Event-Driven Architecture and Serverless with Red Hat ....Part 2

Red Hat

Outline

1. Intros
2. Participating in the demo!
3. Recap of why this matters
4. OpenShift Serverless
5. Sample Business Architecture
6. Demo Architecture
7. Demo!

Laine Vyvyan

Josh Smith

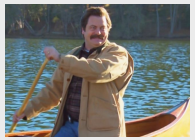
Participating in the Demo

Red Hat

Participating in the demo

1. Go to https://bit.ly/serverless-demo-slack
2. Accept the invite
3. Join channel ***messaging-demo***

Recap: Why This Matters



Red Hat

“Serverless,” Defined

Serverless computing is a cloud computing execution model in which the applications are written by the cloud consumer and the infrastructure is managed by the cloud provider.

“Serverless,” Defined

Serverless computing is…

- a cloud computing execution model
- where the applications are written by the cloud consumer
- and infrastructure is managed by the cloud provider

“Serverless,” Defined

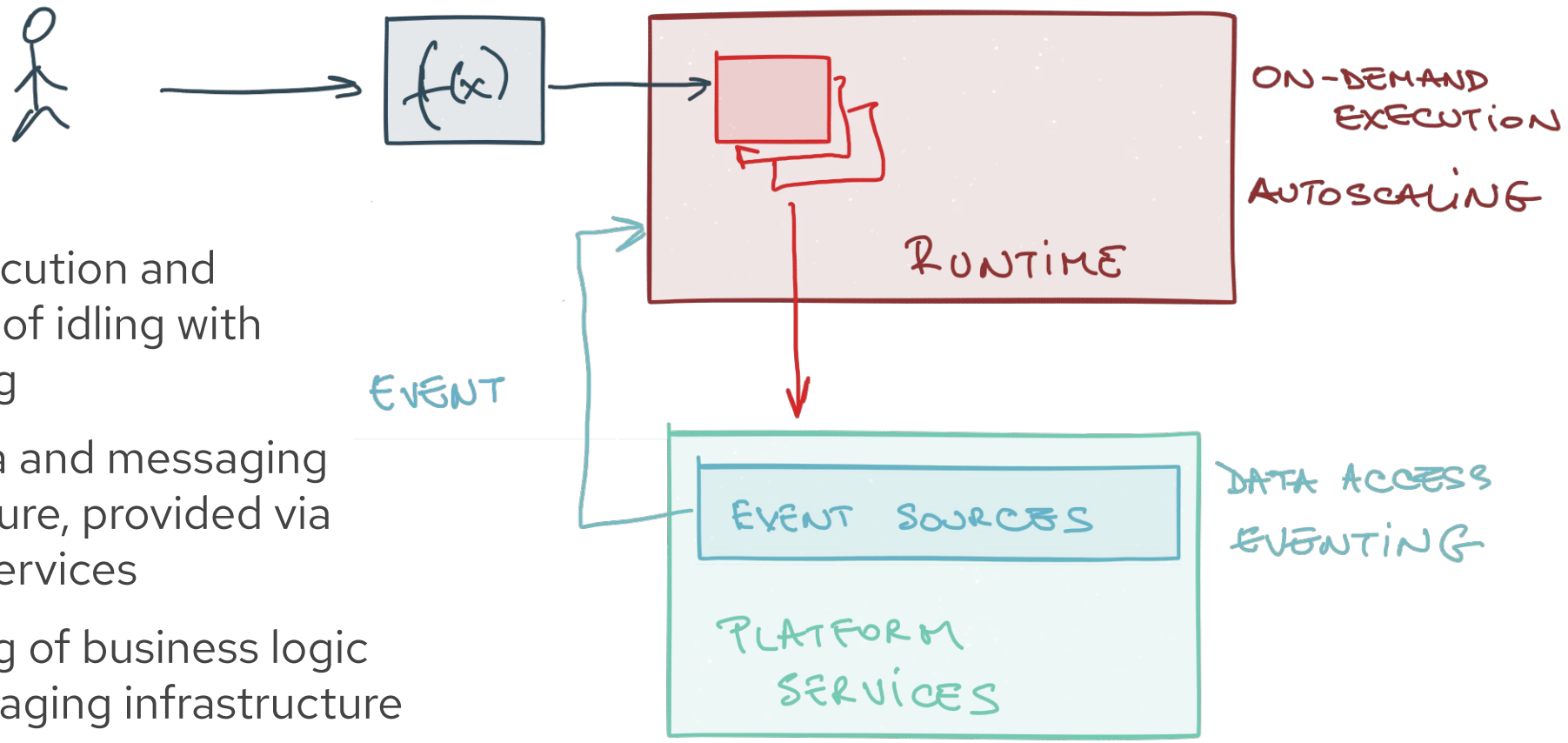
Serverless computing is…

- a cloud computing **execution model**
- where the applications are written by the cloud consumer
- and infrastructure is managed by the cloud provider

“Infrastructure”: from hardware all the way up the stack to the ***number of instances of applications running***

Red Hat

Event-Based Serverless Architecture: A Model



- Elastic execution and avoidance of idling with autoscaling
- Utility data and messaging infrastructure, provided via platform services
- Decoupling of business logic from messaging infrastructure

Architecture Problems	Architecture Solutions
Tight coupling	Event-Driven messages
Cascading failures	Event-Driven messages
Call chain latency	Event-Driven messages
Cloud latency	Event-Driven messages
Getting scaling right	Serverless

OpenShift Serverless

Red Hat

1.0

AWS Lambda, Functions...

Built around the FaaS components and other services such as API Gateways. It enabled a variety of use cases but it is far from ideal for general computing and with room for improvements.

- ➔ HTTP and other few Sources
- ➔ **Functions only**
- ➔ **Limited execution time (5 min)**
- ➔ No orchestration
- ➔ Limited local development experience

1.5

Serverless Containers

With the advent of Kubernetes, many frameworks and solutions started to auto-scale containers. Cloud providers created offerings using managed services completely abstracting Kubernetes APIs.

- ➔ Red Hat joins Knative
- ➔ Kubernetes based auto-scaling
- ➔ **Microservices and Functions**
- ➔ Easy to debug & test locally
- ➔ **Polyglot & Portable**

2.0

Integration & State

The maturity and benefits of Serverless are recognized industry wide and it adds the missing parts to make pattern suitable for general purpose workloads and used on the enterprise.

- ➔ Basic state handling
- ➔ **Enterprise Integration Patterns**
- ➔ Advanced Messaging Capabilities
- ➔ **Blended with your PaaS**
- ➔ Enterprise-ready event sources

Serverless is still evolving...

Serverless Market Trends

"Use Serverless To optimize The Benefits of The cloud" [2]

40%

of enterprises adopted Serverless technologies or practices with expected growth coming in the next 12 to 18 months.[1]

Vendor lock-in is the second biggest concern when adopting Serverless technologies.[1]

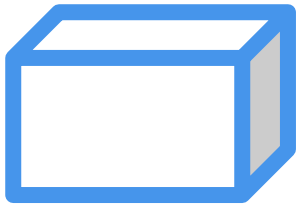
60%

of the serverless practitioners reported *"reduction of operational costs"* with the second biggest benefit being *"scale with demand automatically"*

Source:
1. https://www.oreilly.com/radar/oreilly-serverless-survey-2019-concerns-what-works-and-what-to-expect/
2. Forrester - Now Tech: Serverless, Q4 2019

Red Hat

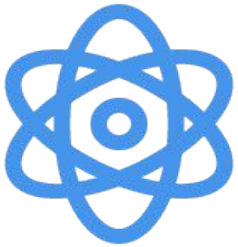
Application Architecture Choices



Monolith

Cloud Native

Serverless

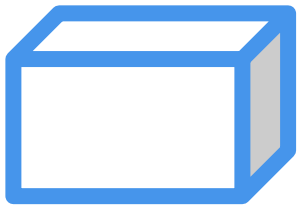


Request/Reply Architecture

Event-Driven Architecture

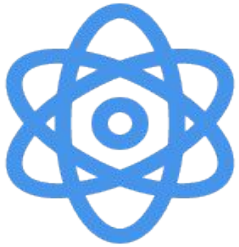
Red Hat

Common Deployment Tools



Monolith

Cloud Native



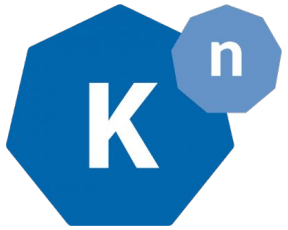
Request/Reply

Serverless

Event-Driven

kubernetes

Istio



Knative

Red Hat

OpenShift Container Platform

Multi-cluster Management

Discovery ⁝ Policy ⁝ Compliance ⁝ Configuration ⁝ Workloads

Manage Workloads	Build Cloud-Native Apps	Developer Productivity
Platform Services	**Application Services**	**Developer Services**
Service Mesh ⁝ Serverless Builds ⁝ CI/CD Pipelines Full Stack Logging Chargeback	Databases ⁝ Languages Runtimes ⁝ Integration Business Automation 150+ ISV Services	Helm ⁝ Developer CLI ⁝ VS Code extensions ⁝ IDE Plugins Code Ready Workspaces CodeReady Containers

Operate Kubernetes

Cluster Services

Automated Ops ⁝ Over-The-Air Updates ⁝ Monitoring ⁝ Telemetry ⁝ Logging ⁝ Registry ⁝ Networking ⁝ Router

Kubernetes

Red Hat Enterprise Linux & RHEL CoreOS


Physical


Virtual


Private cloud


Public cloud

Red Hat

OpenShift Container Platform

Multi-cluster Management

Discovery ⁝ Policy ⁝ Compliance ⁝ Configuration ⁝ Workloads

Manage Workloads	Build Cloud-Native Apps	Developer Productivity
Platform Services	**Application Services**	**Developer Services**
Service Mesh ⁝ **Serverless** Builds ⁝ CI/CD Pipelines Full Stack Logging Chargeback	Databases ⁝ Languages Runtimes ⁝ Integration Business Automation 150+ ISV Services	Helm ⁝ Developer CLI ⁝ VS Code extensions ⁝ IDE Plugins Code Ready Workspaces CodeReady Containers

Operate Kubernetes

Cluster Services

Automated Ops ⁝ Over-The-Air Updates ⁝ Monitoring ⁝ Logging ⁝ Registry ⁝ Networking ⁝ Router

Kubernetes

Red Hat Enterprise Linux & RHEL CoreOS

Physical | Virtual | Private cloud | Public cloud

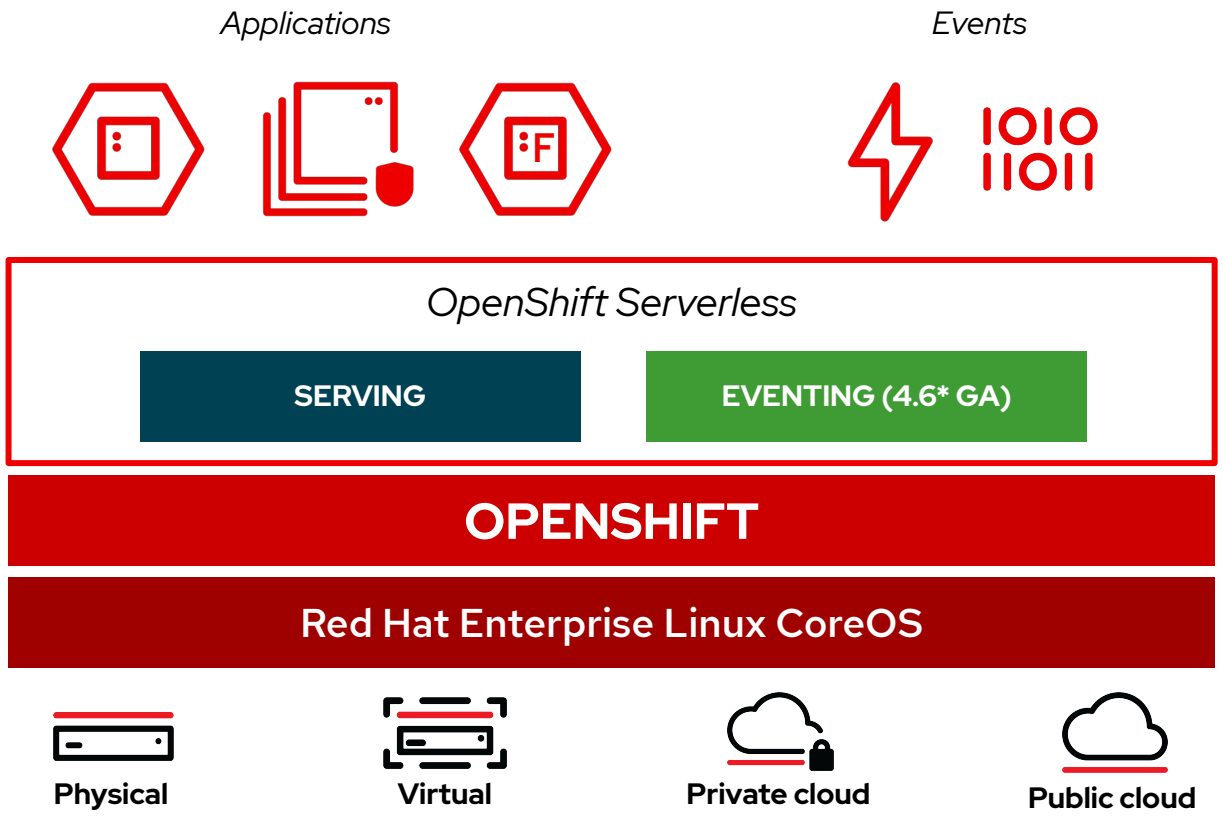
Red Hat

OpenShift Serverless

Event-driven, serverless containers and functions

GA in OpenShift 4.5

- 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



* 4.6 is the target, this could change!

Red Hat

Serverless Themes

Monitoring and Automation

Powerful monitoring capabilities with configuration and automation for GitOps and modern CI/CD practices.

Integrations and Ecosystem

Eventing capabilities enabling a rich ecosystem of Event Sources from Red Hat and Partner products.

Developer Experience

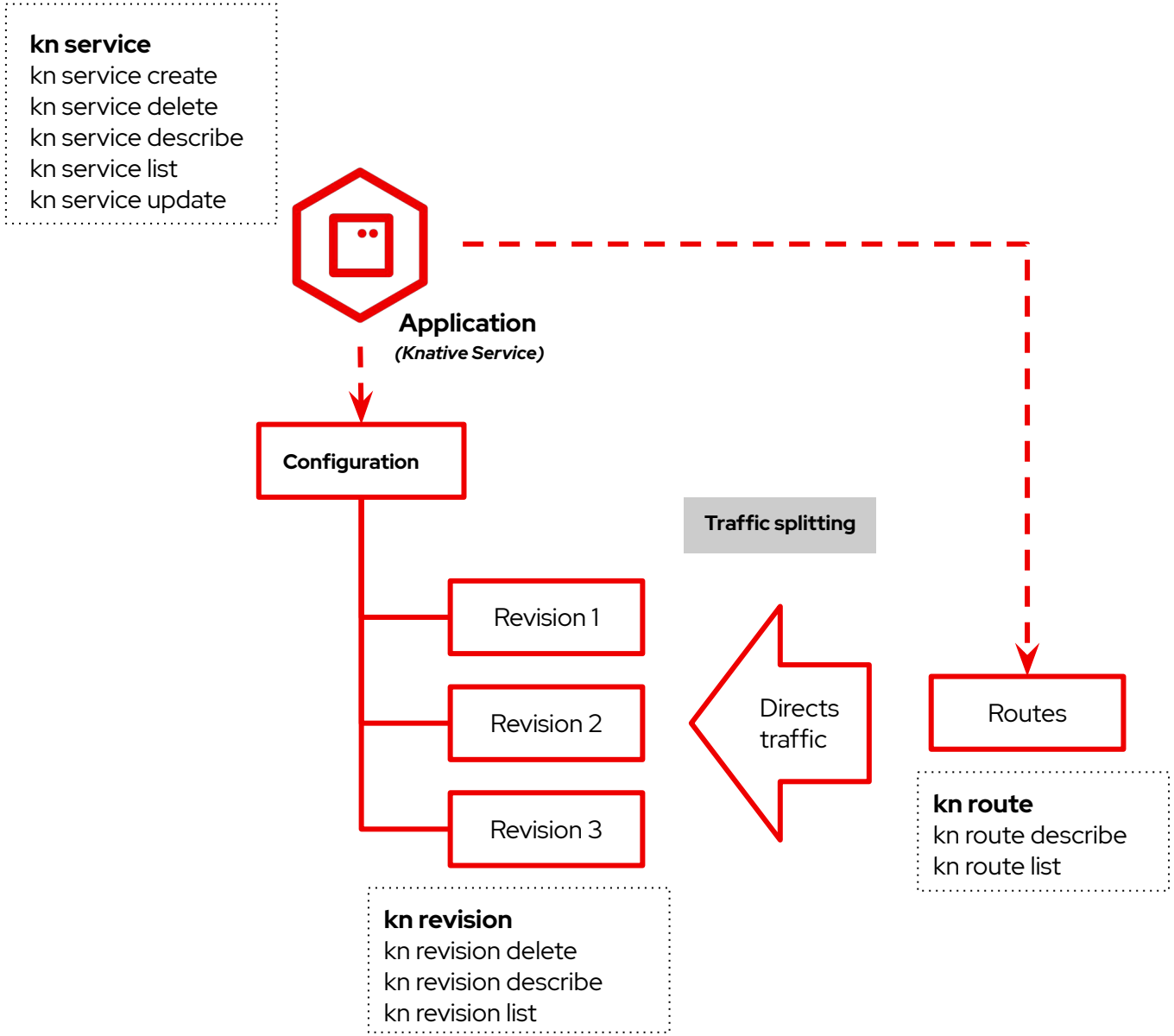
Intuitive developer experience through the Developer Console and CLI/IDE with Functions support.

Generally Available

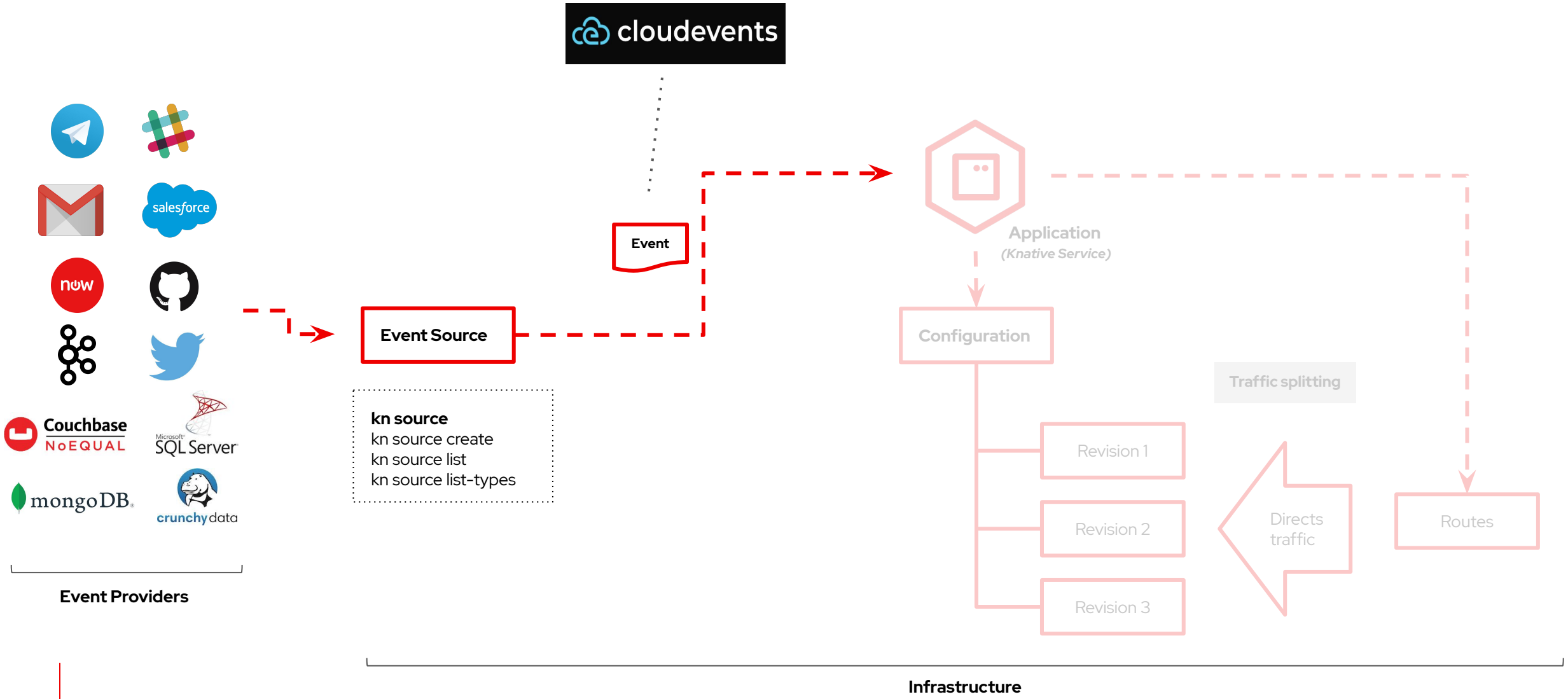
Serving

- From container to URL within seconds
- Easier developer experience for Kubernetes
- Built-in versioning, traffic split and more
- Simplified Installation experience with Kourier *new*
- Automatic TLS/SSL for Applications *new*

```
$ kn service create --image=<container>
```

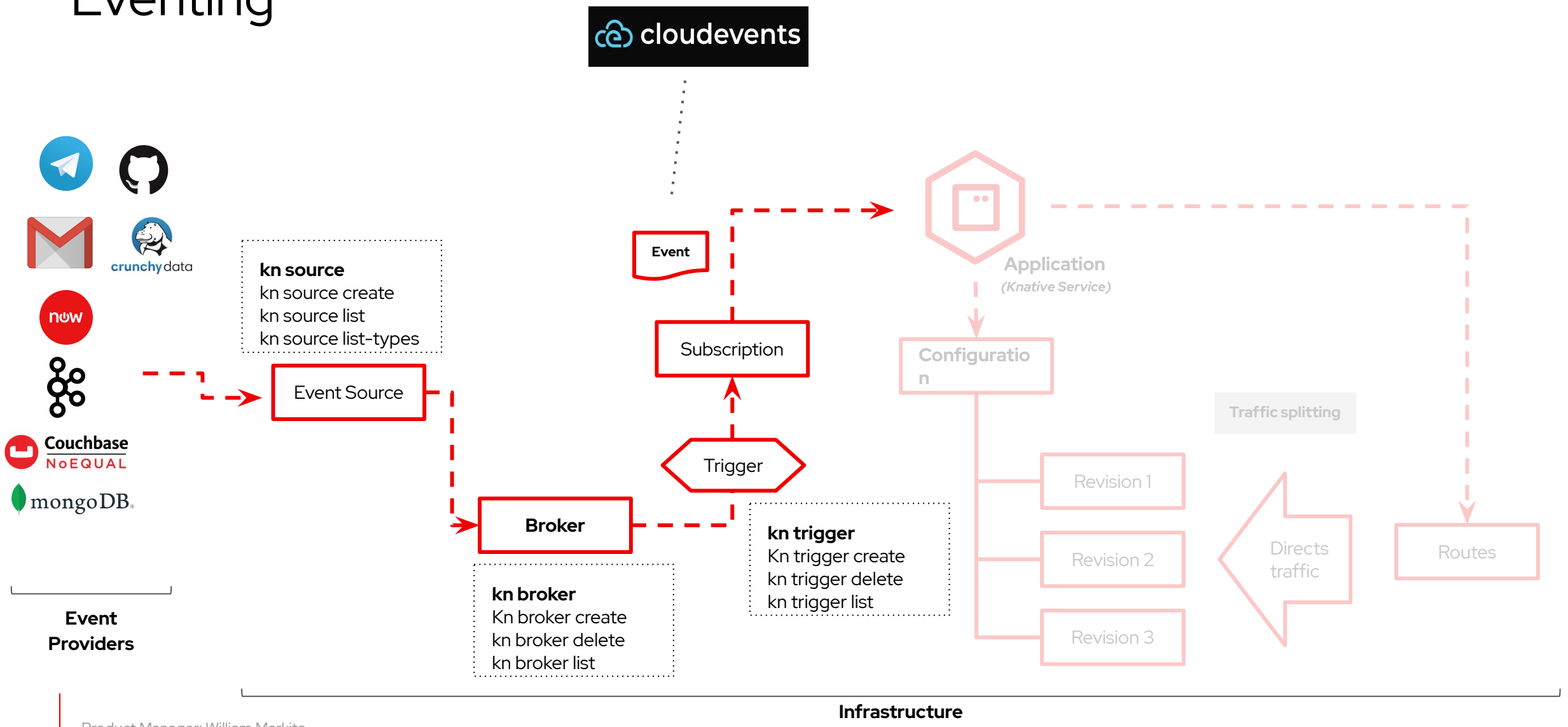


Eventing

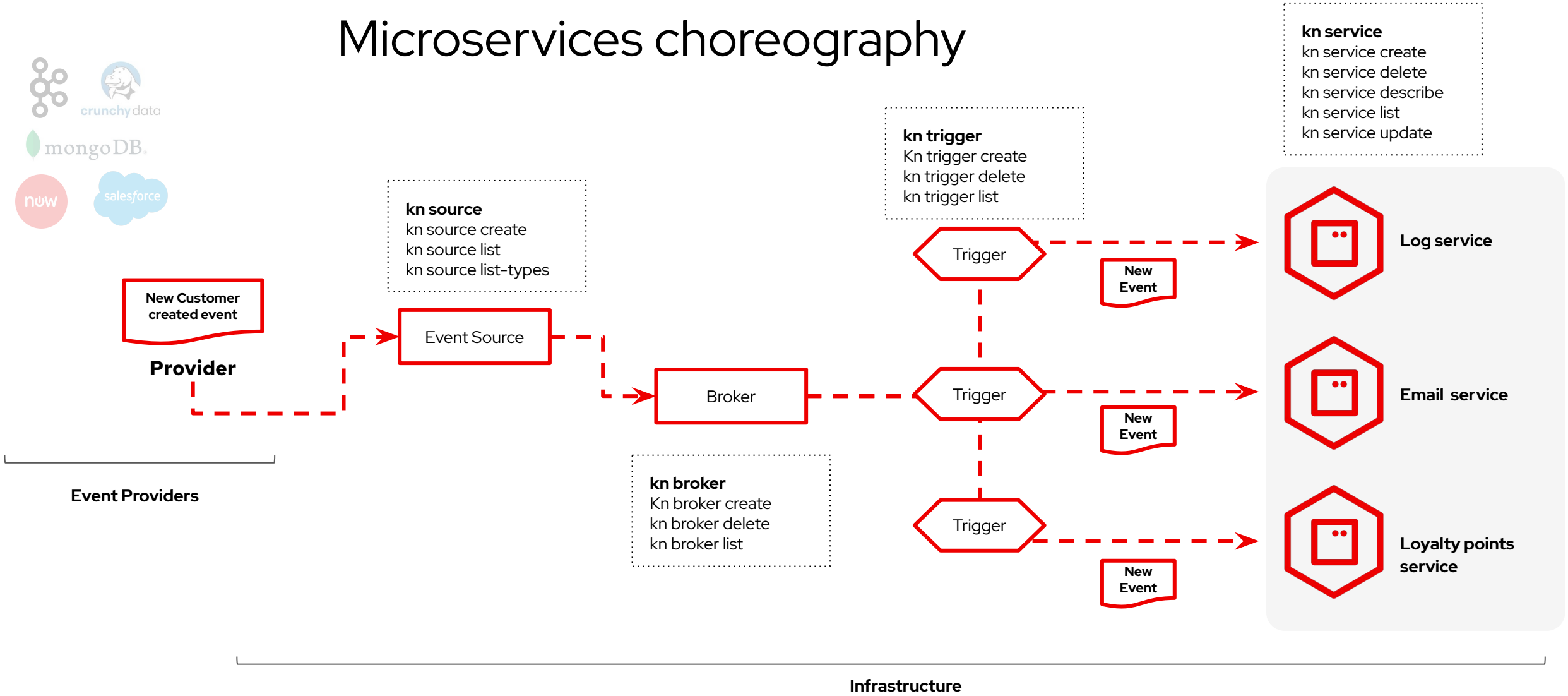

cloudevents
Event
Event Source
kn source
kn source create
kn source list
kn source list-types
Application
(Knative Service)
Configuration
Traffic splitting
Revision 1
Revision 2
Revision 3
Directs traffic
Routes
Couchbase
NoEQUAL
SQL Server
mongoDB
crunchydata
Event Providers
Infrastructure

Technology Preview

Eventing



Microservices choreography



Event Sources in the Developer Console

Red Hat OpenShift Container Platform

kube:admin

Developer

+Add

Topology

Monitoring

Search

Builds

Pipelines

Helm

Project

Config Maps

Secrets

Pods

You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Project: stest2 Application: all applications

Add

No workloads found

To add content to your project, create an application, component or service using one of these options.

From Git
Import code from your git repository to be built and deployed

Container Image
Deploy an existing image from an image registry or image stream tag

From Dockerfile
Import your Dockerfile from your git repo to be built and deployed

YAML
Create resources from their YAML or JSON definitions

From Catalog
Browse the catalog to discover, deploy and connect to services

Database
Browse the catalog to discover database services to add to your application

Operator Backed
Browse the catalog to discover and deploy operator managed services

Helm Chart
Browse the catalog to discover and install Helm Charts

https://console-openshift-console.apps.viraj01006.devcluster.openshift.com/im...

Sink Binding

Container Source

Kafka

Event Sources

Create an event source to register interest in a class of events from a particular system

Type

ApiServerSource | ContainerSource | CronJobSource | KafkaSource | PingSource | SinkBinding | CamelSource

KafkaSource

BootstrapServers *

The address of the Kafka broker

Add Bootstrapservers

Topics *

Virtual groups across Kafka brokers

Add Topics

ConsumerGroup *

A group that tracks maximum offset consumed

Create | Cancel

Type

ApiServerSource | ContainerSource | CronJobSource

SinkBinding

Subject

apiVersion *

Kind *

Match Labels

NAME	VALUE
name	value

Add Values

ntainerSource | CronJobSource | PingSource | SinkBinding

The image to run inside of the container

Name

The name of the image

Arguments

argument

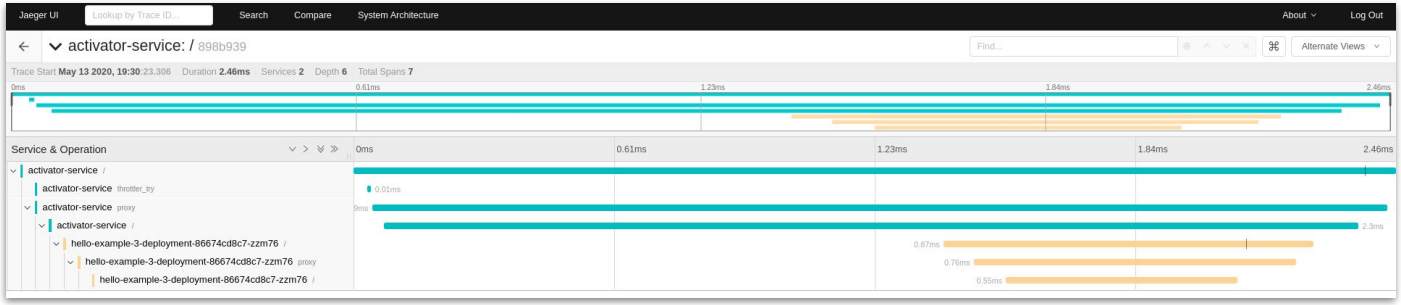
Arguments passed to the container

Add args

Red Hat

Developer Experience

Jaeger Support [2]



GPU Support [1] for Serverless Applications

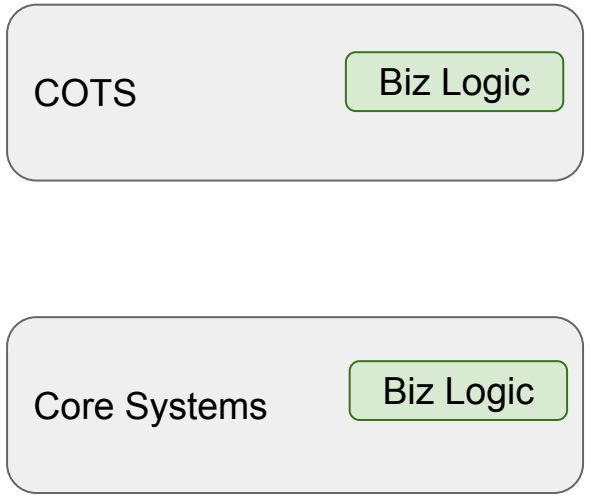
```
kn service create hello --image \ docker.io/knativesamples/hellocuda-go
--limit nvidia.com/gpu=1
```

Product Manager: William Markito

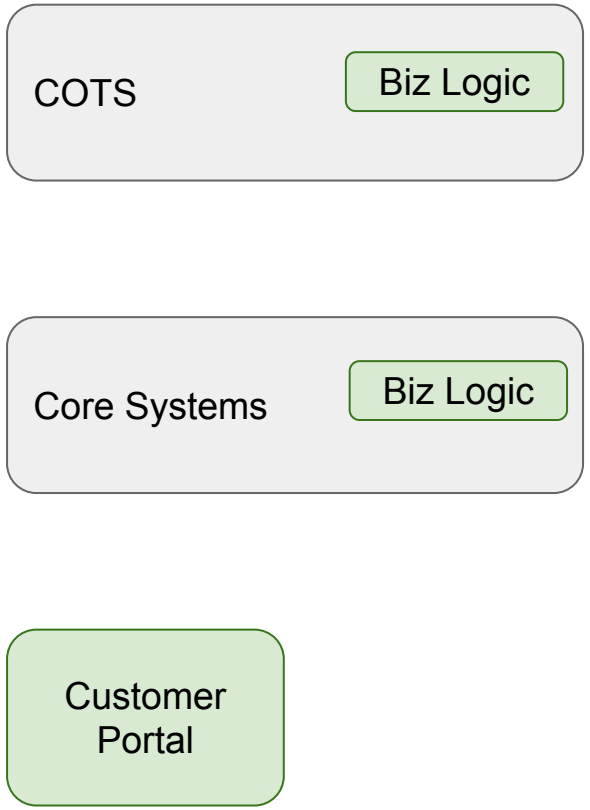
[1] https://docs.nvidia.com/datacenter/kubernetes/openshift-on-gpu-install-guide/index.html
[2] https://docs.openshift.com/container-platform/4.4/serverless/serverless-tracing.html

Sample Business Architecture

Red Hat

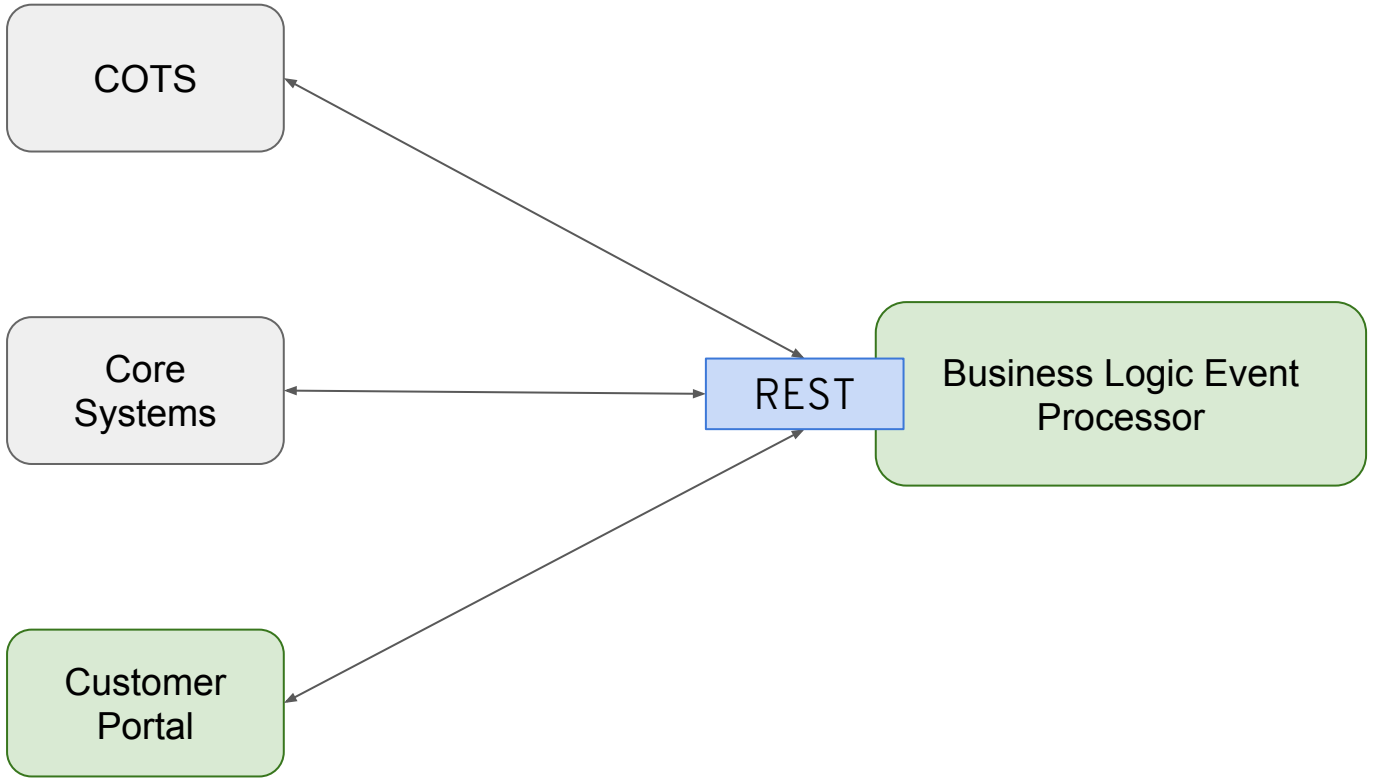


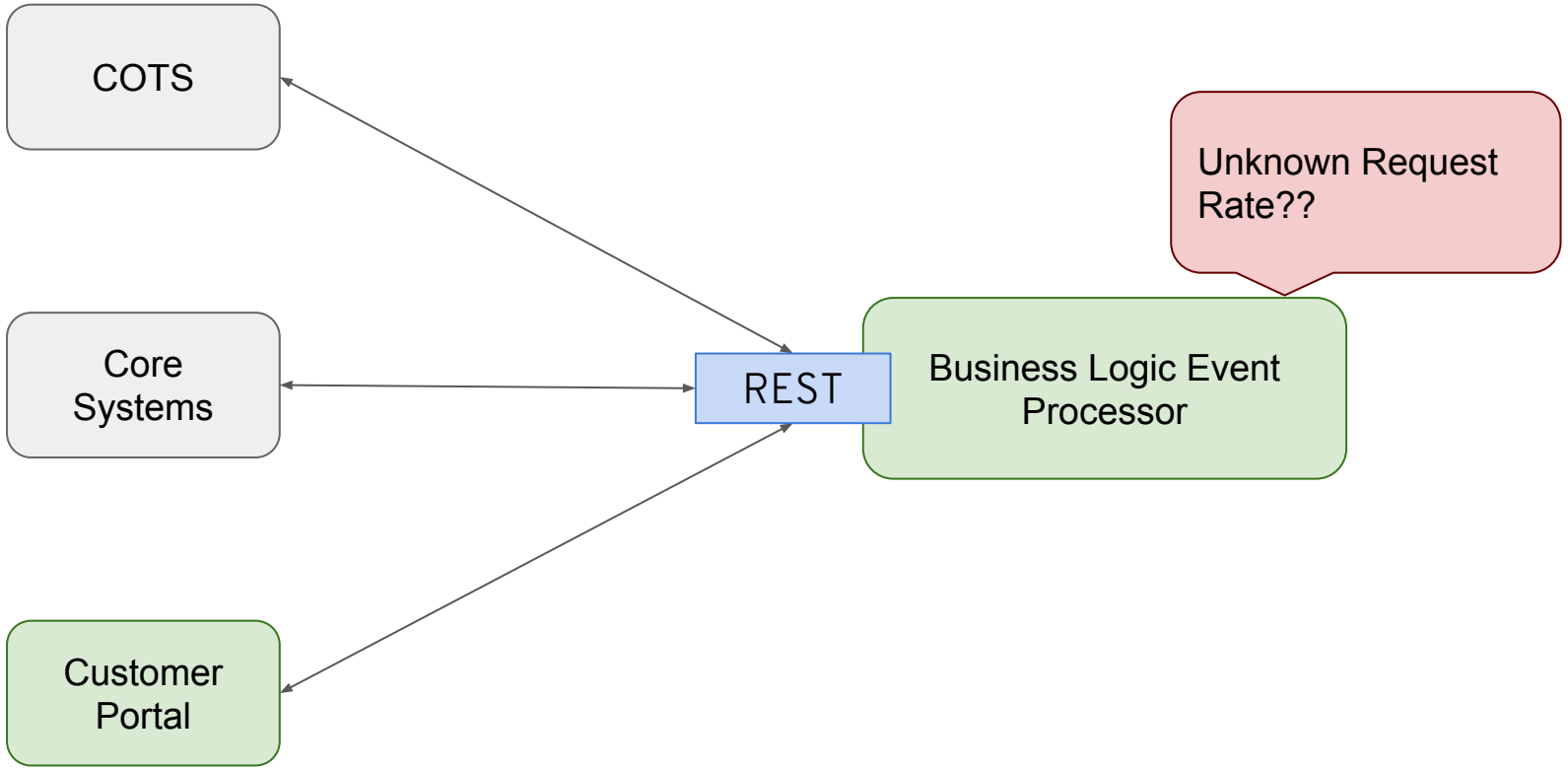
Red Hat

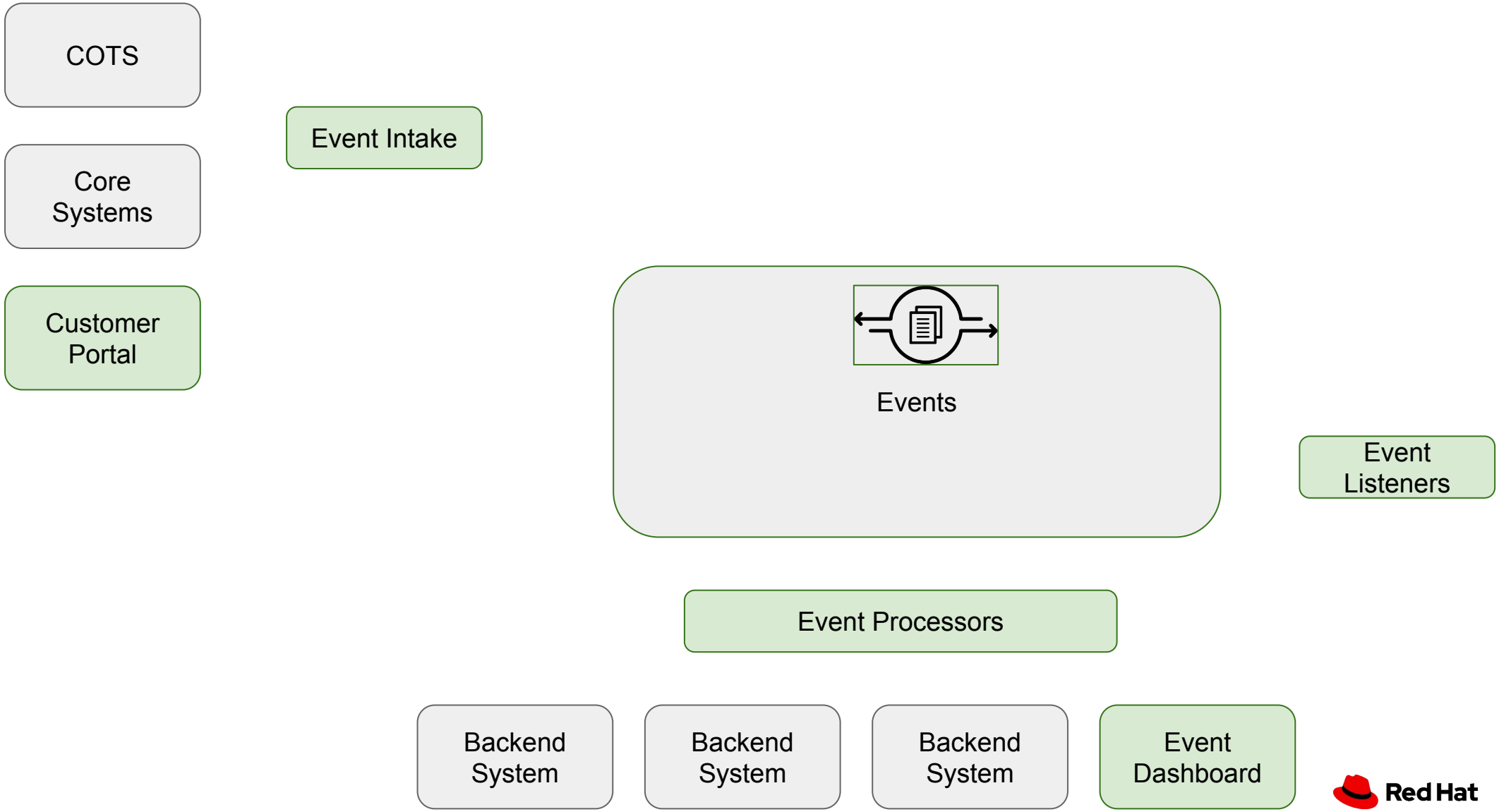

COTS
Biz Logic
Core Systems
Biz Logic
Customer Portal


COTS
Core Systems
Business Logic Event Processor
Customer Portal

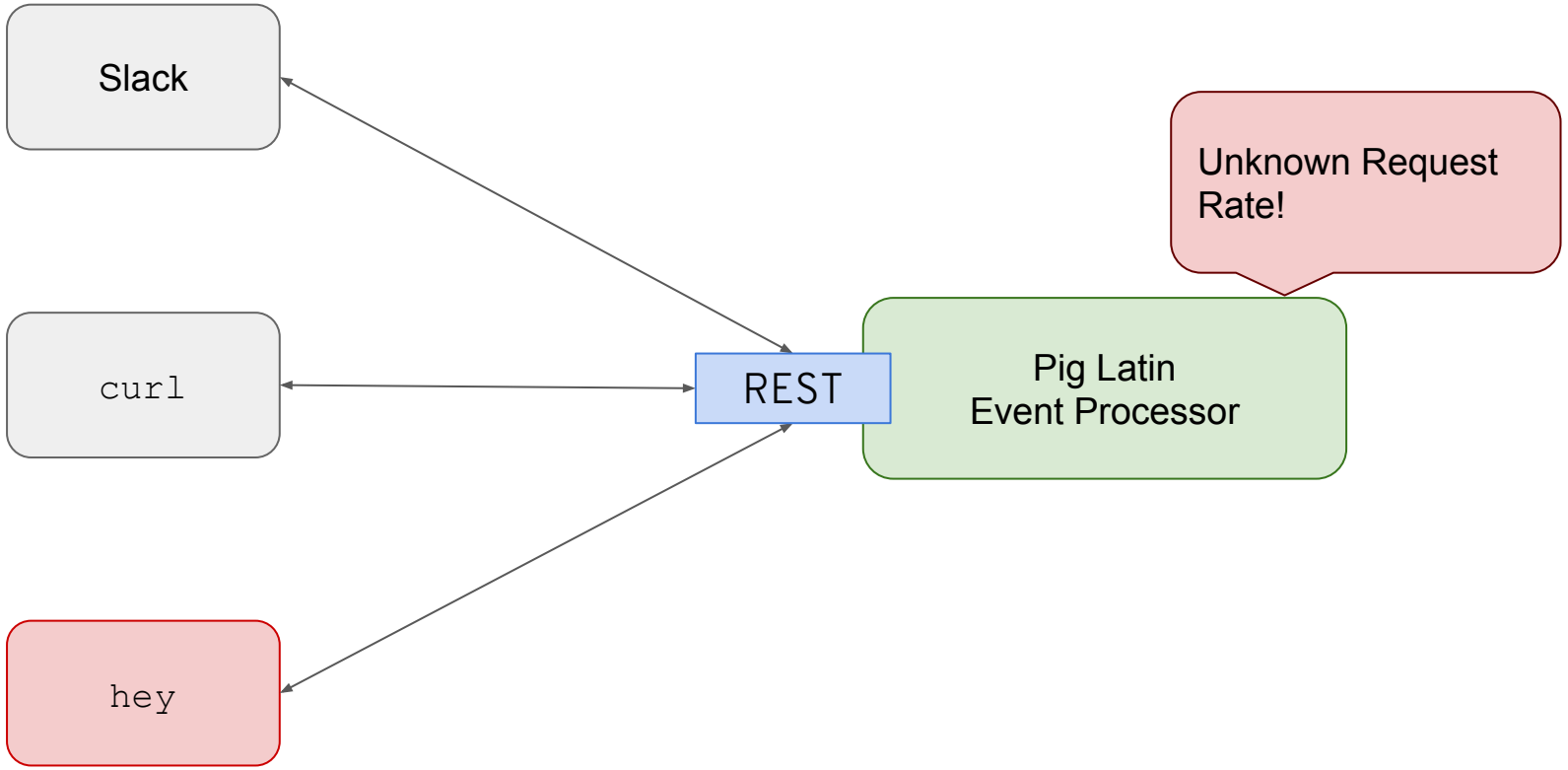
Red Hat


COTS
Core Systems
Customer Portal
REST
Business Logic Event Processor


COTS
Core Systems
Customer Portal
REST
Business Logic Event Processor
Unknown Request Rate??


COTS
Core Systems
Customer Portal
Event Intake
Events
Event Listeners
Event Processors
Backend System
Backend System
Backend System
Event Dashboard

Demo Architecture


Slack
curl
hey
REST
Pig Latin
Event Processor
Unknown Request
Rate!

Demo!

Red Hat

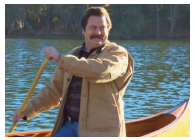
Participating in the demo

1. Go to https://bit.ly/serverless-demo-slack
2. Accept the invite
3. Join channel messaging-demo
4. **/piglatin `this is a cool sentence`**

Red Hat

emo-Day akeaways-Tay

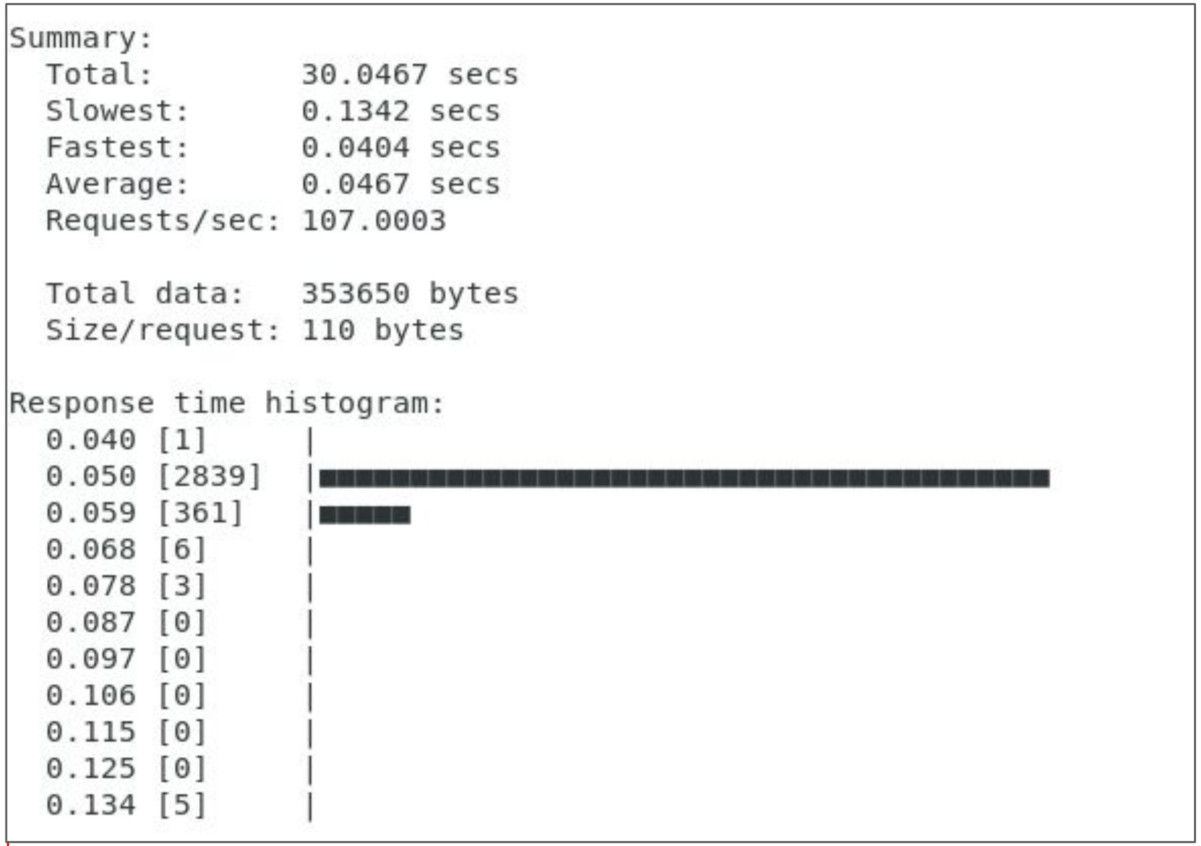
1. OpenShift Serverless (the serving part, anyway) is basically a different way to *deploy* an application. It uses the same built image as any other deployment method, and OpenShift magic does the rest
2. Traffic can cause an app to scale up in configurable ways
   a. This is configurable by CPU, by number of requests, or both
3. App start time ***matters***
4. Having a single instance running all the time may be a good idea and is possible



Red Hat

Demo Takeaways: Performance

Serverless:

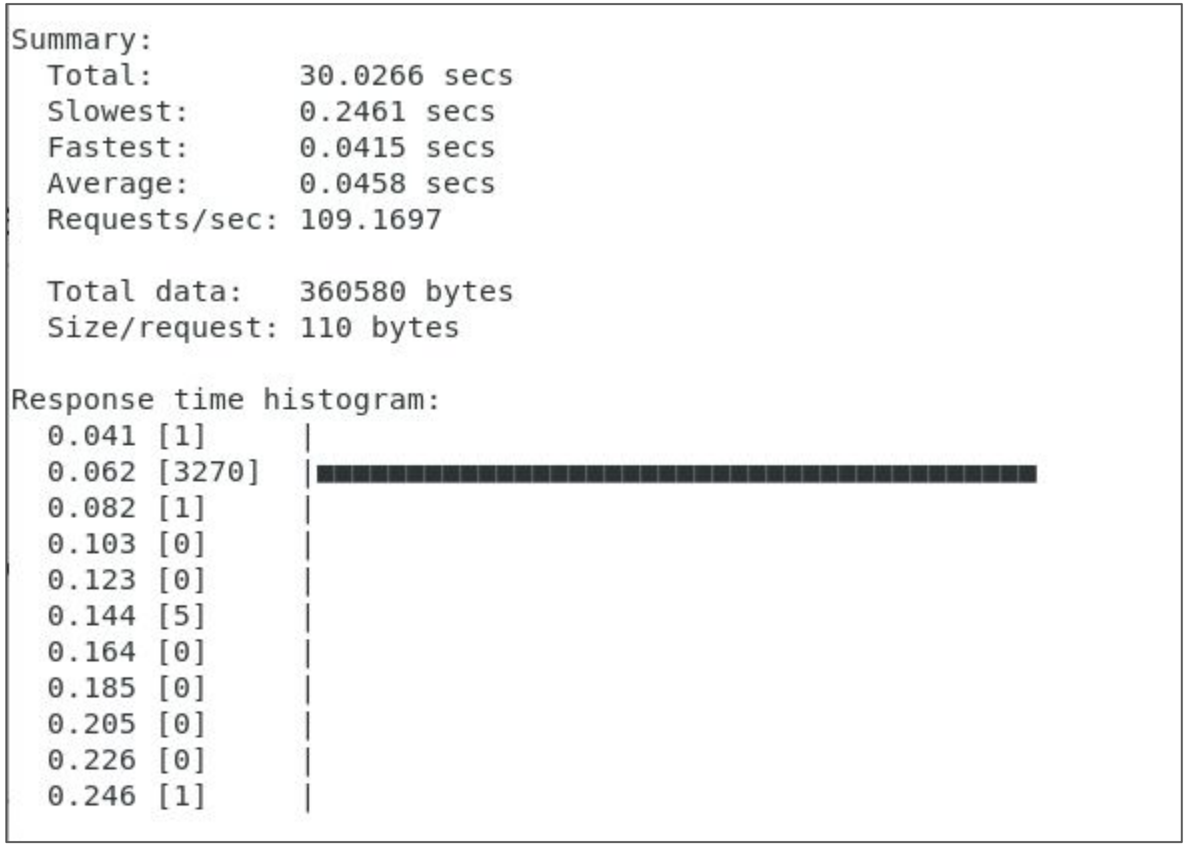


```
Summary:
  Total:        30.0467 secs
  Slowest:      0.1342 secs
  Fastest:      0.0404 secs
  Average:      0.0467 secs
  Requests/sec: 107.0003

  Total data:   353650 bytes
  Size/request: 110 bytes

Response time histogram:
  0.040 [1]     |
  0.050 [2839]  |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.059 [361]   |■■■■■
  0.068 [6]     |
  0.078 [3]     |
  0.087 [0]     |
  0.097 [0]     |
  0.106 [0]     |
  0.115 [0]     |
  0.125 [0]     |
  0.134 [5]     |
```

Static Number of Instances:



```
Summary:
  Total:        30.0266 secs
  Slowest:      0.2461 secs
  Fastest:      0.0415 secs
  Average:      0.0458 secs
  Requests/sec: 109.1697

  Total data:   360580 bytes
  Size/request: 110 bytes

Response time histogram:
  0.041 [1]     |
  0.062 [3270]  |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.082 [1]     |
  0.103 [0]     |
  0.123 [0]     |
  0.144 [5]     |
  0.164 [0]     |
  0.185 [0]     |
  0.205 [0]     |
  0.226 [0]     |
  0.246 [1]     |
```

Next Steps:
Your Technology Radar for Event-Driven and Serverless

- Service Mesh (Istio):
  - Provide microservice interconnectivity and visibility
- Serverless platforms (Knative)
  - Container build and on-demand scheduling
- Container-native frameworks (Quarkus)
  - Optimize Java workloads for serverless architecture

Next Steps:
Your Technology Radar for Event-Driven and Serverless

- Strimzi
  - Kafka operator for Kubernetes/OpenShift
- EnMasse
  - Messaging-as-a-Service for Kubernetes/OpenShift
- FaaS frameworks (e.g. Camel-K)
  - Schedule integration code directly on platform or via Knative

Next Steps:
Red Hat's Technology Radar for Event-Driven and Serverless

Red Hat AMQ

MESSAGING BACKBONE

Red Hat Data Grid

DISTRIBUTE, REACT ON DATA

Red Hat Application Runtimes

REACTIVE / FAAS FRAMEWORKS

Red Hat Middleware Portfolio

RULES EVALUATION
COMPLEX EVENTS
AUTOMATION

Red Hat Integration

CAMEL K
REACTIVE INTEGRATION
SERVERLESS

Red Hat CodeReady

REACTIVE DEVELOPER
TOOLING

Red Hat OpenShift

APP ENVIRONMENT
INFRASTRUCTURE
SERVERLESS / KNATIVE
OPERATOR HUB

Next Steps: Resources

Knative Tutorial on Red Hat Developer

Knative Cookbook on Red Hat Developer

OpenShift Serverless Tech Topic

Red Hat Services Overview of Serverless Blog

Knative Autoscaling Sample App

Knative Documentation re: Configuring Autoscaling

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