Kafka Installation and Configuration

Step 1- Creating a user for Kafka

Log in to your server as your non-root sudo user, then create a user called kafka:

sudo add user kafka

Next, add the kafka user to the sudo group with the adduser command. You need these privileges to install Kafka's dependencies:

sudo add user kafka sudo

Your kafka user is now ready. Log in to the kafka account using su:

su -l kafka

Step 2 — Downloading and Extracting the Kafka Binaries

To start, create a directory in /home/kafka called Downloads to store your downloads:

mkdir ~/Downloads

Use curl to download the Kafka binaries:

curl "https://downloads.apache.org/kafka/2.8.2/kafka_2.13-2.8.2.tgz" -o ~/Downloads/kafka.tgz

Create a directory called kafka and move to this directory. You'll use this directory as the base directory of the Kafka installation:

mkdir ~/kafka && cd ~/kafka

Extract the archive you downloaded using the tar command:

tar -xvzf ~/Downloads/kafka.tgz --strip 1

Step 3 — Configuring the Kafka Server

Kafka's configuration options are specified in server.properties. Open this file with nano or your favorite editor:

nano ~/kafka/config/server.properties

First, add a setting that will allow you to delete Kafka topics. Add the following line to the bottom of the file:

delete.topic.enable = true

Second, you'll change the directory where the Kafka logs are stored by modifying the log.dirs property. Find the log.dirs property and replace the existing route with the highlighted route:

log.dirs=/home/kafka/logs

Step 4 — Creating systemd Unit Files and Starting the Kafka Server

Create the unit file for zookeeper:

sudo nano /etc/systemd/system/zookeeper.service

Enter the following unit definition into the file:

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

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

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

After adding this content, save and close the file.

Next, create the systemd service file for `kafka`:

sudo nano /etc/systemd/system/kafka.service

Enter the following unit definition into the file:

```
[Unit]
Requires=zookeeper.service
After=zookeeper.service

[Service]
Type=simple
User=kafka
ExecStart=/bin/sh -c '/home/kafka/kafka/bin/kafka-server-start.sh
/home/kafka/kafka/config/server.properties > /home/kafka/kafka/kafka.log
2>&1'
ExecStop=/home/kafka/kafka/bin/kafka-server-stop.sh
Restart=on-abnormal

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

Save and close the file.

Now that you have defined the units, start Kafka with the following command:

sudo systemctl start kafka

To ensure that the server has started successfully, check the journal logs for the `kafka` unit:

sudo systemctl status kafka

You will receive output like this:

Output

● kafka.service

Loaded: loaded (/etc/systemd/system/kafka.service; disabled; vendor preset>
Active: active (running) since Wed 2023-02-01 23:44:12 UTC; 4s ago
Main PID: 17770 (sh)
Tasks: 69 (limit: 4677)
Memory: 321.9M
CGroup: /system.slice/kafka.service
├─17770 /bin/sh -c /home/kafka/kafka/bin/kafka-server-start.sh /ho>
└─17793 java -Xmx1G -Xms1G -server -XX:+UseG1GC -XX:MaxGCPauseMill>

You have started the `kafka` service. But if you reboot your server, Kafka will not restart automatically. To enable the `kafka` service on server boot, run the following command:

sudo systemctl enable zookeeper

You'll receive a response that a symlink was created:

Output
Created symlink /etc/systemd/system/multi-user.target.wants/zookeeper.service → /etc/systemd/system/zookeeper.service.

Then run this command:

sudo systemctl enable kafka

You'll receive a response that a symlink was created:

Output
Created symlink /etc/systemd/system/multi-user.target.wants/kafka.service → /etc/systemd/system/kafka.service.

Step 5 — Testing the Kafka Installation

To begin, create a topic named `TutorialTopic`:

~/kafka/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic TutorialTopic

You'll receive a response that the topic was created:

Output
Created topic TutorialTopic.

Now publish the string `"Hello, World"` to the `TutorialTopic` topic:

echo "Hello, World" | ~/kafka/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic TutorialTopic > /dev/null

Next, create a Kafka consumer using the `kafka-console-consumer.sh` script. It expects the ZooKeeper server's hostname and port, along with a topic name, as arguments. The following command consumes messages from `TutorialTopic`. Note the use of the `--from-beginning` flag, which allows the consumption of messages that were published before the consumer was started:

~/kafka/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic TutorialTopic --from-beginning

If there are no configuration issues, you will receive a `Hello, World` response in your terminal:

Output
Hello, World

The script will continue to run, waiting for more messages to publish. To test this, open a new terminal window and log in to your server. Remember to log in as your `kafka` user:

su -l kafka

In this new terminal, start a producer to publish a second message:

echo "Hello World from Sammy at DigitalOcean!" | ~/kafka/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic TutorialTopic > /dev/null

This message will load in the consumer's output in your original terminal:

Output
Hello, World
Hello World from Sammy at DigitalOcean!

Step 6 — Hardening the Kafka Server

Remove the `kafka` user from the sudo group:

sudo deluser kafka sudo

To further improve your Kafka server's security, lock the `kafka` user's password using the `passwd` command. This action ensures that nobody can directly log into the server using this account:

sudo passwd kafka -l

The `-l` flag locks the command to change a user's password (`passwd`).

At this point, only `root` or a `sudo` user can log in as `kafka` with the following command:

sudo su – kafka

In the future, if you want to unlock the ability to change the password, use `passwd` with the `-u` option:

sudo passwd kafka -u

Step 7 — Installing KafkaT (Optional)

Install Ruby and the build-essential package using apt:

sudo apt install ruby ruby-dev build-essential

You can now install KafkaT with the `gem` command:

sudo CFLAGS=-Wno-error=format-overflow gem install kafkat

When the installation has finished, you'll receive a response that it is done:

Output
...
Done installing documentation for json, colored, retryable, highline, trollop, zookeeper, zk, kafkat after 3 seconds
8 gems installed

Create a new file called `.kafkatcfg`:

nano ~/.kafkatcfg

Add the following lines to specify the required information about your Kafka server and Zookeeper instance:

~/.kafkatcfg

```
{
  "kafka_path": "~/kafka",
  "log_path": "/home/kafka/logs",
  "zk_path": "localhost:2181"
}
```

To view details about all Kafka partitions, try running this command:

kafkat partitions

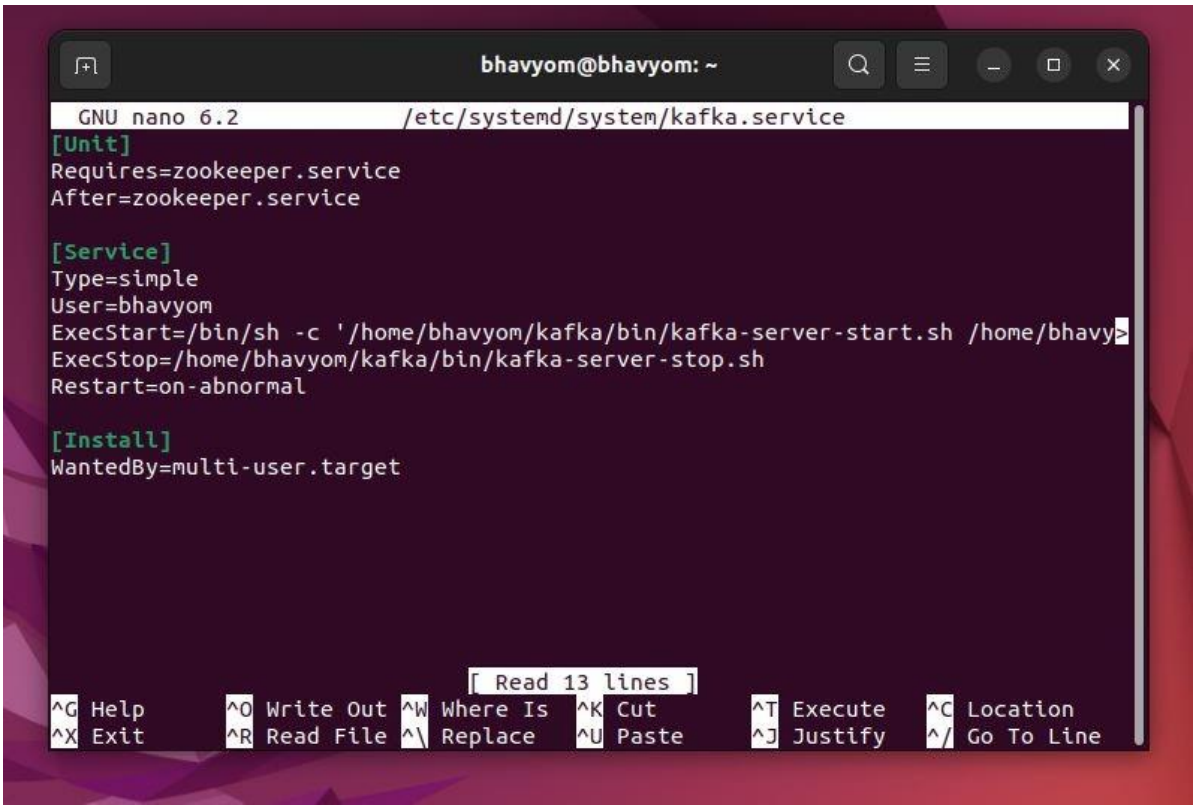
You will receive the following output:

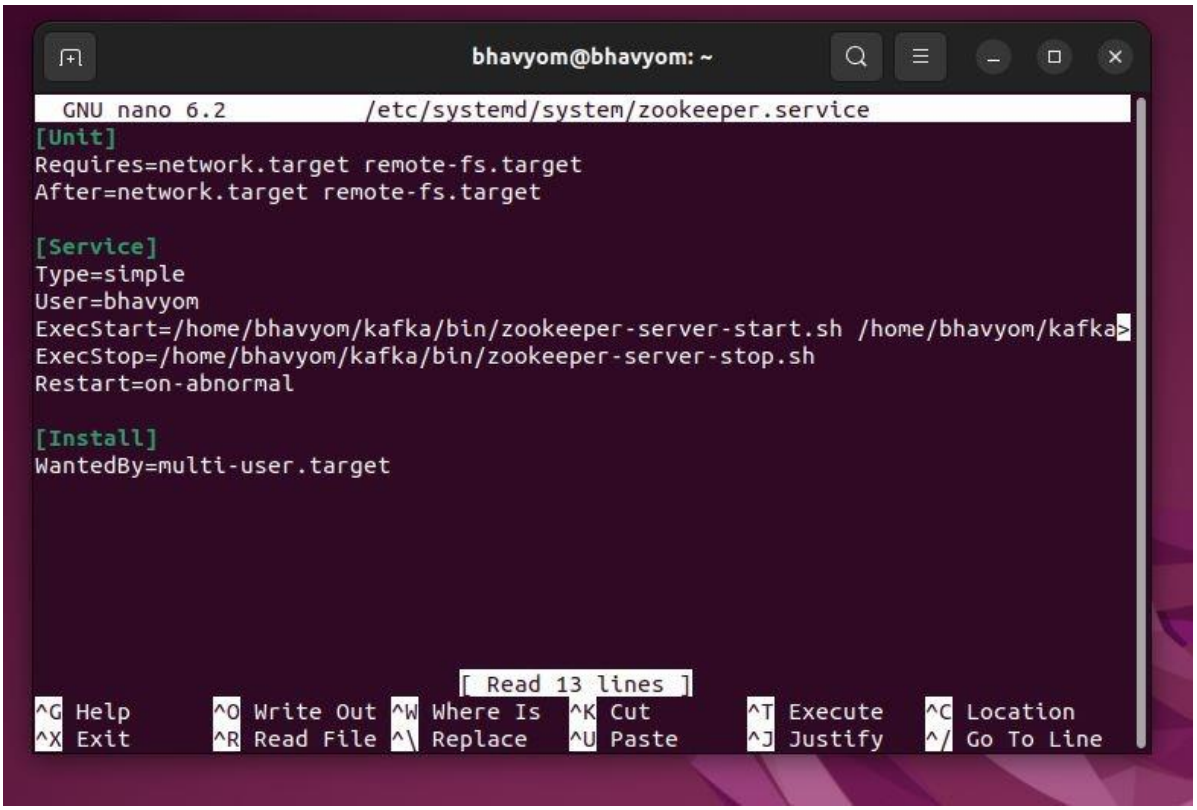
Output
[DEPRECATION] The trollop gem has been renamed to optimist and will no longer be supported. Please switch to optimist as soon as possible.
/var/lib/gems/2.7.0/gems/json-1.8.6/lib/json/common.rb:155: warning: Using the last argument as keyword parameters is deprecated
...
Topic Partition Leader Replicas ISRs
TutorialTopic 0 0 [0] [0]
__consumer_offsets 0 0 [0]
[0]
...
...

The output will include `TutorialTopic` and `__consumer_offsets`, an internal topic used by Kafka for storing client-related information. You can safely ignore lines starting with `__consumer_offsets`.

Conclusion

You now have Apache Kafka running securely on your Ubuntu server. You can integrate Kafka into your favorite programming language using Kafka clients.


bhavyom@bhavyom: ~
GNU nano 6.2 /etc/systemd/system/kafka.service
[Unit]
Requires=zookeeper.service
After=zookeeper.service
[Service]
Type=simple
User=bhavyom
ExecStart=/bin/sh -c '/home/bhavyom/kafka/bin/kafka-server-start.sh /home/bhavy>
ExecStop=/home/bhavyom/kafka/bin/kafka-server-stop.sh
Restart=on-abnormal
[Install]
WantedBy=multi-user.target
[Read 13 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line


bhavyom@bhavyom: ~
GNU nano 6.2 /etc/systemd/system/zookeeper.service
[Unit]
Requires=network.target remote-fs.target
After=network.target remote-fs.target
[Service]
Type=simple
User=bhavyom
ExecStart=/home/bhavyom/kafka/bin/zookeeper-server-start.sh /home/bhavyom/kafka>
ExecStop=/home/bhavyom/kafka/bin/zookeeper-server-stop.sh
Restart=on-abnormal
[Install]
WantedBy=multi-user.target
[Read 13 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line