

Install and Configure Apache Kafka on Ubuntu 16.04

Introduction

Apache Kafka is an open-source scalable and high-throughput messaging system developed by the Apache Software Foundation written in Scala. Apache Kafka is specially designed to allow a single cluster to serve as the central data backbone for a large environment. It has a much higher throughput compared to other message brokers systems like ActiveMQ and RabbitMQ. It is capable of handling large volumes of real-time data efficiently. You can deploy Kafka on single Apache server or in a distributed clustered environment.

Features

The general features of Kafka are as follows :

1. Persist message on disk that provide constant time performance.
2. High throughput with disk structures that supporting hundreds of thousands of messages per second.
3. Distributed system scales easily with no downtime.
4. Supports multi-subscribers and automatically balances the consumers during failure.

This tutorial shows how to install and configure Apache Kafka on a Ubuntu 16.04 server.

Requirements

- A Ubuntu 16.04 server.
- Non-root user account with sudo privilege set up on your server.

Getting Started

Let's start making sure that your Ubuntu 16.04 server is fully up to date.

You can update your server by running the following command:

```
sudo apt-get update -y
sudo apt-get upgrade -y
```

Installing Java

Before installing Kafka, you will need to install Java on your system. You can install Oracle JDK 8 using the Webupd8 team PPA repository.

To add the repository, run the following command:

```
sudo add-apt-repository -y ppa:webupd8team/java
```

You should see the following output:

```
gpg: keyring `/tmp/tmpkjr4mn4m/secring.gpg' created
gpg: keyring `/tmp/tmpkjr4mn4m/pubring.gpg' created
gpg: requesting key EEA14886 from hkp server keyserver.ubuntu.com
gpg: /tmp/tmpkjr4mn4m/trustdb.gpg: trustdb created
gpg: key EEA14886: public key "Launchpad VLC" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:         imported: 1   (RSA: 1)
OK
```

Next, update the metadata of the new repository by running the following command:

```
sudo apt-get update
```

Once you have finished, run the following command to install JDK 8:

```
sudo apt-get install oracle-java8-installer -y
```

You can also verify that JDK 8 is installed properly by running the following command:

```
sudo java -version
```

You should see the output something like this:

```
java version "1.8.0_66"
Java(TM) SE Runtime Environment (build 1.8.0_66-b17)
Java HotSpot(TM) 64-Bit Server VM (build 25.66-b17, mixed mode)
```

Install ZooKeeper

Before installing Apache Kafka, you will need to have zookeeper available and running. ZooKeeper is an open source service for maintaining configuration information, providing distributed synchronization, naming and providing group services.

By default ZooKeeper package is available in Ubuntu's default repository, you can install it by running the following command:

```
sudo apt-get install zookeeperd
```

Once installation is finished, it will be started as a daemon automatically. By default ZooKeeper will run on port 2181.

You can test it by running the following command:

```
netstat -ant | grep :2181
```

If everything's fine, you should see the following Output:

```
tcp6      0      0 :::2181      :::*      LISTEN
```

Install and Start Kafka Server

Now that Java and ZooKeeper are installed, it is time to download and extract Kafka from Apache website. You can use wget to download Kafka:

```
wget http://mirror.fibergrid.in/apache/kafka/0.10.0.1/kafka_2.10-0.10.0.1.tgz
```

Next, create a directory for Kafka installation:

```
sudo mkdir /opt/Kafka
cd /opt/Kafka
```

Extract the downloaded archive using tar command in /opt/Kafka:

```
sudo tar -xvf kafka_2.10-0.10.0.1.tgz -C /opt/Kafka/
```

The next step is to start Kafka server, you can start it by running kafka-server-start.sh script located at /opt/Kafka/kafka_2.10-0.10.0.1/bin/ directory.

```
sudo /opt/Kafka/kafka_2.10-0.10.0.1/bin/kafka-server-start.sh
/opt/Kafka/kafka_2.10-0.10.0.1/config/server.properties
```

You should see the following output, if the server has started successfully:

```
[2016-08-22 21:43:48,279] WARN No meta.properties file under dir /tmp/kafka-logs/meta.properties (kafka.server.BrokerMetadataCheckpoint)
[2016-08-22 21:43:48,516] INFO Kafka version : 0.10.0.1
(org.apache.kafka.common.utils.AppInfoParser)
[2016-08-22 21:43:48,525] INFO Kafka commitId : a7a17cdec9eaa6c5
(org.apache.kafka.common.utils.AppInfoParser)
[2016-08-22 21:43:48,527] INFO [Kafka Server 0], started
(kafka.server.KafkaServer)
[2016-08-22 21:43:48,555] INFO New leader is 0
(kafka.server.ZookeeperLeaderElector$LeaderChangeListener)
```

You can use nohup with script to start the Kafka server as a background process:

```
sudo nohup /opt/Kafka/kafka_2.10-0.10.0.1/bin/kafka-server-start.sh
/opt/Kafka/kafka_2.10-0.10.0.1/config/server.properties /tmp/kafka.log 2>&1 &
```

You now have a Kafka server running and listening on port 9092.

Testing Kafka Server

Now, it is time to verify the Kafka server is operating correctly.

To test Kafka, create a sample topic with name "testing" in Apache Kafka using the following command:

```
sudo /opt/Kafka/kafka_2.10-0.10.0.1/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic testing
```

You should see the following output:

```
Created topic "testing".
```

Now, ask Zookeeper to list available topics on Apache Kafka by running the following command:

```
sudo /opt/Kafka/kafka_2.10-0.10.0.1/bin/kafka-topics.sh --list --zookeeper localhost:2181
```

You should see the following output:

```
testing
```

Now, publish a sample messages to Apache Kafka topic called testing by using the following producer command:

```
sudo /opt/Kafka/kafka_2.10-0.10.0.1/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic testing
```

After running above command, enter some messages like "Hi how are you?" press enter, then enter another message like "Where are you?"

Now, use consumer command to check for messages on Apache Kafka Topic called testing by running the following command:

```
sudo /opt/Kafka/kafka_2.10-0.10.0.1/bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic testing --from-beginning
```

You should see the following output:

```
Hi how are you?  
Where are you?
```

With this above testing you have successfully verified that you have a valid Apache Kafka setup with Apache Zookeeper.

Summary

At this point, we have installed, configured, and tested Kafka on a Ubuntu 16.04 server. You can adapt the setup to make use of it in your production environment. To learn more about Kafka check out the [Kafka documentation](#).