



# Big Data with Kafka

Course overview  
Kafka  
Concepts  
Hands on





## **Intro to BigData**

Overview of BigData  
BigData Ecosystems  
Setting up BigData  
env.

## **BigData with Kafka**

Real-time data  
processing  
Key components of  
Kafka  
Hands on

## **Apache Spark**

Architecture  
Spark streaming  
and jobs  
Hands on

## **Data Lake**

Design &  
Architecture  
Pipeline  
Hands on

What is Kafka?

# Kafka popularity

[kafka](#)

# What is Apache Kafka®?



↑ TOPIC

Topics  
Producers  
Consumers  
Brokers  
Zookeeper

# Welcome to Confluent Cloud, Shekhar

Getting your data in motion is quick and easy

Start generating data and developing your first pipeline by adding a cluster.





Signed up for confluent cloud  
Create 1st Kafka cluster  
Create a topic  
Add a message to the topic  
Use producer and consumer

## CLI and tools

Confluent CLI   Confluent Platform Components   Kafka Connect

### Try it out!

Now that you have a cluster up and running in Confluent Cloud, you can administer using the [Confluent CLI](#).

#### 1. Install / Update the Confluent CLI

Run this command to install the Confluent CLI:

```
$ curl -sL --http1.1 https://cnfl.io/cli | sh -s -- latest
```

[Copy](#)

This script will install the CLI in `./bin` by default. If you want to install it somewhere else, add the path to the end of the command and to your `$PATH` variable.

**Note:** On Windows, you might need to install an appropriate Linux environment to have the `curl` and `sh` commands available, such as the [Windows Subsystem for Linux](#). You can also download and install the [raw binaries](#).

If already installed, update to the latest version with:

```
$ confluent update
```

[Copy](#)

#### 2. Log in to your Confluent Cloud organization using the Confluent CLI

Run this command to log in to the Confluent CLI:

```
$ confluent login --save
```

[Copy](#)

When prompted for your username and password, enter the same credentials that you used to log in to Confluent Cloud.

The optional `--save` flag saves your login credentials to a local file for future

What is Kafka Brokers?

# Brokers



- An computer, instance, or container running Kafka process
- Manage partitions
- Handle write and read requests
- Manage replication of partitions
- Intentionally very simple

**KAFKA 101**



# Kafka Connector



# Connector Plugins

Confluent Cloud offers pre-built, fully managed Kafka connectors that make it easy to instantly connect your clusters to popular data sources and sinks. Connect to external data systems effortlessly with simple configuration and no ongoing operational burden.

Filter by: 

Deployment


Type

Sort by: 


Popular


Displaying 232 connectors

 Fully managed cloud connector




**Sample Data**  
Datagen Source






**Snowflake Sink**  
Sink


Popular






**Google Cloud Storage Sink**  
Sink


Popular






**Elasticsearch Service Sink**  
Sink


Popular






**MongoDB Atlas Source**  
Source


Popular






**MongoDB Atlas Sink**  
Sink


Popular






**Amazon Kinesis Source**  
Source


Popular





**Salesforce CDC Source**  
Source

Popular





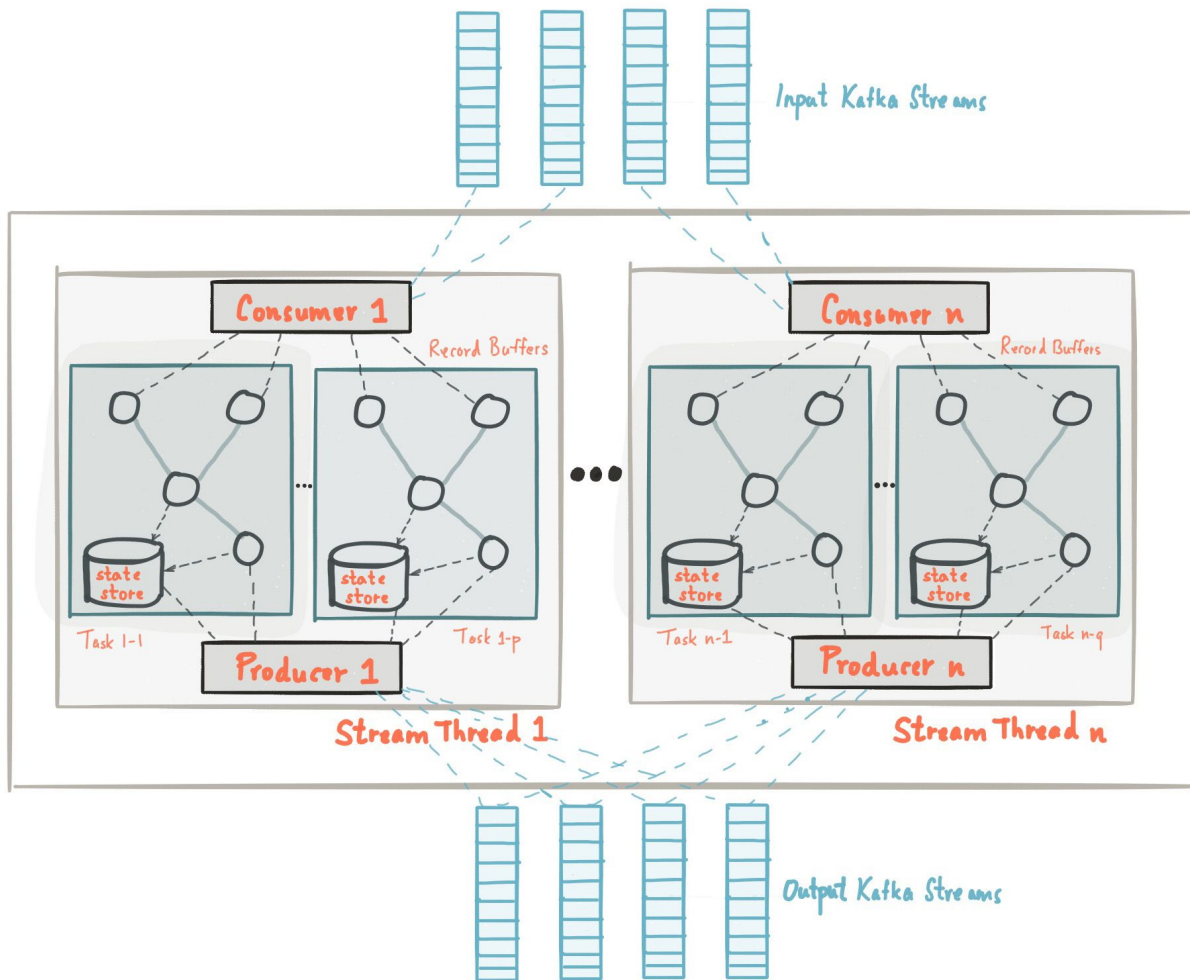
Partitions enable data distribution and parallel processing within Kafka topics.

Offsets track the position of each message within a partition, enabling consumers to process messages in order, resume after failure, and support data replay.

Together, partitions and offsets provide scalability, fault tolerance, and flexibility in data processing within Kafka.



# Kafka Streams



kSQLDB



Hands on: ksqlDB

# Stateful Aggregations (Materialized Views)



```
CREATE MATERIALIZED VIEW mv1 AS  
SELECT * FROM stream1  
WHERE stream1.col1 = 'value1';
```

```
CREATE MATERIALIZED VIEW mv2 AS  
SELECT * FROM stream1  
WHERE stream1.col1 = 'value2';
```

```
CREATE MATERIALIZED VIEW mv3 AS  
SELECT * FROM stream1  
WHERE stream1.col1 = 'value3';
```

```
CREATE MATERIALIZED VIEW mv4 AS  
SELECT * FROM stream1  
WHERE stream1.col1 = 'value4';
```

```
CREATE MATERIALIZED VIEW mv5 AS  
SELECT * FROM stream1  
WHERE stream1.col1 = 'value5';
```

```
CREATE MATERIALIZED VIEW mv6 AS  
SELECT * FROM stream1  
WHERE stream1.col1 = 'value6';
```

```
CREATE MATERIALIZED VIEW mv7 AS  
SELECT * FROM stream1  
WHERE stream1.col1 = 'value7';
```

```
CREATE MATERIALIZED VIEW mv8 AS  
SELECT * FROM stream1  
WHERE stream1.col1 = 'value8';
```

```
CREATE MATERIALIZED VIEW mv9 AS  
SELECT * FROM stream1  
WHERE stream1.col1 = 'value9';
```

```
CREATE MATERIALIZED VIEW mv10 AS  
SELECT * FROM stream1  
WHERE stream1.col1 = 'value10';
```

**kSQLDB 101**

## PUSH QUERY



TELLS YOU:

ALL VALUE CHANGES

EXITS:

NEVER

## PULL QUERY



POINT IN TIME VALUE

IMMEDIATELY

# Lambda Function

# Lambda Functions



Start with a stream containing an array of integers to process

```
CREATE STREAM stream1 (  
  id VARCHAR,  
  numbers ARRAY<INTEGER>  
) WITH (  
  kafka_topic = 'topic3', partitions = 3,  
  value_format = 'json'  
) ;
```

Filter the elements in the stream to produce a new stream

```
CREATE STREAM Filtered  
  AS SELECT id,  
    FILTER(numbers, x => (x%2 = 0)) AS even_numbers  
FROM stream1 ENRICH CHANGES;
```

## ksqlDB 101





Leveraging cloud technology can reduce your carbon footprint by optimizing energy usage and promoting sustainability.



# Carbon Footprint Tools

Measure and reduce your  
IT carbon emissions



world beyond streaming



# Flink

# Core terminologies



Stream

Event

Task

Operator

Job

State

Checkpoint

# Getting started with Flink

Have Questions?

Working with DATA  
is an ART

