Name: Deep Pawar(A20545137)
Professor: Joseph Rosen
Institute: Illinois Institute of Technology

CSP 554: Big Data Technologies

Fall 2024 - Assignment 9

- **Questions and Answers:**

Exercise 1) 5 points

Read the article "Real-time stream processing for Big Data" available on the blackboard in the 'Articles' section and then answer the following questions:

a) (1.25 points) What is the Kappa architecture and how does it differ from the lambda architecture?

Ans:

The Kappa Architecture is a real-time data processing design focused on simplicity by eliminating the batch layer, allowing all data to be processed through a single streaming system. Unlike the Lambda Architecture, which maintains distinct batch and real-time processing routes, Kappa uses the same stream processor to process new and historical data. Through the streaming layer, historical data is replayed when business logic modifications require reprocessing. This solution supports replay functionality without requiring a batch layer by utilizing scalable streaming systems, such as Kafka, to store data for extended periods.

The Lambda Architecture, on the other hand, uses both a speed layer and a batch layer to manage the volume and velocity of Big Data. While the speed layer enables real-time processing and guarantees low-latency updates, the batch layer handles massive volumes of data in periodic batches. Lambda offers real-time insights and thorough historical processing, but its dual-layer architecture makes the system more complex and necessitates independent development and maintenance for each layer. However, the single-layer architecture of the Kappa Architecture makes development and maintenance easier, which makes it appropriate for applications that value low-latency analytics over exhausting historical reprocessing.

b) (1.25 points) What are the advantages and drawbacks of pure streaming versus micro-batch real-time processing systems?

Ans:

Applications that require immediate responses, such as tracking user interactions or IoT sensor data, are best suited for **pure streaming systems**, which analyze data as it comes in and guarantee low latency. In situations where decisions must be made quickly, systems like Apache Storm and Samza offer low-latency processing which is often measured in milliseconds. However, because of messaging and other real-time demands, pure streaming may have a large per-item overhead, which reduces its resource efficiency. Pure streaming sometimes only provides at least one assurance and lacks precise fault tolerance, which means that if there are problems during the data flow, redundant processing may take place.

On the other hand, **micro-batch processing** is used by systems such as Spark Streaming balances batch and real-time processing by managing small data batches. This method improves fault tolerance and lowers development complexity for managing data consistency while increasing efficiency and enabling exactly-once processing semantics. The modest delay that micro-batching usually entails, however, makes it less appropriate for applications that require extremely low latency. Depending on the application's requirements for immediacy and system resource limitations, latency, and throughput are traded off in micro-batch against pure streaming.

c) (1.25 points) In a few sentences describe the data processing pipeline in Storm.

Ans:

Storm enables real-time stream processing by structuring the data processing pipeline into a topology, which is a directed graph in which data flows from one component to the next. Spouts, which serve as the data ingestion points and gather and release data tuples into the topology, are where the pipeline begins. Data then travels to bolts, which carry out the real processing, including data transformation, aggregation, and filtering. In order to create a chain of activities within the pipeline, bolts can also transfer processed data downstream to other bolts or write it to external storage systems.

By using various groupings to regulate the distribution of data among nodes, Storm's pipeline structure enables techniques like hash-partitioning based on particular properties or shuffle grouping. With Nimbus servers controlling job allocation throughout the cluster and ZooKeeper handling coordination, this configuration allows for scalability and robustness. Storm guarantees at-least-once processing through an acknowledgment mechanism that monitors each tuple's path through the pipeline and replays it in the event that processing fails. Storm is perfect for applications that require instant data processing because of its low-latency processing design.

d) (1.25 points) How does Spark streaming shift the Spark batch processing approach to work on real-time data streams?

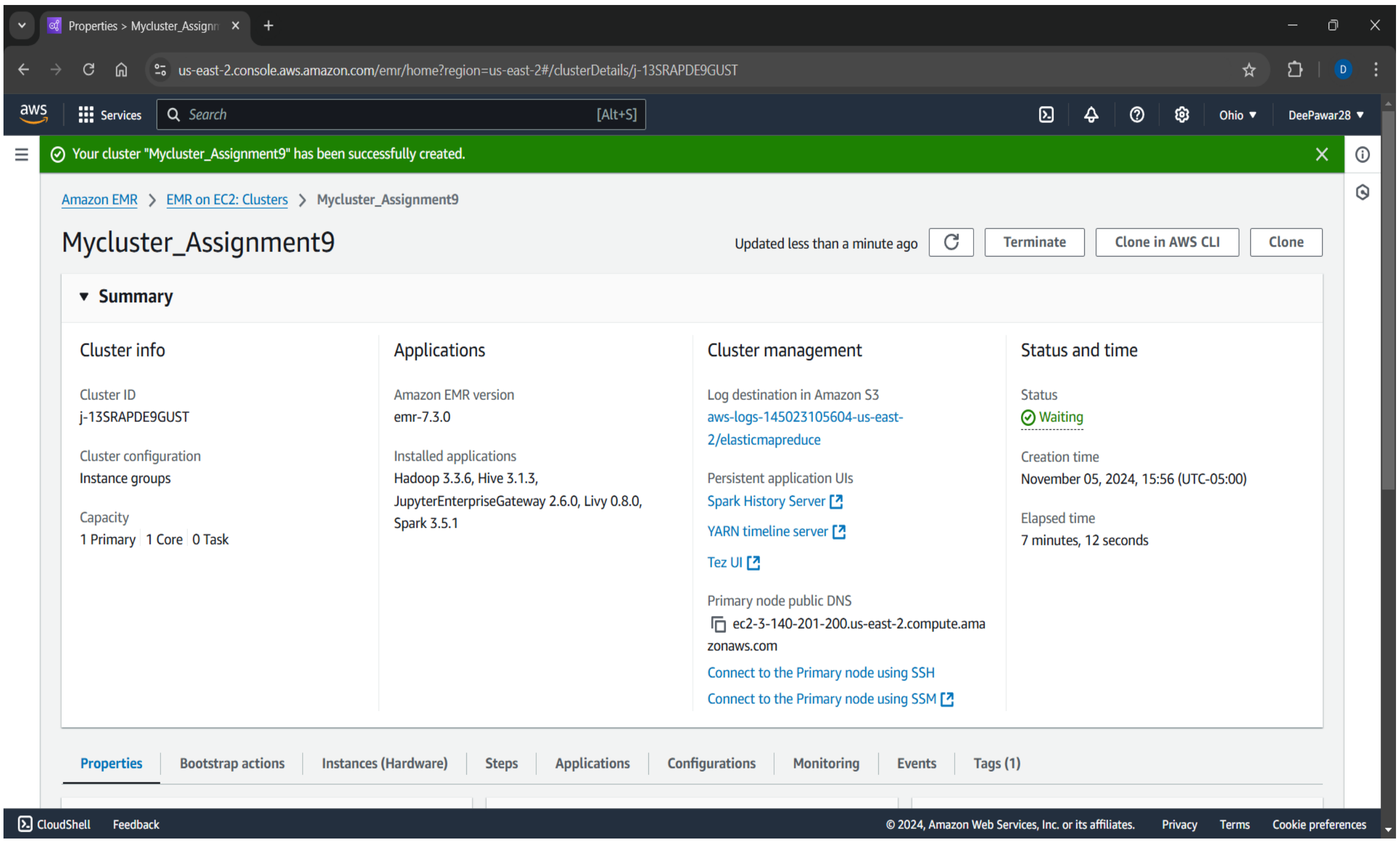
Ans:

By introducing the idea of DStreams (Discretized Streams), which splits incoming data streams into manageable chunks for processing, Spark Streaming applies the Spark batch processing methodology to real-time data. Spark Streaming transforms data into Resilient Distributed Datasets (RDDs), which Spark analyzes in parallel, by grouping data into mini batches rather than treating each data item separately. By using this method, Spark Streaming can reduce latency to support near real-time applications while maintaining Spark's natural batch-processing performance. Spark Streaming allows for continuous computation on real-time data streams by sequentially processing each mini-batch of RDDs but with a marginally higher latency than pure streaming systems.

Spark Streaming incorporates a write-ahead log (WAL) to improve fault tolerance in real-time applications. This logs incoming data for possible replay in the event of errors. Even with dubious sources, this guarantees that no data is lost. Spark Streaming is dependable for applications that need consistency because it uses checkpointing and tracking lineage to preserve exactly-once processing semantics. Spark can manage both batch and streaming data within a single, cohesive framework thanks to the mini-batch technique, which offers a compromise between the high latency of typical batch processing and the quick, per-item overhead of pure streaming systems, despite the slight delay it adds.

Exercise 2) 5 points (extra credit)

Step A – Start an EMR cluster


Properties > Mycluster_Assignm
us-east-2.console.aws.amazon.com/emr/home?region=us-east-2#/clusterDetails/j-13SRAPDE9GUST
Services
Search [Alt+S]
Ohio
DeePawar28

Your cluster "Mycluster_Assignment9" has been successfully created.

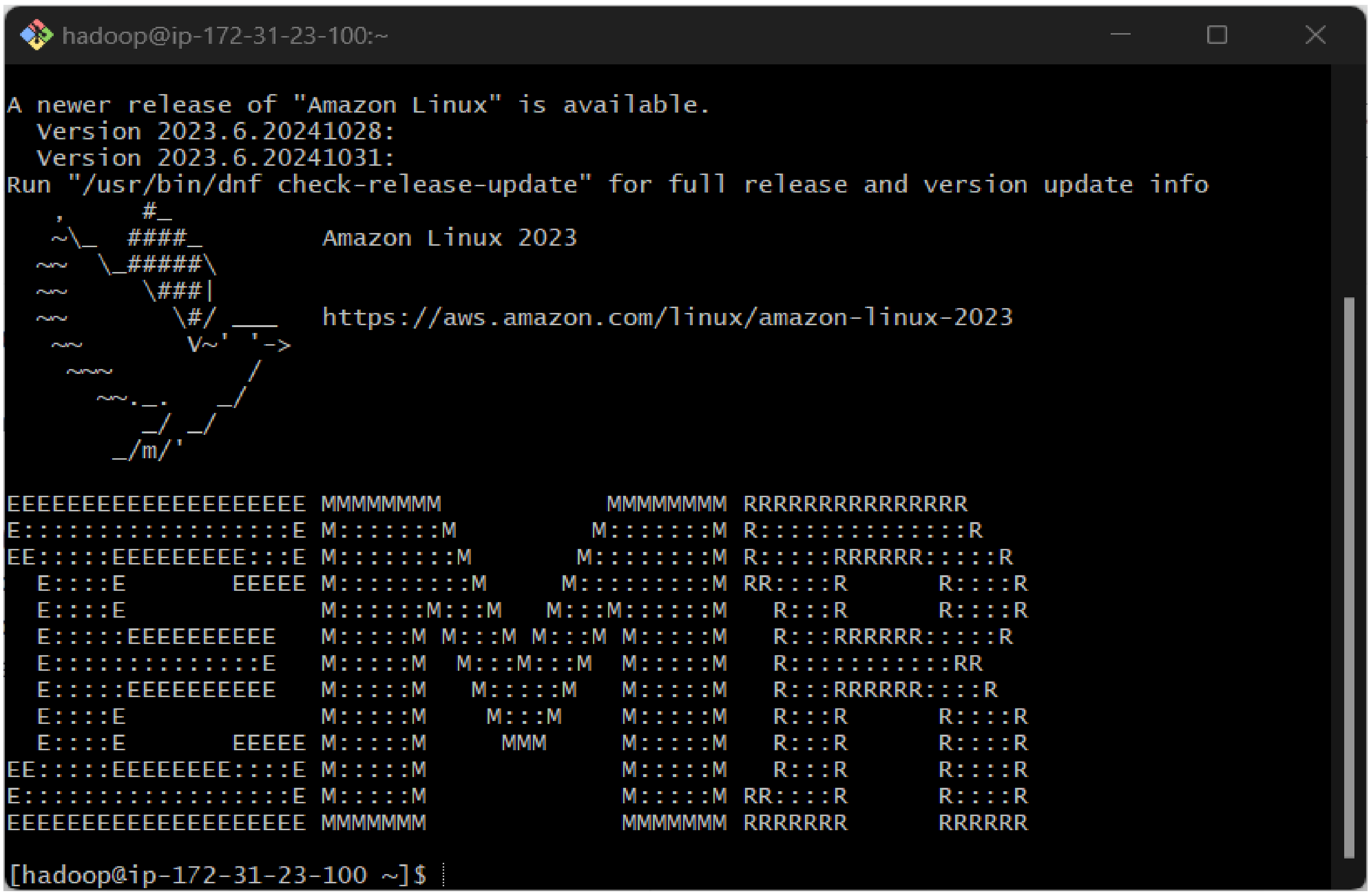
Amazon EMR > EMR on EC2: Clusters > Mycluster_Assignment9

Mycluster_Assignment9

Updated less than a minute ago | Terminate | Clone in AWS CLI | Clone

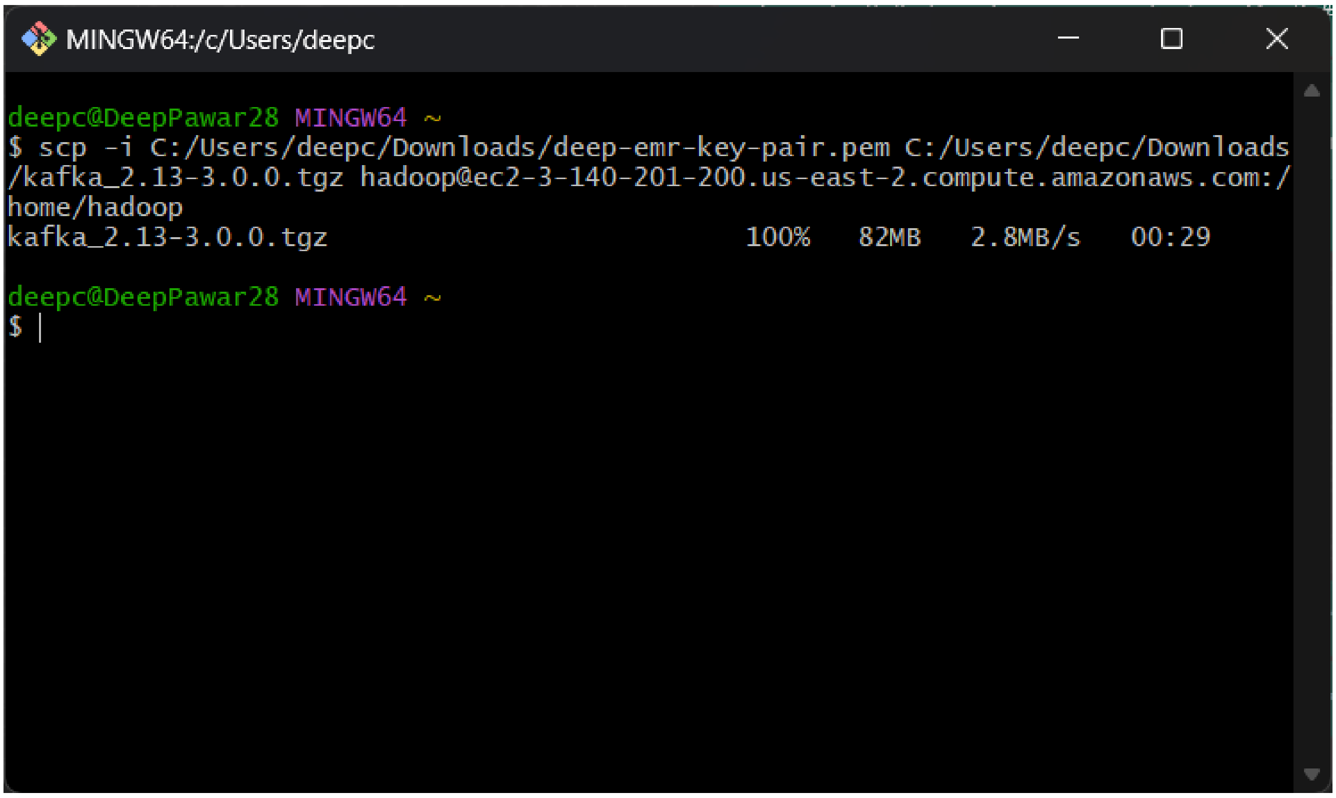
Summary

Cluster info	Applications	Cluster management	Status and time
Cluster ID: j-13SRAPDE9GUST	Amazon EMR version: emr-7.3.0	Log destination in Amazon S3: aws-logs-145023105604-us-east-2/elasticmapreduce	Status: Waiting
Cluster configuration: Instance groups	Installed applications: Hadoop 3.3.6, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5.1	Persistent application UIs: Spark History Server, YARN timeline server, Tez UI	Creation time: November 05, 2024, 15:56 (UTC-05:00)
Capacity: 1 Primary 1 Core 0 Task		Primary node public DNS: ec2-3-140-201-200.us-east-2.compute.amazonaws.com	Elapsed time: 7 minutes, 12 seconds
		Connect to the Primary node using SSH; Connect to the Primary node using SSM	



```
A newer release of "Amazon Linux" is available.
  Version 2023.6.20241028:
  Version 2023.6.20241031:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-23-100 ~]$
```

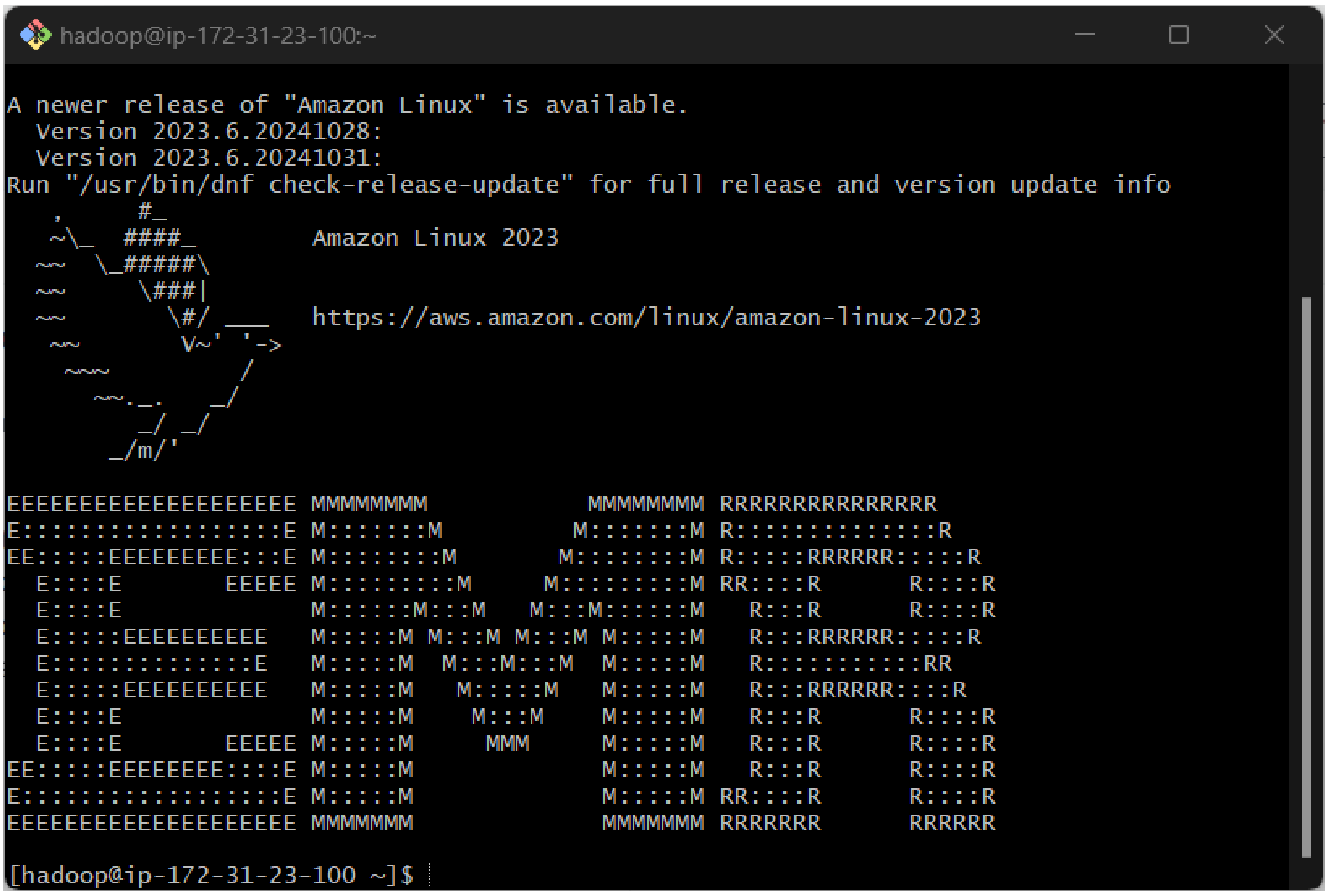
Step B – Copy the Kafka software to the EMR primary node



```
deepc@DeepPawar28 MINGW64 ~
$ scp -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem C:/Users/deepc/Downloads
/kafka_2.13-3.0.0.tgz hadoop@ec2-3-140-201-200.us-east-2.compute.amazonaws.com:/
home/hadoop
kafka_2.13-3.0.0.tgz                          100%   82MB   2.8MB/s   00:29

deepc@DeepPawar28 MINGW64 ~
$ |
```

Step C – Install the Kafka software and start it

- Kafka-Term:



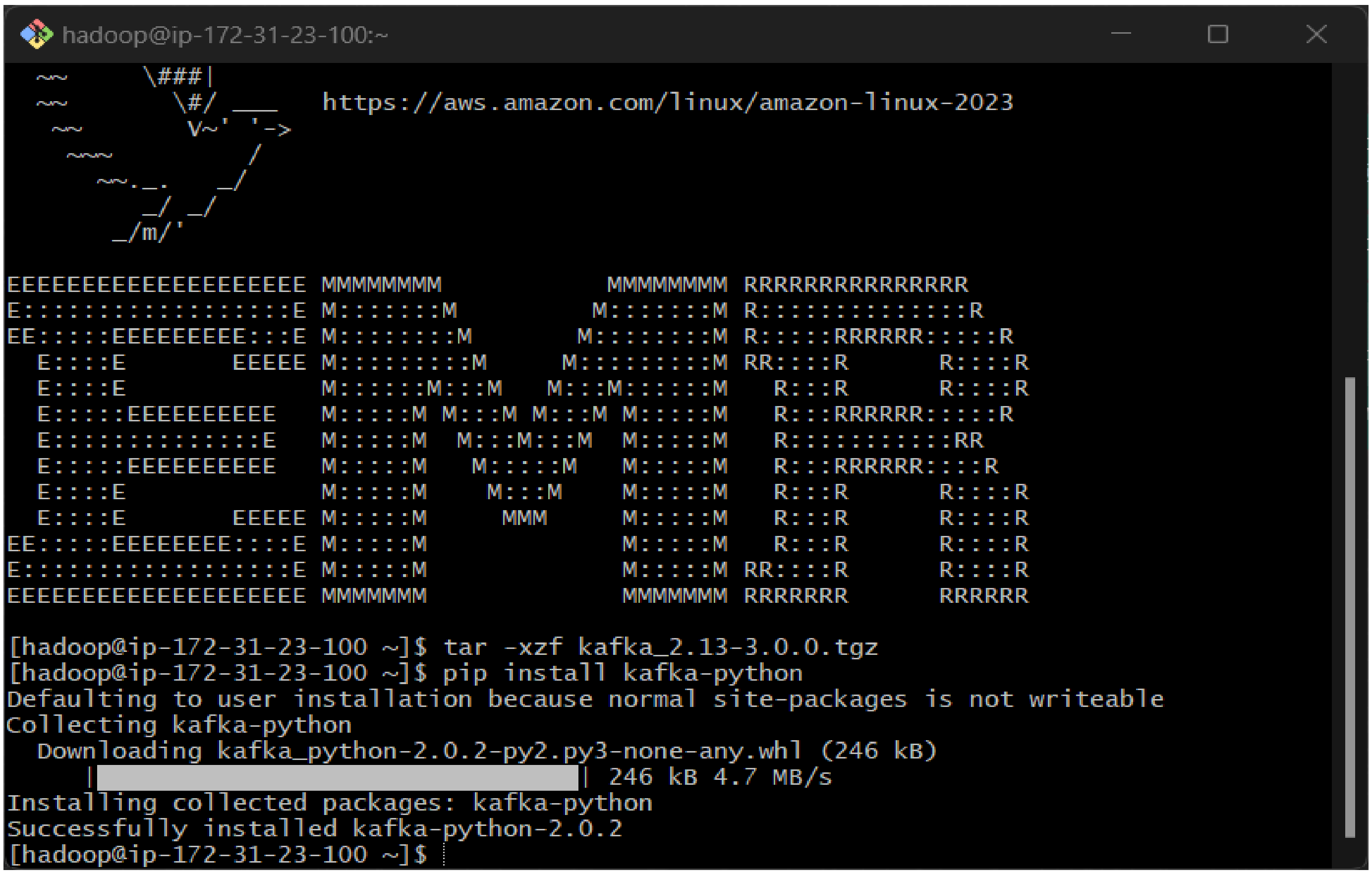
```
A newer release of "Amazon Linux" is available.
  Version 2023.6.20241028:
  Version 2023.6.20241031:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-23-100 ~]$
```

- Enter the following command to create a new directory (kafka_2.13-3.0.0) holding the kafka software release:

tar -xzf kafka_2.13-3.0.0.tgz

- Then enter the following command to install the kafka-python package:

pip install kafka-python



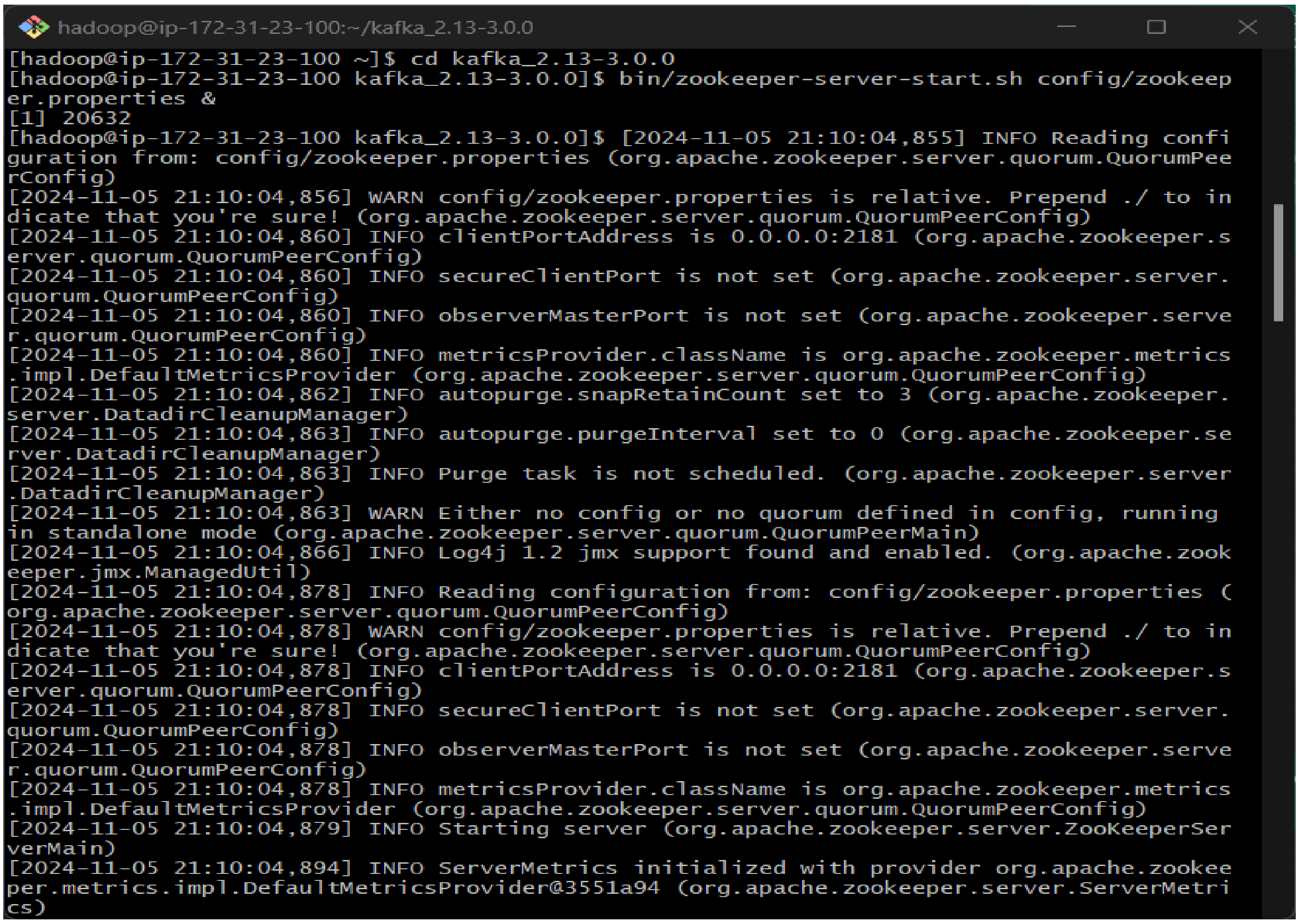
```
  ~~       \###|
  ~~        \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~        V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-23-100 ~]$ tar -xzf kafka_2.13-3.0.0.tgz
[hadoop@ip-172-31-23-100 ~]$ pip install kafka-python
Defaulting to user installation because normal site-packages is not writeable
Collecting kafka-python
  Downloading kafka_python-2.0.2-py2.py3-none-any.whl (246 kB)
     |████████████████████████████████| 246 kB 4.7 MB/s
Installing collected packages: kafka-python
Successfully installed kafka-python-2.0.2
[hadoop@ip-172-31-23-100 ~]$
```

- Now enter the following commands into the terminal to start up a zookeeper instance and the kafka server:

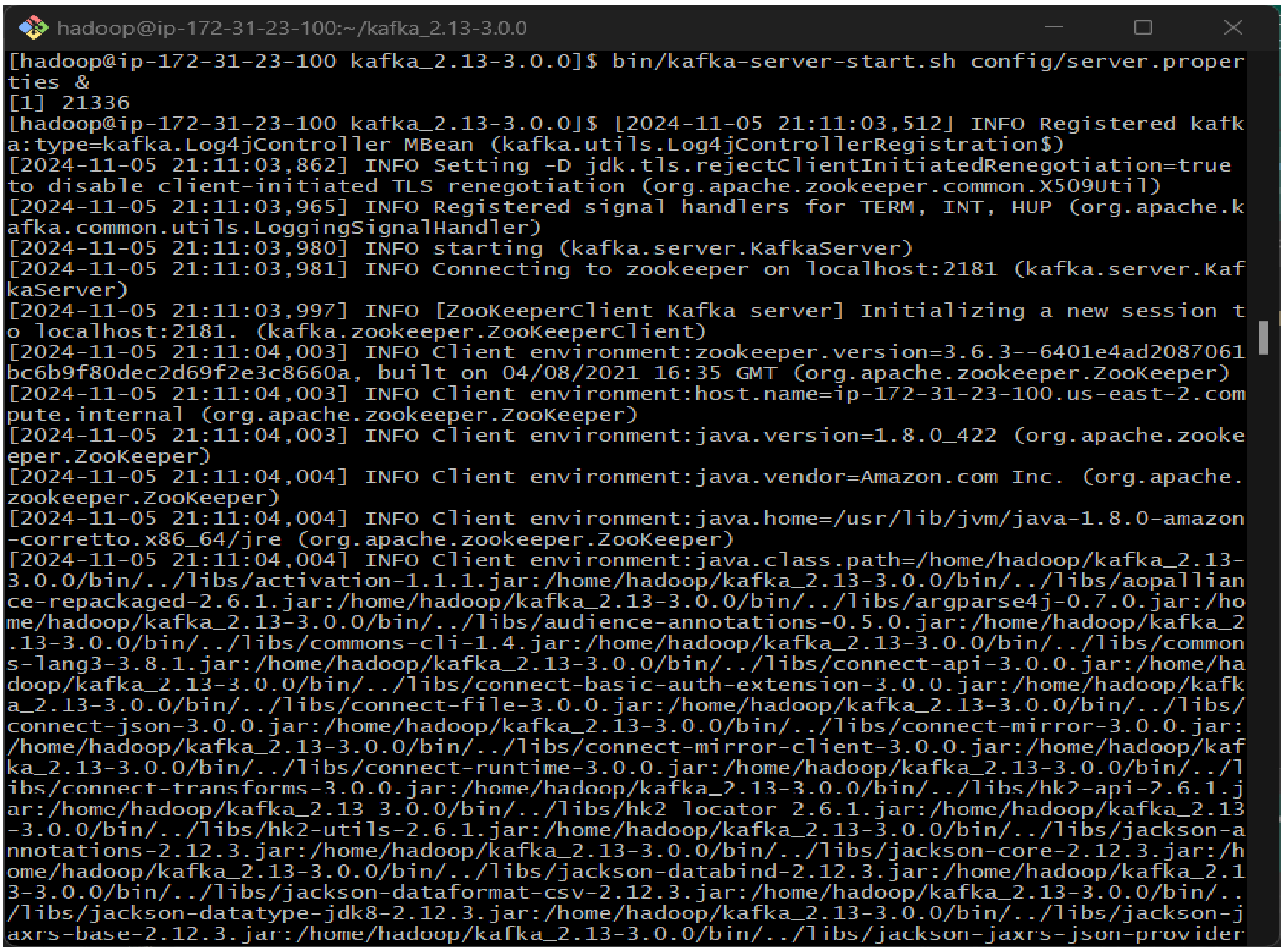
cd kafka_2.13-3.0.0

bin/zookeeper-server-start.sh config/zookeeper.properties &

bin/kafka-server-start.sh config/server.properties &



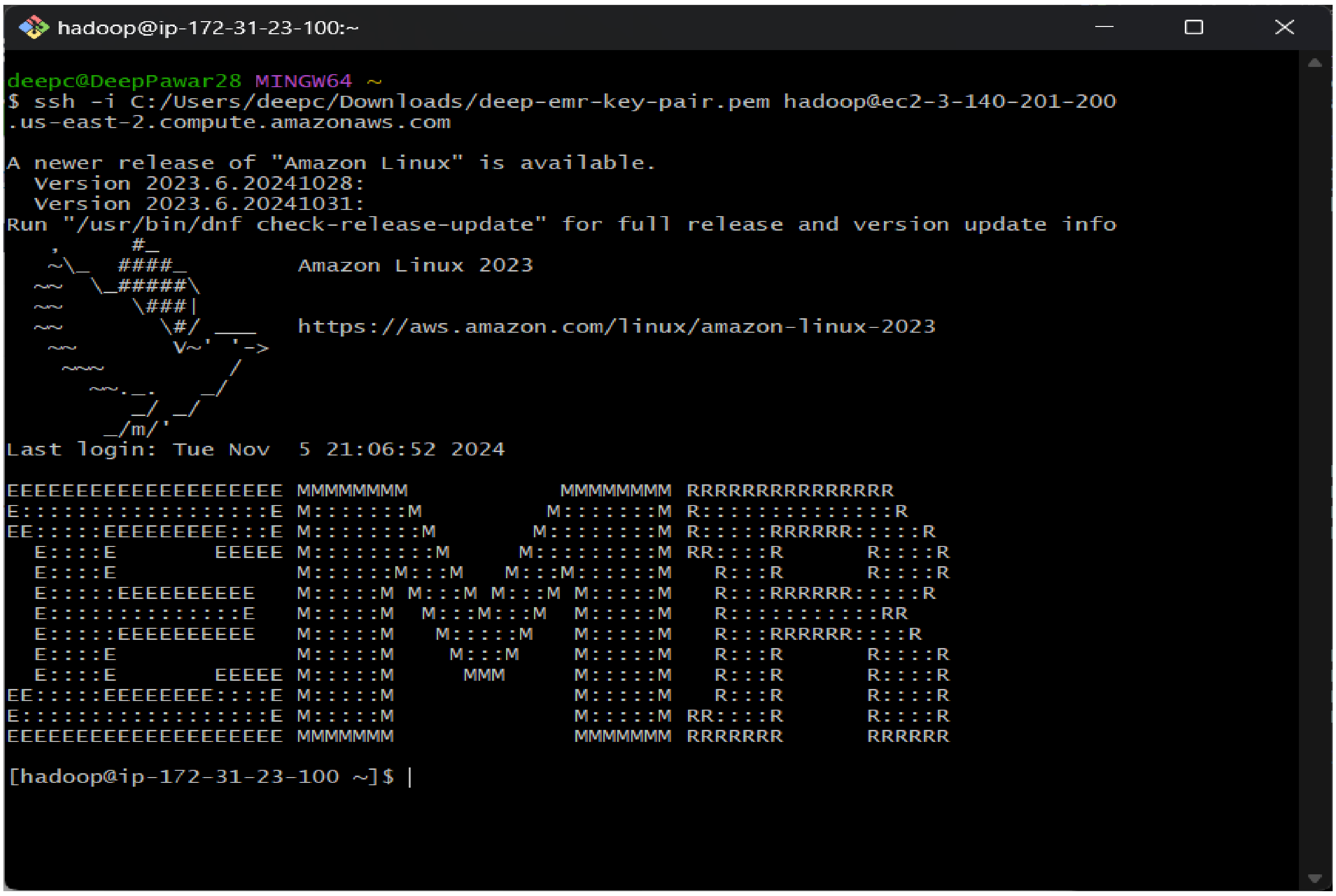
```
[hadoop@ip-172-31-23-100 ~]$ cd kafka_2.13-3.0.0
[hadoop@ip-172-31-23-100 kafka_2.13-3.0.0]$ bin/zookeeper-server-start.sh config/zookeep
er.properties &
[1] 20632
[hadoop@ip-172-31-23-100 kafka_2.13-3.0.0]$ [2024-11-05 21:10:04,855] INFO Reading confi
guration from: config/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPee
rConfig)
[2024-11-05 21:10:04,856] WARN config/zookeeper.properties is relative. Prepend ./ to in
dicate that you're sure! (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-11-05 21:10:04,860] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.s
erver.quorum.QuorumPeerConfig)
[2024-11-05 21:10:04,860] INFO secureClientPort is not set (org.apache.zookeeper.server.
quorum.QuorumPeerConfig)
[2024-11-05 21:10:04,860] INFO observerMasterPort is not set (org.apache.zookeeper.serve
r.quorum.QuorumPeerConfig)
[2024-11-05 21:10:04,860] INFO metricsProvider.className is org.apache.zookeeper.metrics
.impl.DefaultMetricsProvider (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-11-05 21:10:04,862] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.
server.DatadirCleanupManager)
[2024-11-05 21:10:04,863] INFO autopurge.purgeInterval set to 0 (org.apache.zookeeper.se
rver.DatadirCleanupManager)
[2024-11-05 21:10:04,863] INFO Purge task is not scheduled. (org.apache.zookeeper.server
.DatadirCleanupManager)
[2024-11-05 21:10:04,863] WARN Either no config or no quorum defined in config, running
in standalone mode (org.apache.zookeeper.server.quorum.QuorumPeerMain)
[2024-11-05 21:10:04,866] INFO Log4j 1.2 jmx support found and enabled. (org.apache.zook
eeper.jmx.ManagedUtil)
[2024-11-05 21:10:04,878] INFO Reading configuration from: config/zookeeper.properties (
org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-11-05 21:10:04,878] WARN config/zookeeper.properties is relative. Prepend ./ to in
dicate that you're sure! (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-11-05 21:10:04,878] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.s
erver.quorum.QuorumPeerConfig)
[2024-11-05 21:10:04,878] INFO secureClientPort is not set (org.apache.zookeeper.server.
quorum.QuorumPeerConfig)
[2024-11-05 21:10:04,878] INFO observerMasterPort is not set (org.apache.zookeeper.serve
r.quorum.QuorumPeerConfig)
[2024-11-05 21:10:04,878] INFO metricsProvider.className is org.apache.zookeeper.metrics
.impl.DefaultMetricsProvider (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-11-05 21:10:04,879] INFO Starting server (org.apache.zookeeper.server.ZooKeeperSer
verMain)
[2024-11-05 21:10:04,894] INFO ServerMetrics initialized with provider org.apache.zookee
per.metrics.impl.DefaultMetricsProvider@3551a94 (org.apache.zookeeper.server.ServerMetri
cs)
```



```
[hadoop@ip-172-31-23-100 kafka_2.13-3.0.0]$ bin/kafka-server-start.sh config/server.proper
ties &
[1] 21336
[hadoop@ip-172-31-23-100 kafka_2.13-3.0.0]$ [2024-11-05 21:11:03,512] INFO Registered kafk
a:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2024-11-05 21:11:03,862] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true
to disable client-initiated TLS renegotiation (org.apache.zookeeper.common.X509Util)
[2024-11-05 21:11:03,965] INFO Registered signal handlers for TERM, INT, HUP (org.apache.k
afka.common.utils.LoggingSignalHandler)
[2024-11-05 21:11:03,980] INFO starting (kafka.server.KafkaServer)
[2024-11-05 21:11:03,981] INFO Connecting to zookeeper on localhost:2181 (kafka.server.Kaf
kaServer)
[2024-11-05 21:11:03,997] INFO [ZooKeeperClient Kafka server] Initializing a new session t
o localhost:2181. (kafka.zookeeper.ZooKeeperClient)
[2024-11-05 21:11:04,003] INFO Client environment:zookeeper.version=3.6.3--6401e4ad2087061
bc6b9f80dec2d69f2e3c8660a, built on 04/08/2021 16:35 GMT (org.apache.zookeeper.ZooKeeper)
[2024-11-05 21:11:04,003] INFO Client environment:host.name=ip-172-31-23-100.us-east-2.com
pute.internal (org.apache.zookeeper.ZooKeeper)
[2024-11-05 21:11:04,003] INFO Client environment:java.version=1.8.0_422 (org.apache.zooke
eper.ZooKeeper)
[2024-11-05 21:11:04,004] INFO Client environment:java.vendor=Amazon.com Inc. (org.apache.
zookeeper.ZooKeeper)
[2024-11-05 21:11:04,004] INFO Client environment:java.home=/usr/lib/jvm/java-1.8.0-amazon
-corretto.x86_64/jre (org.apache.zookeeper.ZooKeeper)
[2024-11-05 21:11:04,004] INFO Client environment:java.class.path=/home/hadoop/kafka_2.13-
3.0.0/bin/../libs/activation-1.1.1.jar:/home/hadoop/kafka_2.13-3.0.0/bin/../libs/aopallian
ce-repackaged-2.6.1.jar:/home/hadoop/kafka_2.13-3.0.0/bin/../libs/argparse4j-0.7.0.jar:/ho
me/hadoop/kafka_2.13-3.0.0/bin/../libs/audience-annotations-0.5.0.jar:/home/hadoop/kafka_2
.13-3.0.0/bin/../libs/commons-cli-1.4.jar:/home/hadoop/kafka_2.13-3.0.0/bin/../libs/common
s-lang3-3.8.1.jar:/home/hadoop/kafka_2.13-3.0.0/bin/../libs/connect-api-3.0.0.jar:/home/ha
doop/kafka_2.13-3.0.0/bin/../libs/connect-basic-auth-extension-3.0.0.jar:/home/hadoop/kafk
a_2.13-3.0.0/bin/../libs/connect-file-3.0.0.jar:/home/hadoop/kafka_2.13-3.0.0/bin/../libs/
connect-json-3.0.0.jar:/home/hadoop/kafka_2.13-3.0.0/bin/../libs/connect-mirror-3.0.0.jar:
/home/hadoop/kafka_2.13-3.0.0/bin/../libs/connect-mirror-client-3.0.0.jar:/home/hadoop/kaf
ka_2.13-3.0.0/bin/../libs/connect-runtime-3.0.0.jar:/home/hadoop/kafka_2.13-3.0.0/bin/../l
ibs/connect-transforms-3.0.0.jar:/home/hadoop/kafka_2.13-3.0.0/bin/../libs/hk2-api-2.6.1.j
ar:/home/hadoop/kafka_2.13-3.0.0/bin/../libs/hk2-locator-2.6.1.jar:/home/hadoop/kafka_2.13
-3.0.0/bin/../libs/hk2-utils-2.6.1.jar:/home/hadoop/kafka_2.13-3.0.0/bin/../libs/jackson-a
nnotations-2.12.3.jar:/home/hadoop/kafka_2.13-3.0.0/bin/../libs/jackson-core-2.12.3.jar:/h
ome/hadoop/kafka_2.13-3.0.0/bin/../libs/jackson-databind-2.12.3.jar:/home/hadoop/kafka_2.1
3-3.0.0/bin/../libs/jackson-dataformat-csv-2.12.3.jar:/home/hadoop/kafka_2.13-3.0.0/bin/..
/libs/jackson-datatype-jdk8-2.12.3.jar:/home/hadoop/kafka_2.13-3.0.0/bin/../libs/jackson-j
axrs-base-2.12.3.jar:/home/hadoop/kafka_2.13-3.0.0/bin/../libs/jackson-jaxrs-json-provider
```

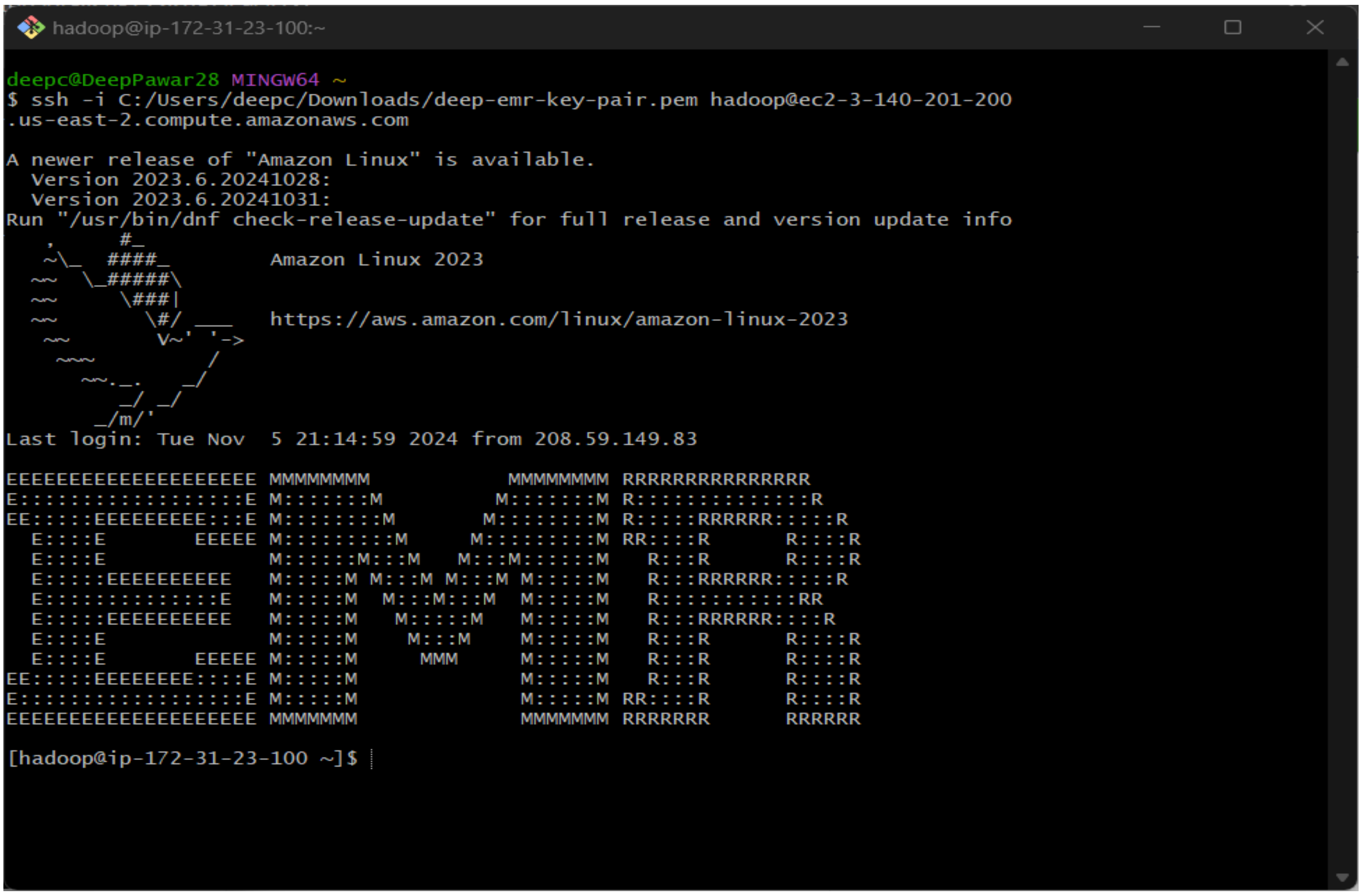
Step D – Prepare to run Kafka producers and consumers

- Producer-Term:



```
hadoop@ip-172-31-23-100:~

deepc@DeepPawar28 MINGW64 ~
$ ssh -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem hadoop@ec2-3-140-201-200
.us-east-2.compute.amazonaws.com

A newer release of "Amazon Linux" is available.
  Version 2023.6.20241028:
  Version 2023.6.20241031:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Nov  5 21:06:52 2024

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-23-100 ~]$ |
```

- Consumer-Term:



```
hadoop@ip-172-31-23-100:~

deepc@DeepPawar28 MINGW64 ~
$ ssh -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem hadoop@ec2-3-140-201-200
.us-east-2.compute.amazonaws.com

A newer release of "Amazon Linux" is available.
  Version 2023.6.20241028:
  Version 2023.6.20241031:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Nov  5 21:14:59 2024 from 208.59.149.83

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-23-100 ~]$ |
```

Step E – Create a Kafka topic

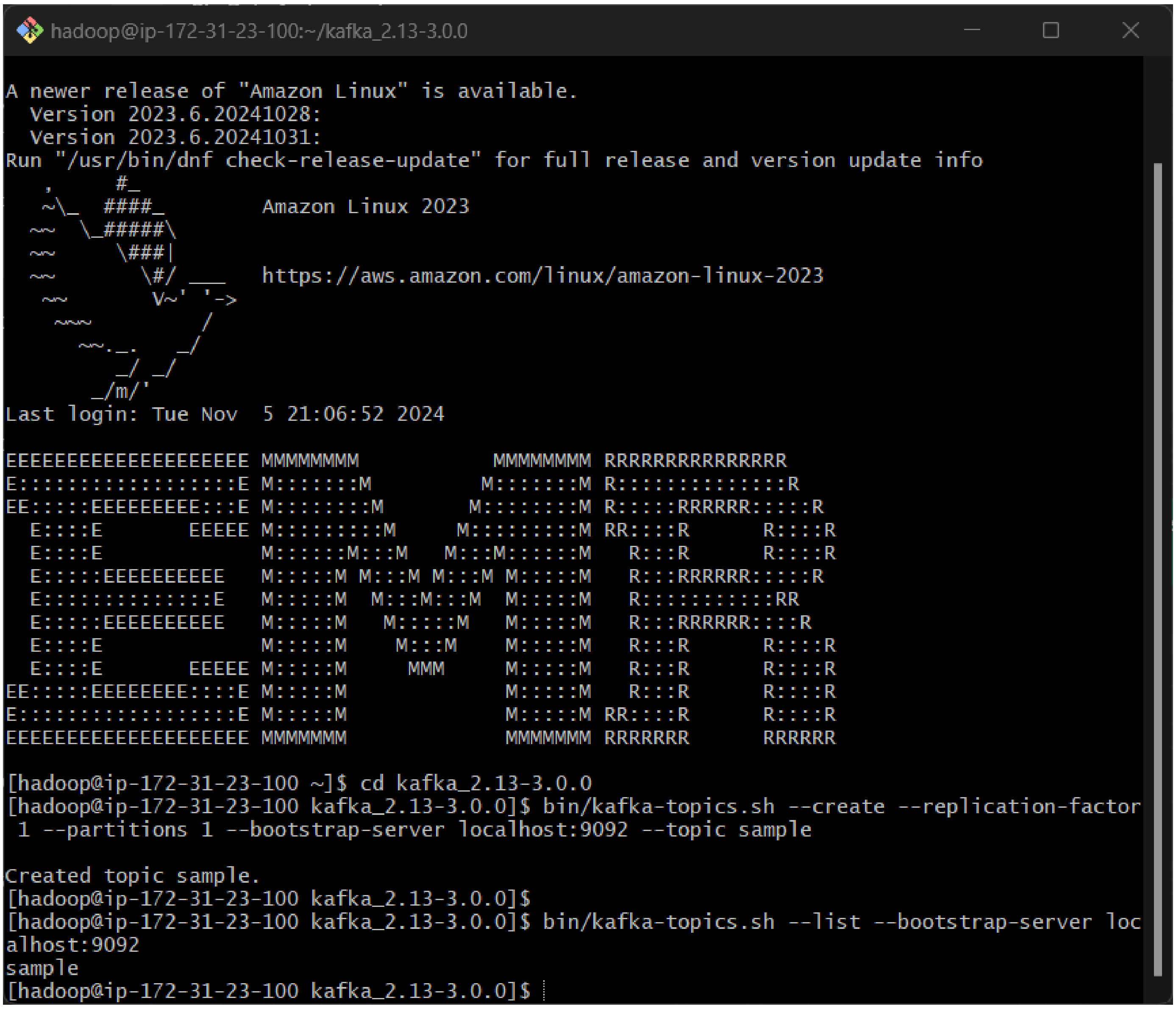
- In the Producer-Term, enter the following command to create a new kafka topic called 'sample':

cd kafka_2.13-3.0.0

bin/kafka-topics.sh --create --replication-factor 1 --partitions 1 --bootstrap-server localhost:9092 --topic sample

- To list the topics that you created you can enter the following into the Producer-Term:

bin/kafka-topics.sh --list --bootstrap-server localhost:9092

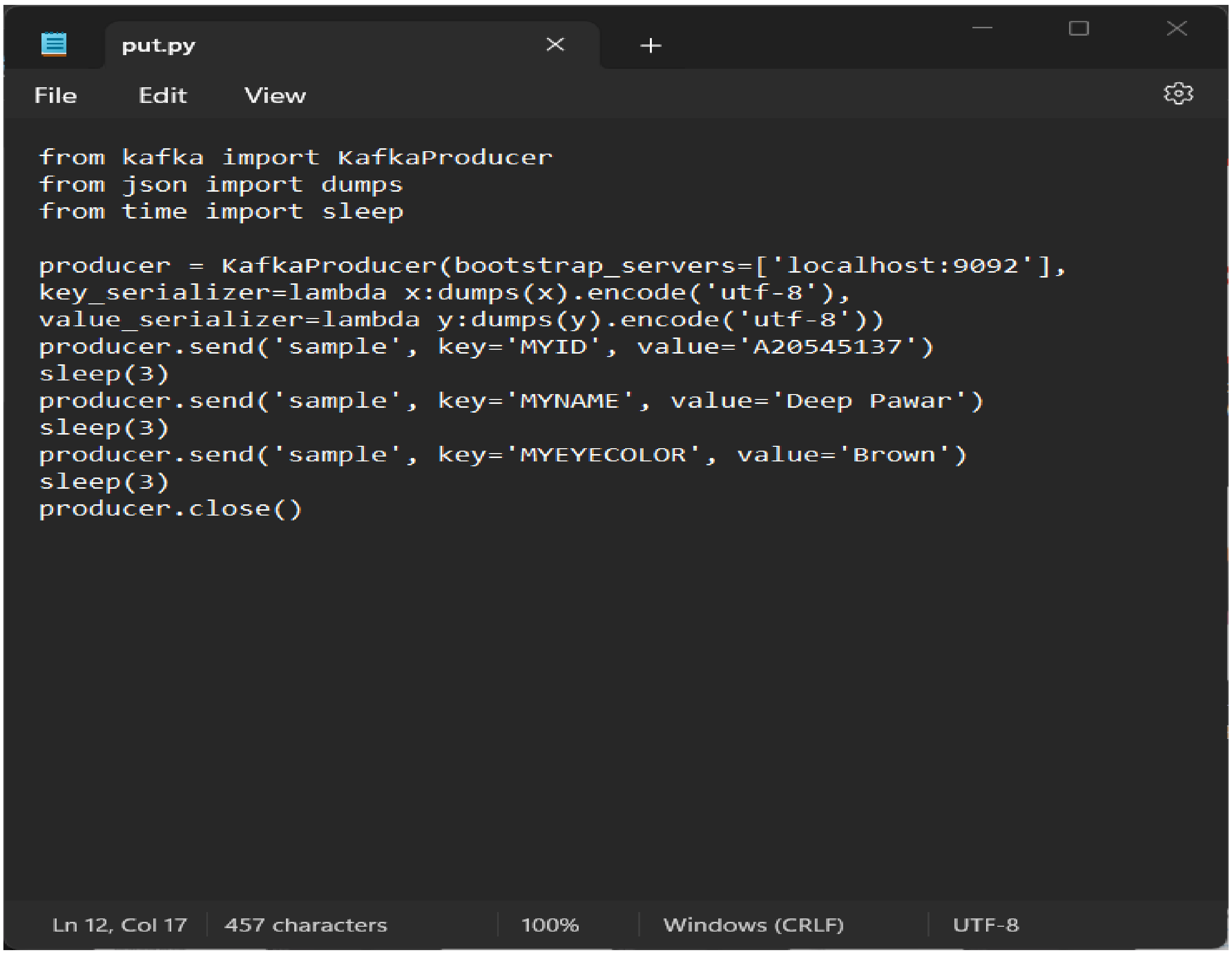


```
A newer release of "Amazon Linux" is available.
  Version 2023.6.20241028:
  Version 2023.6.20241031:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Nov  5 21:06:52 2024

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-23-100 ~]$ cd kafka_2.13-3.0.0
[hadoop@ip-172-31-23-100 kafka_2.13-3.0.0]$ bin/kafka-topics.sh --create --replication-factor
 1 --partitions 1 --bootstrap-server localhost:9092 --topic sample

Created topic sample.
[hadoop@ip-172-31-23-100 kafka_2.13-3.0.0]$
[hadoop@ip-172-31-23-100 kafka_2.13-3.0.0]$ bin/kafka-topics.sh --list --bootstrap-server loc
alhost:9092
sample
[hadoop@ip-172-31-23-100 kafka_2.13-3.0.0]$
```

a)

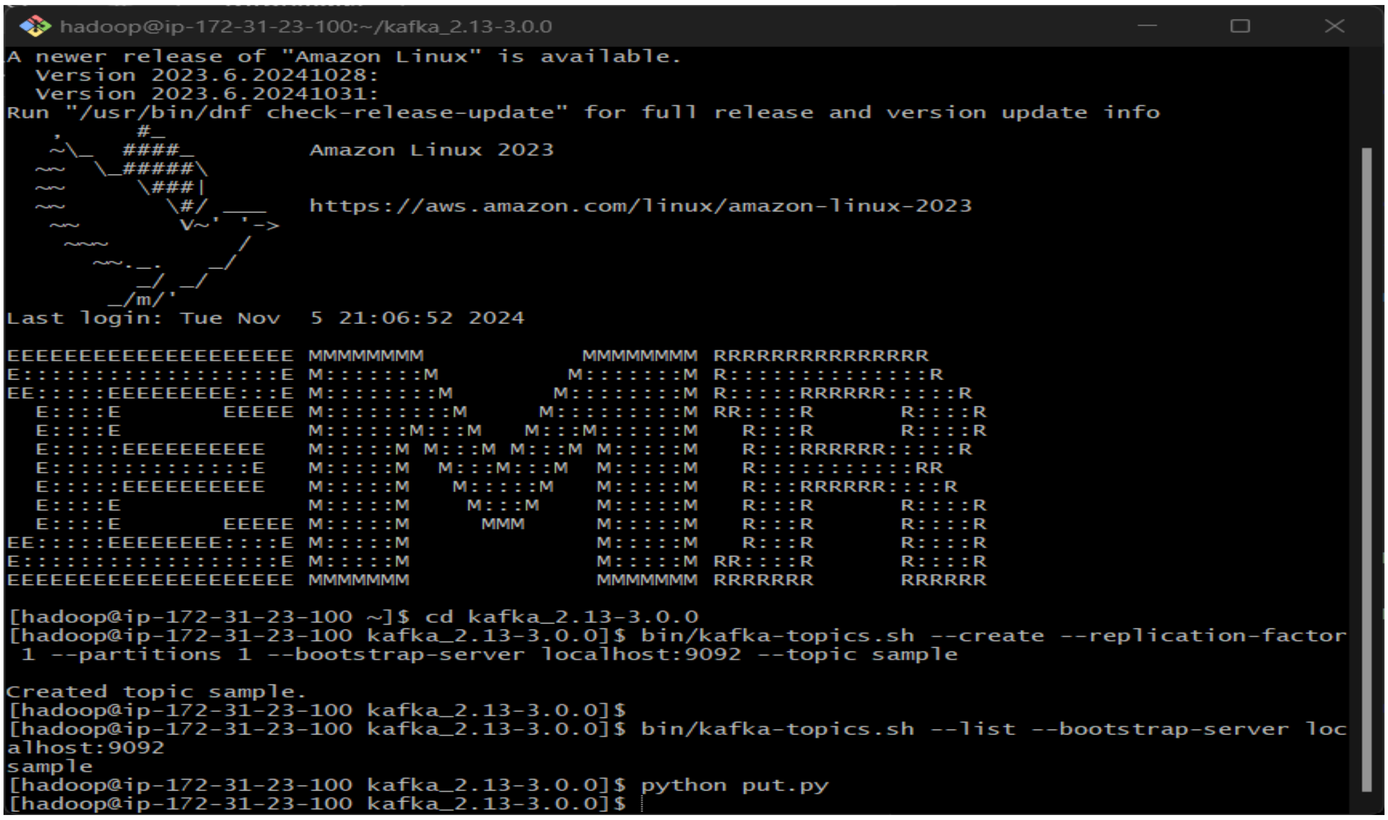
Ans:

- put.py



```
from kafka import KafkaProducer
from json import dumps
from time import sleep

producer = KafkaProducer(bootstrap_servers=['localhost:9092'],
key_serializer=lambda x:dumps(x).encode('utf-8'),
value_serializer=lambda y:dumps(y).encode('utf-8'))
producer.send('sample', key='MYID', value='A20545137')
sleep(3)
producer.send('sample', key='MYNAME', value='Deep Pawar')
sleep(3)
producer.send('sample', key='MYEYECOLOR', value='Brown')
sleep(3)
producer.close()
```

- Execution:

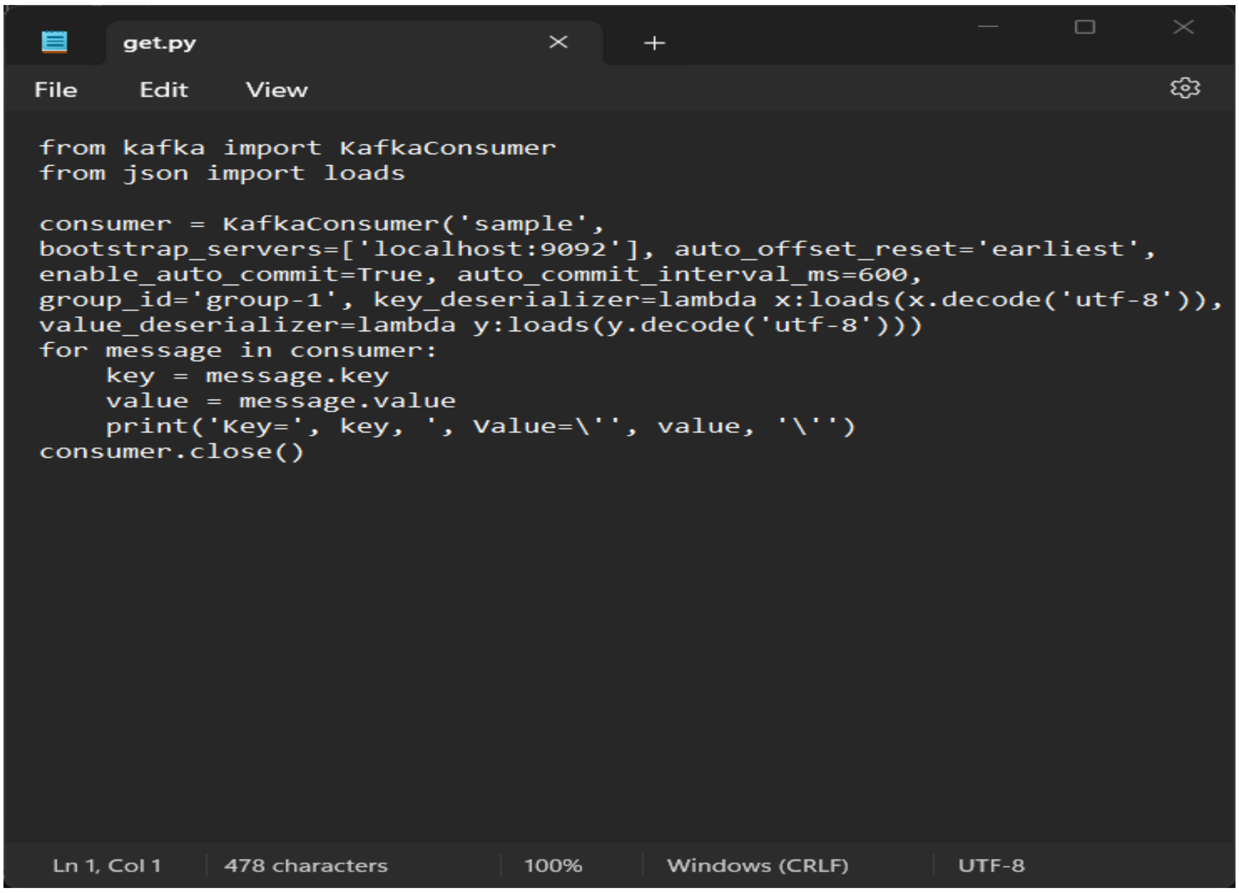


```
A newer release of "Amazon Linux" is available.
  Version 2023.6.20241028:
  Version 2023.6.20241031:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Nov  5 21:06:52 2024

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-23-100 ~]$ cd kafka_2.13-3.0.0
[hadoop@ip-172-31-23-100 kafka_2.13-3.0.0]$ bin/kafka-topics.sh --create --replication-factor
 1 --partitions 1 --bootstrap-server localhost:9092 --topic sample

Created topic sample.
[hadoop@ip-172-31-23-100 kafka_2.13-3.0.0]$
[hadoop@ip-172-31-23-100 kafka_2.13-3.0.0]$ bin/kafka-topics.sh --list --bootstrap-server loc
alhost:9092
sample
[hadoop@ip-172-31-23-100 kafka_2.13-3.0.0]$ python put.py
[hadoop@ip-172-31-23-100 kafka_2.13-3.0.0]$
```

b)

Ans:

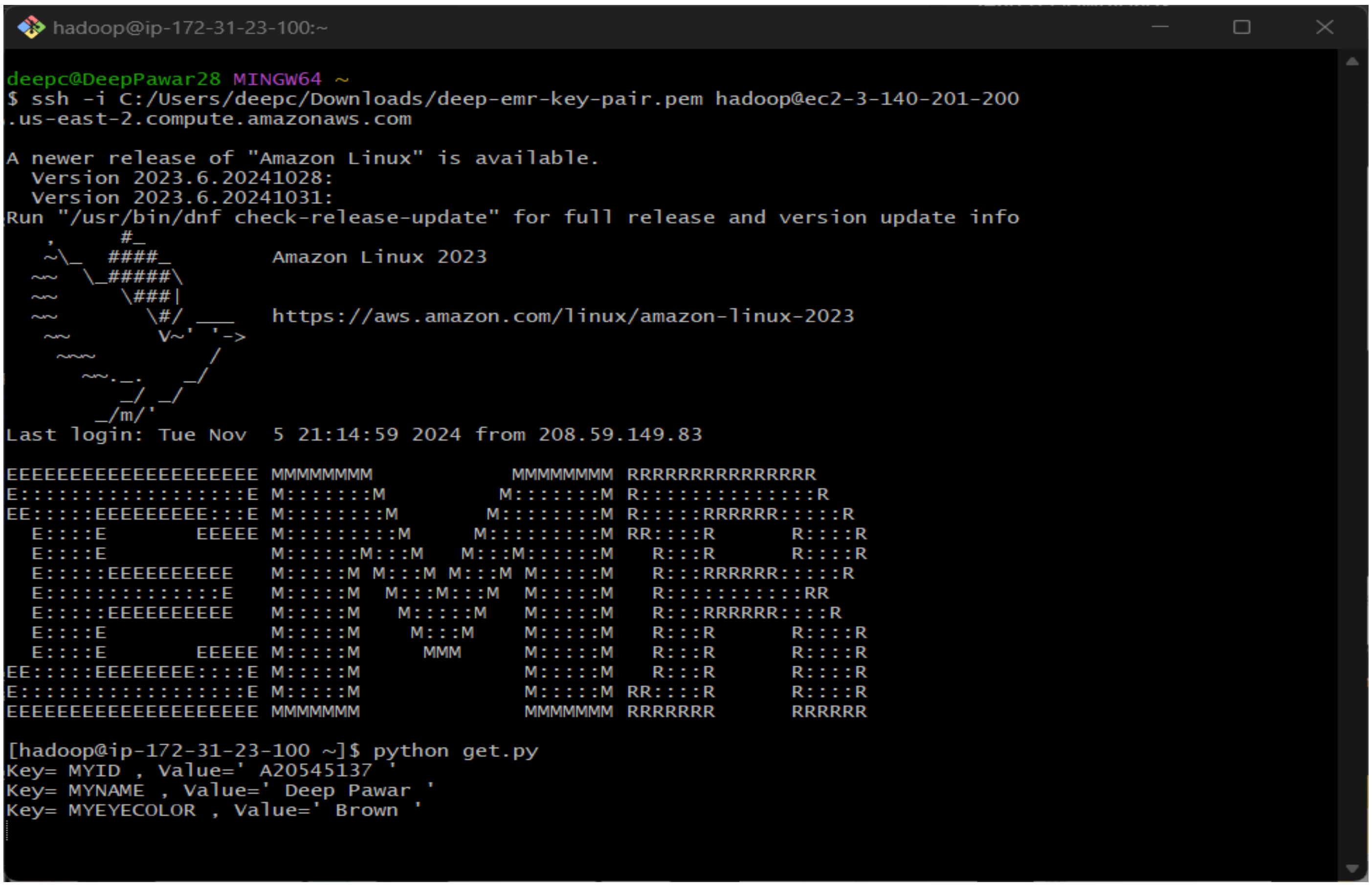
- get.py



```
from kafka import KafkaConsumer
from json import loads

consumer = KafkaConsumer('sample',
bootstrap_servers=['localhost:9092'], auto_offset_reset='earliest',
enable_auto_commit=True, auto_commit_interval_ms=600,
group_id='group-1', key_deserializer=lambda x:loads(x.decode('utf-8')),
value_deserializer=lambda y:loads(y.decode('utf-8')))
for message in consumer:
    key = message.key
    value = message.value
    print('Key=', key, ', Value=\'', value, '\'')
consumer.close()
```

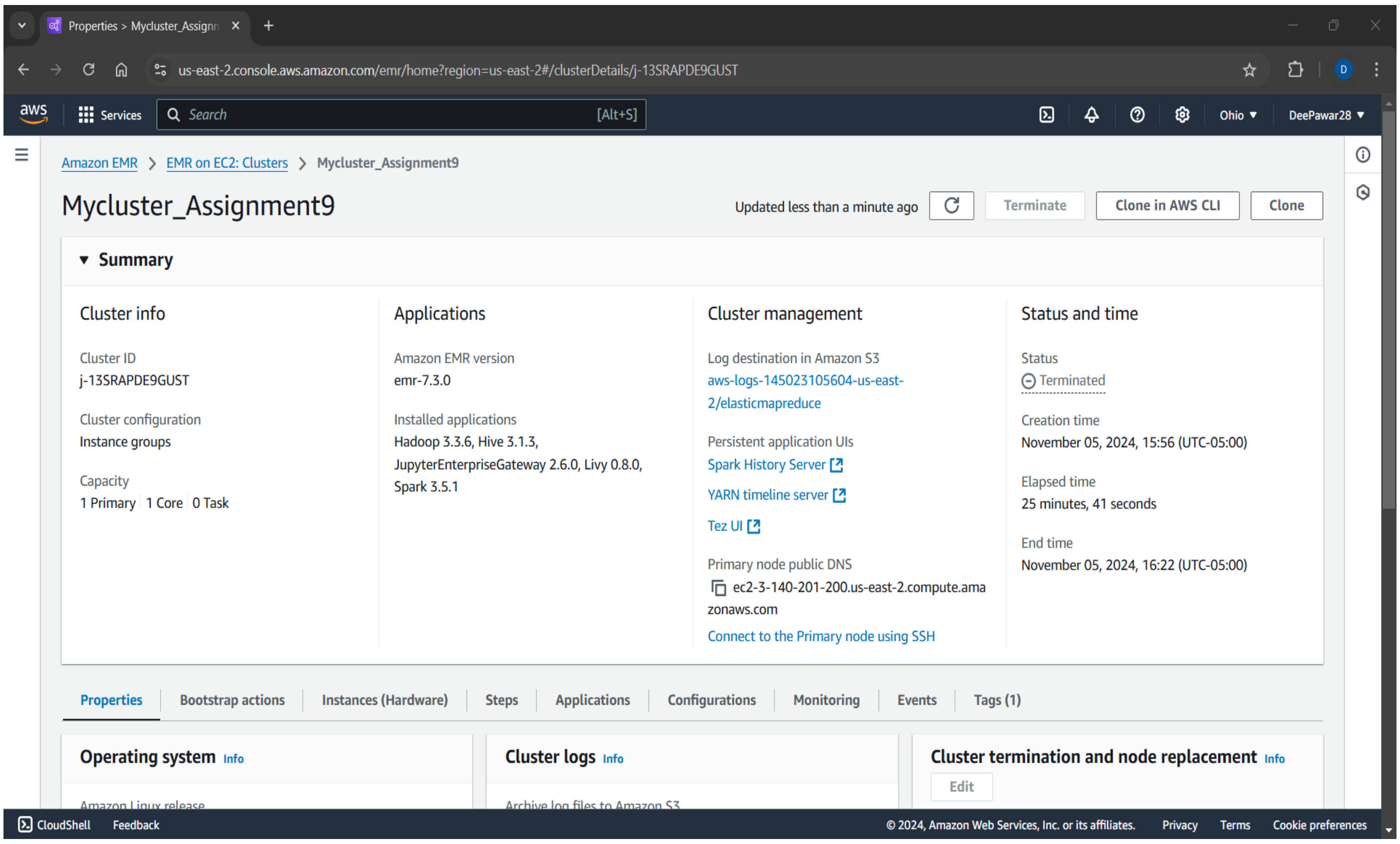
```
Ln 1, Col 1   478 characters   100%   Windows (CRLF)   UTF-8
```

- Execution:



```
hadoop@ip-172-31-23-100:~

deepc@DeepPawar28 MINGW64 ~
$ ssh -i C:/Users/deepc/Downloads/deep-emr-key-pair.pem hadoop@ec2-3-140-201-200
.us-east-2.compute.amazonaws.com

A newer release of "Amazon Linux" is available.
  Version 2023.6.20241028:
  Version 2023.6.20241031:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Nov  5 21:14:59 2024 from 208.59.149.83

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-23-100 ~]$ python get.py
Key= MYID , Value=' A20545137 '
Key= MYNAME , Value=' Deep Pawar '
Key= MYEYECOLOR , Value=' Brown '
```

c) Cluster Termination:



Properties > Mycluster_Assignm
us-east-2.console.aws.amazon.com/emr/home?region=us-east-2#/clusterDetails/j-13SRAPDE9GUST
Services
Search
[Alt+S]
Ohio
DeePawar28
Amazon EMR > EMR on EC2: Clusters > Mycluster_Assignment9
Updated less than a minute ago
Terminate
Clone in AWS CLI
Clone
Properties
Bootstrap actions
Instances (Hardware)
Steps
Applications
Configurations
Monitoring
Events
Tags (1)
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Mycluster_Assignment9

Summary

Cluster info

Cluster ID
j-13SRAPDE9GUST

Cluster configuration
Instance groups

Capacity
1 Primary 1 Core 0 Task

Applications

Amazon EMR version
emr-7.3.0

Installed applications
Hadoop 3.3.6, Hive 3.1.3, JupyterEnterpriseGateway 2.6.0, Livy 0.8.0, Spark 3.5.1

Cluster management

Log destination in Amazon S3
aws-logs-145023105604-us-east-2/elasticmapreduce

Persistent application UIs
Spark History Server
YARN timeline server
Tez UI

Primary node public DNS
ec2-3-140-201-200.us-east-2.compute.amazonaws.com
Connect to the Primary node using SSH

Status and time

Status
Terminated

Creation time
November 05, 2024, 15:56 (UTC-05:00)

Elapsed time
25 minutes, 41 seconds

End time
November 05, 2024, 16:22 (UTC-05:00)

Operating system Info

Cluster logs Info

Cluster termination and node replacement Info
Edit