

OpenShift Serverless

Wanja Pernath

EMEA Partner Enablement, OpenShift Dev
wanja@redhat.com



linkedin.com/company/red-hat



facebook.com/redhatinc



youtube.com/user/RedHatVideos



twitter.com/RedHat

Self introduction

Name: Wanja Pernath

Email: wpernath@redhat.com

Base: Germany (very close to the Alps)

Role: EMEA Technical Partner Development Manager

- OpenShift and MW

Experience: Years of Consulting, Training, PreSales at
Red Hat and before

Twitter: <https://twitter.com/wpernath>

LinkedIn: <https://www.linkedin.com/in/wanjapernath/>



First book just published

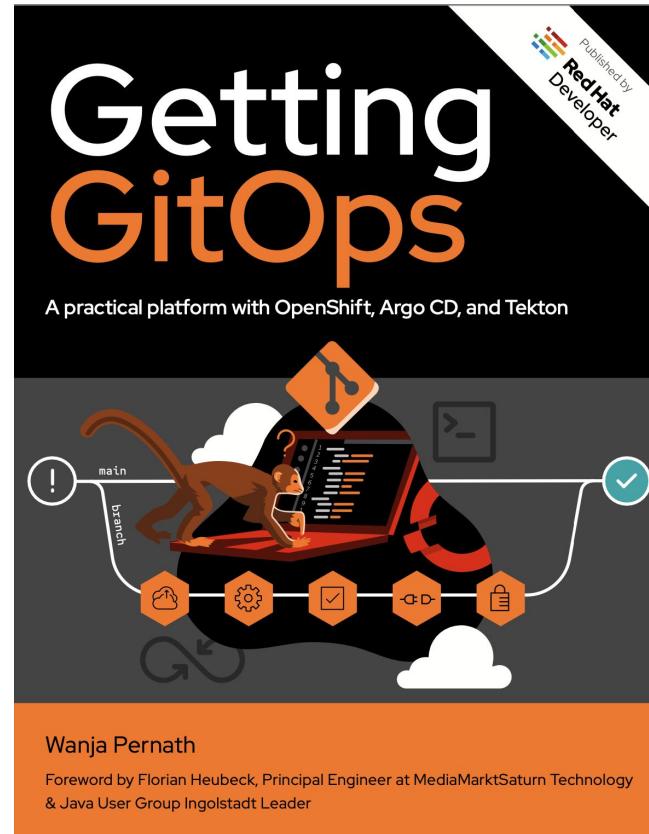
Getting GitOps

A technical blueprint for developing with Kubernetes and OpenShift based on a REST microservice example written with Quarkus

Technologies discussed: Quarkus, Helm Charts, Kustomize, Tekton Pipelines, Kubernetes Operators, OpenShift Templates, ArgoCD, CI/CD, GitOps....

Download for free at:

<https://developers.redhat.com/e-books/getting-gitops-practical-platform-openshift-argo-cd-and-tekton>



Agenda

Agenda

- **OpenShift Serverless**
 - What and why?
 - Knative-serving demo
 - Knative-serving blue/green deployment demo
 - knative -serving autoscaling demo
 - Use cases

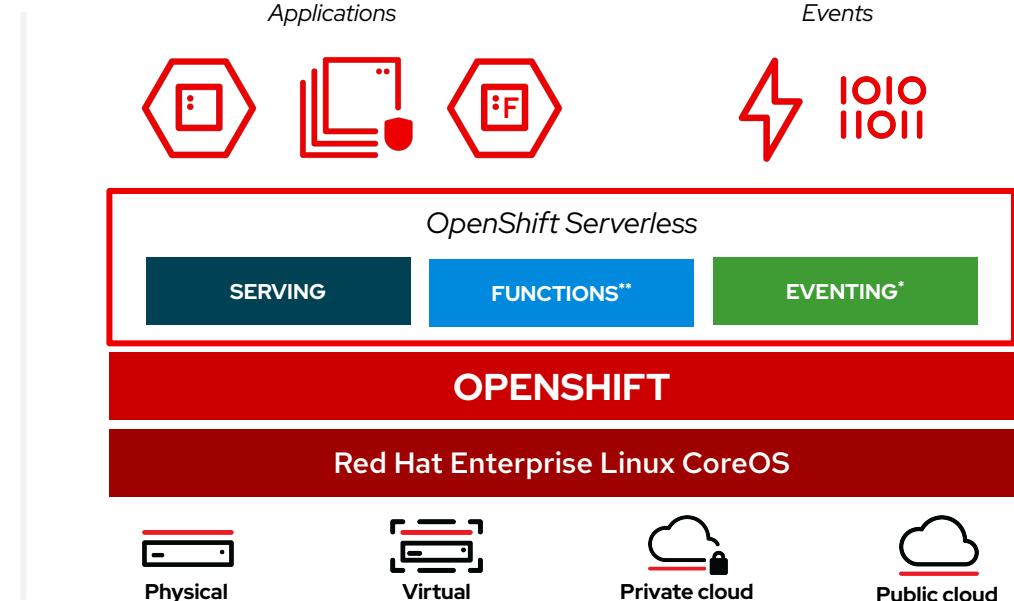
Serverless / knative



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



* Eventing is currently in Technology Preview

** Functions are currently a work in progress initiative

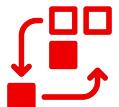
OpenShift Serverless

Key Features



Containers made easy

Simplified developer experience to deploy applications/code on serverless containers abstracting infrastructure & focusing on what matters.



Immutable revisions

Deploy new features: performing canary, A/B or blue-green testing with gradual traffic rollout with no sweat and following best practices.



Automatic scaling

No need to configure number of replicas, or idling. Scale to zero when not in use, auto scale to thousands during peak, with built-in reliability and fault-tolerance.



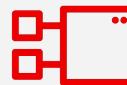
Ready for the Hybrid

Portable serverless running anywhere OpenShift runs, that is on-premises or on any public cloud. Leverage data locality and SaaS when needed.



Any programming language

Use any programming language or runtime of choice. From Java, Python, Go and JavaScript to Quarkus, SpringBoot or Node.js.



Event Driven

Architectures coupled & distributed apps connecting with a variety of built-in or third-party event sources or connectors powered by Operators.

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.

 OpenShift Serverless Operator

1.7.0 provided by Red Hat, Inc.

[Install](#)

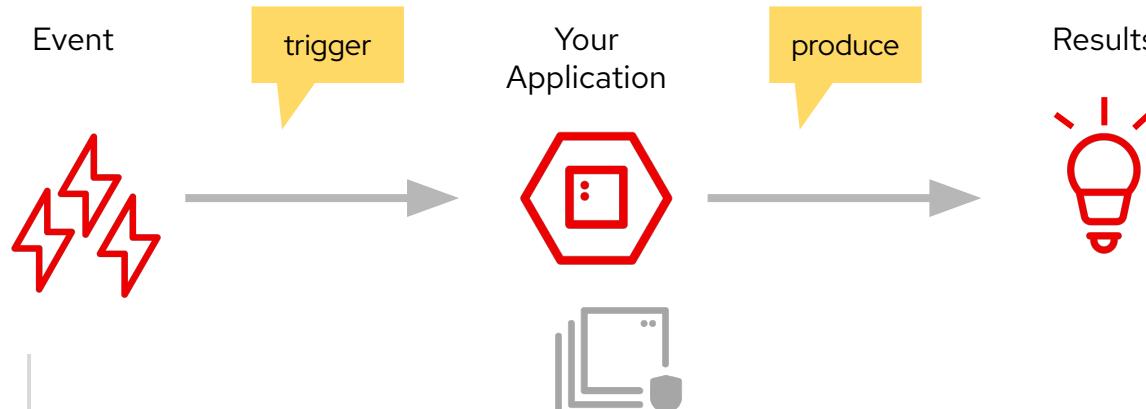
OPERATOR VERSION
1.7.0

PROVIDER TYPE
Red Hat

PROVIDER
Red Hat, Inc.

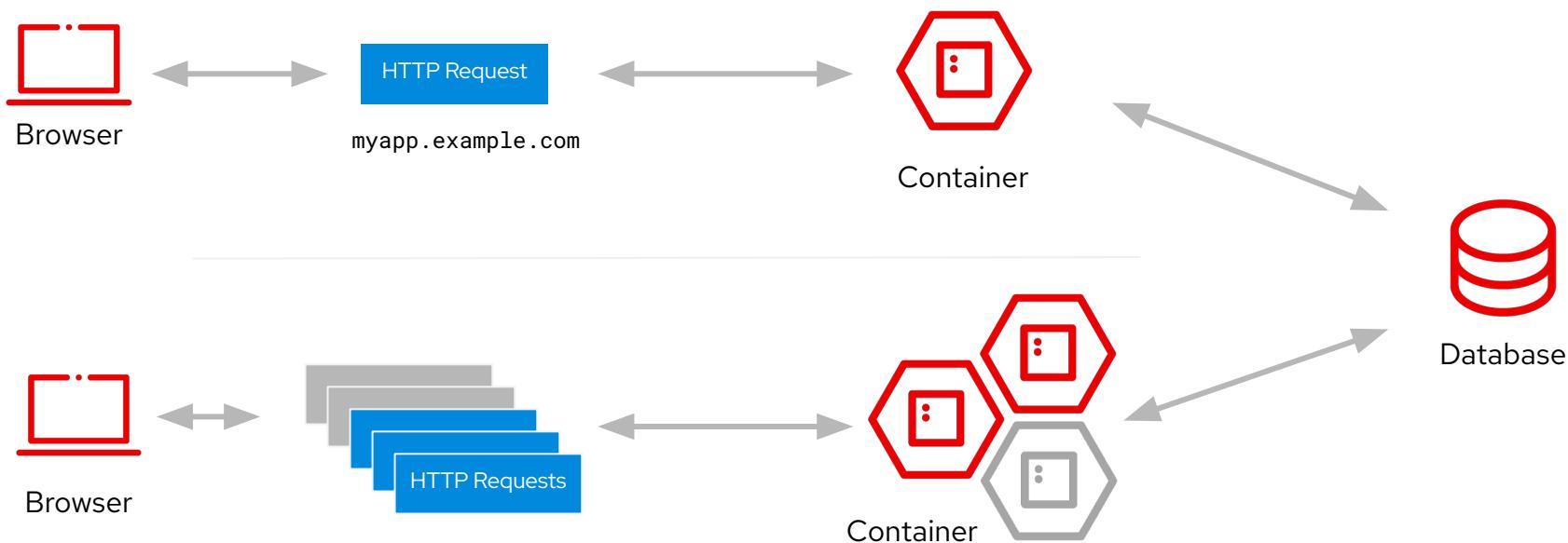
The Red Hat OpenShift Serverless operator provides a collection of APIs that enables containers, microservices and functions to run "serverless". Serverless applications can scale up and down (to zero) on demand and be triggered by a number of event sources. OpenShift Serverless integrates with a number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

The "Serverless Pattern"



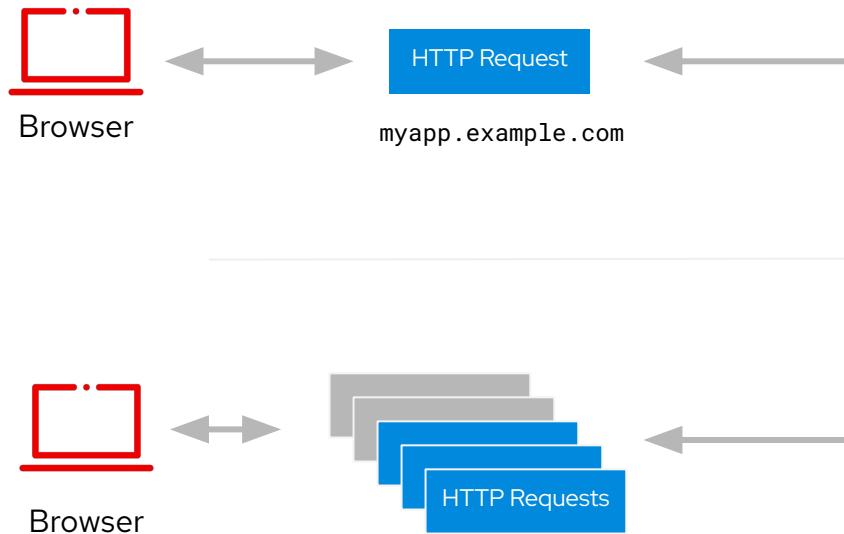
The "Serverless Pattern"

A serverless web application



The "Serverless Pattern"

A serverless web application

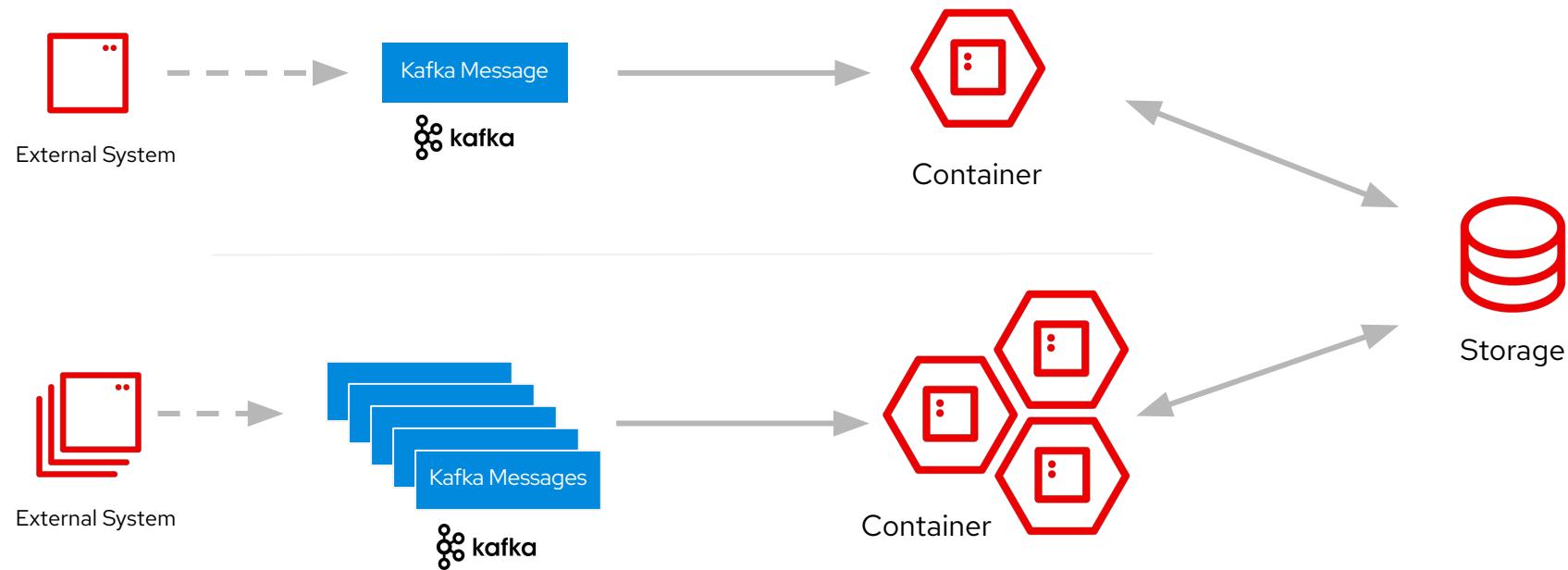


Benefits of this model:

- 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

The "Serverless Pattern"

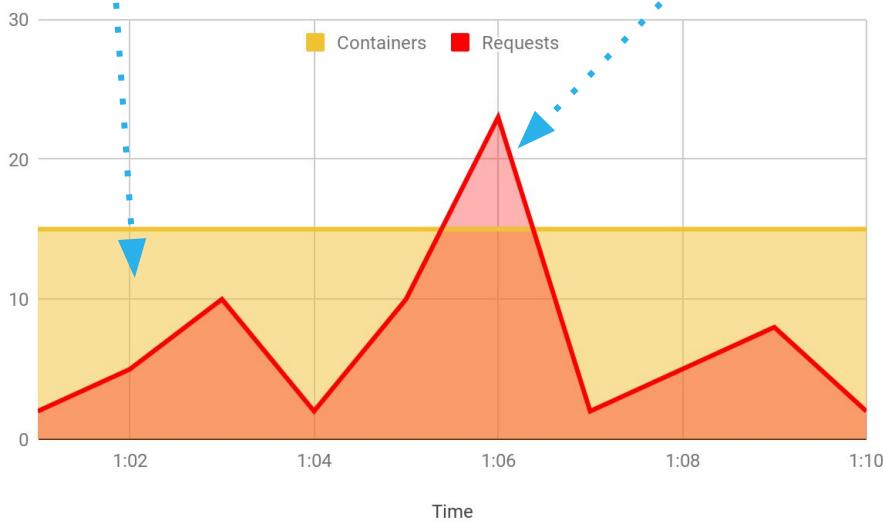
Processing a Kafka message



Serverless Operational Benefits

Over provisioning

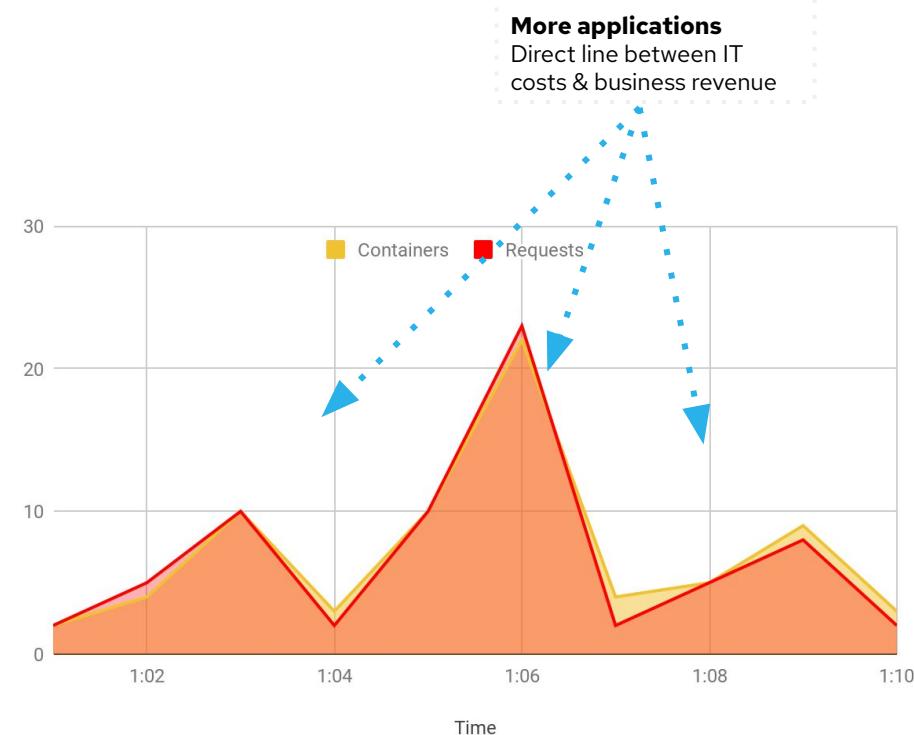
Time in capacity planning
IT cost of idle resources



NOT Serverless

Under provisioning

Lost business revenue
Poor quality of service



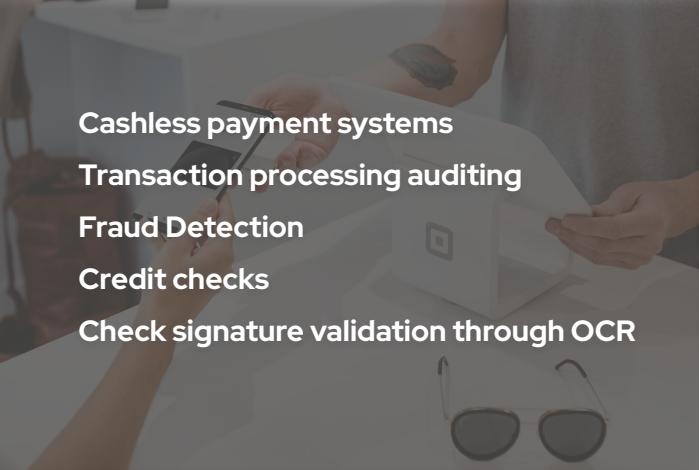
with Serverless

Knative-serving DEMO

Knative-serving A/B Deployments

Knative-serving Canary Deploy

Knative-serving Autoscaling



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

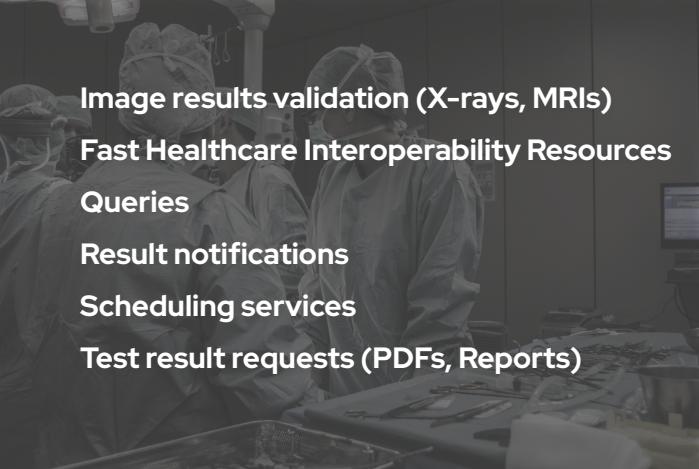
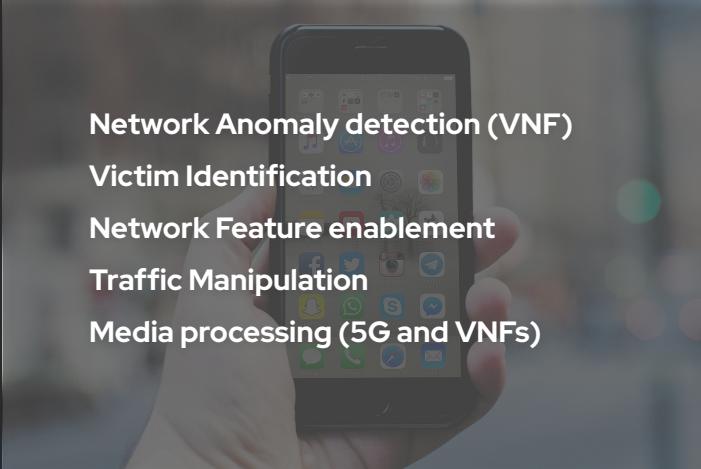


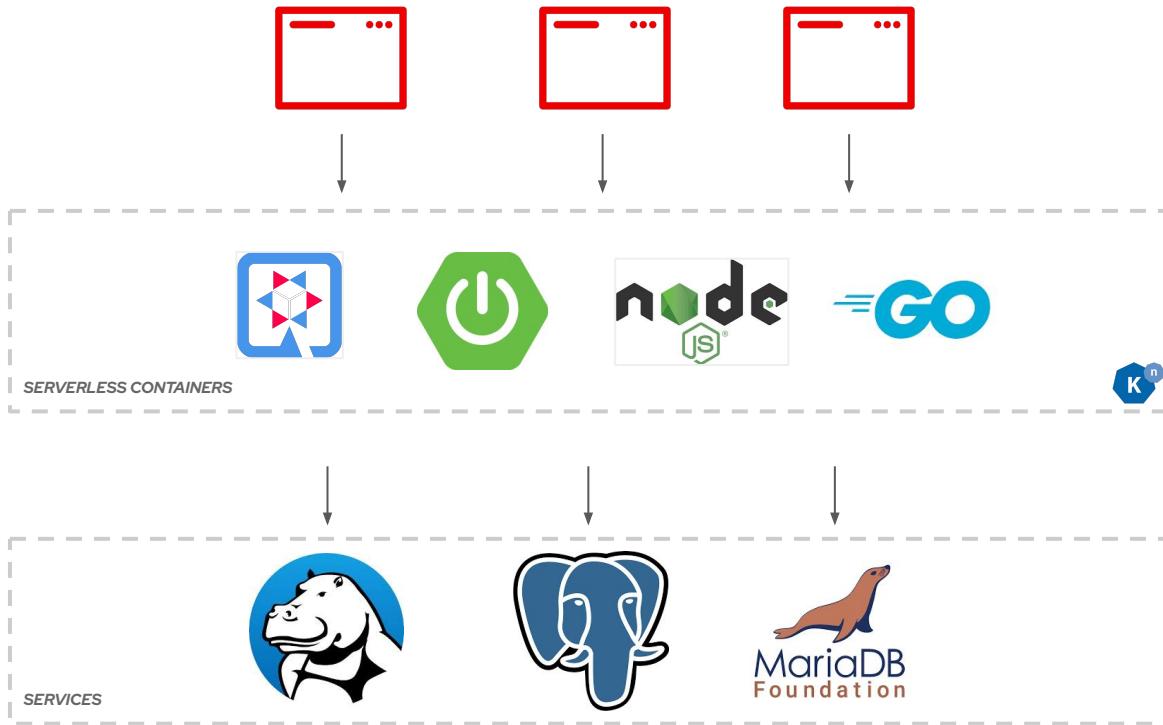
Image results validation (X-rays, MRIs)
Fast Healthcare Interoperability Resources
Queries
Result notifications
Scheduling services
Test result requests (PDFs, Reports)



Network Anomaly detection (VNF)
Victim Identification
Network Feature enablement
Traffic Manipulation
Media processing (5G and VNFs)



Web Applications and APIs



Language or runtime of your choice:

OpenJDK



python™



django



VERT.X

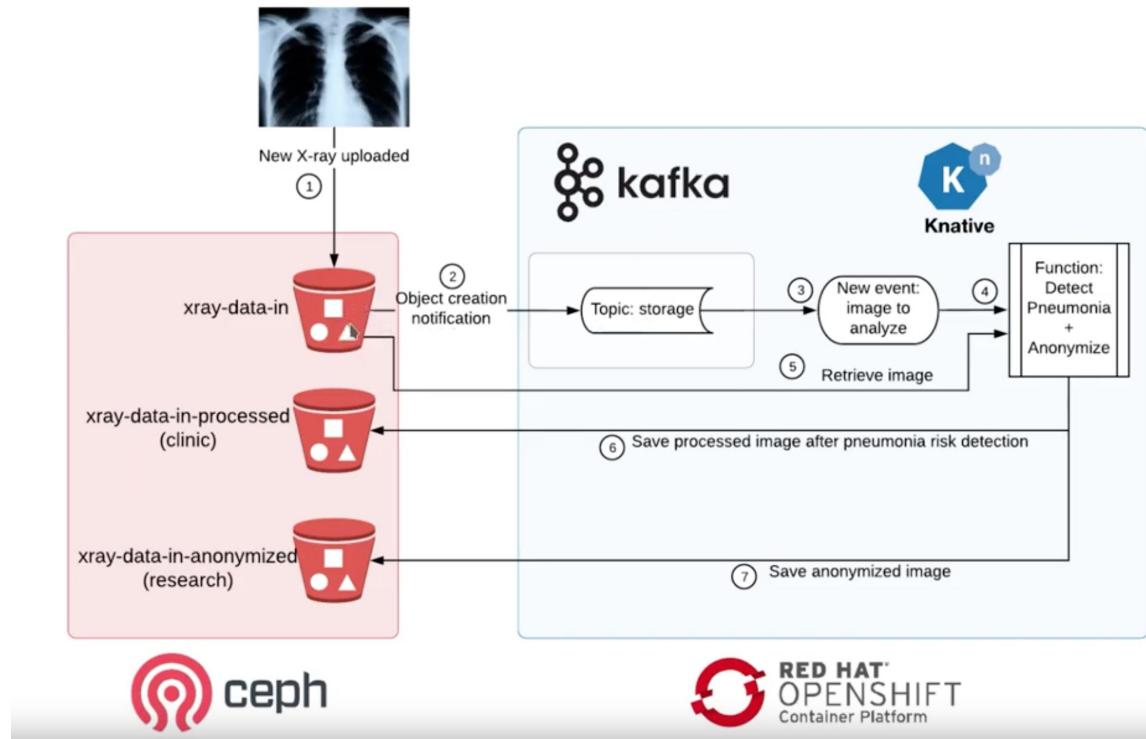




Data Transformation

Common Use cases

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



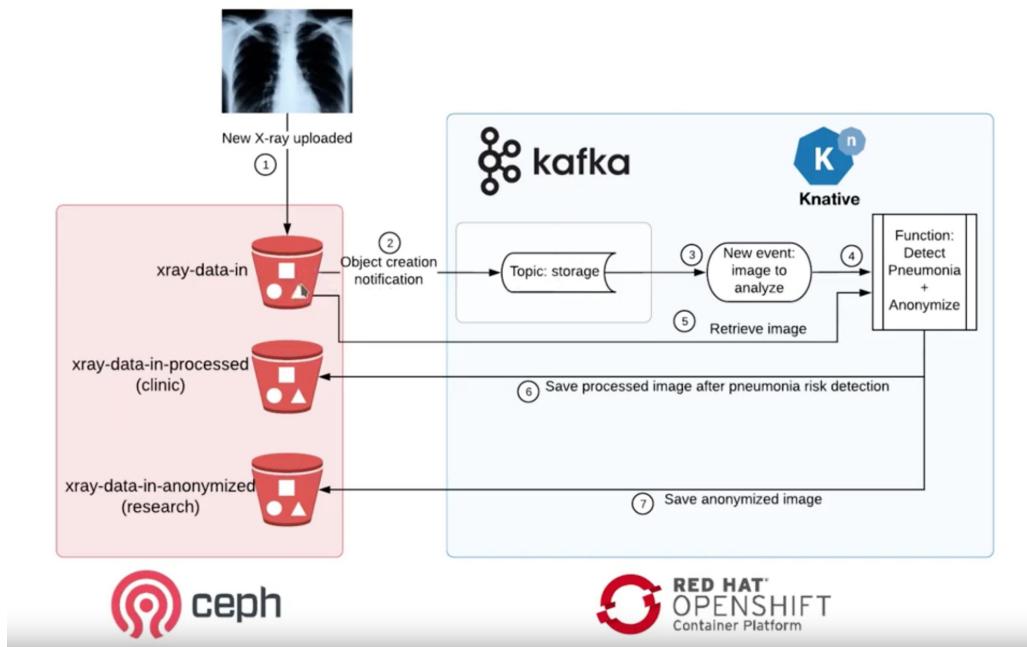


Ceph + Kafka + Serverless

"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





Building on OpenShift Serverless with Red Hat Services

Connected Services

How Knative services interact with the outside world.



Service Orchestrator

Composing multiple services together into an application.



Event Streaming

All modern architectures need some Kafka.



API Gateway

Next gen APIs still require management.



Implementing Services

Functions, languages, and the vagaries of cold starts.



The Dirty Word in Serverless

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

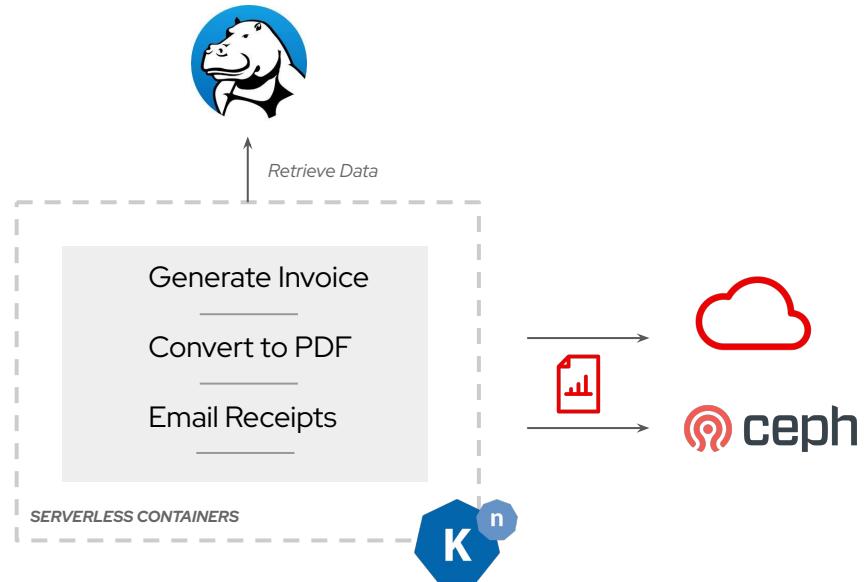




Scheduled Apps Cron Jobs



- Every Hour →
- Every Day →
- Every Week →
- Every Month →





When to use Serverless

- Does your application have an unpredictable or bursty number of requests ?
- Are you trying to build event-driven, loosely coupled systems ?
- (Do you think Kubernetes is hard for developers to get started ?)
- Do you want to perform A/B testing or canary deployments for your applications ?
- Do you have workloads that are seasonal or run on a specific schedule or periodically ?
- Are you building microservices or containers and want to leverage serverless ?

Optional section marker or title

26



Thank you



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



twitter.com/RedHat