

Kafka Brokers & Architecture

1. Kafka Broker Kya Hai?

- **Definition:** Kafka Cluster ek ya ek se zyada servers ka group hota hai. In servers ko **Brokers** kaha jata hai.
- **Role:** Ye "Broker" isliye kehlate hain kyunki ye data receive karte hain, store karte hain, aur consumers ko send karte hain.
- **Identification:** Har Broker ka ek Unique ID (Integer) hota hai.
 - Example: **Broker 101**, **Broker 102**, **Broker 103**.
- **Scale:** Ek cluster mein aap 3 brokers se start kar sakte hain, lekin badi companies mein 100+ brokers bhi hote hain.

[Image of Kafka broker architecture]

2. Brokers and Partitions (Data Distribution)

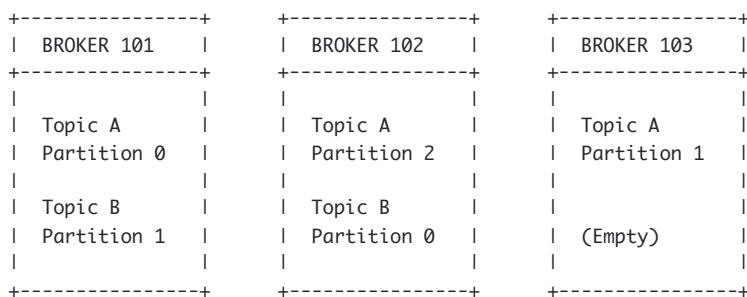
Ek Broker ke paas pure Cluster ka saara data nahi hota. Data distribute hota hai.

- Topic ke **Partitions** alag-alag Brokers par spread hote hain.
- Isse **Horizontal Scaling** milti hai (Zyada load? Zyada brokers add karo, data automatically spread ho jayega).

Example Scenario:

- **Topic A:** 3 Partitions (0, 1, 2)
- **Topic B:** 2 Partitions (0, 1)
- **Cluster:** 3 Brokers (101, 102, 103)

Text Diagram: Data Spread



Observation:

1. **Topic A** teeno brokers par faila hua hai.
2. **Topic B** sirf Broker 101 aur 102 par hai. Broker 103 ke paas Topic B ka koi data nahi hai (jo ki normal hai).
3. Broker ke paas sirf wahi data hota hai jo usse assign kiya gaya hai.

3. Broker Discovery (Bootstrap Servers)

Sabse bada sawal: *Client (Producer/Consumer) ko kaise pata chalega ki kaunsa partition kis broker par hai?*

Kafka mein "**Bootstrap Server**" ka concept use hota hai.

- **Smart Clients:** Aapko cluster ke saare brokers ki list yaad rakhne ki zarurat nahi hai.
- **Rule:** Har Broker ek Bootstrap Server hai.
- Agar aap **kisi bhi ek broker** se connect kar lete hain, toh aap pure cluster se connect ho jate hain.

Connection Flow (Step-by-Step):

1. **Step 1:** Client (e.g., Java App) kisi bhi ek broker (e.g., 101) se connect karta hai.
2. **Step 2:** Client "Metadata Request" bhejta hai.
3. **Step 3:** Broker 101 (jo smart hai) usko pure cluster ki list wapas bhejta hai (Broker 101, 102, 103 aur unke paas kaunse partitions hain).

4. **Step 4:** Ab Client ko sab pata hai, toh wo directly sahi broker se connect karega jahan data bhejna/lenा hai.

Text Diagram: Discovery Mechanism

```
[ Client ]  
|  
| (1. Connect & Request Metadata)  
|  
[ Broker 101 ] <---- (Knows about 102 & 103)  
|  
| (2. Returns List: "101 is here, 102 is there...")  
|  
[ Client ]  
|  
| (3. Automatically connects to required broker)  
|-----> [ Broker 102 ]  
|-----> [ Broker 103 ]
```

4. Metadata Intelligence

- Har Broker ke paas pure cluster ka **Metadata** hota hai.
- Usko pata hota hai ki baaki brokers zinda hain ya mar gaye, aur kiske paas kaunsa partition hai.
- Isi wajah se Client kisi bhi ek darwaze (broker) se andar aaye, usko pura naksha (map) mil jata hai.

Summary Checklist:

- **Broker:** Server jo data hold karta hai.
- **Distribution:** Data (Partitions) brokers ke beech divide hota hai.
- **Bootstrap Server:** Kisi bhi ek node se connect karke pura cluster discover ho jata hai.
- **Metadata:** Har broker ko dusre brokers ki khabar hoti hai.

Next Step: Would you like to understand **Topic Replication Factor** (High Availability & Fault Tolerance) next? Ye explain karega ki agar Broker 102 crash ho gaya toh data ka kya hogा.