

How to Install Apache Kafka on Ubuntu 18.04 & 16.04

Written by [Rahul](#)

🔍 Search Now

Grammarly

Free Grammar Checker



Apache Kafka is a distributed streaming platform. It is useful for building real-time streaming data pipelines to get data between the systems or applications. Another useful feature is real-time streaming applications that can transform streams of data or react on a stream of data. This tutorial will help you to install Apache Kafka on Ubuntu 19.10, 18.04 & 16.04 systems.

Step 1 – Install Java

Free Grammar Checker

Grammarly

Improve grammar, word choice, and sentence structure in your writing with Grammarly.

DOWNLOAD

POPULAR POSTS



How to Install GIMP 2.10 or Ubuntu 20.04 [↑](#)

3

How to Install NetBeans on Ubuntu 20.04

Apache Kafka required Java to run. You must have java installed on your system. Execute below command to install default OpenJDK on your system from the official PPA's. You can also install the specific version of from [here](#).

```
$ sudo apt update
$ sudo apt install default-jdk
```

Step 2 – Download Apache Kafka

Download the Apache Kafka binary files from its official [download](#) website. You can also select any nearby mirror to download.

```
$ wget http://www-us.apache.org/dist/kafka/2.4.0/kafka_2.13-2.4.0.tgz
```

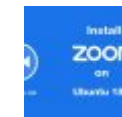
Then extract the archive file

```
$ tar xzf kafka_2.13-2.4.0.tgz
$ mv kafka_2.13-2.4.0 /usr/local/kafka
```

Step 3 – Setup Kafka Systemd Unit Files



2



How to Install Zoom on Ubuntu
18.04

0



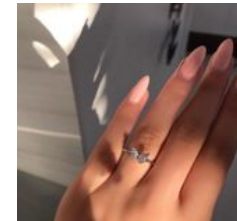
How to Install Docker on Ubuntu
20.04

2

How to Install Dovecot (POP/IMAP) on
Debian 10

0

A Shiny
Surprise



A Shiny Surprise

Your Online Fashion Store



buy now

Next, to create a Kafka systemd unit file using the following command:

```
$ vim /etc/systemd/system/kafka.service
```

Add the below content. Make sure to set the correct **JAVA_HOME** path as per the Java installed on your system.

```
[Unit]
Description=Apache Kafka Server
Documentation=http://kafka.apache.org/documentation.html
Requires=zookeeper.service

[Service]
Type=simple
Environment="JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64"
ExecStart=/usr/local/kafka/bin/kafka-server-start.sh /usr/local/kafka/config
ExecStop=/usr/local/kafka/bin/kafka-server-stop.sh

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

Save file and close.



Reload the systemd daemon to apply new changes.

```
$ systemctl daemon-reload
```

Step 4 – Start Kafka Server

Kafka required ZooKeeper so first, start a ZooKeeper server on your system. You can use the script available with Kafka to get start a single-node ZooKeeper instance.

```
$ sudo systemctl start zookeeper
```

Now start the Kafka server and view the running status:

```
$ sudo systemctl start kafka  
$ sudo systemctl status kafka
```



Next, create systemd unit files for the Zookeeper and Kafka service. This will help to manage Kafka services to start/stop using the systemctl command.

First, create systemd unit file for Zookeeper with below command:

```
$ vim /etc/systemd/system/zookeeper.service
```



Add below content:

```
[Unit]
Description=Apache Zookeeper server
Documentation=http://zookeeper.apache.org
Requires=network.target remote-fs.target
After=network.target remote-fs.target

[Service]
Type=simple
ExecStart=/usr/local/kafka/bin/zookeeper-server-start.sh /usr/local/kafka/conf/zoo.cfg
ExecStop=/usr/local/kafka/bin/zookeeper-server-stop.sh
Restart=on-abnormal


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


Save the file and close it.

Big Promotion on PCB Prototype

Fast and Affordable &
Your PCB Fabrication
Assembly by PCBGC





```
root@tecadmin: ~  
root@tecadmin:~# systemctl status kafka  
● kafka.service - Apache Kafka Server  
   Loaded: loaded (/etc/systemd/system/kafka.service; enabled; vendor preset: enabled)  
   Active: active (running) since Wed 2020-03-04 07:23:45 UTC; 25min ago  
     Docs: http://kafka.apache.org/documentation.html  
    Main PID: 8917 (java)  
      Tasks: 67 (limit: 3571)  
    CGroup: /system.slice/kafka.service  
            └─8917 /usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -Xmx1G -Xms1G -server -XX  
  
Mar 04 07:28:46 tecadmin kafka-server-start.sh[8917]: [2020-03-04 07:28:46,295] INFO [Group  
Mar 04 07:28:46 tecadmin kafka-server-start.sh[8917]: [2020-03-04 07:28:46,341] INFO [Group  
Mar 04 07:28:46 tecadmin kafka-server-start.sh[8917]: [2020-03-04 07:28:46,370] INFO [Group  
Mar 04 07:28:46 tecadmin kafka-server-start.sh[8917]: [2020-03-04 07:28:46,392] INFO [Group  
Mar 04 07:33:51 tecadmin kafka-server-start.sh[8917]: [2020-03-04 07:33:51,761] INFO [Group  
Mar 04 07:42:26 tecadmin kafka-server-start.sh[8917]: [2020-03-04 07:42:26,655] INFO [Group  
Mar 04 07:42:26 tecadmin kafka-server-start.sh[8917]: [2020-03-04 07:42:26,662] INFO [Group  
Mar 04 07:42:26 tecadmin kafka-server-start.sh[8917]: [2020-03-04 07:42:26,669] INFO [Group  
Mar 04 07:43:51 tecadmin kafka-server-start.sh[8917]: [2020-03-04 07:43:51,750] INFO [Group  
Mar 04 07:43:51 tecadmin kafka-server-start.sh[8917]: [2020-03-04 07:43:51,755] INFO [Group  
lines 1-19/19 (END)
```

All done. The Kafka installation has been successfully completed. The part of this tutorial will help you to work with the Kafka server.

Step 5 – Create a Topic in Kafka

Kafka provides multiple pre-built shell script to work on it. First, create a topic named "testTopic" with a single partition with single replica:

```
$ cd /usr/local/kafka  
$ bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 testTopic  
  
Created topic testTopic.
```





Big Promotion on PCB Prototype

PCBgogo

Good Chance to join
PCBgogo to get coupons and
souvenirs.

The replication-factor describes how many copies of data will be created. As we are running with a single instance keep this value 1.

Set the partitions options as the number of brokers you want your data to be split between. As we are running with a single broker keep this value 1.

You can create multiple topics by running the same command as above. After that, you can see the created topics on Kafka by the running below command:

```
$ bin/kafka-topics.sh --list --zookeeper localhost:2181
```

```
testTopic  
TecAdminTutorial1  
TecAdminTutorial2
```



Alternatively, instead of manually creating topics you can also configure your brokers to auto-create topics when a non-existent topic is published to.

Step 6 – Send Messages to Kafka

The “producer” is the process responsible for put data into our Kafka. The Kafka comes with a command-line client that will take input from a file or from standard input and send it out as messages to the Kafka cluster. The default Kafka sends each line as a separate message.

Let’s run the producer and then type a few messages into the console to send to the server.

```
$ bin/kafka-console-producer.sh --broker-list localhost:9092 --topic testTop

>Welcome to kafka
>This is my first topic
>
```

You can exit this command or keep this terminal running for further testing. Now open a new terminal to the Kafka consumer process on the next step.



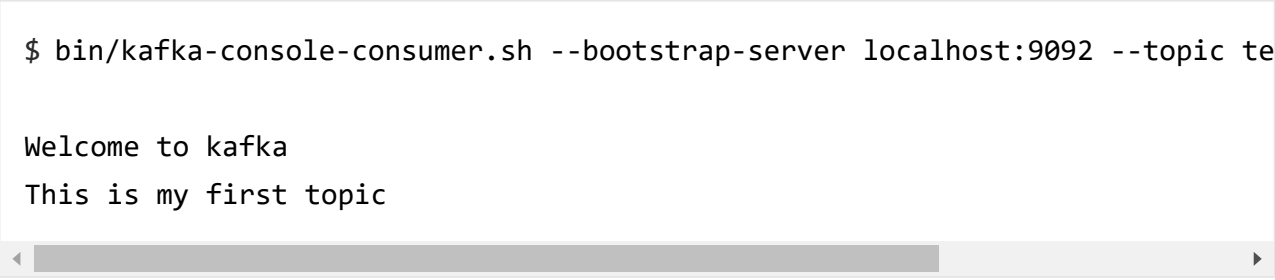
Step 7 – Using Kafka Consumer

Kafka also has a command-line consumer to read data from the Kafka cluster and display messages to standard output.

```
$ bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic te
```

```
Welcome to kafka
```

```
This is my first topic
```



Now, If you have still running Kafka producer (Step #6) in another terminal. Just type some text on that producer terminal. it will immediately visible on consumer terminal. See the below screenshot of Kafka producer and consumer in working:



Conclusion

You have successfully installed and configured the Kafka service on your Ubuntu Linux machine.



SHARE IT!**RAHUL**

I, Rahul Kumar am the founder and chief editor of TecAdmin.net. I am a Red Hat Certified Engineer (RHCE) and working as an IT professional since 2009..

**9 COMMENTS****SIMON**

September 13, 2019 at 2:57 pm

<https://tecadmin.net/install-apache-kafka-ubuntu/>



REPLY TO SIMON TO SIMON" ARIA-LABEL='↩ REPLY TO SIMON TO SIMON'>↩ REPLY TO SIMON





IVAN

June 21, 2020 at 11:45 am

REPLY TO IVAN TO IVAN" ARIA-LABEL='↩ REPLY TO IVAN TO IVAN'>↩ REPLY TO IVAN



TANVEER KHAN

September 6, 2019 at 11:11 am

REPLY TO TANVEER TO TANVEER KHAN" ARIA-LABEL='↩ REPLY TO TANVEER TO TANVEER KHAN'>↩ REPLY TO TANVEER



**ZAHID**

August 17, 2019 at 10:54 am

REPLY TO ZAHID TO ZAHID" ARIA-LABEL='↩ REPLY TO ZAHID TO ZAHID'>↩ REPLY TO ZAHID

**KALAIKKOVAN**

July 5, 2019 at 5:28 am

REPLY TO KALAIKKOVAN TO KALAIKKOVAN" ARIA-LABEL='↩ REPLY TO KALAIKKOVAN TO KALAIKKOVAN'>↩ REPLY TO KALAIKKOVAN



ABDUL

June 19, 2019 at 3:10 pm



REPLY TO ABDUL TO ABDUL" ARIA-LABEL='↩ REPLY TO ABDUL TO ABDUL'>↩ REPLY TO ABDUL



DANIEL OPIYO

June 16, 2019 at 4:04 pm

REPLY TO DANIEL TO DANIEL OPIYO" ARIA-LABEL='↩ REPLY TO DANIEL TO DANIEL OPIYO'>↩ REPLY TO DANIEL



ANIL KUMAR PANIGRAHI

May 8, 2019 at 11:48 am

REPLY TO ANIL TO ANIL KUMAR PANIGRAHI" ARIA-LABEL='↩ REPLY TO ANIL TO ANIL KUMAR PANIGRAHI'>↩ REPLY TO ANIL



**CHRIS**

June 20, 2018 at 7:37 am

REPLY TO CHRIS TO CHRIS" ARIA-LABEL='↩ REPLY TO CHRIS TO CHRIS'>↩ REPLY TO CHRIS

LEAVE A REPLY**COMMENTS *****NAME *****EMAIL *****WEBSITE**

☐ **SAVE MY NAME, EMAIL, AND WEBSITE IN THIS BROWSER FOR THE NEXT TIME I COMMENT.**



Je ne suis pas un robot

reCAPTCHA
Confidentialité - Conditions

SUBMIT

© 2013-2020 [Tecadmin.net](#). All Rights Reserved | [Terms](#) | [Privacy Policy](#)

