Installation of Apache Kafka on Centos

Step 1. Connect to the Server

Log in to the server via SSH as user root using the following command:

```
ssh root@IP_ADDRESS -p PORT_NUMBER
```

replace “IP_ADDRESS” and “PORT_NUMBER” with your actual server IP address and SSH port number.

Step 2: Update OS Packages

Once logged in, make sure that your server OS packages are up-to-date by running the following commands:

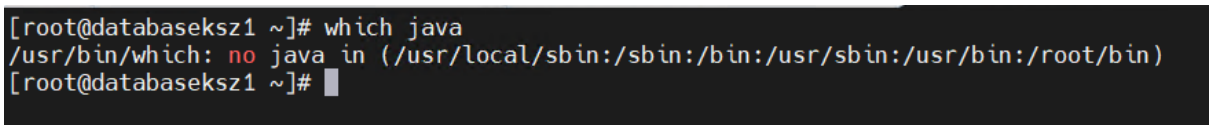
```
yum clean all
```

```
yum update
```

Step 3: Install JAVA

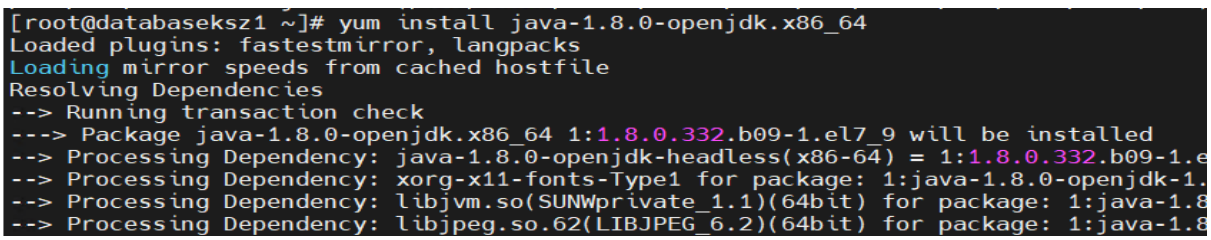
Apache Kafka requires Java, so in order to run it on your server, we need to install Java first. We can check if Java is already installed on the server using this command:

```
which java
```



If there is no output, it means that Java is not installed on the server yet. We can install Java from a RPM package:

```
yum install java-1.8.0-openjdk.x86_64
```



We can check the Java version installed on the server by running the following command:

```
java -version
```

The output should be similar to this:

```
[kafka@databaseksz1 ~]$ java -version
openjdk version "1.8.0_332"
OpenJDK Runtime Environment (build 1.8.0_332-b09)
OpenJDK 64-Bit Server VM (build 25.332-b09, mixed mode)
[kafka@databaseksz1 ~]$
```

Add the "JAVA_HOME" and "JRE_HOME" environment variables at the end of `/etc/bashrc` file:

```
sudo vi /etc/bashrc
```

```
[root@databaseksz1 ~]# sudo vi /etc/bashrc
```

Append the following lines to the original content of the file:

```
export JRE_HOME=/usr/lib/jvm/jre
export JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk
PATH=$PATH:$JRE_HOME:$JAVA_HOME
```

```
# vim:ts=4:sw=4
export JRE_HOME=/usr/lib/jvm/jre
export JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk
PATH=$PATH:$JRE_HOME:$JAVA_HOME
-- INSERT --
```

Open the `~/.bashrc` file and make sure that the following lines exist:

```
[root@databaseksz1 ~]# vim ~/.bashrc
```

```
if [ -f /etc/bashrc ] ; then
  . /etc/bashrc
Fi
```

```
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
```

Run the following command to activate the path settings immediately:

```
source /etc/bashrc
```

```
[root@databaseksz1 ~]# source /etc/bashrc
[root@databaseksz1 ~]#
```

Step 4: Install Apache Kafka

Create a new system user dedicated for the Kafka service using the following command:

```
useradd kafka -m
```

```
[root@databaseksz1 ~]# useradd kafka -m
```

Set a password for the newly created user:

```
passwd kafka
```

```
[root@databaseksz1 ~]# passwd kafka
Changing password for user kafka.
New password:
BAD PASSWORD: The password contains less than 1 digits
Retype new password:
passwd: all authentication tokens updated successfully.
[root@databaseksz1 ~]#
```

Use a strong password and enter it twice. Then, run the following command on the server:

```
sudo usermod -aG wheel kafka
```

```
[root@databaseksz1 ~]# sudo usermod -aG wheel kafka
[root@databaseksz1 ~]#
```

Log in as the newly created user with:

```
su kafka
```

```
[root@databaseksz1 ~]# su kafka
[kafka@databaseksz1 root]$
```

Download the latest version of Apache Kafka available or old versions at https://kafka.apache.org/downloads / https://archive.apache.org/dist/kafka/ and extract it in the home directory of the kafka user account:

```
cd ~

wget https://archive.apache.org/dist/kafka/0.10.1.0/kafka_2.11-0.10.1.0.tgz
```

```
[kafka@databaseksz1 root]$ cd ~
[kafka@databaseksz1 ~]$ wget https://archive.apache.org/dist/kafka/0.10.1.0/kafka_2.11-0.10.1.0.tgz
--2022-05-26 05:57:20--  https://archive.apache.org/dist/kafka/0.10.1.0/kafka_2.11-0.10.1.0.tgz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:172:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 34373824 (33M) [application/x-gzip]
Saving to: 'kafka_2.11-0.10.1.0.tgz'

100%[==========================================================================================
```

```
tar -xvzf kafka_2.12-2.1.0.tgz

mv kafka_2.12-2.1.0/* .
```

```
tar -xvzf kafka_2.11-0.10.1.0.tgz
mv kafka_2.11-0.10.1.0
mv kafka_2.11-0.10.1.0/* .
```

```
rmdir /home/kafka/kafka_2.12-2.1.0
```

```
rmdir /home/kafka/kafka_2.11-0.10.1.0
```

Apache Kafka uses ZooKeeper to store persistent cluster metadata, so we need to install ZooKeeper. The ZooKeeper files are included with Apache Kafka. ZooKeeper is running on port 2181 and it doesn't require much maintenance. The ZooKeeper service is responsible for configuration management, leader detection, synchronization, etc.
Create a ZooKeeper systemd unit file so that we can run ZooKeeper as a service:

```
sudo vi /lib/systemd/system/zookeeper.service
```

```
[kafka@databaseksz1 ~]$ sudo vi /lib/systemd/system/zookeeper.service
```

```
[Unit]

Requires=network.target remote-fs.target

After=network.target remote-fs.target


[Service]

Type=simple

User=kafka

ExecStart=/home/kafka/bin/zookeeper-server-start.sh
/home/kafka/config/zookeeper.properties

ExecStop=/home/kafka/bin/zookeeper-server-stop.sh

Restart=on-abnormal
```

```
[Install]

WantedBy=multi-user.target
```

```
[Unit]
Requires=network.target remote-fs.target
After=network.target remote-fs.target

[Service]
Type=simple
User=kafka
ExecStart=/home/kafka/bin/zookeeper-server-start.sh /home/kafka/config/zookeeper.properties
ExecStop=/home/kafka/bin/zookeeper-server-stop.sh
Restart=on-abnormal

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

Create a systemd unit file for Apache Kafka:

```
sudo vi /etc/systemd/system/kafka.service
```

```
[kafka@databaseksz1 ~]$ sudo vi /etc/systemd/system/kafka.service
[kafka@databaseksz1 ~]$
```

Add the following lines:

```
[Unit]

Requires=network.target remote-fs.target zookeeper.service

After=network.target remote-fs.target zookeeper.service


[Service]

Type=simple

User=kafka

ExecStart=/home/kafka/bin/kafka-server-start.sh
/home/kafka/config/server.properties

ExecStop=/home/kafka/bin/kafka-server-stop.sh

Restart=on-abnormal

[Install]

WantedBy=multi-user.target
```

```
[Unit]
Requires=network.target remote-fs.target zookeeper.service
After=network.target remote-fs.target zookeeper.service

[Service]
Type=simple
User=kafka
ExecStart=/home/kafka/bin/kafka-server-start.sh /home/kafka/config/server.properties
ExecStop=/home/kafka/bin/kafka-server-stop.sh
Restart=on-abnormal

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

Edit the `server.properties` file and add/modify the following settings:

```
vi /home/kafka/config/server.properties
```

```
[kafka@databaseksz1 ~]$ vi /home/kafka/config/server.properties
[kafka@databaseksz1 ~]$
```

```
listeners=PLAINTEXT://:9092
```

```
log.dirs=/var/log/kafka-logs
```

```
# Timeout in ms for connecting to zookeeper
zookeeper.connection.timeout.ms=6000
listeners=PLAINTEXT://:9092
log.dirs=/var/log/kafka-logs
```

After we make changes to a unit file, we should run the '`systemctl daemon-reload`' command for the changes to take effect:

```
systemctl daemon-reload
```

```
[kafka@databaseksz1 ~]$ systemctl daemon-reload
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ===
Authentication is required to reload the systemd state.
Authenticating as: kafka
Password:
==== AUTHENTICATION COMPLETE ===
[kafka@databaseksz1 ~]$
```

Create a new directory 'kafka-logs' in the '`/var/log/`' directory on your server:

```
sudo mkdir -p /var/log/kafka-logs
```

```
chown kafka:kafka -R /var/log/kafka-logs
```

```
[kafka@databaseksz1 ~]$ sudo mkdir -p /var/log/kafka-logs
[kafka@databaseksz1 ~]$ chown kafka:kafka -R /var/log/kafka-logs
chown: changing ownership of '/var/log/kafka-logs': Operation not permitted
[kafka@databaseksz1 ~]$ sudo chown kafka:kafka -R /var/log/kafka-logs
[kafka@databaseksz1 ~]$
```

This can be useful for troubleshooting. Once that's done, start the ZooKeeper and Apache Kafka services:

```
sudo systemctl start zookeeper.service
```

```
sudo systemctl start kafka.service
```

```
[kafka@databaseksz1 ~]$ sudo systemctl start zookeeper.service
[kafka@databaseksz1 ~]$ sudo systemctl start kafka.service
[kafka@databaseksz1 ~]$
```

Enable the ZooKeeper and Apache Kafka services to automatically start on server boot:

```
systemctl enable zookeeper.service
```

```
systemctl enable kafka.service
```

```
[kafka@databaseksz1 ~]$ systemctl enable zookeeper.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ===
Authentication is required to manage system service or unit files.
Authenticating as: kafka
Password:
==== AUTHENTICATION COMPLETE ===
Created symlink from /etc/systemd/system/multi-user.target.wants/zookeeper.service to /usr/lib/systemd/system/zookeeper.service.
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ===
Authentication is required to reload the systemd state.
Authenticating as: kafka
Password:
==== AUTHENTICATION COMPLETE ===
[kafka@databaseksz1 ~]$ systemctl enable kafka.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ===
Authentication is required to manage system service or unit files.
Authenticating as: kafka
Password:
==== AUTHENTICATION COMPLETE ===
Created symlink from /etc/systemd/system/multi-user.target.wants/kafka.service to /etc/systemd/system/kafka.service.
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ===
Authentication is required to reload the systemd state.
Authenticating as: kafka
Password:
==== AUTHENTICATION COMPLETE ===
[kafka@databaseksz1 ~]$
```

In order to check if ZooKeeper and Kafka services are up and running, run the following commands on the VPS:

`systemctl status zookeeper.service`

We should receive an output similar to this:

```
[kafka@databaseksz1 ~]$ systemctl status zookeeper.service
● zookeeper.service
   Loaded: loaded (/usr/lib/systemd/system/zookeeper.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2022-05-26 06:25:47 UTC; 2min 46s ago
 Main PID: 94796 (java)
   CGroup: /system.slice/zookeeper.service
           └─94796 java -Xmx512M -Xms512M -server -XX:+UseG1GC -XX:MaxGCPauseMillis=20 -XX:InitiatingHeap

May 26 06:25:59 databaseksz1 zookeeper-server-start.sh[94796]: [2022-05-26 06:25:59,734] INFO Creating ne
May 26 06:25:59 databaseksz1 zookeeper-server-start.sh[94796]: [2022-05-26 06:25:59,744] INFO Established
May 26 06:25:59 databaseksz1 zookeeper-server-start.sh[94796]: [2022-05-26 06:25:59,768] INFO Got user-le
May 26 06:25:59 databaseksz1 zookeeper-server-start.sh[94796]: [2022-05-26 06:25:59,775] INFO Got user-le
May 26 06:25:59 databaseksz1 zookeeper-server-start.sh[94796]: [2022-05-26 06:25:59,781] INFO Got user-le
May 26 06:25:59 databaseksz1 zookeeper-server-start.sh[94796]: [2022-05-26 06:25:59,808] INFO Got user-le
May 26 06:25:59 databaseksz1 zookeeper-server-start.sh[94796]: [2022-05-26 06:25:59,965] INFO Got user-le
May 26 06:25:59 databaseksz1 zookeeper-server-start.sh[94796]: [2022-05-26 06:25:59,993] INFO Got user-le
May 26 06:26:00 databaseksz1 zookeeper-server-start.sh[94796]: [2022-05-26 06:26:00,047] INFO Got user-le
May 26 06:26:00 databaseksz1 zookeeper-server-start.sh[94796]: [2022-05-26 06:26:00,048] INFO Got user-le
Hint: Some lines were ellipsized, use -l to show in full.
```

`systemctl status kafka.service`

The output of this command should be similar to this one:

```
[kafka@databaseksz1 ~]$ systemctl status kafka.service
● kafka.service
   Loaded: loaded (/etc/systemd/system/kafka.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2022-05-26 06:25:58 UTC; 3min 1s ago
 Main PID: 95074 (java)
   CGroup: /system.slice/kafka.service
           └─95074 java -Xmx1G -Xms1G -server -XX:+UseG1GC -XX:MaxGCPauseMillis=20 -XX:InitiatingHeapOccupancyPercent=35 -XX:+Di

May 26 06:26:00 databaseksz1 kafka-server-start.sh[95074]: [2022-05-26 06:26:00,008] INFO [Group Metadata Manager on Broker 0]:
May 26 06:26:00 databaseksz1 kafka-server-start.sh[95074]: [2022-05-26 06:26:00,022] INFO Will not load MX4J, mx4j-tools.jar is
May 26 06:26:00 databaseksz1 kafka-server-start.sh[95074]: [2022-05-26 06:26:00,045] INFO Creating /brokers/ids/0 (is it secure?
May 26 06:26:00 databaseksz1 kafka-server-start.sh[95074]: [2022-05-26 06:26:00,045] INFO New leader is 0 (kafka.server.Zookeepe
May 26 06:26:00 databaseksz1 kafka-server-start.sh[95074]: [2022-05-26 06:26:00,052] INFO Result of znode creation is: OK (kafka
May 26 06:26:00 databaseksz1 kafka-server-start.sh[95074]: [2022-05-26 06:26:00,057] INFO Registered broker 0 at path /brokers/i
May 26 06:26:00 databaseksz1 kafka-server-start.sh[95074]: [2022-05-26 06:26:00,057] WARN No meta.properties file under dir /var
May 26 06:26:00 databaseksz1 kafka-server-start.sh[95074]: [2022-05-26 06:26:00,074] INFO Kafka version : 0.10.1.0 (org.apache.k
May 26 06:26:00 databaseksz1 kafka-server-start.sh[95074]: [2022-05-26 06:26:00,074] INFO Kafka commitId : 3402a74efb23d1d4 (org
May 26 06:26:00 databaseksz1 kafka-server-start.sh[95074]: [2022-05-26 06:26:00,074] INFO [Kafka Server 0], started (kafka.serve
Hint: Some lines were ellipsized, use -l to show in full.
```

We can also use the `netstat` command to check if Kafka and ZooKeeper services are listening on ports 9092 and 2181 respectively:

```
sudo netstat -tunlp | grep -e \:9092 -e \:2181
```

Kafka Versions:

`Latest versions:` https://kafka.apache.org/downloads

`Old versions:` https://archive.apache.org/dist/kafka/