Stream Processing With Apache Flink

Big Data Demystified 2024

About the presenter

Dunith Dhanushka

Senior Developer Advocate, Redpanda Data

- Event streaming, real-time analytics, and stream processing enthusiast
- Frequent blogger, speaker, and an educator

twitter.com/dunithd

Medium

medium.com/event-driven-utopia

linkedin.com/in/dunithd/

© 2023 REDPANDA DATA

Agenda

1. An introduction to stream processing and Apache Flink
2. Deploying a Flink cluster
3. Generating a Flink application project for Java
4. Stateless operations
5. Q & A

© 2023 REDPANDA DATA

Code is in GitHub!

https://github.com/dunithd/kafka-meetup-frankfurt

© 2023 REDPANDA DATA

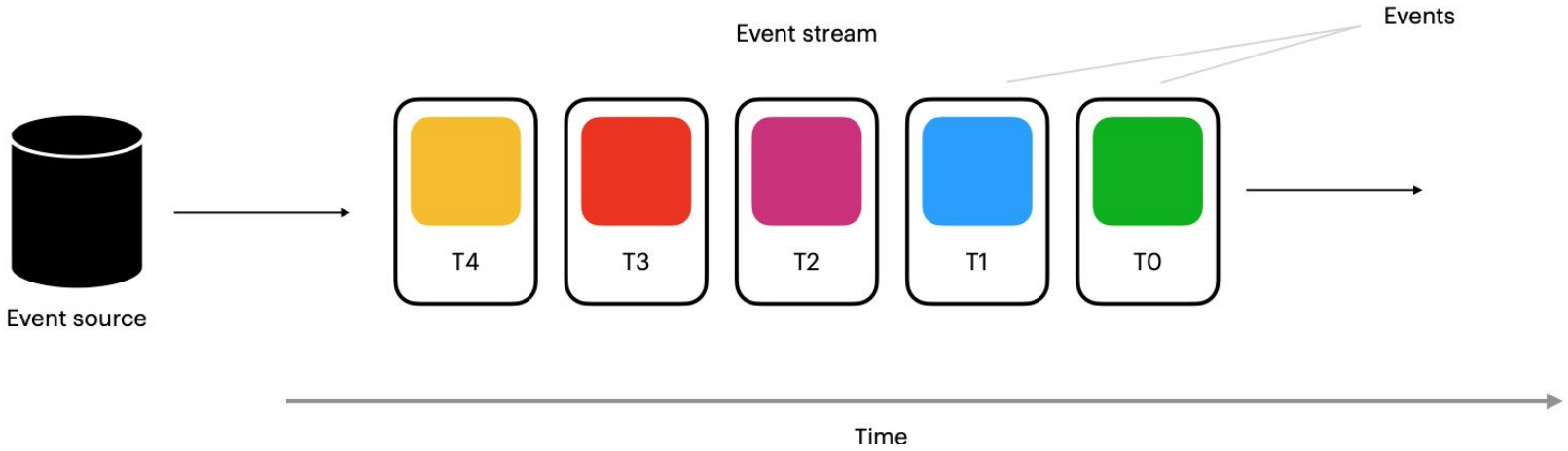
What is Stream Processing?

© 2023 REDPANDA DATA

Event streams

A data stream consists of a series of data points **ordered in time**.

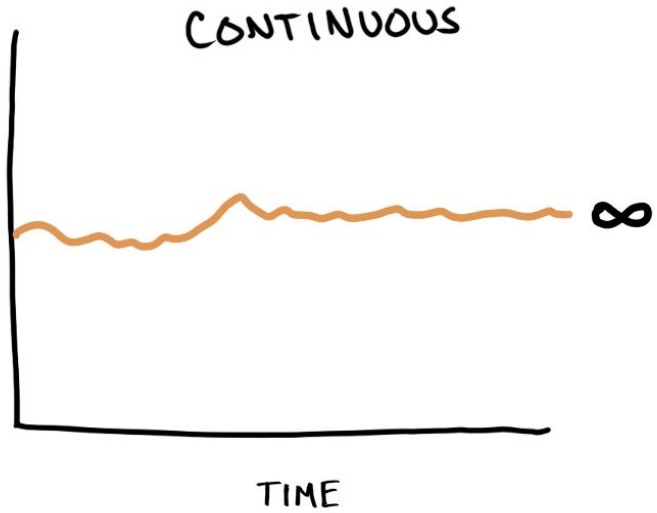
Each data point represents an "**event**" or a change in the state of the business.



© 2023 REDPANDA DATA

Stream processing

- Enriching, transforming, and reacting to data **continuously**
- Great for time-sensitive business use cases
- Jobs run indefinitely



© 2024 Redpanda Data

What is Flink?

© 2023 Redpanda Data

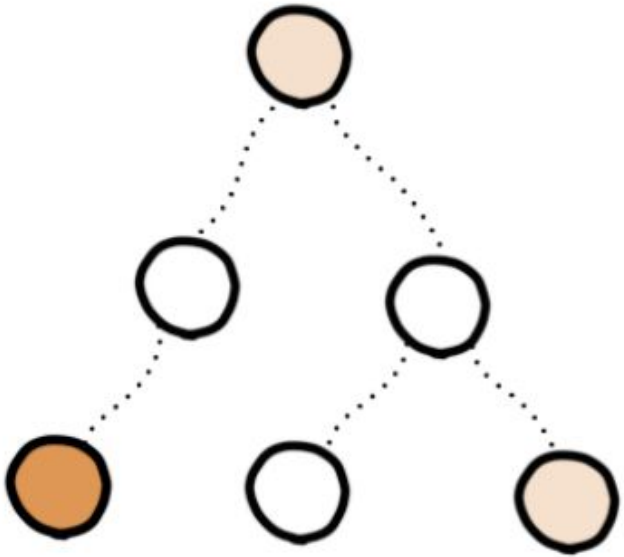
What is Apache Flink?

- Data processing framework
- Open-source project
- Very active community
- Battle-tested

© 2024 Redpanda Data

Dataflow programming

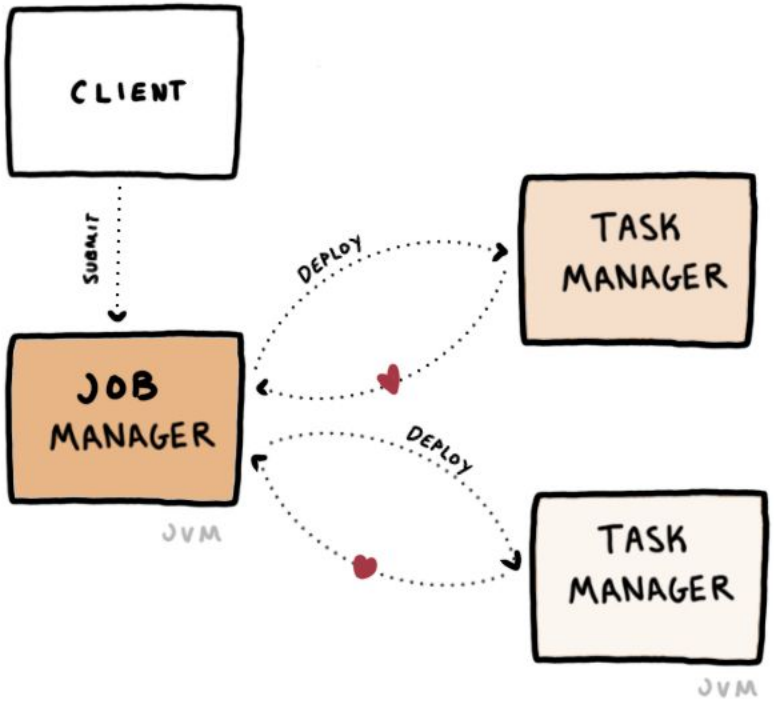
- Dataflow graphs, or ***Job graphs***
- Nodes represent **operators** (functions)
- Edges represent the flow of data between operators
- Logical representation
- Can be **parallelized**



© 2024 Redpanda Data

Components

- Clients
- Job manager
- Task manager



© 2024 Redpanda Data

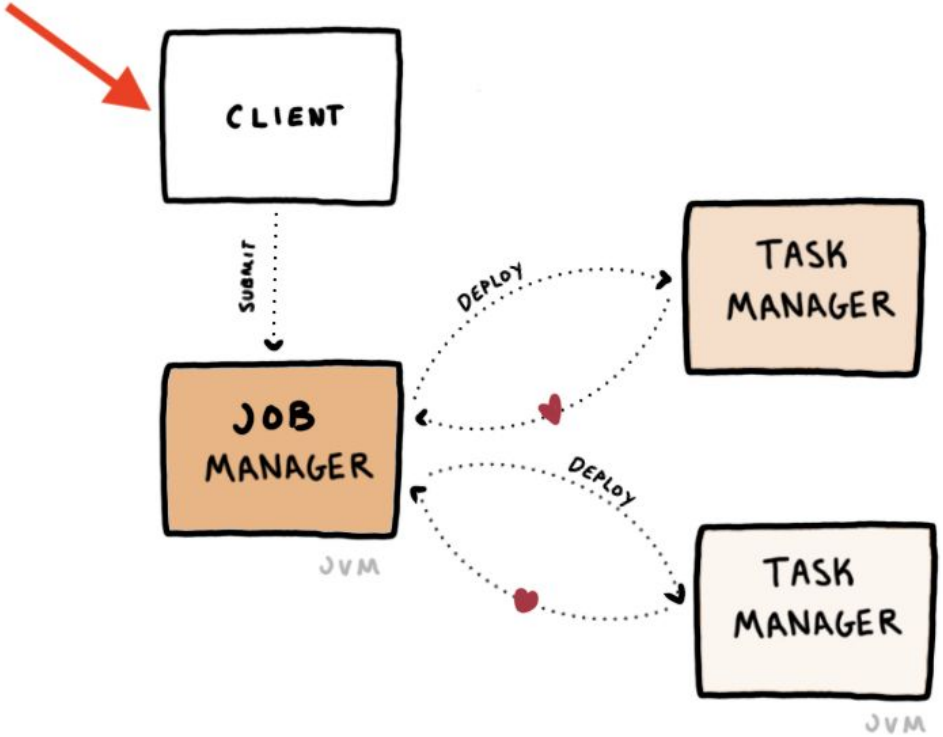
Clients

- SQL client
- PyFlink
- REST client

  `POST jobs/`

- CLI

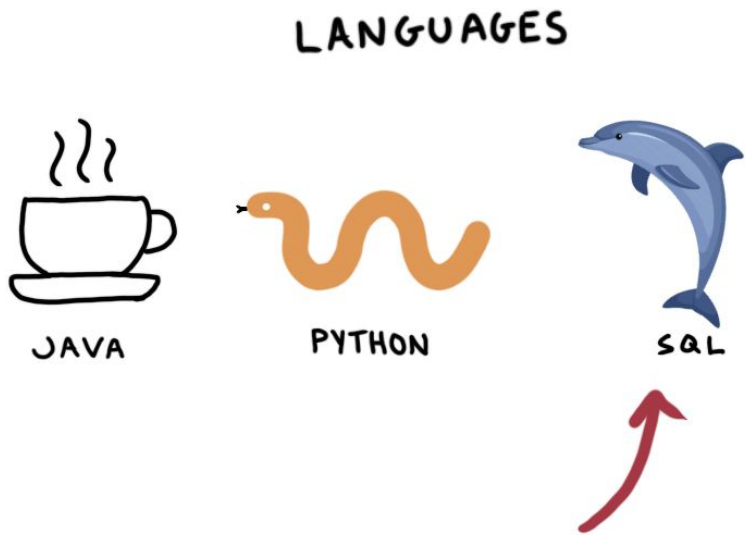
  `./bin/flink run`



© 2024 Redpanda Data

Languages

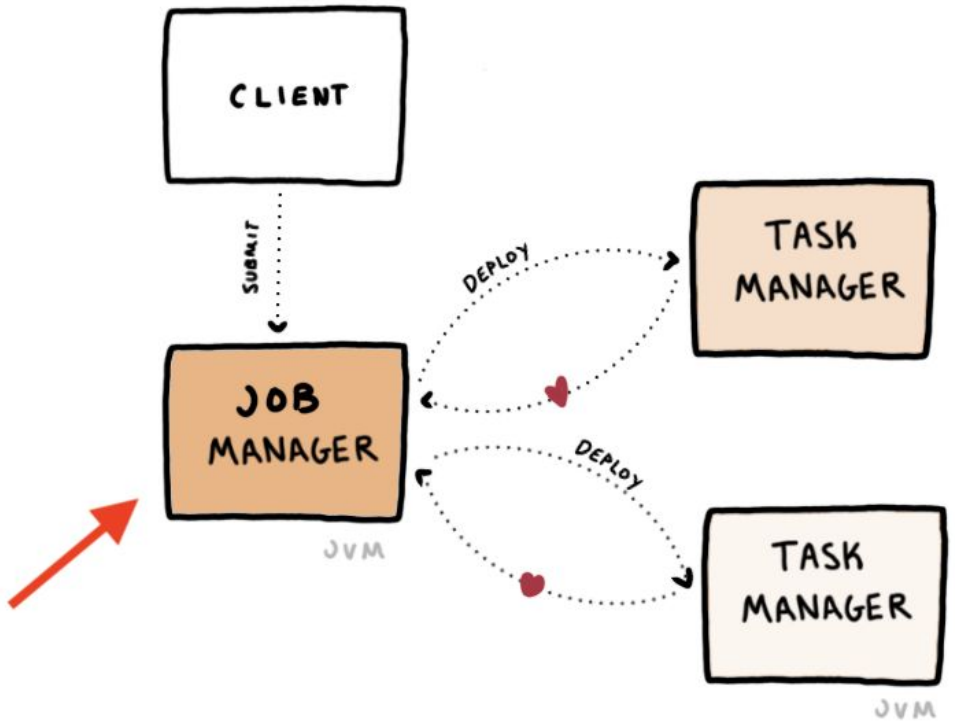
- JVM
- PyFlink
- Flink SQL



© 2024 Redpanda Data

Job manager

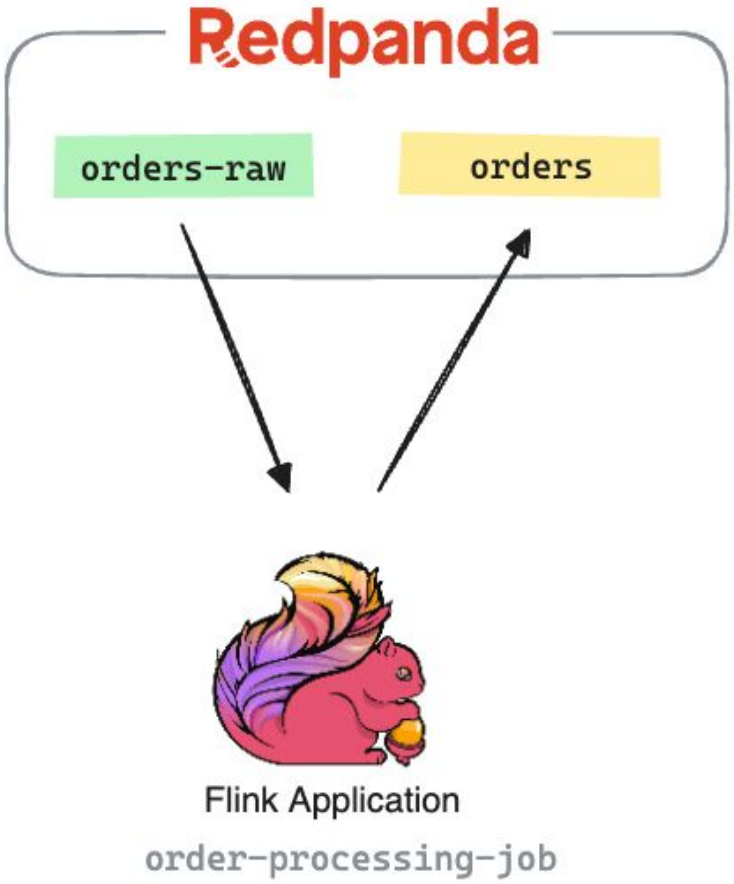
- Coordinator
- Initiates checkpoints
- UI
- Different deployment strategies and cluster modes
- HA optional



© 2024 Redpanda Data

Demo

© 2023 REDPANDA DATA


Redpanda
orders-raw
orders
Flink Application
order-processing-job

What is Redpanda?

© 2023 Redpanda Data

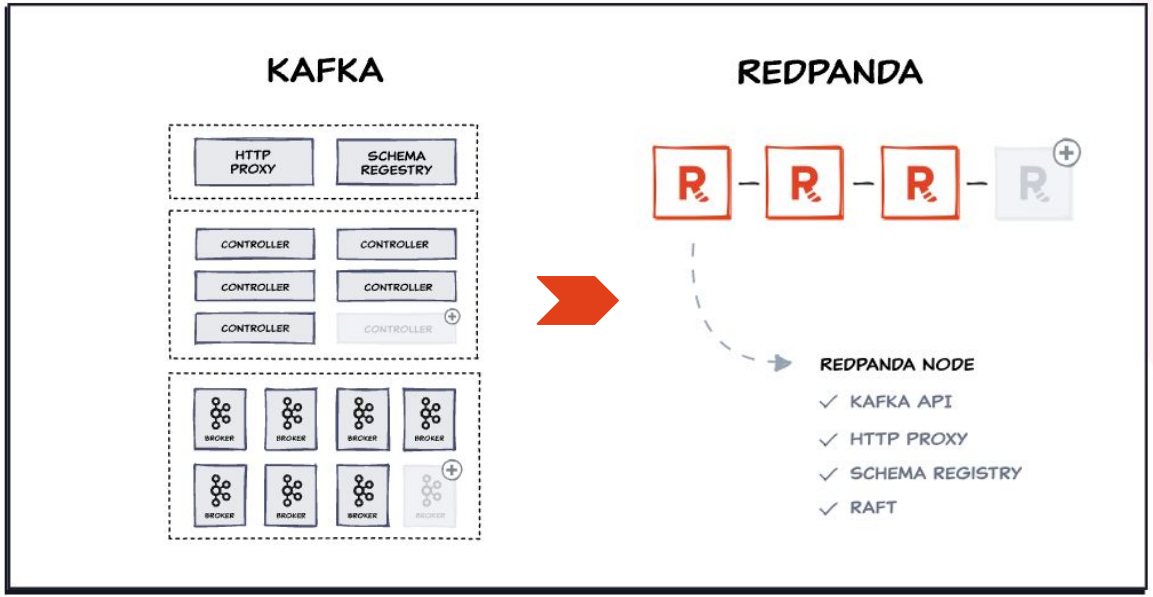
Redpanda is a Kafka API compatible streaming data platform

- Written in C++
- Thread-per-core architecture
- Designed for modern hardware

© 2023 REDPANDA DATA

Simple to deploy, use and manage

- Single binary
- Kafka-compatible APIs
- Easy Day 2 Ops
- Dev-friendly interface



© 2023 REDPANDA DATA

What you need?

- Docker Compose (Docker Desktop installed locally)

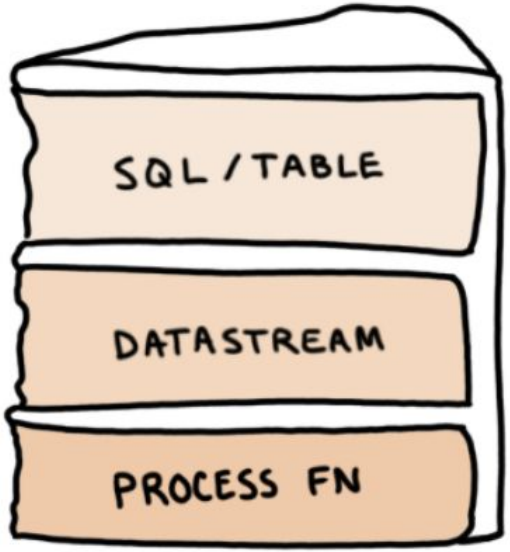
© 2023 REDPANDA DATA

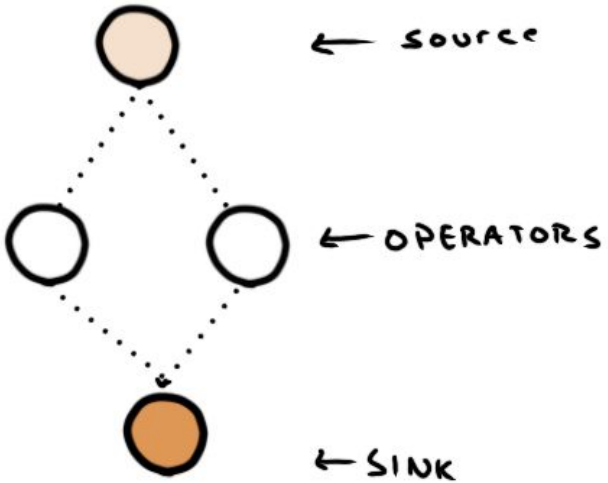
Developing Flink Applications

With Java

Redpanda

© 2023 Redpanda Data

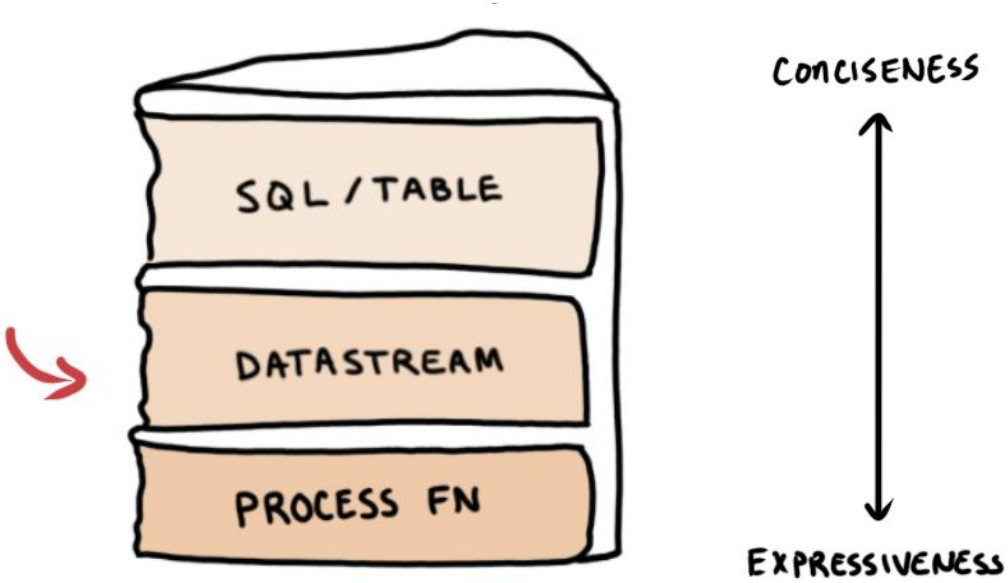




© 2024 Redpanda Data

Lower levels

- More expressive
- Imperative approach
- Use for more complex use cases



© 2024 Redpanda Data

DataStream API

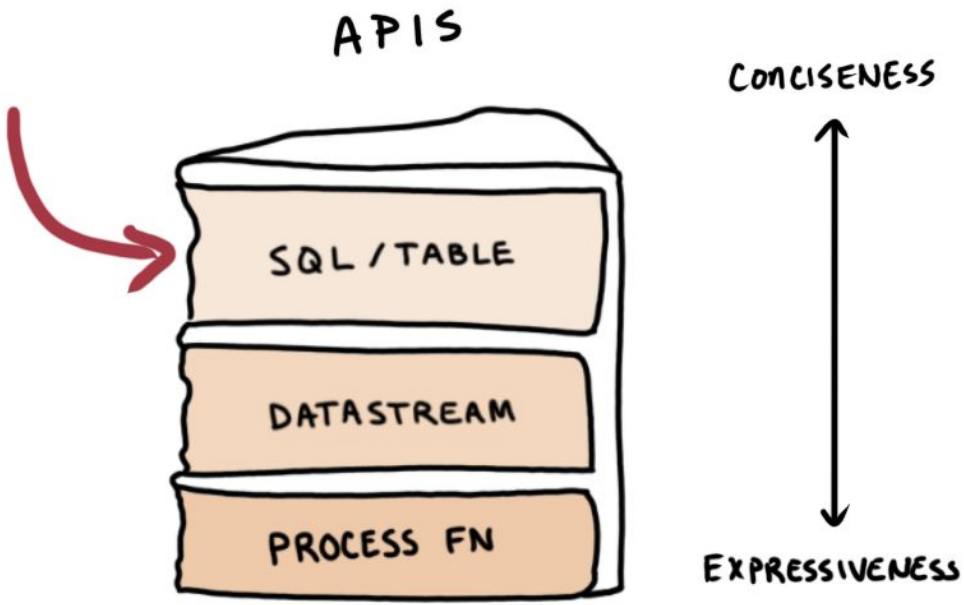
```
StreamExecutionEnvironment env=
      StreamExecutionEnvironment.getExecutionEnvironment();


DataStream<Tuple2<String, Integer>> dataStream = env
        .socketTextStream("localhost", 9999)
        .flatMap(new Splitter())
        .keyBy(value -> value.f0)
        .window(TumblingProcessingTimeWindows.of(Time.seconds(5)))
        .sum(1);
dataStream.print();
env.execute("Window WordCount");
```

Redpanda

© 2024 Redpanda Data

Higher levels

- More concise
- Declarative approach
- Use when lower-level state / time access isn't needed



© 2024 Redpanda Data

Table API

```
EnvironmentSettings settings = …

TableEnvironment tEnv = TableEnvironment.create(settings);

// specify table program

Table orders = tEnv.from("Orders"); // schema (a, b, c, rowtime)

Table counts = orders

        .groupBy($("a"))

        .select($("a"), $("b").count().as("cnt"));

counts.execute().print();
```


© 2024 Redpanda Data

Choosing an API

Table API

- If you want a **declarative** interface or closer alignment with **SQL**
- You don't need low-level control of **state** or time-based operations

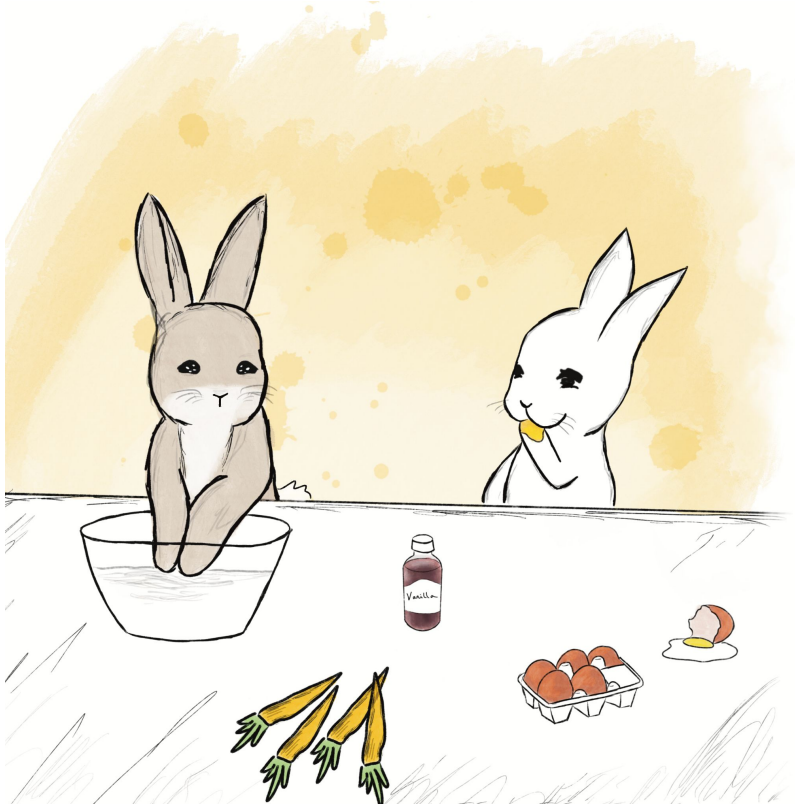
DataStream API

- If you need more fine-grained control of streaming primitives (state, time)
- Imperative interface
- You want to work with a more **raw** stream of events
- Custom business logic

© 2024 Redpanda Data

Mixing APIs

- Mixing is possible
- Example:
  - Table API for cleansing, filtering, or re-shaping input data
  - DataStream API for the primary application logic



Demo

© 2023 REDPANDA DATA

What you need?

- Java 11 or higher
- Maven 3+
- Internet connection
- Terminal
- IDE - IntelliJ IDEA or Eclipse

© 2023 REDPANDA DATA

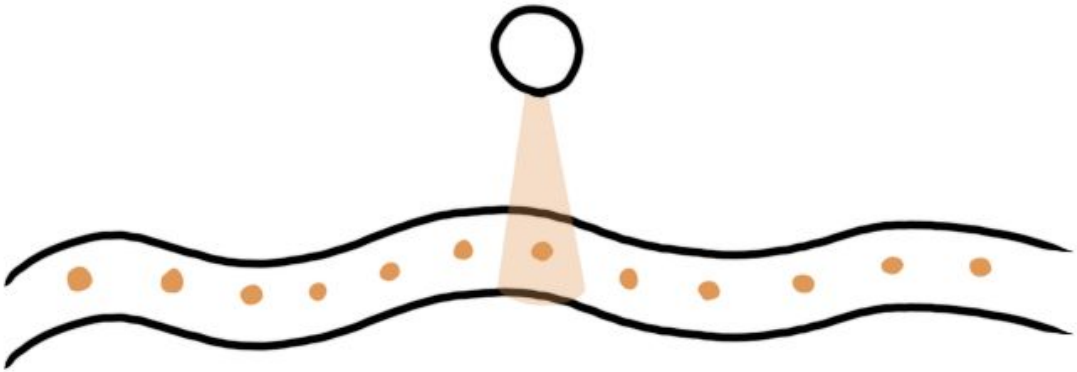
```
mvn archetype:generate \
-DarchetypeGroupId=org.apache.flink \
-DarchetypeArtifactId=flink-quickstart-java \
-DarchetypeVersion=1.16.2 \
-DgroupId=kafka.meetup.flink \
-DartifactId=flink-quickstart-project \
-DinteractiveMode=false
```

Stateless Operations

© 2023 Redpanda Data

Stateless operators

observe events without historical context



© 2024 Redpanda Data

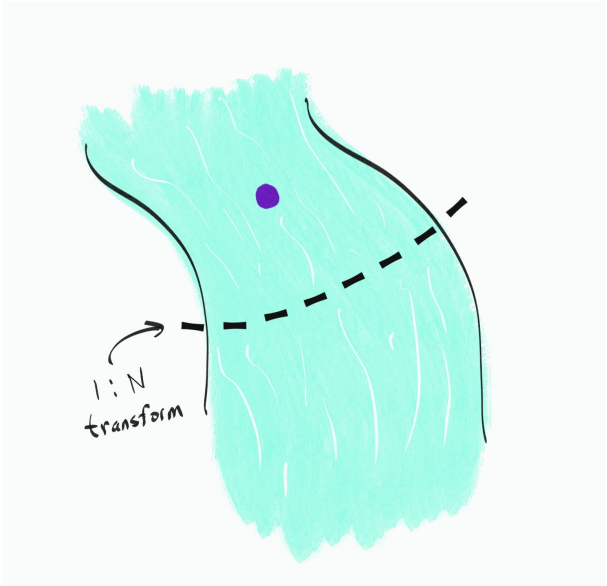
Stateless applications

- No memory
- Lightweight storage requirements
- Simple to deploy

© 2024 Redpanda Data

Examples of stateless operations

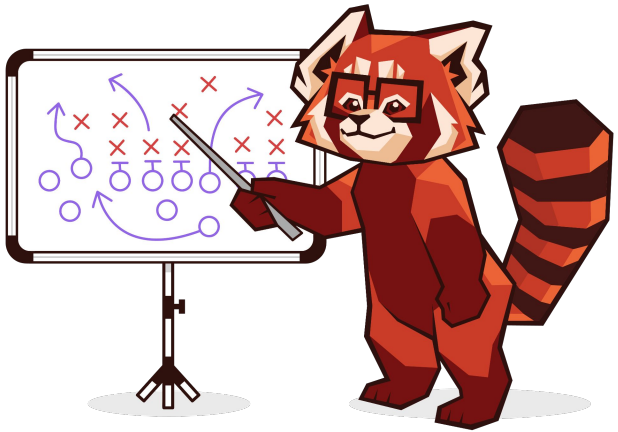
- Filtering data
- Single record data transformations
- Scrubbing records of sensitive information



© 2024 Redpanda Data

Redpanda University

Free, self-paced online learning
https://university.redpanda.com



- Learn the fundamentals of data streaming and Redpanda
- Install Redpanda and use the rpk CLI to configure it
- Create producers and consumers in Java, Python and NodeJS
- Sign up today for free!

© 2023 REDPANDA DATA

Thanks for joining!

Let’s keep in touch

@redpandadata

redpanda-data

redpanda-data

hello@redpanda.com

© 2023 Redpanda Data