

Node.js Cloud-Native Apps with Knative and Istio



Carlos Santana @csantanapr
STSM – Cloud Native GSE
IBM Cloud Garage



Last Update: 2019-11

Serverless for everyone

How can I use
existing binaries

Don't vendor lock me

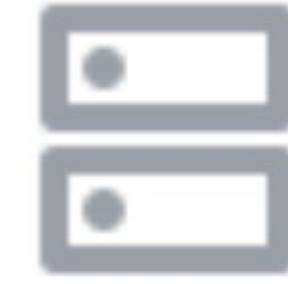
I want pick the
Linux OS

Can you support language x?

Don't make me choose between
containers and serverless



Serverless usage models



No servers



Pay only for usage



Event-driven



Portable

Knative



OSS: Apache License 2.0

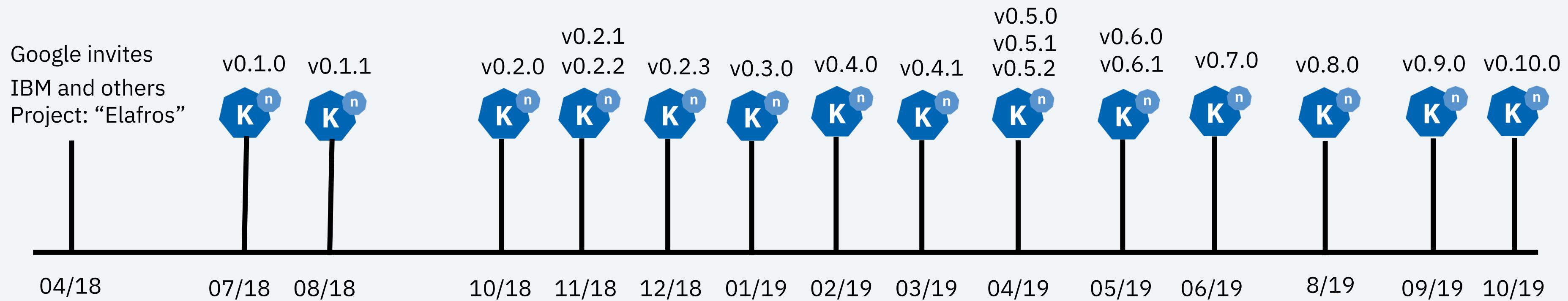


- Covers, ~~build~~, **serving**, **eventing** ...
- Common API and runtimes environments for serving workloads
- Focus on **modern** development patterns
- Implements **learnings** from over 50 companies contributing
- **Portability** of experiences, tooling, and workloads between Knative environments on any Kubernetes, Public or **On-Prem**

<https://github.com/knative>

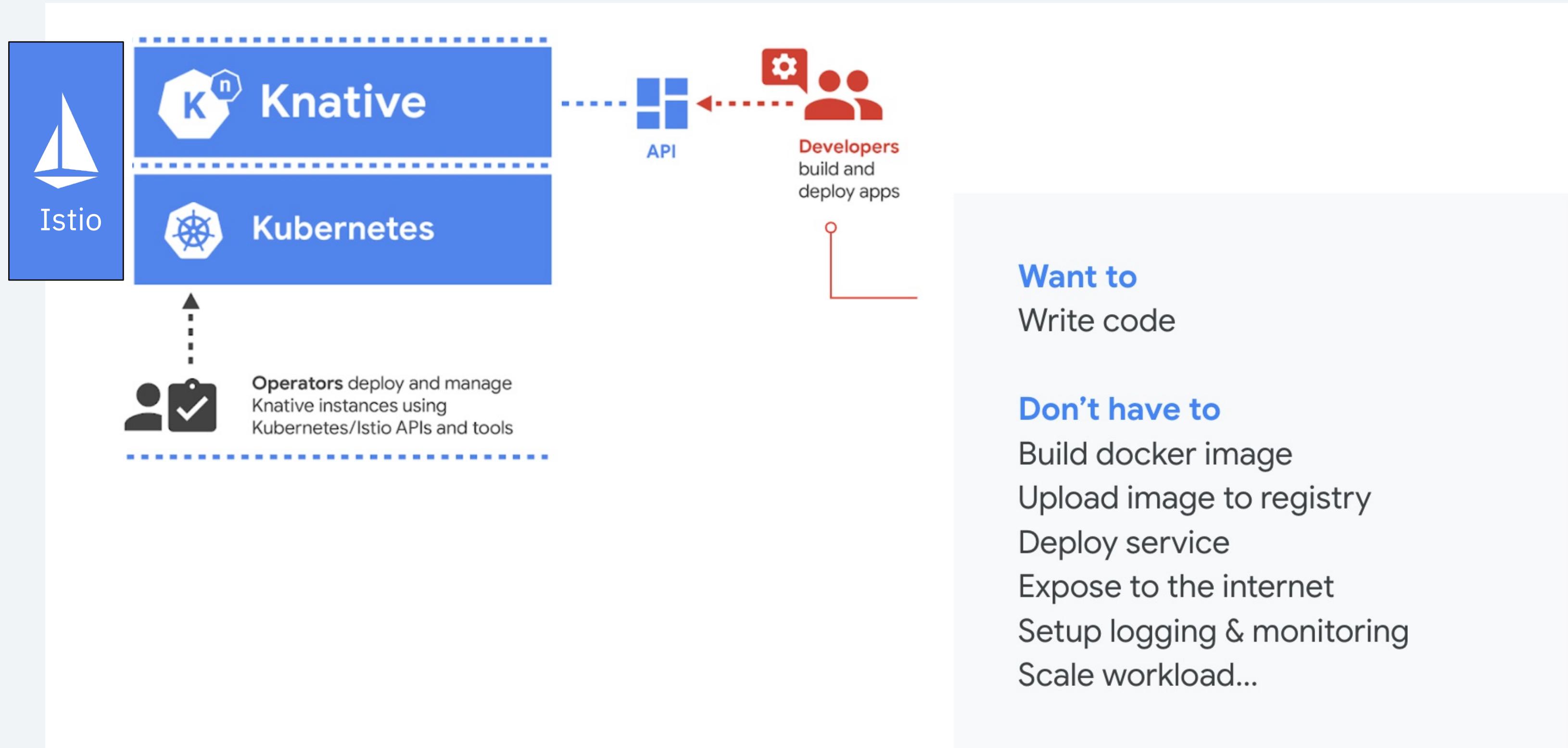
<https://knative.dev>

Knative releases

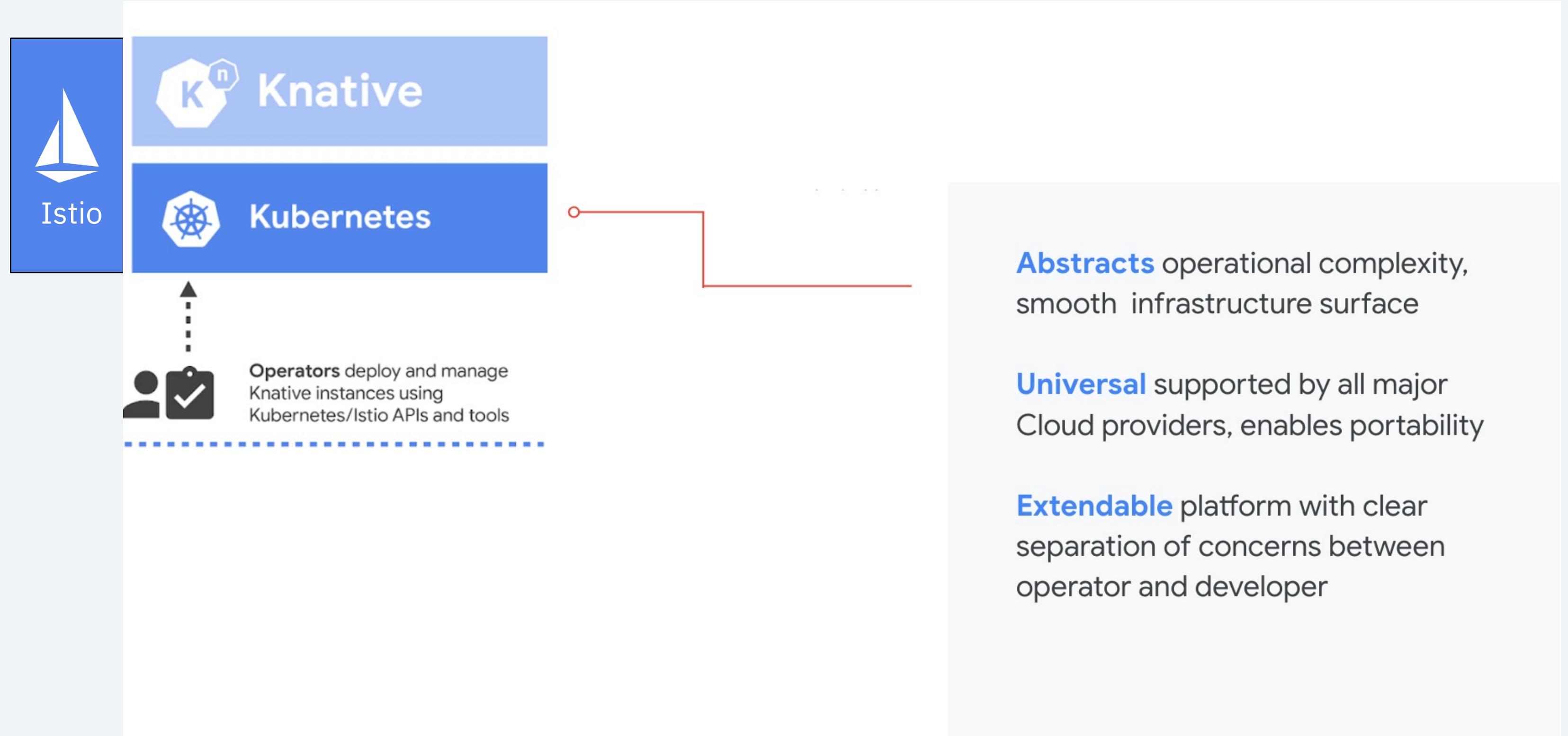


<https://github.com/knative/serving/commit/7f5da0b7438f562ea4a45c7021a674e2f90a5ddf>

Knative for developers



Knative for operators



Containers

Any language

Any library

Any binary

Ecosystem of base images

Industry standard



Hello World

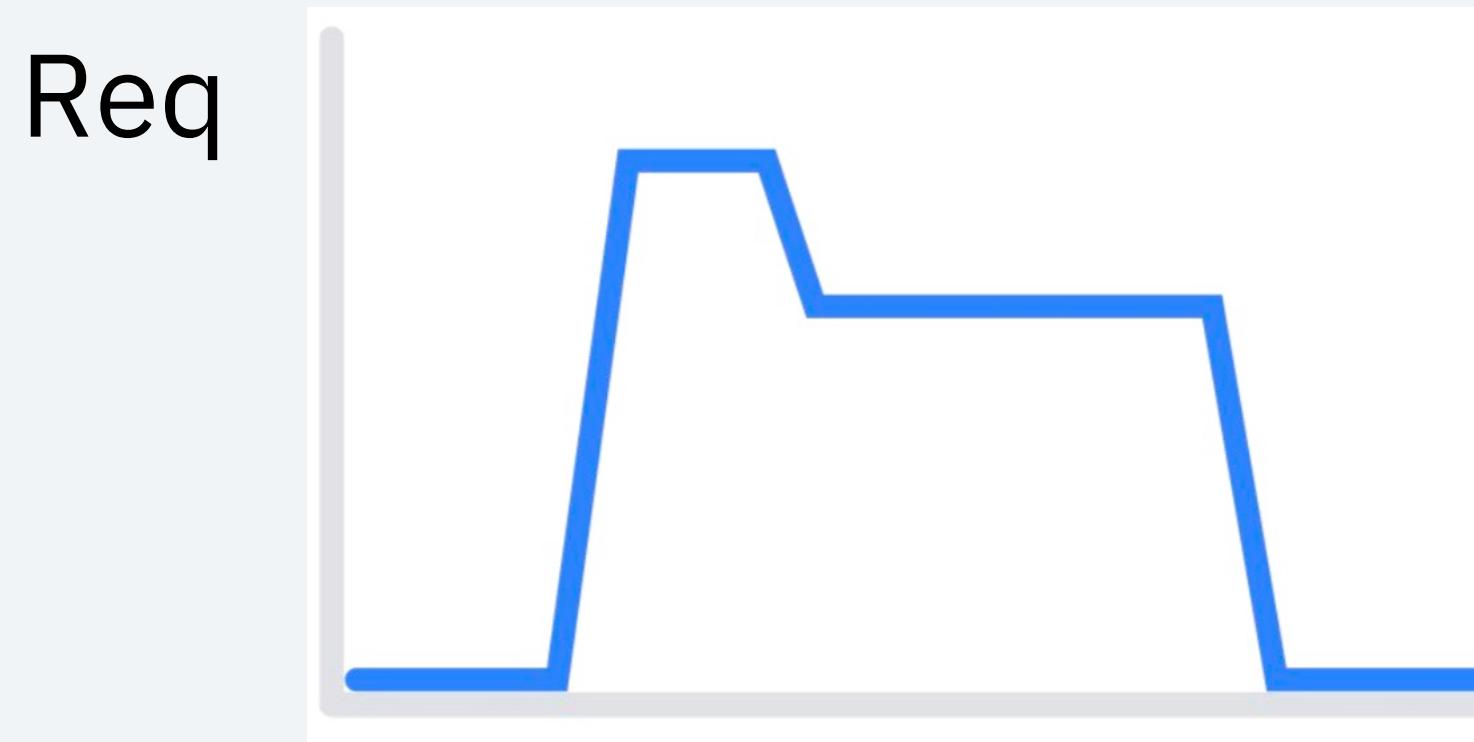
```
▶ kn service create helloworld --image csantanapr/helloworld  
Service 'helloworld' successfully created in namespace 'default'.
```

```
▶ kn service list
```

NAME	DOMAIN	GENERATION	AGE	CONDITIONS	READY	REASON
helloworld	helloworld-default.csantana-demos.us-south.containers.appdomain.cloud	1	45s	3 OK / 3	True	



Automatically scales based on requests



Scales out fast

Scales down to zero

Higher Utilization

Knative Serving



Benefits

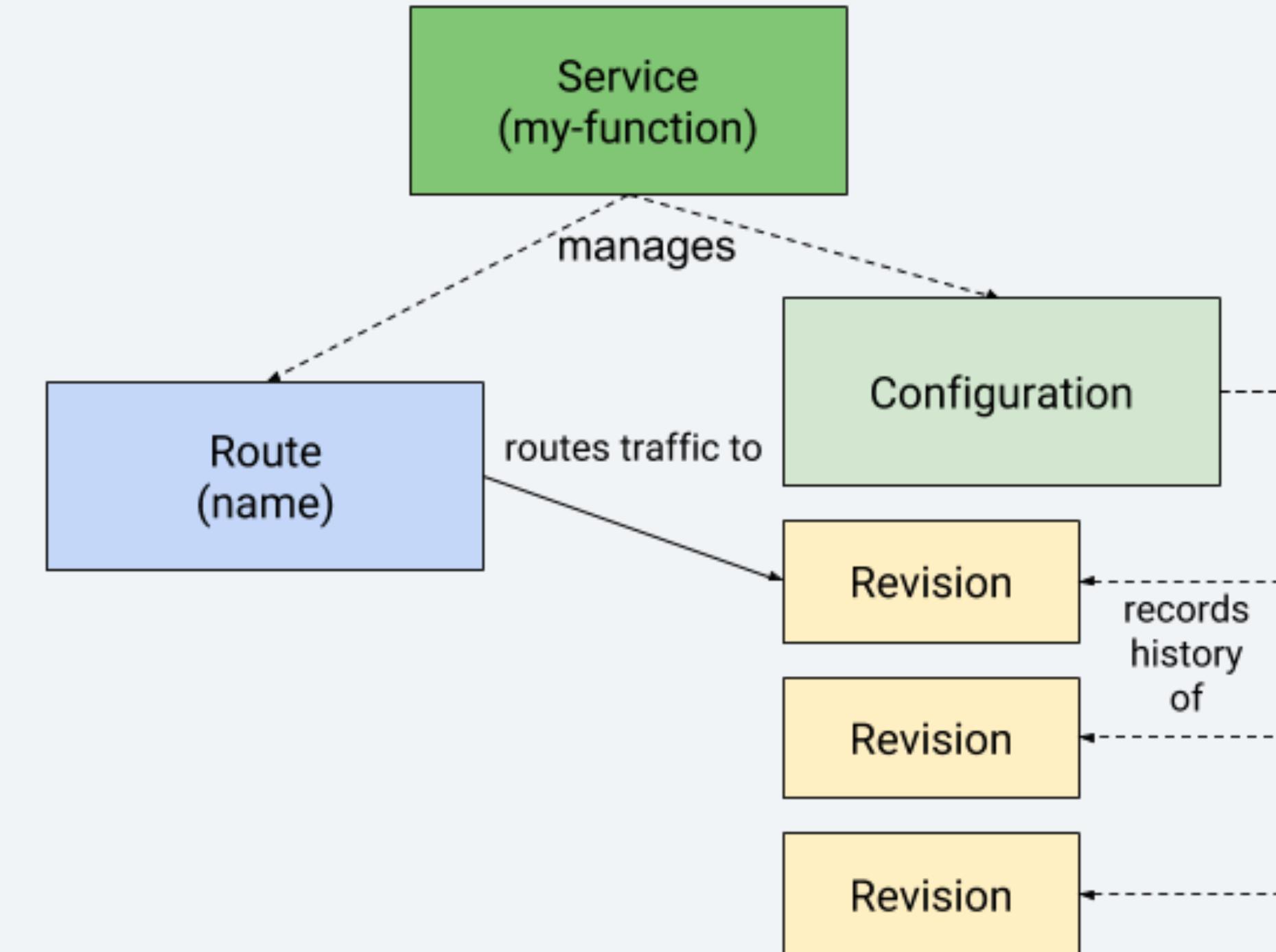
- Automatic request-based activation and scaling (0-n)
- Code/config revision management (rollback)
- Traffic splitting between revisions/versions (“canary”)
- Request path access controls (internal & external services)

Pluggable

- Connect to your own logging and monitoring platform, or use built-in system
- Auto-scaler can be tuned or swapped out for custom code
- Swappable ingress gateway Istio is optional

Knative Service Resource Types

- A **Route** provides a named endpoint that routes traffic to one or more revisions.
- **Revisions**, which are immutable snapshots of code + config, created by a Configuration
- **Configuration**, which acts as a stream of environment Revisions.
- **Service** is a top level abstraction for managing Routes and Configurations



Knative Service yaml (minimum)

```
1  apiVersion: serving.knative.dev/v1beta1
2  kind: Service
3  metadata:
4    name: helloworld
5  spec:
6    template:
7      spec:
8        containers:
9          - image: csantanapr/helloworld
```

```
▶ kubectl apply -f service.yaml
service.serving.knative.dev/helloworld configured
```

<https://github.com/knative/serving/blob/master/docs/runtime-contract.md>

<https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.10/#containerport-v1-core>

* Default port is 8080, expose as environment variable PORT to the container, can override with containerPort

Knative custom resources (CR)

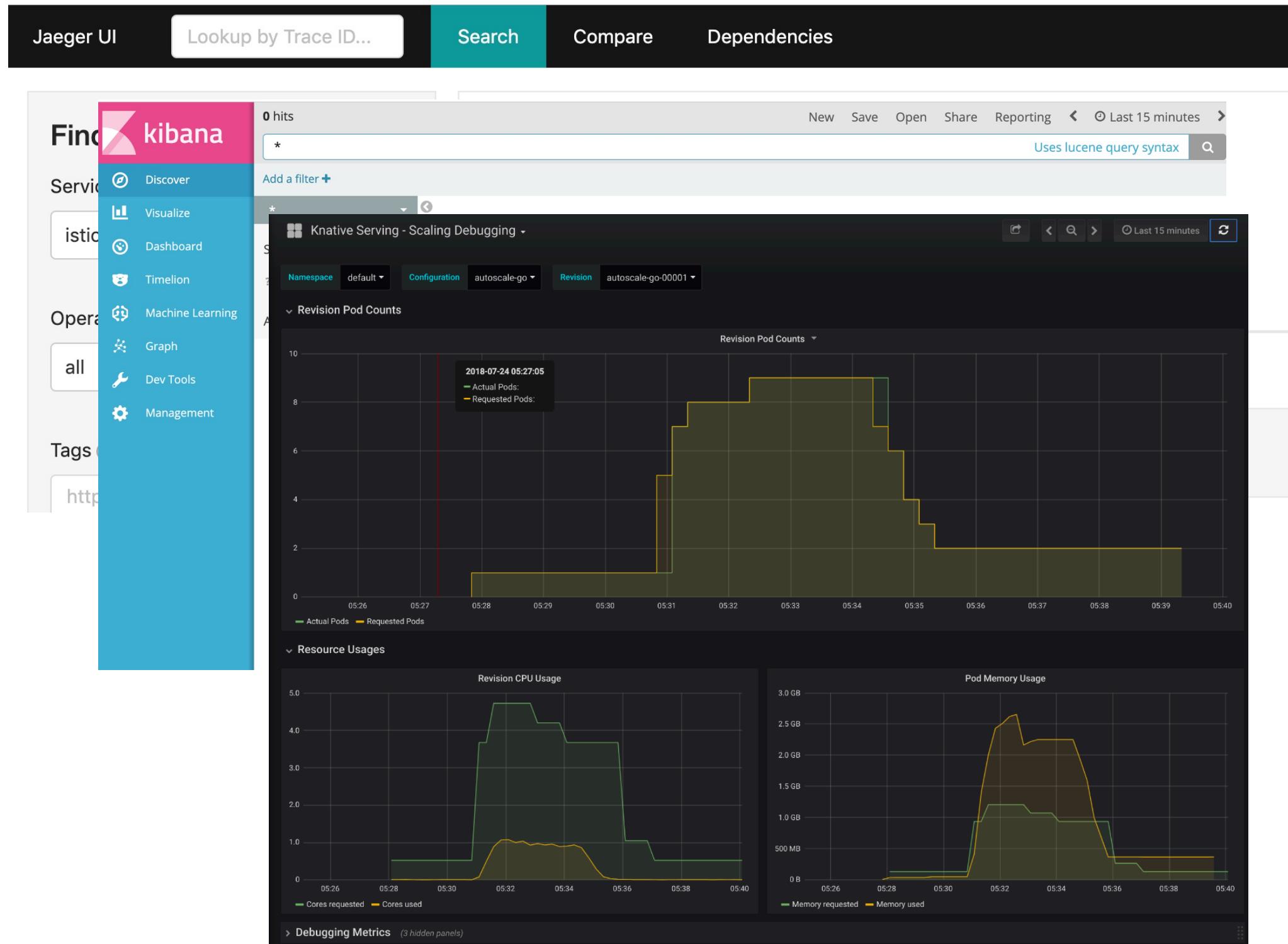
▶ kubectl apply -f service.yaml							
	service.serving.knative.dev/helloworld configured						
▶ kubectl get ksvc							
NAME	URL						
helloworld	http://helloworld-default.csantana-demos.us-south.containers.appdomain.cloud				LATESTCREATED		LATESTREADY
					helloworld-xg59c		READY
▶ kubectl get configuration							
NAME	LATESTCREATED	LATESTREADY	READY	REASON			
helloworld	helloworld-xg59c	helloworld-xg59c	True				
▶ kubectl get revision							
NAME	SERVICE NAME	GENERATION	READY	REASON			
helloworld-bs225	helloworld-bs225	1	True				
helloworld-v76f8	helloworld-v76f8	2	True				
helloworld-xg59c	helloworld-xg59c	3	True				
▶ kubectl get kpa							
NAME	READY	REASON					
helloworld-bs225	False	NoTraffic					
helloworld-v76f8	False	NoTraffic					
helloworld-xg59c	True						
▶ kubectl get route							
NAME	URL				READY	REASON	
helloworld	http://helloworld-default.csantana-demos.us-south.containers.appdomain.cloud				True		
▶ kubectl get virtualservice -n knative-serving							
NAME	GATEWAYS				HOSTS		
route-43c53ba4-976d-11e9-870d-feddda1c832c	[knative-ingress-gateway mesh]				[helloworld-default.csantana-demos.us-south.containers.appdomain.cloud]		

Kubernetes resources

▶ kubectl get deployments					
NAME	READY	UP-TO-DATE	AVAILABLE	AGE	
helloworld-bs225-deployment	0/0	0	0	132m	
helloworld-v76f8-deployment	0/0	0	0	129m	
helloworld-xg59c-deployment	1/1	1	1	38m	
▶ kubectl get replicaset					
NAME	DESIRED	CURRENT	READY	AGE	
helloworld-bs225-deployment-55cf49588d	0	0	0	132m	
helloworld-v76f8-deployment-846bd476d5	0	0	0	130m	
helloworld-xg59c-deployment-7ffcb44896	1	1	1	38m	
▶ kubectl get pods					
NAME	READY	STATUS	RESTARTS	AGE	
helloworld-xg59c-deployment-7ffcb44896-tcbgk	2/2	Running	0	3m24s	
▶ kubectl get service					
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
helloworld	ExternalName	<none>	istio-ingressgateway.istio-system.svc.cluster.local	<none>	133m
helloworld-bs225	ClusterIP	172.21.235.60	<none>	80/TCP	133m
helloworld-bs225-metrics	ClusterIP	172.21.148.62	<none>	9090/TCP	133m
helloworld-bs225-priv	ClusterIP	172.21.94.11	<none>	80/TCP	133m
helloworld-v76f8	ClusterIP	172.21.61.212	<none>	80/TCP	130m
helloworld-v76f8-metrics	ClusterIP	172.21.204.145	<none>	9090/TCP	130m
helloworld-v76f8-priv	ClusterIP	172.21.228.11	<none>	80/TCP	130m
helloworld-xg59c	ClusterIP	172.21.97.209	<none>	80/TCP	39m
helloworld-xg59c-metrics	ClusterIP	172.21.71.241	<none>	9090/TCP	39m
helloworld-xg59c-priv	ClusterIP	172.21.228.157	<none>	80/TCP	39m

Build to manage

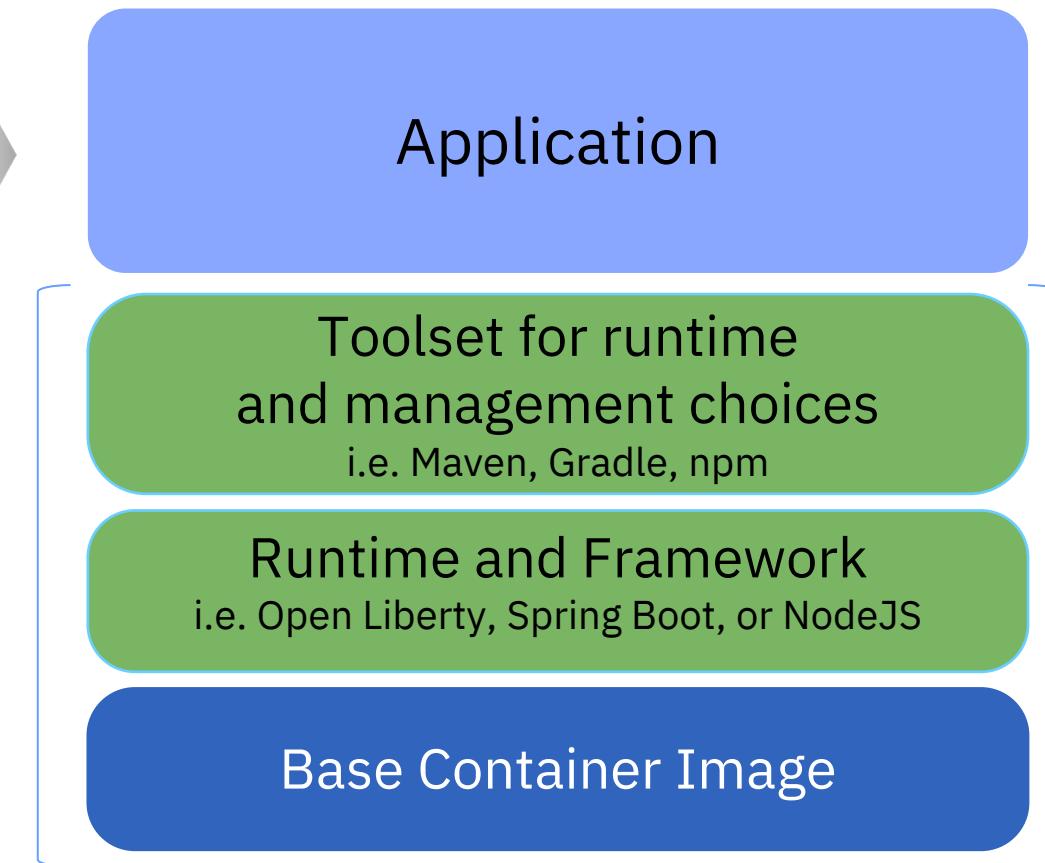
- Probe checks
 - Live, Ready, Health
- Logging
 - Formatting and Error catalog
 - Diagnostic Report (FFDC)
- Metrics
- Distributed Tracing
- Service Mesh
- Topology



Stacks



**Appsody
Project
Template**



Stack

Runtimes & frameworks in pre-built container images, ready to go or customize them yourself.

Optimized for the target platform

- **Open Tracing**
- **Metrics/Monitoring**
- **Logging**
- **Security and Auth**
- ... etc



Knative Autoscaling

KPA (Knative Pod Autoscaler)

```

1  apiVersion: serving.knative.dev/v1alpha1
2  kind: Service
3  metadata:
4    name: helloworld
5  spec:
6    template:
7      metadata:
8        annotations:
9          autoscaling.knative.dev/target: "50"
10         autoscaling.knative.dev/minScale: "2"
11         autoscaling.knative.dev/maxScale: "100"
12    spec:
13      containers:
14        - image: csantanapr/helloworld

```

HPA (Horizontal Pod Autoscaler)

```

1  apiVersion: serving.knative.dev/v1alpha1
2  kind: Service
3  metadata:
4    name: helloworld
5  spec:
6    template:
7      metadata:
8        annotations:
9          autoscaling.knative.dev/class: hpa.autoscaling.knative.dev
10         autoscaling.knative.dev/metric: cpu
11         autoscaling.knative.dev/target: "85"
12    spec:
13      containers:
14        - image: csantanapr/helloworld
15        resources:
16          limits:
17            cpu: "1"
18          requests:
19            cpu: 100m

```

The screenshot shows a terminal window with the title 'watch kubectl get pods (watch)'. It displays the status of two pods: 'helloworld-57mxc-deployment-7c98f94b85-9crwh' and 'helloworld-57mxc-deployment-7c98f94b85-p9mr5'. Both pods are ready with 2/2 replicas, running, and have 0 restarts. The age of the pods is 7m11s and 7m3s respectively. The command 'Every 2.0s: kubectl get pods' is shown at the bottom.

```

● ● ●  ▶ 2
watch kubectl get pods (watch)
Click here to configure status bar
Every 2.0s: kubectl get pods
iMac.local: Tue Jun 25 15:57:39 2019

NAME           READY   STATUS    RESTARTS   AGE
helloworld-57mxc-deployment-7c98f94b85-9crwh  2/2     Running   0          7m11s
helloworld-57mxc-deployment-7c98f94b85-p9mr5   2/2     Running   0          7m3s

```

▶ kubectl get hpa						
NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	AGE
helloworld-hxlhc	Deployment/helloworld-hxlhc-deployment	167%/85%	1	2147483647	28	4m30s

<https://github.com/knative/docs/blob/master/docs/serving/configuring-the-autoscaler.md>

<https://github.com/knative/docs/blob/master/docs/serving/samples/autoscale-go/README.md>

Knative driving Istio Traffic Splitting

Configurations

```

1  apiVersion: serving.knative.dev/v1alpha1
2  kind: Configuration
3  metadata:
4    name: hello-blue
5  spec:
6    revisionTemplate:
7      metadata:
8        labels:
9          knative.dev/type: container
10   spec:
11     container:
12       image: gcr.io/knative-samples/knative-route-demo:blue
13       imagePullPolicy: Always
14     env:
15       - name: T_VERSION
16         value: "blue"
17 ---
18 apiVersion: serving.knative.dev/v1alpha1
19 kind: Configuration
20 metadata:
21   name: hello-green
22 spec:
23   revisionTemplate:
24     metadata:
25       labels:
26         knative.dev/type: container
27     spec:
28       container:
29         image: gcr.io/knative-samples/knative-route-demo:green
30         env:
31           - name: T_VERSION
32             value: "green"
```

Route (blue 90%, green 10%)

```

34  apiVersion: serving.knative.dev/v1alpha1
35  kind: Route
36  metadata:
37    name: helloworld
38  spec:
39    traffic:
40      - configurationName: hello-blue
41        name: v1
42        percent: 90
43      - configurationName: hello-green
44        name: v2
45        percent: 10
```

* Route.spec.traffic could also split using revisionName instead of configurationName

```
kubectl get route -o yaml | grep url
  url: http://helloworld.default.svc.cluster.local
  url: http://helloworld-v1-default.csantana-demos.us-south.containers.appdomain.cloud
  url: http://helloworld-v2-default.csantana-demos.us-south.containers.appdomain.cloud
  url: http://helloworld-default.csantana-demos.us-south.containers.appdomain.cloud
```

Knative supported by IBM

- IBM Cloud Kubernetes Service (Knative add-on)
- IBM Cloud Pack for Applications (ICP4Apps)
 - Kabanero Enterprise (Appsody)
 - Red Hat OpenShift Serverless Operator

Resources

<https://knative.dev>

<https://developer.ibm.com/components/knative>

<https://github.com/IBM/knative101>

<https://github.com/csantanapr/knative-kubecon19eu>

<https://github.com/csantanapr/knative-ibm-developer>

<https://github.com/csantanapr/knative-eventing-basics>

<https://github.com/csantanapr/knative-think2019>

Thank you