

GUIA

DE

KAFKA



Iniciar Kafka com KRaft (sem Zookeeper)

Dentro da pasta kafka

```
/ bin/kafka-storage.sh format -t $(bin/kafka-storage.sh random-uuid) -c config/kraft/server.properties  
bin/kafka-server-start.sh config/kraft/server.properties
```



CLI Kafka – Criar Tópicos

```
bin/kafka-topics.sh --create --topic temperature --bootstrap-server localhost:9092 --partitions 1 --  
replication-factor 1
```

```
bin/kafka-topics.sh --create --topic humidity --bootstrap-server localhost:9092 --partitions 1 --  
replication-factor 1.
```



CLI Kafka – Produzir mensagens

```
bin/kafka-console-producer.sh --topic sensor-data --bootstrap-server localhost:9092
```

```
temp:32.1
```

```
temp:33
```

```
temp:28.7
```



CLI Kafka – Consumir mensagens

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



Python – KAKFA com Python

1 - Instalar o PYTHON

```
pip install kafka-python
```

2 - PRODUCER.py (gera mensagens de temperatura e umidade a cada 1 segundo)

```
from kafka import KafkaProducer

producer = KafkaProducer(bootstrap_servers='localhost:9092')

for i in range(5):

    value = f"temp={20+i*0.5}"

    producer.send('sensor-data', value.encode('utf-8'))

producer.flush()
```

2 - CONSUMER.py (lê o tópico de temperatura e salva os valores > 23 em `alertas.txt`)

```
from kafka import KafkaConsumer

consumer = KafkaConsumer(
    'sensor-data',
    bootstrap_servers='localhost:9092',
    auto_offset_reset='earliest',
    enable_auto_commit=True
)

for message in consumer:
    print(message.value.decode('utf-8'))
```



Python – DESAFIO

1. Alterar o producer.py para gerar também "**humidity=XX**" aleatórios
2. Alterar o consumer.py para **filtrar e mostrar só temperaturas acima de 22**