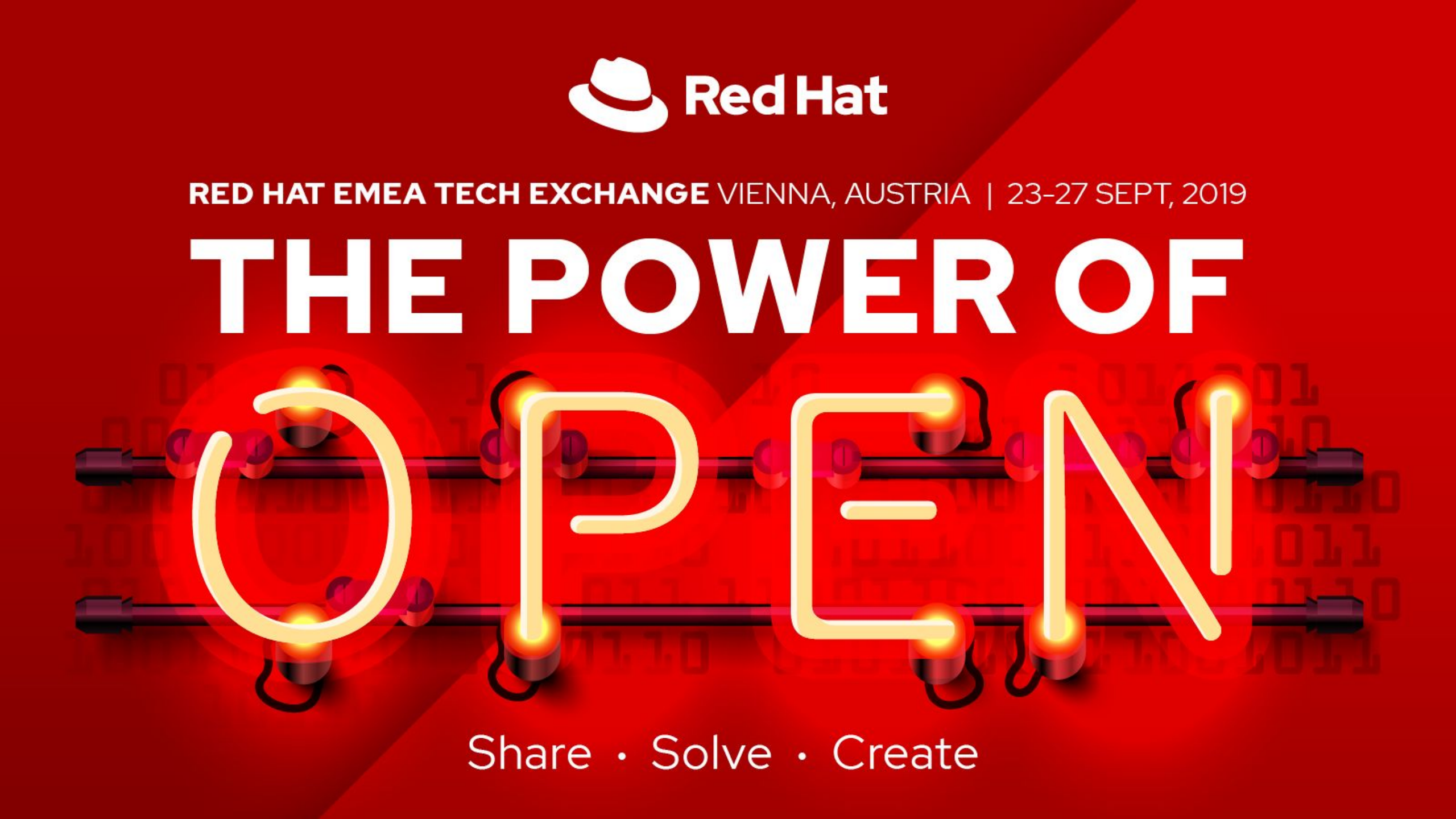

Red Hat
RED HAT EMEA TECH EXCHANGE VIENNA, AUSTRIA | 23-27 SEPT, 2019
THE POWER OF
OPEN
Share · Solve · Create

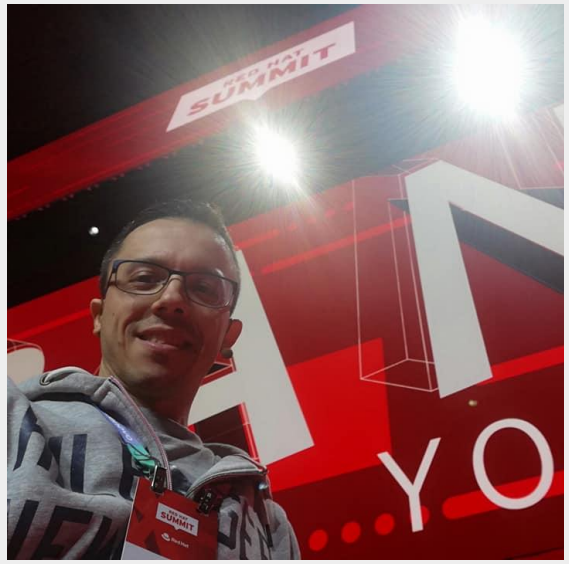
# Building data streaming applications with Apache Kafka on OpenShift using AMQ Streams


Paolo Patierno
Principal Software Engineer

Stanislav Knot
Software Engineer




```
apiVersion: redhat/v1
kind: PrincipalSoftwareEngineer
metadata:
  name: Paolo Patierno
  namespace: Red Hat, Messaging & IoT team
  annotations:
    eclipse/committer: Vert.x, Hono & Paho
    microsoft/mvp: Azure & IoT
  labels:
    family: dad of two, husband of one
    sports: running, swimming, motogp, vr46, ssc napoli
spec:
  replicas: 1
  containers:
    - image: patiernohub.io/paolo:latest
```




**Paolo Patierno**
Principal Software Engineer



Red Hat

```
apiVersion: redhat/v1
kind: SoftwareEngineer
metadata:
  name: Stanislav Knot
  namespace: Red Hat, Messaging & IoT team
  annotations:
    activemq/artemis: committer
  labels:
    hobbies: 3D graphics, Unity programming, sports
spec:
  replicas: 1
  containers:
    - image: knothub.io/stanislav:latest
```


**Stanislav Knot**
Software Engineer

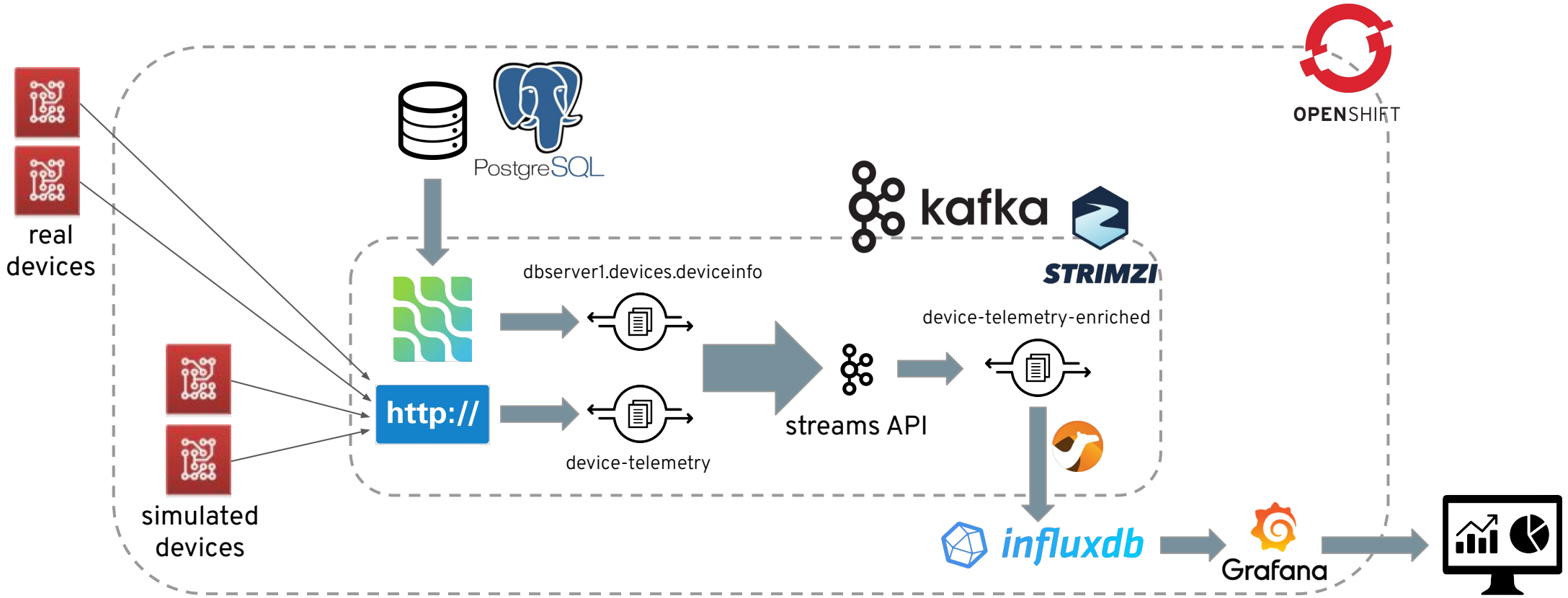


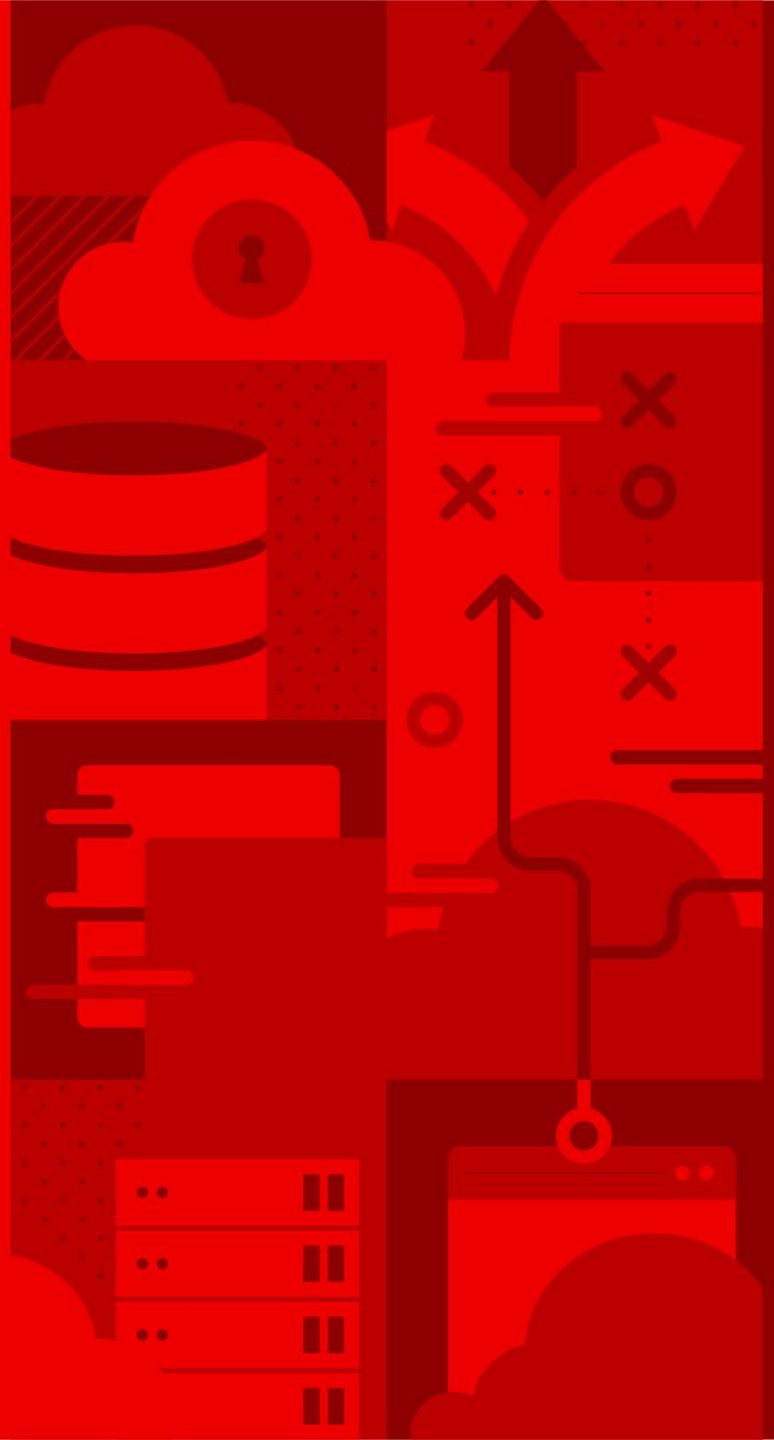
Red Hat

# Agenda

1. **Apache Kafka on OpenShift**
   - Challenges
   - AMQ Streams operator
   - Demo
2. **Apache Kafka Connect**
   - Introduction
   - Change Data Capture & Debezium
   - Demo
3. **Apache Kafka Streams**
   - Introduction
   - Demo
4. **Apache Kafka HTTP Bridge**
   - Why? Use cases
   - API & endpoints
   - Demo

# Scenario





# Apache Kafka on OpenShift: AMQ Streams



Red Hat

# The challenges


**A Kafka cluster requires ...**

... a stable broker identity and network address

... a way for brokers to discover each other

... durable state and storage recovery

... brokers accessible from clients, directly

... and we also have Zookeeper


**OpenShift provides ...**

... Statefulsets for stable identity and network

... together with Headless services for discovery

... Services for accessing the cluster

... Secrets and ConfigMaps for configurations

... PersistentVolume and PersistentVolumeClaim for durable storage

# Welcome to Strimzi

STRIMZI

CLOUD NATIVE
COMPUTING FOUNDATION

**Open source project licensed under Apache License 2.0**

- CNCF Sandbox project

**Focuses on running Apache Kafka on Kubernetes and OpenShift**

- Container images for Apache Kafka and Apache Zookeeper
- Operators for managing and configuring Kafka clusters, topics or users

**Provides Kubernetes-native experience**

- Kafka cluster, topic and user as Kubernetes custom resources



Red Hat

# Red Hat AMQ Streams

## Part of the Red Hat AMQ Suite

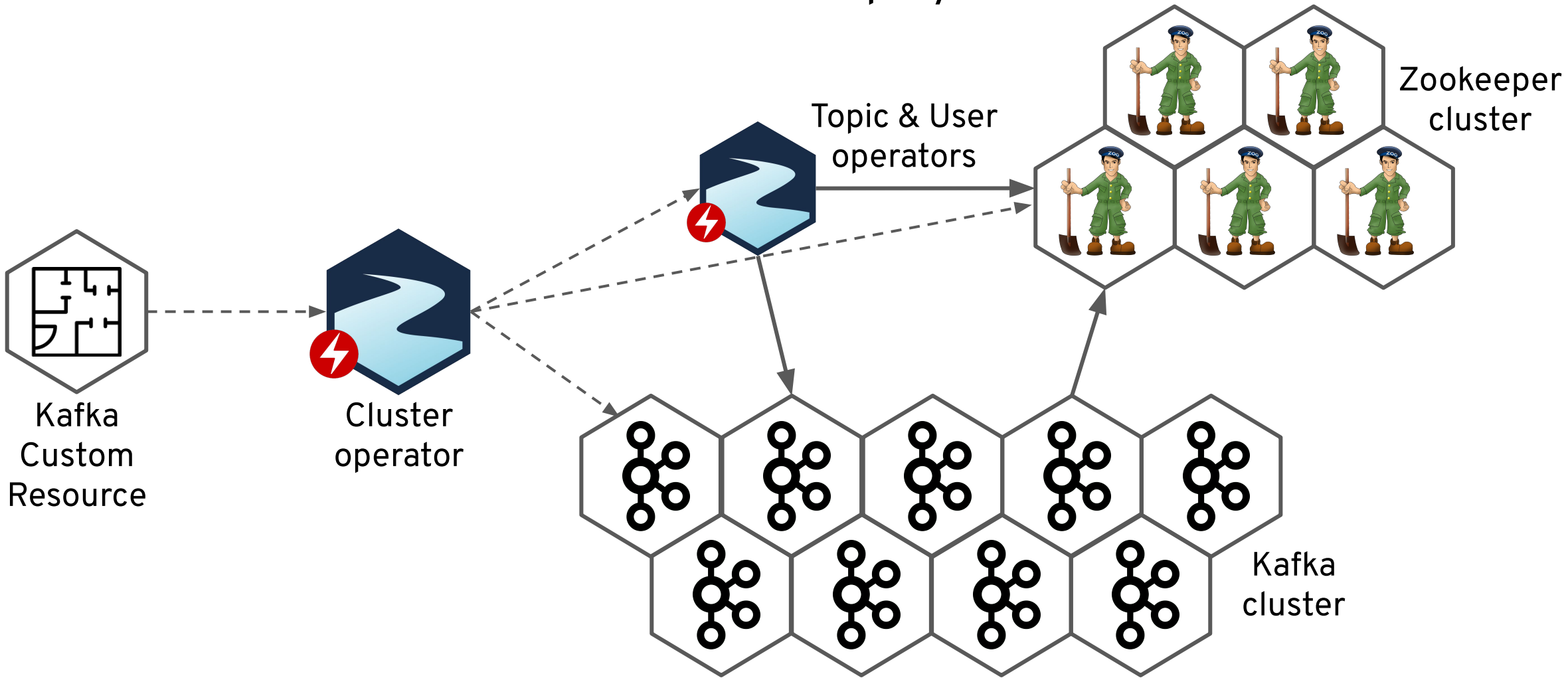
## AMQ Streams on OCP

- Running Apache Kafka on OpenShift Container Platform
- Based on the upstream Strimzi project

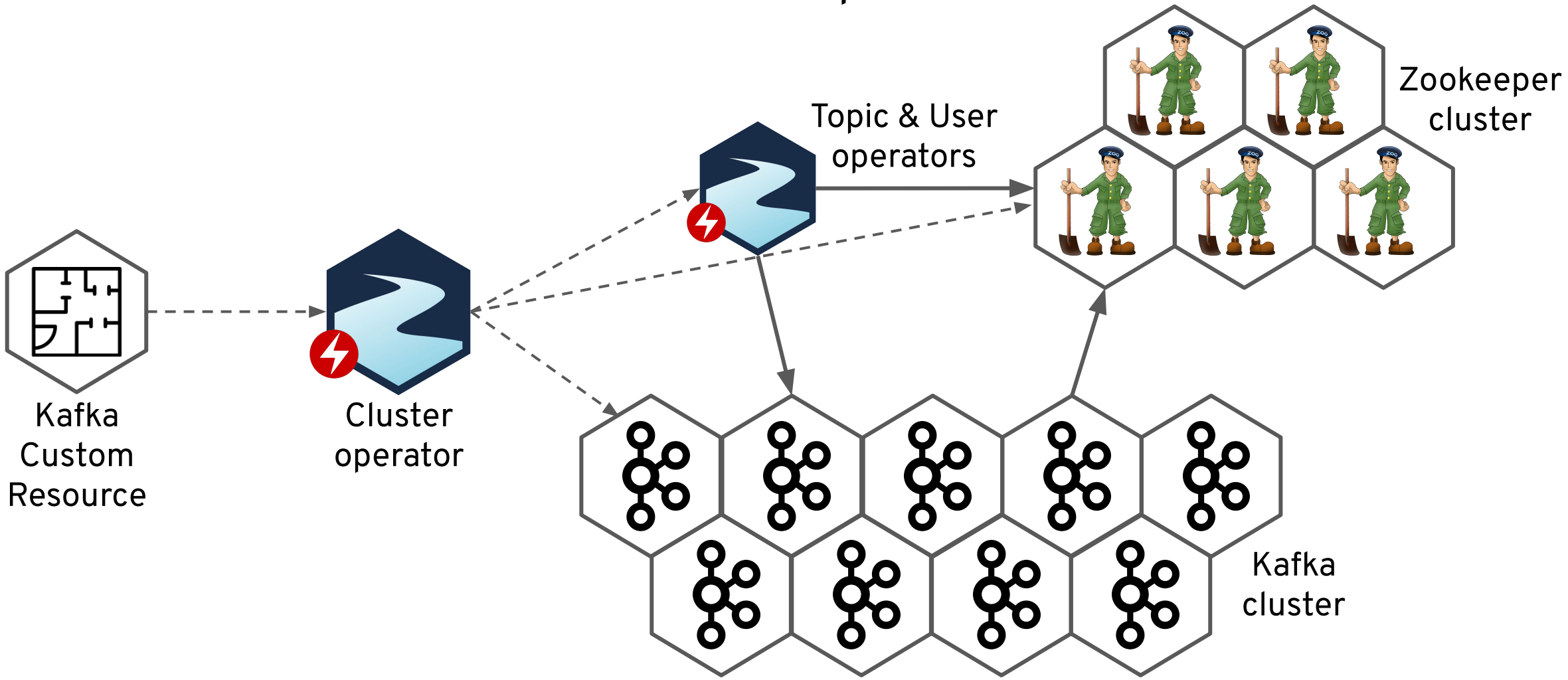
## AMQ Streams on RHEL

- Running Apache Kafka on “bare metal”

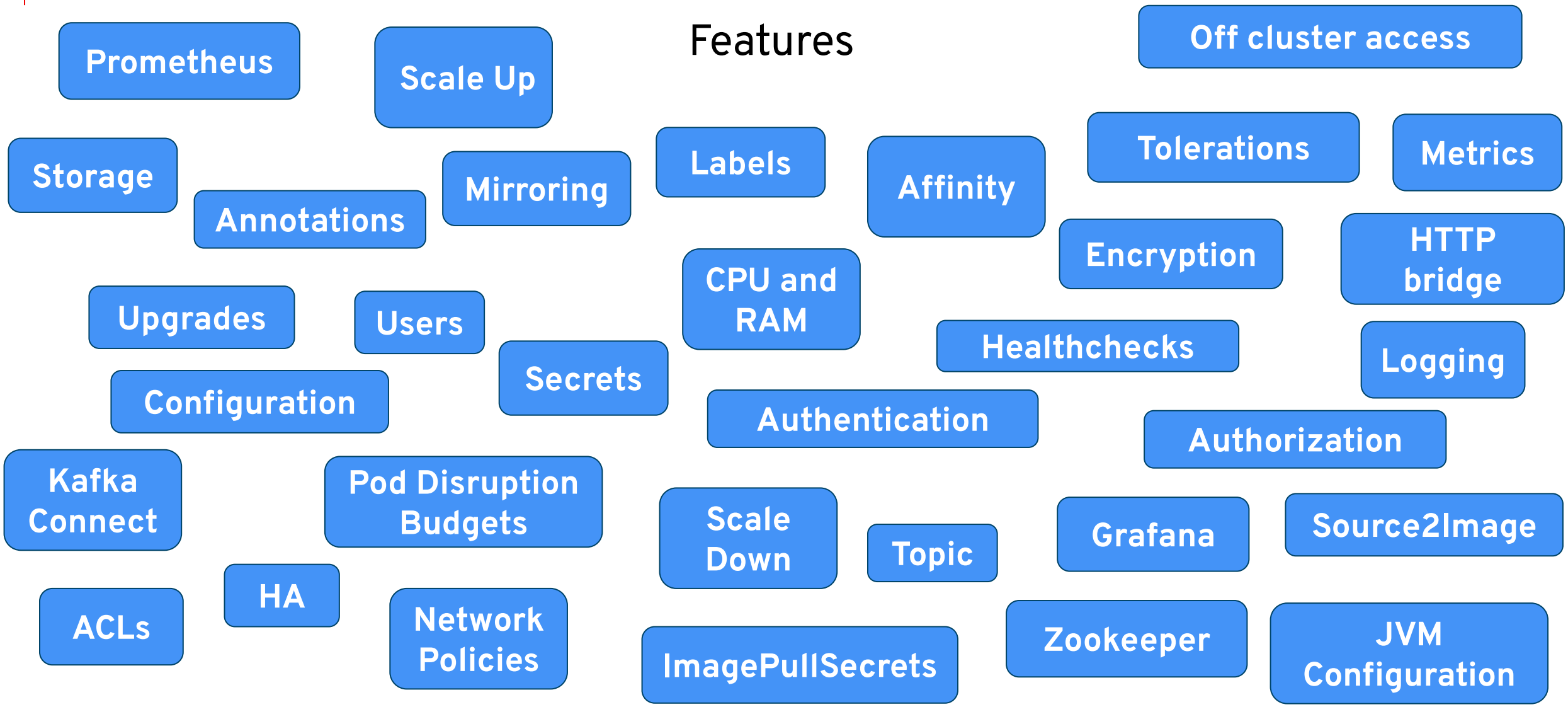
# How to deploy



# How to update



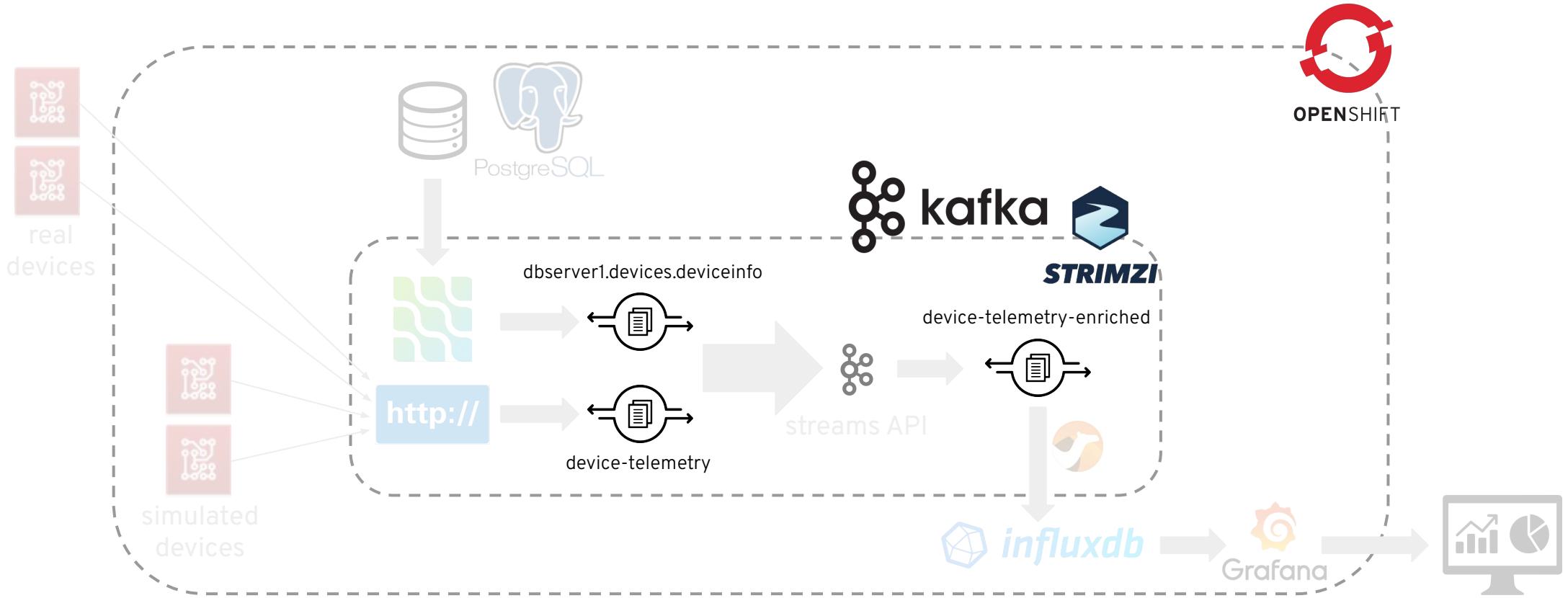
# Features

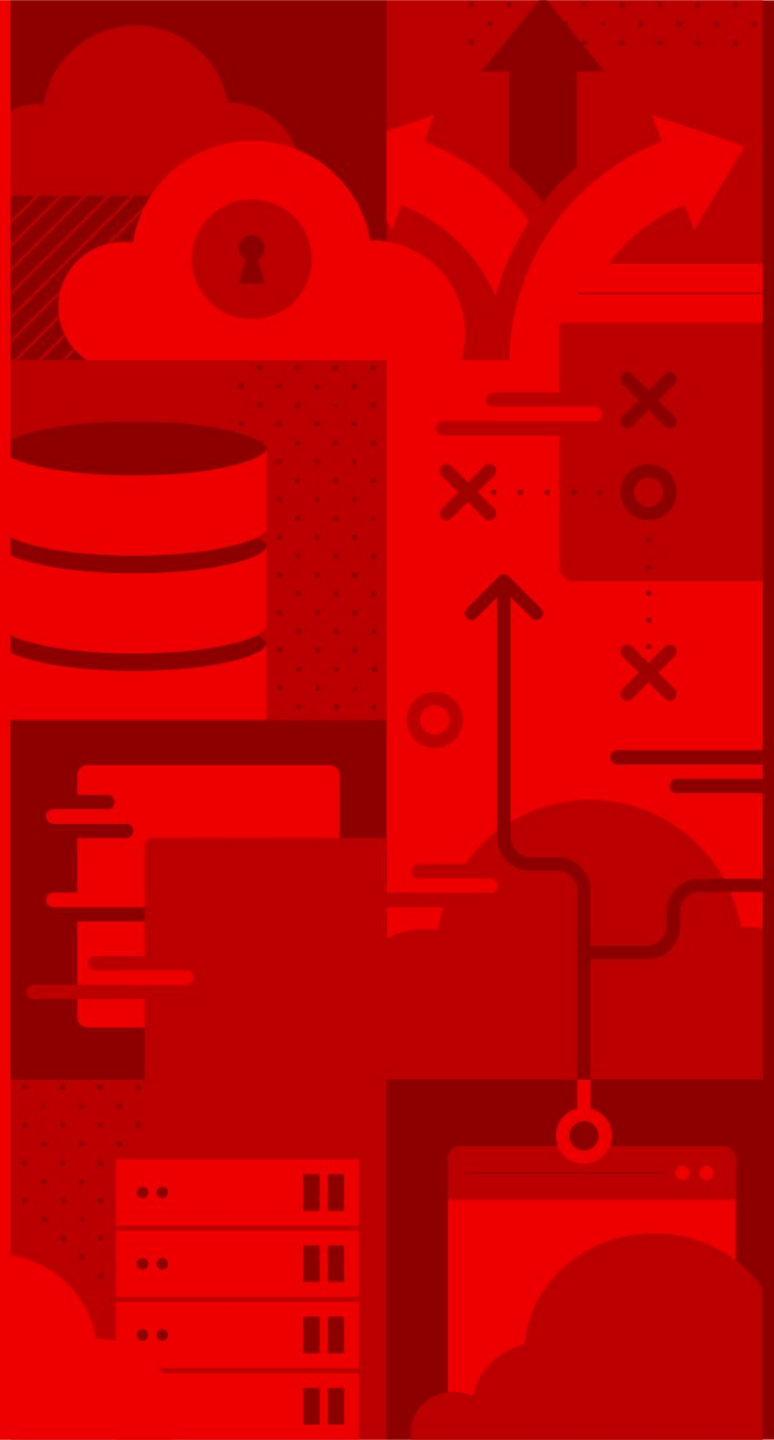





Red Hat

# Deploy the Apache Kafka cluster, creating topics and users





# Apache Kafka Connect & Debezium

# Apache Kafka Connect


**Framework for transferring data between Kafka and other data systems**

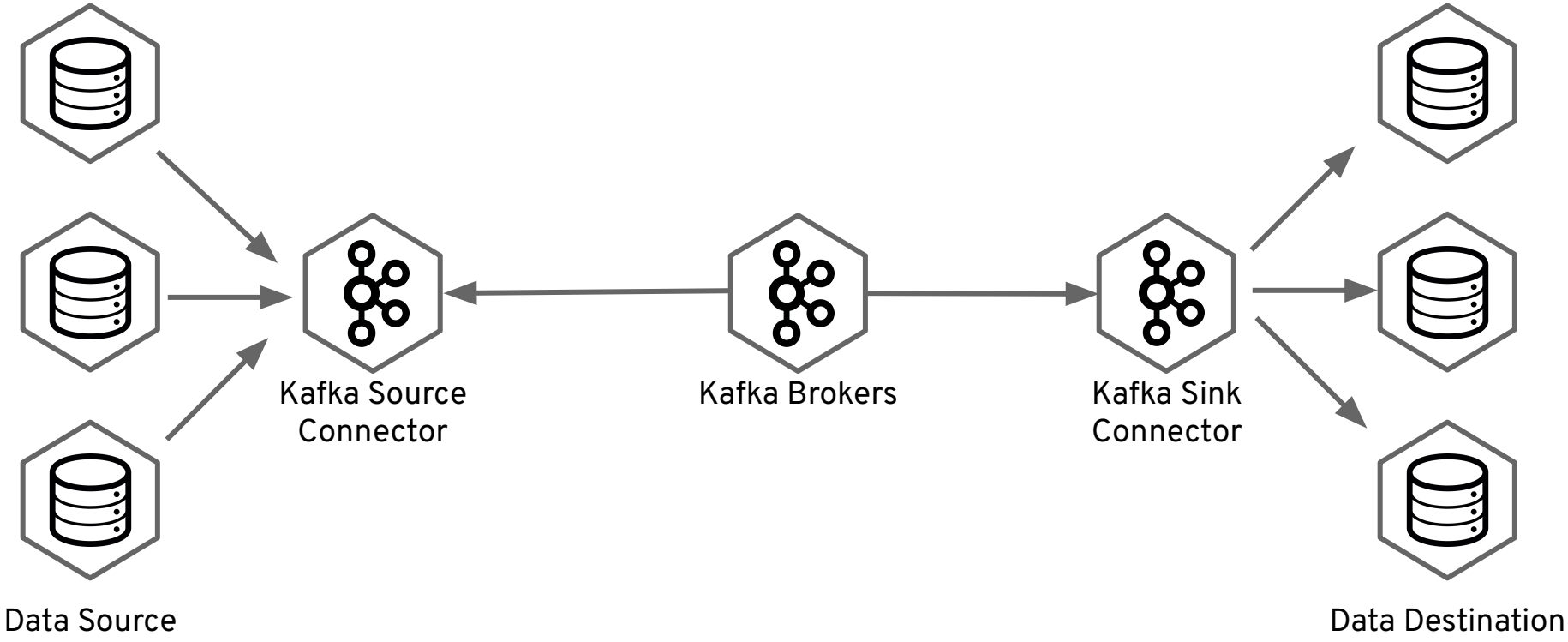
**Addresses more requirements against using just consumer/producer**

- Data conversion (serialization)
- Parallelism/scaling
- Load balancing
- Fault tolerance/failover
- General management

**Connector plugins are deployed into Kafka Connect cluster**

- Well defined API for creating new connectors (with Sink/Source)
- Apache Kafka itself includes only FileSink and FileSource plugins
- Many additional plugins are available outside of Apache Kafka

# Apache Kafka Connect



# Debezium

**Change Data Capture (CDC) connectors for Kafka Connect**

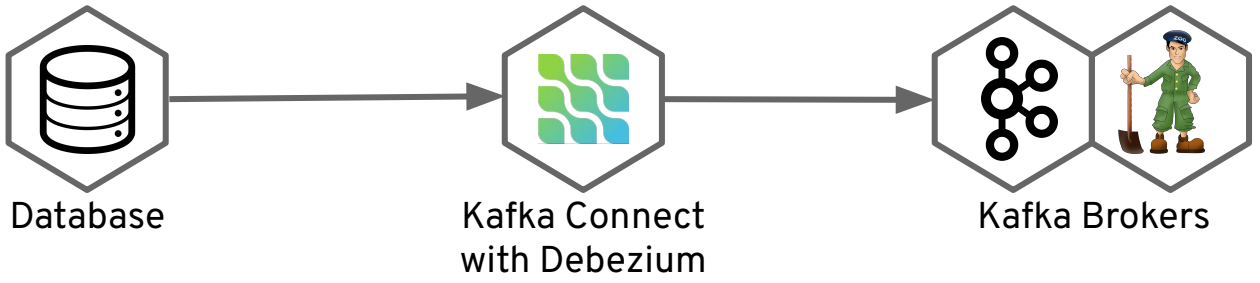
**Connects to DB, reads transaction log and publish to Kafka**

- Supported DB are MySQL, PostgreSQL, MongoDB and SQL Server
- The Kafka messages can be send for example in JSON format

**Makes it easy to integrate DB based applications into Kafka**

- No need to write data to DB and send to Kafka
- Use cases like, for example, microservices integration and data replication

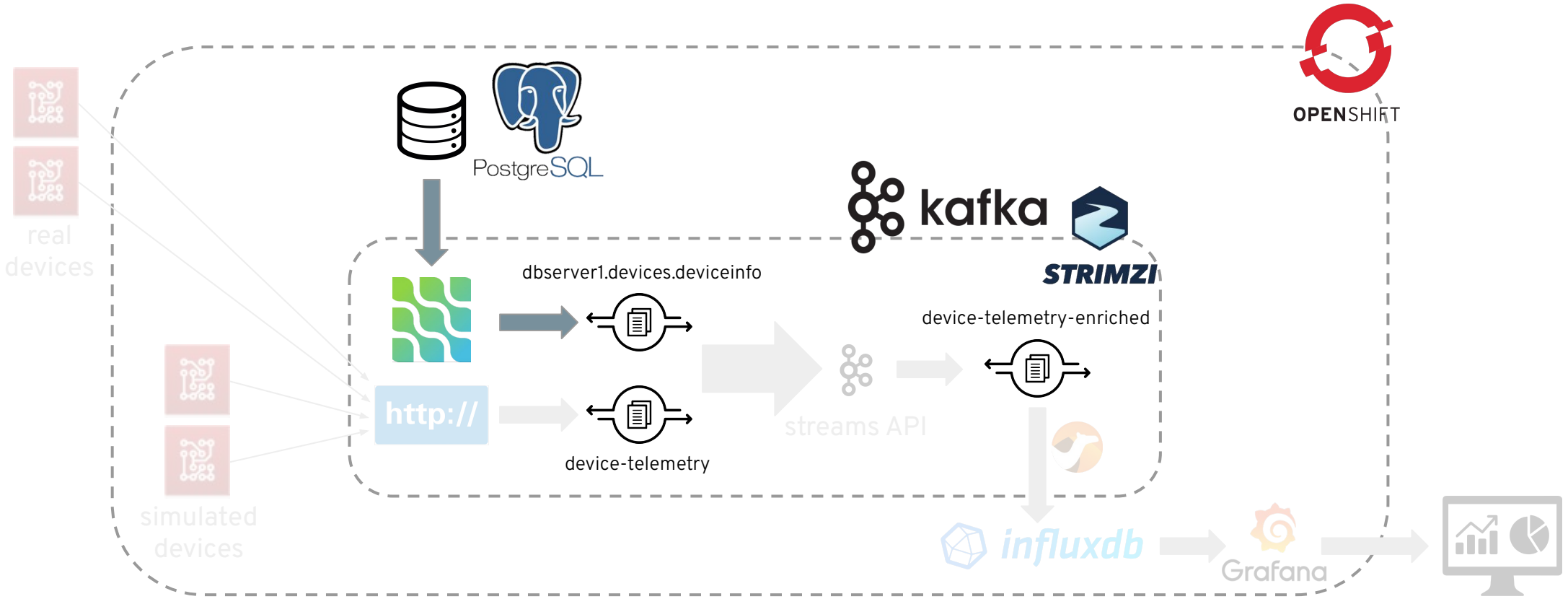
# Debezium

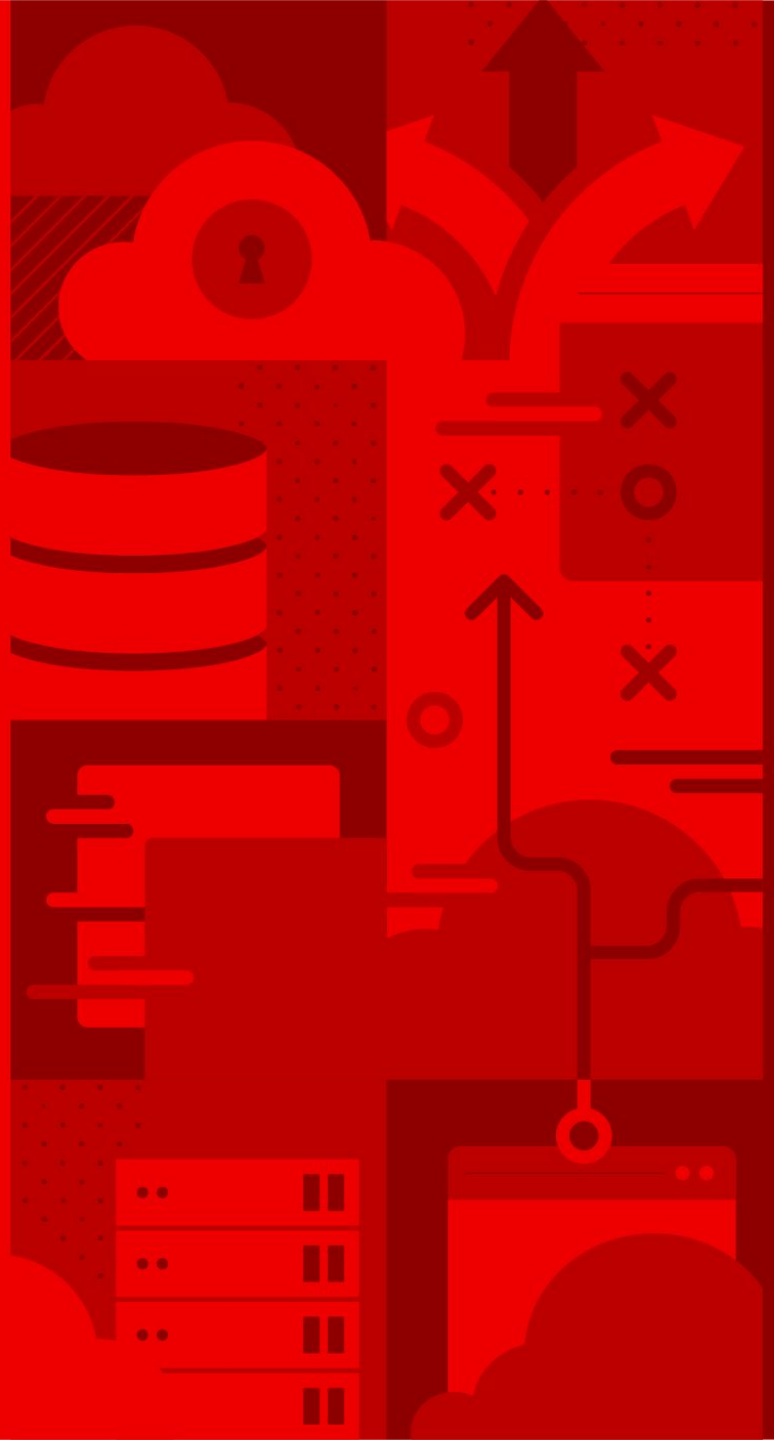


# Debezium- Use cases

- Update or invalidate caches
- Data replication
- Microservices data exchange
- Enable streaming queries
- Update CQRS read models
- Enable full-text search via Elasticsearch, …

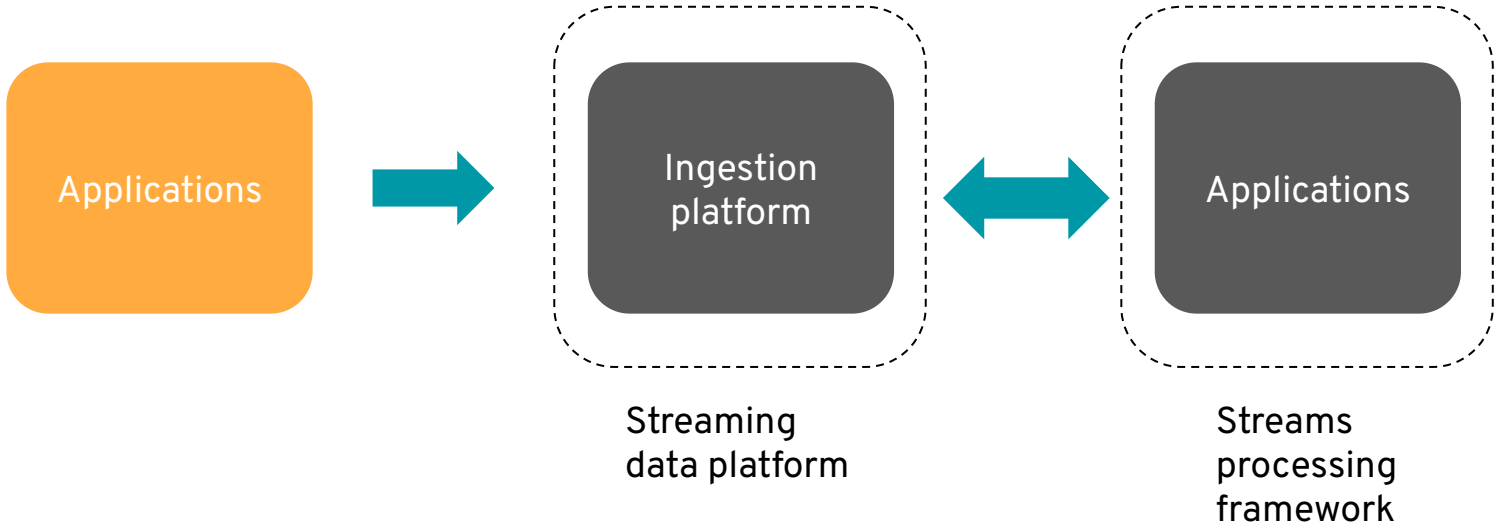
# Deploy PostgreSQL, Kafka Connect and Debezium





# Apache Kafka Streams

# Stream processing



# It's a wild west out there

- Streaming data platform
  - Apache Kafka
- Streams processing frameworks
  - Apache Spark (Streaming)
  - Apache Samza
  - Apache Flink
  - …

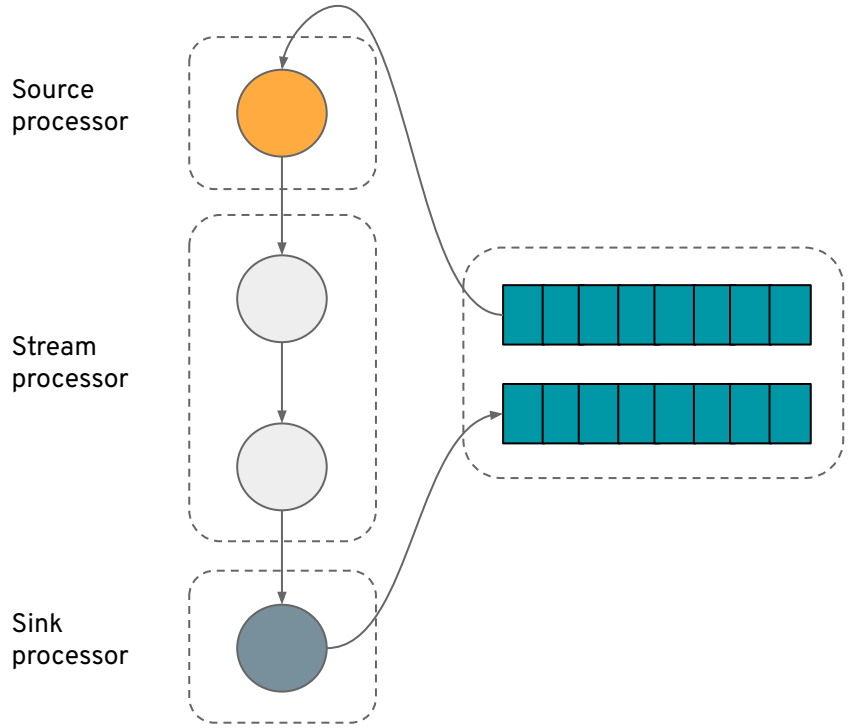



# Let's use just one

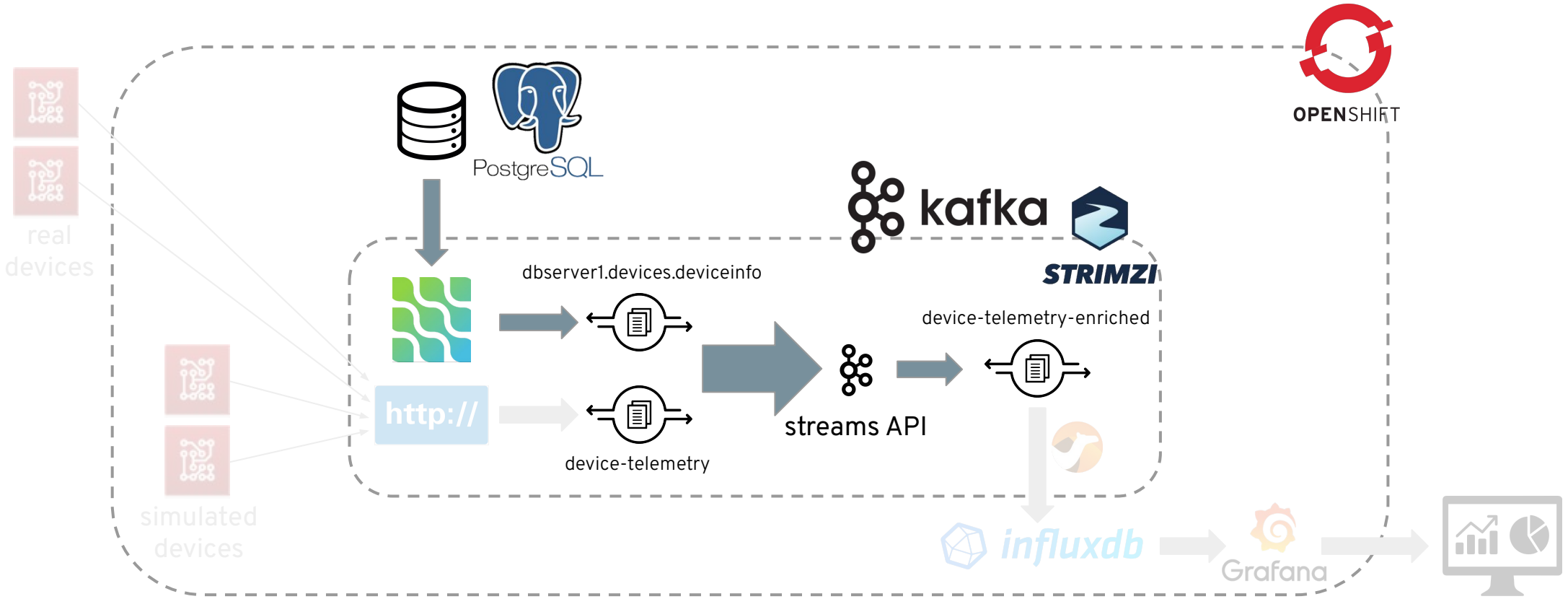

Apache Kafka + Kafka Streams API

# Kafka Streams

- Stream processing framework, just a Java lib!
- Streams are Kafka topics (as input and output)
- Scaling the stream application horizontally
- Creates a topology of processing nodes (filter, map, join etc) acting on a stream
  - Low level processor API
  - High level DSL
- Using “internal” topics (when re-partitioning is needed or for “stateful” transformations)

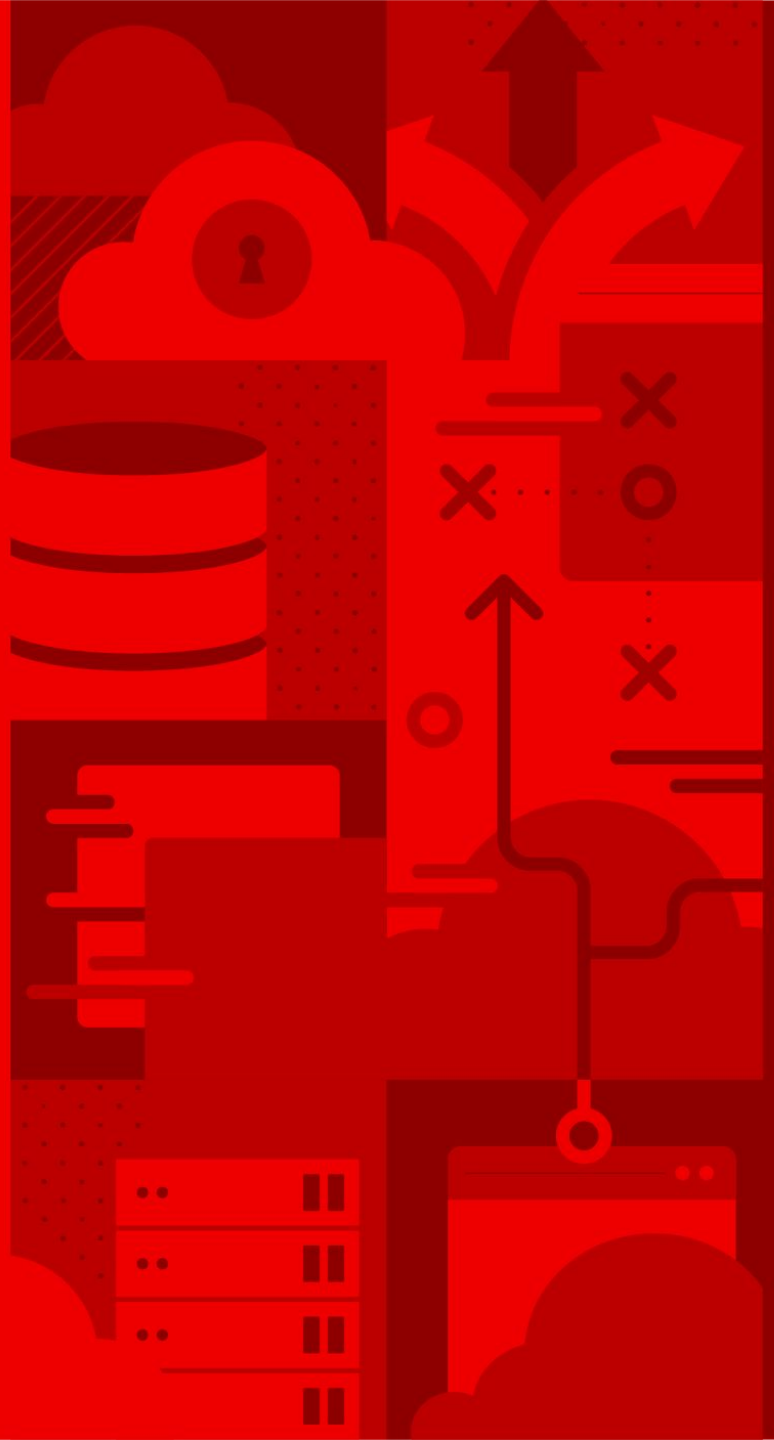


# Deploy Kafka Streams application






Red Hat



# Apache Kafka HTTP Bridge

# Why HTTP to Apache Kafka?



**No native Apache Kafka client implementation for your language**

- HTTP is pretty simple to use
- Quite often the best choice for mobile applications

**All Apache Kafka brokers have to be accessible from clients**

- For security reasons you wouldn't want that
- Using a more controlled HTTP "single" entry point

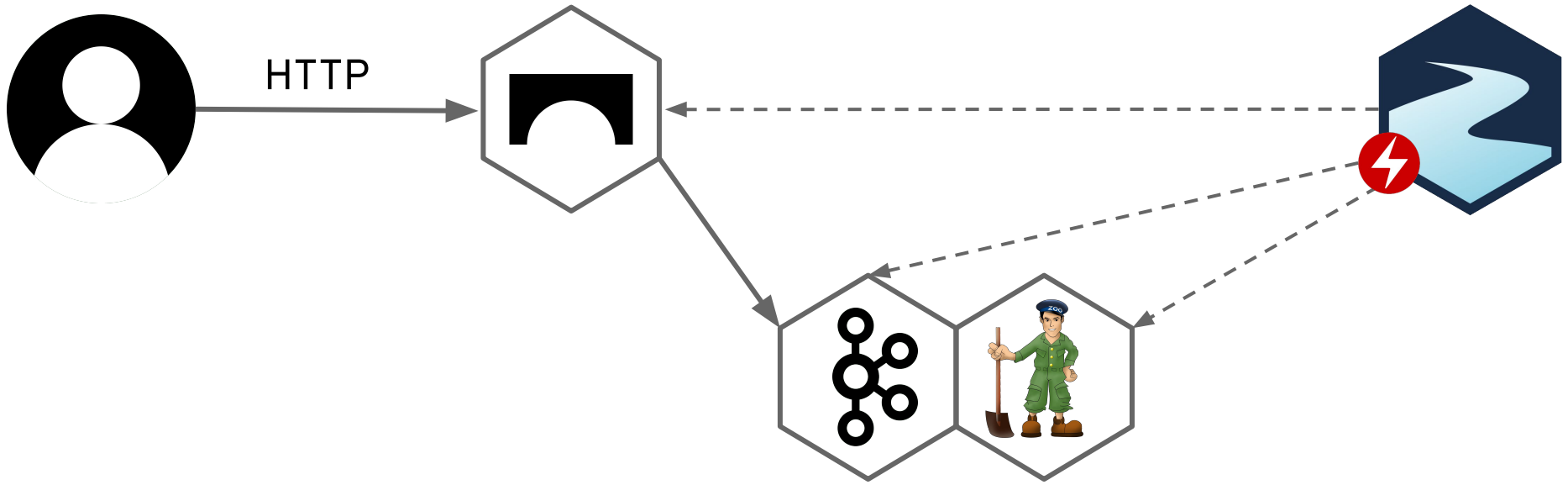
**IoT solutions**

- Constrained devices, too many TCP connections to Apache Kafka
- "Always on" connection doesn't save battery
- Just send/receive when needed with HTTP

# HTTP bridge

- In Tech Preview in AMQ Streams 1.2, GA in 1.3 (in October)
- Available on RHEL and on OCP
  - On OCP deployed through the Cluster Operator
  - On RHEL installed and configured manually
- Used to access Apache Kafka using HTTP REST API
  - Tries to maintain compatibility with the Confluent REST proxy
  - Implements only selected features (for now!)
- AMQP bridge (part of the same upstream code) is not supported!

# Deployed by Cluster Operator



# HTTP bridge - Producers

- The Bridge allows to produce and consume messages
  - Supported formats are JSON and Binary
  - JSON messages can be produced and consumed directly
  - Binary messages can be used also for other formats such as String, it will just be Base64 encoded
- Producers are mostly straight forward
  - POST to `/topics/{topicname}`
  - Content-Type header is important

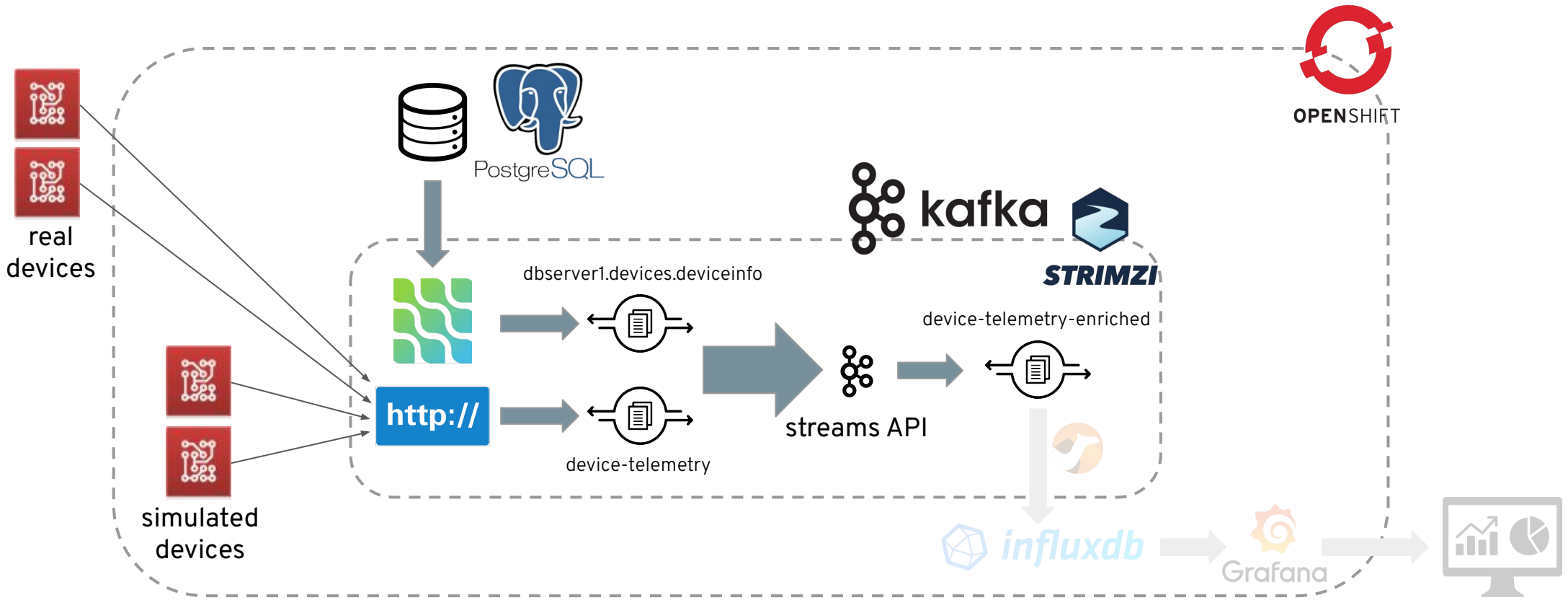
# HTTP bridge - Consumers

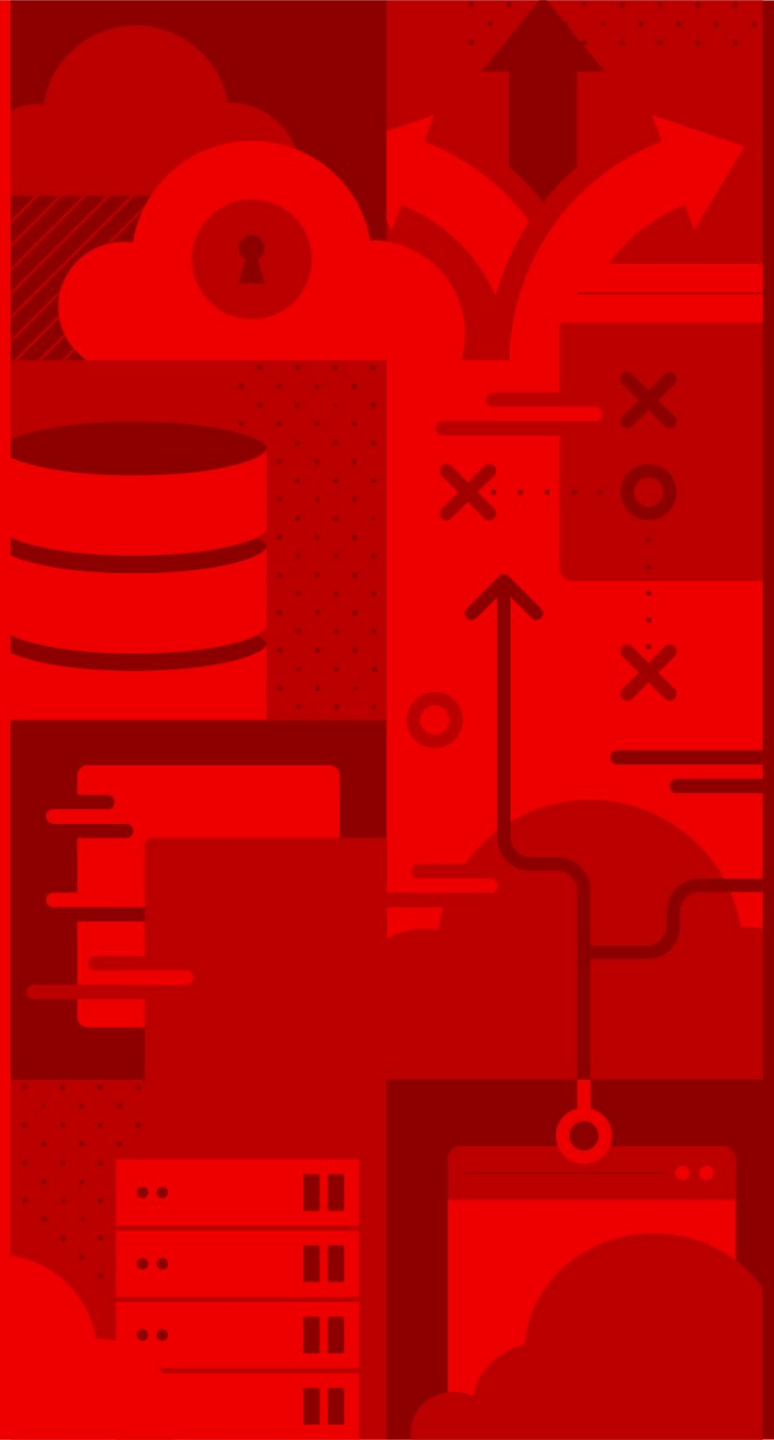
- Consumers are more complicated
- They are stateful
  - Create a consumer with a POST to `/consumers/{groupid}`
  - Subscribe to topics with a POST to `/consumers/{groupid}/instances/{name}/subscription`
  - Consume messages with a GET to `/consumers/{groupid}/instances/{name}/records`
  - Delete the consumer with a DELETE to `/consumers/{groupid}/instances/{name}`
- The statefulness is needed to handle the consumer groups, rebalances etc.
- But it makes scaling complicated
  - For consumers, only 1 replica should be used
  - Multiple separate and independent bridges can be setup to scale

# HTTP bridge - Security

- Connecting to Kafka supports our standard set of protocols
  - mTLS, SASL-SCRAM-SHA, SASL-PLAIN
- The HTTP interface is currently unsecured
  - No encryption (TLS), no authentication
  - API Gateways (3Scale)
  - Network policies
- More security features will be planned for the next releases

# Deploy the Kafka HTTP bridge and devices



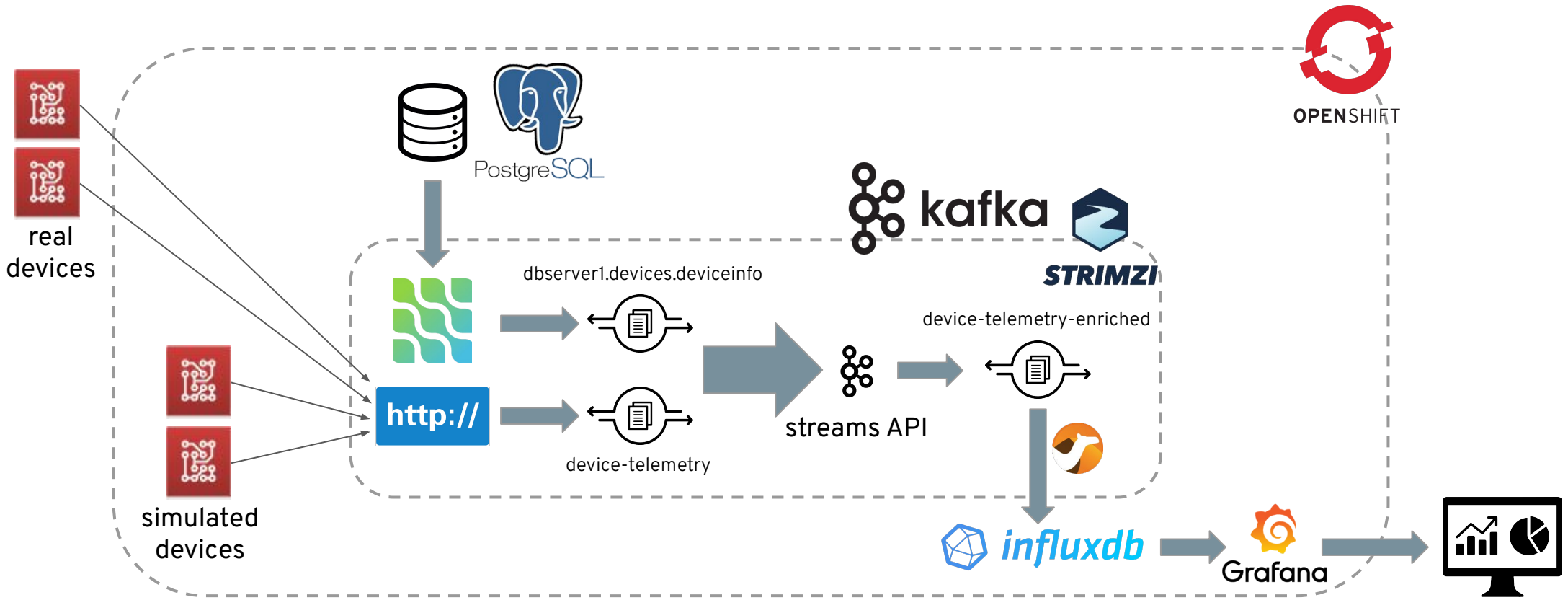


# Bonus module: IoT dashboard with InfluxDB and Grafana

# IoT dashboard

- InfluxDB
  - … as time-series database for ingested data
  - … as datasource for a Grafana dashboard
- Apache Camel based application for “routing” ingested and enriched data to the InfluxDB database

# Deploy InfluxDB and Apache Camel application



# Resources

- AMQ Streams: https://access.redhat.com/products/red-hat-amq#streams
- Strimzi: https://strimzi.io
- Debezium: https://debezium.io/
- Apache Kafka: https://kafka.apache.org/
- CNCF Sandbox projects: https://www.cncf.io/sandbox-projects/
- Workshop: https://github.com/ppatierno/rhte-2019

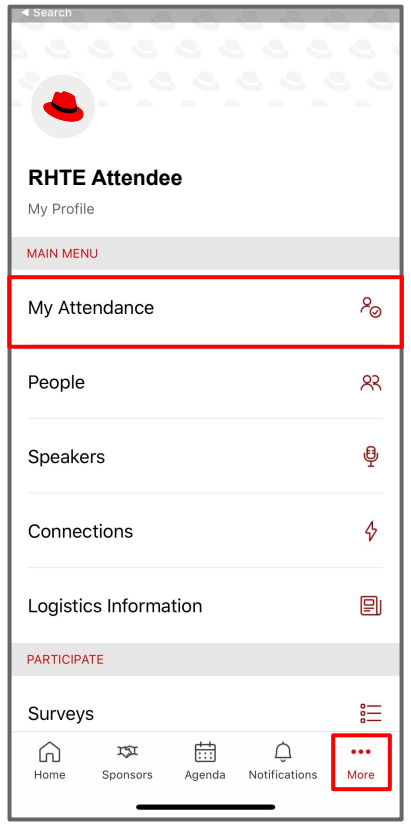
# Code for Attendance + Session Survey

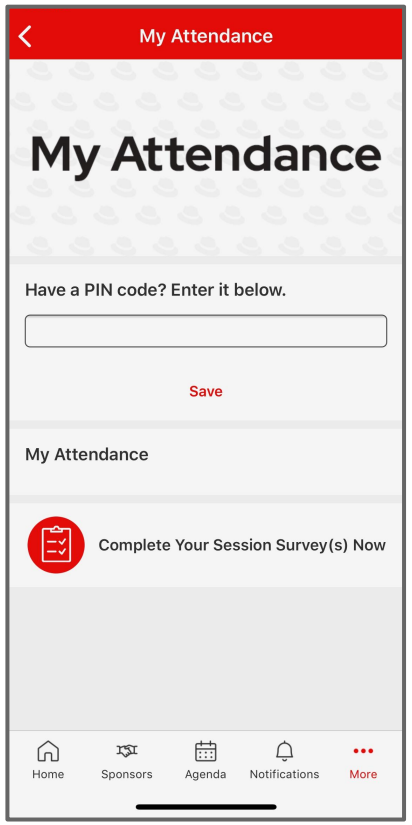
REGION - TIME

1. In the mobile app, go to the ***My Attendance*** page by clicking "***More***" at the bottom navigation bar
2. On the ***My Attendance*** page, please enter the below PIN code in the designated box

**MGDK**

3. Tap *Save* to submit your PIN



# Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat



Red Hat