```
&lt;p&gt;# Start a Zookeeper session
bin/zookeeper-server-start.sh config/zookeeper.properties
# Configuration file: zookeeper.properties

# Create a Kafka cluster with 3 brokers
cp config/server.properties config/server1.properties
cp config/server.properties config/server2.properties
cp config/server.properties config/server3.properties

# Edit server1.properties
echo "broker.id=1" &gt;&gt; config/server1.properties
echo "listeners=PLAINTEXT://:9092" &gt;&gt; config/server1.properties
echo "log.dirs=/tmp/kafka-logs-1" &gt;&gt; config/server1.properties

# Edit server2.properties
echo "broker.id=2" &gt;&gt; config/server2.properties
echo "listeners=PLAINTEXT://:9093" &gt;&gt; config/server2.properties
echo "log.dirs=/tmp/kafka-logs-2" &gt;&gt; config/server2.properties

# Edit server3.properties
echo "broker.id=3" &gt;&gt; config/server3.properties
echo "listeners=PLAINTEXT://:9094" &gt;&gt; config/server3.properties
echo "log.dirs=/tmp/kafka-logs-3" &gt;&gt; config/server3.properties

# Start Kafka brokers
bin/kafka-server-start.sh config/server1.properties
bin/kafka-server-start.sh config/server2.properties
bin/kafka-server-start.sh config/server3.properties

# Create topic 'test1' with 3 partitions
bin/kafka-topics.sh --create --topic test1 --partitions 3 --replication-factor 3 --bootstrap-server localhost:9092
```

```
# Create topic 'test2' with 2 partitions
bin/kafka-topics.sh --create --topic test2 --partitions 2 --replication-factor 3 --bootstrap-server localhost:9092

# List available topics and partition details
bin/kafka-topics.sh --list --bootstrap-server localhost:9092
bin/kafka-topics.sh --describe --topic test1 --bootstrap-server localhost:9092
bin/kafka-topics.sh --describe --topic test2 --bootstrap-server localhost:9092

# Create 3 Kafka console consumers for topic 'test1'
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test1 --from-beginning
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test1 --from-beginning
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test1 --from-beginning

# Create a Kafka console producer for topic 'test1'
bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test1

# Produce 10 messages to topic 'test1'
# Type each line and hit Enter
message1
message2
message3
message4
message5
message6
message7
message8
message9
message10
```

```
# Consumers will display the received messages

# Close the 3 console consumers (Ctrl+C on each consumer terminal)

# Create 3 Kafka console consumers for topic 'test1' using consumer group 'group1'
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test1 --group group1
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test1 --group group1
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test1 --group group1

# Produce 10 messages using the console producer
# Type each line and hit Enter
message11
message12
message13
message14
message15
message16
message17
message18
message19
message20

# Consumers will display the received messages in a load-balanced manner

# Create a new console producer with additional options
bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test1 --property "parse.key=true" --property "key.separator=:"

# Produce 10 keyed messages
# Type each line and hit Enter
key1:value1
key2:value2
key3:value3
key4:value4
```

```
key5:value5
key6:value6
key7:value7
key8:value8
key9:value9
key10:value10

# Consumers will display the messages with keys
# Explanation of options:
# --property &quot;parse.key=true&quot;: Enables parsing of the key from the message.
# --property &quot;key.separator=:&quot;: Specifies the separator between the key and the value in the message.
&lt;/p&gt;
```