

Apache Kafka 4.1.1 – Single Node KRaft Mode on CentOS 8 (No ZooKeeper)

Apache Kafka 4.1.1 – Single Node KRaft Mode on CentOS 8 (No ZooKeeper)

This guide shows how to install and run **Apache Kafka 4.1.1** in **KRaft mode** (no ZooKeeper) on **CentOS 8**, using a **single-node broker + controller** setup.

1. Install Java 17 (Required for Kafka 4.x)

Kafka 4.x requires **Java 17+**.

```
sudo dnf install -y java-17-openjdk java-17-openjdk-devel
```

Verify:

```
java -version
```

You should see something like:

```
openjdk version "17.x.x" ...
```

2. Create a Dedicated Kafka User

```
sudo useradd kafka -m  
sudo usermod -s /bin/bash kafka
```

3. Download & Install Kafka 4.1.1

From your home directory:

```
cd ~  
wget https://dlcdn.apache.org/kafka/4.1.1/kafka_2.13-4.1.1.tgz  
tar -xvzf kafka_2.13-4.1.1.tgz
```

Move Kafka to `/opt` and set permissions:

```
sudo mv kafka_2.13-4.1.1 /opt/kafka  
sudo chown -R kafka:kafka /opt/kafka
```

4. Create Data Directory

```
sudo mkdir -p /opt/kafka/data  
sudo chown -R kafka:kafka /opt/kafka
```

5. Configure Kafka for Single-Node KRaft

Edit the main Kafka config file:

```
sudo nano /opt/kafka/config/server.properties
```

Replace the contents with the **working KRaft config**:

```
process.roles=broker,controller
node.id=1
controller.listener.names=CONTROLLER
controller.quorum.voters=1@localhost:9093

listeners=PLAINTEXT://:9092,CONTROLLER://:9093
advertised.listeners=PLAINTEXT://localhost:9092
inter.broker.listener.name=PLAINTEXT

num.partitions=1
offsets.topic.replication.factor=1
transaction.state.log.replication.factor=1
transaction.state.log.min.isr=1

log.dirs=/opt/kafka/data
```

Explanation (short):

- `process.roles=broker,controller` → single process runs both broker + controller.
- `controller.quorum.voters=1@localhost:9093` → 1-node KRaft quorum.
- `listeners` / `advertised.listeners` → Kafka is reachable at `localhost:9092`.
- `offsets.topic.*` and `transaction.state.*` → required so internal topics (like `__consumer_offsets`) can be created.
- `log.dirs` → where Kafka stores data.

6. Generate KRaft Cluster UUID

Run as **any user** (root or your normal user is fine):

```
UUID=$(/opt/kafka/bin/kafka-storage.sh random-uuid)
echo $UUID
```

Copy the printed UUID.

7. Format the KRaft Storage

This initializes the KRaft metadata directory.

```
sudo -u kafka /opt/kafka/bin/kafka-storage.sh format -t $UUID -c
/opt/kafka/config/server.properties
```

Expected output includes something like:

```
Formatting metadata directory /opt/kafka/data with metadata.version 4.1-IV1.
```

8. (Optional) SELinux Configuration

On CentOS 8, SELinux may block Kafka when installed under `/opt`.

Check mode:

```
getenforce
```

If it shows `Enforcing`, you can temporarily relax it:

```
sudo setenforce 0
```

For a permanent setting (optional, depends on your security policy):

```
sudo nano /etc/selinux/config
# Set:
SELINUX=permissive
```

Then reboot later to apply permanently.

9. Test Kafka Manually (Foreground)

Run Kafka as the `kafka` user:

```
sudo -u kafka /opt/kafka/bin/kafka-server-start.sh /opt/kafka/config/server.properties
```

If all is correct, you should see logs ending with something like:

```
Kafka Server started in KRaft mode
```

Press **Ctrl + C** to stop Kafka.

10. Create a Systemd Service for Kafka

Create a systemd unit:

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

Paste:

```
[Unit]
Description=Kafka KRaft Server
After=network.target

[Service]
User=kafka
Environment="JAVA_HOME=/usr/lib/jvm/java-17-openjdk"
Environment="PATH=/usr/lib/jvm/java-17-openjdk/bin:/usr/bin:/bin"
ExecStart=/opt/kafka/bin/kafka-server-start.sh /opt/kafka/config/server.properties
ExecStop=/opt/kafka/bin/kafka-server-stop.sh
Restart=on-failure
LimitNOFILE=100000

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

Reload systemd and start Kafka:

```
sudo systemctl daemon-reload
sudo systemctl start kafka
sudo systemctl enable kafka
sudo systemctl status kafka
```

Status should show:

```
Active: active (running)
```

11. Verify Data Directory & Internal Topics

Check Kafka's data directory:

```
ls -l /opt/kafka/data
```

You should eventually see directories like:

- `__cluster_metadata-0`
- `__consumer_offsets-0`
- `test-topic-0` (after you create a topic)

The presence of `__consumer_offsets-0` confirms that Kafka successfully auto-created the internal offsets topic required for consumer groups.

12. Create and Test a Topic

12.1 Create a Topic

```
/opt/kafka/bin/kafka-topics.sh --create --topic test-topic --bootstrap-server localhost:9092
```

List topics:

```
/opt/kafka/bin/kafka-topics.sh --list --bootstrap-server localhost:9092
```

Expected:

```
test-topic
```

12.2 Run a Producer

```
/opt/kafka/bin/kafka-console-producer.sh --topic test-topic --bootstrap-server localhost:9092
```

Type a few messages and press **ENTER** after each:

```
hi
hello
this is a test
```

12.3 Run a Consumer

In another terminal:

```
/opt/kafka/bin/kafka-console-consumer.sh --topic test-topic --bootstrap-server
localhost:9092 --from-beginning
```

You should see:

```
hi
hello
this is a test
```

If you want to test with a new consumer group:

```
/opt/kafka/bin/kafka-console-consumer.sh --topic test-topic --bootstrap-server
localhost:9092 --group debug-group-1 --from-beginning
```

13. Debugging Tips

13.1 Consumer Shows No Messages

Check:

```
ls -l /opt/kafka/data
```

If `__consumer_offsets-0` is missing:

1. Ensure these properties exist in `server.properties`:

```
num.partitions=1
offsets.topic.replication.factor=1
transaction.state.log.replication.factor=1
transaction.state.log.min.isr=1
```

2. Restart Kafka:

```
sudo systemctl restart kafka
```

3. Check logs:

```
journalctl -u kafka -n 200 --no-pager
```

13.2 Check Consumer Groups

```
/opt/kafka/bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list
```

Describe a group:

```
/opt/kafka/bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --group debug-group-1 --describe
```

14. Summary

You now have:

- **Kafka 4.1.1 installed** on CentOS 8
- Running in **single-node KRaft mode** (no ZooKeeper)
- Proper `server.properties` for KRaft
- Kafka managed by **systemd**
- Verified producer and consumer flow on topic `test-topic`

This setup is ideal for **development, POC, and local testing** without ZooKeeper.

Check Version

```
opt/kafka/bin/kafka-storage.sh version
```

Revision #3

Created 25 November 2025 10:21:22 by Admin

Updated 26 November 2025 07:11:47 by Admin