

Assignment

- Course Title: Software Tools for Data Science
- Course Code: STD912S
- Assessment: Second Assignment

Group members: Olivia T Haikali

Kornelia N Nghinaunye

Fransina N Petrus

Step 1: Installation and setup of the cluster. (15%)

The setup of the Apache Spark cluster was a critical part of this assignment, which required the deployment of a reliable and efficient three-node cluster. Below are the steps and configurations used to set up the cluster for real-time data processing.

- We deployed a three-node cluster, which includes one master node that is responsible for managing cluster resources and distributing tasks across worker nodes. We however cloned the master node into two worker nodes. We initially created a Hadoop environment before cloning the main node. We then installed Apache Spark and the necessary dependencies for communication within the clusters. Necessary configuration of files was performed in all the nodes where the master managed the task distribution, and the workers processed the data streams in parallel.

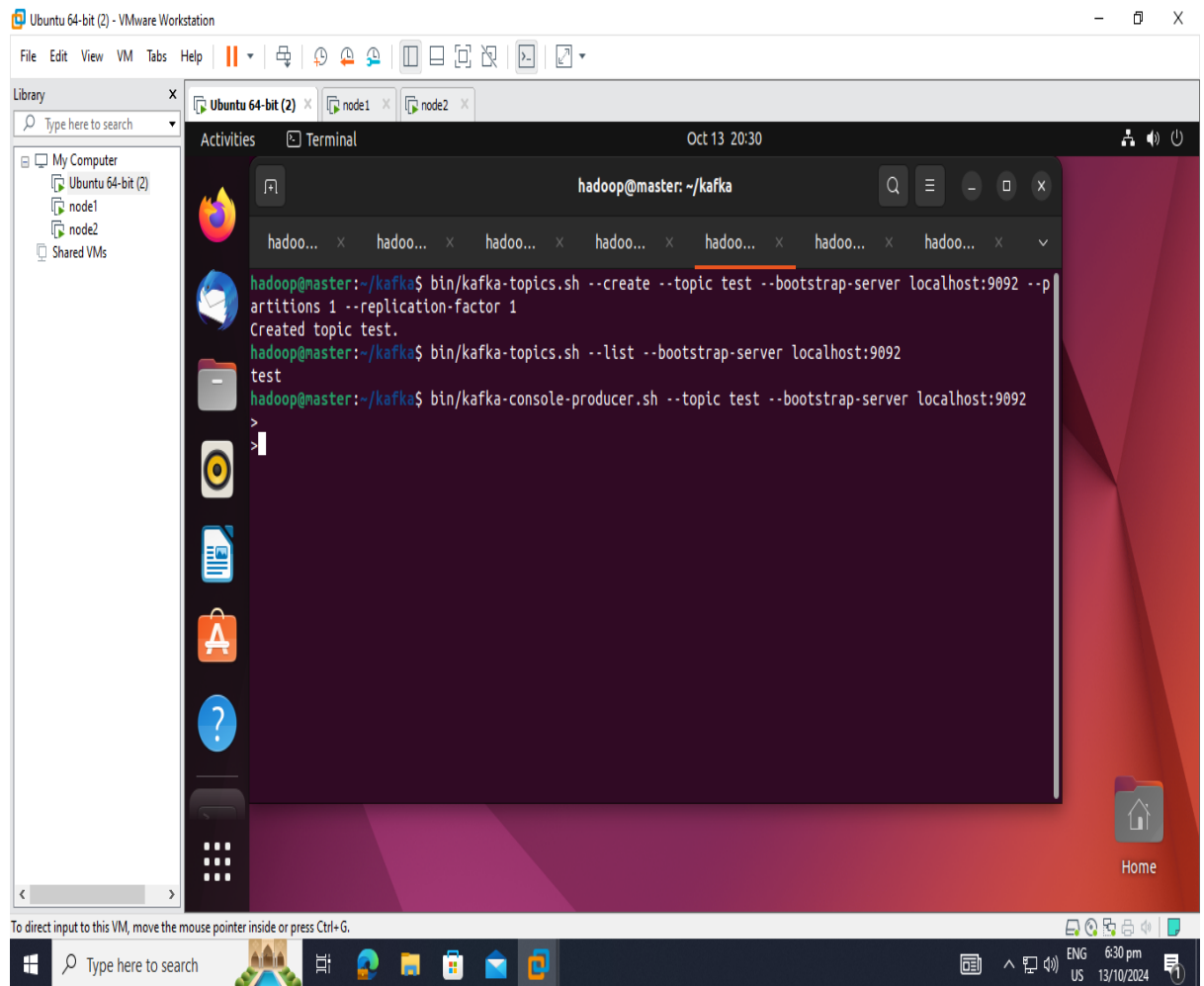
2. Installation and setup of Kafka. (35%)

Initially, we confirmed that java was successfully installed since it operates on it.

The following are the steps used to install and setup Kafka which we did on all the nodes.

- Navigate to the official Apache Kafka downloads page and download the kafka version
- We extracted kafka using the tar command
- We configured kafka specifically on the following files: Zookeeper configuration file, Kafka's server properties, Key settings to configure: broker.id: **Unique ID** for each Kafka broker, **log.dirs**: Directory where Kafka will store logs, as well as **zookeeper.connect**: Zookeeper instance to connect to, usually localhost for a local setup.
- Started the Zookeeper server that Kafka will rely on

- We then started the Kafka broker
- Creation of topics

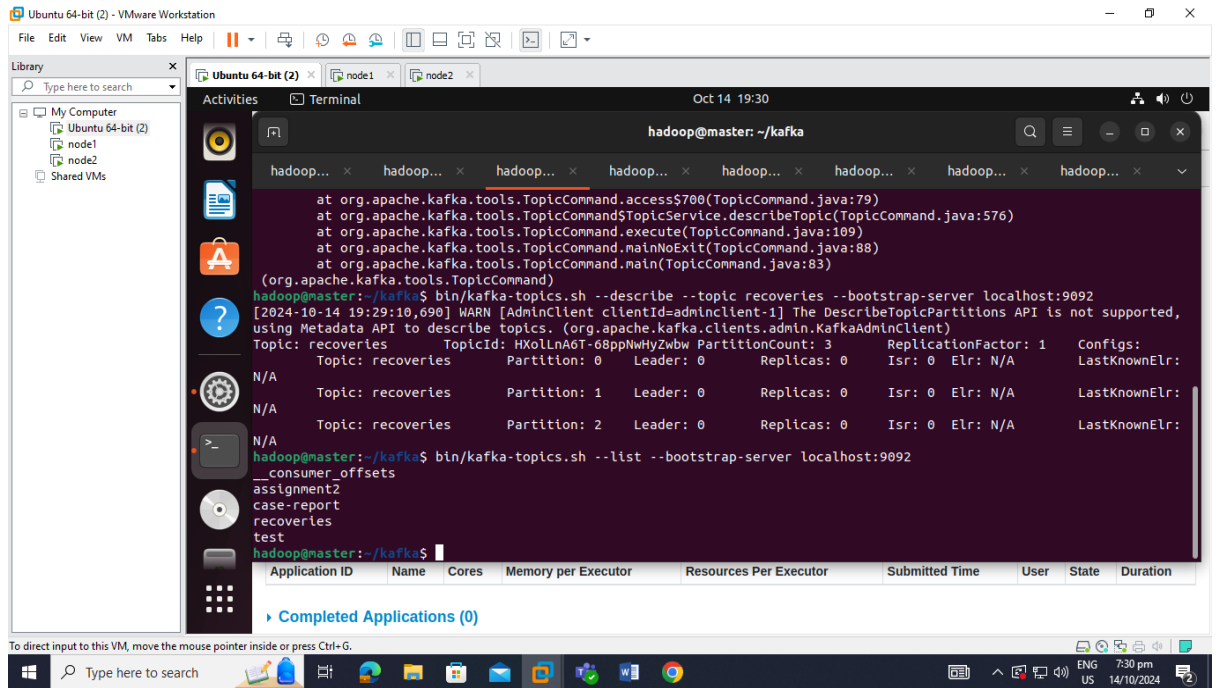


The screenshot shows a VMware Workstation interface with an Ubuntu 64-bit (2) VM. The terminal window is open, showing the following commands and output:

```
hadoop@master: ~/kafka
hadoop@master:~/kafka$ bin/kafka-topics.sh --create --topic test --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1
Created topic test.
hadoop@master:~/kafka$ bin/kafka-topics.sh --list --bootstrap-server localhost:9092
test
hadoop@master:~/kafka$ bin/kafka-console-producer.sh --topic test --bootstrap-server localhost:9092
>
>
```

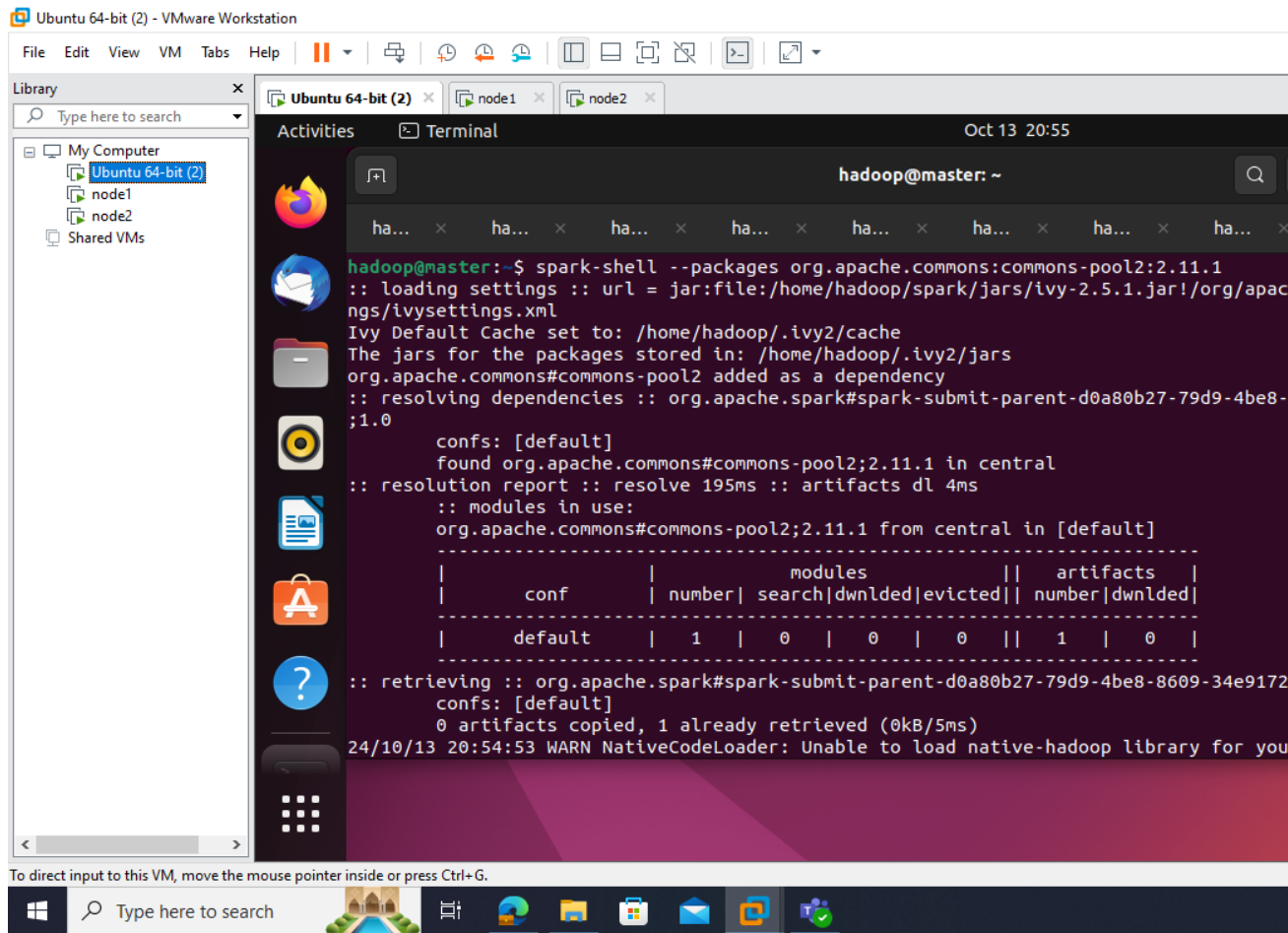
The terminal window also shows a search bar and a list of open windows (hadoop...). The VMware interface includes a menu bar (File, Edit, View, VM, Tabs, Help), a toolbar, and a sidebar with a library of VMs (My Computer, Ubuntu 64-bit (2), node1, node2, Shared VMs). The bottom status bar indicates the date and time (Oct 13 20:30) and the language (ENG US).

- Describing the topic created for recoveries



- Monitor kafka

statring kafka



Downloading the necessary jar file for spark

Ubuntu 64-bit (2) - VMware Workstation

File Edit View VM Tabs Help

Library

Type here to search

- My Computer
 - Ubuntu 64-bit (2)
 - node1
 - node2
- Shared VMs

Activities

Terminal

Home / spark / jars

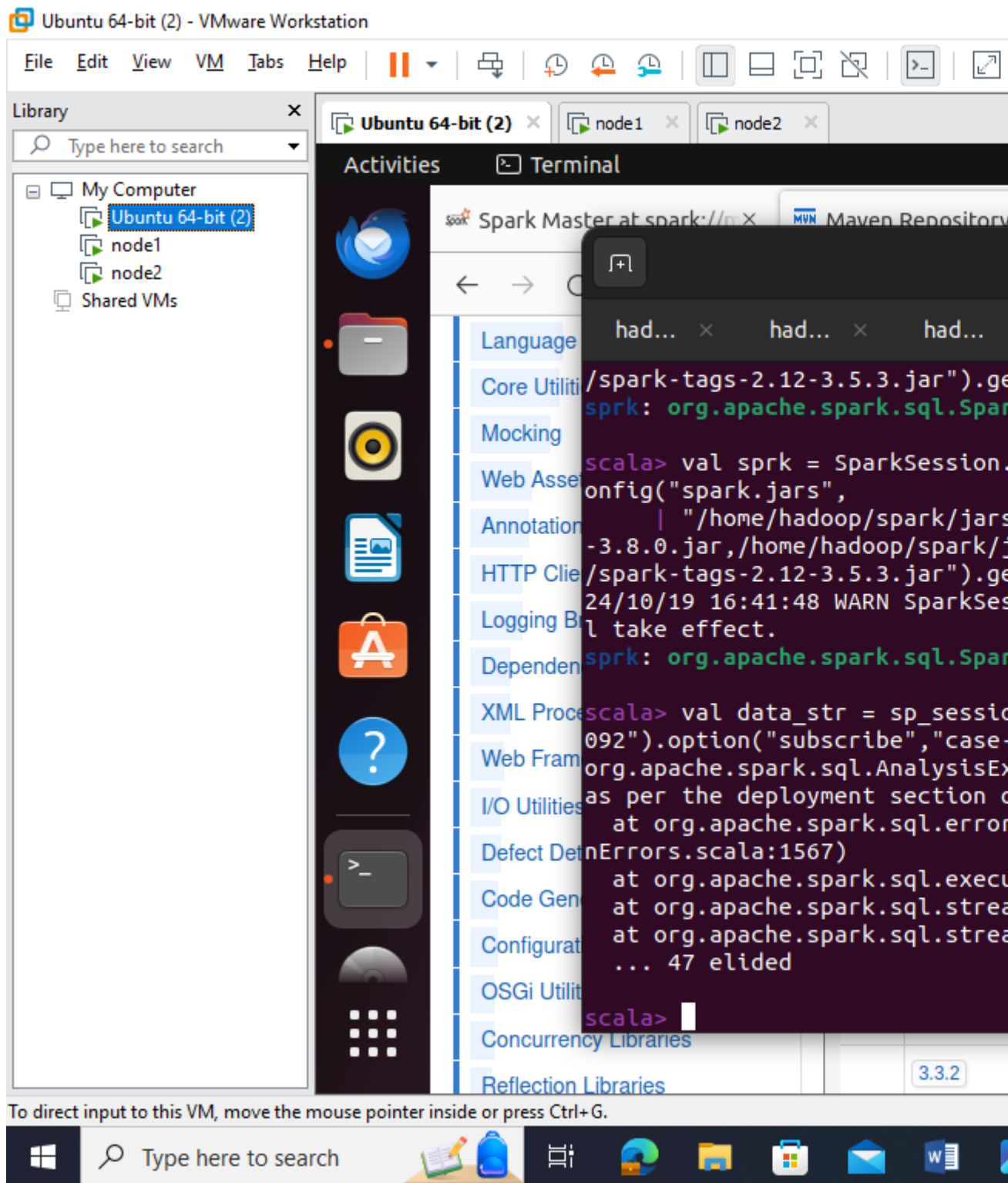
```
transaction-api-1.1.jar
univocity-parsers-2.9.1.jar
xbean-asm9-shaded-4.23.jar
xz-1.9.jar
zjsonpatch-0.3.0.jar
zookeeper-3.6.3.jar
zookeeper-jute-3.6.3.jar
zstd-jni-1.5.5-4.jar
hadoop@master:~/spark/jars$ wget https://repo1.maven.org/maven2/org/apache/spark/spark-sql-kafka-0-10_2.13/3.5.3/spark-sql-kafka-0-10_2.13-3.5.3.jar
--2024-10-19 16:27:55-- https://repo1.maven.org/maven2/org/apache/spark/spark-sql-kafka-0-10_2.13/3.5.3/spark-sql-kafka-0-10_2.13-3.5.3.jar
Resolving repo1.maven.org (repo1.maven.org)...
Connecting to repo1.maven.org (repo1.maven.org)...
HTTP request sent, awaiting response...
Length: 440563 (430K) [application/java-archive]
Saving to: 'spark-sql-kafka-0-10_2.13-3.5.3.jar'

spark-sql-kafka-0-10_2.13-3.5.3.jar 100% [3.7 MB/s]
2024-10-19 16:27:56 (847 KB/s)

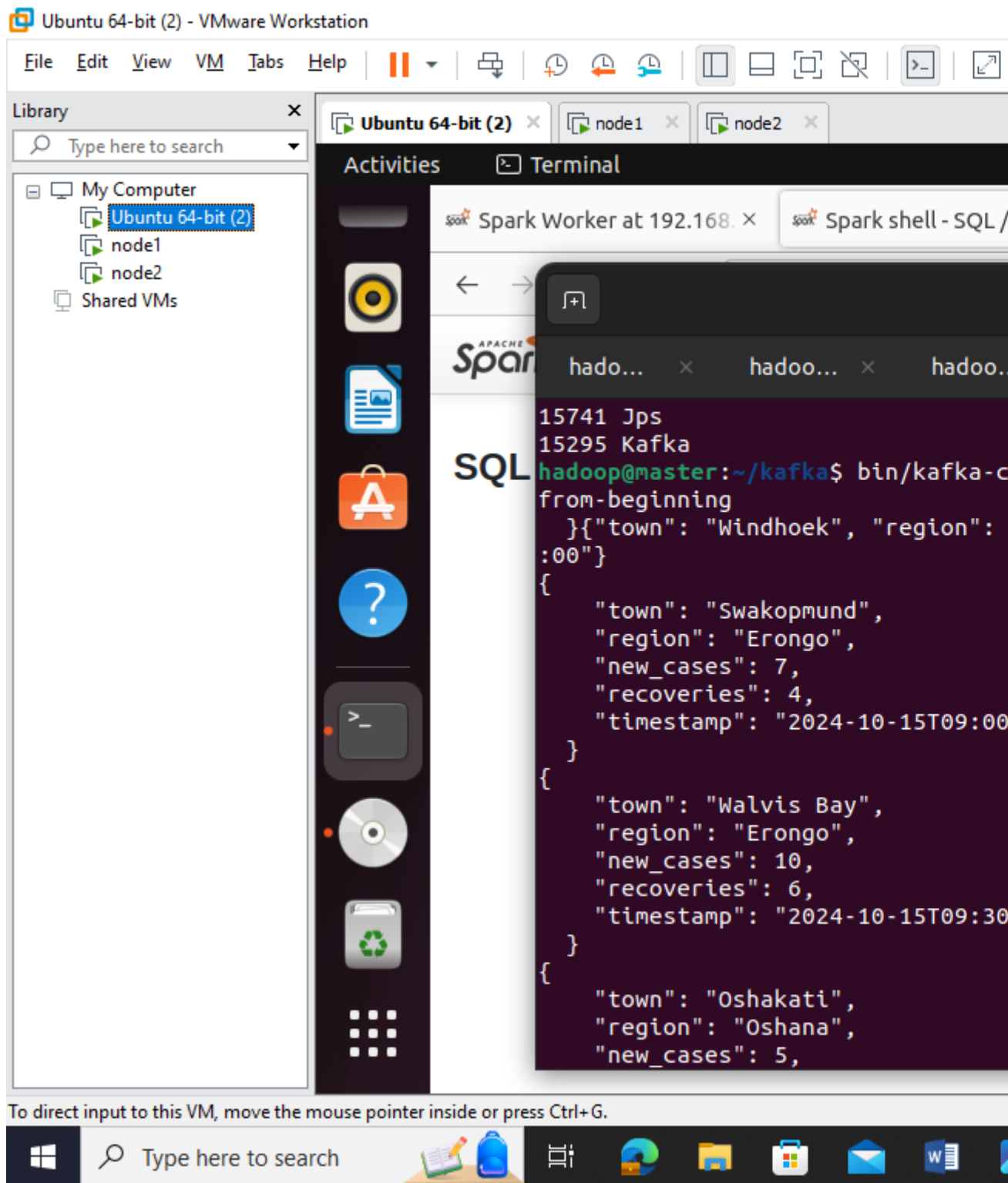
hadoop@master:~/spark/jars$ ls
activation-1.1.1.jar
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Type here to search



Consuming the data from the producer in to kafka topic case-report



Error when trying to create a spark session

Ubuntu 64-bit (2) - VMware Workstation

File Edit View VM Tabs Help

Library

Type here to search

- My Computer
 - Ubuntu 64-bit (2)
 - node1
 - node2
- Shared VMs

Activities

Terminal

Spark Worker at 192.168. x

Spark shell - SQL /

hadoo... x hadoo... x hadoo...

```
scala> import org.apache.spark.sql
import org.apache.spark.sql.SparkS

scala> import org.apache.spark.sql
import org.apache.spark.sql.stream

scala> import org.apache.commons.p
import org.apache.commons.pool2.im

scala> val sprk = SparkSession.bui
g("spark.jars",
  | "/home/hadoop/spark/jars/sp
.0.jar,/home/hadoop/spark/jars/spa
ags-2.12-3.5.3.jar").getOrElse()
24/10/19 18:45:16 WARN SparkSessio
ke effect.
sprk: org.apache.spark.sql.SparkSe

scala> val data_str = sprk.readStr
on("subscribe","case-report").opti
java.lang.NoClassDefFoundError: sc
at org.apache.spark.sql.kafka010
idateStreamOptions(KafkaSourceProv
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Type here to search