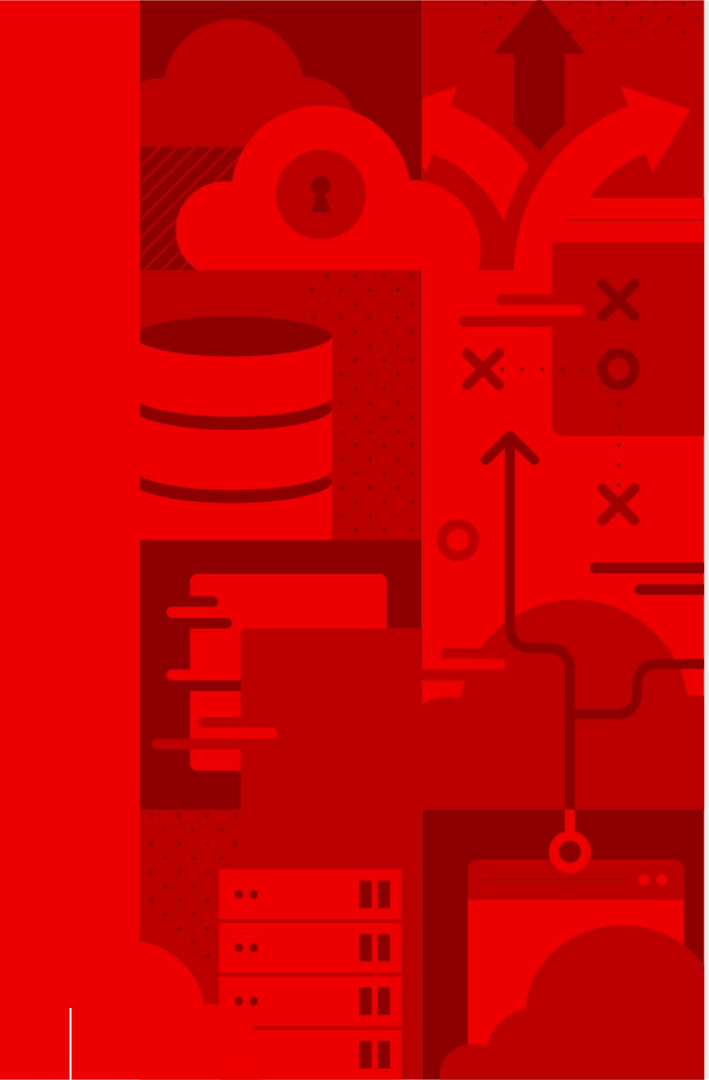


OpenShift Serverless

Wanja Pernath

EMEA Partner Enablement, OpenShift Dev
wanja@redhat.com

Demo Repository:
https://github.com/wpernath/knative-workshop

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat

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

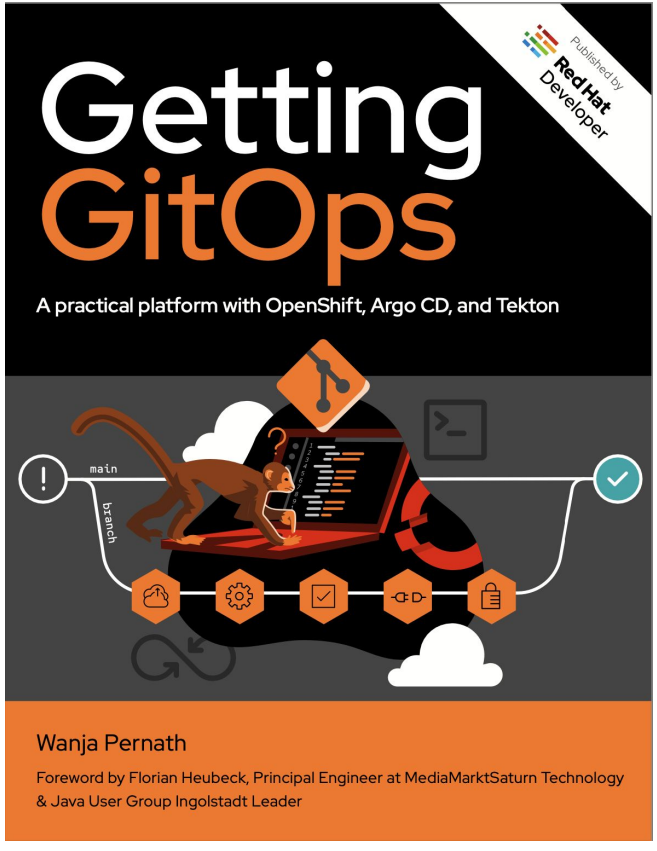
Getting GitOps

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

The demo repository:
https://github.com/wpernath/knative-workshop

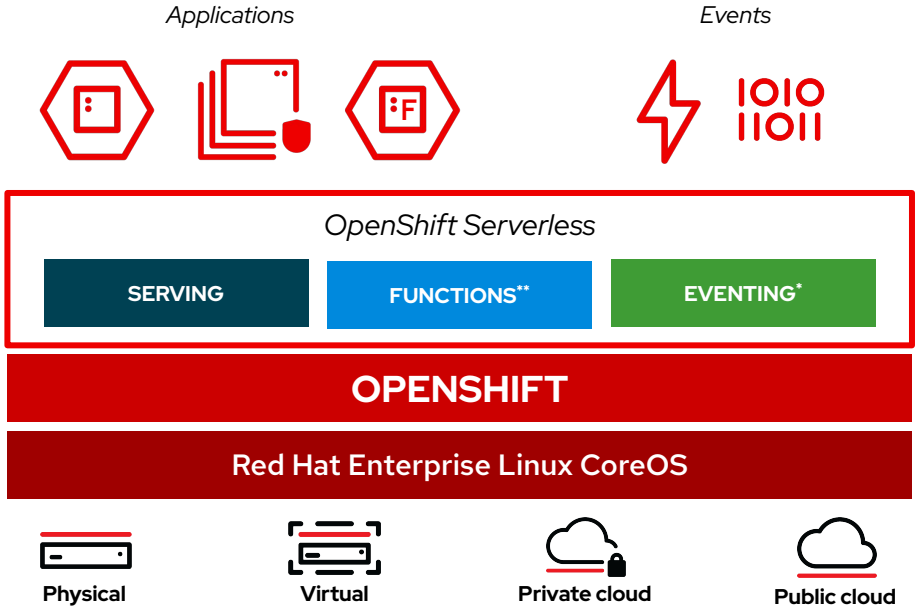
Serverless / knative

Generally Available

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

Red Hat

OpenShift Serverless

Key Features

Containers made easy

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

Ready for the Hybrid Cloud

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

Immutable revisions

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

Any programming language

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

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.

Event Driven Architectures

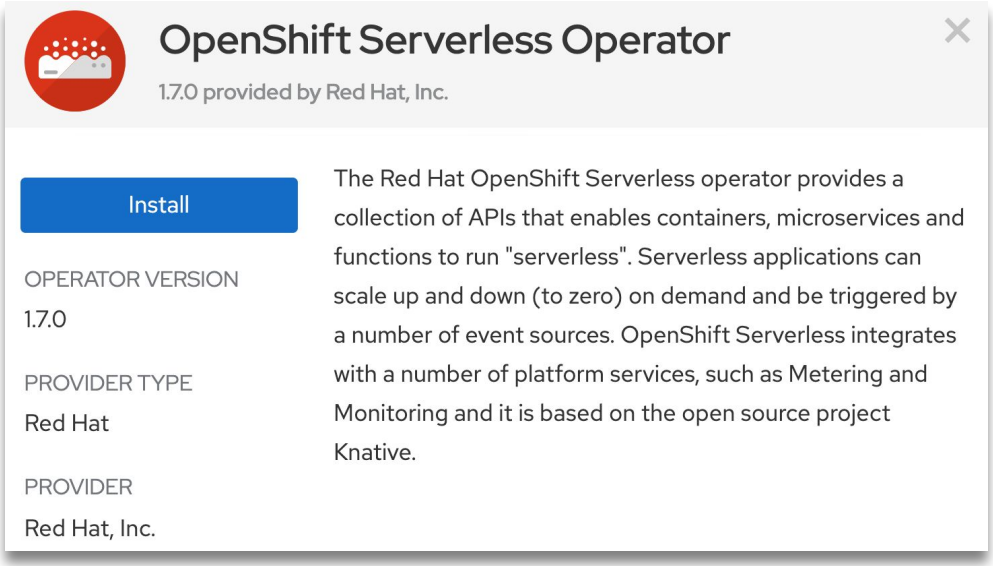
Build loosely coupled & distributed apps connecting with a variety of built-in or third-party event sources or connectors powered by Operators.

Red Hat

Installation experience

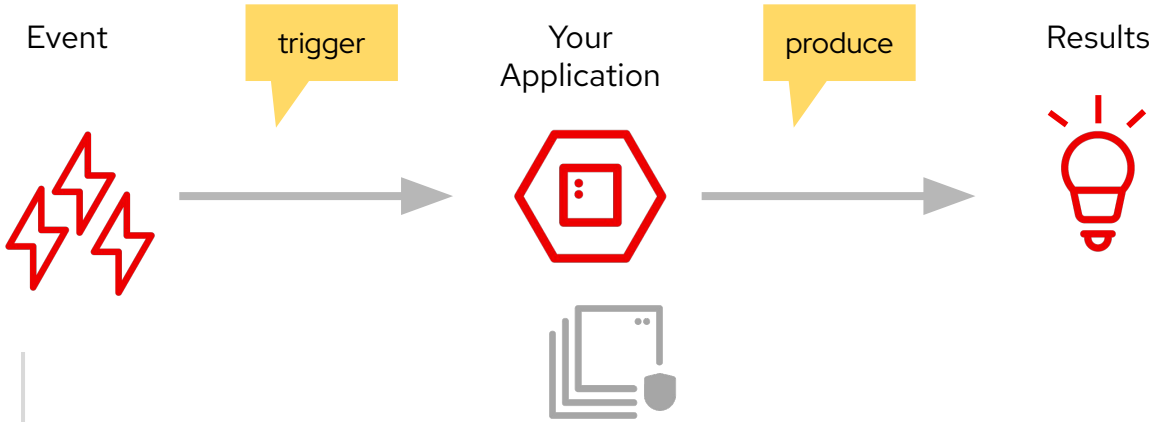
"Easy day 1 and even better for day 2"

- Click Install experience
- Developer & admin experience in Console
- Built-in event sources
- No external dependencies.



Red Hat

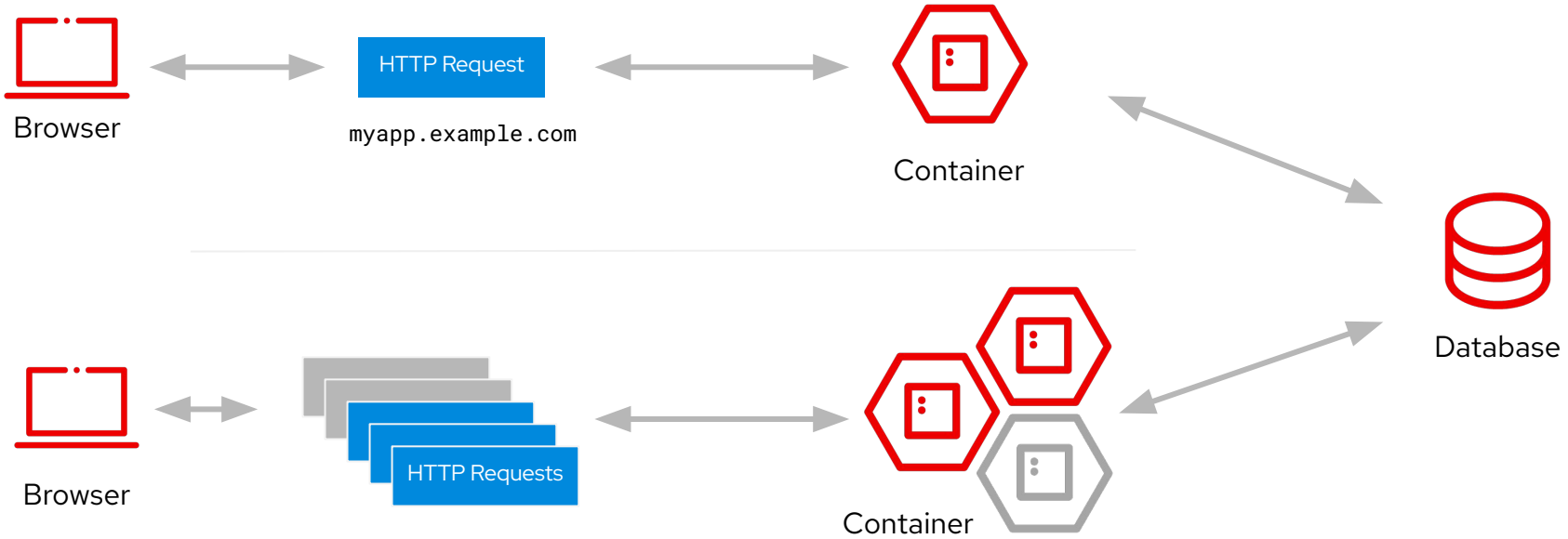
The "Serverless Pattern"



- HTTP Requests
- Kafka Messages
- Image Uploaded
- New Order
- Login from user

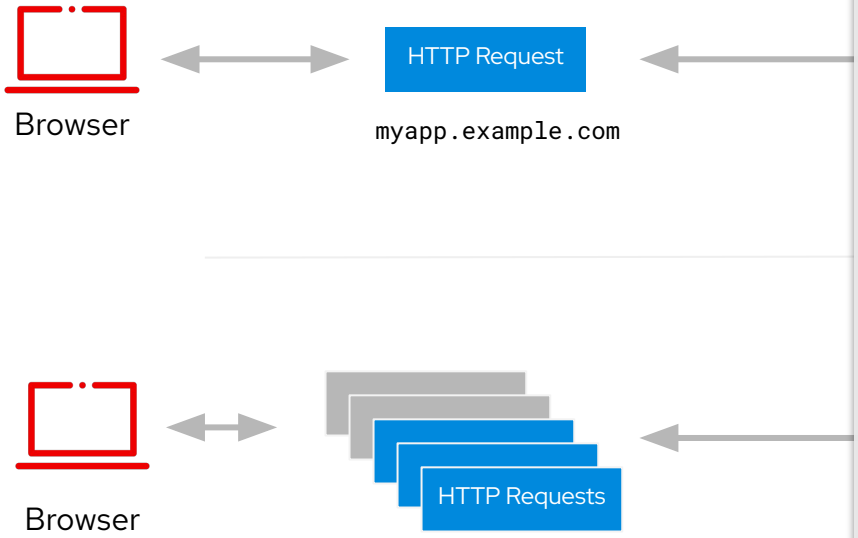
The "Serverless Pattern"

A serverless web application



The "Serverless Pattern"

A serverless web application

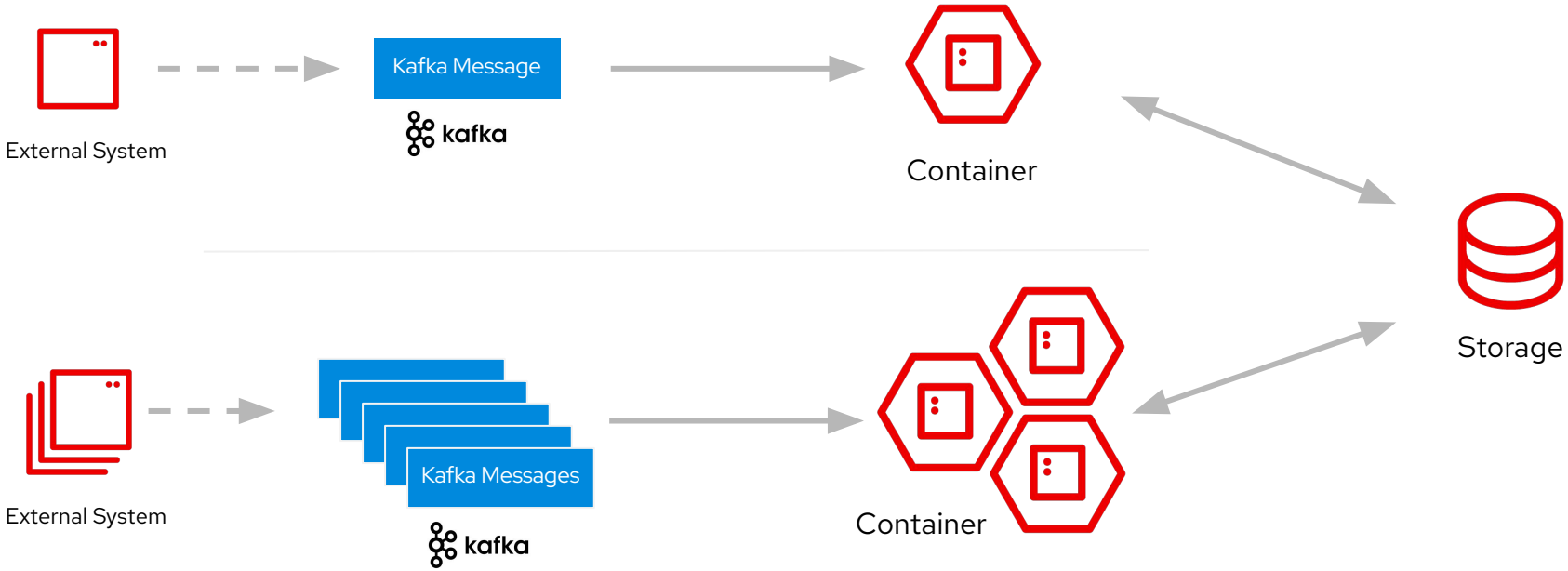


Benefits of this model:

- 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"

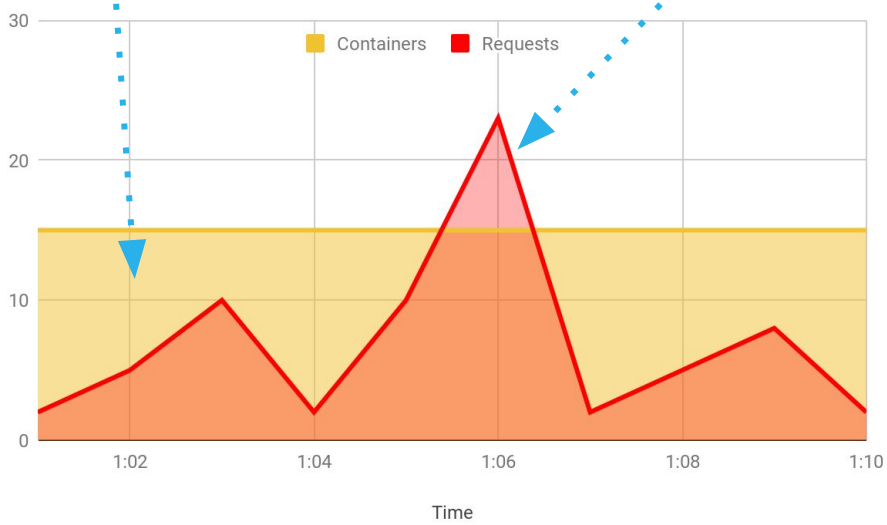
Processing a Kafka message



Serverless Operational Benefits

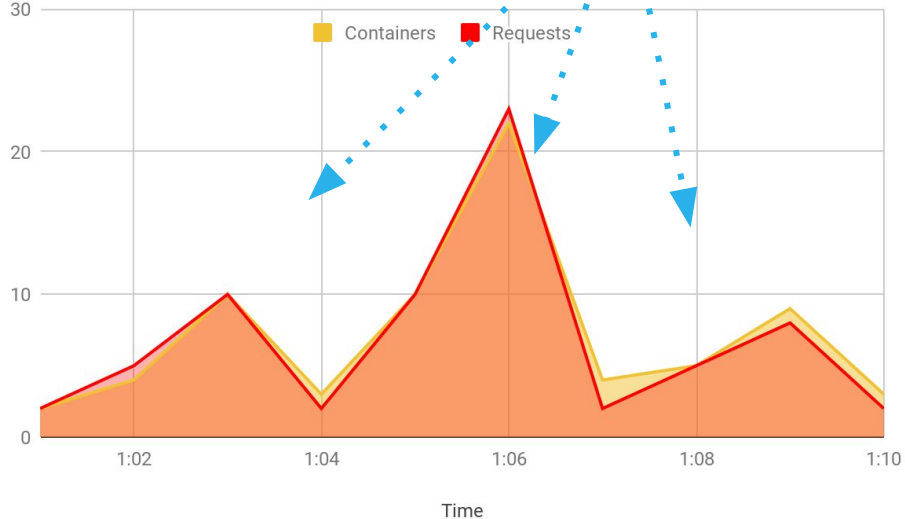
Over provisioning
Time in capacity planning
IT cost of idle resources

Under provisioning
Lost business revenue
Poor quality of service



NOT Serverless

More applications
Direct line between IT
costs & business revenue



with Serverless

Knative-serving DEMO

https://github.com/wpernath/knative-workshop

Knative-serving A/B Deployments

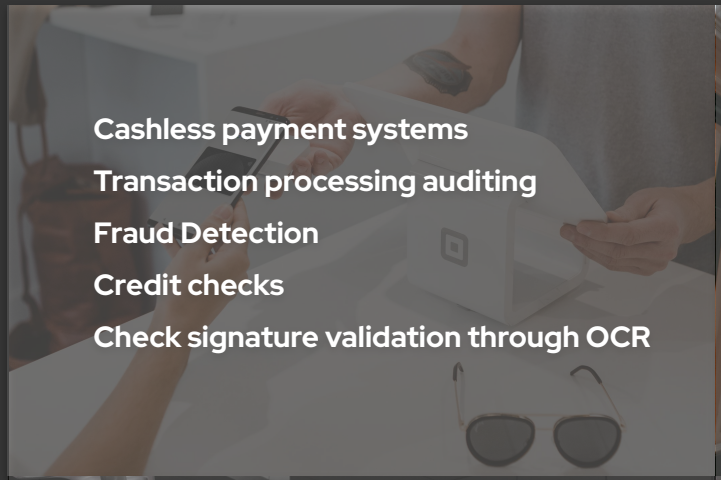
https://github.com/wpernath/knative-workshop

Knative-serving Canary Deploy

https://github.com/wpernath/knative-workshop

Knative-serving Autoscaling

https://github.com/wpernath/knative-workshop



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

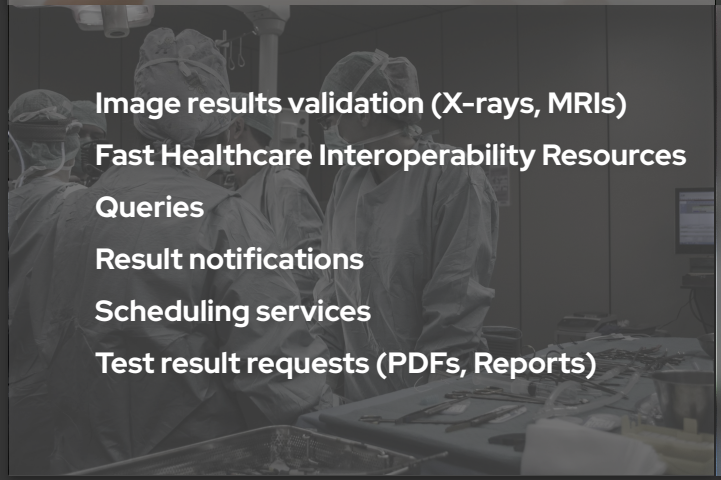
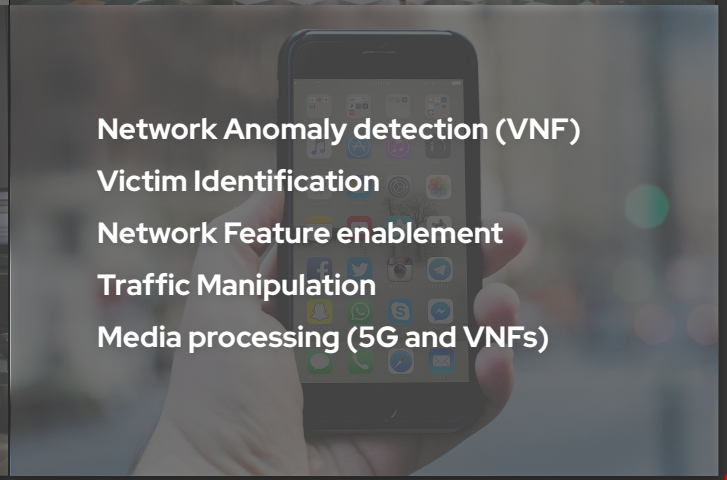
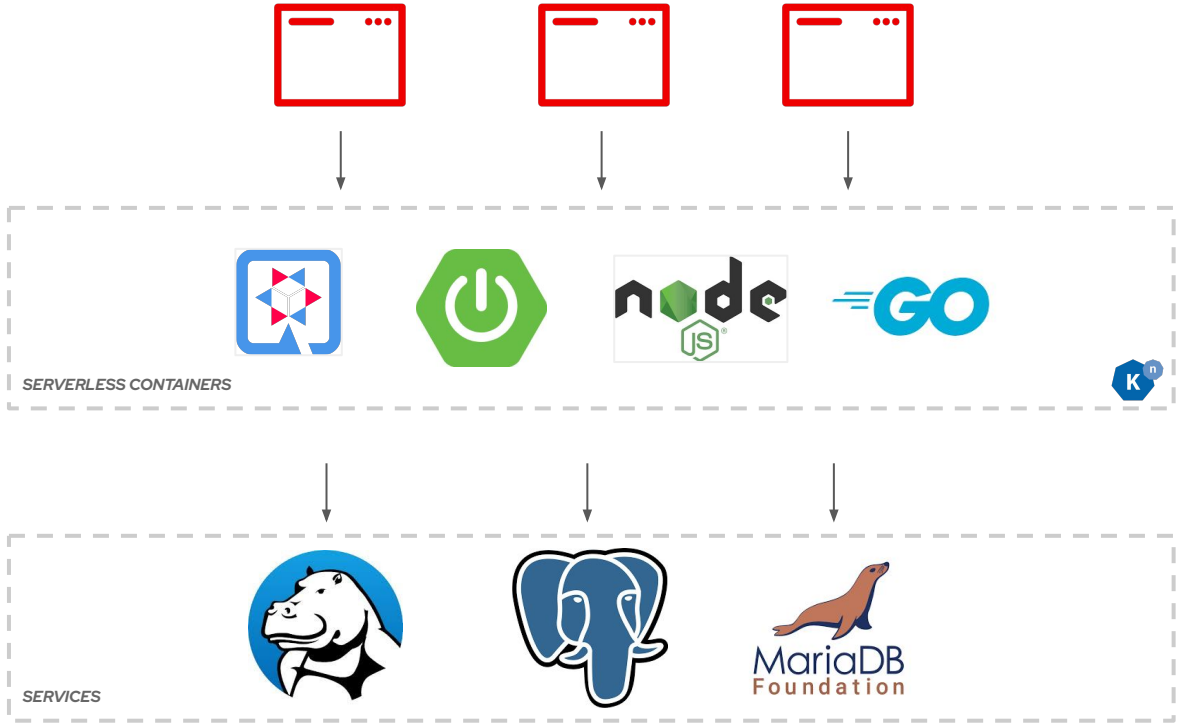


Image results validation (X-rays, MRIs)
Fast Healthcare Interoperability Resources
Queries
Result notifications
Scheduling services
Test result requests (PDFs, Reports)



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:

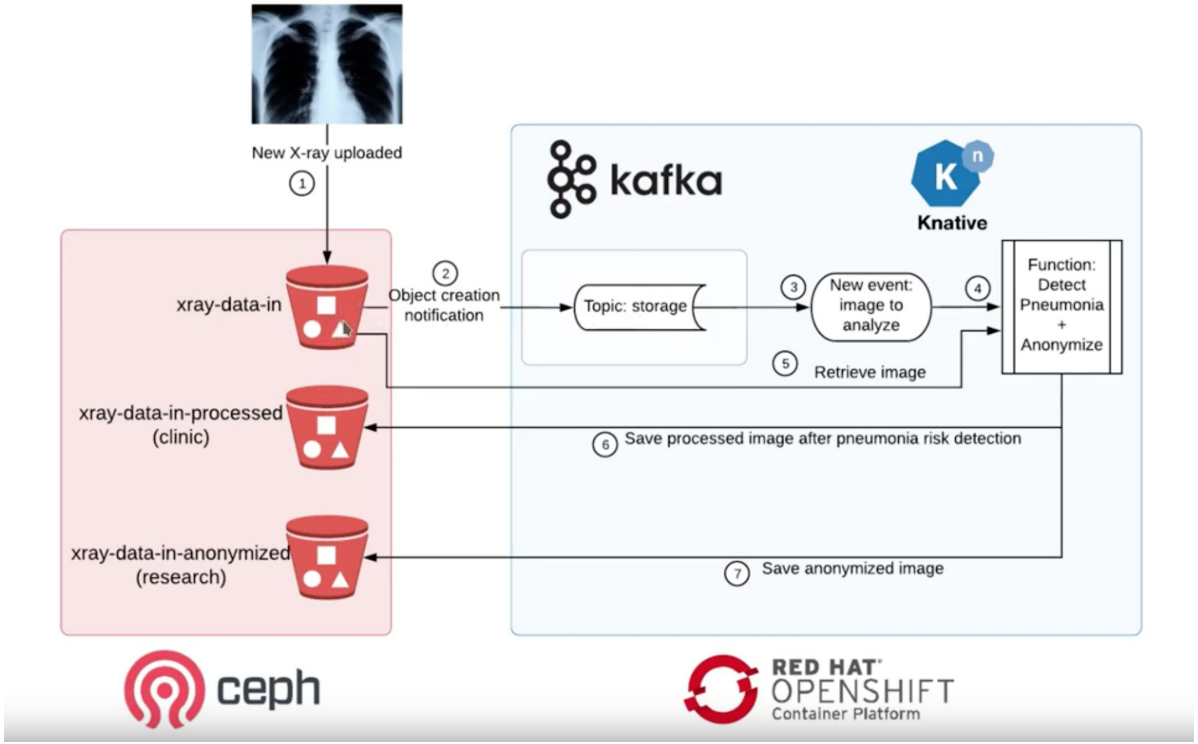
OpenJDK
python
django
VERT.X
JS
RAILS

Red Hat

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



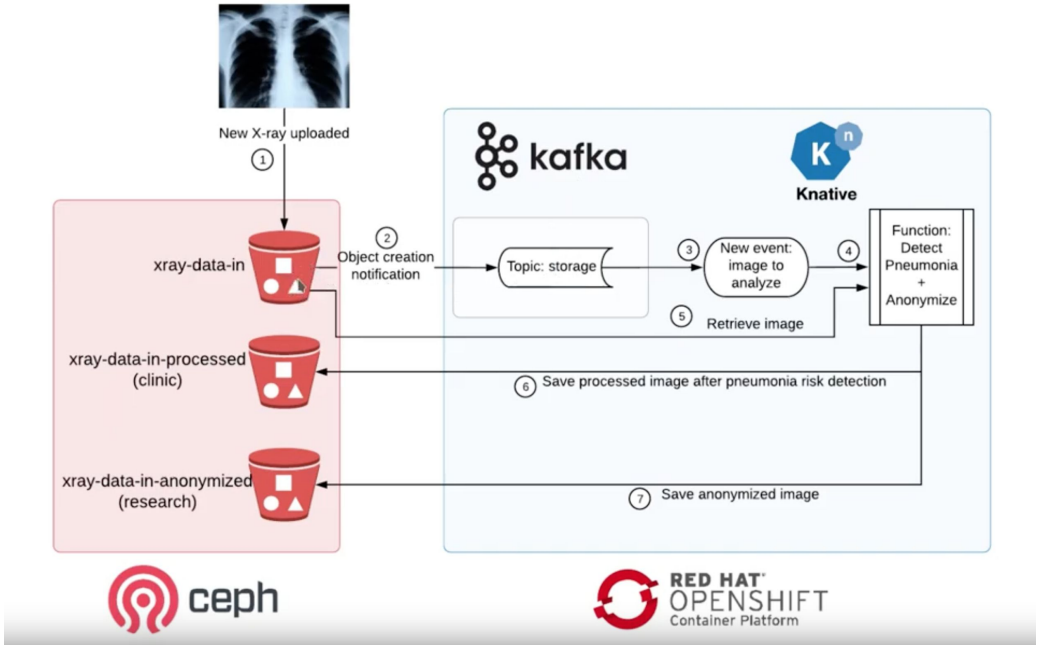
* Article: https://medium.com/analytics-vidhya/automated-data-pipeline-using-ceph-notifications-and-kserving-5e1e9b996661

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



TensorFlow

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

Red Hat

Building on OpenShift Serverless with Red Hat Services

Connected Services
How Knative services interact with the outside world.

API Gateway
Next gen APIs still require management.

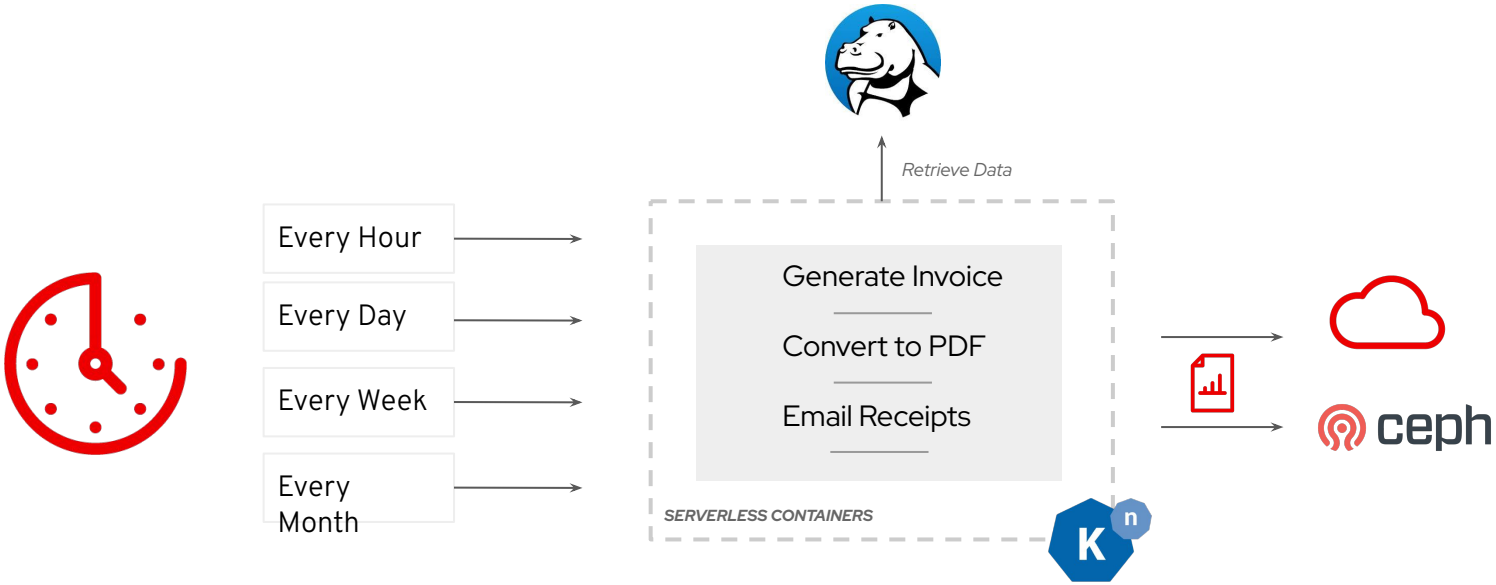
Service Orchestrator
Composing multiple services together into an application.

Implementing Services
Functions, languages, and the vagaries of cold starts.

Event Streaming
All modern architectures need some Kafka.

The Dirty Word in Serverless
Yep, you still need state to handle long-lived orchestration.

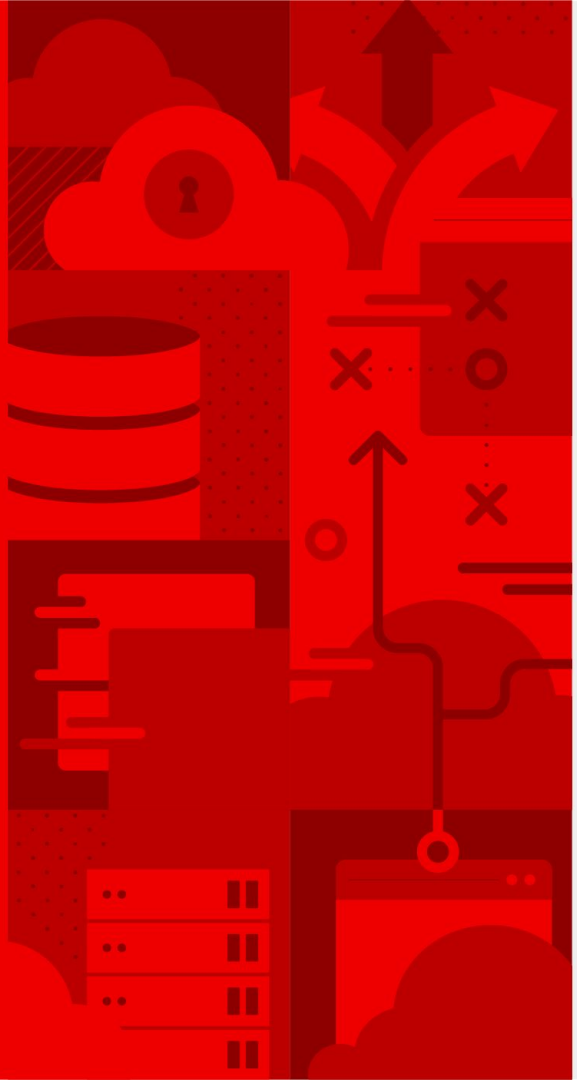
Scheduled Apps Cron Jobs



Red Hat

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 ?



Thank you

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat