Red Hat
OpenShift
Application Services

Serverless and Streams Meetup

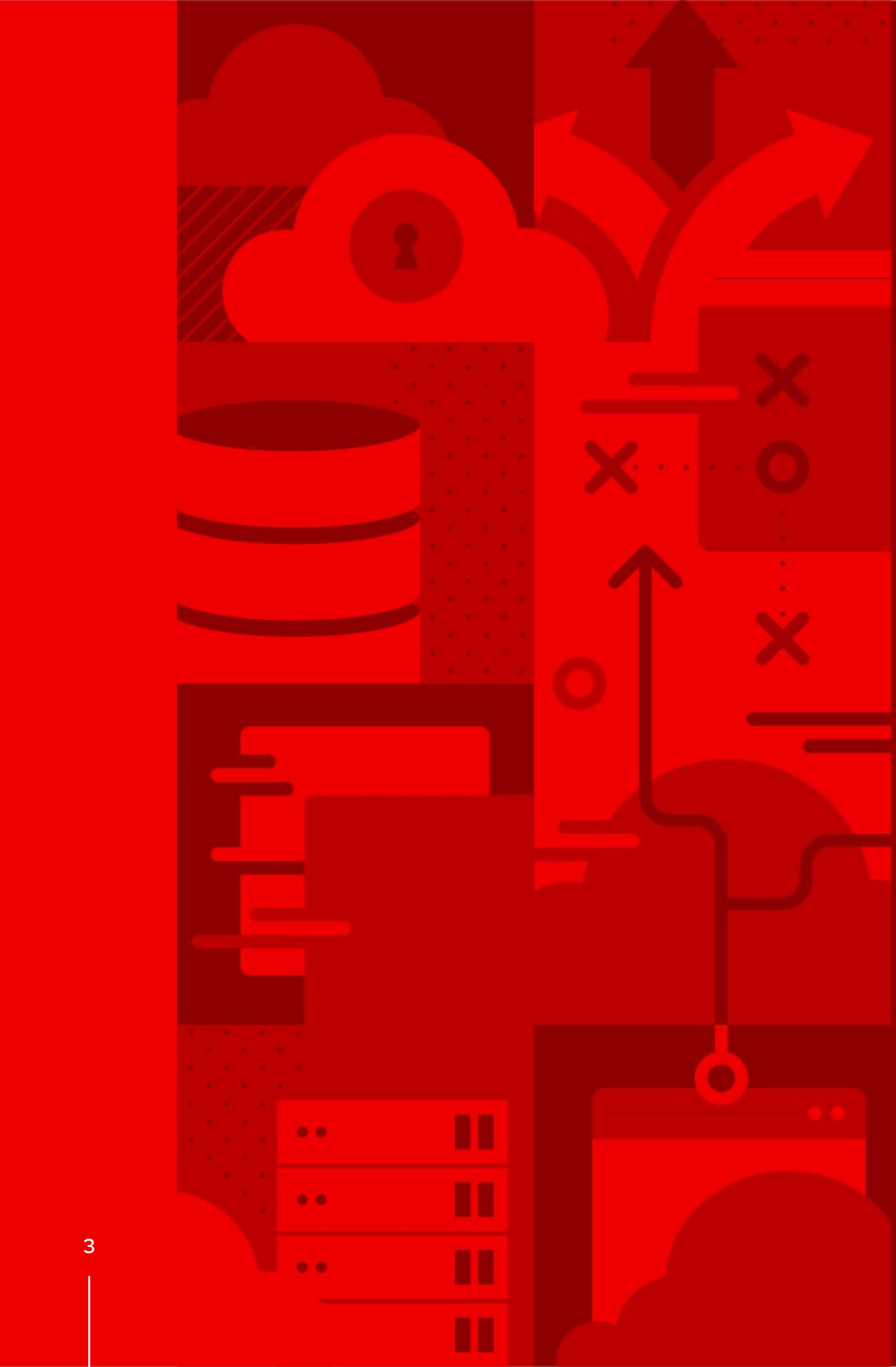
Pramod Padmanabhan
Associate Principal Consultant

Ramy ElEssawy
Specialist Solutions Architect

Red Hat

Agenda

- Apache Kafka Overview
- Serverless Overview
- Demo
- Extended Q&A



Apache Kafka

Red Hat

Red Hat AMQ

Overview

Streams

Streams simplifies the deployment, configuration, management and use of Apache Kafka on OpenShift using the Operator concept

Interconnect

Message router to build large-scale messaging networks using the AMQP protocol to create a redundant application-level messaging network

Broker

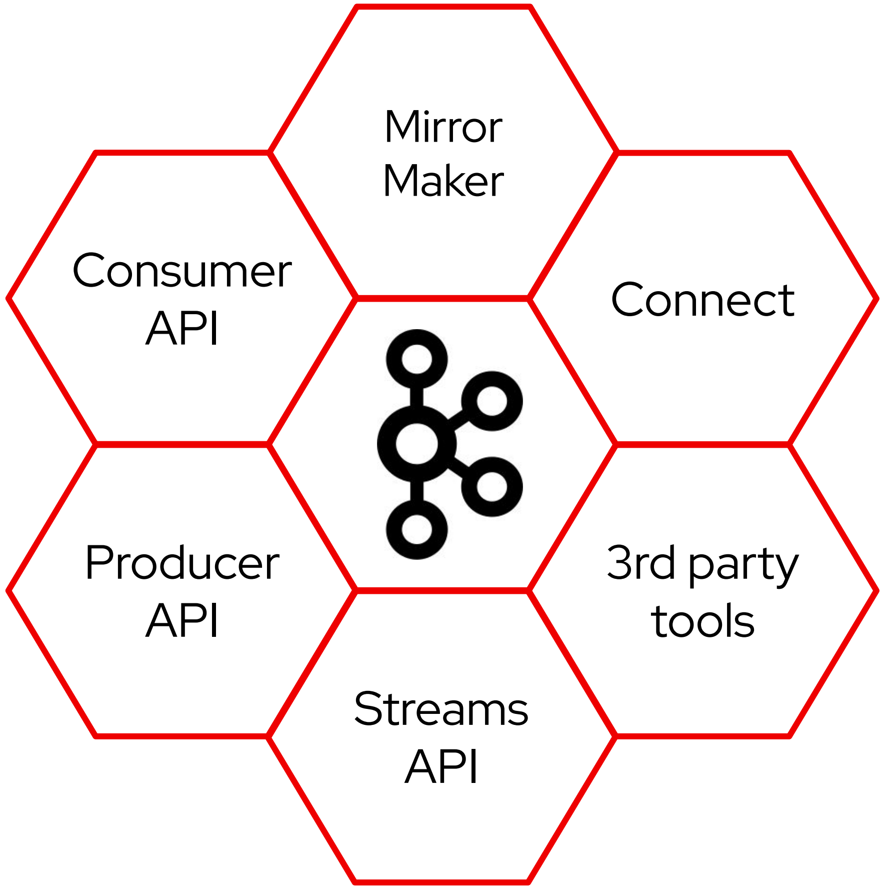
High-performance messaging implementation based on ActiveMQ Artemis

What is Apache Kafka?

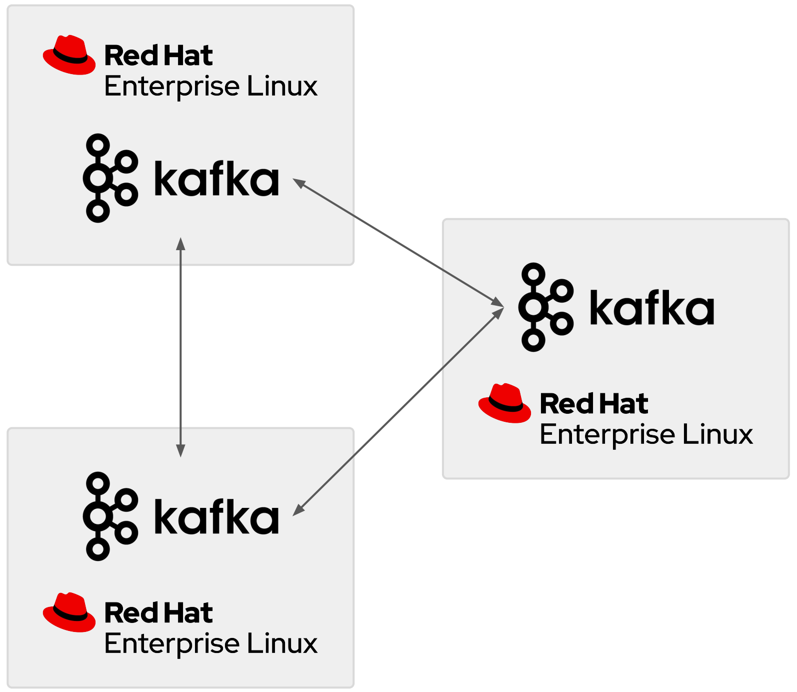
Apache Kafka is a distributed system designed for streams. It is built to be an **horizontally-scalable**, **fault-tolerant**, **commit log**, and allows distributed data streams and stream processing applications.

Apache Kafka ecosystem

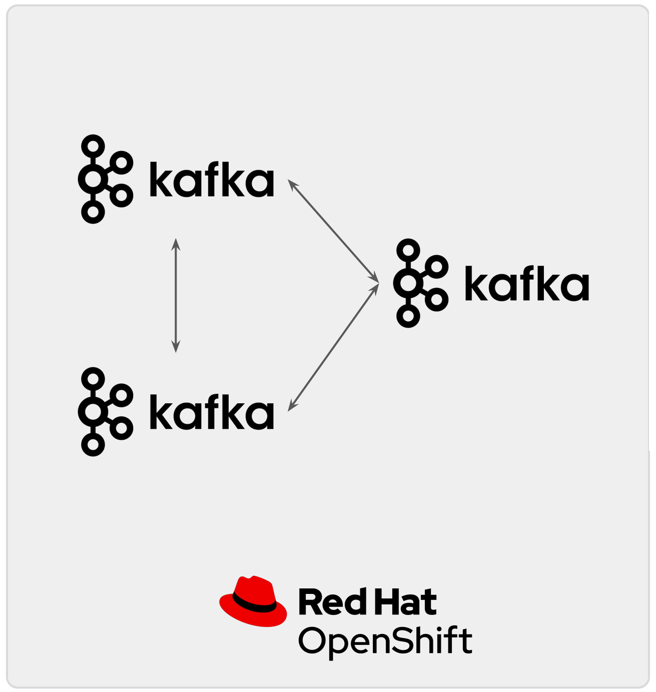
- Kafka Core
  - Broker
  - Producer API, Consumer API, Admin API
  - Management tools
- Kafka Connect
- Kafka Streams API
- Mirror Maker / Mirror Maker 2
- REST Proxy for bridging HTTP and Kafka
- Schema Registry



AMQ Streams Deployment Options



AMQ streams on RHEL



AMQ Streams on OpenShift

STRIMZI

Apache Kafka implementation optimized for OpenShift

Why should you use AMQ Streams?

- Scalability and Performance
  - Designed for horizontal scalability
  - Cluster sizes from few brokers up to 1000s of brokers
    - 3 nodes usually seen as minimum for production (HA, message durability)
    - Most clusters are under 50 nodes
  - Scaling has minimal impact on throughput and latency
  - Adding nodes to running cluster is easy

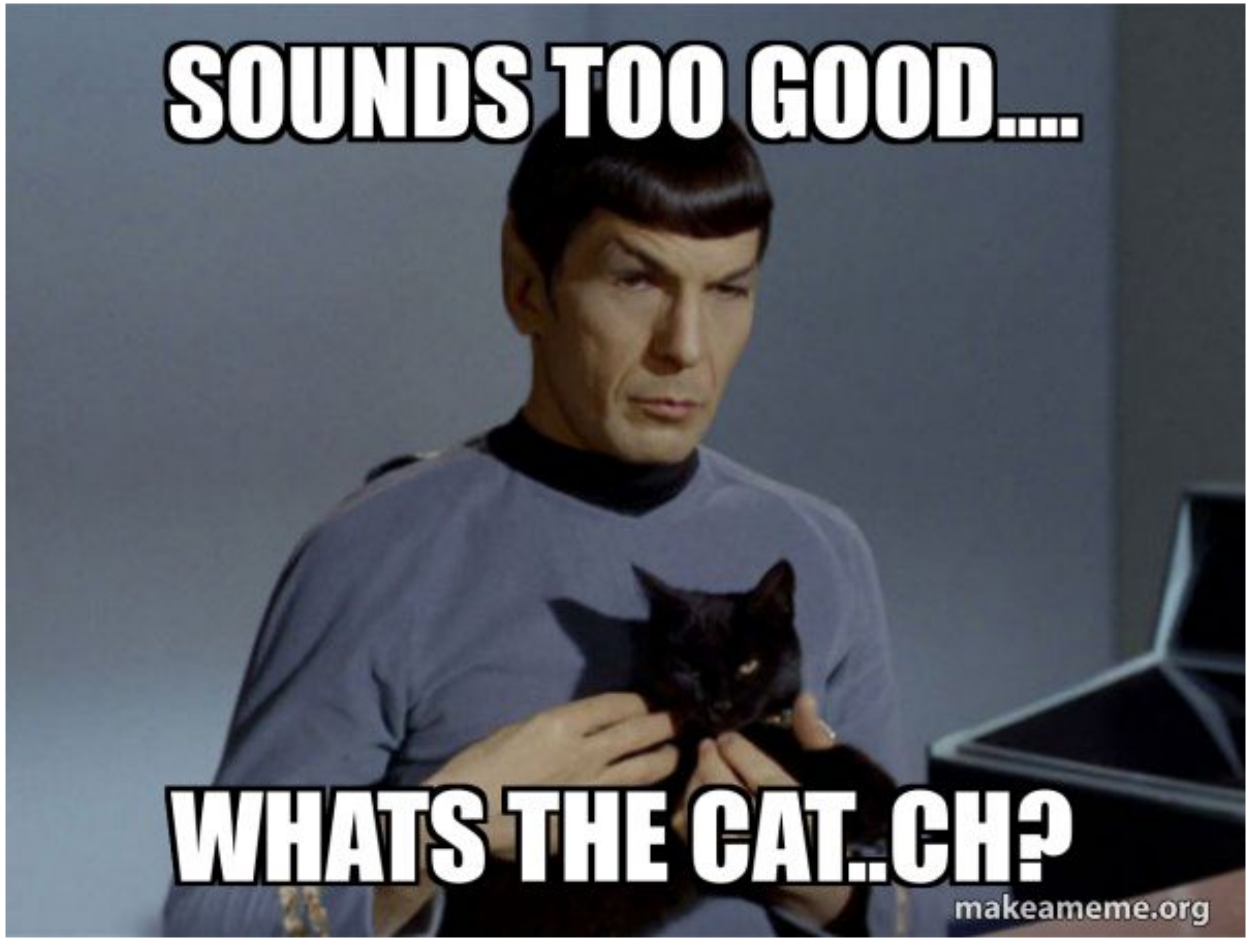
- Message ordering guarantee
  - Messages are written to disk in the same order as received by the broker
  - Messages are read from disk from the requested offset

Red Hat

Why should you use AMQ Streams?

- Message rewind / replay
    - Limited only by available disk space
        - Amount of stored messages has no impact on performance
    - Topic / Partition size has no direct impact on performance
    - Allows to reconstruct application state by replaying the messages
    - Combined with compacted topics allows to use Kafka as key-value store
    - Event sourcing (https://martinfowler.com/eaaDev/EventSourcing.html)
    - Parallel running (https://en.wikipedia.org/wiki/Parallel_running)

What’s the catch?



What’s the catch?

- Proxying Kafka protocol can be difficult
    - Exposing clusters to the “outside world” might be complicated
    - Clients need access to all brokers in the cluster
    - Producers / consumers might need to maintain large number of TCP connections
    - Can be solved by proxying to another protocol
        - HTTP REST proxy
        - AMQP-Kafka bridge
- Kafka protocol cannot be load-balanced
    - To balance the cluster, the topics / partitions have to be reassigned between nodes

- Dumb broker, smart clients
    - Architecture has to be carefully thought through
    - Carefully decide what would be the number of partitions for each topic
        - Too many partitions => Too many brokers / Too much load per broker
        - Too few partitions => Not enough clients running in parallel
    - Removing partitions is not possible
    - How should the partitioning be done
        - What should be the key?
        - Balancing between the ordering guarantee and scalability

Use Cases and Applications

Web Site Activity Tracker

Rebuild user activity tracking pipeline as a set of real-time publish-subscribe feeds.

Metrics

Aggregation of statistics from distributed applications to produce centralized feeds of operational data.

Log Aggregation

Centralized collection of log files in a highly-available store, enabling real-time streaming access to log activity.

Stream Processing

Enables continuous, real-time applications built to react to, process, or transform streams.

Data Integration

Captures streams of events or data changes and feeds these to other data systems.

Source: https://kafka.apache.org/uses

Red Hat
OpenShift Streams
for Apache Kafka

Try Kafka!

No cost - no strings attached

red.ht/TryKafka

Managed Kafka cluster

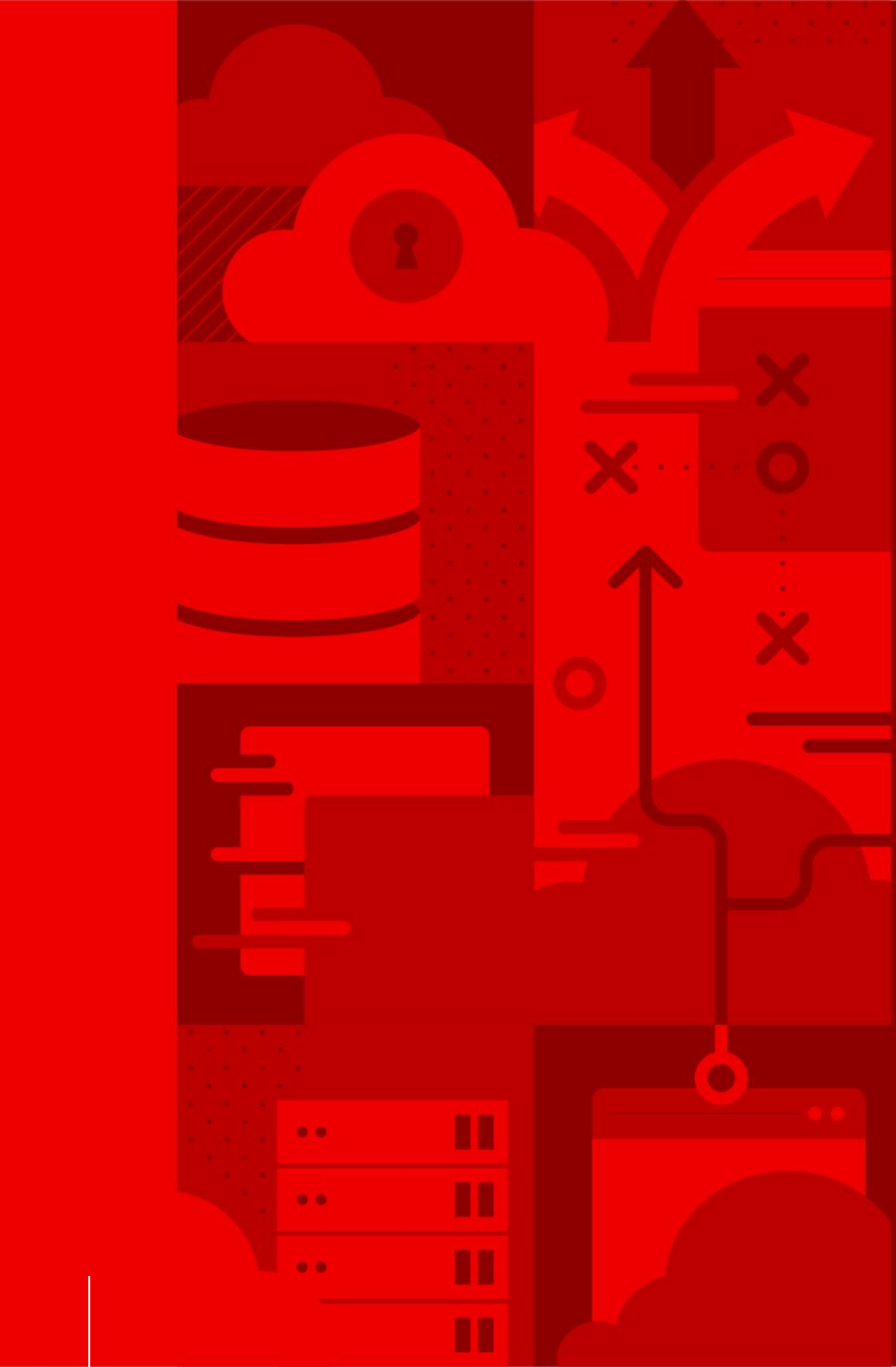
- Spin up your own Kafka cluster
- Create your topics and its partitions
- Connect your producers and consumers
- Get started with the quick starts
- Integrate your apps to the service

Time and resource limited

- Access for 48 hours
- Limited number of topics & brokers

Sign-up

- Go to: red.ht/TryKafka
- Create your own Red Hat account
- Sign-in to try the service



Serverless

Red Hat

What is Serverless?

“Serverless computing refers to the concept of **building and running** applications that **do not require server management**. It describes a finer-grained deployment model where applications, bundled as one or more function are uploaded to a platform and then **executed, scaled, and billed** in response to the exact **demand** needed at the moment.”

- CNCF Definition

https://www.cncf.io/blog/2018/02/14/cncf-takes-first-step-towards-serverless-computing/

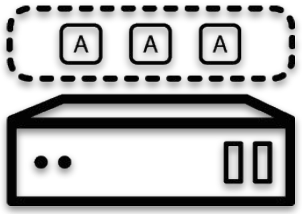
What is Knative ?

- Knative is a **Kubernetes**-based platform to **deploy** and **manage** modern **serverless workloads**
- Open Source project that was started by Google in 2018
- Backed by Google, **Red Hat**, IBM, VMware, TriggerMesh, SAP and more.
    - https://github.com/knative
    - https://knative.dev

What is Knative ?

SERVING

An event-driven model that serves the container with your application and can "scale to zero".



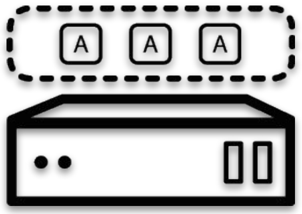
EVENTING

Common infrastructure for consuming and producing events that will stimulate applications.

Knative Serving

SERVING

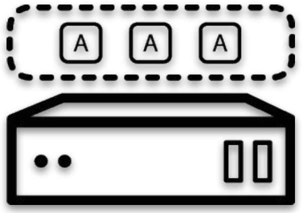
An event-driven model that serves the container with your application and can "scale to zero".

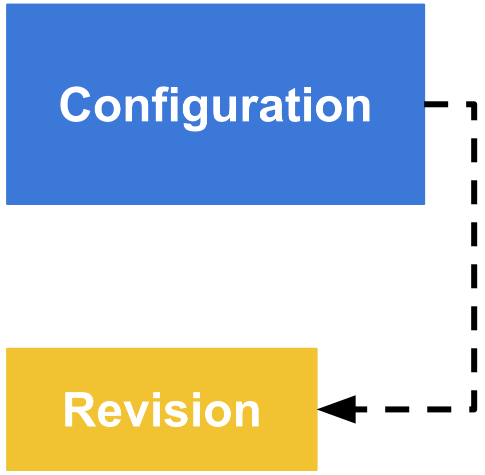


Knative Serving

SERVING

An event-driven model that serves the container with your application and can "scale to zero".

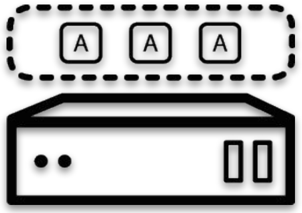


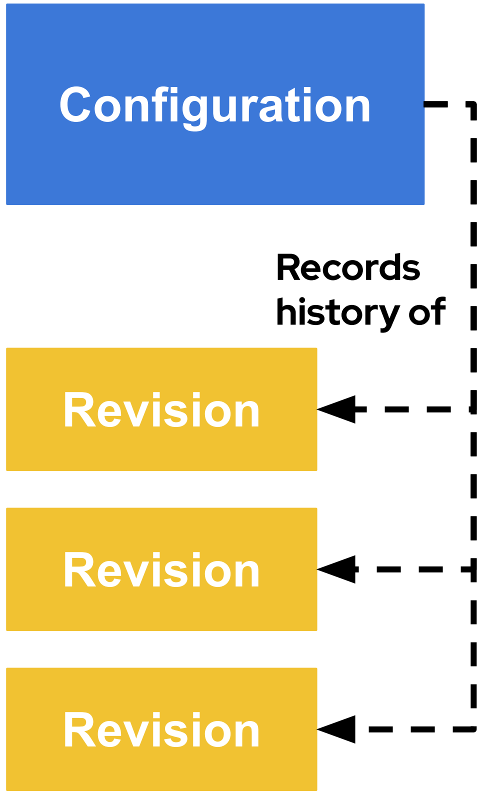


Knative Serving

SERVING

An event-driven model that serves the container with your application and can "scale to zero".

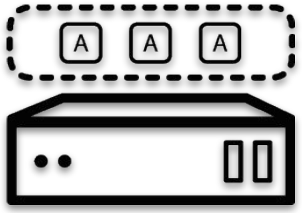


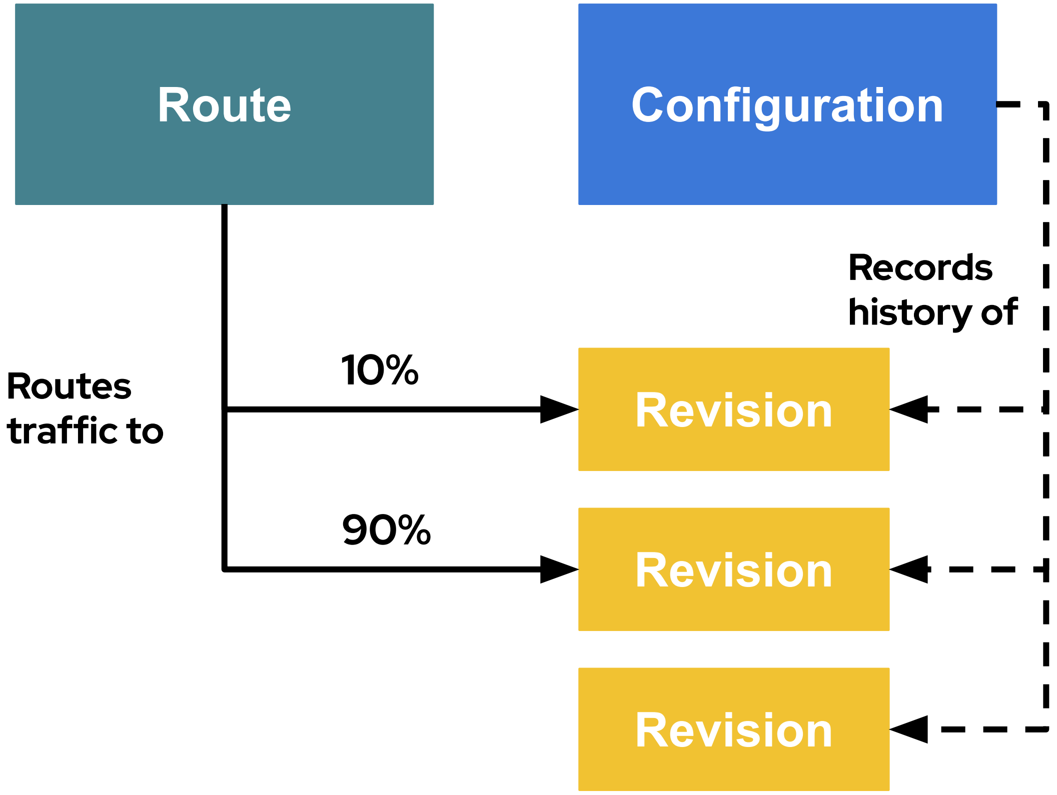


Knative Serving

SERVING

An event-driven model that serves the container with your application and can "scale to zero".

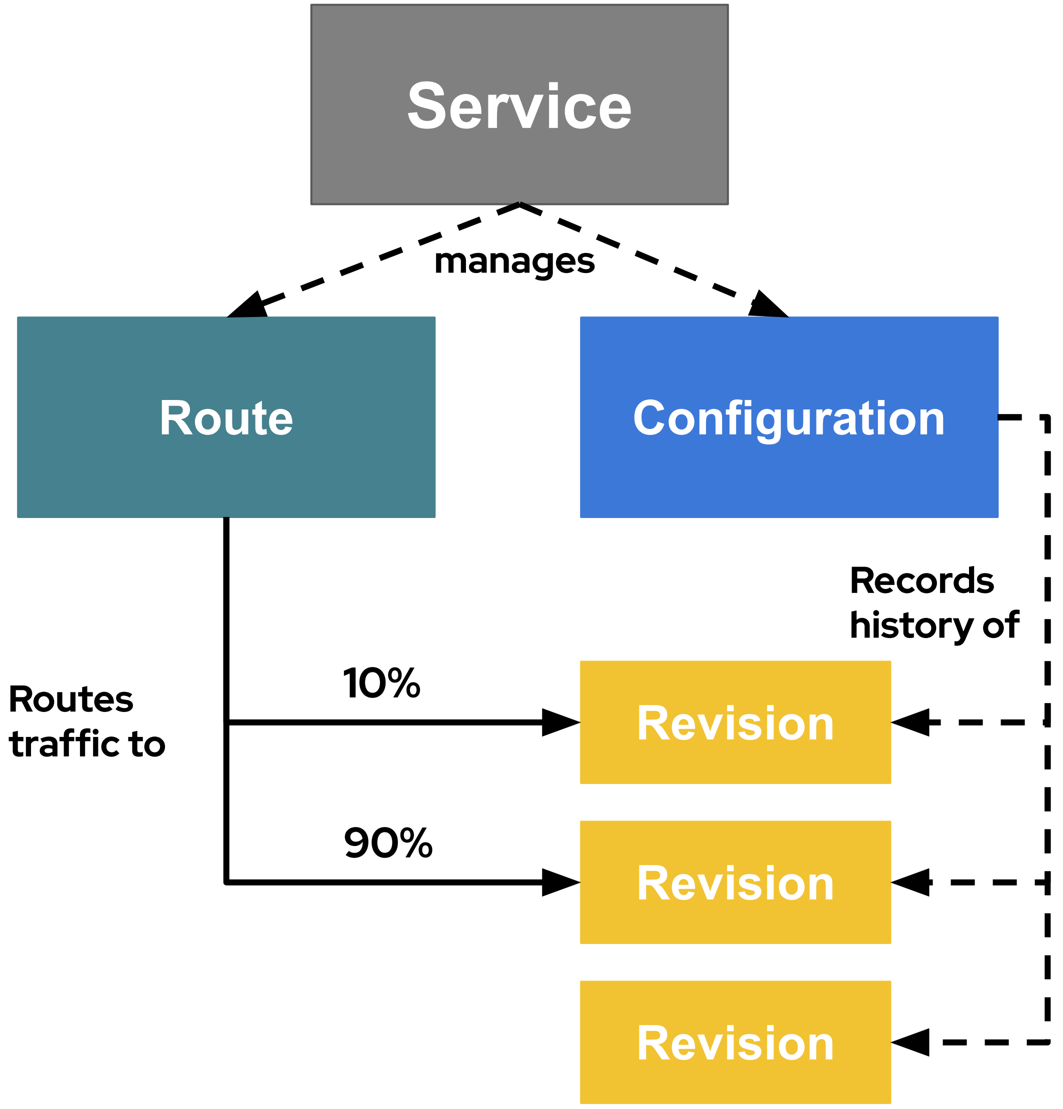




Knative Serving

SERVING

An event-driven model that serves the container with your application and can "scale to zero".



Kubernetes

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
name: frontend
```

Kubernetes

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
name: frontend
labels:
  app: guestbook
spec:
selector:
  matchLabels:
    app: guestbook
    tier: frontend
replicas: 1
```

Kubernetes

```
apiVersion: apps/v1
kind: Deployment
metadata:
name: frontend
labels:
  app: guestbook
spec:
selector:
  matchLabels:
    app: guestbook
    tier: frontend
replicas: 1
template:
  metadata:
    labels:
      app: guestbook
      tier: frontend
```

Kubernetes

```
apiVersion: apps/v1
kind: Deployment
metadata:
name: frontend
labels:
  app: guestbook
spec:
selector:
  matchLabels:
    app: guestbook
    tier: frontend
replicas: 1
template:
  metadata:
    labels:
      app: guestbook
      tier: frontend
  spec:
    containers:
    - image: relessawy/guestbook
      name: guestbook
      resources:
        requests:
          cpu: 100m
          memory: 100Mi
      env:
      - name: GET_HOSTS_FROM
        value: dns
      ports:
      - containerPort: 80
```

30 lines

Kubernetes

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
name: frontend
labels:
  app: guestbook
spec:
selector:
  matchLabels:
    app: guestbook
    tier: frontend
replicas: 1
template:
  metadata:
    labels:
      app: guestbook
      tier: frontend
  spec:
    containers:
    - image: relessawy/guestbook
      name: guestbook
      resources:
        requests:
          cpu: 100m
          memory: 100Mi
      env:
      - name: GET_HOSTS_FROM
        value: dns
      ports:
      - containerPort: 80
```

30 lines

Knative

```yaml
apiVersion: serving.knative.dev/v1
kind: Service
metadata:
name: frontend
labels:
  app: guestbook
spec:
selector:
  matchLabels:
    app: guestbook
    tier: frontend
replicas: 1
template:
  metadata:
    labels:
      app: guestbook
      tier: frontend
  spec:
    containers:
    - image: relessawy/guestbook
      name: guestbook
      resources:
        requests:
          cpu: 100m
          memory: 100Mi
      env:
      - name: GET_HOSTS_FROM
        value: dns
      ports:
      - containerPort: 80
```

Kubernetes

```
apiVersion: apps/v1
kind: Deployment
metadata:
name: frontend
labels:
  app: guestbook
spec:
selector:
  matchLabels:
    app: guestbook
    tier: frontend
replicas: 1
template:
  metadata:
    labels:
      app: guestbook
      tier: frontend
  spec:
    containers:
    - image: relessawy/guestbook
      name: guestbook
      resources:
        requests:
          cpu: 100m
          memory: 100Mi
      env:
      - name: GET_HOSTS_FROM
        value: dns
      ports:
      - containerPort: 80
```

30 lines

Knative

```
apiVersion: apps/v1
kind: Deployment
metadata:
name: frontend
labels:
  app: guestbook
spec:
selector:
  matchLabels:
    app: guestbook
    tier: frontend
replicas: 1
template:
  metadata:
    labels:
      app: guestbook
      tier: frontend
  spec:
    containers:
    - image: relessawy/guestbook
      name: guestbook
      resources:
        requests:
          cpu: 100m
          memory: 100Mi
      env:
      - name: GET_HOSTS_FROM
        value: dns
      ports:
      - containerPort: 80
```

Kubernetes

```
apiVersion: apps/v1
kind: Deployment
metadata:
name: frontend
labels:
  app: guestbook
spec:
selector:
  matchLabels:
    app: guestbook
    tier: frontend
replicas: 1
template:
  metadata:
    labels:
      app: guestbook
      tier: frontend
  spec:
    containers:
    - image: relessawy/guestbook
      name: guestbook
      resources:
        requests:
          cpu: 100m
          memory: 100Mi
      env:
      - name: GET_HOSTS_FROM
        value: dns
      ports:
      - containerPort: 80
```

30 lines

Knative

```
apiVersion: serving.knative.dev/v1
kind: Service
metadata:
name: frontend
spec:
template:
  metadata:
    labels:
      app: guestbook
      tier: frontend
  spec:
    containers:
    - image: relessawy/guestbook
      resources:
        requests:
          cpu: 100m
          memory: 100Mi
      env:
      - name: GET_HOSTS_FROM
        value: dns
      ports:
      - containerPort: 80
```

22 lines

Kubernetes

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
name: frontend
labels:
  app: guestbook
spec:
selector:
  matchLabels:
    app: guestbook
    tier: frontend
replicas: 1
template:
  metadata:
    labels:
      app: guestbook
      tier: frontend
  spec:
    containers:
    - image: relessawy/guestbook
      name: guestbook
      resources:
        requests:
          cpu: 100m
          memory: 100Mi
      env:
      - name: GET_HOSTS_FROM
        value: dns
      ports:
      - containerPort: 80
```

```yaml
---
apiVersion: v1
kind: Service
metadata:
name: frontend-service
labels:
  app: guestbook
  tier: frontend
spec:
ports:
- port: 80
selector:
  app: guestbook
  tier: frontend
---
apiVersion: route.openshift.io/v1
kind: Route
metadata:
name: frontend-route
spec:
to:
  kind: Service
  name: frontend-service
```

51 lines

Knative

```yaml
apiVersion: serving.knative.dev/v1
kind: Service
metadata:
name: frontend
spec:
template:
  metadata:
    labels:
      app: guestbook
      tier: frontend
  spec:
    containers:
    - image: relessawy/guestbook
      resources:
        requests:
          cpu: 100m
          memory: 100Mi
      env:
      - name: GET_HOSTS_FROM
        value: dns
      ports:
      - containerPort: 80
```

22 lines

Kubernetes

```
apiVersion: apps/v1
kind: Deployment
metadata:
name: frontend
labels:
 app: guestbook
spec:
selector:
 matchLabels:
  app: guestbook
  tier: frontend
replicas: 1
template:
 metadata:
  labels:
   app: guestbook
   tier: frontend
 spec:
  containers:
  - image: relessawy/guestbook
   name: guestbook
   resources:
    requests:
     cpu: 100m
     memory: 100Mi
   env:
   - name: GET_HOSTS_FROM
    value: dns
   ports:
   - containerPort: 80
```

```
apiVersion: extensions/v1beta1
kind: HorizontalPodAutoscaler
metadata:
 name: guestbook
 namespace: default
spec:
 scaleRef:
   kind: ReplicationController
   name: guestbook
   namespace: default
   subresource: scale
 minReplicas: 1
 maxReplicas: 10
 cpuUtilization:
   targetPercentage: 50
```

```
metadata:
name: frontend-service
labels:
 app: guestbook
 tier: frontend
spec:
ports:
- port: 80
selector:
 app: guestbook
 tier: frontend
---
apiVersion: route.openshift.io/v1
kind: Route
metadata:
name: frontend-route
spec:
to:
 kind: Service
 name: frontend-service
```

66 lines

Knative

```
apiVersion: serving.knative.dev/v1
kind: Service
metadata:
name: frontend
spec:
template:
 metadata:
  labels:
   app: guestbook
   tier: frontend
 spec:
  containers:
  - image: relessawy/guestbook
   resources:
    requests:
     cpu: 100m
     memory: 100Mi
   env:
   - name: GET_HOSTS_FROM
    value: dns
   ports:
   - containerPort: 80
```

22 lines

Knative Eventing

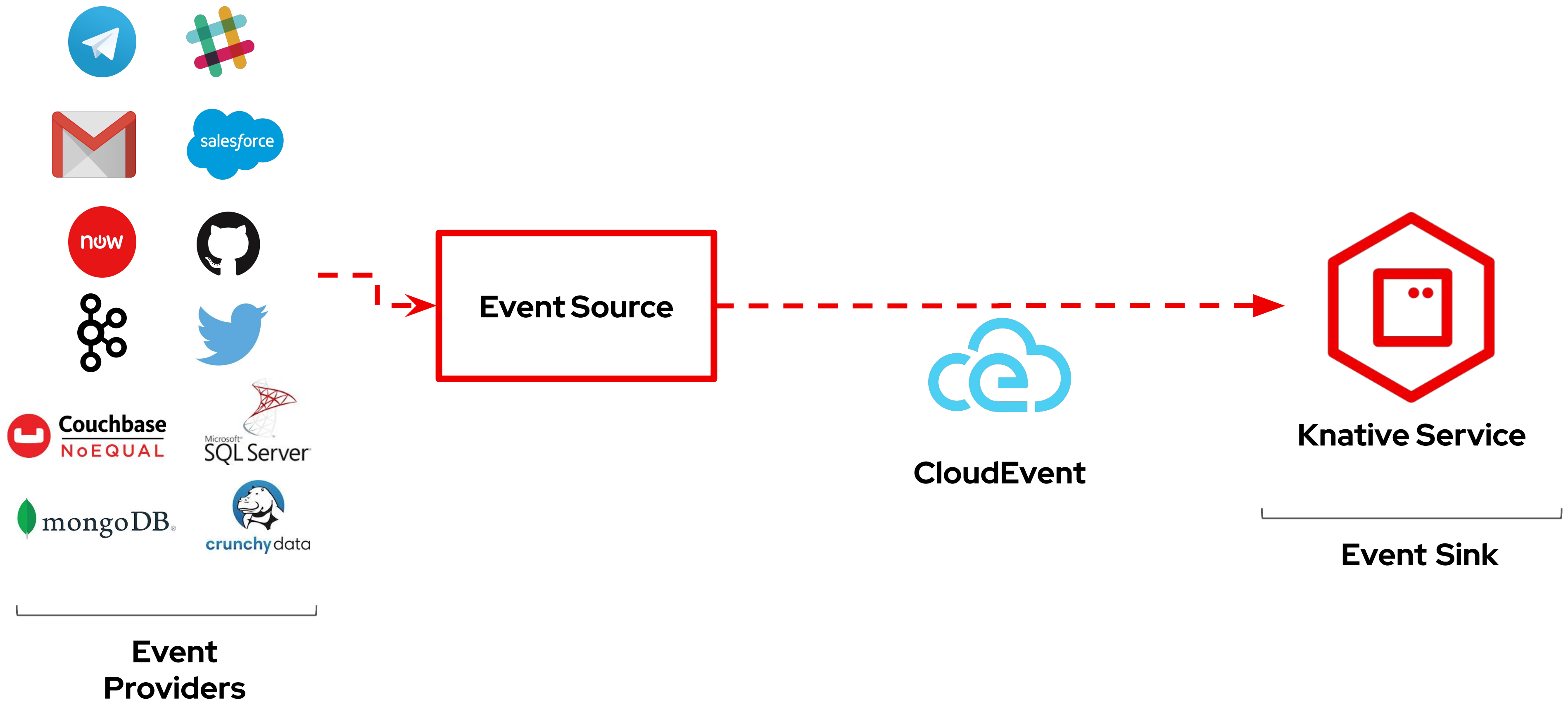
- Based on CloudEvents (CNCF Standard)
- Flexible routing of events from Source to Sink

CloudEvents

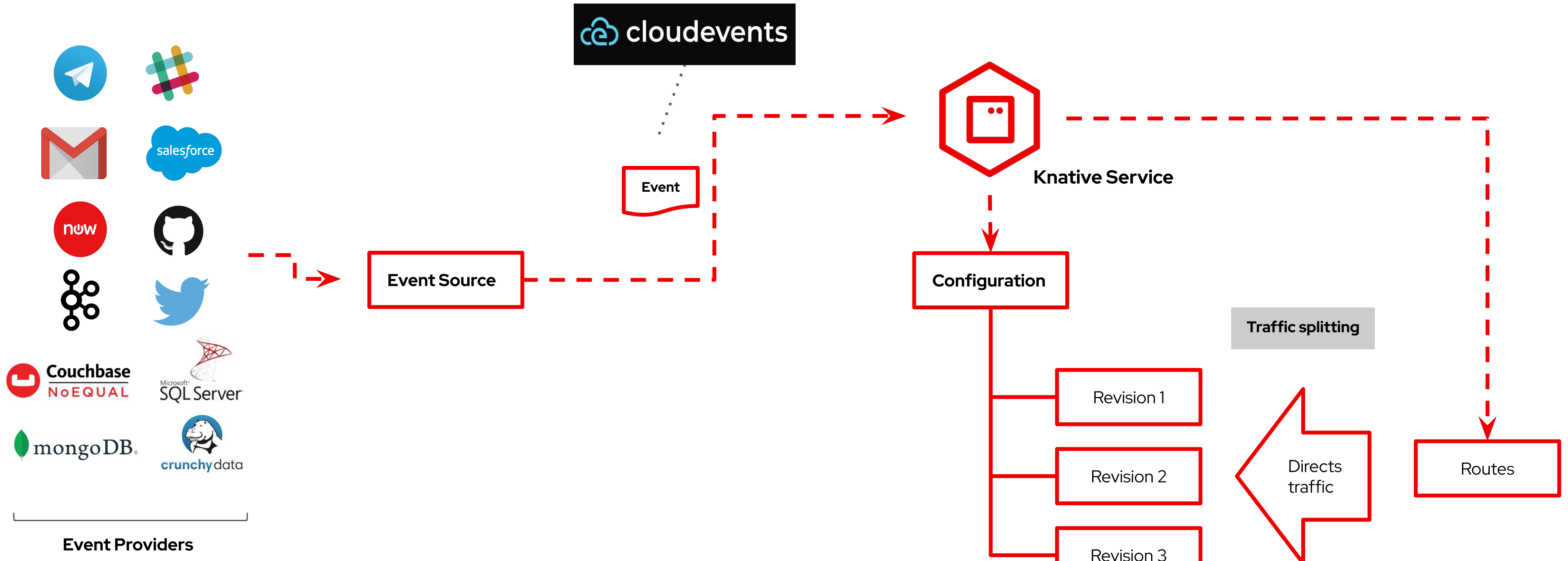
EVENTING

Common infrastructure for consuming and producing events that will stimulate applications.

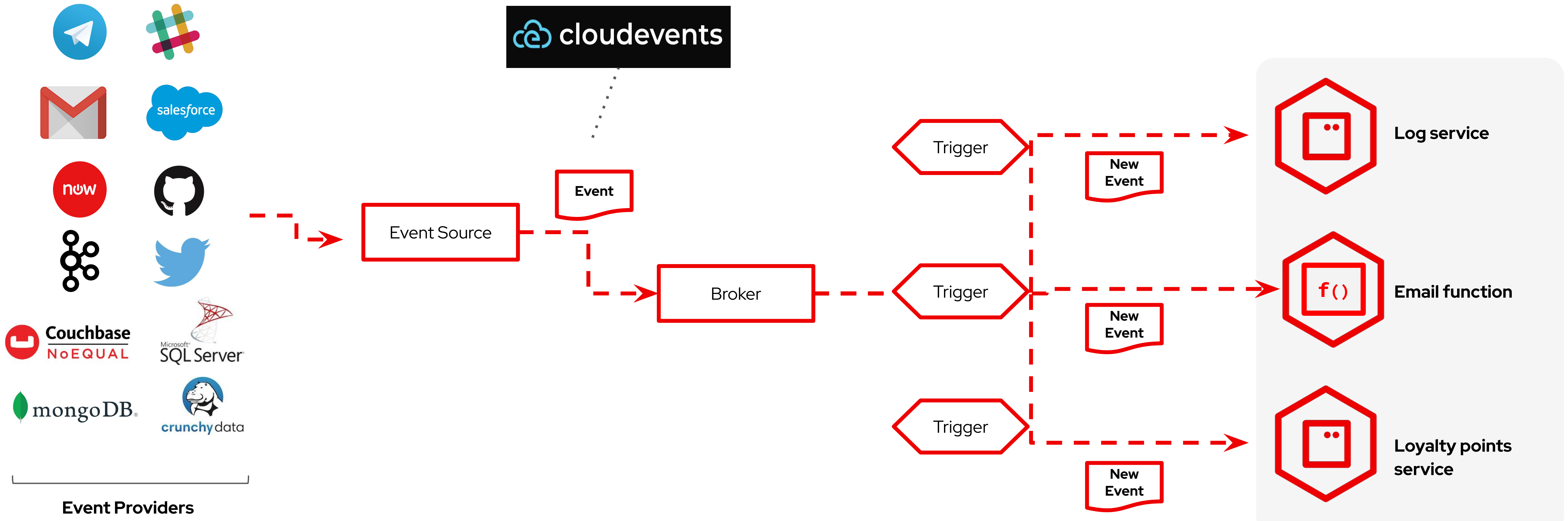
Knative Eventing - Source to Sink pattern



Knative Eventing - Source to Sink pattern



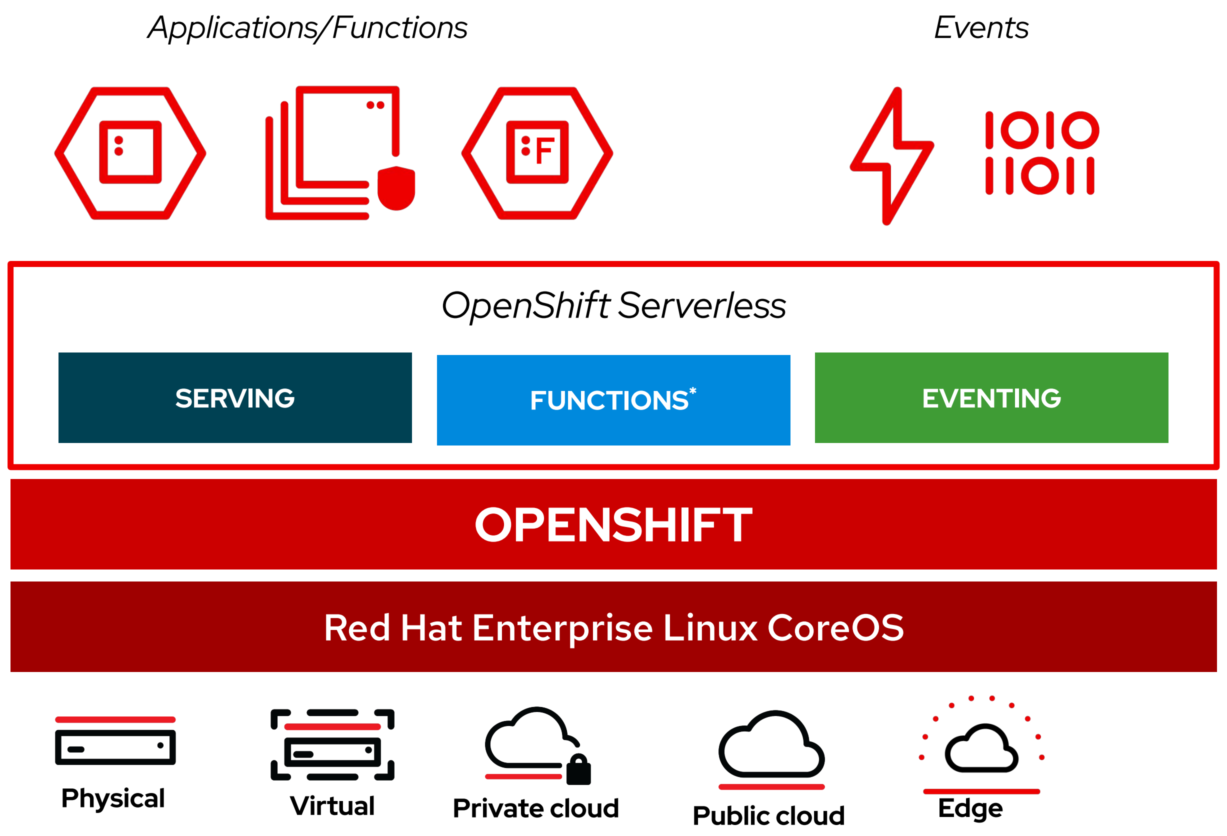
Knative Eventing - Broker and Trigger pattern



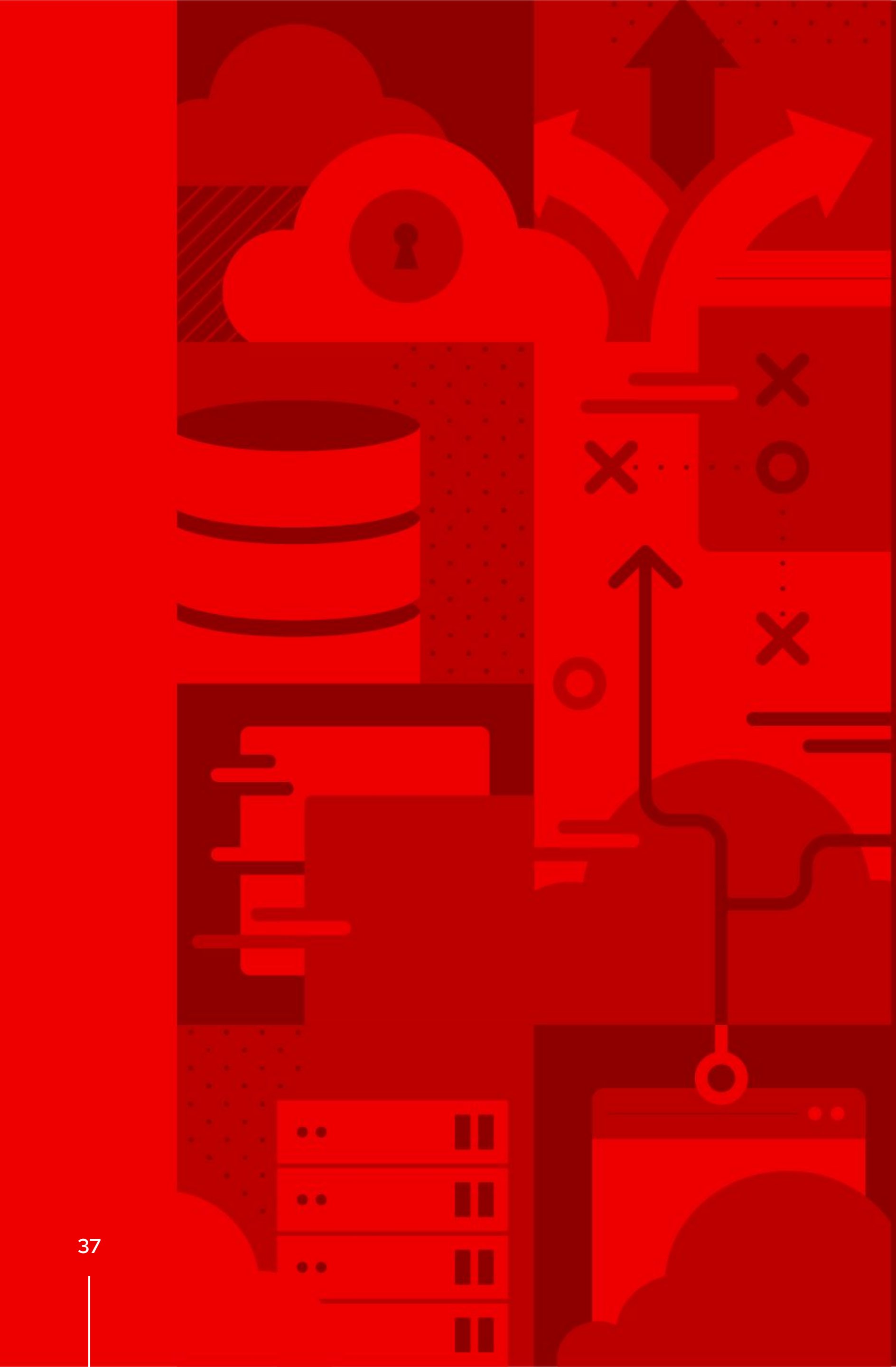
Openshift Serverless

Event-driven serverless containers and functions

- Deploy and run **serverless containers**
- Use any programming language or runtime
- Modernize existing applications to run serverless
- Powered by a rich ecosystem of event sources
- Manage serverless apps natively in Kubernetes
- Based on open source project **Knative**
- Run anywhere OpenShift runs



* Functions is currently in Technology Preview



Red Hat
OpenShift
Application Services

Streams & Serverless Demo

Video game revenue exceeds sports and movies industries

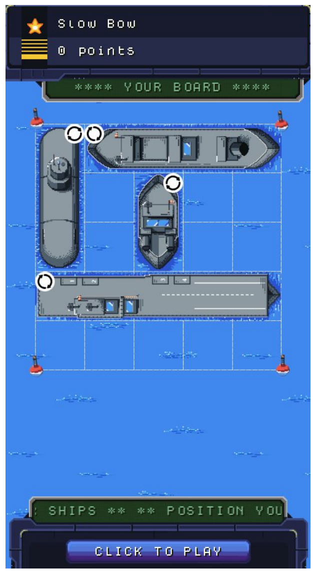
New York Post

Move over, NBA and Wonder Woman, there's a new top dog in town.

Source: Foxbusiness

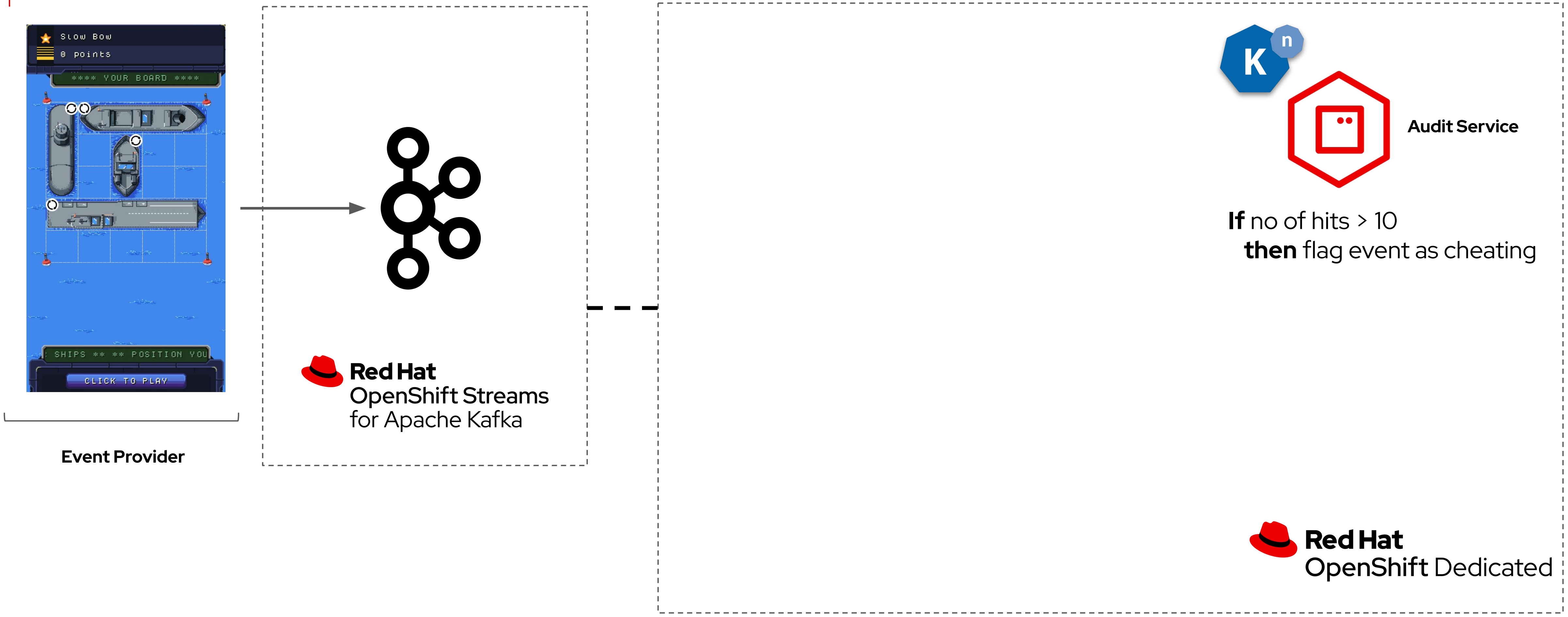
Cheating in Video Games – Business Impact

More than 2.5 billion players worldwide are playing video games – including cheaters. The Irdeto Global Gaming Survey revealed that 60% of gamers across the globe have had their multiplayer gaming experience negatively impacted by other players cheating. And 77% are likely to walk away from a multiplayer online game if they feel that other players are gaining an unfair advantage through cheating. Plus a Global Gaming Survey found that nearly half of gamers are less likely to buy in-game content if the encounter cheating.

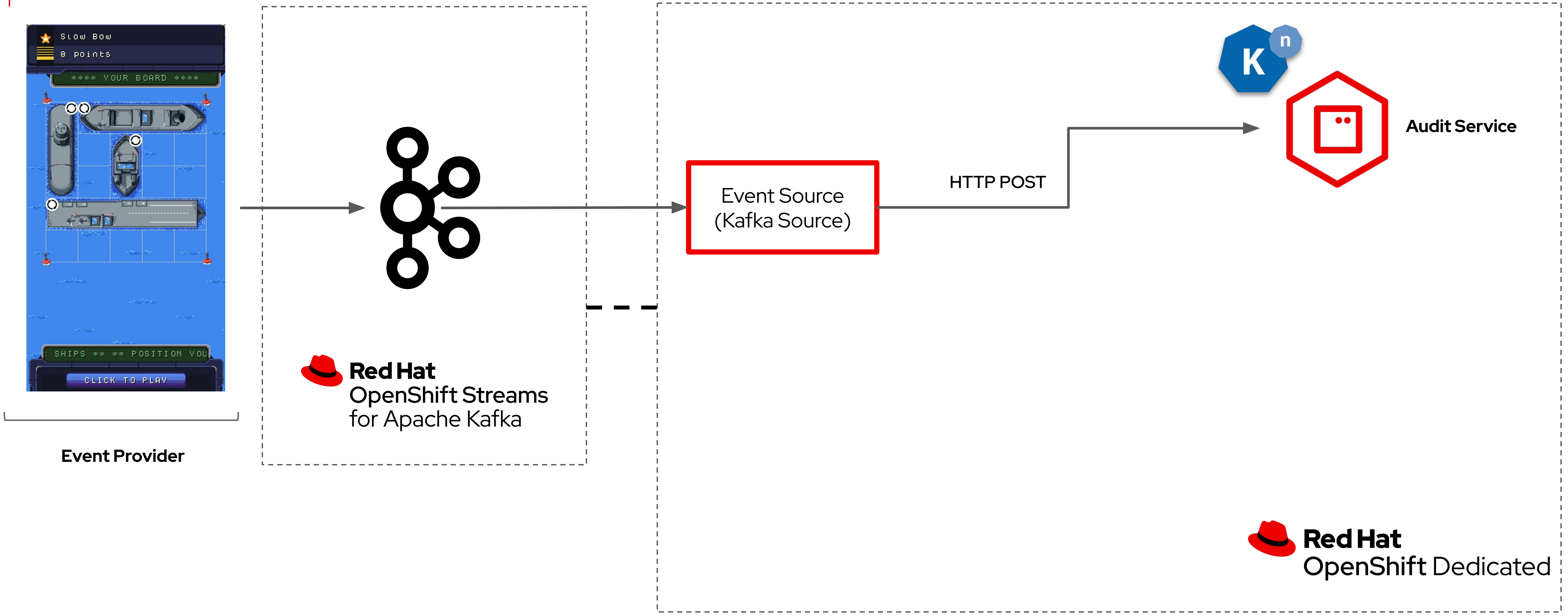


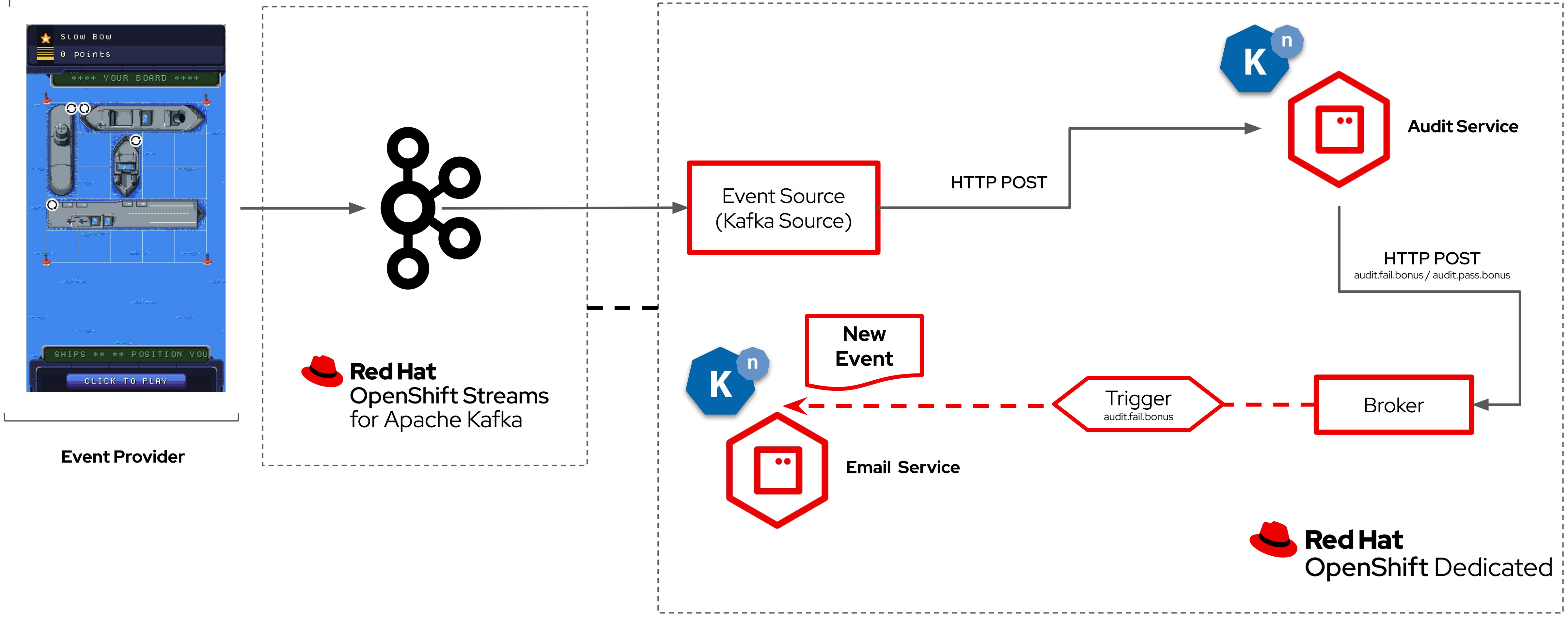
Event Provider

Red Hat
OpenShift Streams
for Apache Kafka



If no of hits > 10
then flag event as cheating


Slow Bow
0 points
**** YOUR BOARD ****
SHIPS ** ** POSITION YOU
CLICK TO PLAY
Event Provider
Red Hat
OpenShift Streams
for Apache Kafka
Event Source
(Kafka Source)
HTTP POST
Audit Service
Red Hat
OpenShift Dedicated


Slow Bow
0 points
**** YOUR BOARD ****
SHIPS ** ** POSITION YOU
CLICK TO PLAY
Event Provider
Red Hat
OpenShift Streams
for Apache Kafka
Event Source
(Kafka Source)
HTTP POST
Audit Service
HTTP POST
audit.fail.bonus / audit.pass.bonus
Broker
Trigger
audit.fail.bonus
New
Event
Email Service
Red Hat
OpenShift Dedicated

SCED-00816
demo one
EUROPE
SONY
COMPUTER ENTERTAINMENT
PlayStation
PAL
PBPX 95001
COMPACT disc
Compilation © 1997 Sony Computer Entertainment Europe.
[A Division of Sony Electronic Publishing Ltd.]. All Rights Reserved.
• EJEMPLAR DE DEMOSTRACIÓN - PROHIBIDA LA VENTA • KUN TIL DEMONSTRATION - IKKE TIL VIDERESALG •
• ENBART DEMO - FÅR EJ SÄLJAS VIDARE • APENAS PARA DEMONSTRAÇÃO - NÃO É PARA REVENDA •
• DEMOVERSION - NICHT ZUM VERKAUF • DÉMO UNIQUEMENT - REVENTE INTERDITE •
• NÄYTE - EI JÄLLEENMYYNTIIN • ALLEEN DEMO - NIET VOOR VERKOOP •
• ΜΟΝΟ ΓΙΑ ΕΠΙΔΕΙΞΗ ΑΠΑΓΟΡΕΥΕΤΑΙ Η ΜΕΤΑΠΩΛΗΣΗ •
• VERSIONE DIMOSTRATIVA - VIETATA LA VENDITA •
• DEMO ONLY - NOT FOR RESALE •

Source:
https://crimson-ceremony.net

Q
&
A

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