

does broker means one more machine?

A **Kafka broker** is basically **one Kafka server instance**.

This was confusing me.

Can i add multiple brokers to same kafka cluster?

Yes ✅, you **can and should** add multiple brokers to the same Kafka cluster — that’s how Kafka achieves **scalability and fault tolerance**.

**suppose i have single topic and single broker machine? How many partitions i can do?**

Great question 👍

**If you have a single topic and only one Kafka broker (machine):?**

### 🔹 Number of partitions

* You can create **as many partitions as you want** on a single broker.
* Kafka does **not limit partitions per broker** (other than system resources like CPU, RAM, and disk).

👉 For example:

kafka-topics.sh --create \

--topic orders \

--partitions 10 \

--replication-factor 1 \

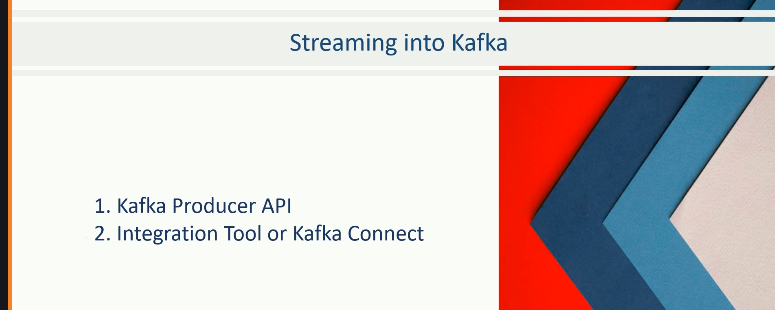
--bootstrap-server localhost:9092

This will create **10 partitions** for the orders topic, **all on the same broker**.

### 🔹 But there’s a catch 🚨

* **Replication factor** must be ≤ number of brokers.
  + With 1 broker, replication-factor=1 is the only option.
  + If you try replication-factor=2, Kafka will throw an error.
* All partitions are stored on the **same broker**, so there’s **no fault tolerance**.
  + If that broker goes down → the topic and its partitions are completely unavailable.

**Note: We can create a kafka cluster by using different machine(Machine here Brokers), we can create a number of topics and number of partitions, but replication factor depends on the number of Brokers(or Machines).**



**Kafka connect architecture:**

# Kafka Connect Architecture

+----------------+

| External System|

| (e.g. MySQL) |

+--------+-------+

|

Source Connector

|

+-------v--------+

| Kafka Topic |

+-------+--------+

|

Sink Connector

|

+--------v-------+

| External System|

| (e.g. Elastic) |

+----------------+

* **Workers** run the connectors.
* **Tasks** are parallel units of work inside a connector.
* **Configs** tell Kafka Connect what topic, DB, or file to use.

**Difference between Source Connector and Sink Connector?**

✅ In short:

* **Source = import into Kafka**
* **Sink = export from Kafka**

## 🔹 Kafka Connectors

In Kafka Connect, connectors are of **two types**:

### 1. ****Source Connector****

* **Moves data INTO Kafka** from an external system.
* Think: external system → Kafka topic.
* Examples:
  + **JDBC Source Connector** → Reads from MySQL/Postgres tables and pushes rows into Kafka topics.
  + **Twitter Source Connector** → Streams tweets into Kafka.
  + **Splunk Source Connector** (custom) → Pulls Splunk logs/events into Kafka.

📌 Analogy: A pipeline that ***fills Kafka*** with data from outside.

### 2. ****Sink Connector****

* **Moves data OUT of Kafka** to an external system.
* Think: Kafka topic → external system.
* Examples:
  + **Elasticsearch Sink Connector** → Streams Kafka events into Elasticsearch for search & analytics.
  + **JDBC Sink Connector** → Pushes Kafka events into a database table.
  + **Splunk Sink Connector** (official from Splunk) → Sends Kafka data into Splunk (via HEC).