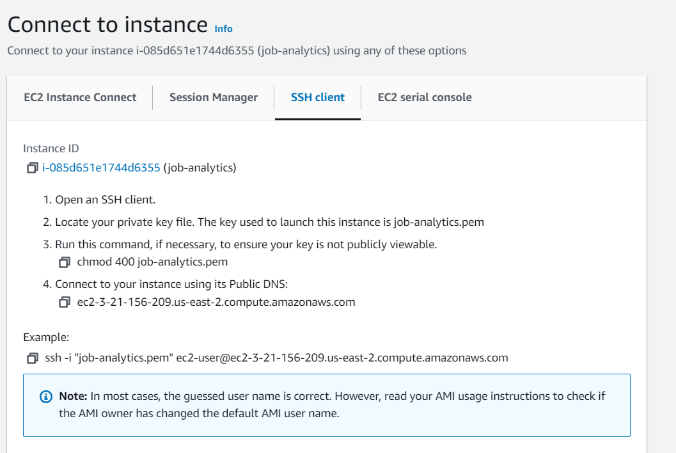
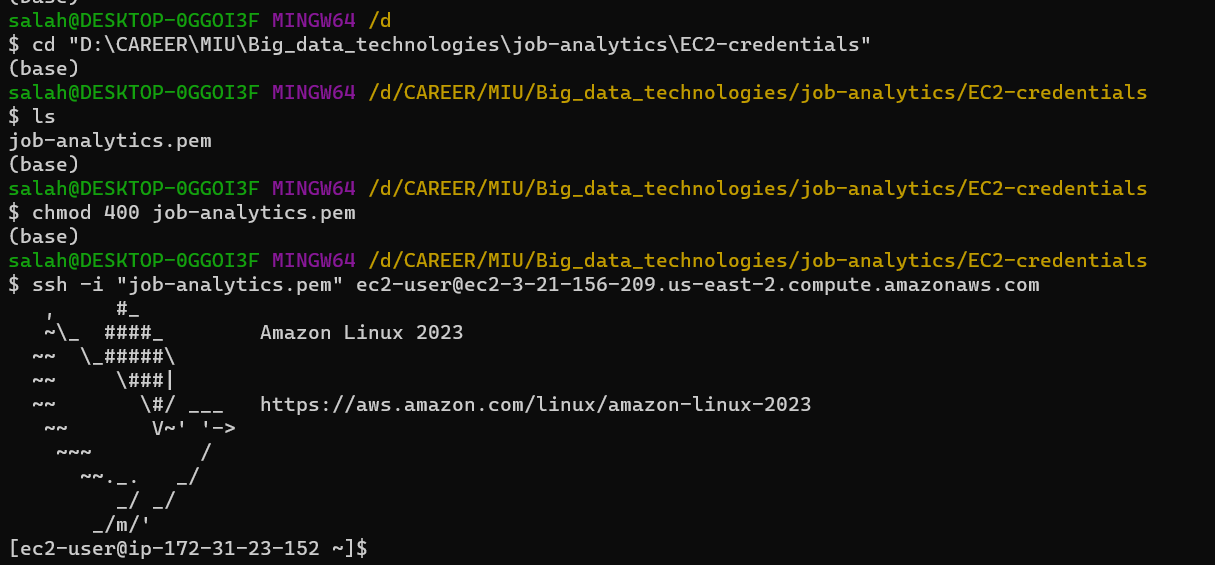
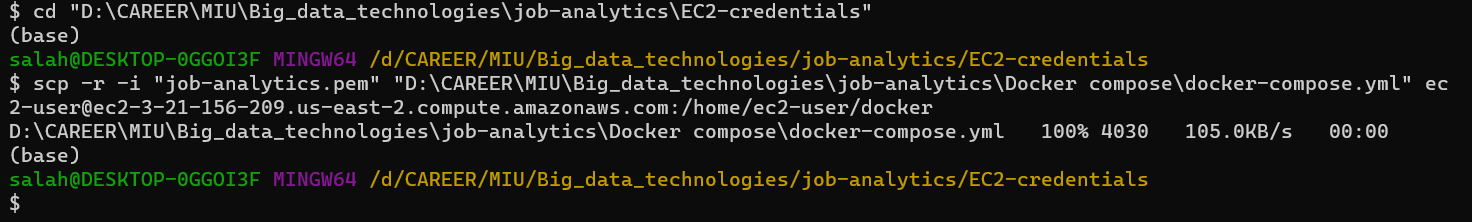
# Create X-large EC2 instance

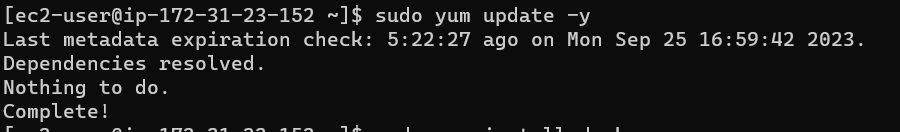
# Here is how to connect to EC2



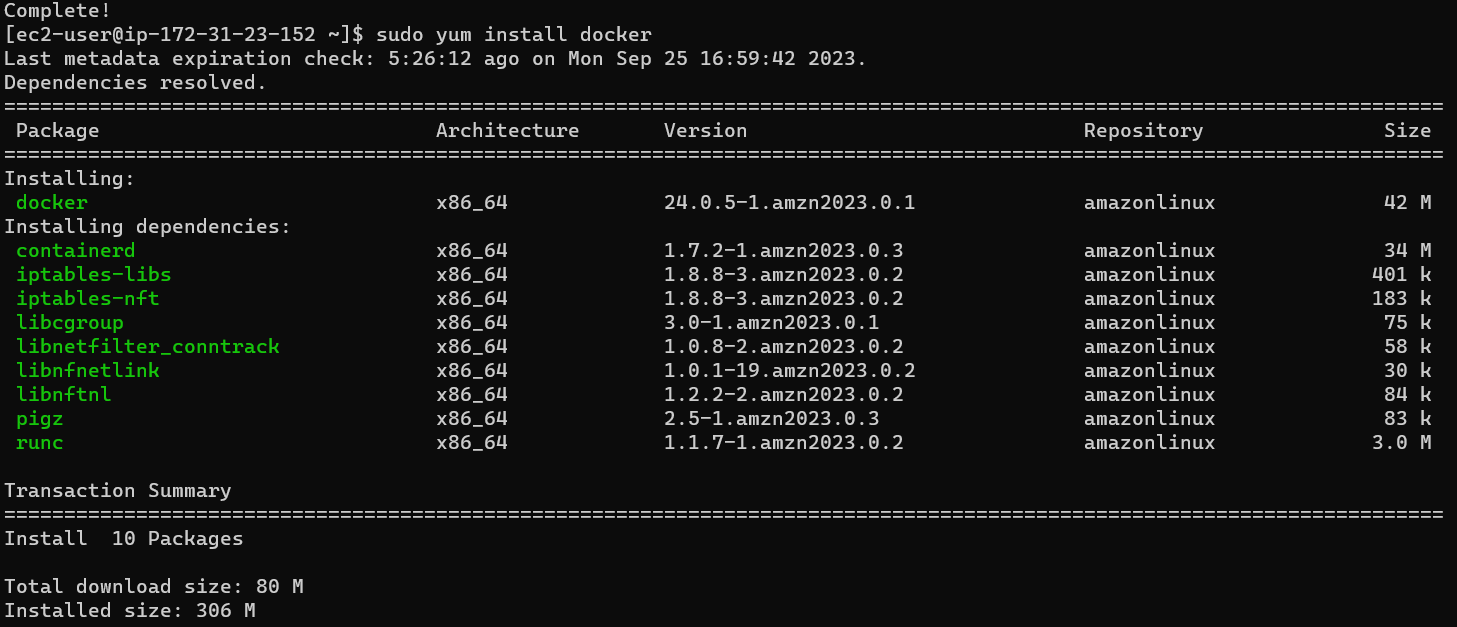


# Copy docker file to EC2

 updates all installed packages and their dependencies on a Linux system using the Yum package manager



# Install docker



# Install docker compose

Curl: allows you to download a file over Https. -L : allows you to give the link

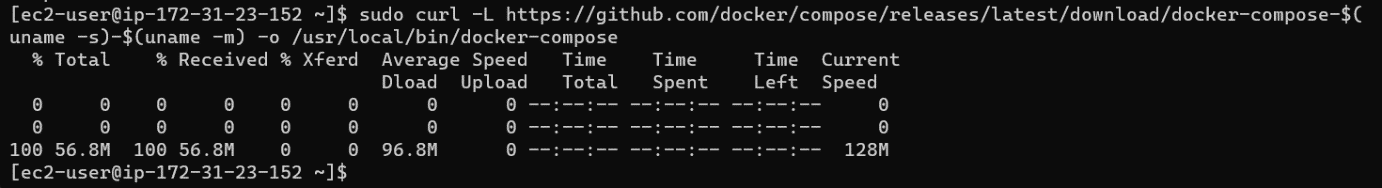
The $(uname -s) and $(uname -m) parts are command substitutions that allow the script to dynamically determine the OS and architecture to download the right Docker Compose binary.

Specifically:

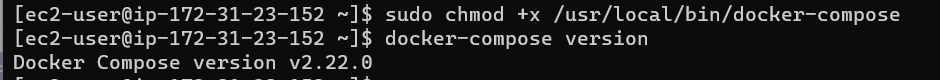
$(uname -s) - uname -s prints the name of the operating system. So on Linux this would print "Linux", on MacOS it would print "Darwin". This allows picking the right OS binary.

$(uname -m) - uname -m prints the system architecture. For example "x86\_64" on 64-bit Intel/AMD systems. This allows picking the right architecture binary.

-o: allows you to specify where you want to store the file you downloaded



# Setting the execute permission on the docker-compose



# Use docker compose up to pull images and run containers

# Then go to the shell of kafka to create a topic

You first need to use kafka’s bash. In docker here is the command:

Docker exec -i -t kafka bash

Then you create the topic with the command:

Kafka-topic.sh --create --topic your-topic-name –partitions 1 –replication-factor 1 –if-not-exists –zookeeper zookeeper:2181

A black screen with white text

Description automatically generated

When you set your producer and send messages. If you want to check if the messages are indeed sent you can use the command:  
kafka-console-consumer.sh --topic your\_topic\_name --bootstrap-server localhost:9092 --from-beginning

# Create Cassandra keyspace and table

Go to Cassandra terminal using :

docker exec -i -t cassandra bash

interact with Cassandra cluster using CQL by typing command : (username and password are Cassandra by default)

cqlsh -u myusername -p mypassword

create keyspace:

CREATE KEYSPACE IF NOT EXISTS jobs\_analytics WITH replication = {'class': 'SimpleStrategy', 'replication\_factor': 1}

CREATE TABLE IF NOT EXIST jobs\_analytics.job

A screenshot of a computer screen

Description automatically generated

Create a table:

CREATE TABLE IF NOT EXISTS jobs\_analytics.job (title Text, company\_name Text, job\_date Date, job\_link Text, job\_location Text, job\_seniority\_level Text, job\_employment\_type Text, job\_function Text, job\_industries Text, number\_applicants int, job\_description Text, Primary key ((job\_date, job\_seniority\_level, job\_location), company\_name, title));A computer screen with white text

Description automatically generated

# Create a folder in hdfs:

Docker exec -i -t namenode bash

hdfs dfs -mkdir -p output1/scraped\_jobs/

A black background with white text

Description automatically generated