Red Hat Summit

Proč zvolit Red Hat pro cloud-native vývoj?

OpenShift Serverless & Knative project

Václav Tunka
Engineering Manager,
OpenShift Serverless
@vtunka

Red Hat

Open Hybrid Cloud

MULTICLOUD

noun • \ *muhl-tee* \ *klaud* \

Using **multiple clouds** from multiple private or public providers, for multiple workloads or tasks, **without interconnectivity** between clouds.

aws | openstack. | Microsoft Azure

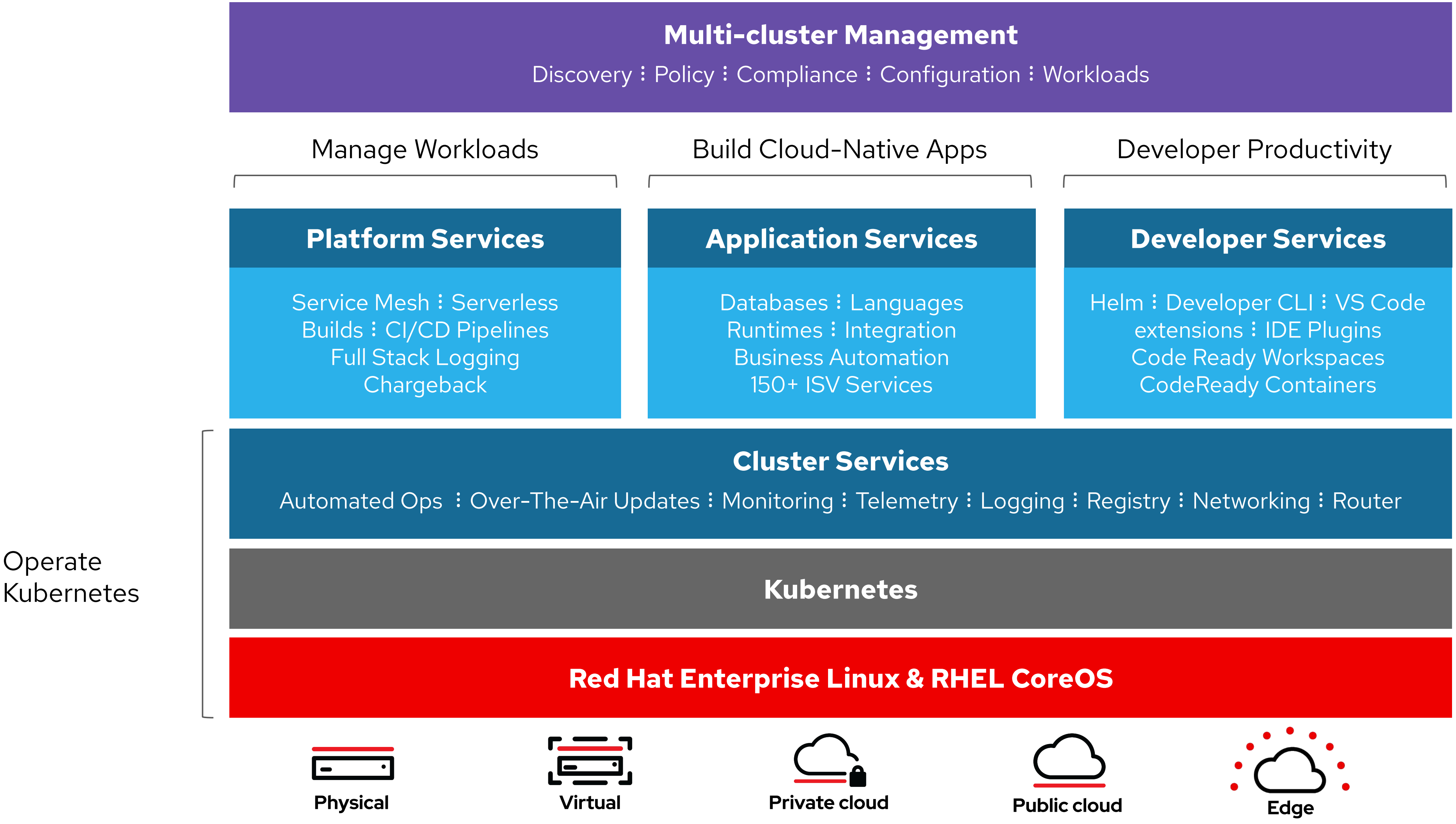
HYBRID CLOUD

noun • \ *hī-bred* \ *klaud* \

A combination of **public** and **private** clouds, possibly with some degree of workload portability, integration, orchestration, and unified management across said clouds.

aws ↔ openstack.

OpenShift Container Platform



Multi-cluster Management

Discovery ⁝ Policy ⁝ Compliance ⁝ Configuration ⁝ Workloads

Manage Workloads

Build Cloud-Native Apps

Developer Productivity

Platform Services

Service Mesh ⁝ Serverless
Builds ⁝ CI/CD Pipelines
Full Stack Logging
Chargeback

Application Services

Databases ⁝ Languages
Runtimes ⁝ Integration
Business Automation
150+ ISV Services

Developer Services

Helm ⁝ Developer CLI ⁝ VS Code
extensions ⁝ IDE Plugins
Code Ready Workspaces
CodeReady Containers

Cluster Services

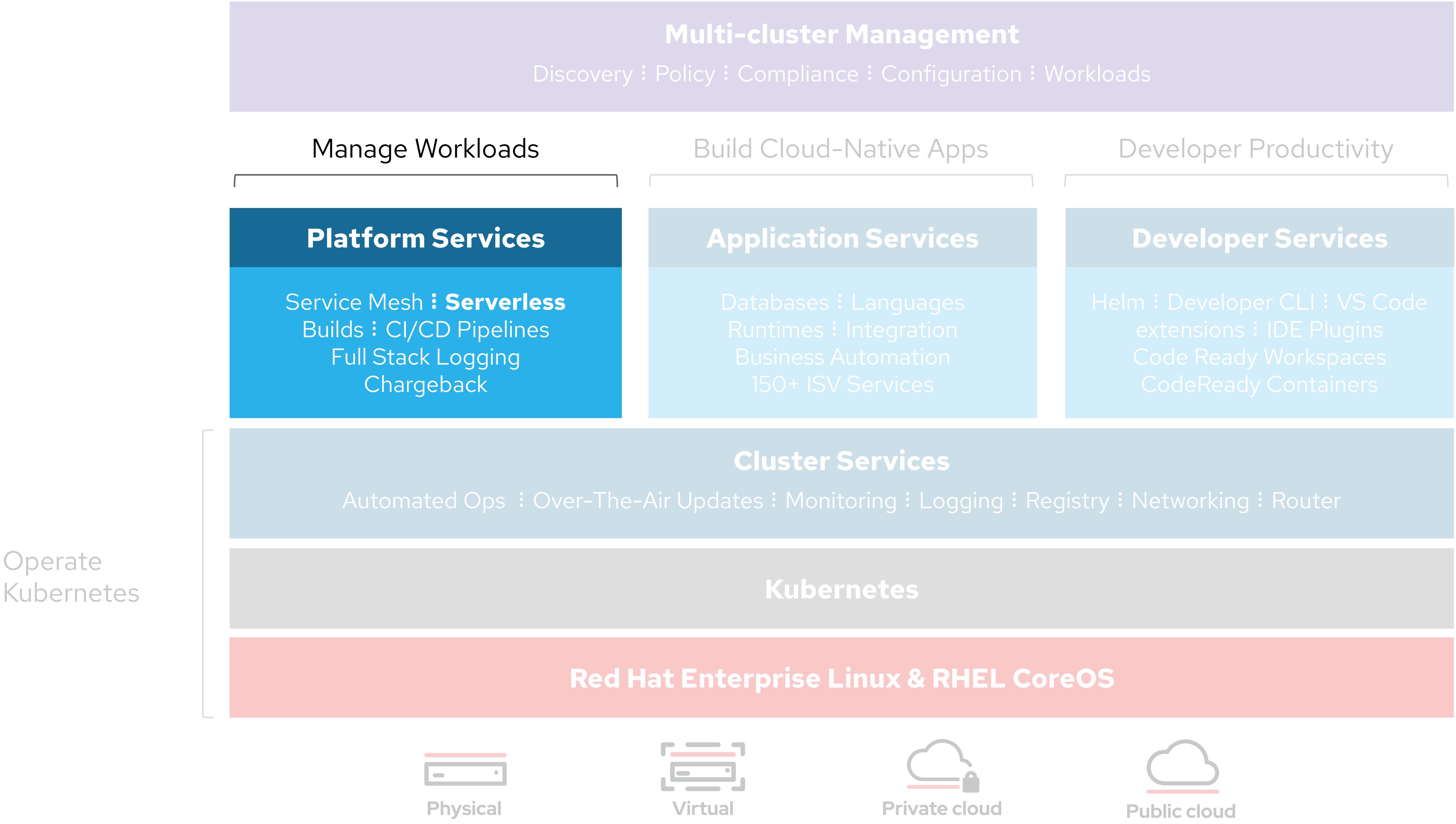
Automated Ops ⁝ Over-The-Air Updates ⁝ Monitoring ⁝ Telemetry ⁝ Logging ⁝ Registry ⁝ Networking ⁝ Router

Kubernetes

Red Hat Enterprise Linux & RHEL CoreOS

Operate
Kubernetes

OpenShift Container Platform



Multi-cluster Management

Discovery ⁝ Policy ⁝ Compliance ⁝ Configuration ⁝ Workloads

Manage Workloads

Build Cloud-Native Apps

Developer Productivity

Platform Services

Service Mesh ⁝ **Serverless**
Builds ⁝ CI/CD Pipelines
Full Stack Logging
Chargeback

Application Services

Databases ⁝ Languages
Runtimes ⁝ Integration
Business Automation
150+ ISV Services

Developer Services

Helm ⁝ Developer CLI ⁝ VS Code extensions ⁝ IDE Plugins
Code Ready Workspaces
CodeReady Containers

Cluster Services

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

Kubernetes

Red Hat Enterprise Linux & RHEL CoreOS

Operate Kubernetes

Physical

Virtual

Private cloud

Public cloud

First Principles

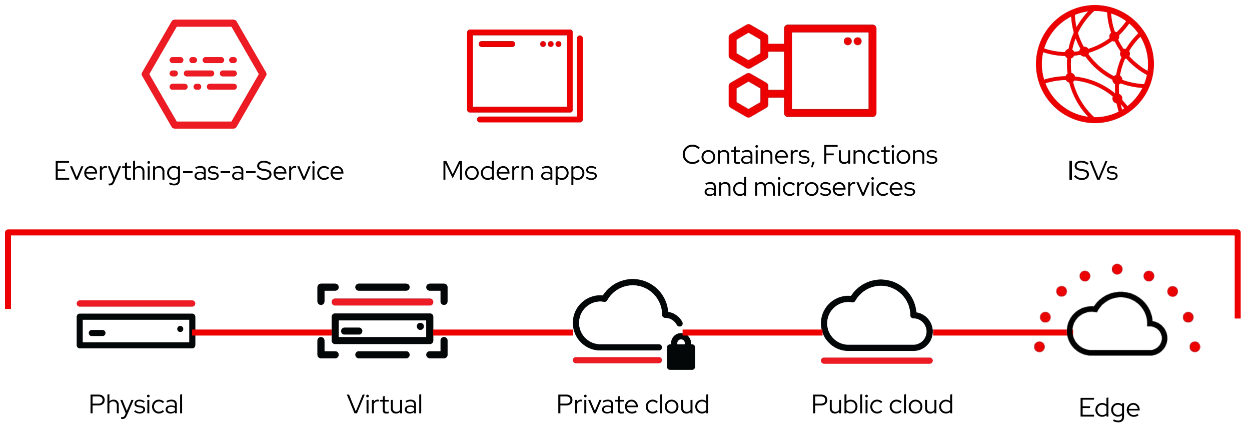
> “In a recent Gartner survey of public cloud users, 81% of respondents said they are working with **two or more providers**.
>
> Gartner 2019 Why Choose a Multicloud Strategy

> “Enterprises with a hybrid strategy grew to **58 percent** in 2019 from 51 percent in 2018.”
>
> RightScale 2019 State of the Cloud Report

Core Platform

Event-Driven Applications

Event Format



Everything-as-a-Service

Modern apps

Containers, Functions and microservices

ISVs

Physical

Virtual

Private cloud

Public cloud

Edge

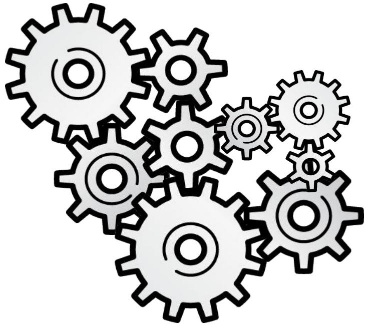
Open

Open **standards** and **community** driven development and a commitment to Open Source, and a broad ecosystem of partners, avoiding lock-in.

Hybrid

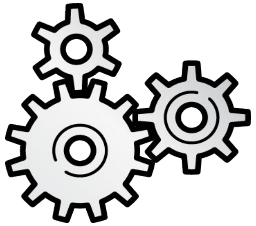
A combination of **public** and **private** clouds, with a degree of workload portability, integration, orchestration, and unified management

Serverless concepts evolution



Service

> Autonomous
> Loosely-coupled



Microservice

> Single Purpose
> Stateless
> Independently Scalable
> Automated

$f(\,)$

Function

> Single Action
> Ephemeral

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**
- ➔ Microsoft & Red Hat create **KEDA**

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
- ➔ **Solutions and outcome focused**

Serverless is still evolving...

Serverless benefits

„Přešli jsme z 6-7 sekund času spuštění s Javou na AWS Lambda na 10 milisekund s Quarkusem“

Platform that…

- Scale to 0
- Has rapid scaling (up & down)
- Enables Event Driven Architecture (EDA)
- Routing and Networking

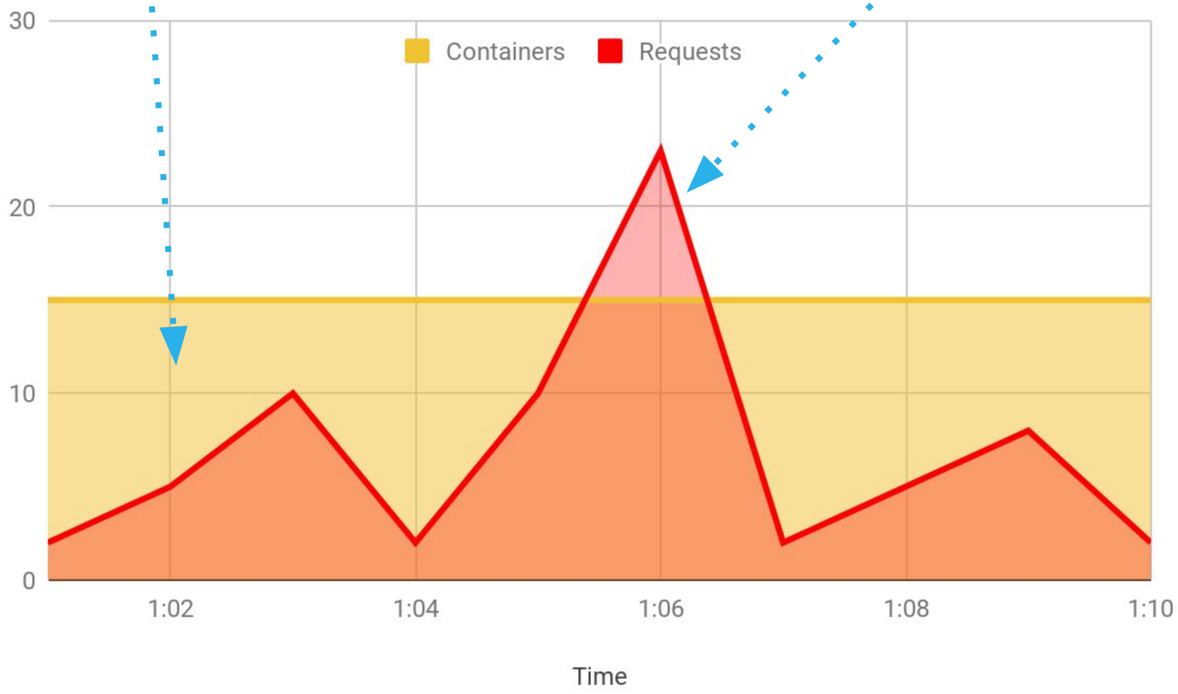
Applications that…

- Start quickly
- Respond quickly
- Are low on memory
- Require small disk size

Serverless Operational Benefits

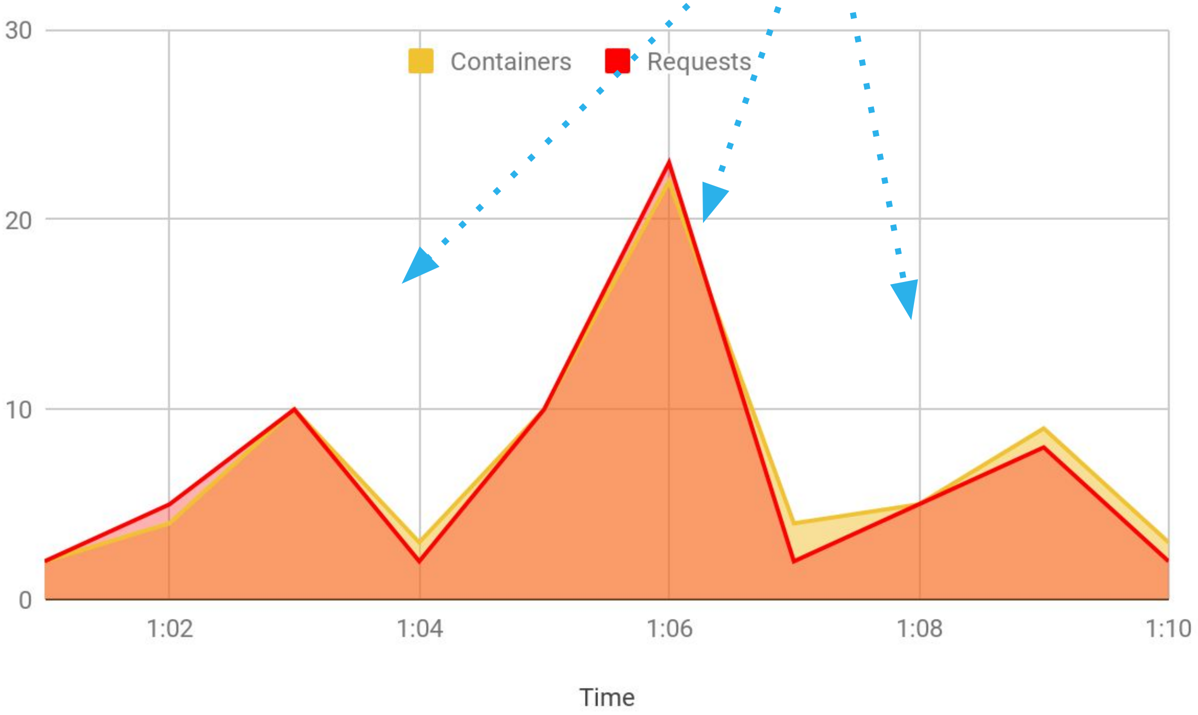
Over provisioning
Time in capacity planning
IT cost of idle resources

Under provisioning
Lost business revenue
Poor quality of service



NOT Serverless

More applications
Direct line between IT
costs & business revenue

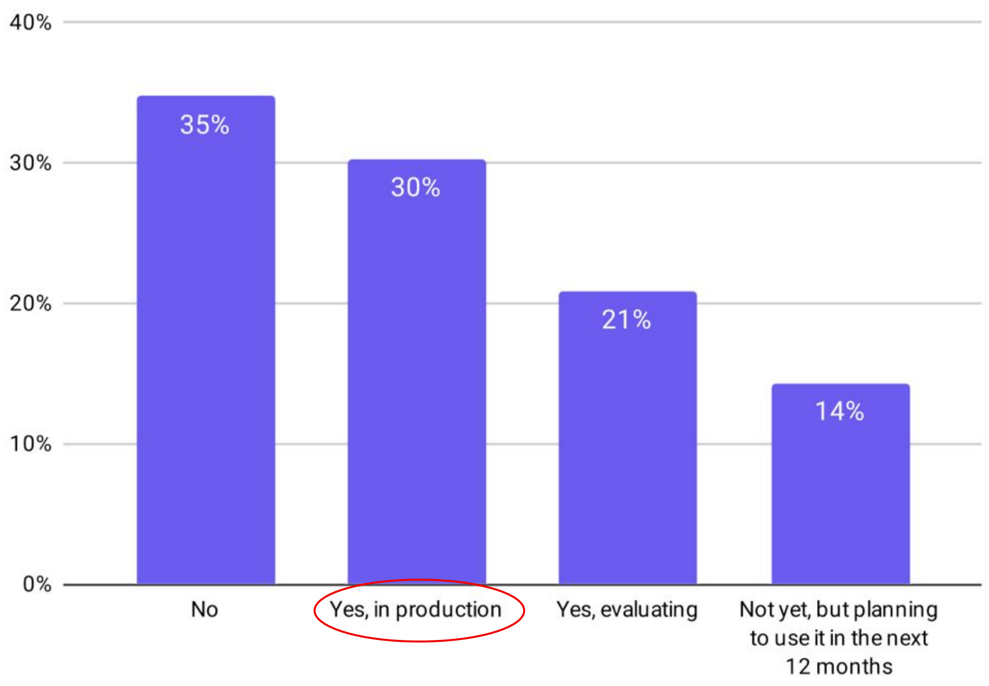


with Serverless

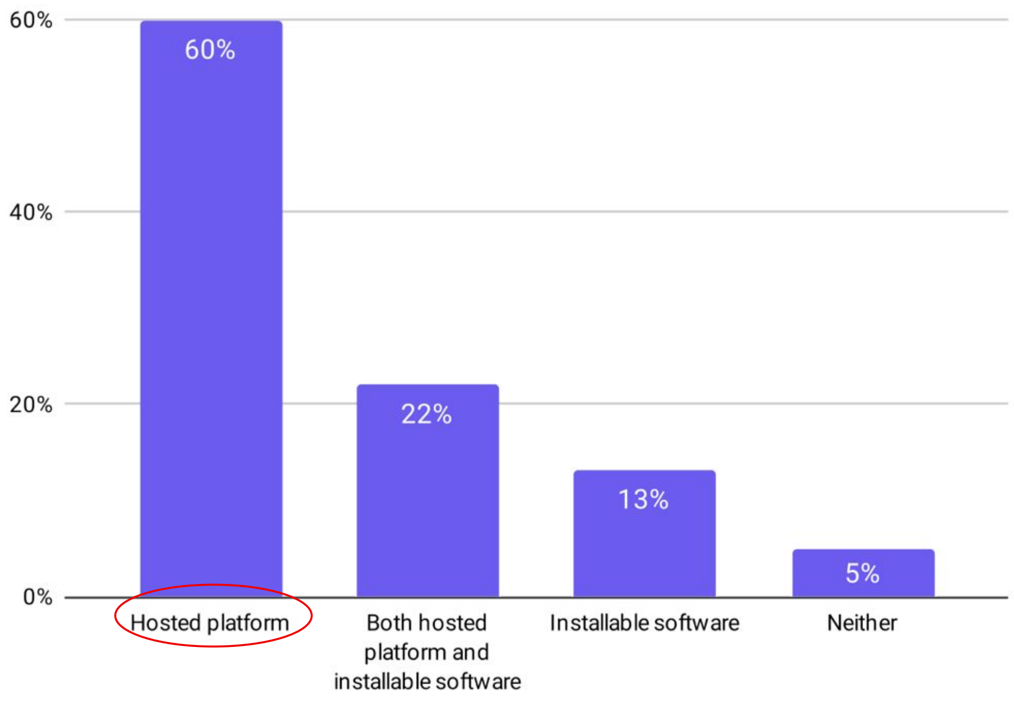
Serverless usage – CNCF study*1

30% již používá Serverless technologii v komerčním prostředí a 60% využívá služby od poskytovatelů cloudu

Is your organization using serverless technology?



How is your organization using serverless technology?



*1 CNCF Survey 2020, https://www.cncf.io/wp-content/uploads/2020/11/CNCF_Survey_Report_2020.pdf

Serverless usage – CNCF study*1

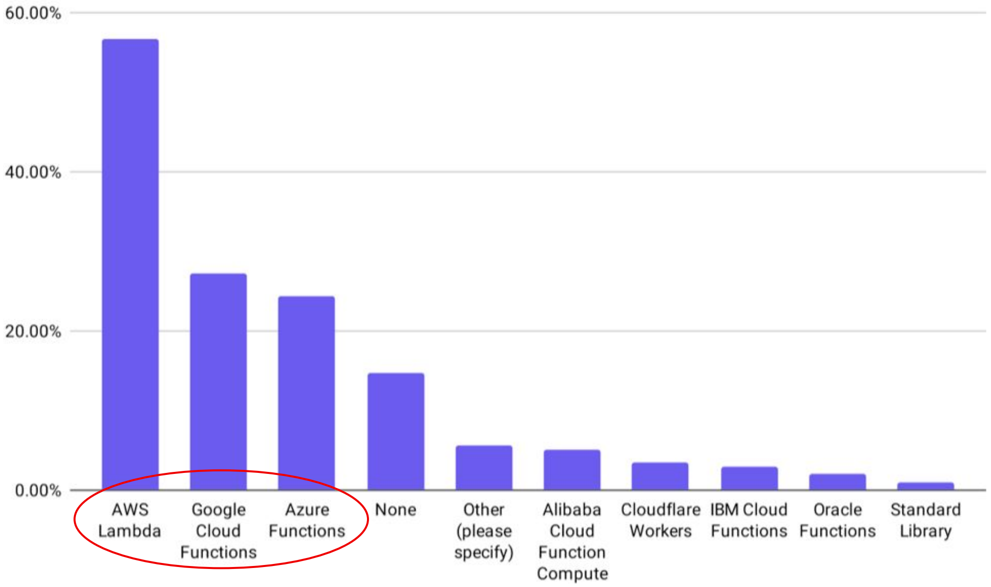
AWS Lambda vede, pokud jde o využití Serverless technologie poskytované poskytovateli cloudu

- AWS Lambda (57 %), Google Cloud Functions (27 %) a Azure Functions (24 %)

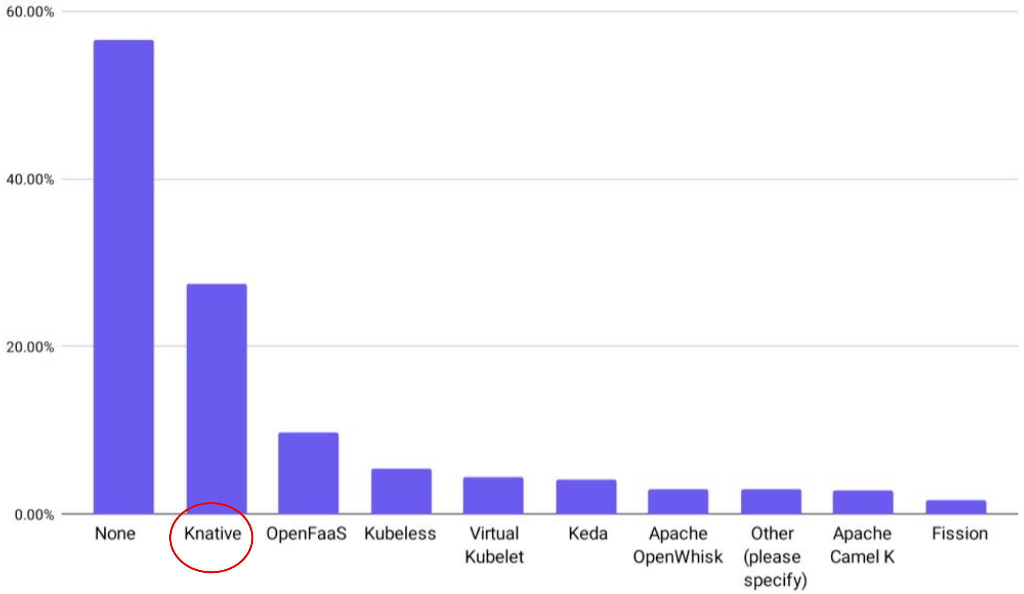
Knative vede, jakmile jde o využití vlastní instalace software

- Knative (27 %), OpenFaaS (10 %) a Kubeless (5 %)

If your organization is using serverless via a "hosted platform", which hosted serverless platform do they use?



If your organization is using serverless via a "installable software", which hosted installable software do they use?



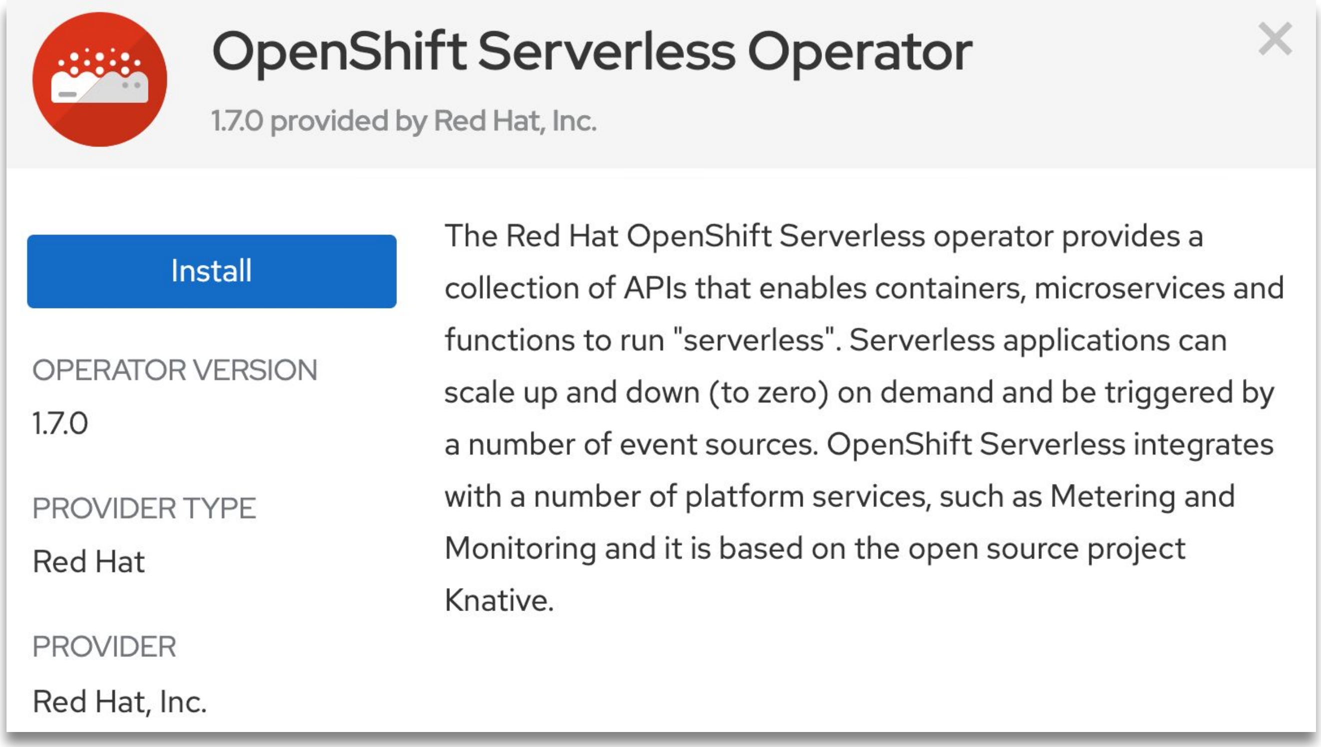
*1 CNCF Survey 2020, https://www.cncf.io/wp-content/uploads/2020/11/CNCF_Survey_Report_2020.pdf

OpenShift Serverless instalace & použití

Installation experience

"Easy day 1 and even better for day 2"

- ➤ Click Install experience
- ➤ Developer & admin experience in Console
- ➤ Built-in event sources
- ➤ No external dependencies.
- ➤ ***"Just works."***



OpenShift Serverless Operator

1.7.0 provided by Red Hat, Inc.

OPERATOR VERSION
1.7.0

PROVIDER TYPE
Red Hat

PROVIDER
Red Hat, Inc.

The Red Hat OpenShift Serverless operator provides a collection of APIs that enables containers, microservices and functions to run "serverless". Serverless applications can scale up and down (to zero) on demand and be triggered by a number of event sources. OpenShift Serverless integrates with a number of platform services, such as Metering and Monitoring and it is based on the open source project Knative.

Developer experience

- </> Java
- </> Node
- </> Python
- </> TypeScript
- </> Go
- </> Rust
- </> Ruby
- </> Ruby on Rails
- </> PHP
- </> Perl
- </> .NET Core

```
$ kn service create --image=
$ kn func create -l
```

CLI

UI

OpenShift Serverless

OPENSHIFT

Red Hat Enterprise Linux CoreOS

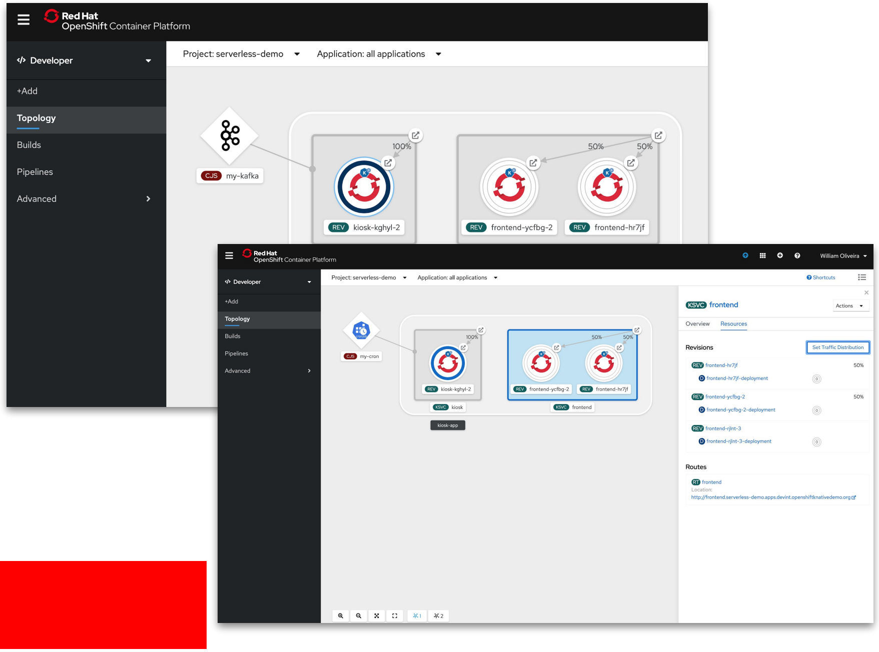
Physical

Virtual

Private cloud

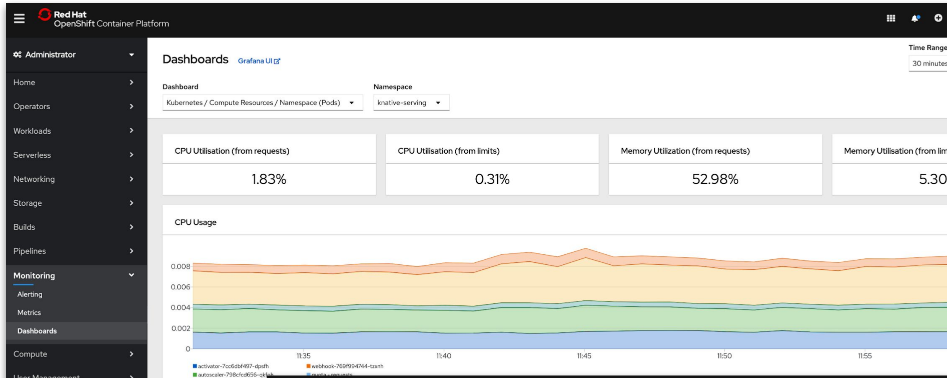
Public cloud

Edge



Admin experience

- Monitoring, Metering and Logging
- Disconnected install support (air-gapped)
- Egress proxy with TLS support
- Over the air updates and patches



Dashboards

Dashboard: Kubernetes / Compute Resources / Namespace (Pods)

Namespace: knative-serving

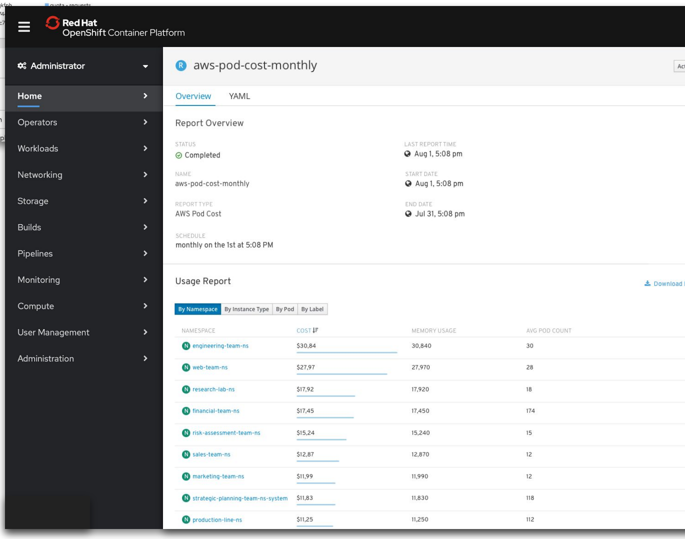
30 minutes

CPU Utilisation (from requests)	CPU Utilisation (from limits)	Memory Utilization (from requests)	Memory Utilisation (from lim
1.83%	0.31%	52.98%	5.30

CPU Usage

CPU Quota

Pod
activator-7cc6dbf497-dpsfh
autoscaler-798cfcd656-qkfp



aws-pod-cost-monthly

Report Overview

STATUS
Completed

NAME
aws-pod-cost-monthly

REPORT TYPE
AWS Pod Cost

SCHEDULE
monthly on the 1st at 5:08 PM

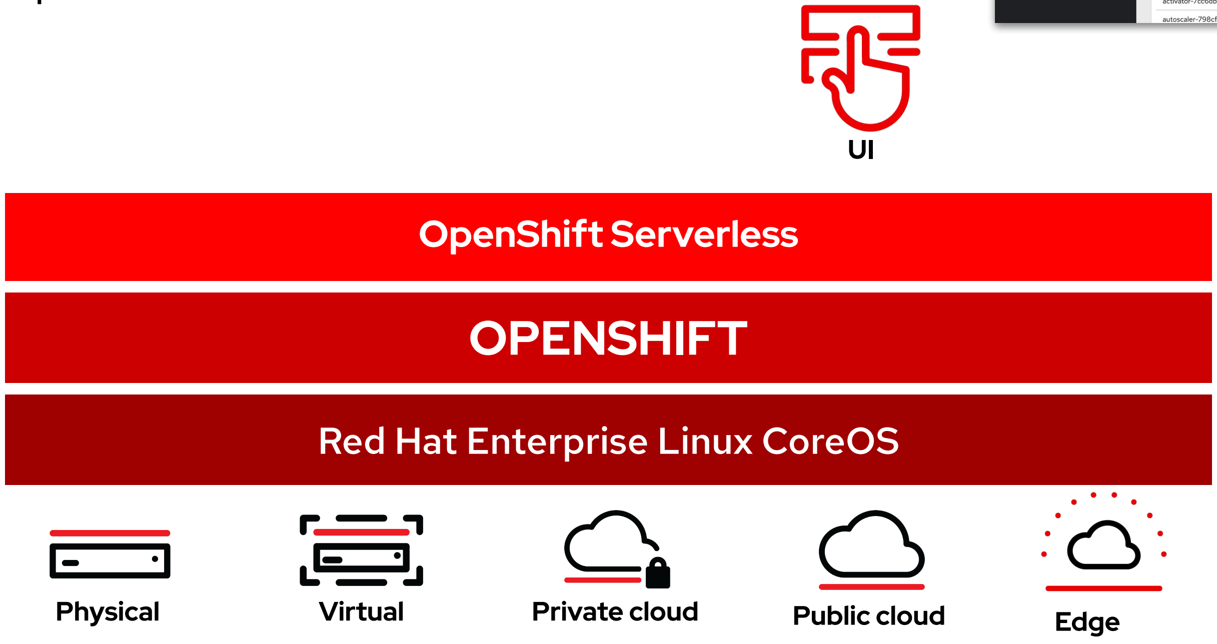
LAST REPORT TIME
Aug 1, 5:08 pm

START DATE
Aug 1, 5:08 pm

END DATE
Jul 31, 5:08 pm

Usage Report

NAMESPACE	COST	MEMORY USAGE	AVG POD COUNT
engineering-team-ns	$30,84	30,840	30
web-team-ns	$27,97	27,970	28
research-lab-ns	$17,92	17,920	18
financial-team-ns	$17,45	17,450	174
risk-assessment-team-ns	$15,24	15,240	15
sales-team-ns	$12,87	12,870	12
marketing-team-ns	$11,99	11,990	12
strategic-planning-team-ns-system	$11,83	11,830	118
production-line-ns	$11,25	11,250	112


UI

OpenShift Serverless

OPENSHIFT

Red Hat Enterprise Linux CoreOS

Physical

Virtual

Private cloud

Public cloud

Edge

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: markusthoemmes/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
```

```
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
```

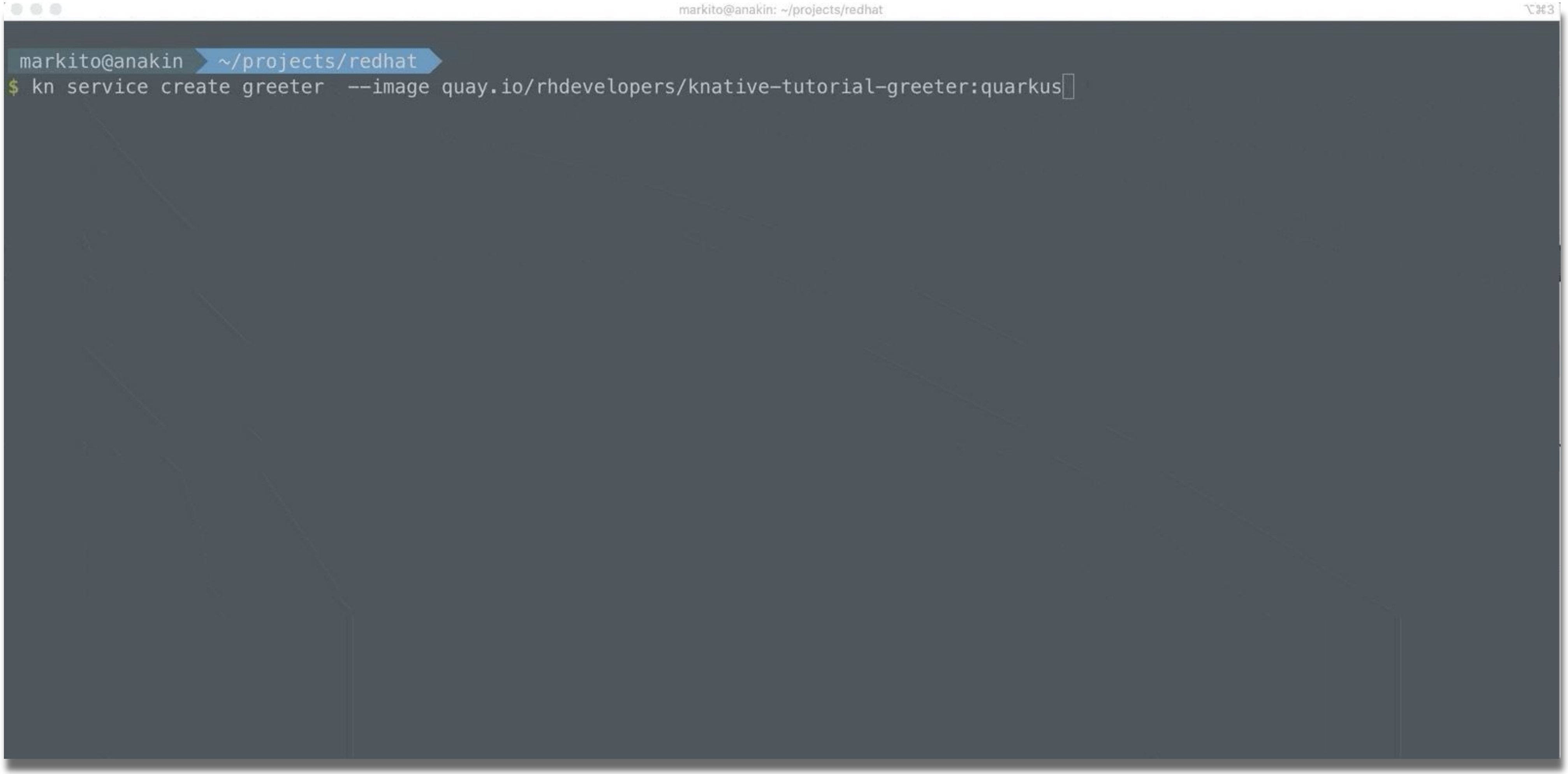
~70 lines

Knative

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

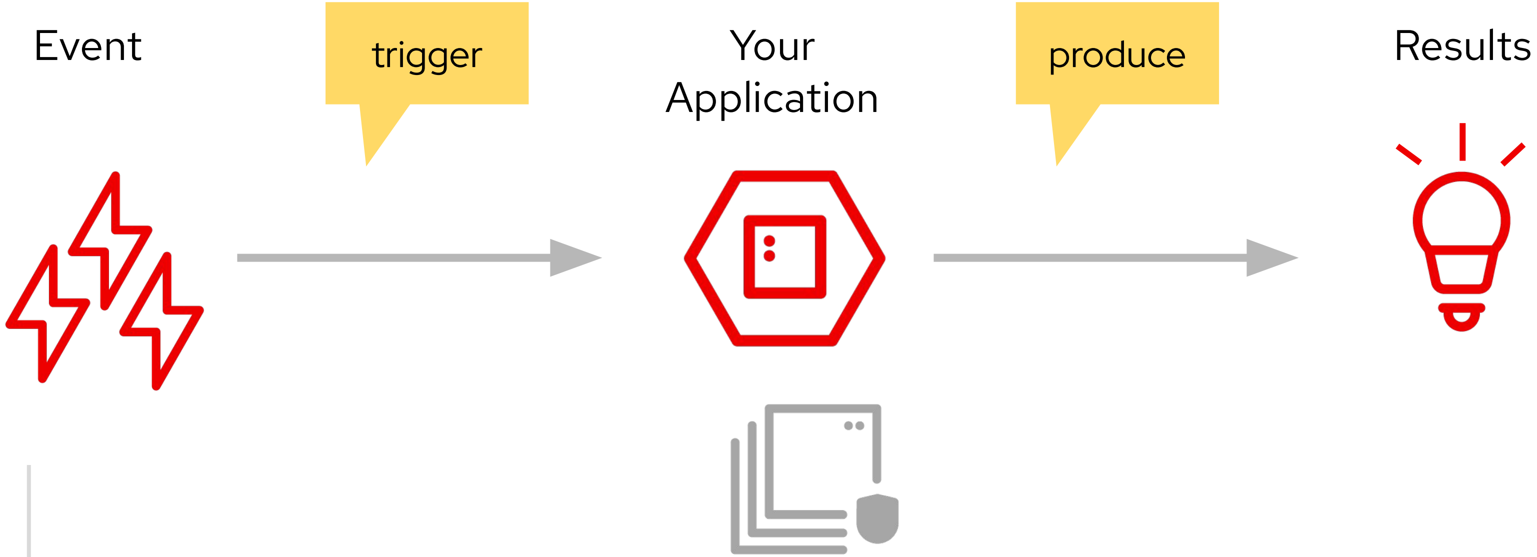
22 lines

Hello World with Quarkus!



```
markito@anakin  ~/projects/redhat
$ kn service create greeter  --image quay.io/rhdevelopers/knative-tutorial-greeter:quarkus
```

The "Serverless Pattern"



HTTP Requests

Kafka Messages

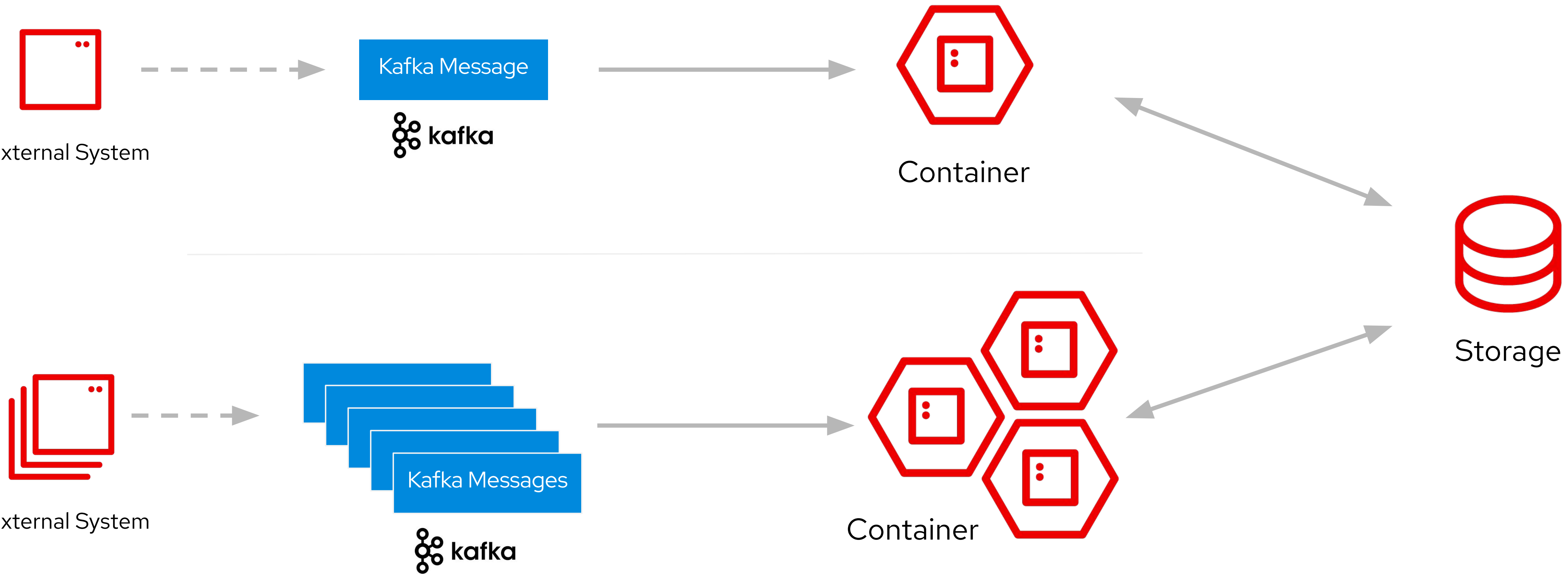
Image Uploaded

New Order

Login from user

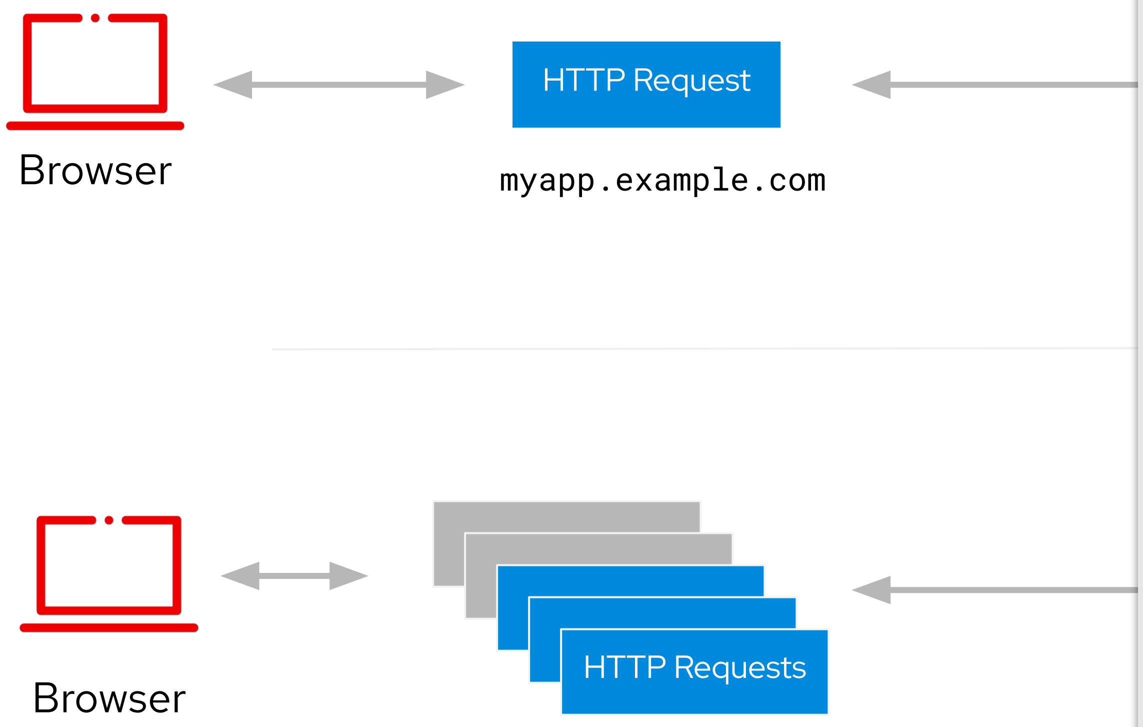
The "Serverless Pattern"

Processing a Kafka message



The "Serverless Pattern"

A serverless web application

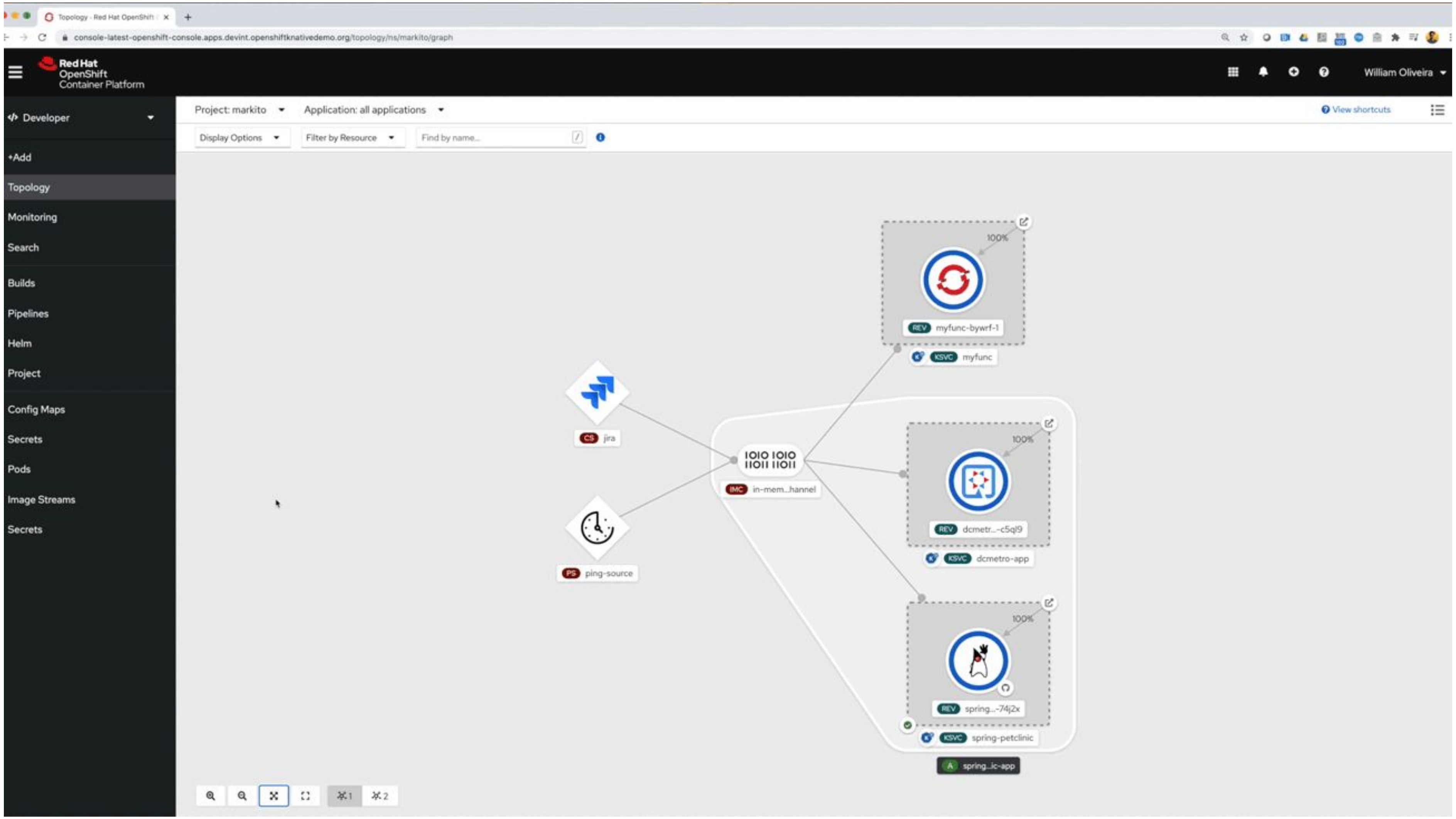


Benefits of this model:

- No need to setup auto-scaling and load balancers
  - Scale down and save resources when needed.
  - Scale up to meet the demand.
- No tickets to configure SSL for applications
- Enable Event Driven Architectures (EDA) patterns
- Enable teams to associate cost with IT
- Modernize existing applications to run as serverless containers

Generally Available

Serverless User Experience



console-latest-openshift-console.apps.devint.openshiftknativedemo.org/topology/ns/markito/graph

Project: markito

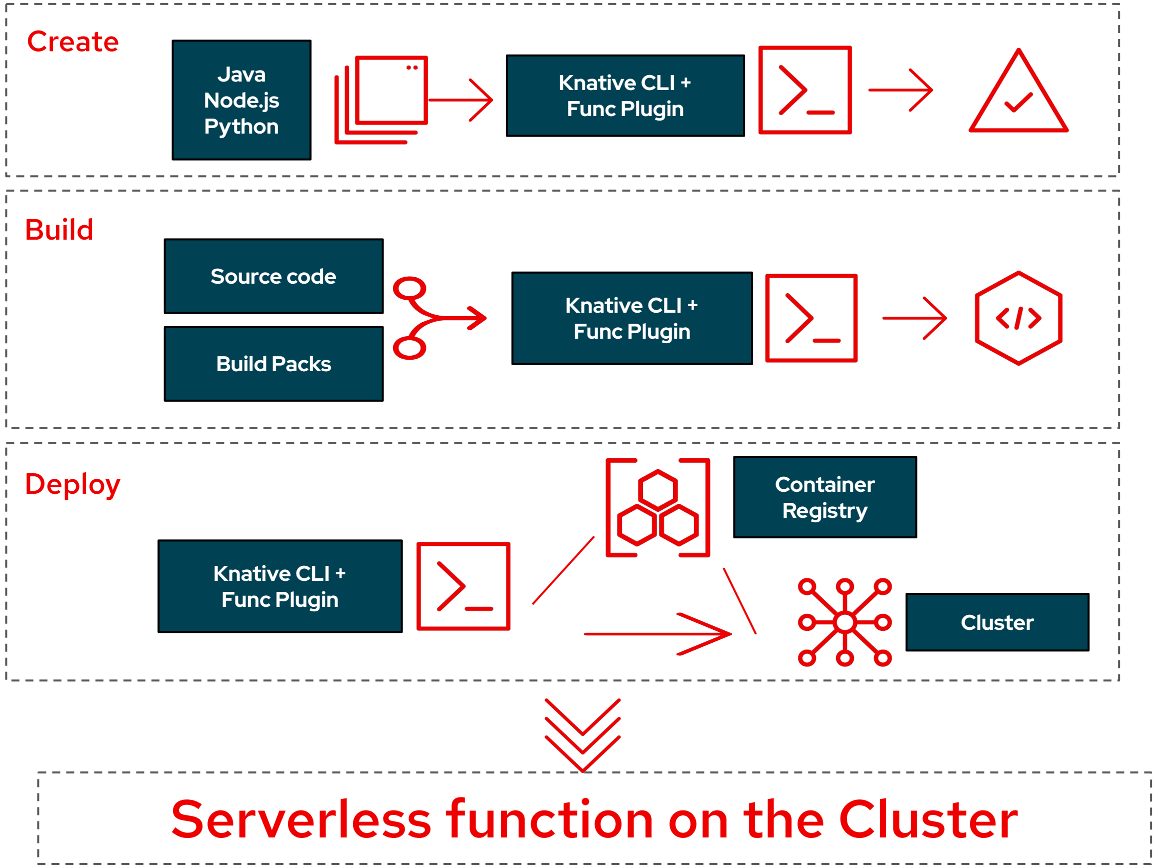
Application: all applications

OpenShift Serverless functions

Simple and quick Creation, Build, and Deploy of Applications

OpenShift Serverless Functions allows users to consume events via functions based APIs and provide a simplified programming model for developers and data scientists alike.

- Simplified deployments
- Reduced programming complexity
- Secure, consistent programming models
- Quarkus, Node.js, Python, Go and Spring Boot.
- Event Driven applications made easy



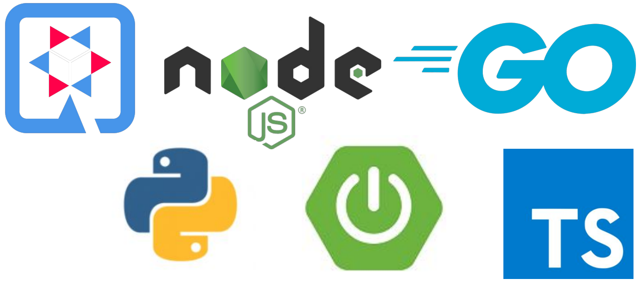
Technology Preview

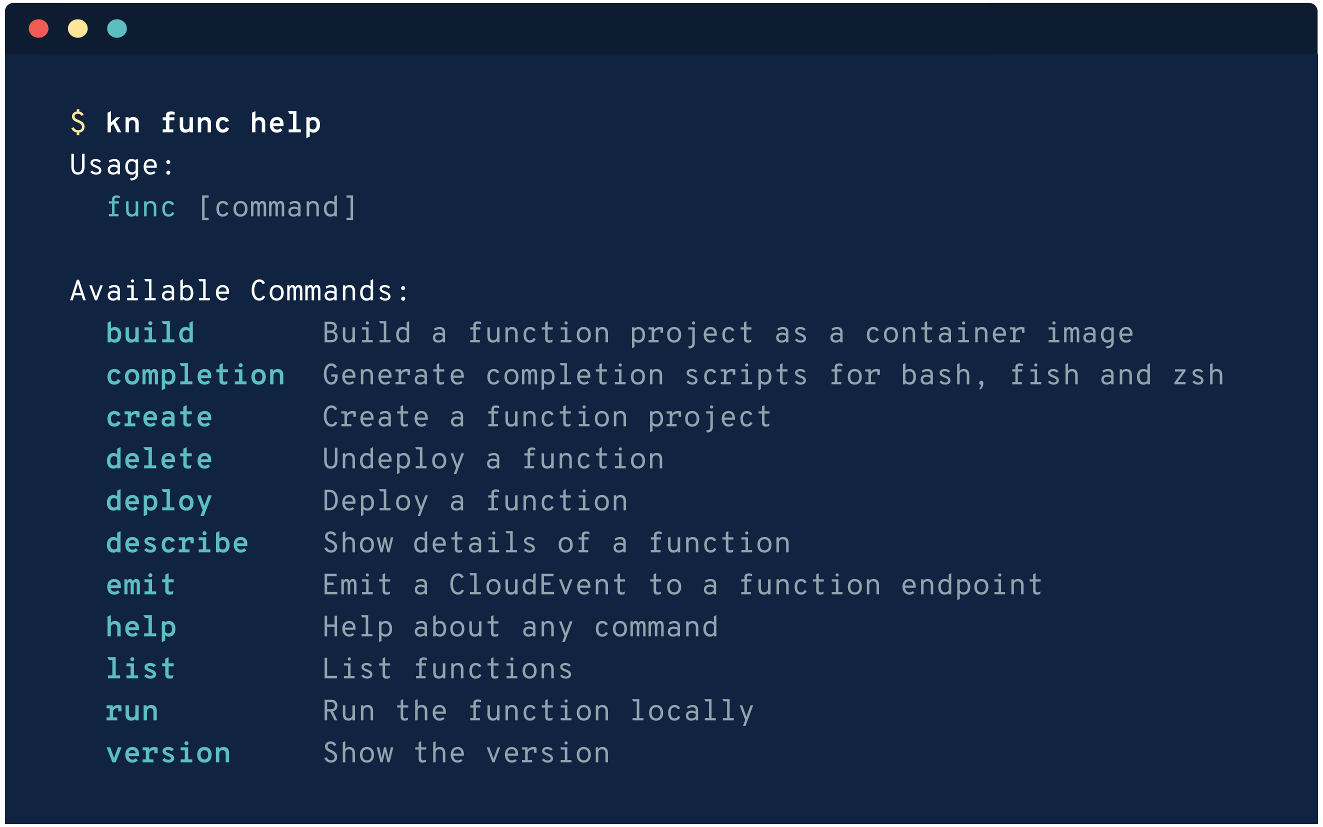
Serverless functions

Since OpenShift Serverless 1.15

Powerful CLI experience

- ✓ Local Developer Experience
- ✓ Based on CNCF Buildpacks
- ✓ Deploy as Knative Service
- ✓ Project templates
- ✓ Support for Cloud Events/HTTP
- ✓ **Runtimes:**



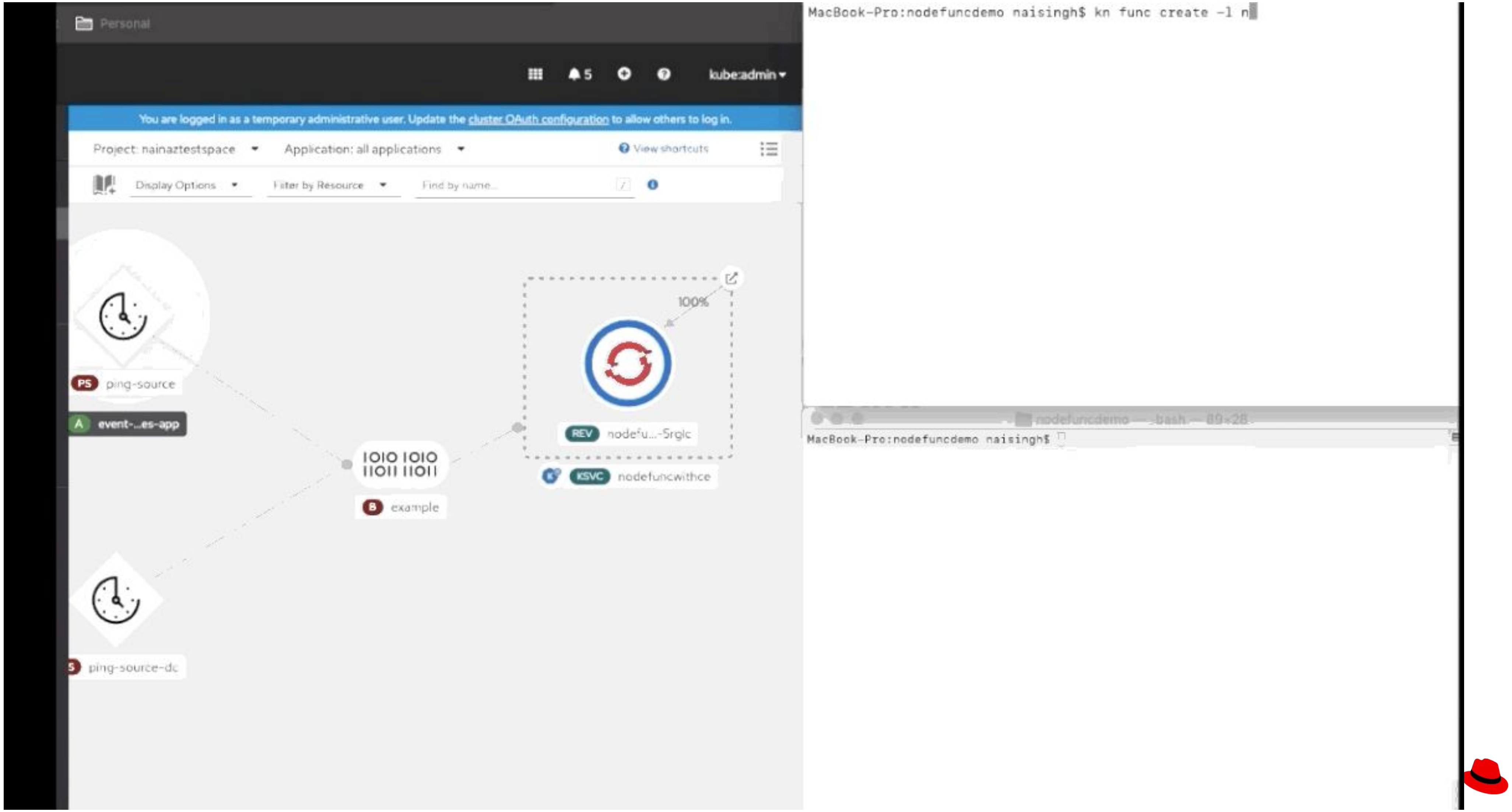


```
$ kn func help
Usage:
  func [command]

Available Commands:
  build       Build a function project as a container image
  completion  Generate completion scripts for bash, fish and zsh
  create      Create a function project
  delete      Undeploy a function
  deploy      Deploy a function
  describe    Show details of a function
  emit        Emit a CloudEvent to a function endpoint
  help        Help about any command
  list        List functions
  run         Run the function locally
  version     Show the version
```

Technology Preview

Functions



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

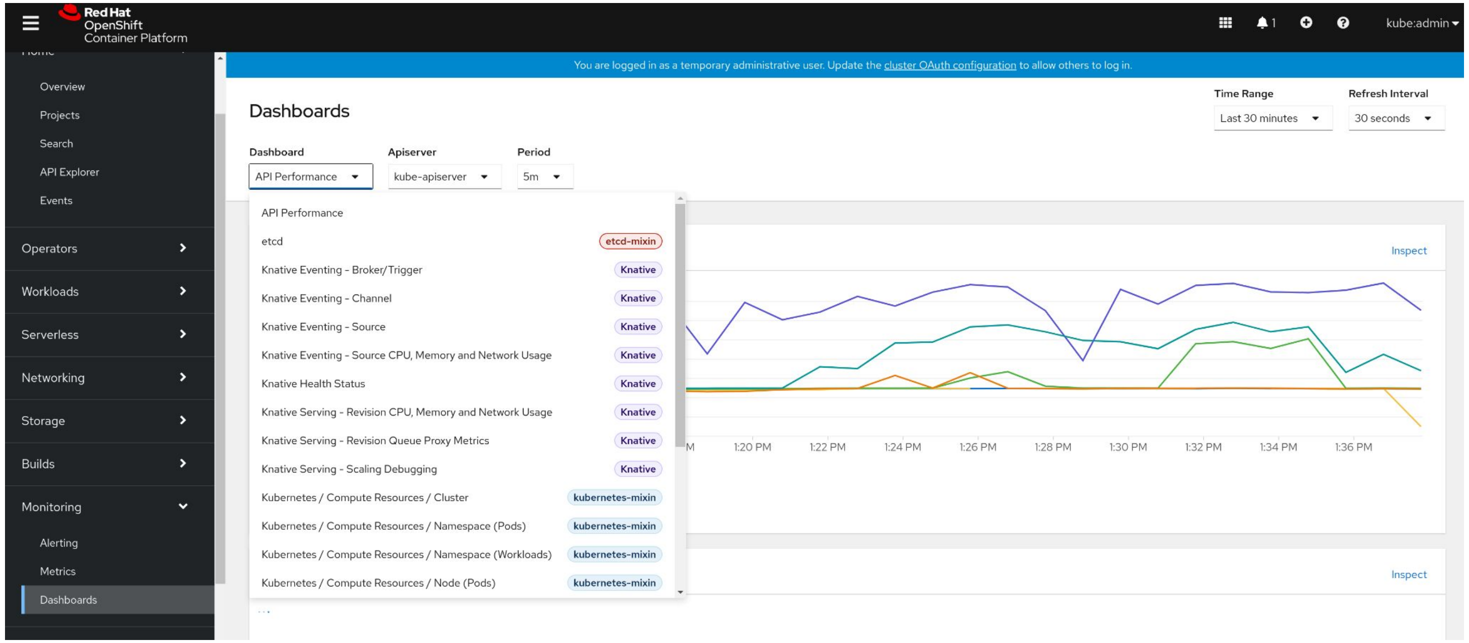
```
MacBook-Pro:nodefuncdemo naisingh$ kn func create -l n
```

```
MacBook-Pro:nodefuncdemo naisingh$
```

Serverless Monitoring - Dashboards

Metrics visualization for:

- Knative Serving, Eventing, Kafka readiness status
- Knative Eventing Broker/Trigger event handling
- Knative Eventing Sources event emission (eg. event counts)
- Knative Eventing Sources resource consumption (eg. CPU, MEM etc)
- Knative In-Memory, Kafka Channels event handling
- Knative Serving Autoscaling behavior
- Knative Queue proxy request handling
- Knative Service resource consumption (eg. CPU, MEM etc)



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

Dashboards

Time Range: Last 30 minutes

Refresh Interval: 30 seconds

Dashboard: API Performance

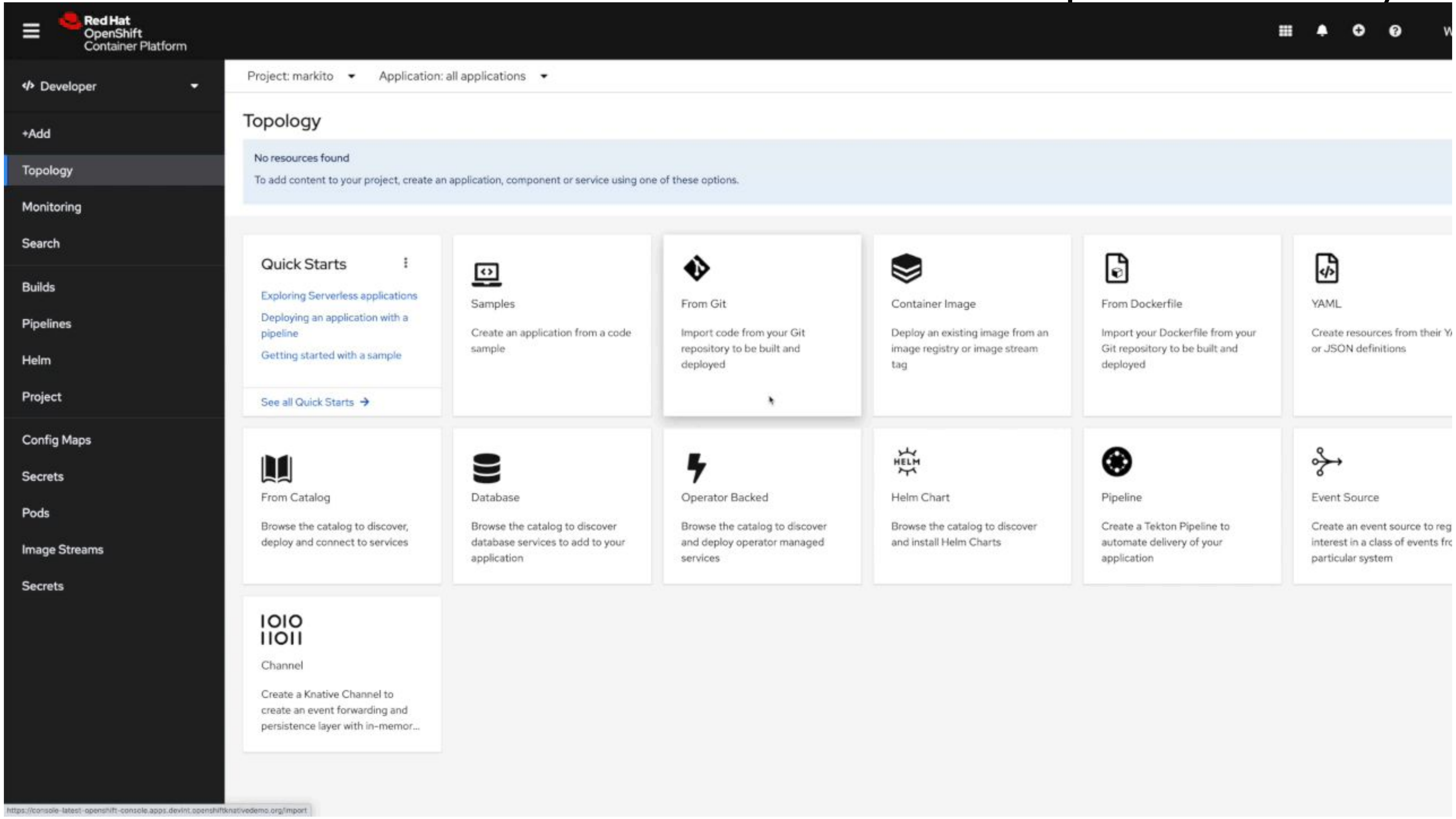
Apiserver: kube-apiserver

Period: 5m

API Performance	
etcd	etcd-mixin
Knative Eventing - Broker/Trigger	Knative
Knative Eventing - Channel	Knative
Knative Eventing - Source	Knative
Knative Eventing - Source CPU, Memory and Network Usage	Knative
Knative Health Status	Knative
Knative Serving - Revision CPU, Memory and Network Usage	Knative
Knative Serving - Revision Queue Proxy Metrics	Knative
Knative Serving - Scaling Debugging	Knative
Kubernetes / Compute Resources / Cluster	kubernetes-mixin
Kubernetes / Compute Resources / Namespace (Pods)	kubernetes-mixin
Kubernetes / Compute Resources / Namespace (Workloads)	kubernetes-mixin
Kubernetes / Compute Resources / Node (Pods)	kubernetes-mixin

marked with the *knative* label

Serverless & Pipelines for CI/CD



Topology

No resources found

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

Quick Starts

Exploring Serverless applications

Deploying an application with a pipeline

Getting started with a sample

See all Quick Starts →

Samples

Create an application from a code sample

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 repository to be built and deployed

YAML

Create resources from their Y
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

Pipeline

Create a Tekton Pipeline to automate delivery of your application

Event Source

Create an event source to reg
interest in a class of events fro
particular system

Channel

Create a Knative Channel to create an event forwarding and persistence layer with in-memor...

https://console-latest-openshift-console.apps.devint.openshiftknativedemo.org/import

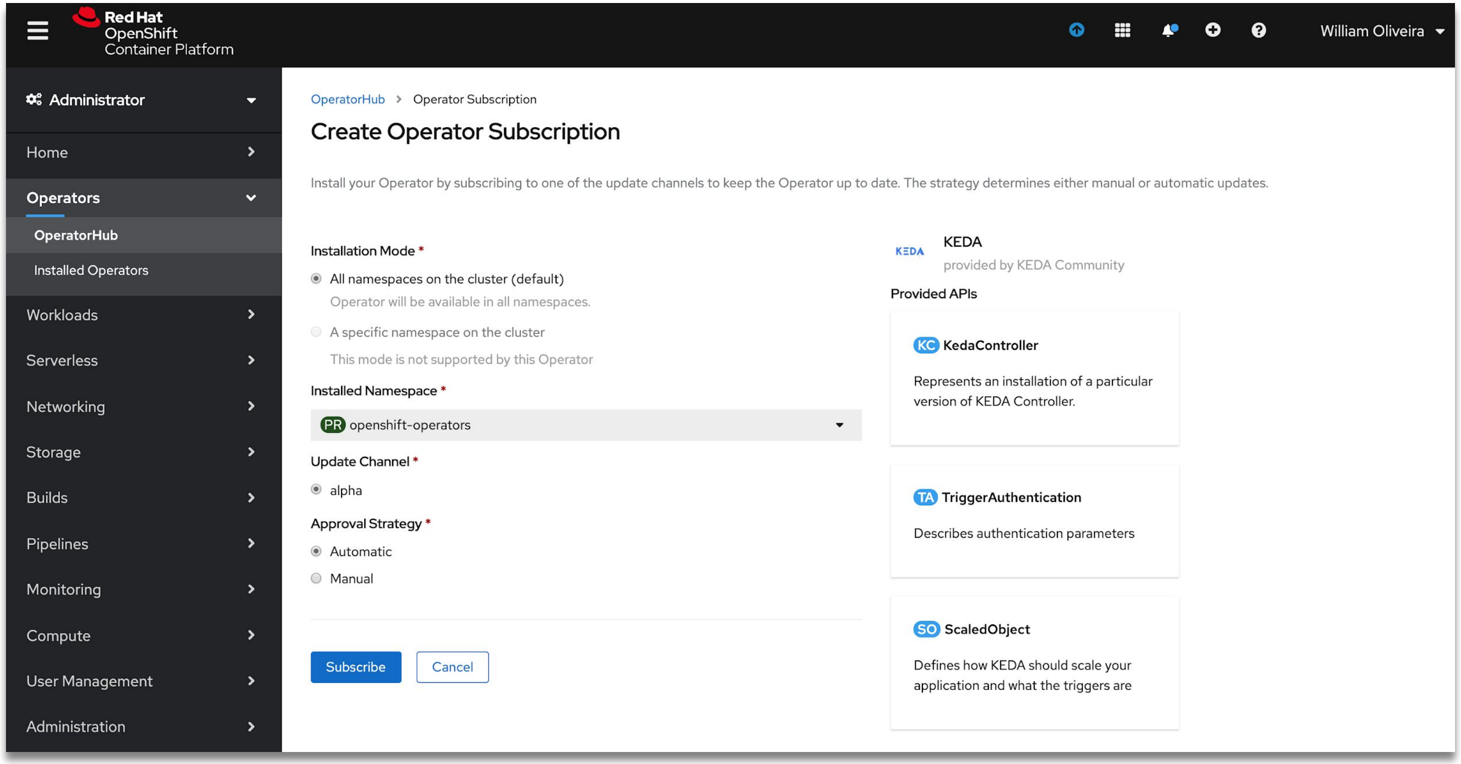
OPENSHIFT SERVERLESS

Red Hat | Microsoft Azure

KEDA

Star 3k Fork 334

CLOUD NATIVE SANDBOX



Create Operator Subscription

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Installation Mode *

● All namespaces on the cluster (default)
Operator will be available in all namespaces.

○ A specific namespace on the cluster
This mode is not supported by this Operator

Installed Namespace *

PR openshift-operators

Update Channel *

● alpha

Approval Strategy *

● Automatic

○ Manual

Subscribe Cancel

KEDA
provided by KEDA Community

Provided APIs

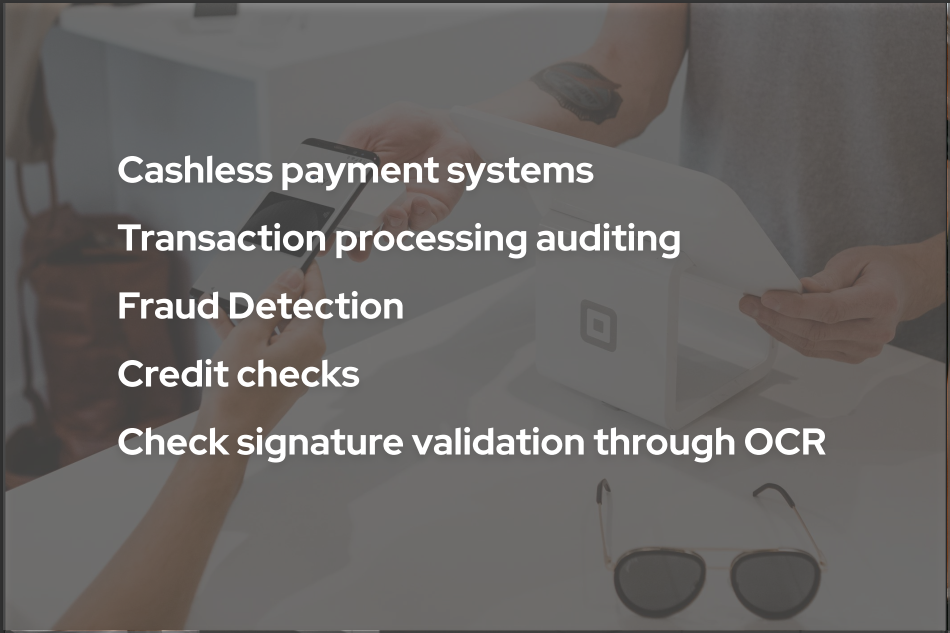
KC KedaController
Represents an installation of a particular version of KEDA Controller.

TA TriggerAuthentication
Describes authentication parameters

SO ScaledObject
Defines how KEDA should scale your application and what the triggers are

"KEDA serves as a Kubernetes Metrics Server and allows users to define autoscaling rules and scalers for any containerized workload.
It enables Azure Functions on Kubernetes.

Serverless
Use cases



Cashless payment systems

Transaction processing auditing

Fraud Detection

Credit checks

Check signature validation through OCR

Product thumbnail generation

Chatbots and CRM functions

Marketing Campaign notifications

Sales Audit

Content Push

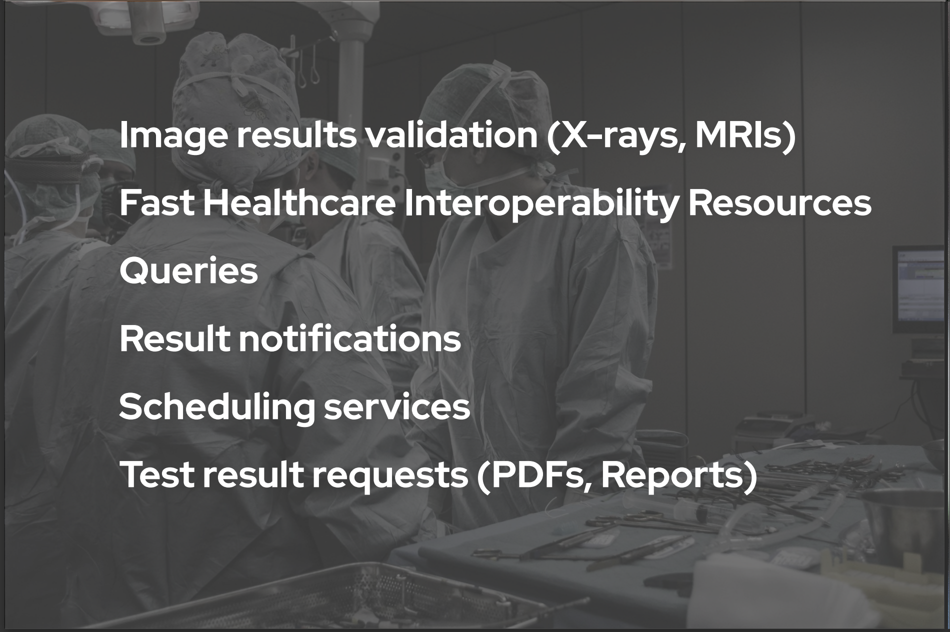


Image results validation (X-rays, MRIs)

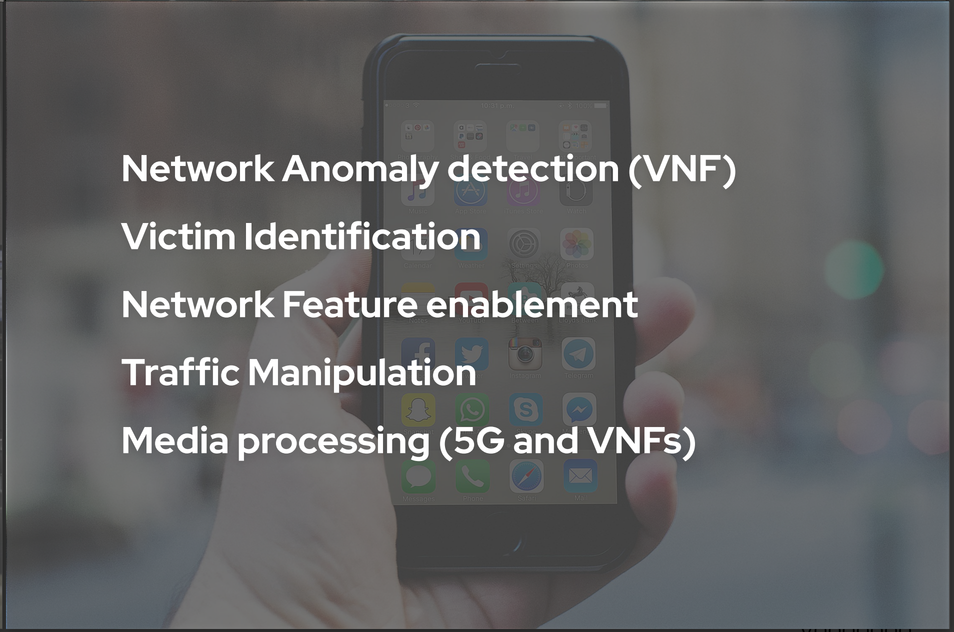
Fast Healthcare Interoperability Resources

Queries

Result notifications

Scheduling services

Test result requests (PDFs, Reports)



Network Anomaly detection (VNF)

Victim Identification

Network Feature enablement

Traffic Manipulation

Media processing (5G and VNFs)

Building on OpenShift Serverless with Red Hat Services

Connected Services

How Knative services interact with the outside world.

APACHE Camel

Service Orchestrator

Composing multiple services together into an application.

APACHE Camel

Event Streaming

All modern architectures need some Kafka.

STRIMZI

API Gateway

Next gen APIs still require management.

3scale BY RED HAT

Implementing Services

Functions, languages, and the vagaries of cold starts.

QUARKUS

The Dirty Word in Serverless

Yep, you still need state to handle long-lived orchestration.

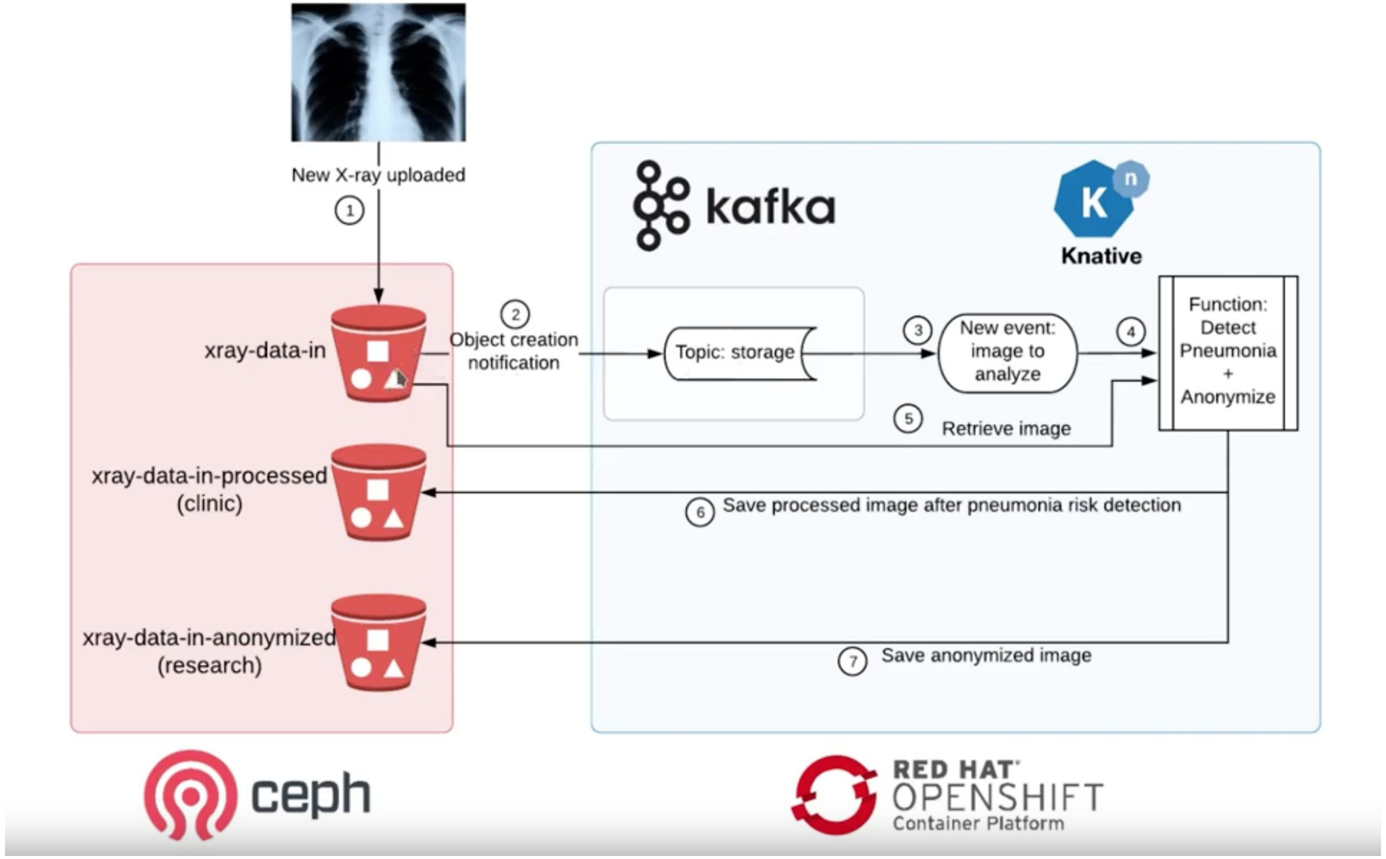
Kogito

Ceph + Kafka + Serverless

"Automated AI/ML Data Pipelines"

Common Use cases

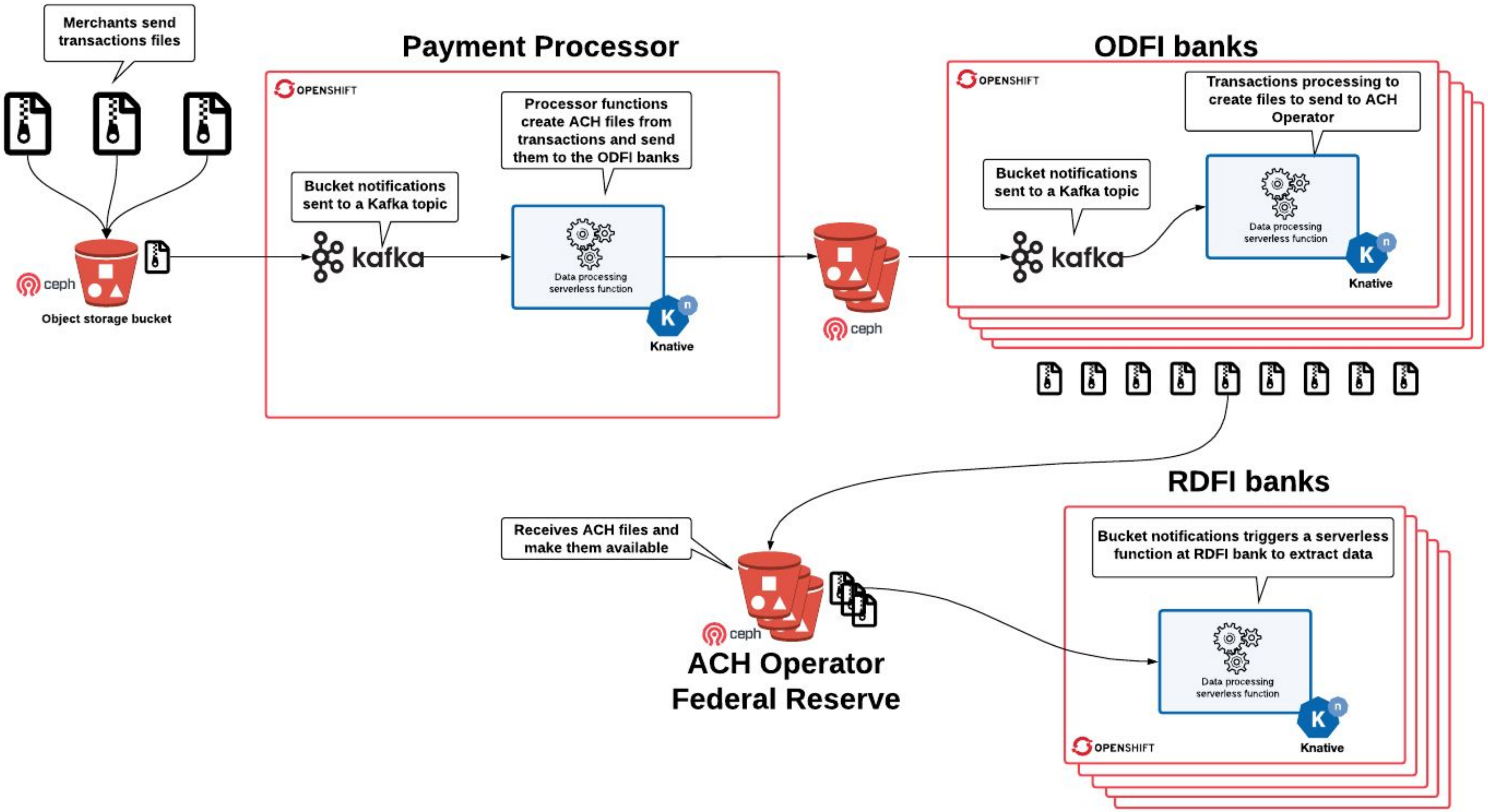
- Image analysis*
  (medical, AI/ML, media)
- Video format transcoding
- File type conversion
  (financial, medical)
- Data extraction
- Data Transformation
- Invoice Generation


New X-ray uploaded
1
xray-data-in
2
Object creation notification
kafka
Knative
Topic: storage
3
New event: image to analyze
4
Function: Detect Pneumonia + Anonymize
5
Retrieve image
xray-data-in-processed (clinic)
6
Save processed image after pneumonia risk detection
xray-data-in-anonymized (research)
7
Save anonymized image
ceph
RED HAT OPENSHIFT Container Platform

TensorFlow

https://medium.com/analytics-vidhya/automated-data-pipeline-using-ceph-notifications-and-kserving-5e1e9b996661

Serverless Open Banking



Merchants send transactions files

Payment Processor

Bucket notifications sent to a Kafka topic

Processor functions create ACH files from transactions and send them to the ODFI banks

ODFI banks

Bucket notifications sent to a Kafka topic

Transactions processing to create files to send to ACH Operator

RDFI banks

Bucket notifications triggers a serverless function at RDFI bank to extract data

Receives ACH files and make them available

ACH Operator
Federal Reserve

ceph
Object storage bucket
OPENSHIFT
kafka
Data processing serverless function
Knative
ceph
OPENSHIFT
kafka
Data processing serverless function
Knative
ceph
Data processing serverless function
Knative
OPENSHIFT

OpenShift Serverless

Links

- ✓ https://www.openshift.com/serverless
- ✓ Knative Tutorial by Red Hat Developers
- ✓ Knative Cookbook by Red Hat Developers
- ✓ try.openshift.com/serverless (coming soon)

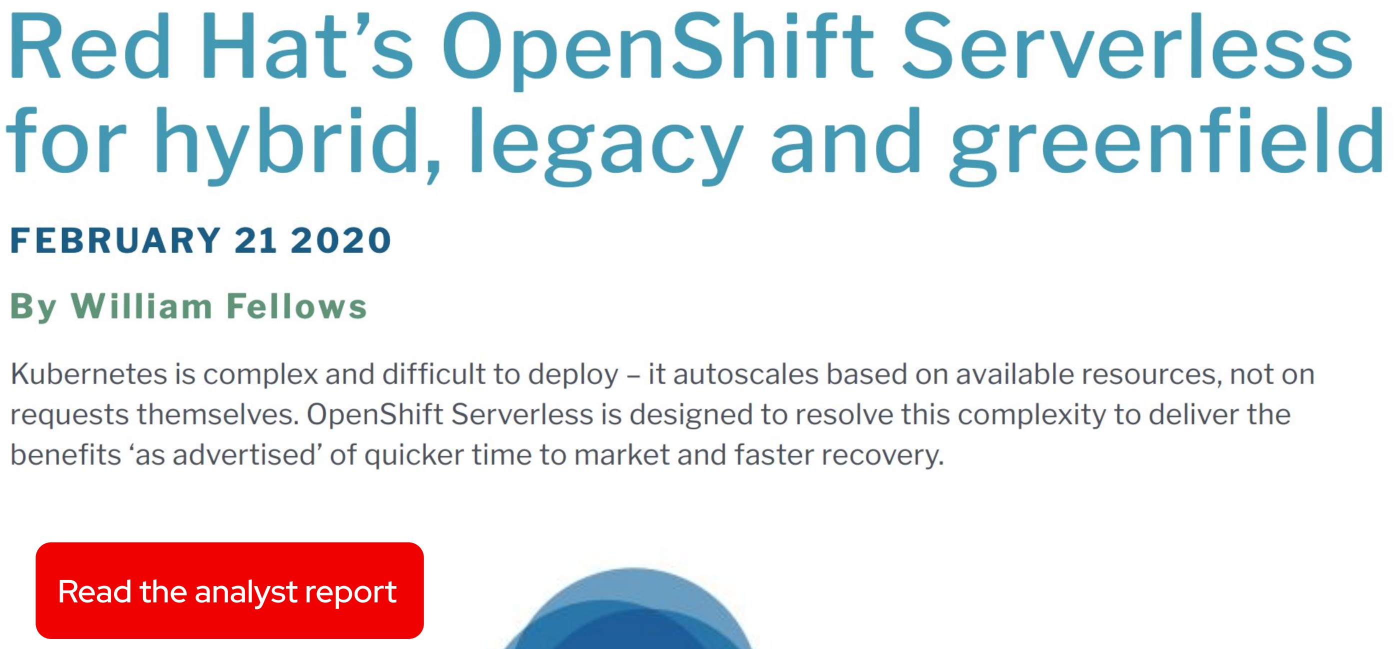
O'REILLY

Knative Cookbook

Building Effective Serverless Applications with Kubernetes and OpenShift

Compliments of Red Hat

Burr Sutter & Kamesh Sampath



Red Hat's OpenShift Serverless for hybrid, legacy and greenfield

FEBRUARY 21 2020

By William Fellows

Kubernetes is complex and difficult to deploy – it autoscales based on available resources, not on requests themselves. OpenShift Serverless is designed to resolve this complexity to deliver the benefits 'as advertised' of quicker time to market and faster recovery.

Read the analyst report

451 Research®

Red Hat
Summit

THANK YOU

linkedin.com/company/Red-Hat

facebook.com/RedHatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat