000 :

```
high_salaries = sal.filter(sal["Salary"] > 100000)
high_salaries.show()
```

`description`:**Cela filtre les lignes du DataFrame où la valeur de la colonne "Salary" est supérieure à 100 000 et affiche le résultat.**



7. Calculer le salaire moyen par entreprise :

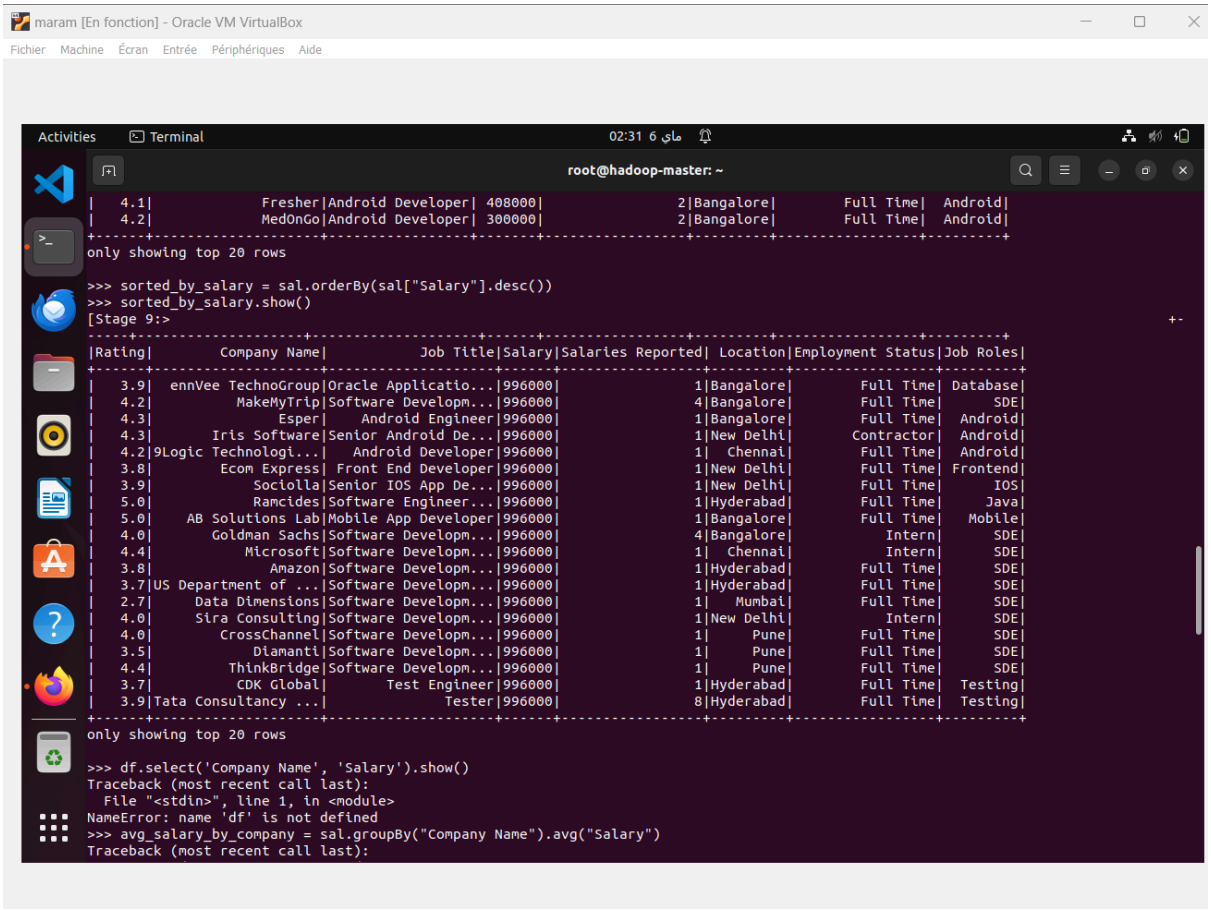
```
avg_salary_by_company = sal.groupBy("Company Name").avg("Salary")
avg_salary_by_company.show()
```

`description`:**Cela regroupe les données par entreprise et calcule la moyenne du salaire pour chaque entreprise, puis affiche le résultat.**

8. Trier le DataFrame par salaire décroissant :

```
sorted_by_salary = sal.orderBy(sal["Salary"].desc())
sorted_by_salary.show()
```

`description`:**Cela trie le DataFrame en fonction de la colonne "Salary" de manière décroissante et affiche le résultat.**



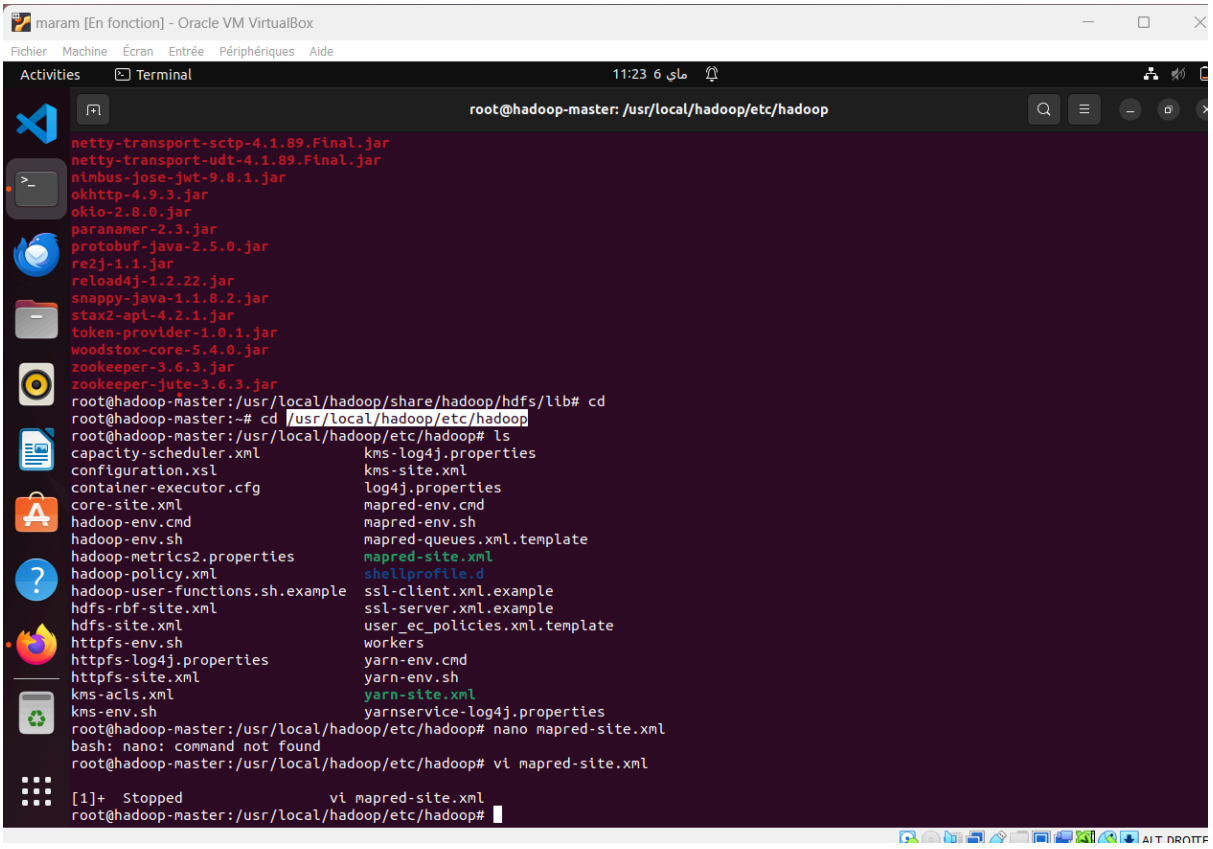
IIII-les erreurs et problèmes rencontrés:

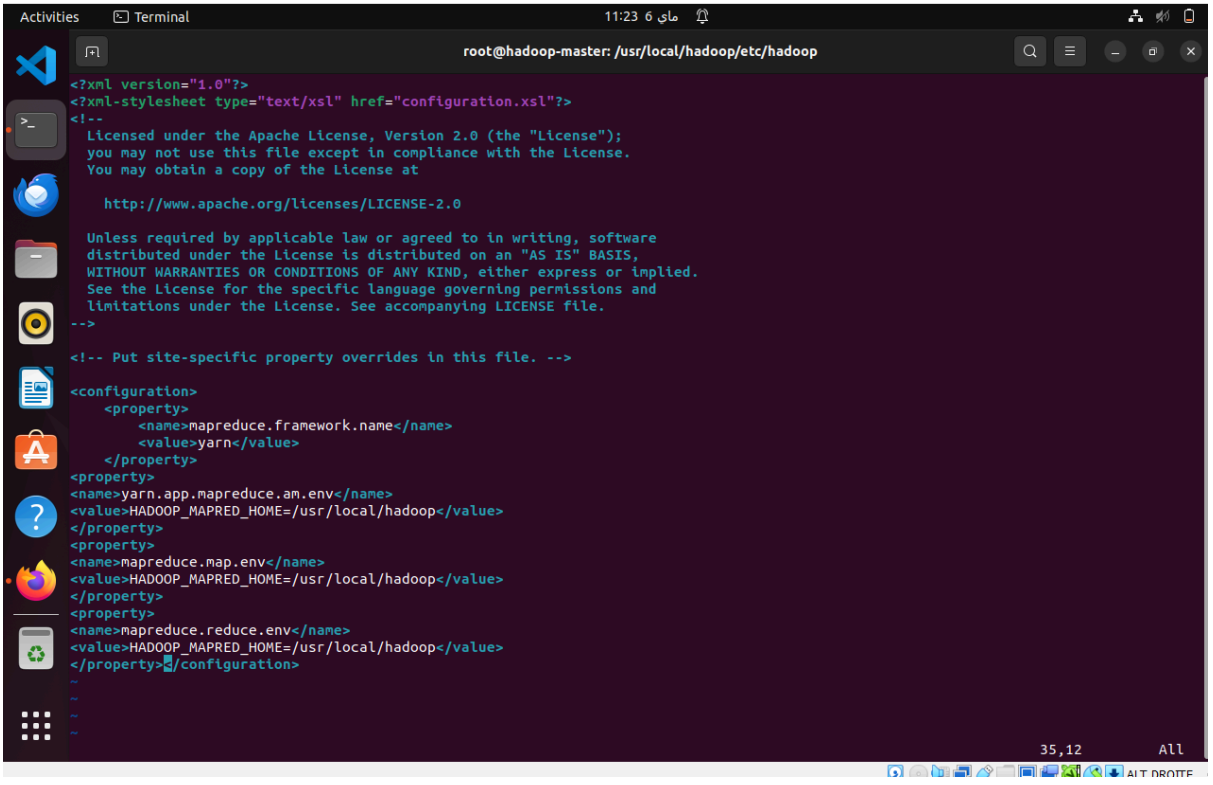
1/Lorsque vous avez télécharger l'image(image lilia dans ma projet) et vous voulez de nouveau éxécuter le root hadoop-master vous devez start tous les workers et le master en exécutant:

sudo docker start hadoop-master

sudo docker start hadoop-worker 1 …. sudo docker start hadoop-worker n

2/lorsque j'ai exécuteé la commande mapreduce j'ai besoin de corriger le fichier mapred car il était incomplet il faut éxécutez :

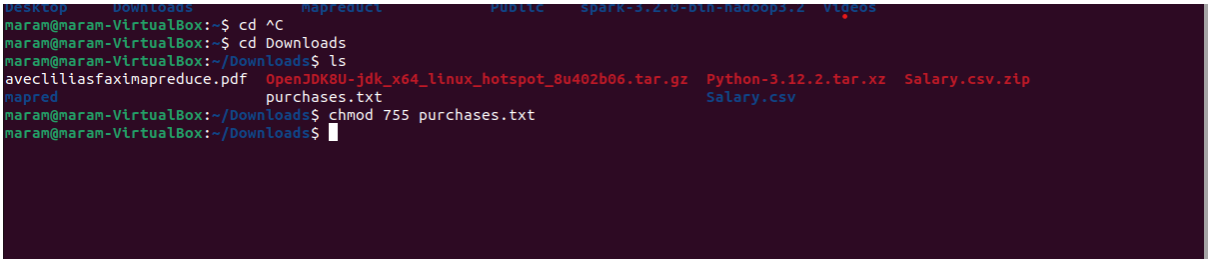




3. Erreur de connexion au réseau lors de la création du réseau Docker :
    - Problème : Vous pourriez rencontrer des erreurs lors de la création du réseau Docker en raison de problèmes de connectivité réseau.

- Solution : Assurez-vous d'avoir une connexion réseau active sur votre machine. Vérifiez également si Docker est autorisé à accéder au réseau en vérifiant les paramètres de votre pare-feu ou de votre proxy.

4. Erreur lors de la copie de fichiers vers le conteneur Hadoop :
    - Problème : Vous pourriez rencontrer des erreurs lors de la copie de fichiers vers le conteneur Hadoop en raison de permissions insuffisantes ou de chemins de fichiers incorrects.
    - Solution : Assurez-vous d'avoir les permissions nécessaires pour copier les fichiers vers le conteneur Docker. Utilisez la commande `sudo` si nécessaire. Vérifiez également que le chemin de fichier spécifié est correct.



5/lors de l'éxécution du mapreduce vous pouvez rencontrer l'erreur de safemode vous devez le désactiver en éxécutant :
hdfs dfsadmin -safemode leave