

State of Serverless & Service Mesh

William Markito Oliveira
Product Manager
`@william_markito`

Giuseppe Bonocore
Solution Architect

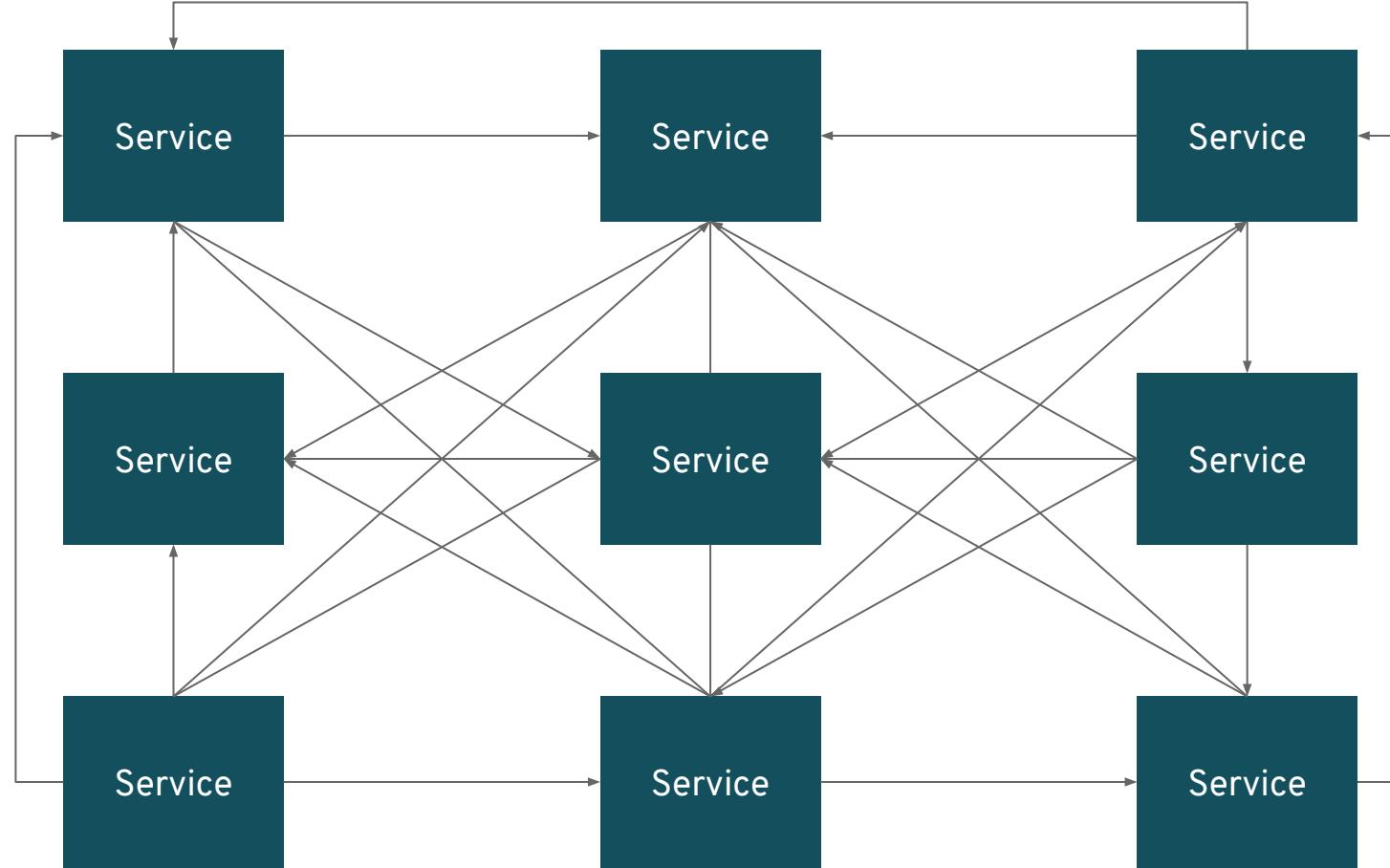
Sep 2019

Sep 2019, Milano

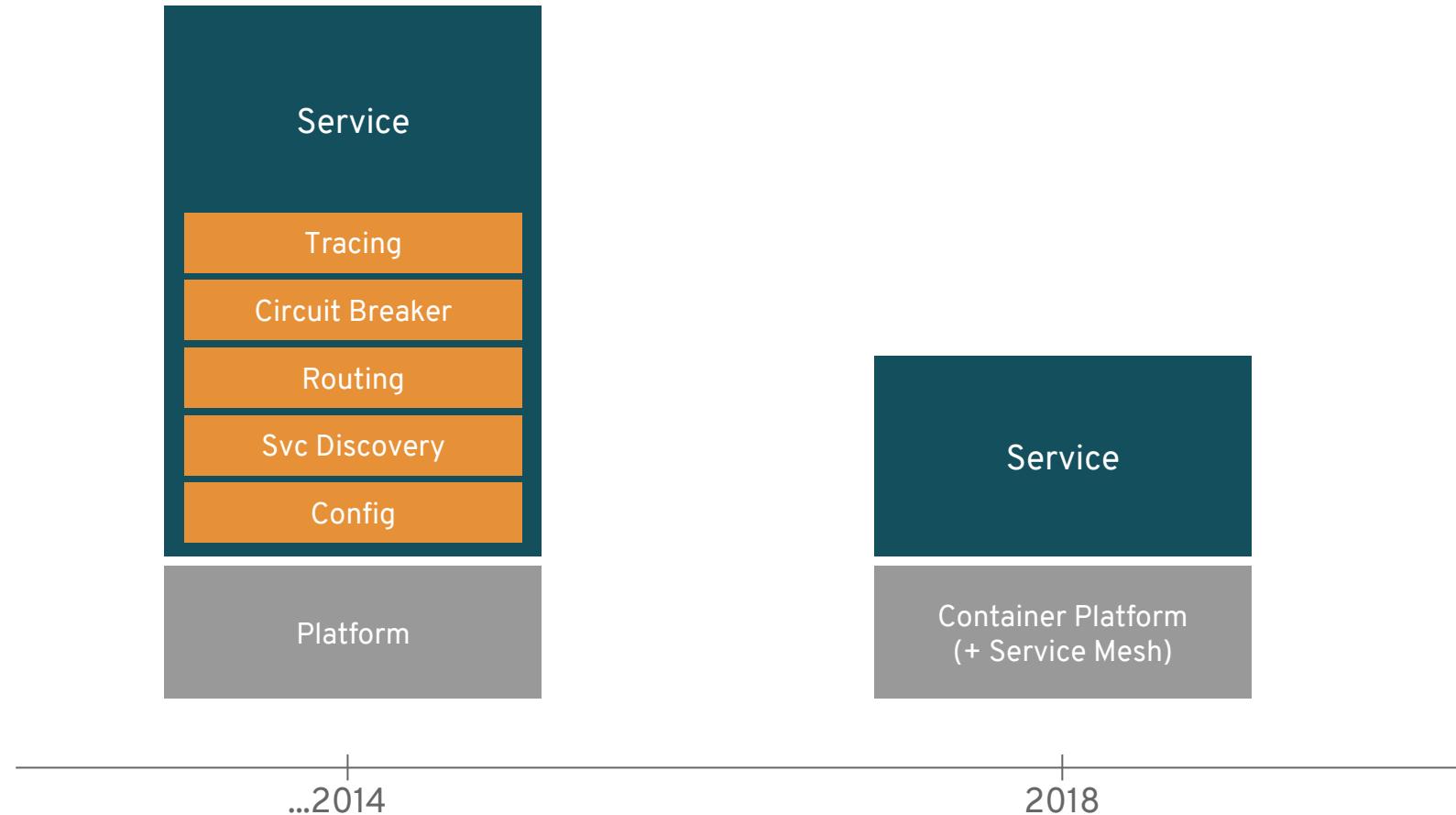


Service Mesh

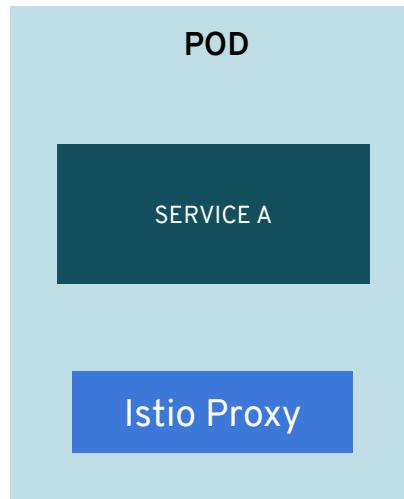
Distributed Architecture



Microservices Evolution



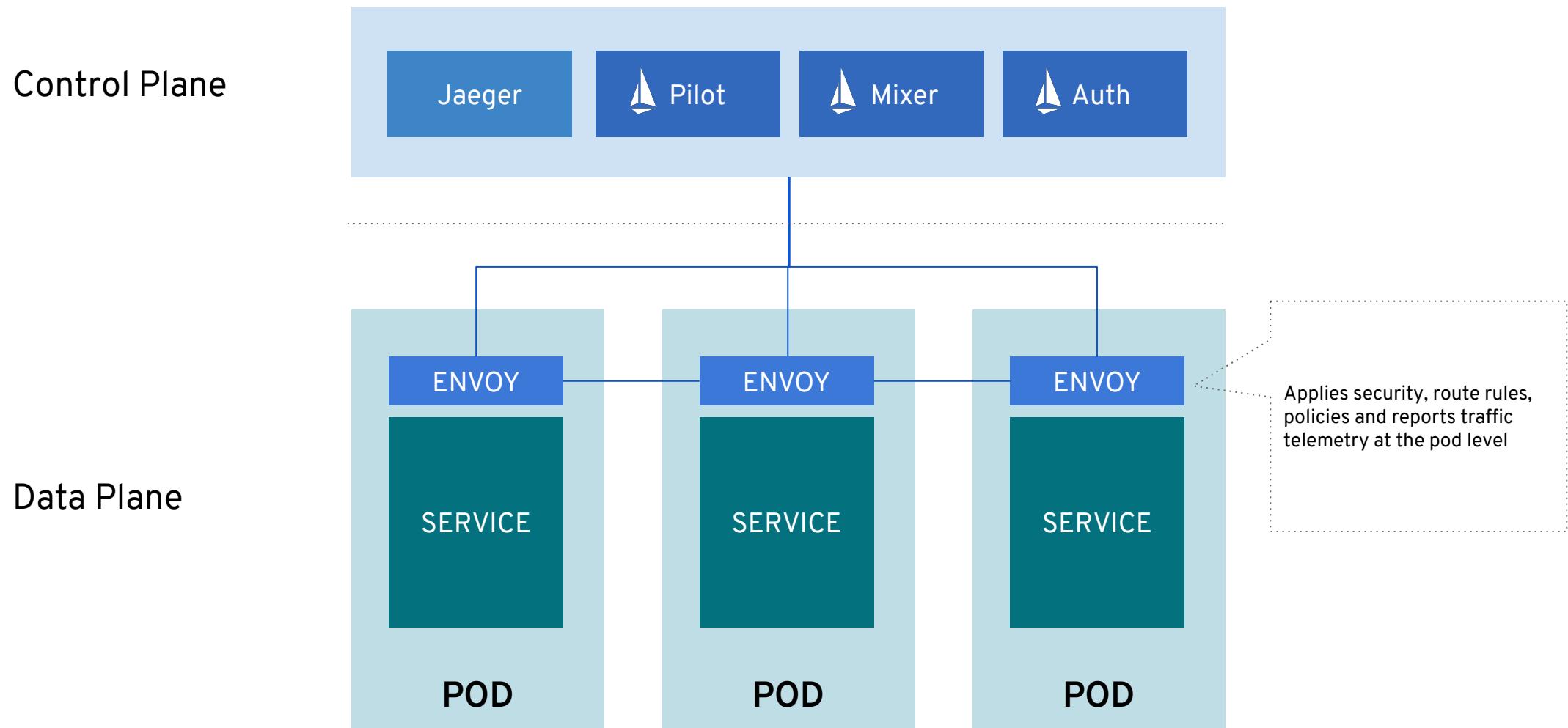
Sidecars



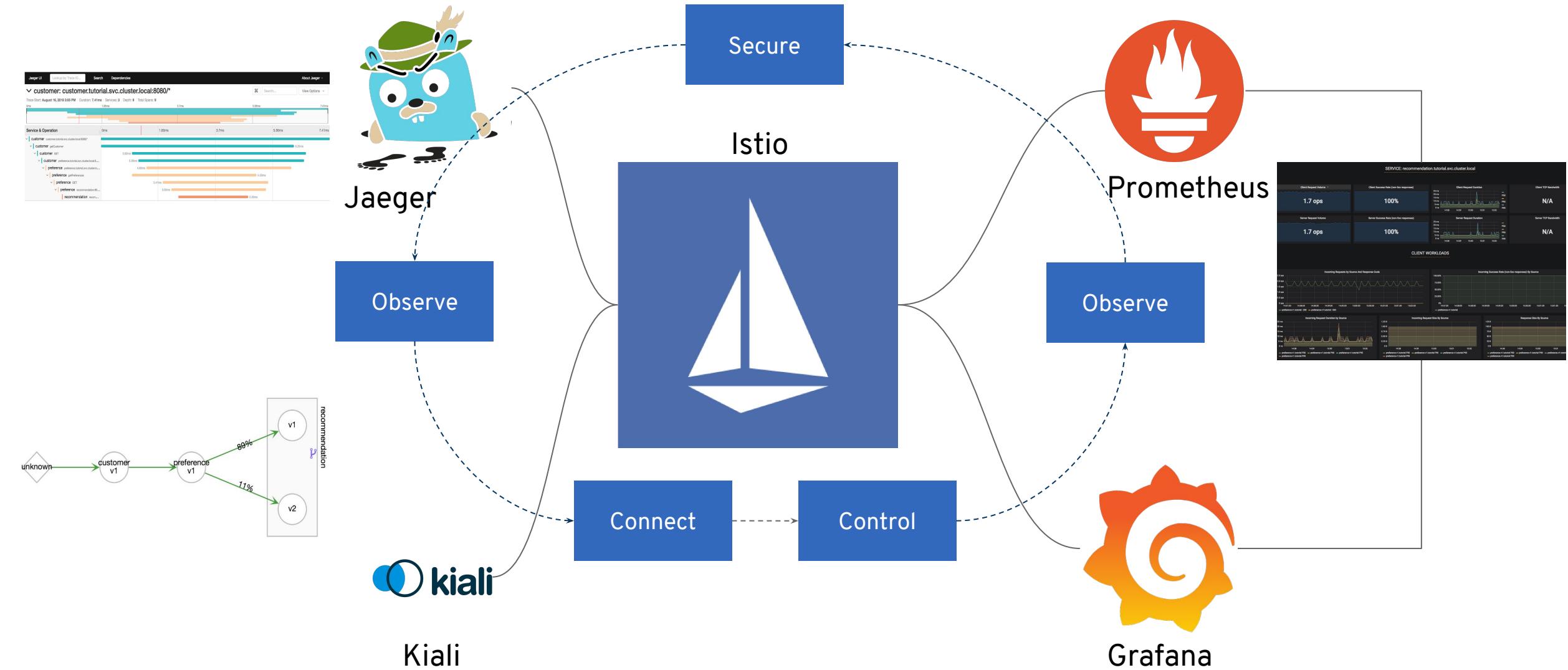
- Two or more containers deployed to same pod
- Share
 - Same
 - Namespace
 - Pod IP
 - Shared lifecycle
- Used to enhance the co-located containers
- Istio Proxy (L7 Proxy)
 - Proxy all network traffic in and out of the app container

Source: <http://blog.kubernetes.io/2015/06/the-distributed-system-toolkit-patterns.html>

Service Mesh Architecture



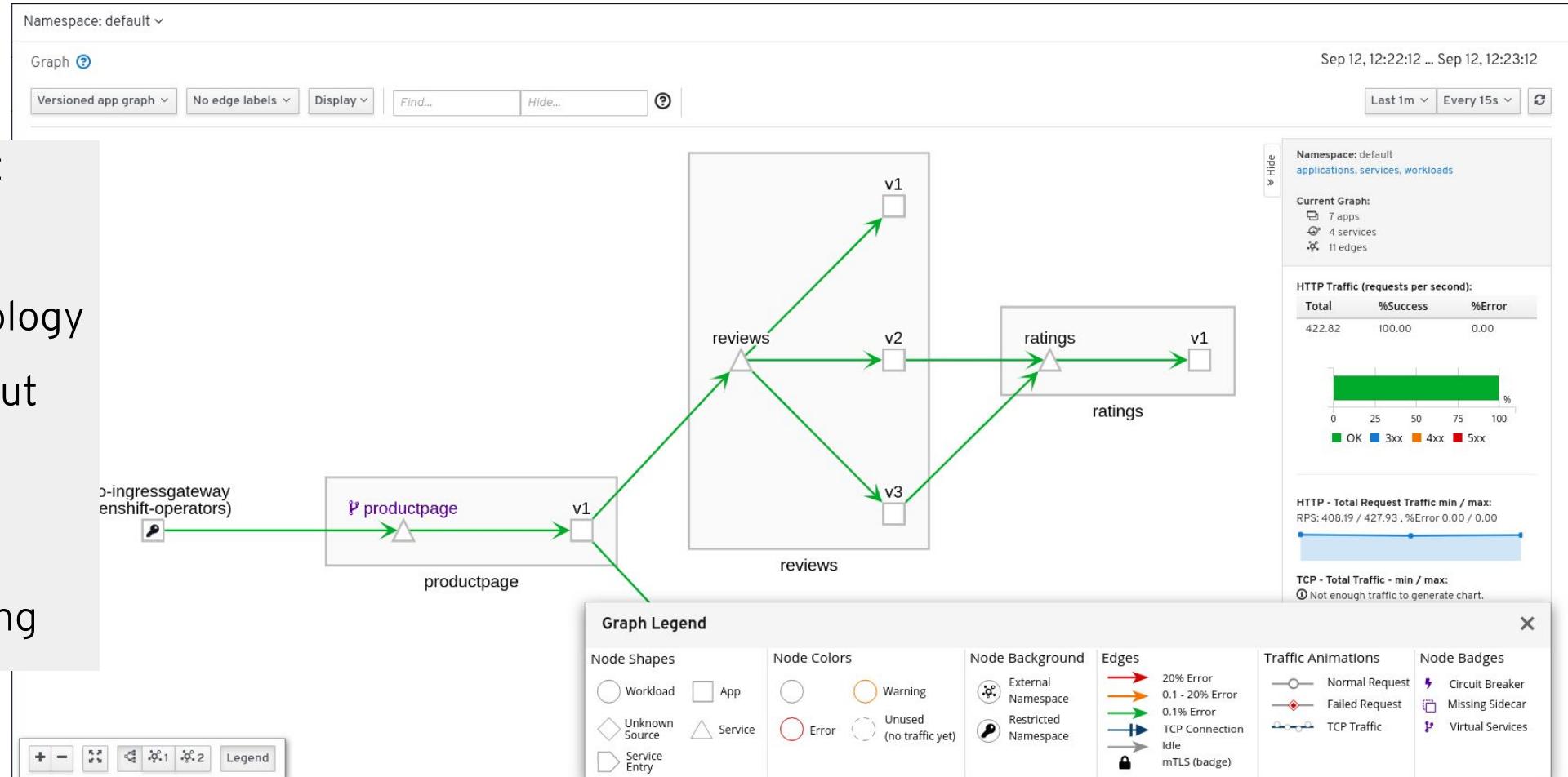
Service Mesh Ecosystem



Enhanced Visualization of Cluster Traffic With Kiali

Visualization of what
Matters most:

- Application Topology
- Traffic throughput
- Error Rates
- Service Latency
- Service Versioning



Convenient Overviews of Individual Services

Services > Namespace: default > Service: reviews

reviews (Show on graph)

Overview Traffic Inbound Metrics

Last 1m ▾ Actions ▾

Labels	Ports	Endpoints	Health
app reviews service reviews	TCP http (9080)	10.128.2.27 : reviews-v1-989d5ffdf-w8gmn 10.128.2.28 : reviews-v3-757c4f7849-rs7sw 10.131.0.35 : reviews-v2-6ff8648d69-tlqhn	Healthy
Selectors			Error Rate over last 1m: 0.00%
app reviews			
Type ClusterIP IP 172.30.17.223 Created at 9/12/2019, 12:06:32 PM Resource Version 5094640			

Workloads (3) Virtual Services (0) Destination Rules (0)

Name	Type	Labels	Created at	Resource version
reviews-v1	Deployment	app reviews version v1	9/12/2019, 12:06:32 PM	5095051
reviews-v2	Deployment	app reviews version v2	9/12/2019, 12:06:32 PM	5094998
reviews-v3	Deployment	app reviews version v3	9/12/2019, 12:06:32 PM	5095043

Guided Configuration of Traffic Policies

Create Weighted Routing X

WORKLOAD	TRAFFIC WEIGHT
 reviews-v1	<input type="range" value="5"/> 5
 reviews-v2	<input type="range" value="80"/> 80
 reviews-v3	<input type="range" value="15"/> 15

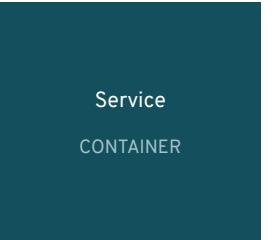
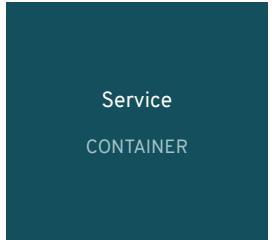
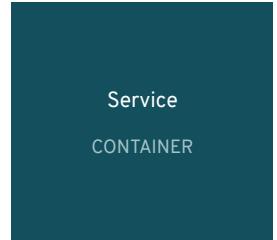
Evenly distribute traffic

▼ Hide Advanced Options

VirtualService Hosts	<input type="text" value="reviews"/>
The destination hosts to which traffic is being sent. Enter one or multiple hosts separated by comma.	
 TLS	<input type="button" value="DISABLE ▾"/>
Add LoadBalancer	<input type="button" value="OFF"/>
Add Gateway	<input type="button" value="OFF"/>

Distributed Services With Red Hat OpenShift Service Mesh

SERVICE



ANY
APPLICATION

SERVICE
OPS

OpenShift Service Mesh
(Istio + Jaeger + Kiali)

INFRA OPS

OpenShift Container Platform
(Enterprise Kubernetes)

INFRA



Laptop



Datacenter



OpenStack



Amazon Web Services



Microsoft Azure



Google Cloud

ANY
INFRASTRUCTURE



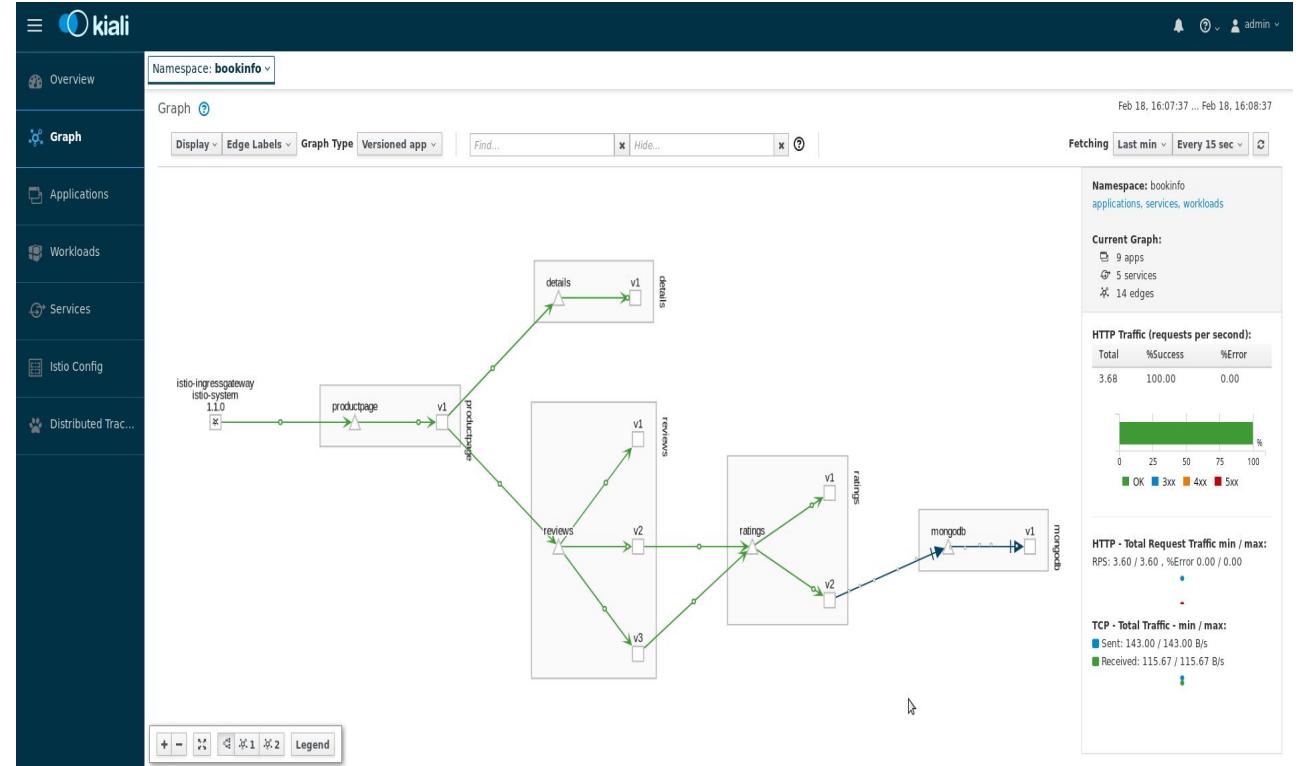
OpenShift Service Mesh

Sep 2019, Milano

Generally Available

Key Features

- A dedicated network for service to service communications
- Observability and distributed tracing
- Policy-driven security
- Routing rules & chaos engineering
- Powerful visualization & monitoring
- Will be available via OperatorHub



Serverless

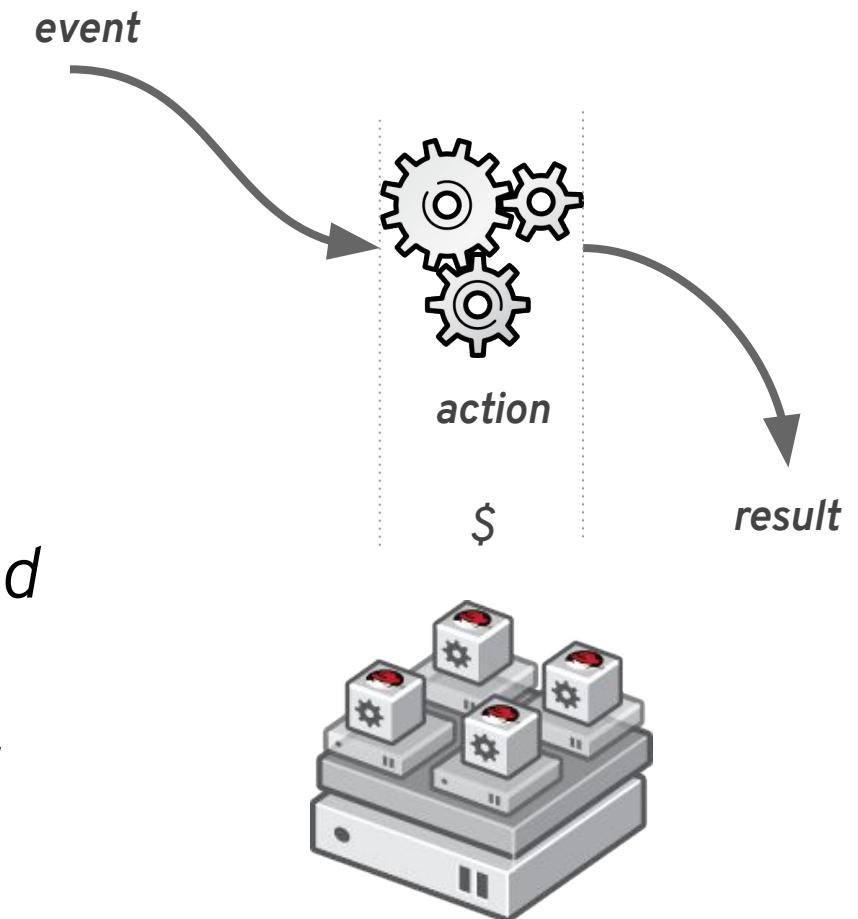
Wait... wat ?

<https://github.com/ghuntley/serverless-to-cgi-bin>



Serverless Defined

“computing execution model that depends on services to manage server-side logic and state where business logic run in stateless, event-triggered compute linux containers”



Serverless > Functions

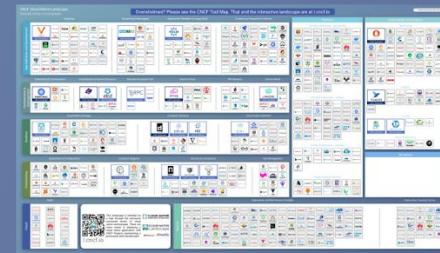
Serverless > Microservices

Serverless > Containers

"Function as a Service (FaaS) is serverless in the same way a square is a rectangle"



Serverless computing refers to a new model of cloud native computing, enabled by architectures that do not require server management to build and run applications. This landscape illustrates a finer-grained deployment model where applications, bundled as one or more functions, are uploaded to a platform and then executed, scaled, and billed in response to the exact demand needed at the moment



Application Concerns

Routing & transformation
Technology Adapters
Error Handling
Development Patterns

Declarative programming
Event orchestration
Activation & scale-to-zero
Service Binding

Continuous Integration
GitOps
Continuous Delivery

Traffic Routing
Network Resilience
Security

Infrastructure Concerns

Provisioning
Scheduling & Deployment
Scaling & Service Discovery
Monitoring and Recovery



Application Concerns

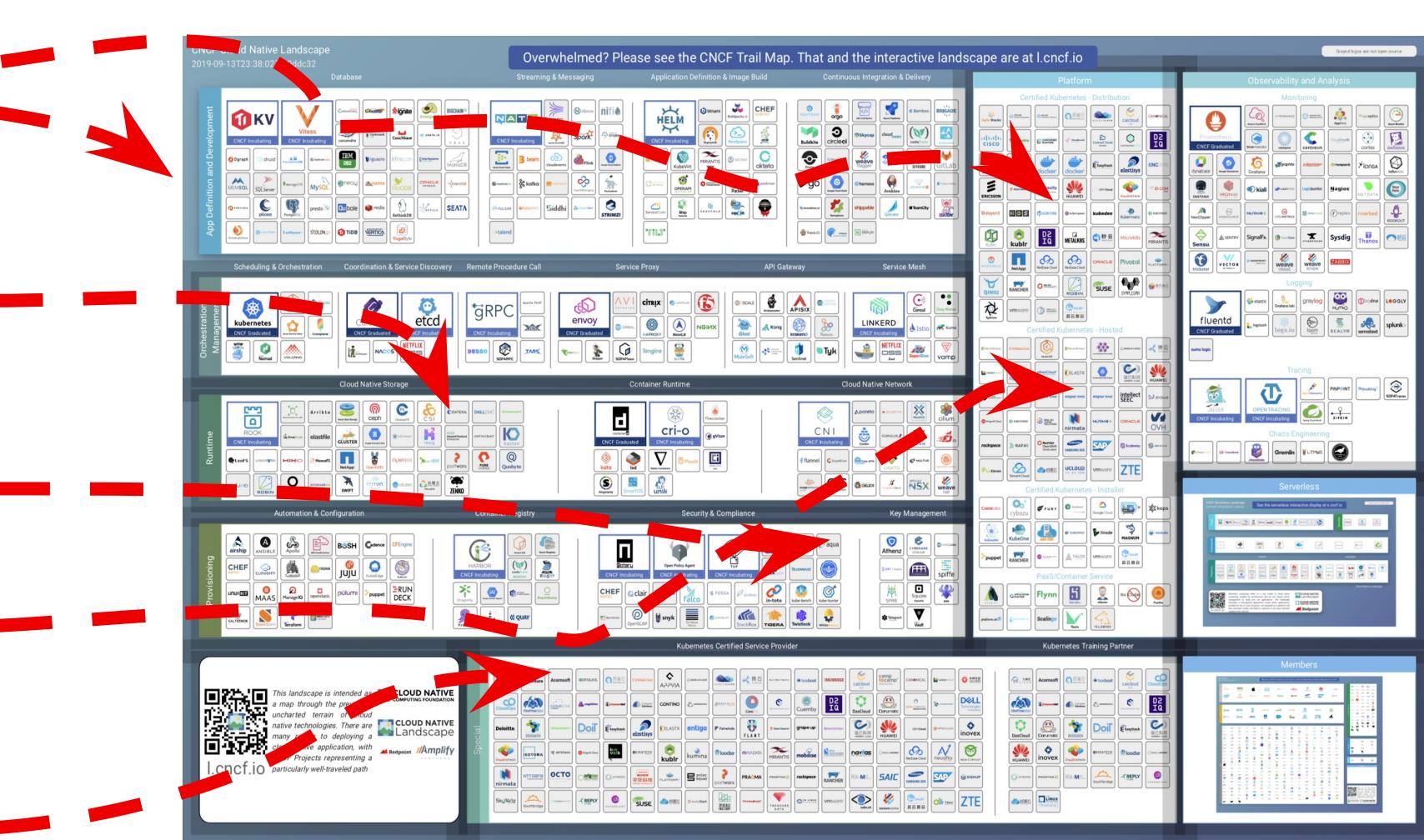
Routing & transformation
Technology Adapters
Error Handling
Development Patterns

Declarative programming
Event orchestration
Activation & scale-to-zero
Service Binding

Continuous Integration
GitOps
Continuous Delivery

Traffic Routing Network Resilience Security

- Provisioning
- Scheduling & Deployment
- Scaling & Service Discovery
- Monitoring and Recovery



Application Concerns



Routing & transformation
Technology Adapters
Error Handling
Development Patterns



Stateless	Batch Job	Microservices
Stateful	Functions	Singleton	

Declarative programming
Event orchestration
Activation & scale-to-zero
Service Binding



SERVING API	EVENTING API
-------------	--------------

Continuous Integration
GitOps
Continuous Delivery



Task	Pipeline	Step
------	----------	------

Traffic Routing
Network Resilience
Security



ServiceEntry	Gateway	VirtualService
--------------	---------	----------------

Provisioning
Scheduling & Deployment
Scaling & Service Discovery
Monitoring and Recovery



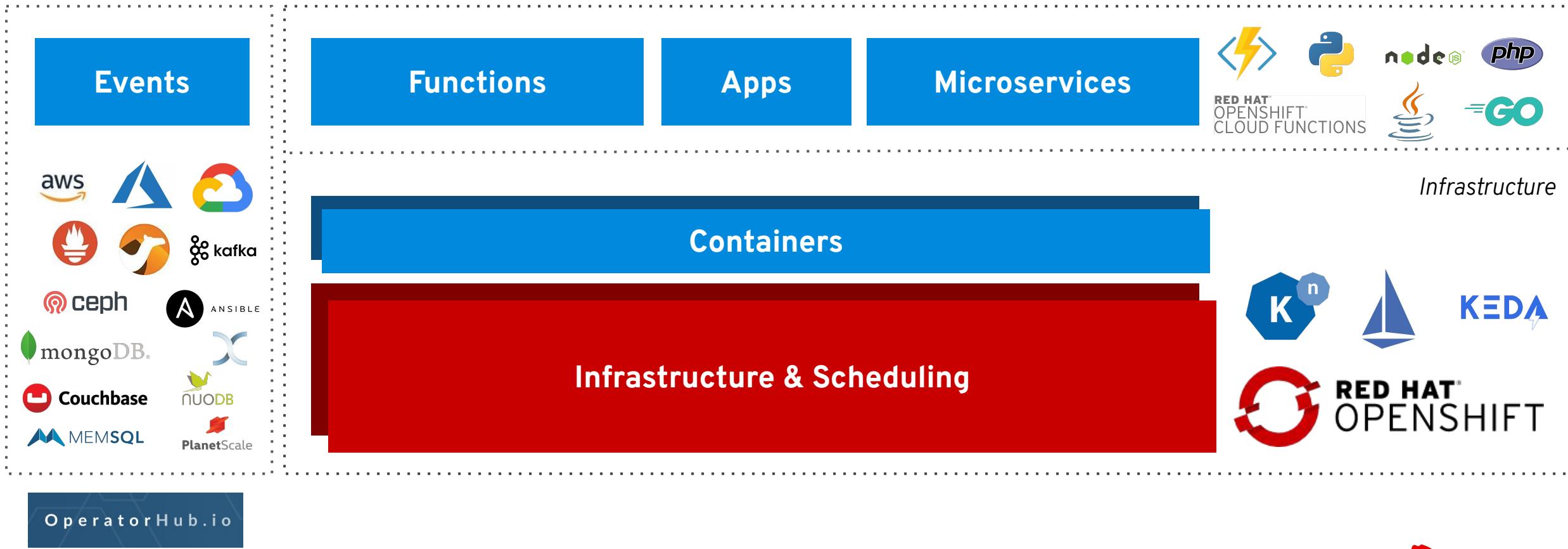
ReplicaSet	Stateful	Deployment
Service	Container	Logs
CronJob	ConfigMap	Health

High-level primitives

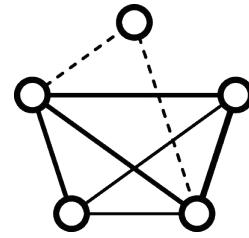


Low-level primitives

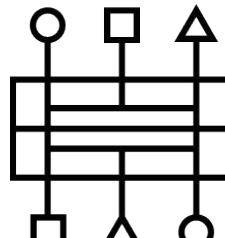
Microservices, Functions and Apps + Events



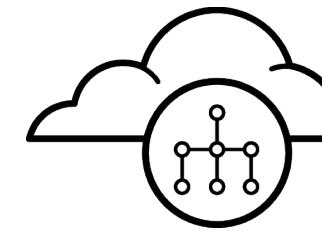
First Principles



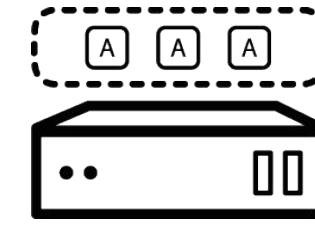
Distributed



API Centric



Multi-cloud



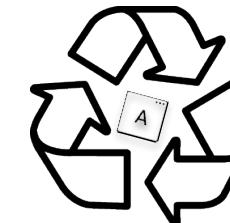
Scalable



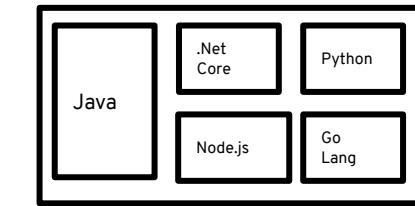
Secure



Event Driven



Disposable



Polyglot

Serverless & Knative

Range	Last month ▾	Metric	Contributions ▾
Knative Companies statistics (Contributions, Range: Last month), bots excluded ▾			
Company			
All		Number ▾	8077
Google			5431
Red Hat			1001
IBM			457
Pivotal			103
Dropbox			45
TriggerMesh			27
Alfresco			21
Microsoft			20
Mirantis			7
SAP			6
Cisco			6
III			5
BoCloud			5
The Elegant Monkeys			2
Stark & Wayne			2
Schibsted			2
StatDNA			1
EVRY			1
China Mobile			1

Companies
contributing to Knative

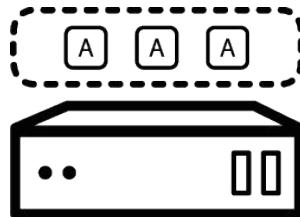
Source:

<https://knative.teststats.cnfc.io/d/8/dashboards?refresh=15m&orgId=1>

What is Knative ?

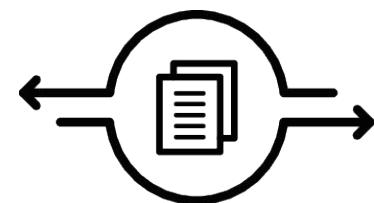
SERVING

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.



Community Operators

The screenshot shows the Red Hat OpenShift Container Platform interface. The left sidebar includes links for Home, Projects, Status, Search, Events, Catalog (Developer Catalog, Installed Operators), OperatorHub (selected), Operator Management, Workloads (Pods, Deployments, Deployment Configs, Stateful Sets, Secrets, Config Maps), Cron Jobs, Jobs, Daemon Sets, and Replica Sets. The main content area displays the OperatorHub for the default project, with a search bar containing "kn". It lists categories like All Items, AI/Machine Learning, Application Monitoring, Big Data, Cloud Provider, Database, Developer Tools, Integration & Delivery, Logging & Tracing, Monitoring, Networking, OpenShift Optional, Security, Storage, Streaming & Messaging. Two operators are highlighted: the Knative Eventing Operator (provided by Knative Community) and the Knative Serving Operator (provided by Knative Community). Both operators are associated with the Knative logo and a "Community" badge.

Project: default

OperatorHub

Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. Operators can be installed in your cluster to manage specific applications or system components.

All Items

All Items

2 items

AI/Machine Learning

Application Monitoring

Big Data

Cloud Provider

Database

Developer Tools

Integration & Delivery

Logging & Tracing

Monitoring

Networking

OpenShift Optional

Security

Storage

Streaming & Messaging

Knative Eventing Operator
provided by Knative Community

Knative Eventing is a system that is designed to address a common need for cloud native development and provides a simple way to handle events from various sources.

Knative Serving Operator
provided by Knative Community

Knative Serving builds on Kubernetes to support deploying and serving of serverless applications and functions.

kn

You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to log in.

Project: default

Installed Operators > Operator Details

 Serverless Operator
1.0.0 provided by Red Hat

Actions ▾

Overview YAML Subscription Events Knative Serving

Provided APIs

 Knative Serving
Represents an installation of a particular version of Knative Serving

 Create Instance

Description

The Red Hat Serverless Operator provides a collection of API's to install various "serverless" services.

Knative Serving

Provider
Red Hat

Created At
 Sep 13, 11:12 pm

Links
Documentation
<https://developers.redhat.com/topics/serverless-architecture/>

Source Repository
<https://github.com/openshift-knative/serverless-operator>

Maintainers
Serverless Team
knative@redhat.com

Productized Operator

User Experience

```
kn service create hello --image gcr.io/knative-samples/helloworld-go --env NAME=value
Service 'hello' successfully created in namespace 'default'.
```

Options: `--limits-cpu 100m --limits-memory 1024m`

kn service get hello					
NAME	DOMAIN	GENERATION	AGE	CONDITIONS	READY
hello	hello.default.example.com	1	3m5s	3 OK / 3	True

Other commands available: `update`, `replace`, `get`, `describe`

"Lift & Shift" to Serverless



```

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

```

Kubernetes

```

---
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

```

53 lines

Knative

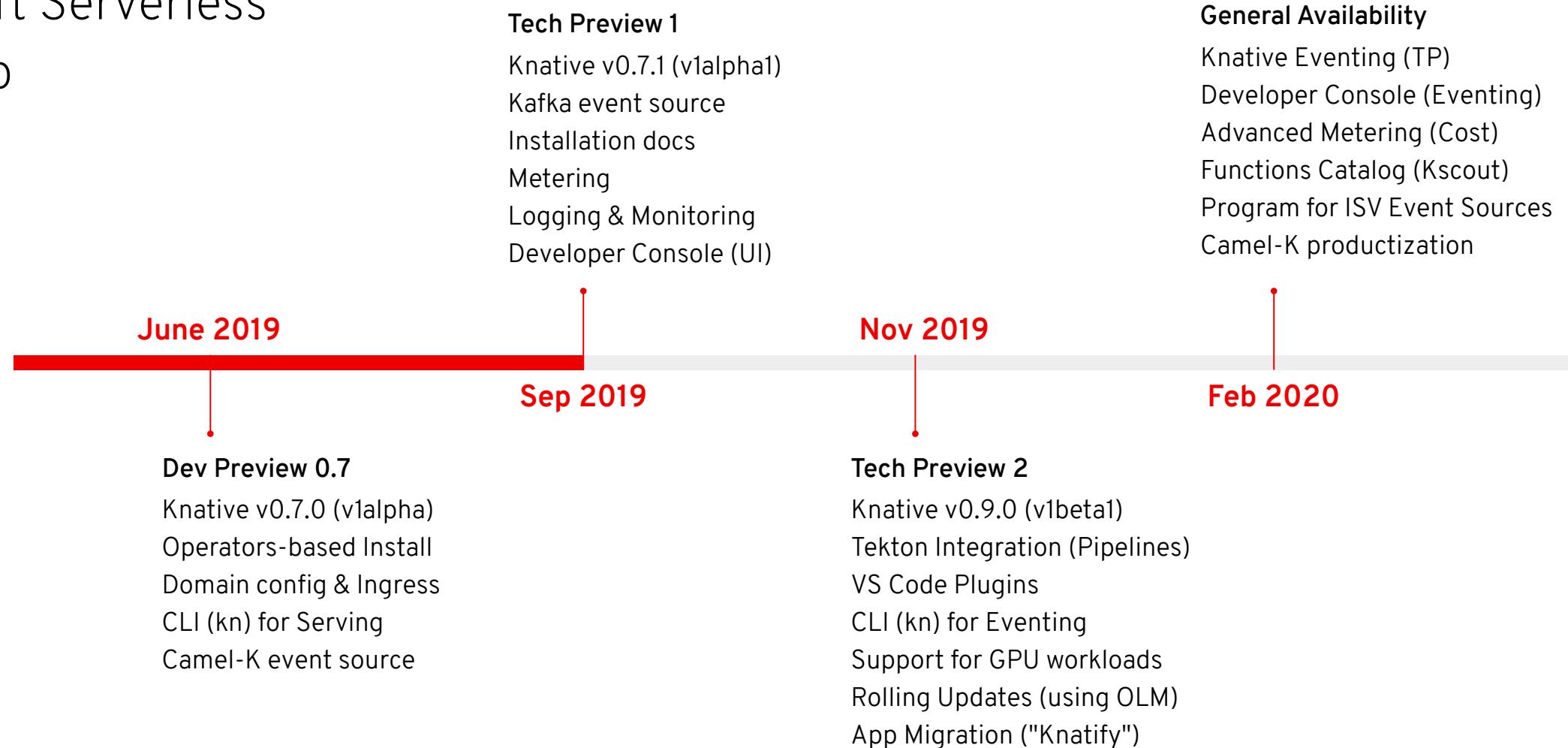
```

apiVersion: serving.knative.dev/v1alpha1
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

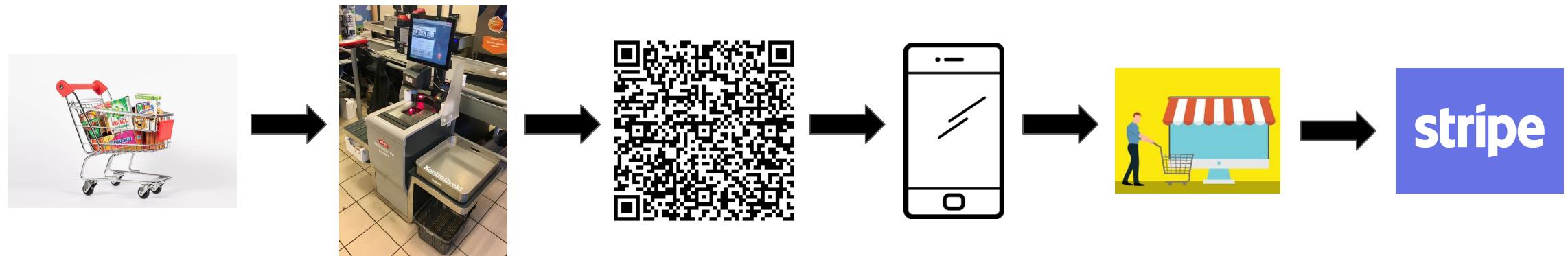
OpenShift Serverless Roadmap





DEMO

The Journey to Profit\$! with Serverless



Admin User Experience

The screenshot shows the Red Hat OpenShift Container Platform Admin UI. The left sidebar includes sections like Home, Catalog (Developer Catalog, Installed Operators), OperatorHub (Operator Management), Workloads, Networking, Storage, Builds, Monitoring, Compute, and Administration. The main content area displays the OperatorHub interface for the project 'openshift-operators'. It shows a message: 'You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.' Below this, there are two tabs: 'All Items' and 'All Items' (with a note '5 items'). The items listed are:

- Knative Apache Camel Operator (provided by Red Hat)
- Knative Apache Kafka Operator (provided by Red Hat)
- Knative Eventing Kafka (manages the Kafka source and channel provisioner)
- Knative Serving Operator (provided by Red Hat)
- TriggerMesh (A serverless management platform that runs on Knative. TriggerMesh provides continuous delivery of)

At the bottom left, there's a search bar with 'kn' typed in. At the bottom right, it says 'INSTALL STATE' and 'Installed (0)'.

User Experience

Applies to:

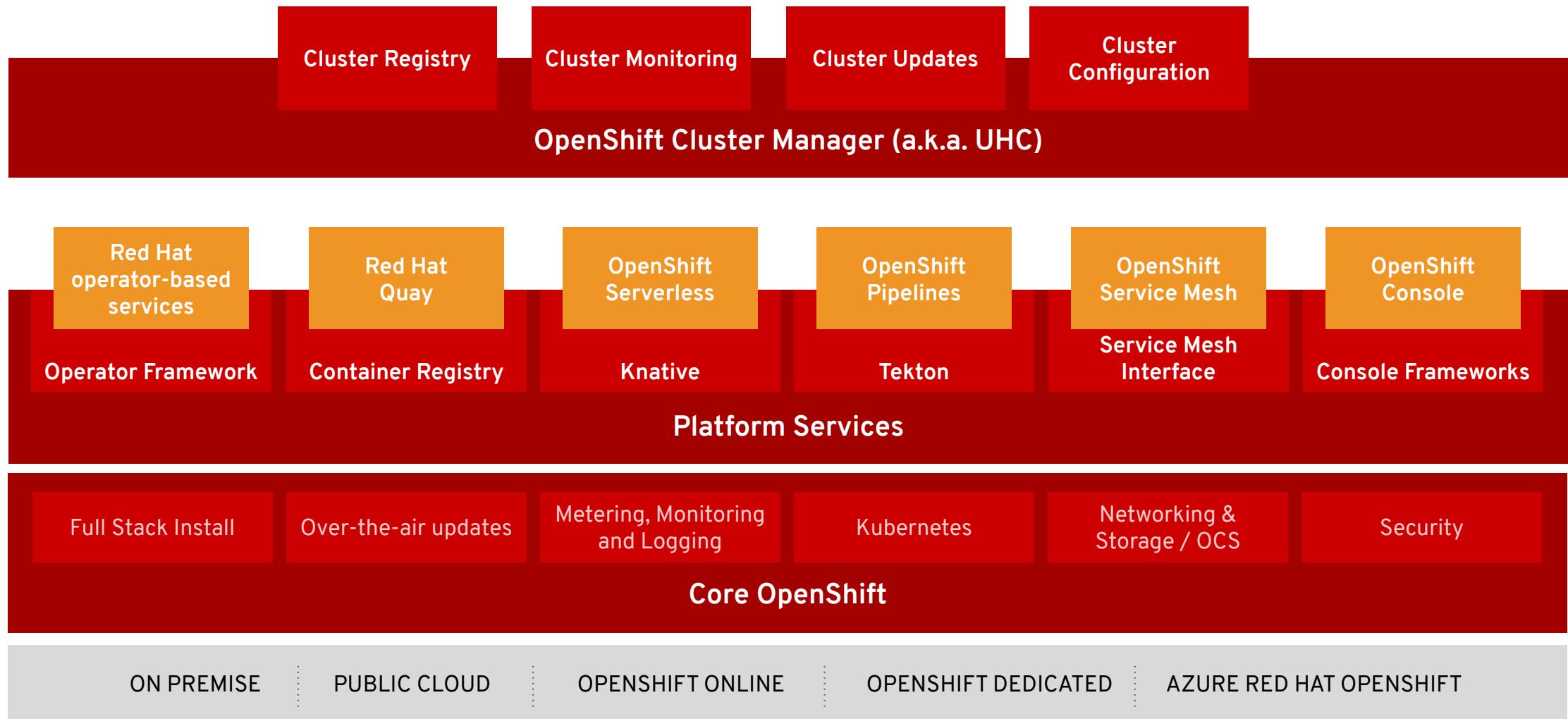
- Deploy Image
- Import from Git
- Create new App

The screenshot shows the Red Hat OpenShift Container Platform web interface. The top navigation bar includes the Red Hat logo, 'Red Hat OpenShift Container Platform', a user dropdown for 'kube:admin', and a 'Import YAML' button. The left sidebar, titled 'Administrator', lists various menu items: Home, Operators, Workloads, Serverless, Networking, Storage, Builds, Monitoring, Compute, and Administration. The main content area is titled 'Topology' and contains five cards: 'Import from Git' (with a 'Import from Git' button), 'Browse Catalog' (with a 'Browse Catalog' button), 'Deploy Image' (with a 'Deploy Image' button), 'Import YAML' (with a 'Import YAML' button), and 'Add Database' (with a 'Add Database' button). A blue banner at the top states: 'You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.'

* Developer Perspective coming in OpenShift 4.2

Summary & Roadmap

OpenShift Platform Services



Azure Functions & KEDA

Key features

- Enable FaaS in OpenShift
- Familiar developer experience using VS Code and Azure CLI
- Polling based auto-scaling for Azure Queues, Kafka...
- Reuse Knative event sources, HTTP auto-scaling
- On premise or Any cloud.
- Familiar to Kubernetes users. Native.

Learn more

- <https://github.com/kedacore/keda>

**Red Hat****Microsoft Azure**

OpenShift Serverless



Key Features

- Familiar to Kubernetes users. Native.
- Scale to 0 and autoscale to N based on demand
- Applications and functions. Any container workload.
- Powerful eventing model with multiple event sources.
- Operator available via OperatorHub
- Knative v0.8 (v1beta1 APIs)
- No vendor lock in

Learn more

<https://openshift.com/learn/topics/serverless>

<https://redhat-developer-demos.github.io/knative-tutorial>

The screenshot shows the Red Hat OpenShift Container Platform interface. On the left, the navigation sidebar includes sections for Administrator, Home, Dashboards, Projects, Search, Explore, Events, Operators (selected), Workloads, Serverless (selected), Services, Revisions, Routes, and Networking. The main content area displays the 'Operator Details' for the 'Serverless Operator' (version 1.0.0 provided by Red Hat). It shows a summary for the 'Project: default' and 'Project: markito-rhte'. In the 'markito-rhte' project, it shows an application named 'all applications' with a deployment named 'spring-petclinic-bchpw-deployment'. A large circular progress bar indicates '4 scaling to 10'. Below the progress bar, deployment details are listed: Name: spring-petclinic-bchpw-deployment, Namespace: markito-rhte, Labels: app=... (multiple labels), Update Strategy: RollingUpdate, Max Unavailable: 25% of 10 pods, Max Surge: 25% greater than 10 pods, Progress Deadline: 2m 0s, Min Ready Seconds: Not Configured.

See you soon at...



Prodotti

Soluzioni

Servizi e supporto

Resources

L'open source secondo Red Hat



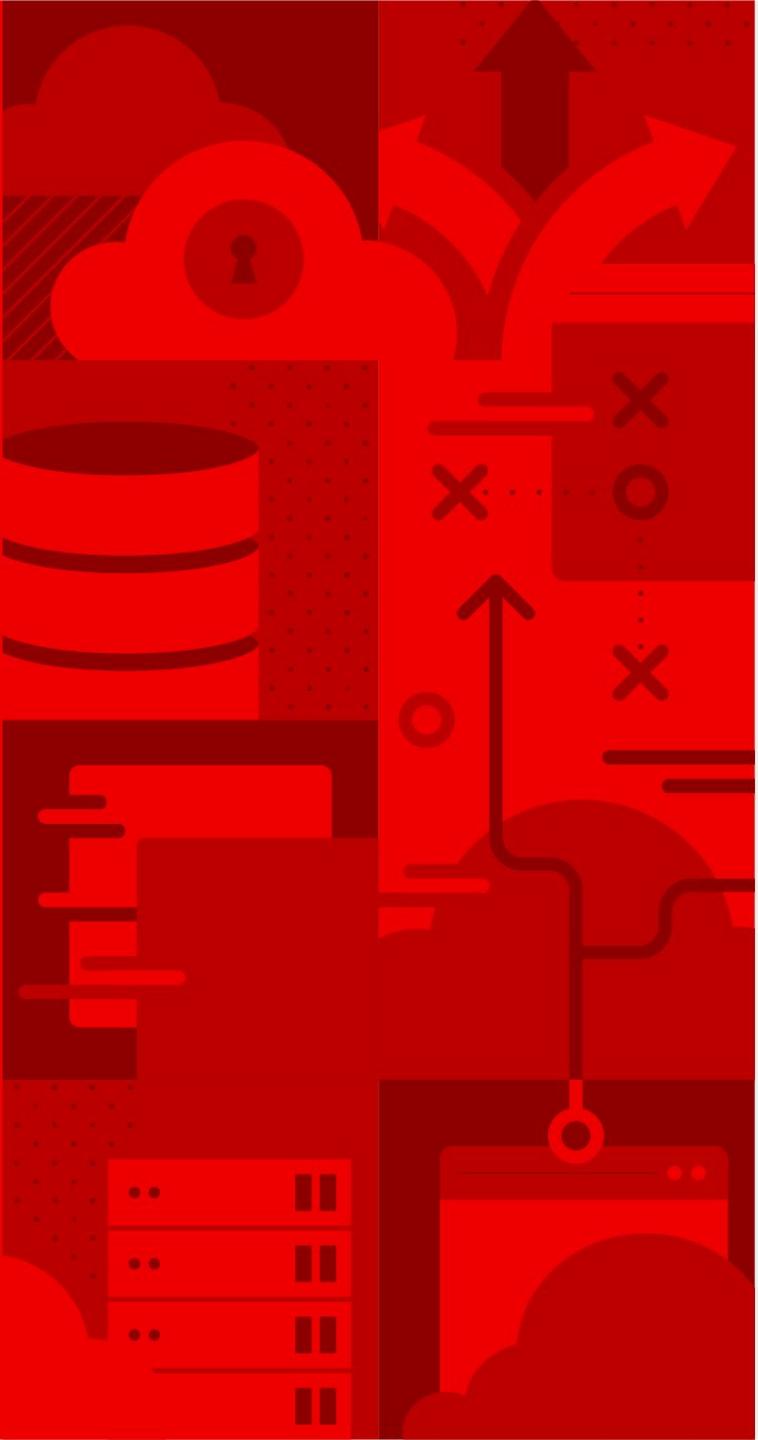
EXPAND YOUR POSSIBILITIES

Red Hat Open Source Day 2019

Roma 20 novembre e Milano 3 dicembre

Registrati Roma

Registrati Milano



Thank you

- openshift.com/learn/topics/knative
- <https://redhat-developer-demos.github.io/knative-tutorial>
- github.com/kedacore/keda



linkedin.com/company/red-hat



facebook.com/redhatinc



youtube.com/user/RedHatVideos



twitter.com/RedHat