

# Apache Kafka

Međuprocesna komunikacija i komunikacija u  
realnom vremenu

# Apache Kafka

- *Distributed Streaming Platform*
- Počiva na PUB-SUB mehanizmu razmene poruka
- Skladišti poruke na robustan način
- Omogućava
  - Skaliranje
  - Brz protok poruka
  - Replikaciju poruka



# Apache Kafka - osnovni koncepti

- Broker
  - Kafka klaster se sastoji iz više brokera
- Producer
  - Klijentska aplikacija koja šalje poruke
- Consumer
  - Klijentska aplikacija koja prima i obrađuje poruke
- Partition
  - Topici su podeljeni na particije što omogućava paralelizaciju
- Consumer Group
  - Poruka dolazi tačno jednom članu grupe
- Leader & Replica
  - Leader - broker na koji se šalju poruke i koji je zadužen za održavanje replikacija
- Offset
  - Za svakog *consumer*-a topika postoji *offset* tj. do kog rednog broja su poruke pročitane

# Instalacija

- Preduslov - JRE
- Download
  - `wget https://www-eu.apache.org/dist/kafka/2.8.2/kafka_2.12-2.8.2.tgz`
  - `tar -xzf kafka_2.12-2.8.2.tgz`
  - `rm kafka_2.12-2.8.2.tgz`
  - `cd kafka_2.12-2.8.2/`

# Pokretanje Kafka klastera

- Pokretanje Zookeeper-a

- `bin/zookeeper-server-start.sh config/zookeeper.properties`
- Sluša na portu 2181

- Pokretanje Kafka brokera

- `bin/kafka-server-start.sh config/server.properties`
- Sluša na portu 9092

# Kreiranje topika

- Kreiranje *test* topika sa jednom particijom i faktorom replikacije 1
  - `bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test`
- Prikaz svih topika
  - `bin/kafka-topics.sh --list --zookeeper localhost:2181`

# Razmena poruka

- Slanje poruka na *test* topik

- `bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test --property parse.key=true --property key.separator=":"`

- Prijem poruka sa *test* topika

- `bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --from-beginning --property print.timestamp=true --property print.key=true --property print.value=true --property print.offset=true --property key.separator=":"`
- Moguće je specificirati još
  - Offset
  - Consumer group id
  - Partition
  - Key-deserializer
  - Value-deserializer
  - ...

# Kafka multi-broker

- Kreiranje dodatna dva Kafka server.properties fajla
  - `cp config/server.properties config/server-1.properties`
  - `cp config/server.properties config/server-2.properties`
- Promeniti sledeća podešavanja
  - `broker.id`
    - redom 1 i 2
  - `listeners`
    - redom 9093 i 9094
  - `log.dirs`
    - Redom `/tmp/kafka-1` i `/tmp/kafka-2`
- Pokrenuti dva nova brokera u klasteru
  - `bin/kafka-server-start.sh config/server-1.properties &`
  - `bin/kafka-server-start.sh config/server-2.properties &`



# Rad sa repliciranim topikom

- **Kreiranje repliciranog topika**

- `bin/kafka-topics.sh --create --zookeeper localhost:2181  
--replication-factor 3 --partitions 1 --topic my-replicated-topic`

- **Prikaz strukture repliciranog topika**

- `bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic  
my-replicated-topic`

- **Slanje poruka**

- `bin/kafka-console-producer.sh --broker-list localhost:9092 --topic  
my-replicated-topic`

- **Prijem poruka**

- `bin/kafka-console-consumer.sh --bootstrap-server localhost:9092  
--from-beginning --topic my-replicated-topic`

# Simulacija otkaza

- Namerno stopiranje brokera 2
  - `ps aux | grep server-2.properties`
  - `kill -9 PID`
- Prikaz strukture repliciranog topika
  - `bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic my-replicated-topic`
- Poruke je i dalje moguće pročitati iako je *leader* broker stopiran
  - `bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --from-beginning --topic my-replicated-topic`

# Consumer grupe

- Kreiranje topika sa 3 particije i replikacijom

- `bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 3 --partitions 3 --topic my-topic`

- Slanje poruka

- `bin/kafka-console-producer.sh --broker-list localhost:9092 --topic my-topic`

- Pokretanje više consumer-a s istom grupom

- `bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic my-topic --consumer-property group.id=g1`
  - `bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic my-topic`
  - `bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list`
  - `bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe --group g1`

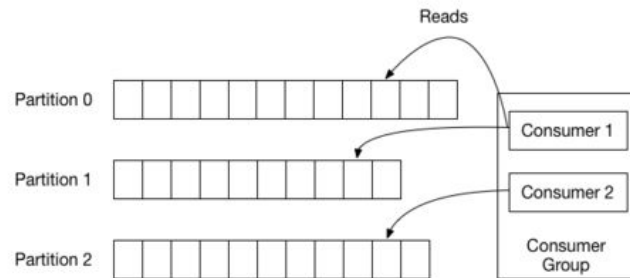
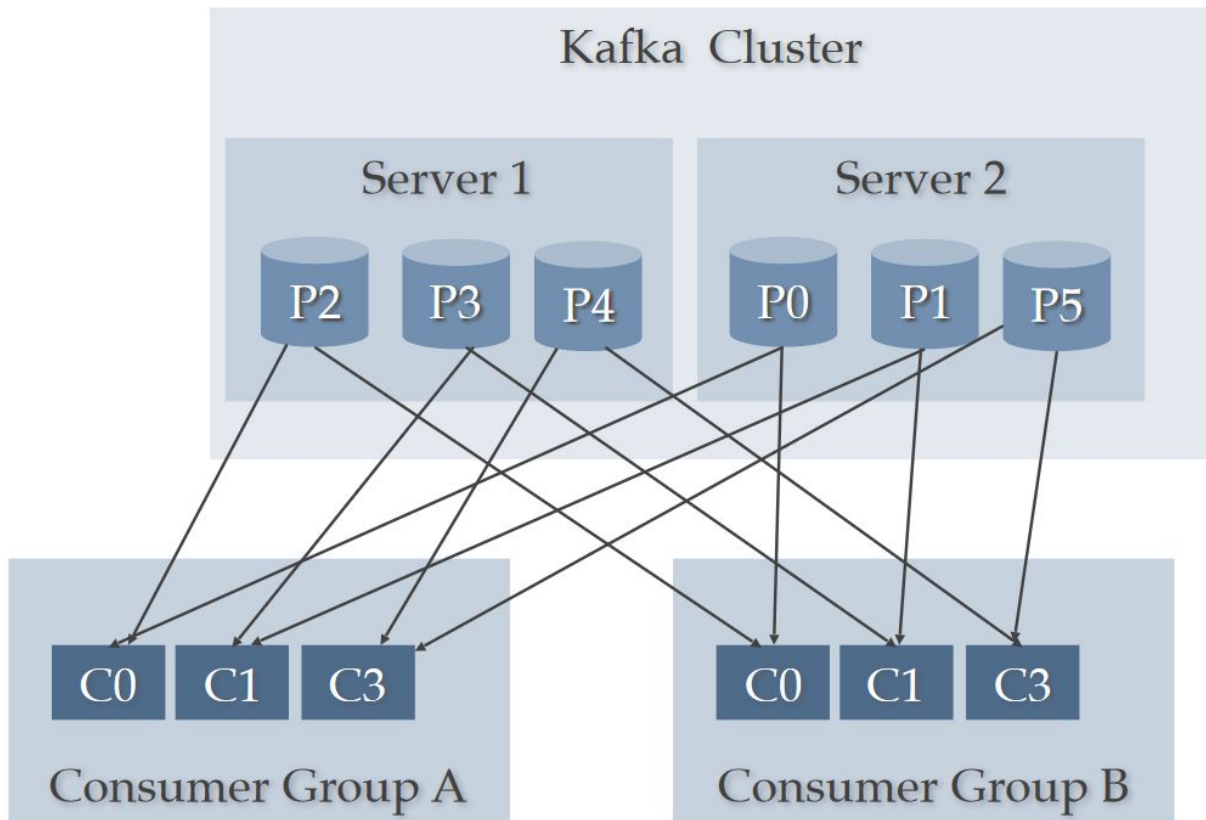


Figure 1: Consumer Group

# Consumer grupe



# Replikacija i distribucija topika

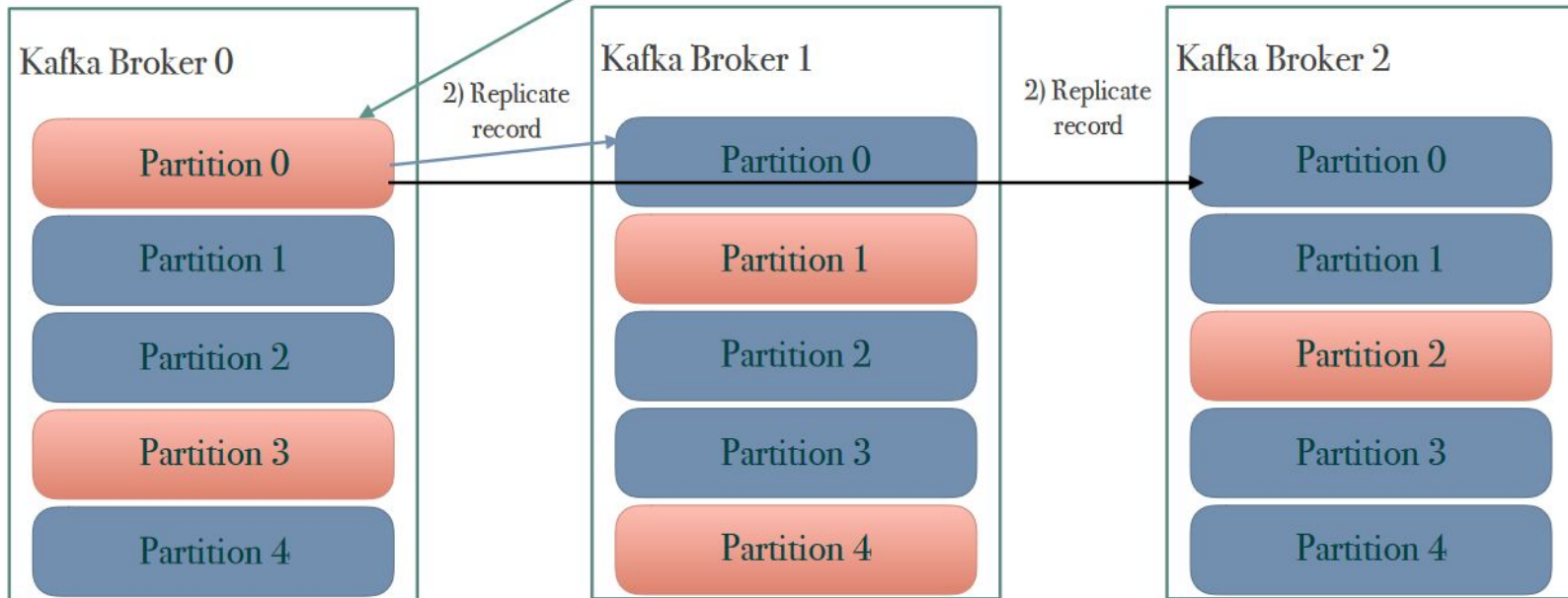
Record is considered "committed" when all ISR for partition wrote to their log.

Client Producer

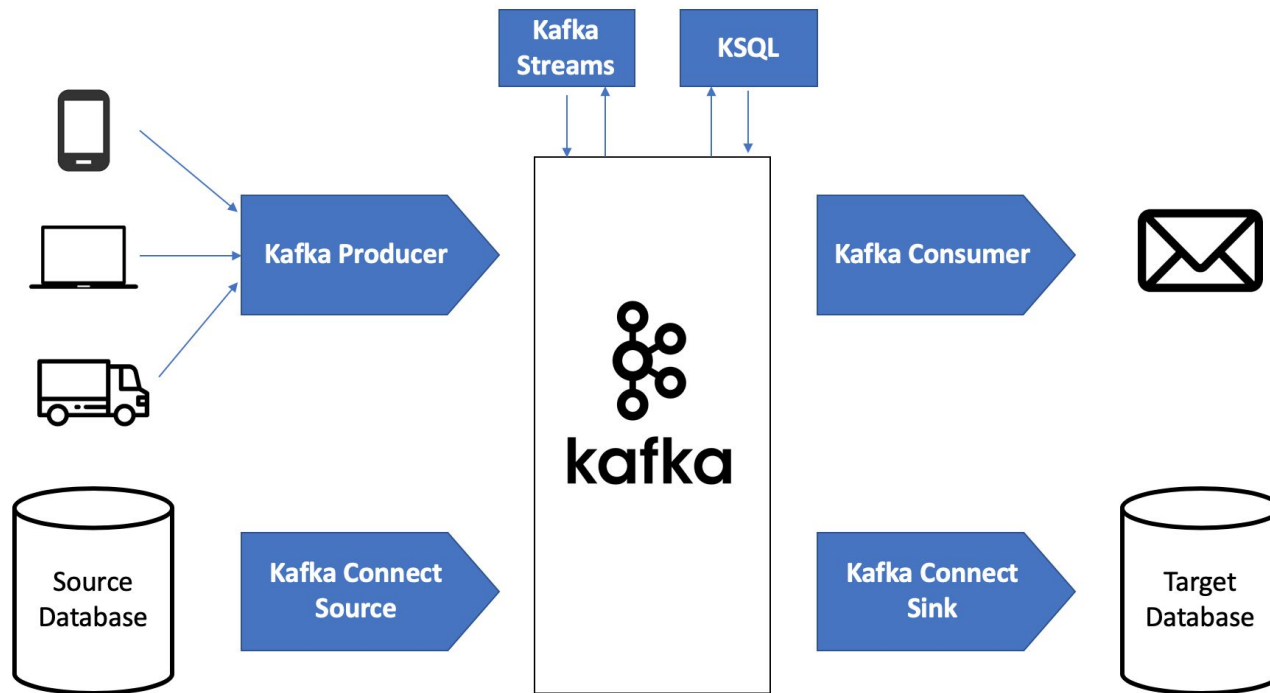
Leader **Red**  
Follower **Blue**

**Only committed records are readable from consumer**

1) Write record



# Kafka APIs



# Kafka klijenti

- Pored izvornog API-ja u Javi postoje i klijentske biblioteke za većinu programskih jezika opšte namene
  - C/C++, Python, goLang, Node.js, Ruby, .NET, ...
- Više o dostupnim implementacijama klijentskih Kafka biblioteka na sledećem linku
  - <https://cwiki.apache.org/confluence/display/KAFKA/Clients>
- Confluent-kafka API za Python
  - `python3 -m venv .venv`
  - `source .venv/bin/activate`
  - `pip install confluent-kafka`
  - `python3 kafka-python/consumer|producer-example.py`

# Kafka klaster u docker-u

- Multi-broker Kafka klaster kroz docker-compose
  - Apache Zookeeper servise
  - Proizvoljan broj Apache Kafka servisa
- Primeri “dockerizovane” Kafke
  - <https://github.com/wurstmeister/kafka-docker>
  - <https://github.com/simplesteph/kafka-stack-docker-compose>
    - Sadrži dodatne servise koji su deo Conluent platforme:
      - Servis za upravljanje shemom topika
      - Connect servis
      - UI za topike, sheme i Connect