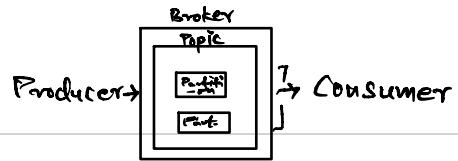


→ A bit more about Kafka:



☞ Kafka Broker: a service that is part of Apache Kafka cluster.

It is responsible for the following key tasks:

- o Message storage: the broker stores messages in topics, which are partitioned for scalability and fault tolerance.
Each partition can be replicated across multiple brokers to ensure data durability.
- o Message Management: It manages the storage and retrieval of messages by maintaining the logs and handling read and write requests from producers and consumers.
- o Coordination: Brokers coordinate with each other to ensure data consistency and reliability. They handle leader elections for partitions, manage replica sets, and track the state of consumers and producers.

☞ Kafka Cluster: is a distributed system consisting of multiple Kafka brokers. Key characteristics of a Kafka cluster include:

1. Scalability: By adding more brokers to the clusters, Kafka can handle more messages and distribute the load effectively.

2. Fault Tolerance: Data in Kafka is replicated across multiple brokers. If one broker fails, another broker with the replicated data can take over, ensuring no data loss.
3. High Throughput: Kafka clusters can handle high-throughput data streams, making them suitable for large-scale data processing and real-time analytics.
4. Distributed Architecture: A Kafka cluster ensures that the workload is distributed across multiple brokers, which can spread across different servers or data centers.

Key Components in Kafka Architecture:

1. **Topics**: category or feed name to which records are sent. Topics are partitioned and distributed across brokers.
2. **Partition**: Each topic is divided into partitions, which are the basic unit of parallelism in Kafka. Each partition is an ordered, immutable sequence of records.
3. **Producers**: Producers are clients that send records to a Kafka topic.
4. **Consumers**: Consumers are clients that read records from a Kafka topic.

5. Zookeeper: Originally, Zookeeper was used for managing and coordinating Kafka brokers.

Newer Kafka versions are moving towards KRaft (Kafka Raft) mode, eliminating the need for Zookeeper.

* In docker-compose.yml, some aspects to consider:

- ① Broker ID: Unique ID for each broker in the cluster.
- ② Zookeeper Connection: Necessary for managing broker metadata and leader election (in older versions).
- ③ Listeners: Defines the protocols and addresses for client connections.
- ④ Replication factors: Set how many copies of data are stored for fault tolerance.
- ⑤ Ports
- ⑥ Environment Variables: Specifies various operational parameters for brokers.

→ 'nc -z' Command used in healthcheck because it allows to verify service availability, lightweight check monitor critical port.

↓ ↴ -z in NetCat is used for scanning purpose, specifically used to scan for open ports without sending any data. To get more detailed output, used with combination of -z and -v (verbose).

↳ versatile networking tool that can be used for various network-related tasks such as reading from and writing to network connections using TCP or UDP.

Syntax: nc -z [hostname/IP address] [Port range]