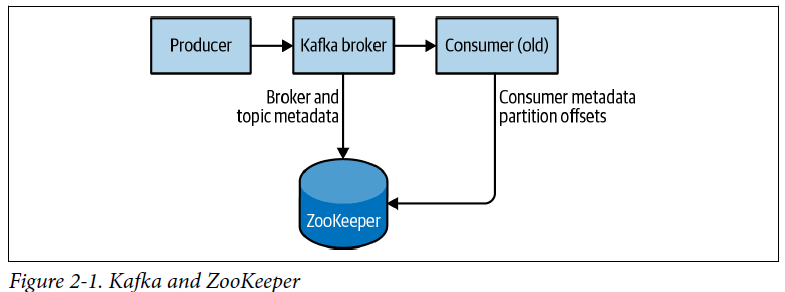
**Install Kafka**

**Install ZooKeeper**

Apache Kafka uses Apache ZooKeeper to store metadata about the Kafka cluster, as well as consumer client details, as shown in Figure 2-1. ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.



Khawja@DESKTOP-ADO10PL MINGW64 /c/kafka/bin

$ ./windows/zookeeper-server-start.bat ../config/zookeeper.properties

Starting zookeeper.. STARTED

You can now validate that ZooKeeper is running correctly in standalone mode by connecting to the client port and sending the four-letter command srvr. This will return basic ZooKeeper information from the running server:

**#** telnet localhost 2181

Trying 127.0.0.1...

Connected to localhost.

Escape character is '^]'.

srvr

Zookeeper version: 3.5.9-83df9301aa5c2a5d284a9940177808c01bc35cef, built on

01/06/2021 19:49 GMT

Latency min/avg/max: 0/0/0

Received: 1

Sent: 0

Connections: 1

Outstanding: 0

Zxid: 0x0

Mode: standalone

Node count: 5

Connection closed by foreign host.

**#**

ZooKeeper is designed to work as a cluster, called an *ensemble*, to ensure high availability. Due to the balancing algorithm used, it is recommended that ensembles contain an odd number of servers (e.g., 3, 5, and so on) as a majority of ensemble members (a *quorum*) must be working in order for ZooKeeper to respond to requests. This means that in a three-node ensemble, you can run with one node missing. With a five-node ensemble, you can run with two nodes missing.

Installing a Kafka Broker

Khawja@DESKTOP-ADO10PL MINGW64 /c/kafka/bin

$ ./windows/kafka-server-start.bat ../config/server.properties

Once the Kafka broker is started, we can verify that it is working by performing some simple operations against the cluster: creating a test topic, producing some messages, and consuming the same messages.

Creating a Kafka Topic

Khawja@DESKTOP-ADO10PL MINGW64 /c/kafka/bin

$ kafka-topics.bat --zookeeper 127.0.0.1:2181 --topic forwardingData --create --partitions 3 --replication-factor 1

Created topic "forwardingData".

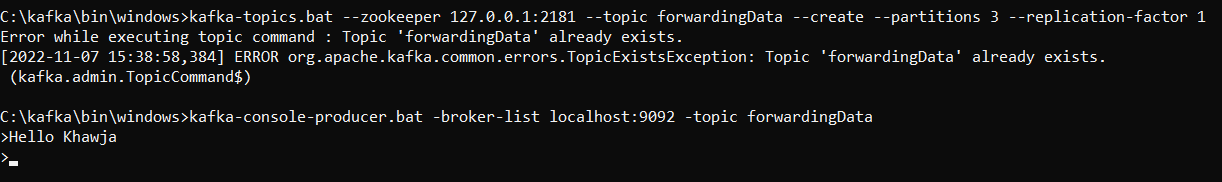
Create a Producer on topic **forwardingData**

Khawja@DESKTOP-ADO10PL MINGW64 /c/kafka/bin

$ kafka-console-producer.bat -broker-list localhost:9092 -topic forwardingData

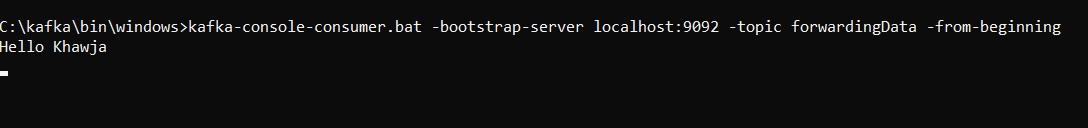
>

Producer created…



Create a Consumer

kafka-console-consumer.bat -bootstrap-server localhost:9092 -topic forwardingData -from-beginning



**Kafka Broker Parameters:**

These parameters deal with the basic configuration of the broker, and most of them must be changed to run properly in a cluster with other brokers.

1. **broker.Id**

Every Kafka broker must have an integer identifier, which is set using the broker.id configuration. By default, this integer is set to 0, but it can be any value. It is essential that the integer must be unique for each broker within a single Kafka cluster.

1. **listeners**

**listeners** - Comma-separated list of URIs we will listen on and their protocols. Specify hostname as 0.0.0.0 to bind to all interfaces. Leave hostname empty to bind to default interface. Examples of legal listener lists:

* PLAINTEXT://myhost:9092,TRACE://:9091
* PLAINTEXT://0.0.0.0:9092, TRACE://localhost:9093

**advertised.listeners** - Listeners to publish to ZooKeeper for clients to use, if different than the listeners above. In IaaS environments, this may need to be different from the interface to which the broker binds. If this is not set, the value for listeners will be used.

1. **zookeeper.connect**

The location of the ZooKeeper used for storing the broker metadata is set using the zookeeper.connect configuration parameter. The example configuration uses a Zoo‐Keeper running on port 2181 on the local host, which is specified as localhost:2181. The format for this parameter is a semicolon-separated list of hostname:port/path.

1. **log.dir**

Kafka persists all messages to disk, and these log segments are stored in the directory

specified in the log.dir configuration. For multiple directories, the config log.dirs is preferable. If this value is not set, it will default back to log.dir. log.dirs is a comma-separated list of paths on the local system. If more than one path is specified, the broker will store partitions on them in a “least-used” fashion, with one partition’s log segments stored within the same path.

1. **delete.topic.enable**

Depending on your environment and data retention guidelines, you may wish to lock down a cluster to prevent arbitrary deletions of topics. Disabling topic deletion can be set by setting this flag to false.

1. **auto.create.topics.enable**

The default Kafka configuration specifies that the broker should automatically create

a topic under the following circumstances:

• When a producer starts writing messages to the topic

• When a consumer starts reading messages from the topic

• When any client requests metadata for the topic

In many situations, this can be undesirable behavior, especially as there is no way to

validate the existence of a topic through the Kafka protocol without causing it to be

created. If you are managing topic creation explicitly, whether manually or through a

provisioning system, you can set the auto.create.topics.enable configuration to false.

1. **default.replication.factor**

If auto-topic creation is enabled, this configuration sets what the replication factor should be for new topics. Replication strategy can vary depending on the desired durability or availability of a cluster and will be discussed more in later chapters. The following is a brief recommendation if you are running Kafka in a cluster that will prevent outages due to factors outside of Kafka’s internal capabilities, such as hardware failures. It is highly recommended to set the replication factor to at least 1 above the min.insync.replicas setting. For more fault-resistant settings, if you have large enough clusters and enough hardware, setting your replication factor to 2 above the min.insync.replicas (abbreviated as RF++) can be preferable.

1. **num.partition**

The num.partitions is the default partitions for auto created topics.