The Journey to Cloud

Kubernetes Advanced Concepts I

Enterprise K8s, *ISTIO and knative*

Niklaus Hirt
DevOps Architect / Cloud Architect
nikh@ch.ibm.com

IBM

Who am I?

Niklaus Hirt

Passionate about tech for over 35 years

- High-school in Berne
- Degree in Computer Science at EPFL
- ELCA
- CAST
- IBM

✉ nikh@ch.ibm.com

@nhirt

Agenda - Kubernetes Advanced Concepts

Module - ADVANCED

Module 1: Enterprise grade Kubernetes

BREAK

Module 2: Mesh Networking with ISTIO

Module 3: Serverless with Knative

Module 4: GitOps with ArgoCD

BREAK

Module 5: Mesh Networking Hands-On

Wrap-up

Sources and documentation will be available here:

https://github.com/niklaushirt/k8s_training_public

The Journey to Cloud
Prepare the Labs

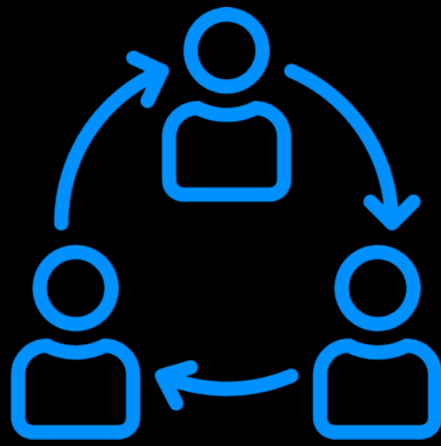
00

IBM **Cloud**

Session Objectives

Attendees will be grouped in ***teams*** wherever it makes sense to facilitate collaborative work.

Following some lectures will be ***hands-on*** work that each team collaborates to complete.



Teams

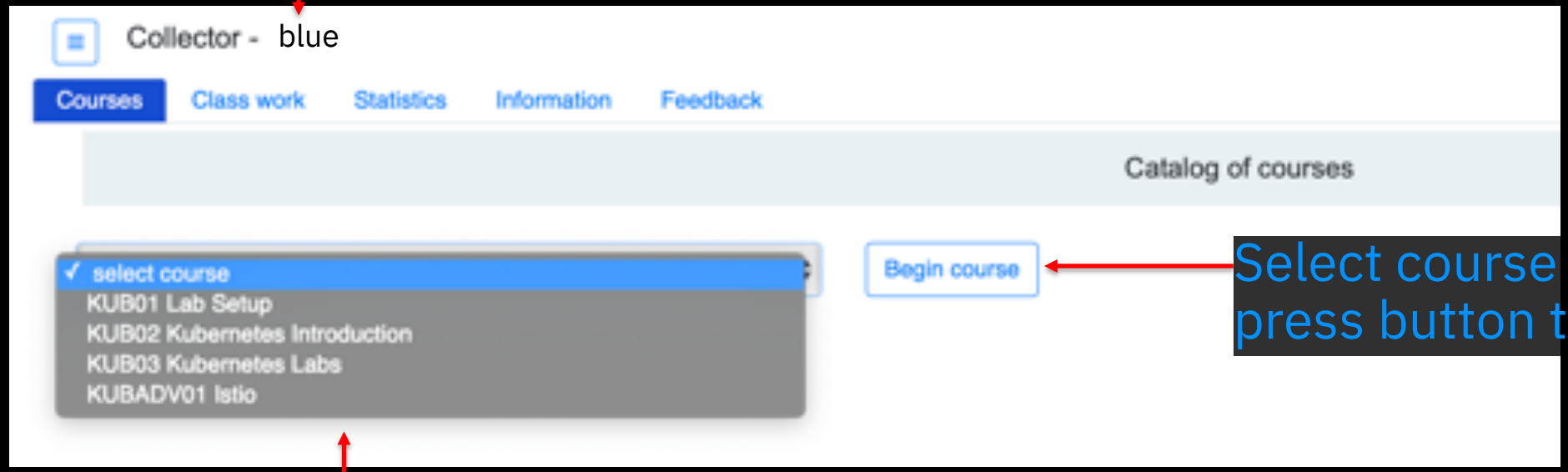
black 31701	olive 31711	peru 31715
white 31702	brown 31712	chocolate 31716
red 31703	lightblue 31713	orchid 31717
blue 31704	orange 31708	gold 31718
yellow 31705	purple 31709	pink 31719
lime 31706	maroon 31710	violet 31720
cyan 31707	firebrick 31714	

Collector - Accessing team web site

http://158.177.137.195:{port#}

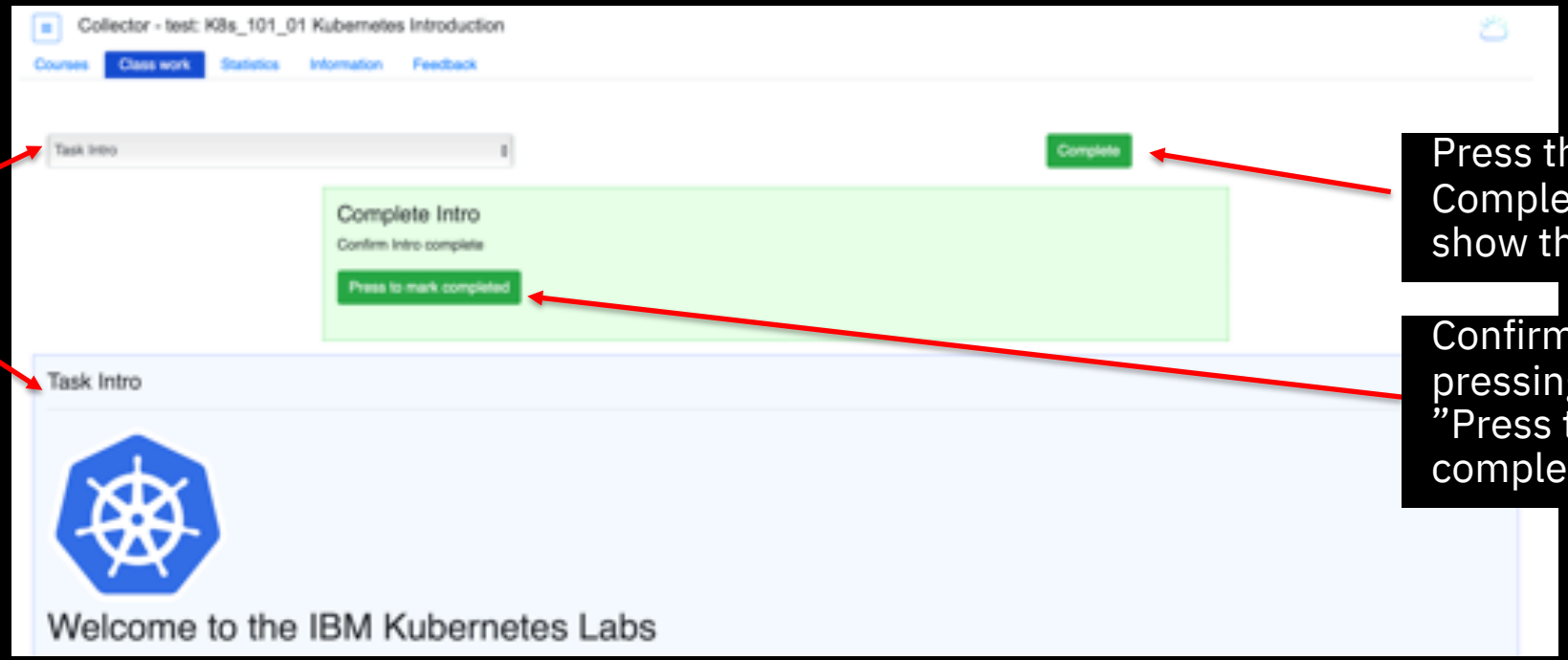
blue 31704

Team name / color will be shown



Collector – Class work

Select class work and the blue portion of the screen is shown



Press the green Complete button to show the green portion.

Confirm completion by pressing the green ”Press to mark completed” button.

The Complete Button might not show instantly depending on the course settings

Collector – Track course completed work

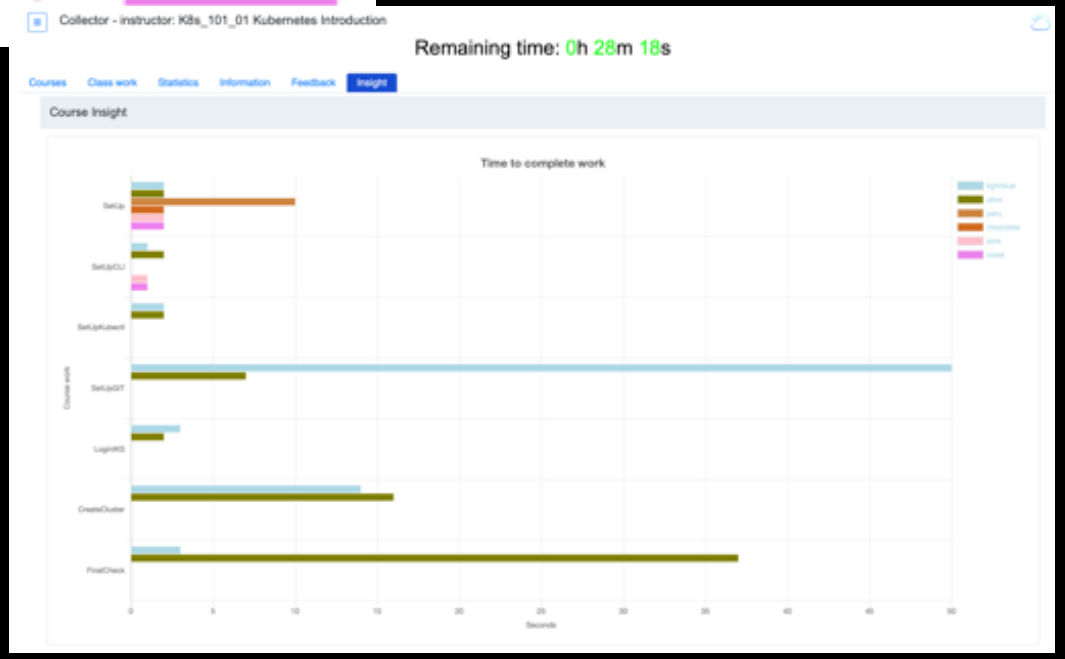
Course title

The number of items tracked will change based on the current course selected.

Green checkmark - item is completed
Red circle - item is waiting to be completed

Collector – Instructor Dashboard

Remaining Time for the Lab



JTC90 Kubernetes Lab Setup

Task 1: Setup Minikube

Task 2: Setup kubectl

Task 3: Setup git

Task 4: Final Check

EVERYBODY OK ?

JTC10 Istio

Lab 0 : Introduction

Lab 1 - Make sure minikube is running

Lab 2 - Installing Istio

Lab 3 - Deploy the Bookinfo App

Lab 4 - Monitoring with Kiali

Lab 5 - Traffic flow management

Lab 6 - Access policy enforcement

Lab 7 - Telemetry data aggregation

JTC10 Istio

Install

```
git clone https://github.com/niklaushirt/istio.git
cd istio
export PATH=$PWD/bin:$PATH
for i in install/kubernetes/helm/istio-init/files/crd*yaml;
      do kubectl apply -f $i; done
kubectl apply -f install/kubernetes/istio-demo.yaml
kubectl label namespace default istio-injection=enabled
```

Check

```
kubectl get ns --show-labels
kubectl get svc -n istio-system
kubectl get pods -n istio-system
```

Be patient!!!

http://158.177.137.195:{port#}

IBM

QUESTIONS?

IBM **Cloud**

IBM

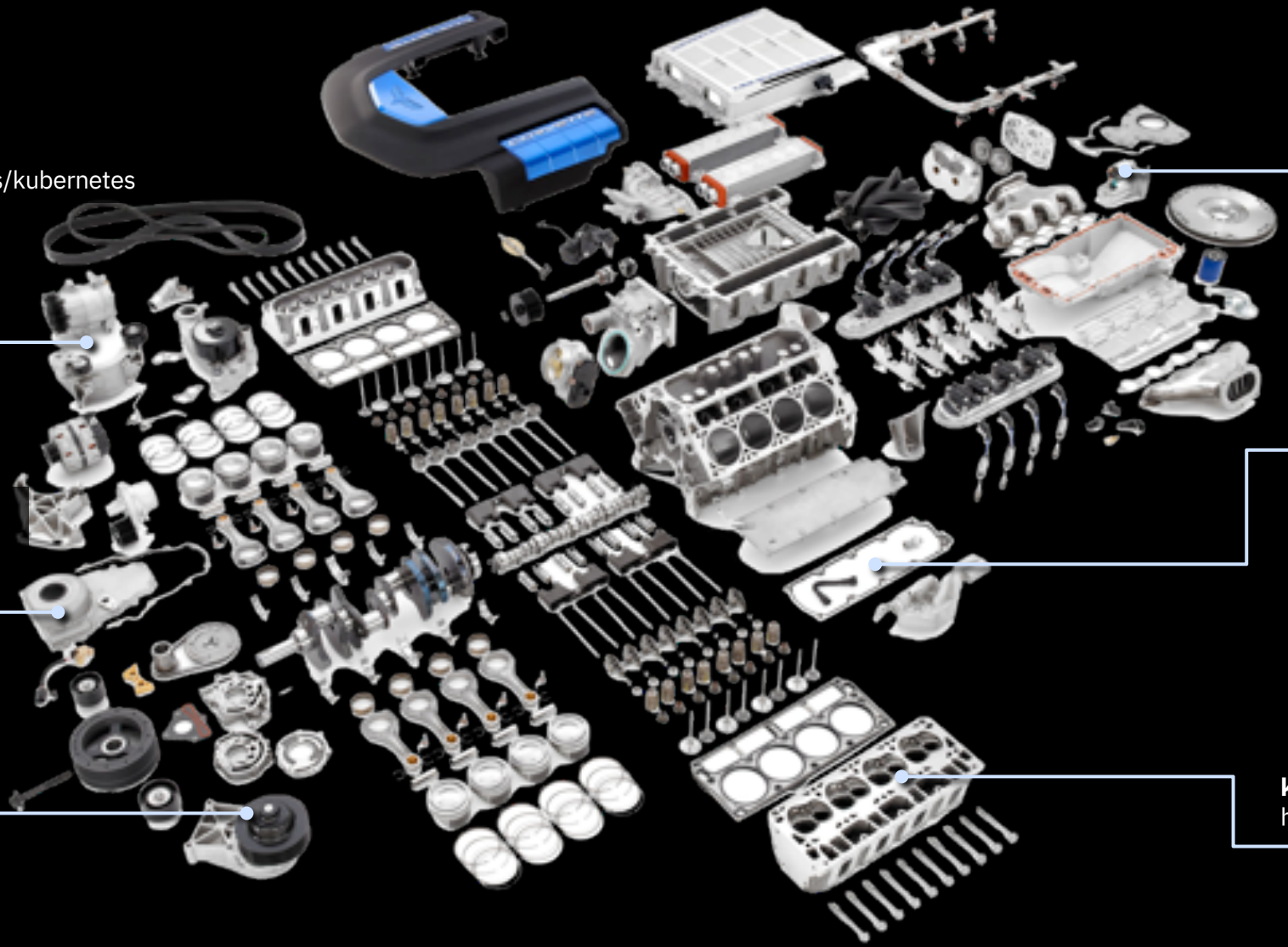
The Journey to Cloud

Elements of an enterprise grade Kubernetes Cluster

01

IBM **Cloud**

Kubernetes
Assemble vs. Operate



kubernetes
https://github.com/kubernetes/kubernetes

istio
https://github.com/istio/istio

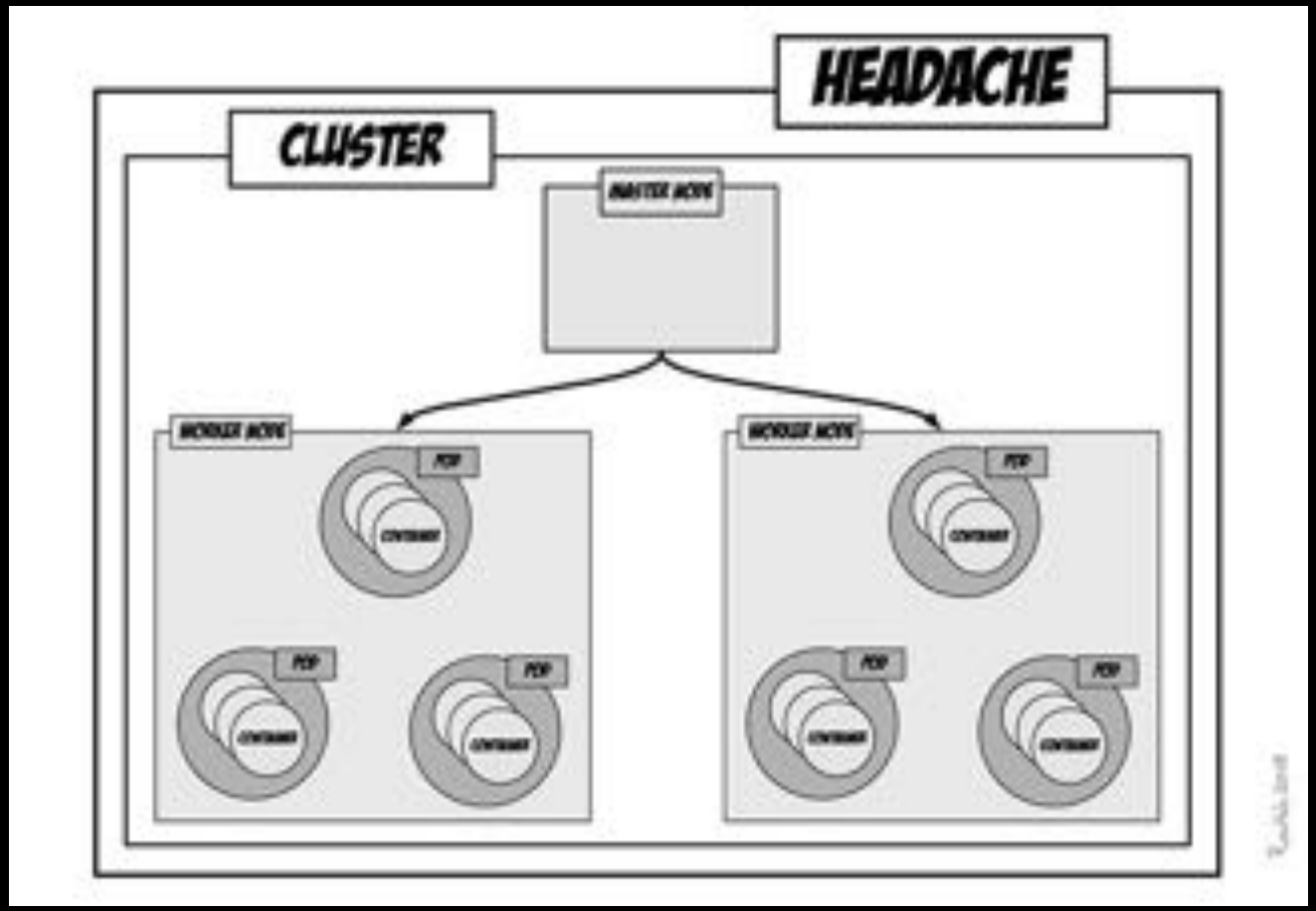
calico
https://github.com/projectcalico/

grafana
https://github.com/grafana/grafana

helm
https://helm.sh/

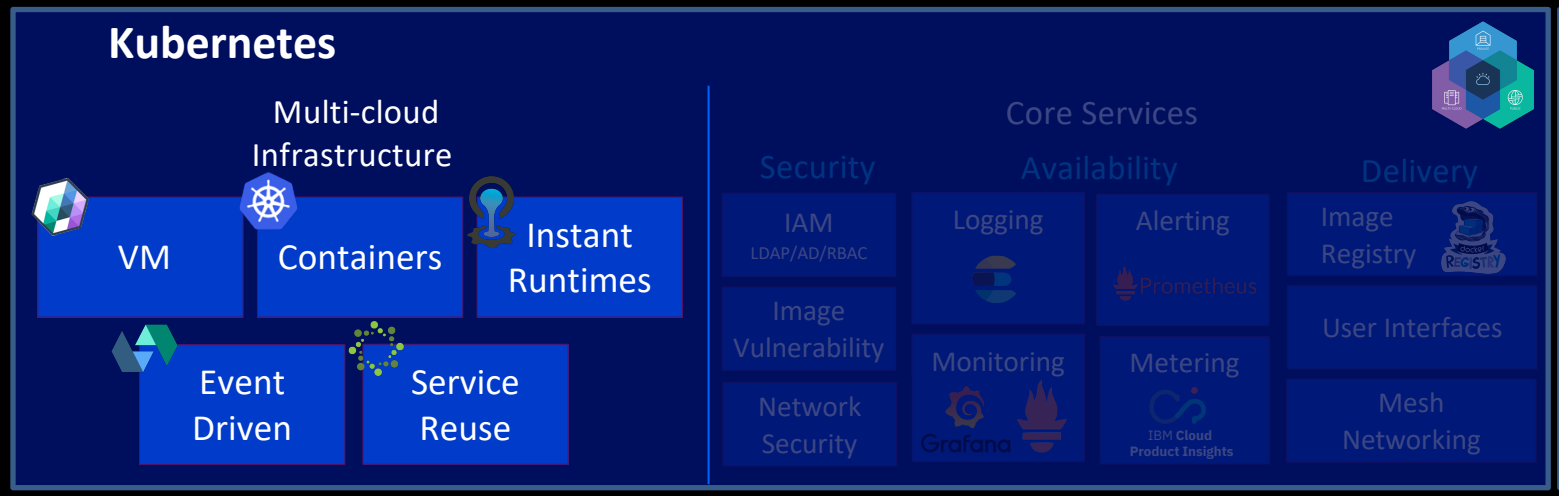
knative
https://github.com/knative/

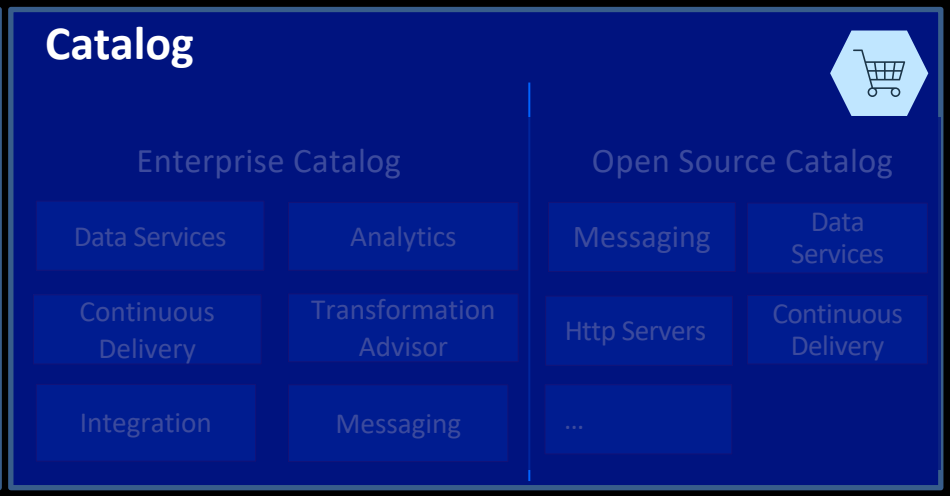
Kubernetes
Assemble vs. Operate



Source: ROELBOB - https://devops.com/kubernetes-in-the-real-world/

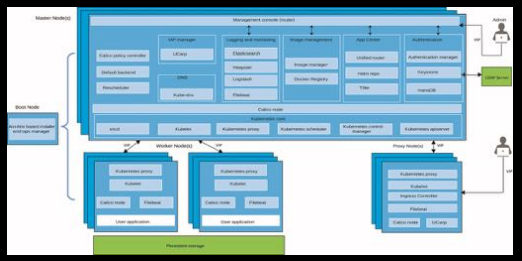
Kubernetes – Multi Cloud



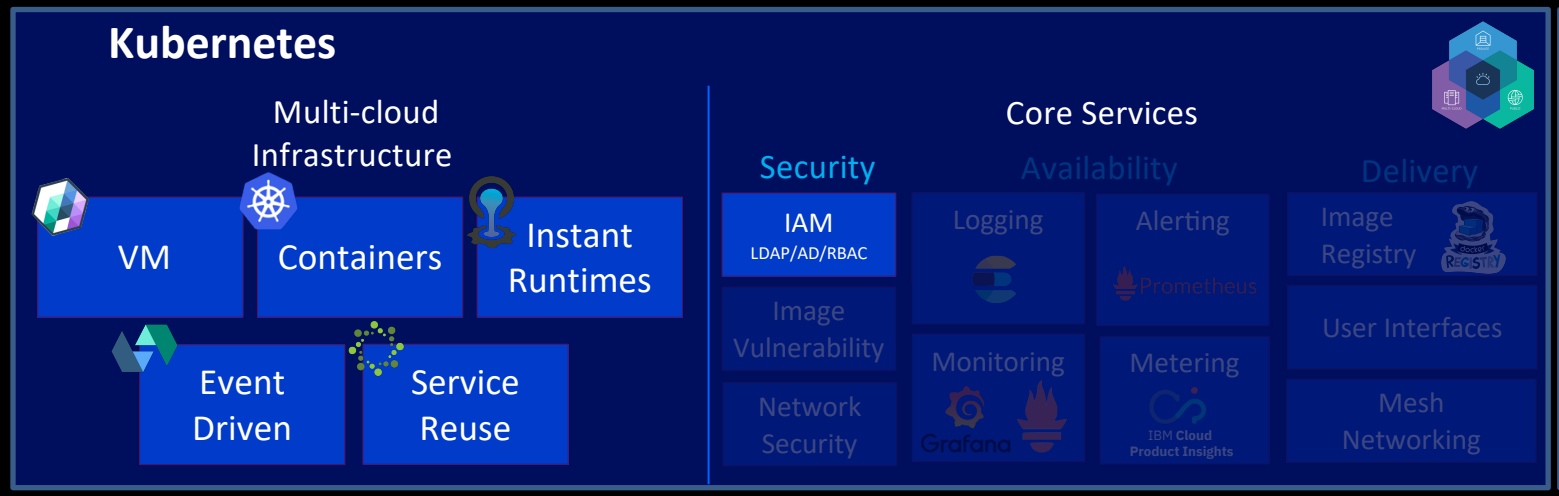


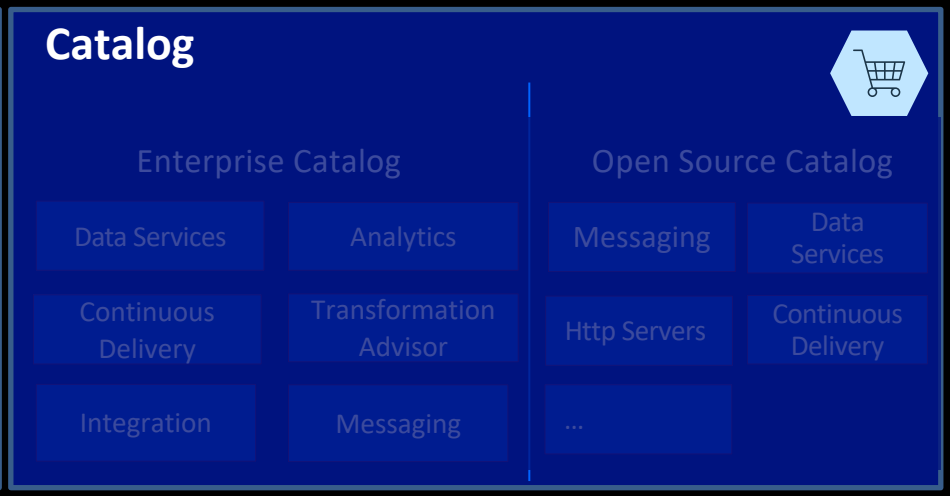
Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications

Cloud Foundry is a code-centric platform that takes the code, written in any language or framework, and runs it on any cloud.

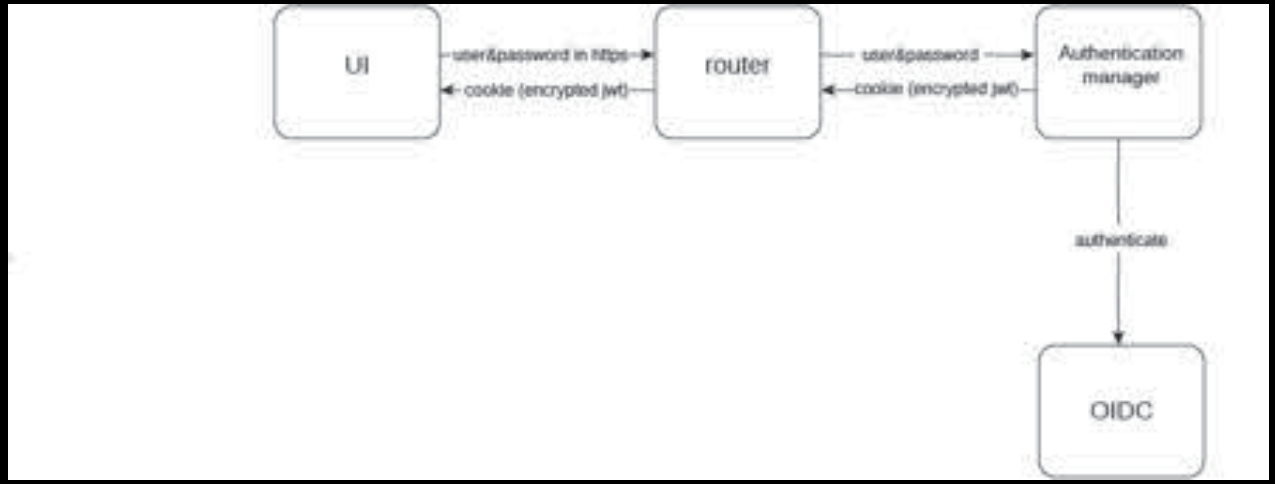


Kubernetes – Core Services - Security



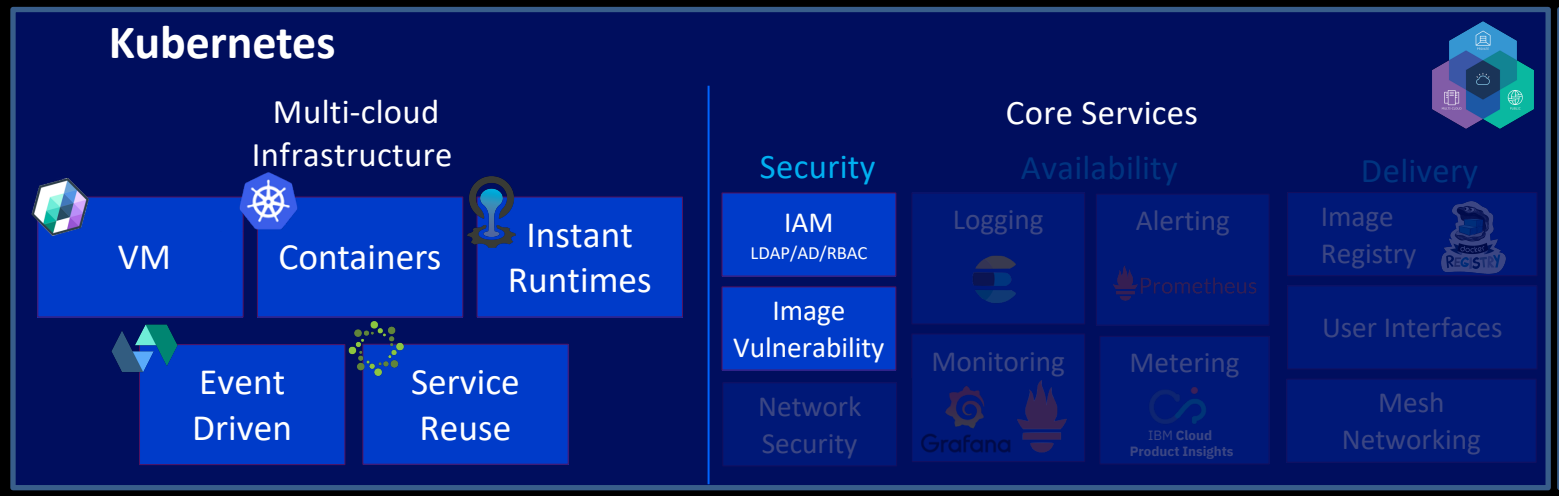


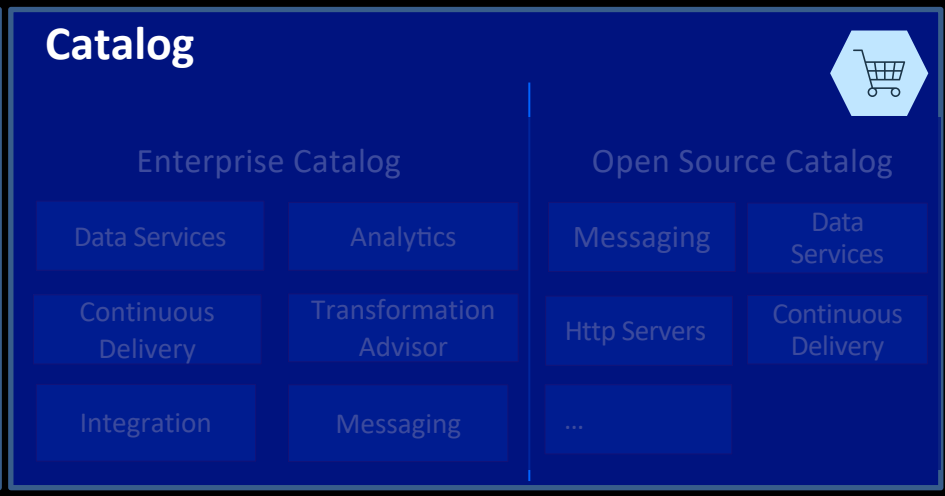
- **Authentication Manager:**
  Provides an HTTP API for managing users.
  Provides external LDAP integration



- RBAC with Teams and Users
- Can assign Roles, Namespaces, Catalog Items

Kubernetes – Core Services - Security

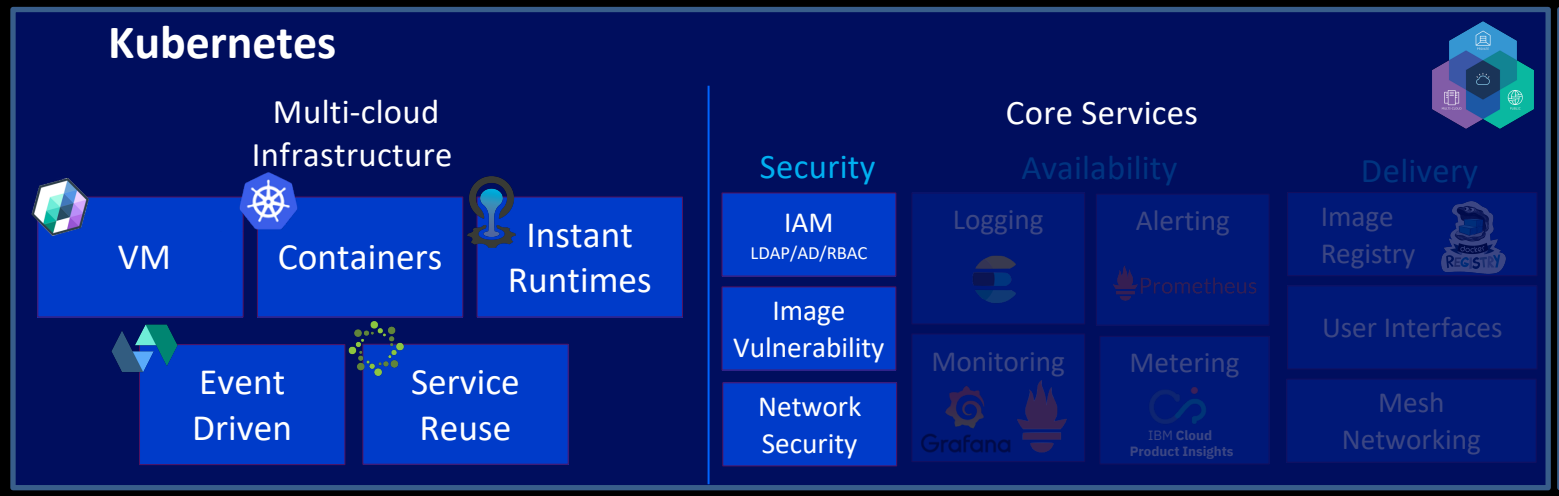


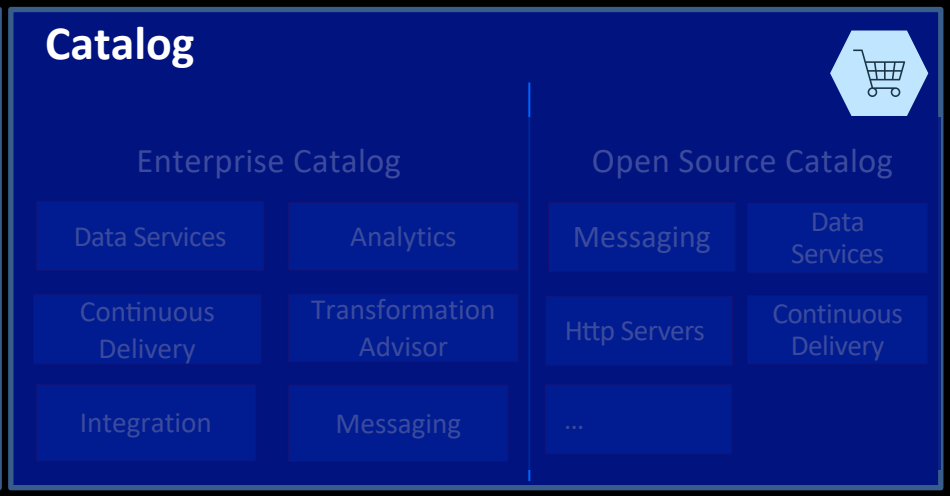


- **Vulnerability scanning**: creates vulnerability reports for containers and images.

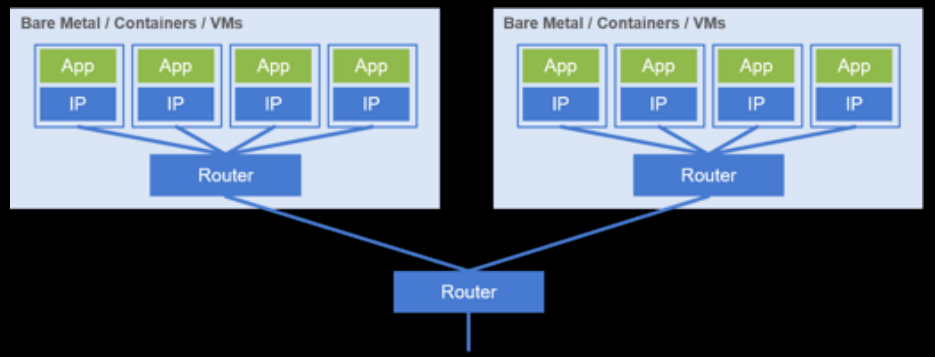
- Provides live container scanning
- Policies for access and password configurations, through policy management framework.
- CIS benchmarks, misconfiguration and drift analytics, and risk analysis by using IBM XForce Exchange.

Kubernetes – Core Services - Security





- **CNI – Common Network Interface**
- **Calico:** Virtual networking and network security for containers, VMs, and bare metal services.

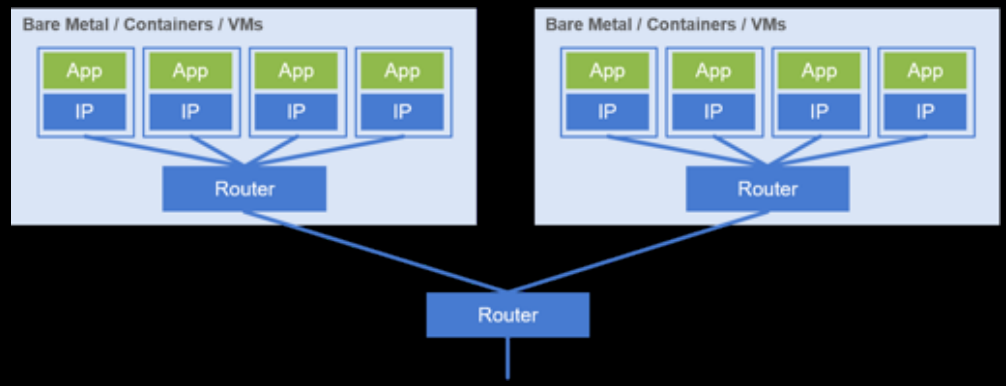


- Provides a rich set of security enforcement capabilities running on top of a highly scalable and efficient virtual network

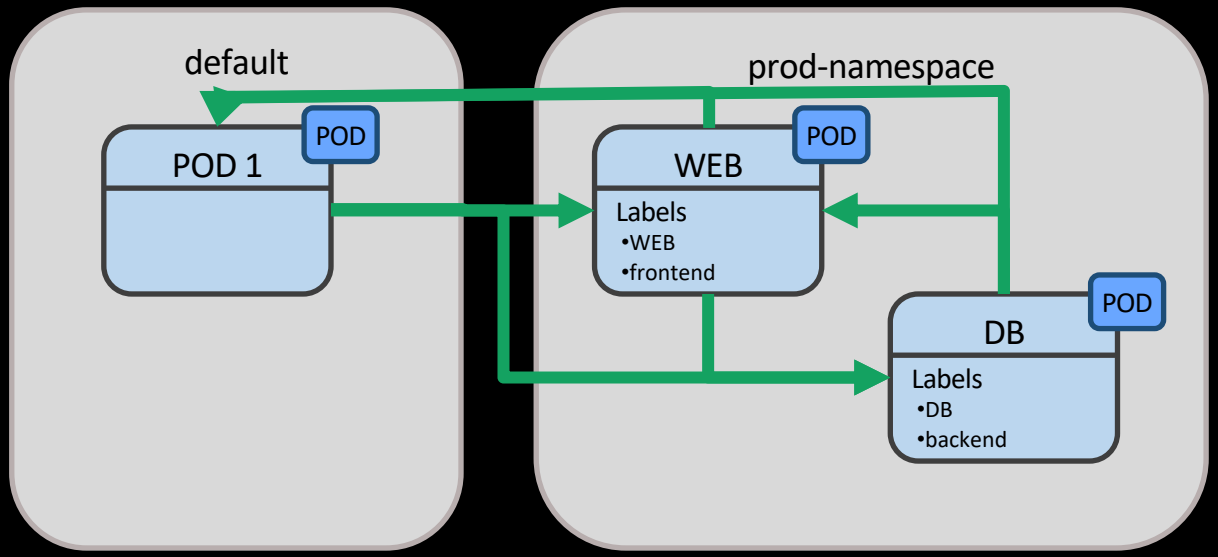
Kubernetes – Core Services - Security

Operates at **Layer 3**, which is the network layer

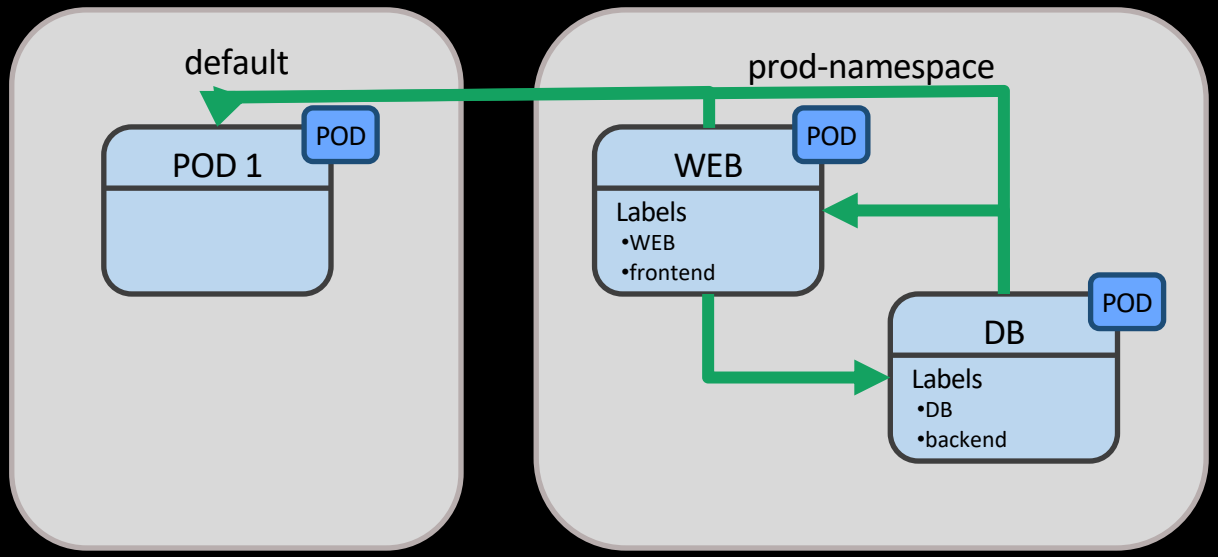
- Has the advantage of being **universal** (DNS, SQL, real-time streaming, …)
- Can extend beyond the service mesh (including to **bare metal or VM** endpoints not under the control of Kubernetes).
- Calico's policy is enforced at the host node, outside the network namespace of the guest pods.
- Based on **iptables**, which are packet filters implemented in the standard Linux kernel, it is extremely fast.



Kubernetes – Core Services - Security

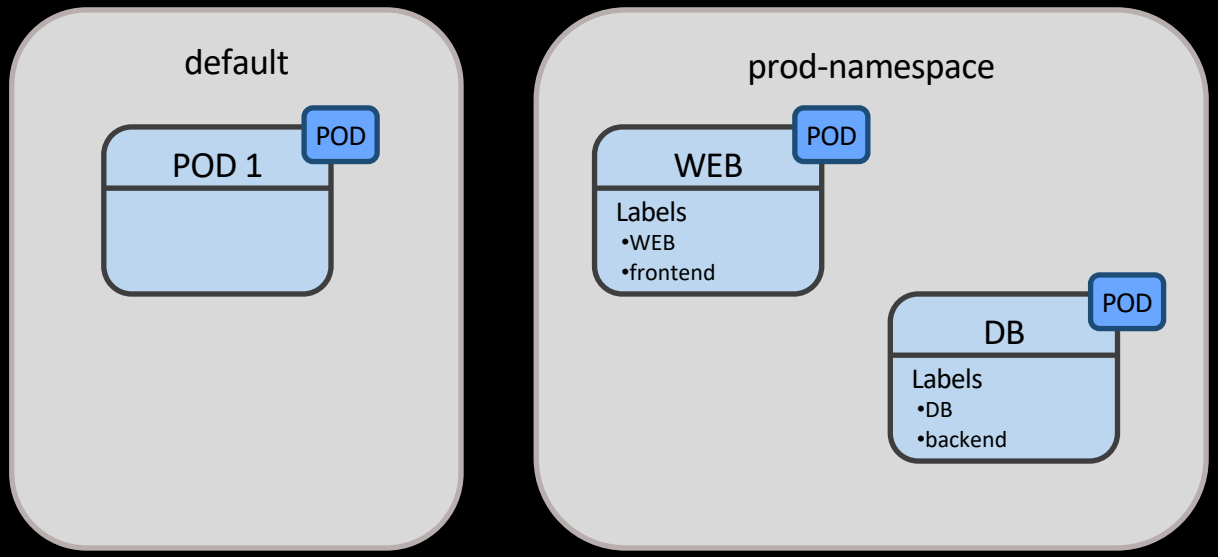


Kubernetes – Core Services - Security



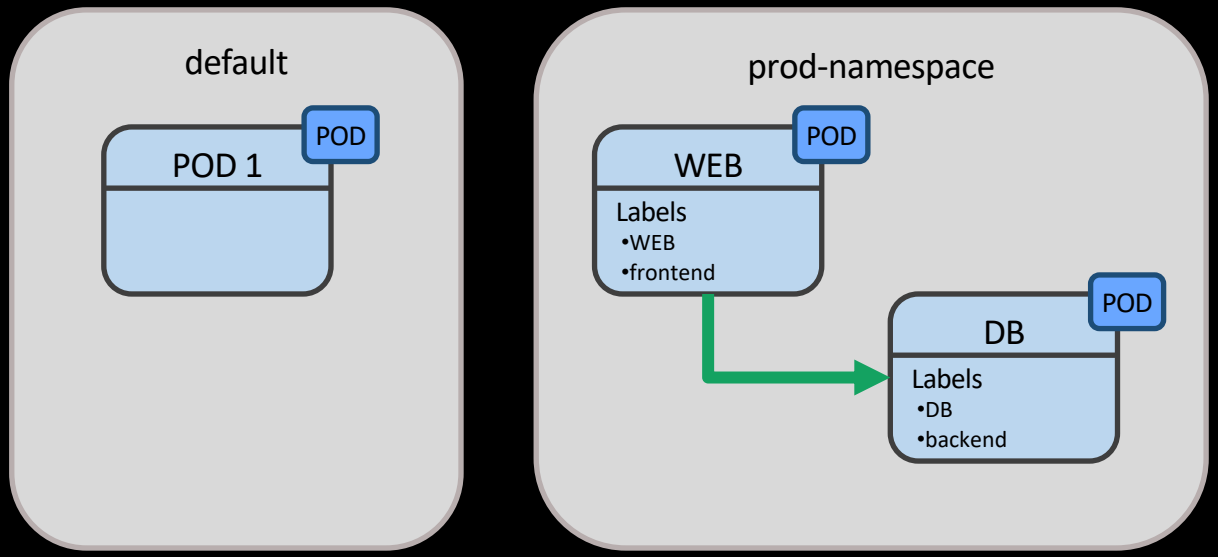

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: default-deny
  namespace: demo-namespace
spec:
  podSelector:
    matchLabels: {}
```

Kubernetes – Core Services - Security



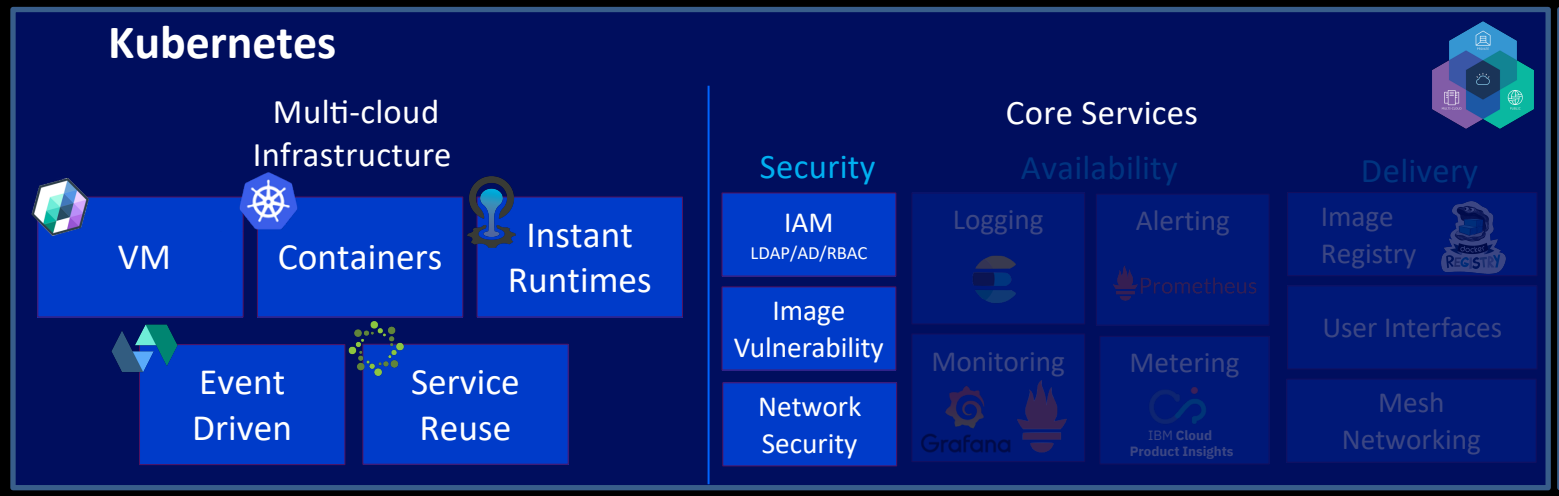

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: default-deny-all
spec:
  podSelector:
    matchLabels: {}
```

Kubernetes – Core Services - Security




```
kind: NetworkPolicy
metadata:
  name: access-frontend-backend
  namespace: prod-namespace
spec:
  podSelector:
    matchLabels:
      run: DB
  ingress:
    - from:
      - podSelector:
          matchLabels:
            run: WEB
```

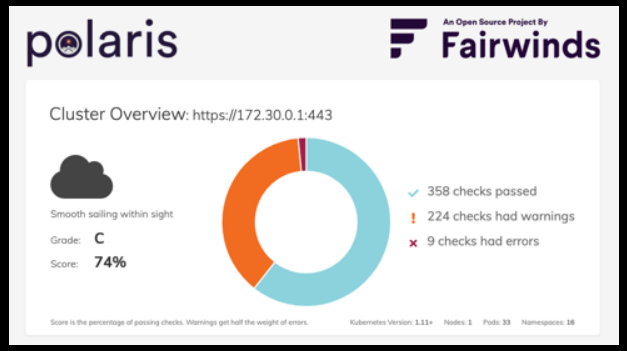
Kubernetes – Core Services - Security

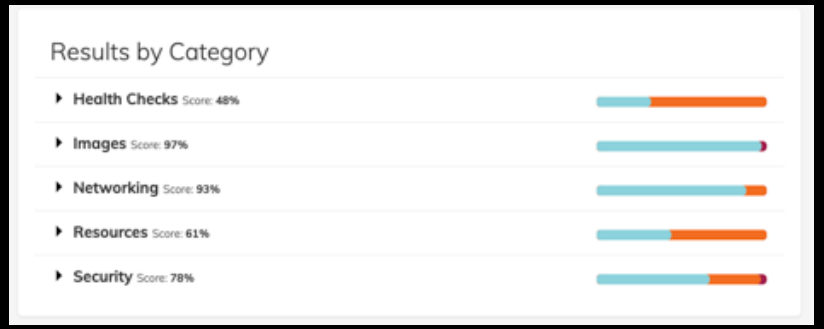


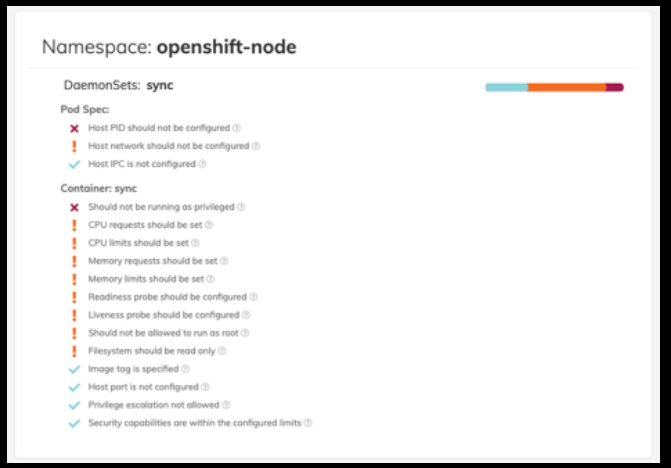
POWER TIP:

Polaris – Validation of best practices in your Kubernetes clusters

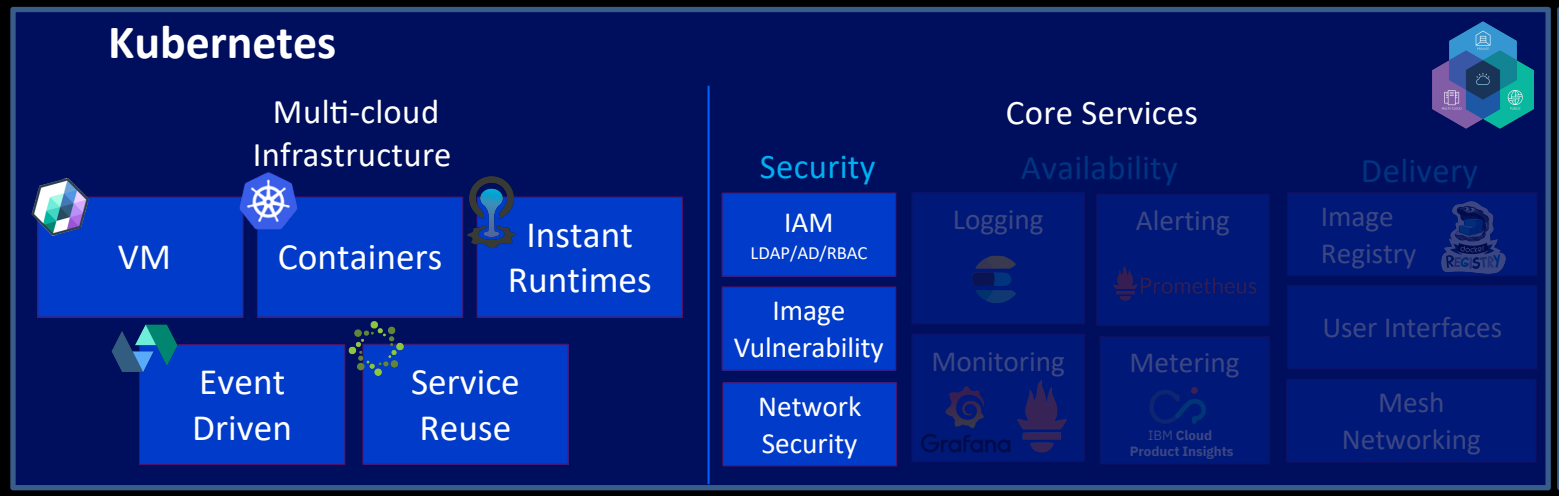
https://github.com/FairwindsOps/polaris

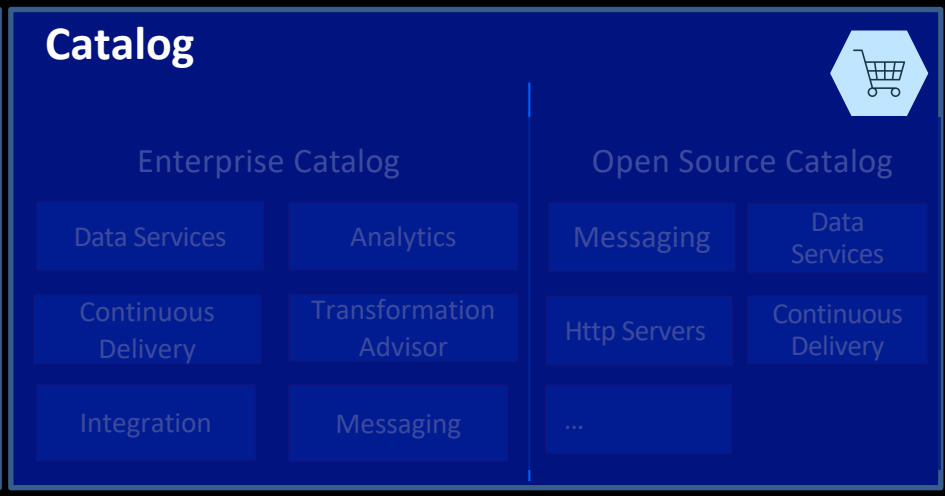






Kubernetes – Core Services - Security





POWER TIP:

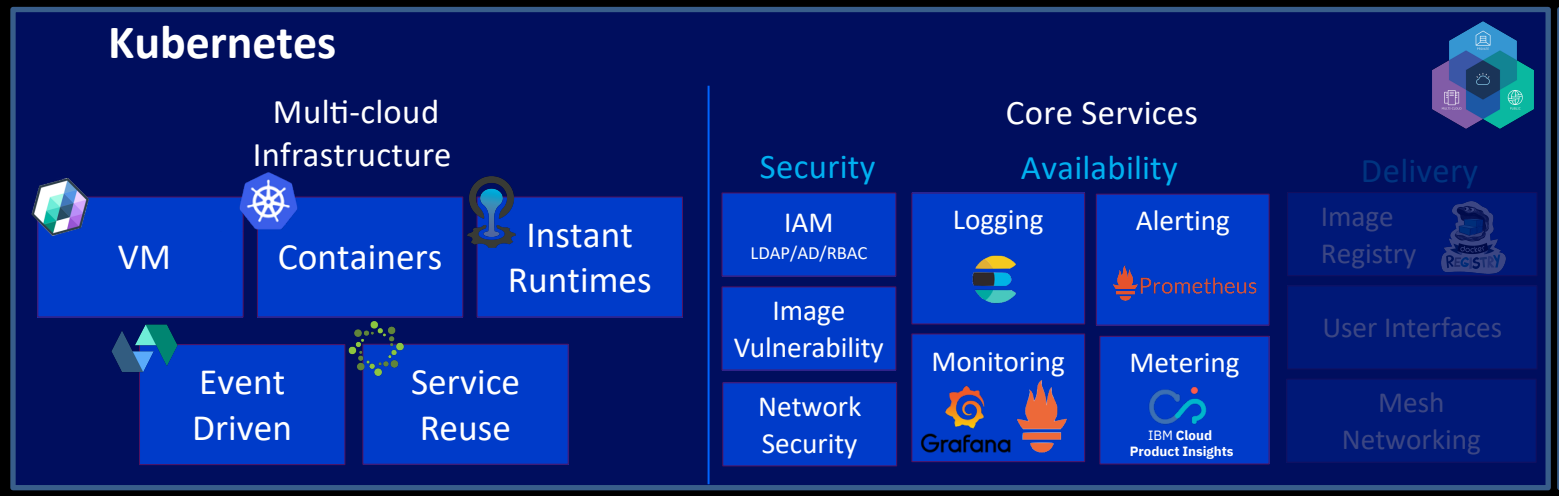
RBAC-Lookup – Easily find roles and cluster roles attached to any user

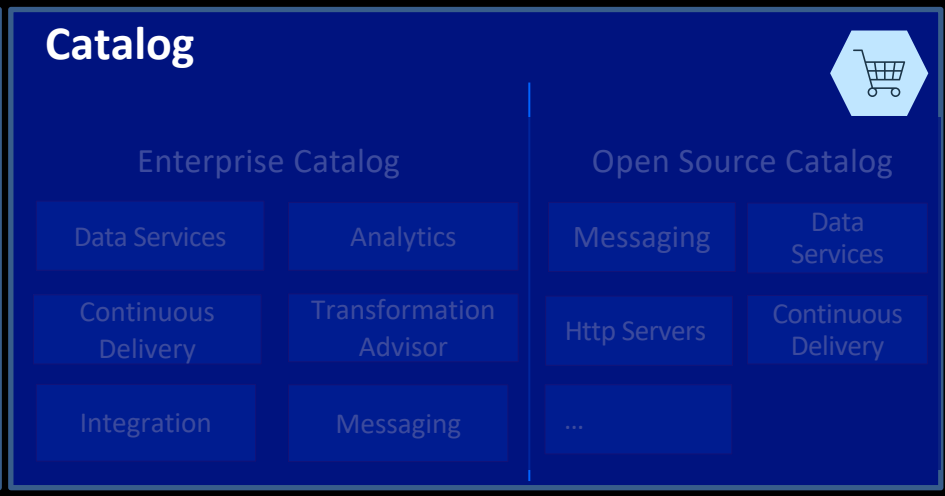
https://github.com/FairwindsOps/rbac-lookup

```
~ rbac-lookup root
SUBJECT   SCOPE          ROLE
root      cluster-wide   ClusterRole/cluster-admin
```

```
~ rbac-lookup openshift
SUBJECT                                                 SCOPE                                   ROLE
system:serviceaccounts:openshift                        openshift                               ClusterRole/system:image-puller
system:serviceaccounts:openshift-console                openshift-console                       ClusterRole/system:image-puller
system:serviceaccounts:openshift-console                openshift-console                       ClusterRole/system:image-puller
system:serviceaccounts:openshift-infra                  openshift-infra                         ClusterRole/system:image-puller
system:serviceaccounts:openshift-logging                openshift-logging                       ClusterRole/system:image-puller
system:serviceaccounts:openshift-logging                openshift-logging                       ClusterRole/system:image-puller
system:serviceaccounts:openshift-metrics-server         openshift-metrics-server                ClusterRole/system:image-puller
system:serviceaccounts:openshift-metrics-server         openshift-metrics-server                ClusterRole/system:image-puller
system:serviceaccounts:openshift-monitoring             openshift-monitoring                    ClusterRole/system:image-puller
system:serviceaccounts:openshift-monitoring             openshift-monitoring                    ClusterRole/system:image-puller
system:serviceaccounts:openshift-node                   openshift-node                          ClusterRole/system:image-puller
system:serviceaccounts:openshift-sdn                    openshift-sdn                           ClusterRole/system:image-puller
system:serviceaccounts:openshift-sdn                    openshift-sdn                           ClusterRole/system:image-puller
system:serviceaccounts:openshift-template-service-broker openshift-template-service-broker     ClusterRole/system:image-puller
system:serviceaccounts:openshift-template-service-broker openshift-template-service-broker     ClusterRole/system:image-puller
system:serviceaccounts:openshift-web-console            openshift-web-console                   ClusterRole/system:image-puller
system:serviceaccounts:openshift-web-console            openshift-web-console                   ClusterRole/system:image-puller
```

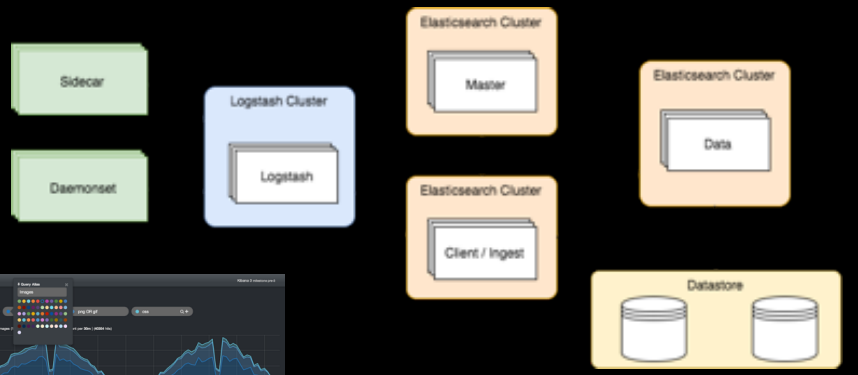
Kubernetes – Core Services - Availability



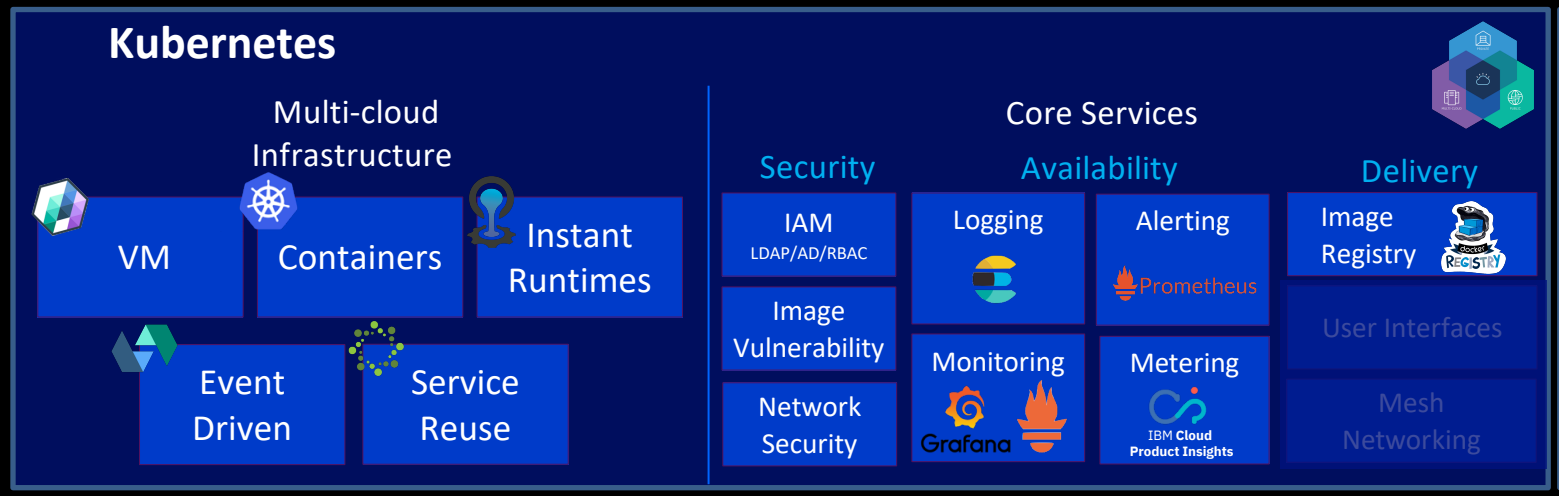


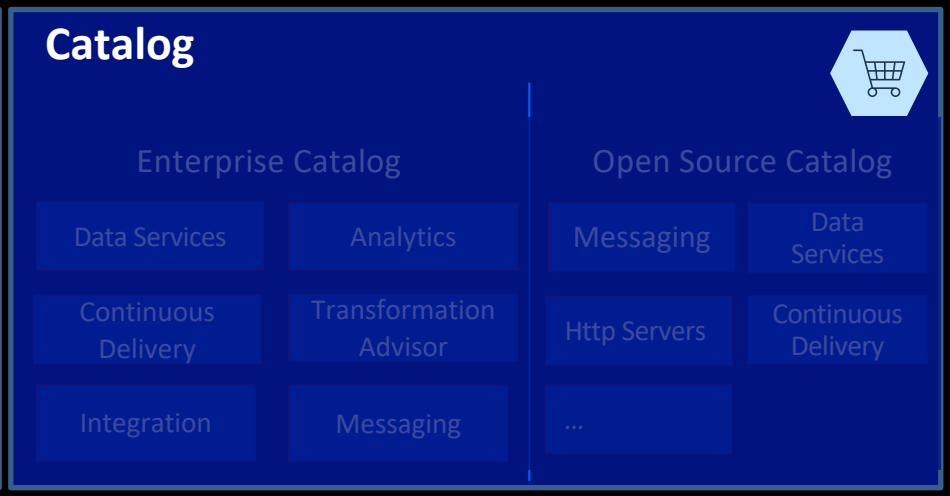
- **Monitoring/Alerting**: Open-source systems monitoring and alerting with general purpose dashboard and graph composer, which runs as a web application.

- **Logging (ELK):** The easiest and most embraced open-source logging solution for containerized applications

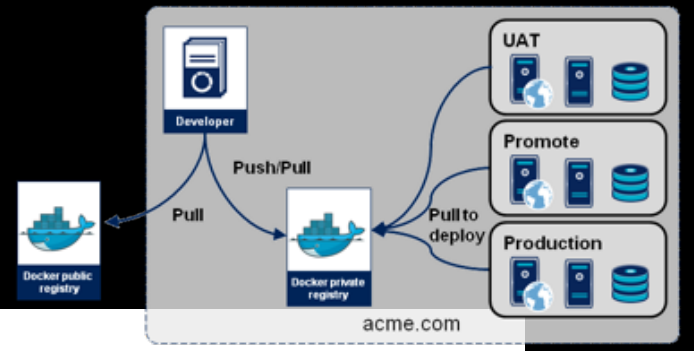


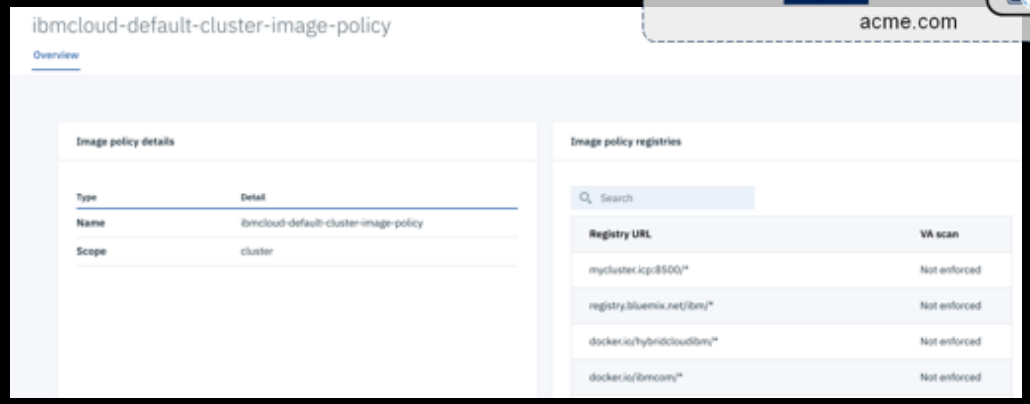
Kubernetes – Core Services - Delivery



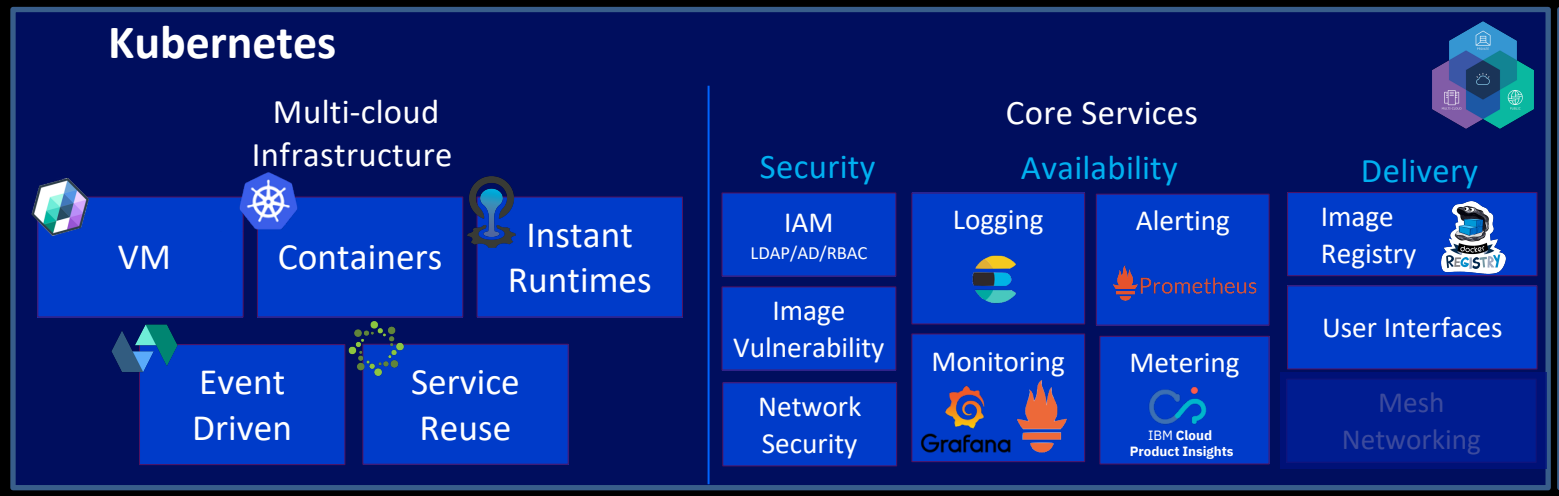


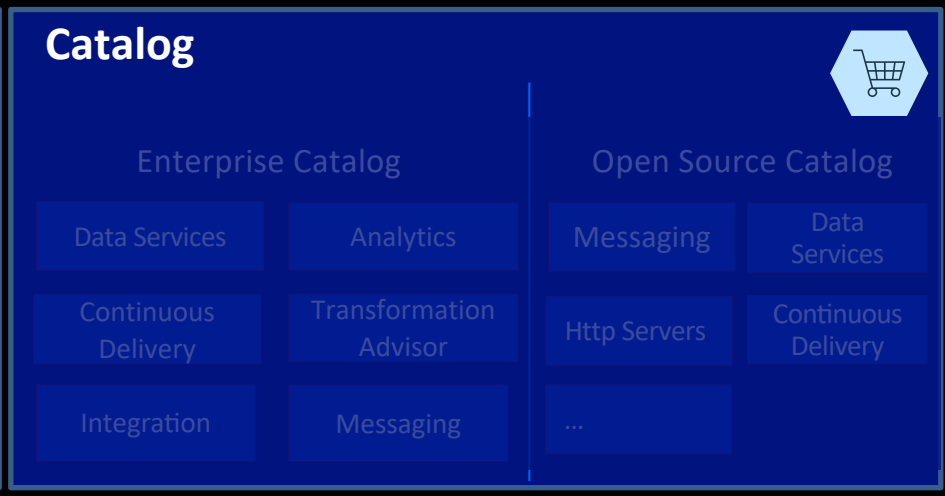
- **Private Docker Registry:** Create a Docker image and push it to a private registry before deploying it to Kubernetes
- Images can be made visible by namespace or globally via Authentication Manager/RBAC
- Image policies – control where images come from





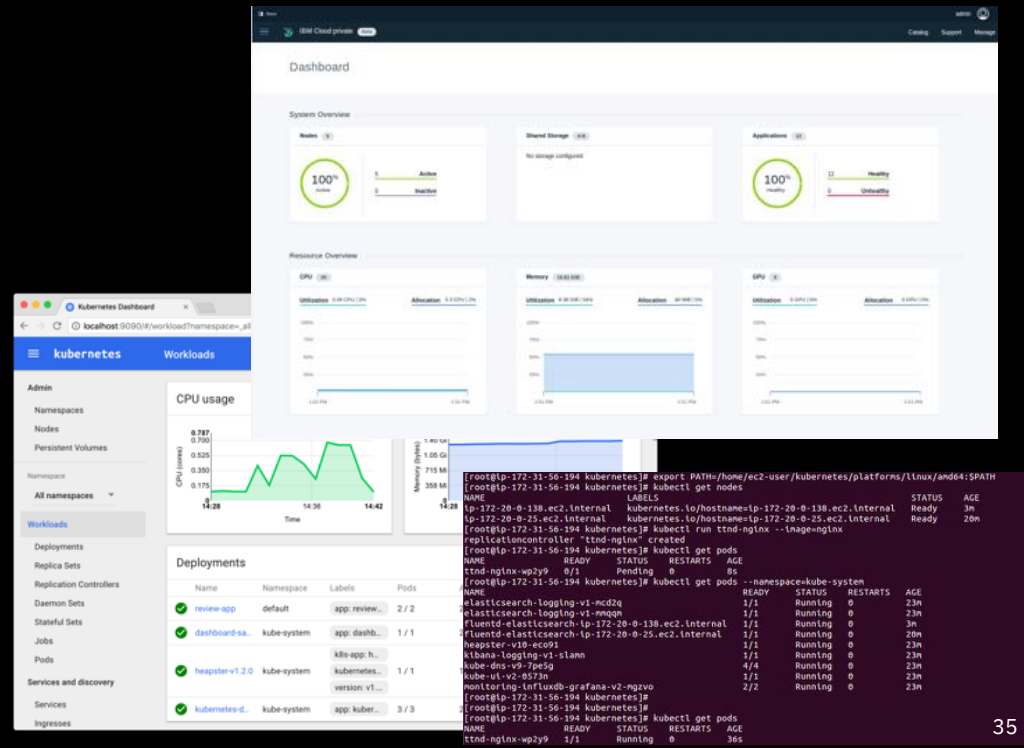
Kubernetes – Core Services - Delivery



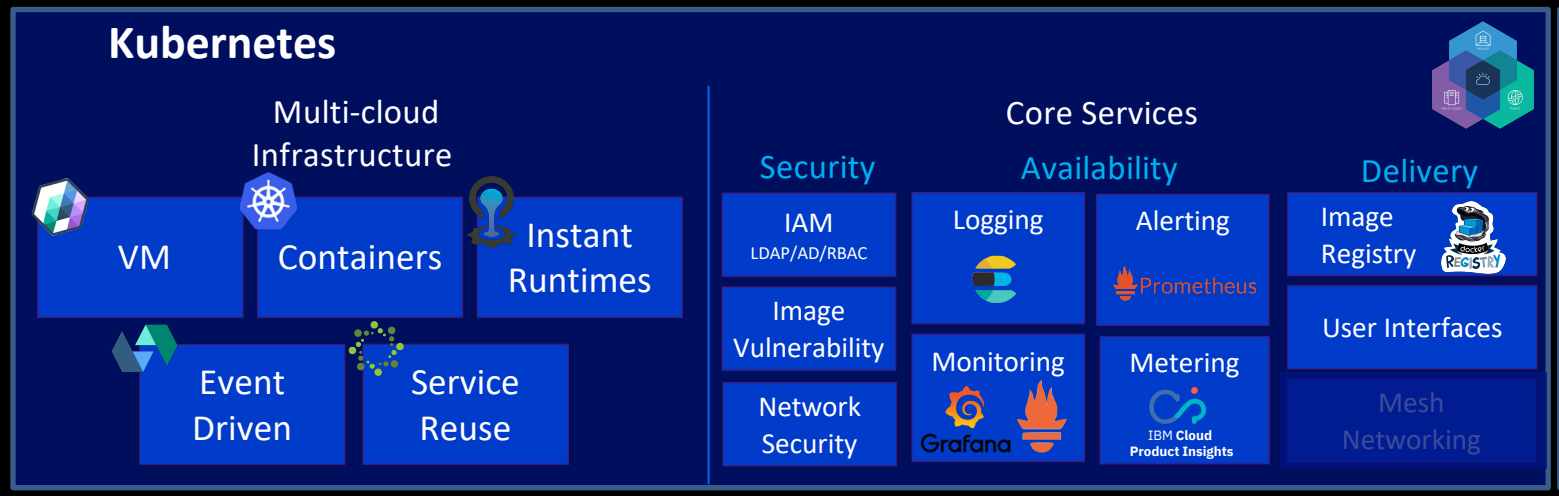


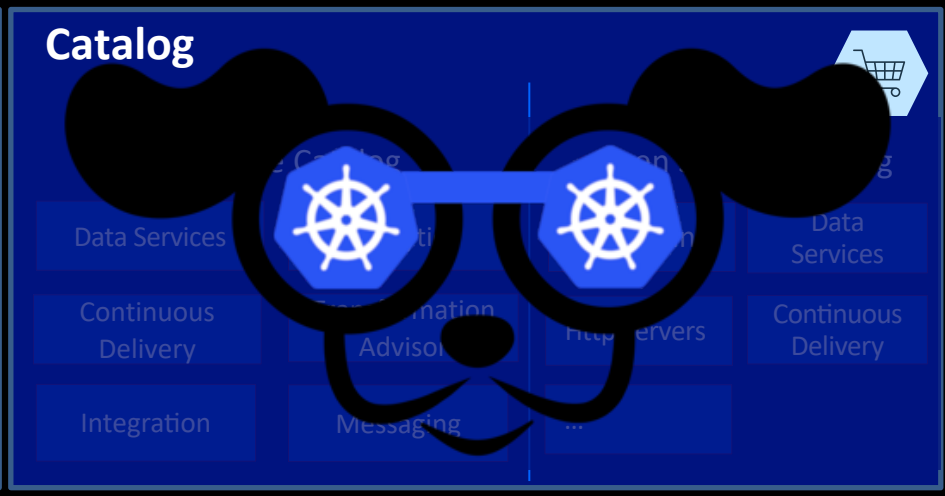
- **User Interfaces:**

  Cluster Management Console: Use to manage, monitor, and troubleshoot your applications and cluster from a single, centralized, and secure management console.
  K8s Web UI: Can use to deploy containerized applications to a Kubernetes cluster, troubleshoot your containerized application, and manage the cluster itself along with its attendant resources.
  kubectl: A command-line interface for running commands against Kubernetes clusters.



Kubernetes – Core Services - Delivery

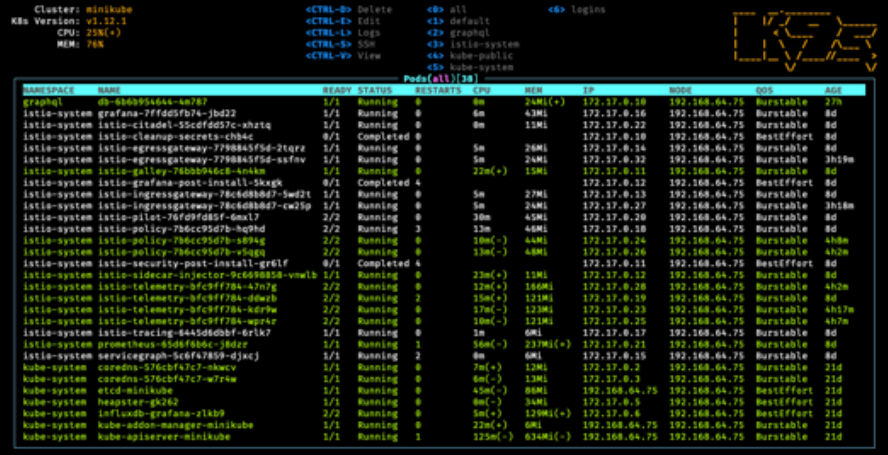




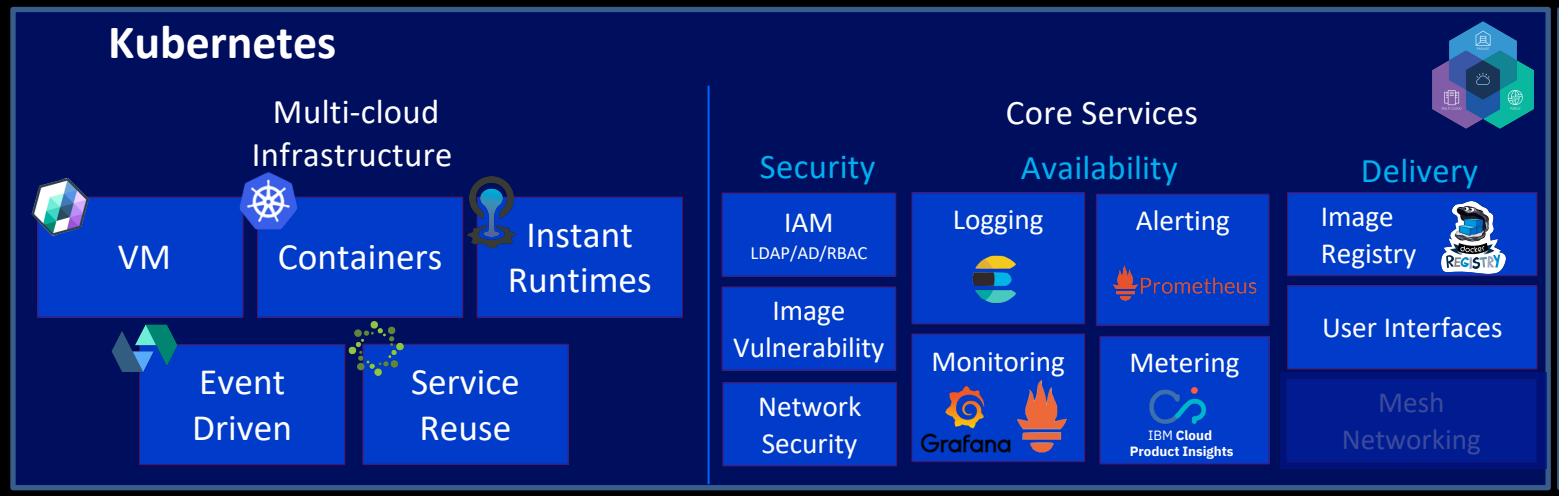
POWER TIP:

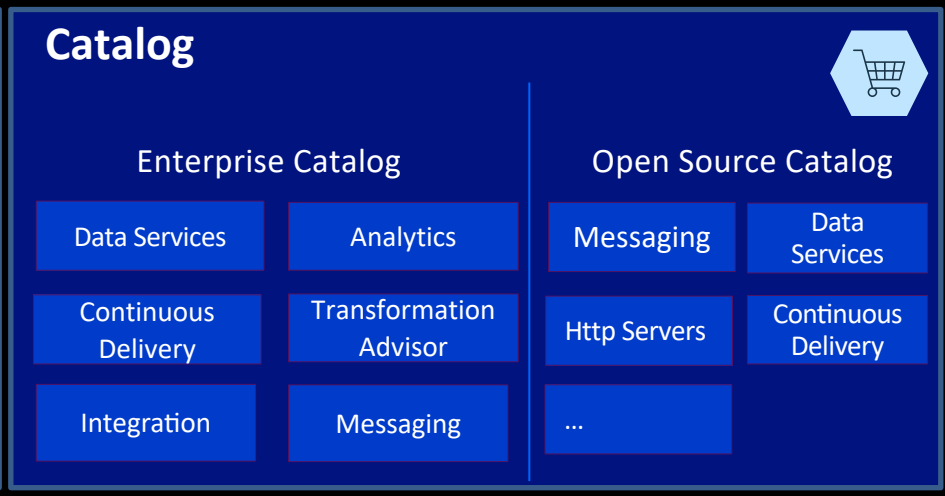
K9s – The Norton Commander for K8s

https://k9ss.io



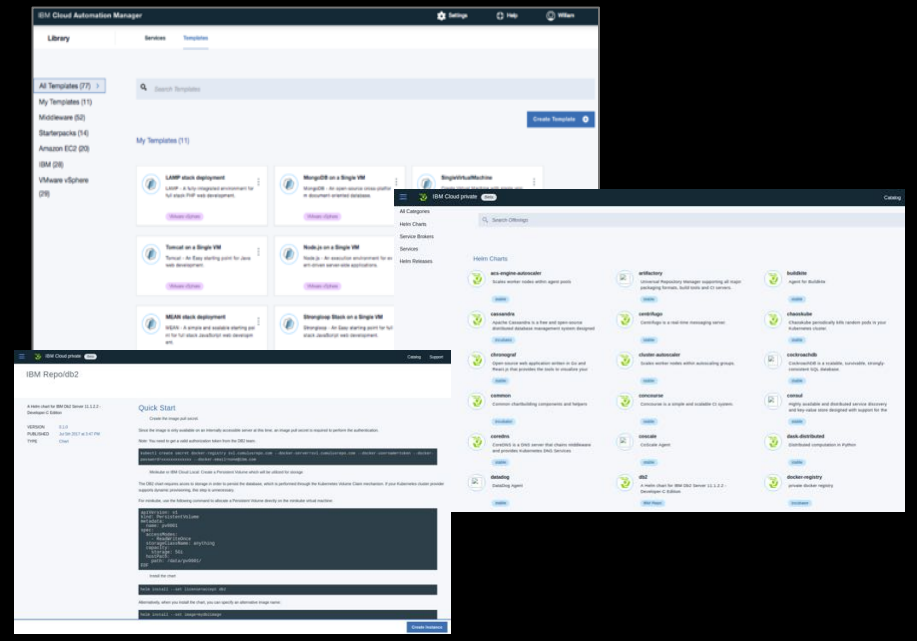
Kubernetes – Catalog – Helm & Operators



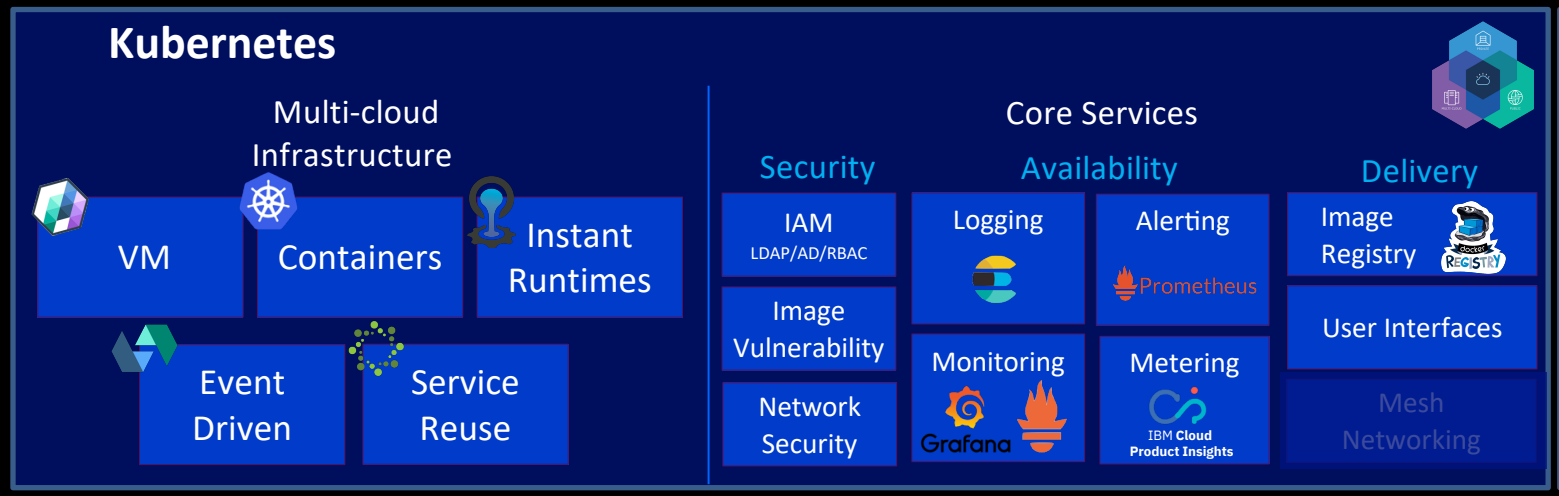


Provide a centralized location from which you can install packages in your cluster.

- Catalog of Open source content
- Leverage pre-build curated Helm Chart, Terraform Configurations and Chef scripts
- Reusable and customizable
- Day 2 Operations



Kubernetes – Storage

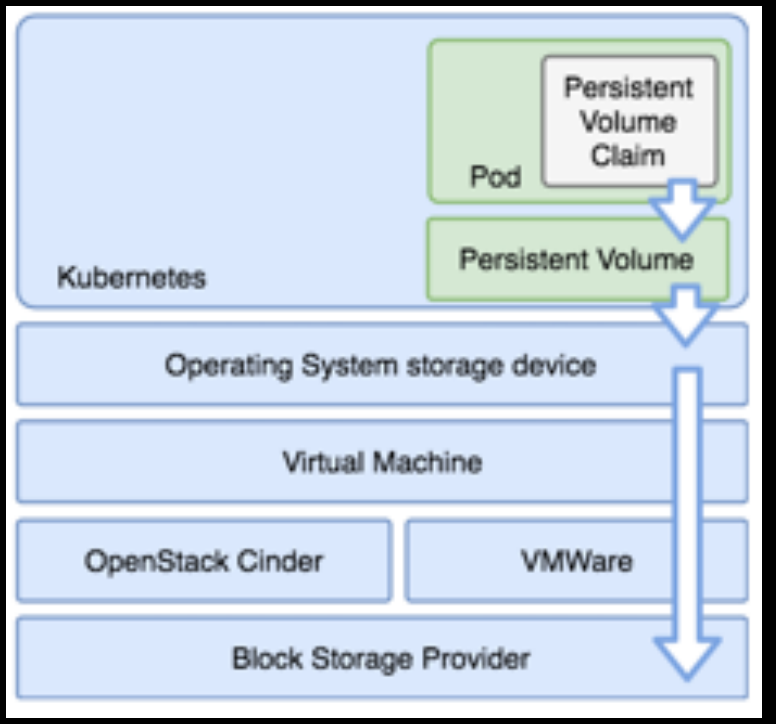


Kubernetes Persistent Storage Support: HostPath, NFS, GlusterFS, vSphereVolume

Note: Reclaim policy and access modes and behaviors can vary

Access Modes:

- ReadWriteOnce – the volume can be mounted as read-write by a single node
- ReadOnlyMany – the volume can be mounted read-only by many nodes
- ReadWriteMany – the volume can be mounted as read-write by many nodes



Kubernetes – Common Storage Interface

- GlusterFS
- Rook/Ceph
- OpenEBS
- NFS

GlusterFS

Gluster is a scalable, distributed network filesystem. Using common off-the-shelf hardware, you can create large, distributed storage solutions for media streaming, data analysis, and other data- and bandwidth-intensive tasks. Gluster is free.

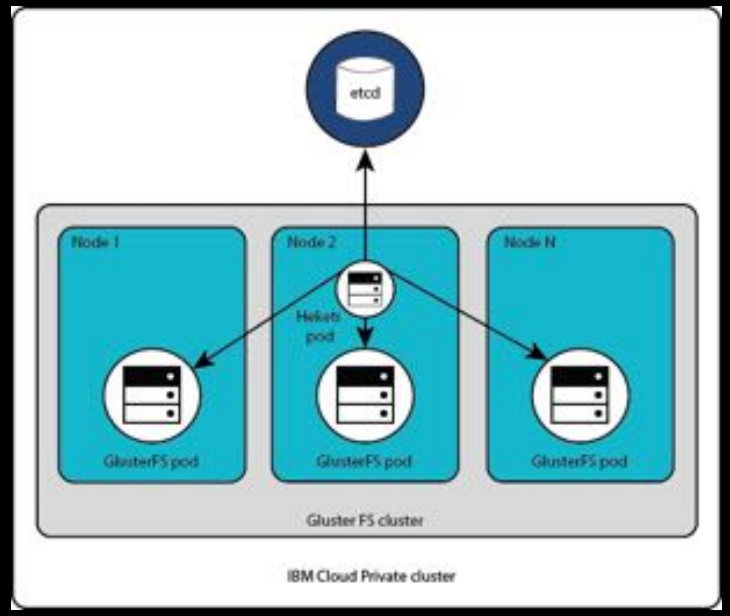
Physical install

- Format and mount the bricks
- Installing GlusterFS
- Configure the trusted pool
- Set up a GlusterFS volume
- Install Heketi & Topology (for dynamic provisioning)
- Create K8s Storage Class

Or container based

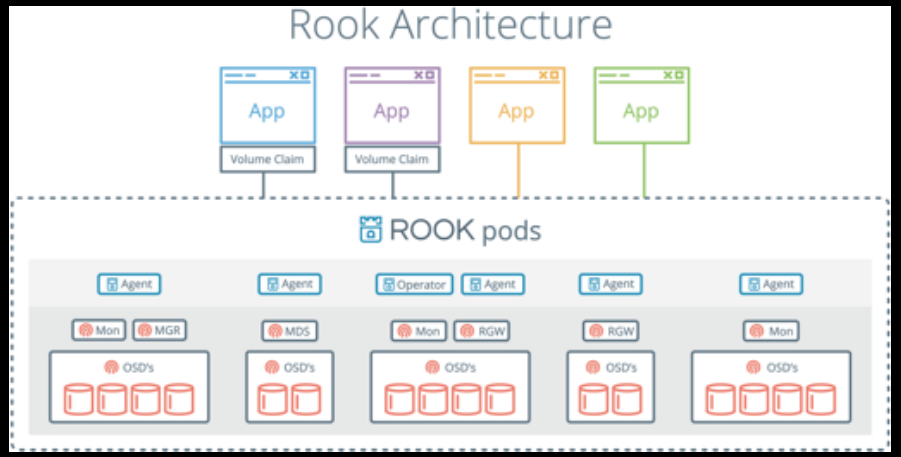
- https://hub.docker.com/r/glusterfs/gluster_centos
- Must run as priviledged → OpenShift?

https://docs.gluster.org/en/latest/Quick-Start-Guide/Quickstart/



Rook/Ceph

Rook turns distributed storage systems into self-managing, self-scaling, self-healing storage services.
Ceph is a highly scalable distributed storage solution for **block storage**, **object storage**, and **shared file systems** with years of production deployments.



Install on K8s

- git clone https://github.com/rook/rook.git
- kubectl create -f common.yaml
- kubectl create -f operator(-openshift).yaml
- kubectl create -f cluster-test.yaml
- kubectl create -f csi/rbd/storageclass-test.yaml

cassandra	crds: Set annotations pods, deployments and so on
ceph	We were defaulting to `ext4` at first and then moved to
cockroachdb	crds: Set annotations pods, deployments and so on
edgefs	edgefs image version update in examples and docs
minio	minio: add necessary update verb in minio RBAC
nfs	NFS: Fix operator.yaml line endings
noobaa	Rook-NooBaa Design Doc
yugabytedb	yugabytedb: Documentation, user guides & examples

https://rook.io/docs/rook/v0.9/ceph-quickstart.html

OpenEBS

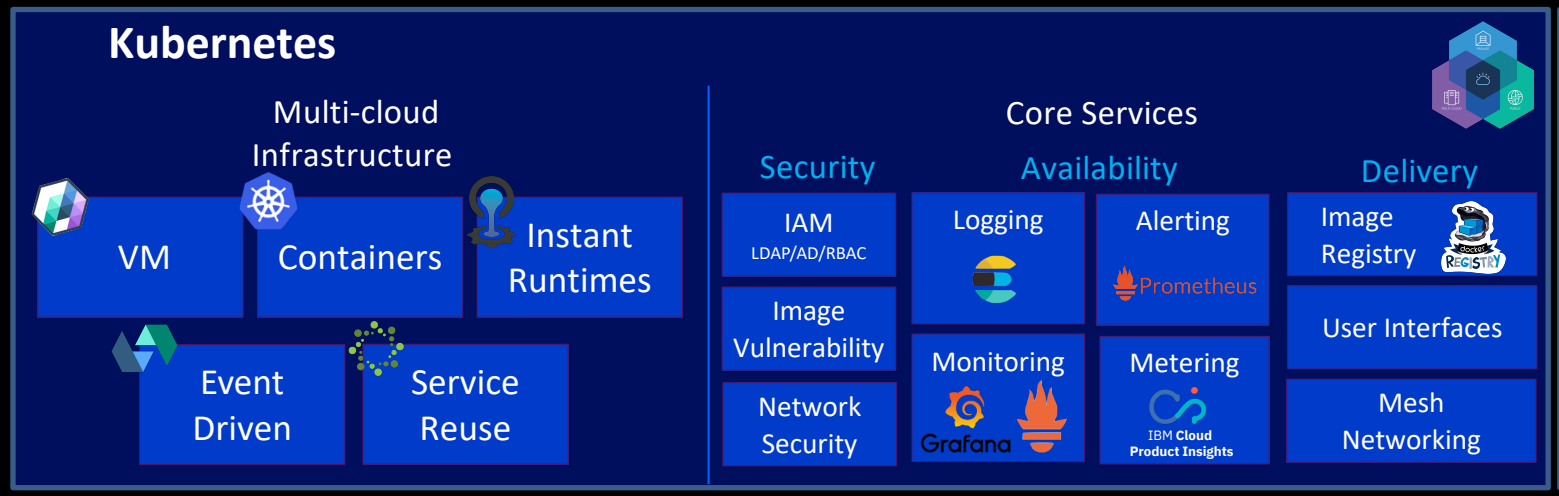
OpenEBS is truly Kubernetes native.
It adopts Container Attached Storage (CAS) approach, where each workload is provided with a dedicated storage controller. It implements granular storage policies and isolation that enable users to optimize storage for each specific workload.

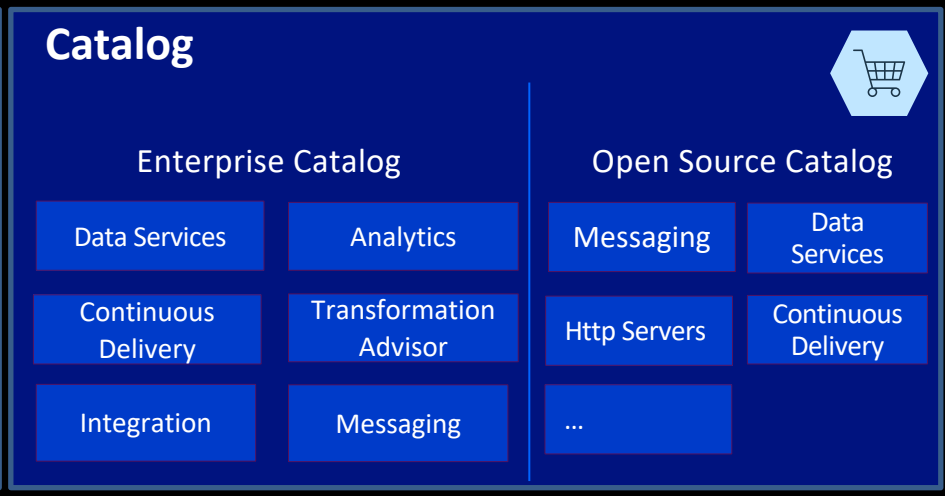
Install on K8s

- sudo apt-get install open-iscsi
- kubectl apply -f https://openebs.github.io/charts/openebs-operator.yaml
- kubectl create -f csi/rbd/storageclass-test.yaml
- kubectl apply -f openebs-storageclasses.yaml

https://docs.openebs.io/docs/next/installation.html

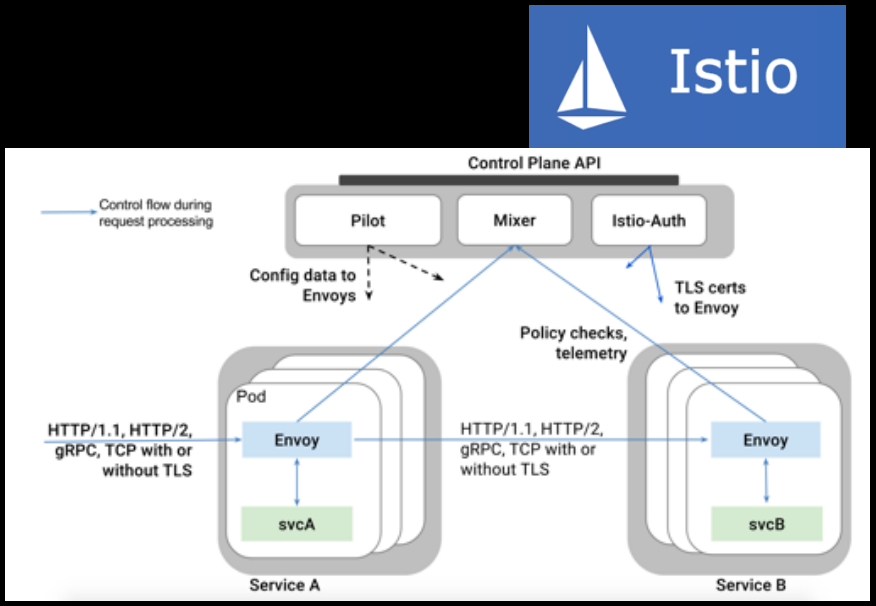
Kubernetes – Core Services - Delivery





Networking – Service Mesh - ISTIO:

- Open platform to connect, manage, and secure microservices.
- Developed by IBM, Google and Lyft
- Create a network of deployed services with load balancing, service-to-service authentication, monitoring, and more, without requiring any changes in service code.



IBM

QUESTIONS?

IBM **Cloud**

IBM

The Journey to Cloud
Mesh Networking

02

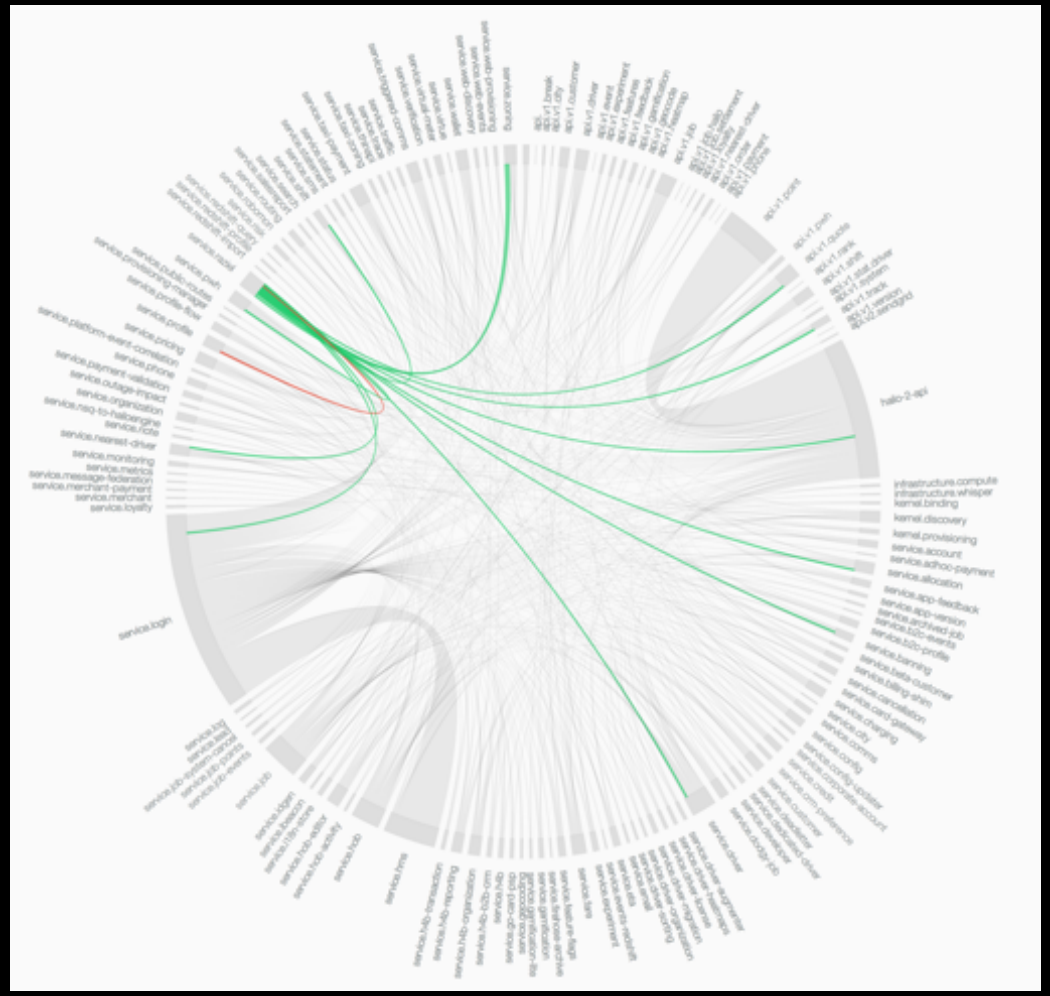
IBM **Cloud**

The trade off

Improved delivery velocity

in exchange for

increased operational complexity



Common DevOps Challenge 1

How do I **roll out** a newer version of my microservice **without down time**?

How do I **ensure traffic** continue to go to the current version before the newer version is tested and ready?

Common DevOps Challenge 2

How do I do **canary testing**?

How do I proceed to a **full rollout** after satisfactory testing of the new version?

Common DevOps Challenge 3

How do I do **A/B testing**?

- Release a new version to a subset of users in a precise way

I want to leverage crowdsourced testing. How do I **test** the new version **with a subset of users**?

I have **launched B in the dark**, but how can I keep B to myself or a small testing group?

Other common DevOps Challenges

4. Things don’t always go correctly in production… How do I **inject fault** to my microservices to prepare myself?

5. My services can only **handle certain rate**, how can I limit rate for some of my services?

Other common DevOps Challenges

6. I need to **view and monitor** what is going on with each of my services when crisis arises.

7. How can I **secure my services**.

Service Mesh

Dedicated infrastructure layer
to make
service-to-service communication
fast, safe and reliable

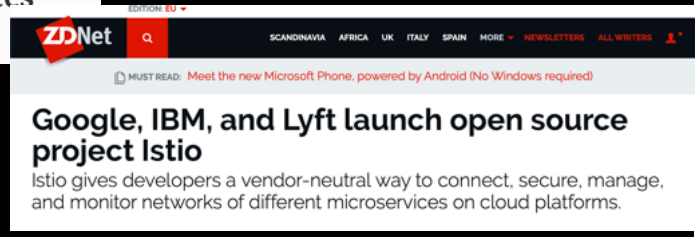
Istio

A service mesh designed to connect, manage and secure micro services


Forbes

3,859 views | May 25, 2017, 01:39am

Google, IBM And Lyft Want To Simplify Microservices Management With Istio

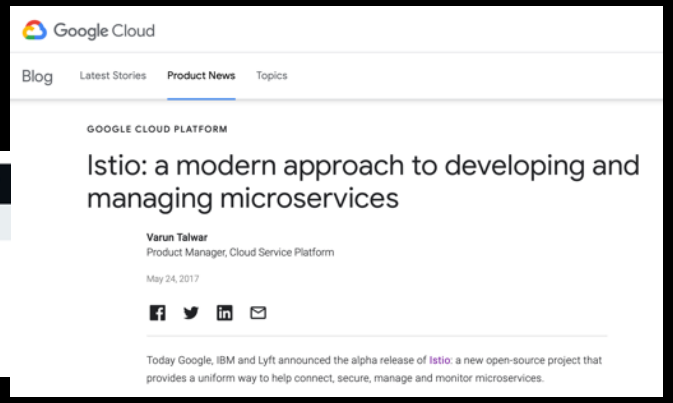

ZDNet

SCANDINAVIA AFRICA UK ITALY SPAIN MORE NEWSLETTERS ALL WRITERS

MUST READ: Meet the new Microsoft Phone, powered by Android (No Windows required)

Google, IBM, and Lyft launch open source project Istio

Istio gives developers a vendor-neutral way to connect, secure, manage, and monitor networks of different microservices on cloud platforms.


Google Cloud

Blog Latest Stories Product News Topics

GOOGLE CLOUD PLATFORM

Istio: a modern approach to developing and managing microservices

Varun Talwar
Product Manager, Cloud Service Platform

May 24, 2017

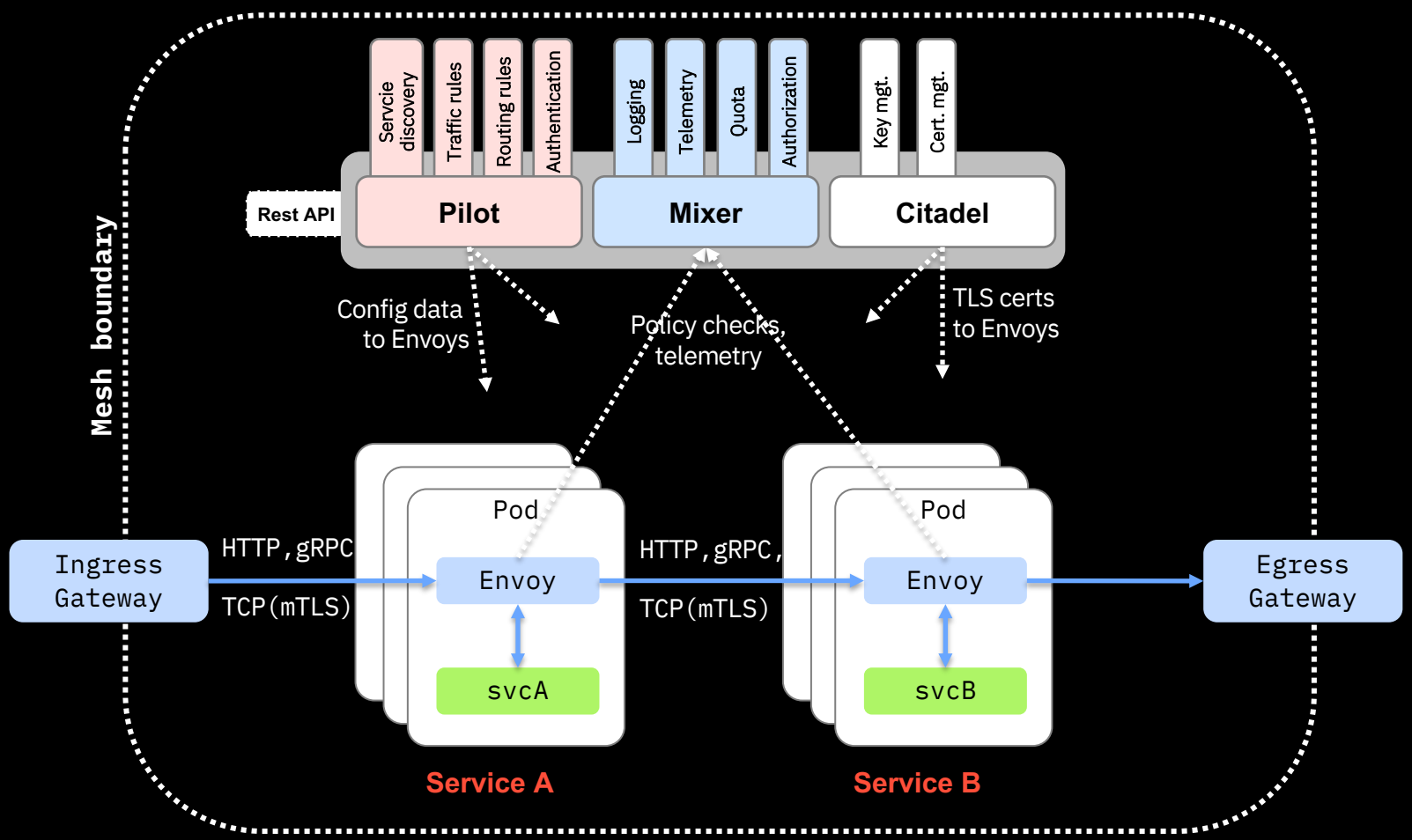
Today Google, IBM and Lyft announced the alpha release of Istio: a new open-source project that provides a uniform way to help connect, secure, manage and monitor microservices.

Launched in May 2017 by Google, Lyft and IBM

Open Source

Zero Code Changes

ISTIO - Architecture



Istio control plane traffic. Request routing rules, resilience configuration (circuit breakers, timeouts, retries), policies (ACLs, rate limits, auth), and metrics/reports from proxies.

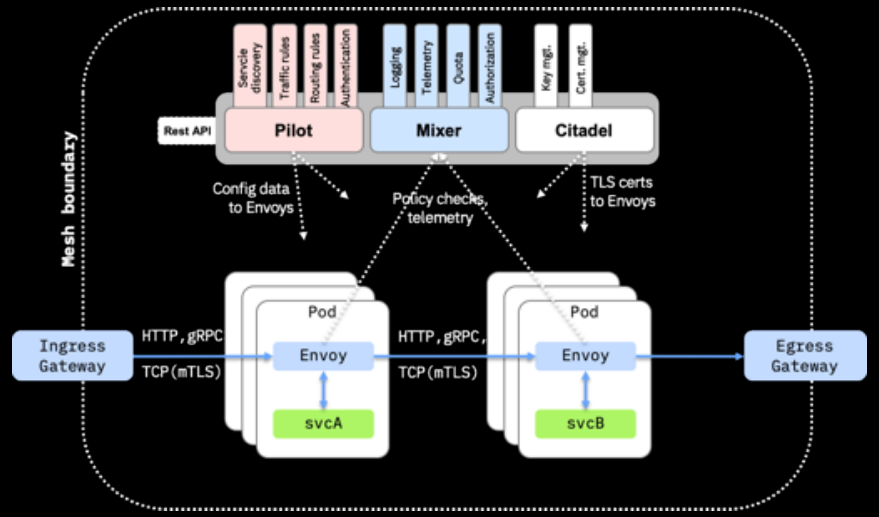
User/application traffic. HTTP/1.1, HTTP/2, gRPC, TCP with or without TLS

ISTIO - Architecture

Istio

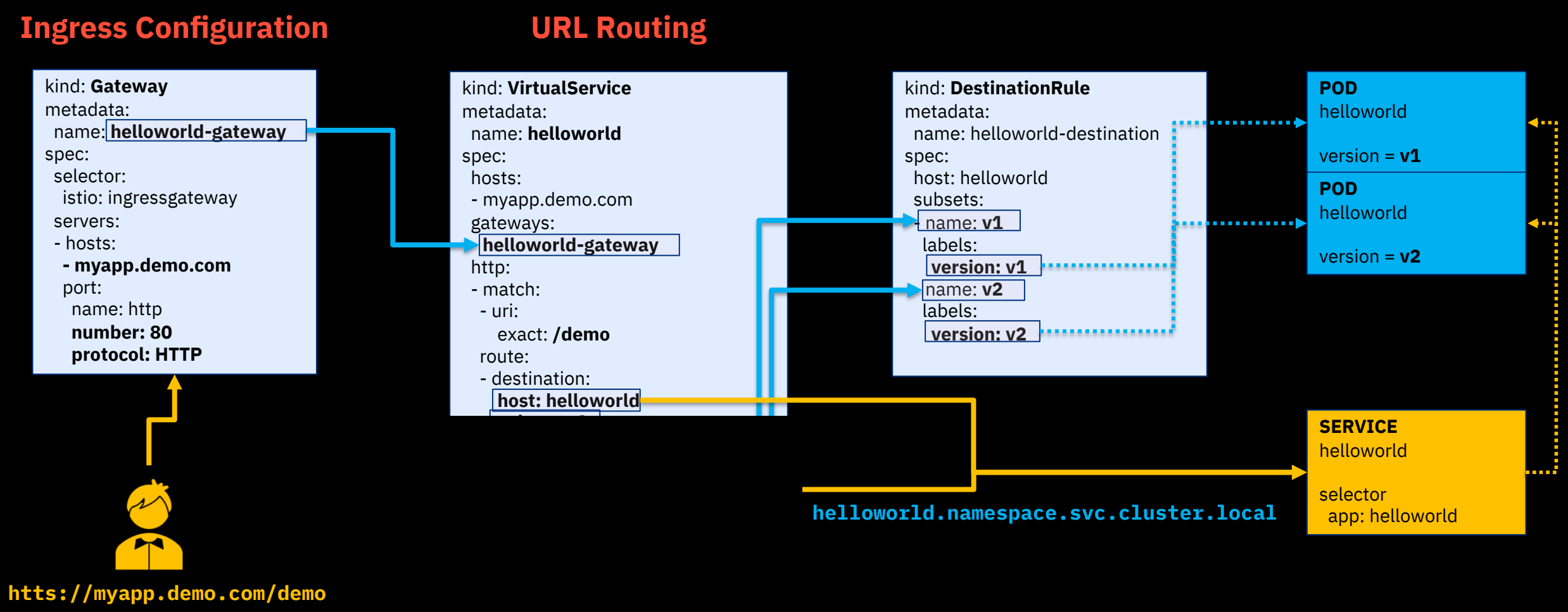
Operates at **Layer 7**

- policies can be applied based on virtual host, URL, or other HTTP headers.
- Flexibility in processing.
- Allows it to be distributed



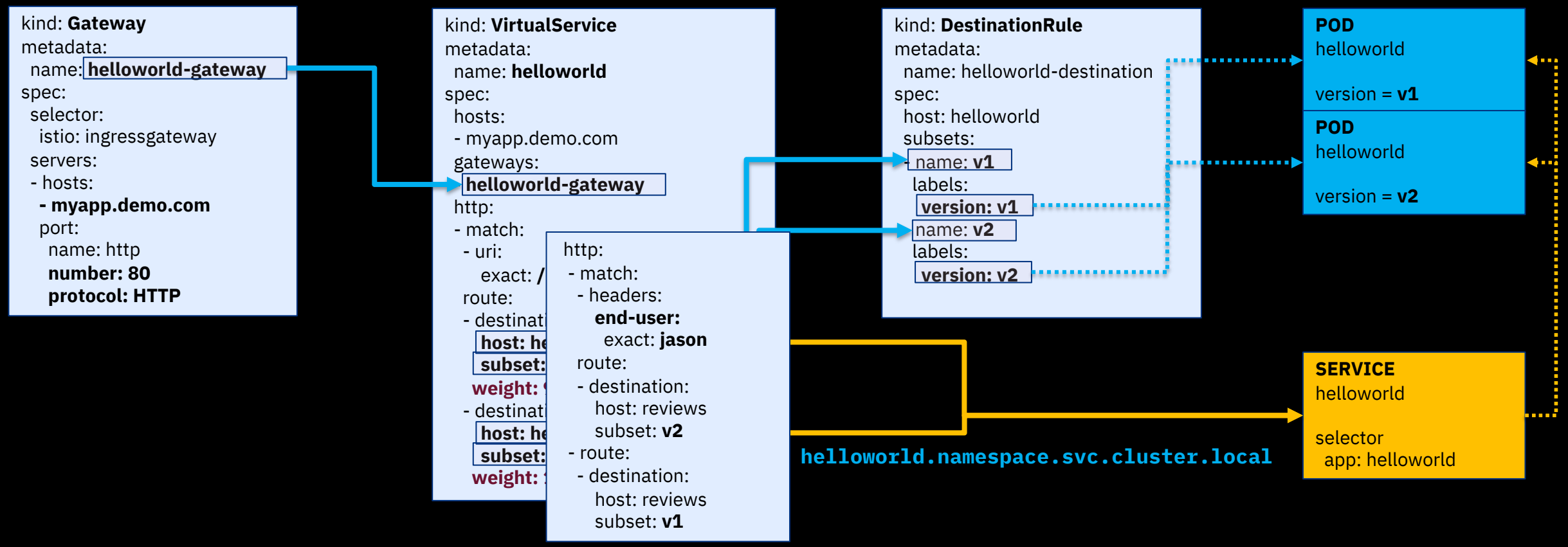
ISTIO

Custom resource definitions



ISTIO

Custom resource definitions



ISTIO
Sidecar injection

Manual Injection

```
kubectl apply -f <(istioctl kube-inject -f myapp.yaml)
```

Automatic Injection

For automatic sidecar injection, Istio relies on Mutating Admission Webhooks.
Label the namespace where you are deploying the app with `istio-injection=enabled`

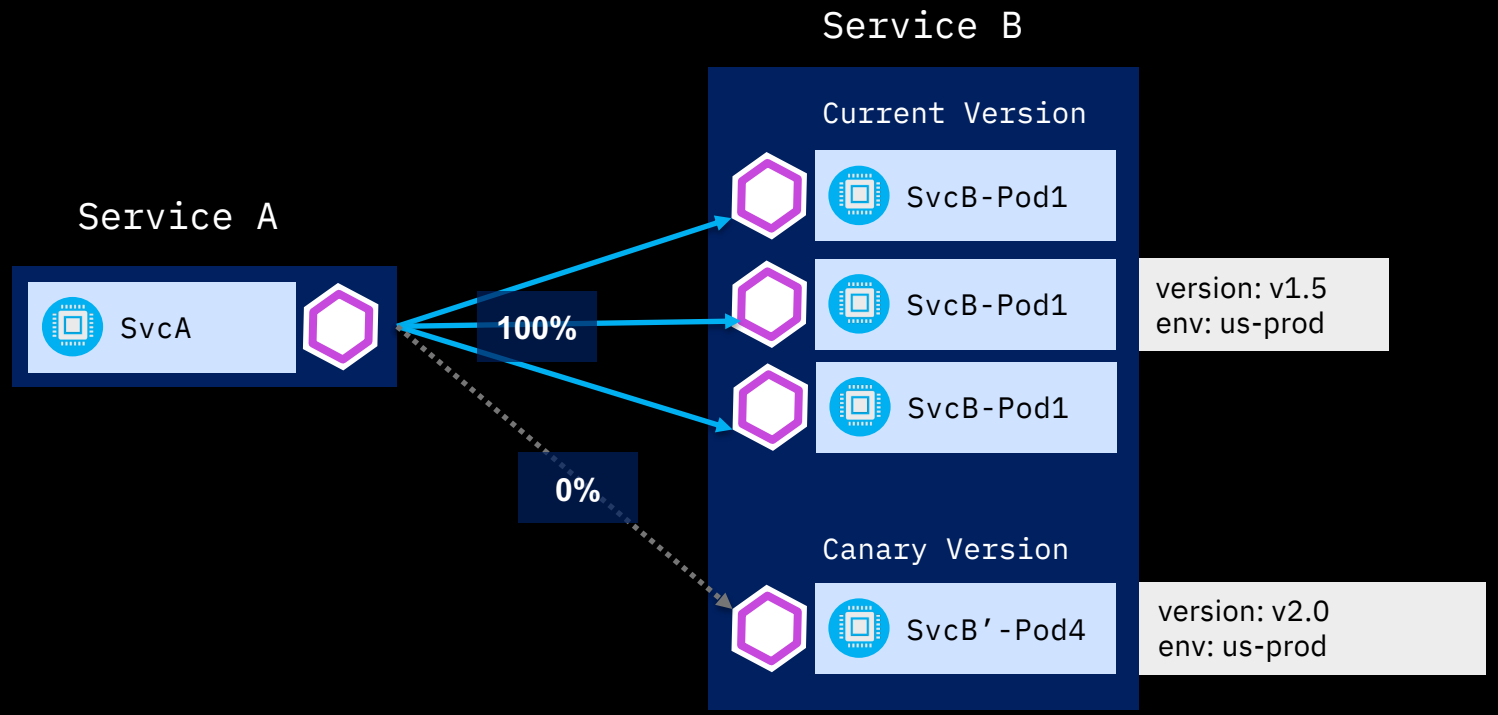
```
root@please1:~# kubectl get namespaces --show-labels
NAME                STATUS        AGE   LABELS
default             Active        35d   istio-injection=enabled
dev-namespace       Active        35d   <none>
godemo              Terminating   16d   istio-injection=enabled
istio-system        Active        35d   icp=system
kube-public         Active        35d   <none>
kube-system         Active        35d   icp=system
platform            Active        35d   <none>
prod-namespace      Active        35d   <none>
services            Active        35d   <none>
test-namespace      Active        35d   <none>
```

```
kind: Deployment
metadata:
  name: no-injection
spec:
  template:
    metadata:
      annotations:
        sidecar.istio.io/inject: "false"
    spec:
      containers:
      - name: no-injection
        image: nginx
```

Addressing DevOps Challenges

#	CHALLENGE	ISTIO SOLUTION
CHALLENGE 1	ROLL OUT NEW VERSION WITHOUT DOWNTIME OR CHANGING CODE	TRAFFIC CONTROL
CHALLENGE 2	HOW TO DO CANARY TESTING	TRAFFIC SPLITTING
CHALLENGE 3	HOW TO DO A/B TESTING	TRAFFIC STEERING
CHALLENGE 4	THINGS DON'T ALWAYS GO CORRECTLY IN PRODUCTION…	TRAFFIC MIRRORING RESILIENCY RESILIENCY TESTING
CHALLENGE5	HOW CAN I LIMIT RATE FOR SOME OF MY SERVICES?	RATE LIMITING
CHALLENGE 6	I NEED TO VIEW AND MONITOR WHAT IS GOING ON WHEN CRISIS ARISES	TELEMETRY
CHALLENGE 7	HOW CAN I SECURE MY SERVICES?	AUTHENTICATION AUTHORIZATION CALICO

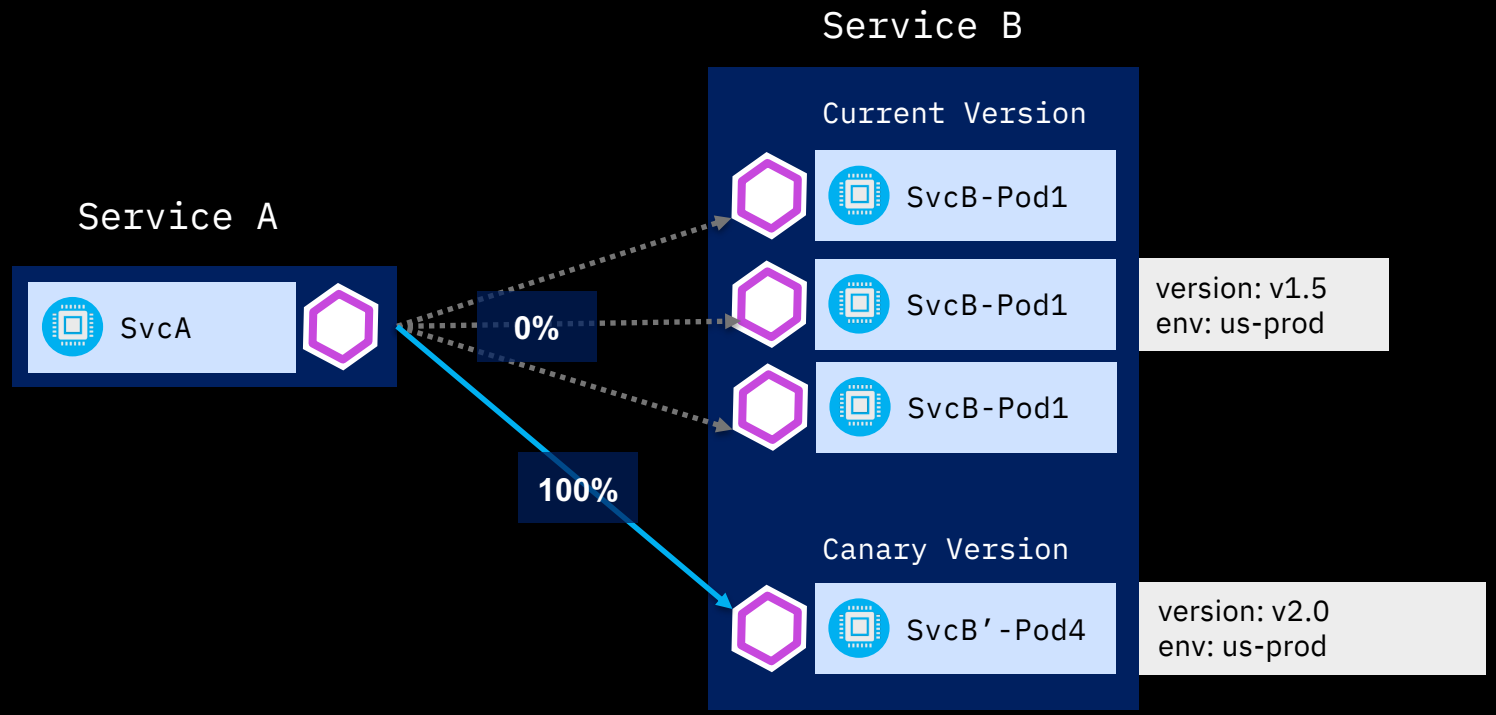
Traffic Control




```
kind: VirtualService
destination:
  serviceB.cluster.local
match:
  source:
    serviceA.cluster.local
route:
  - labels:
    version: v1.5
    env: us-prod
  weight: 100
```

CHALLENGE 1
ROLL OUT NEW VERSION WITHOUT DOWNTIME OR CHANGING CODE

Traffic Control

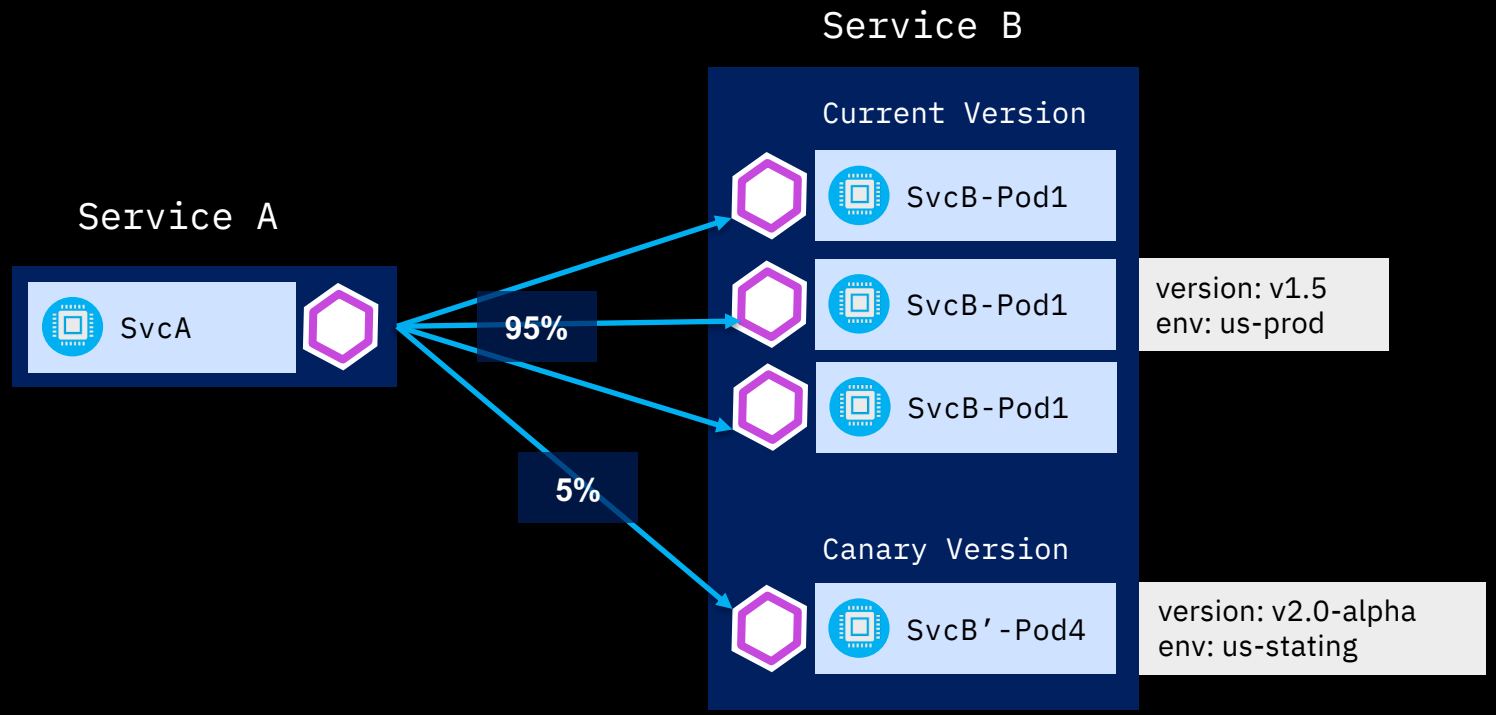



```
kind: VirtualService
destination:
  serviceB.cluster.local
match:
  source:
    serviceA.cluster.local
route:
  - labels:
    version: v2.0
    env: us-prod
  weight: 100
```

Update

CHALLENGE 1
ROLL OUT NEW VERSION WITHOUT DOWNTIME OR CHANGING CODE

Traffic Splitting

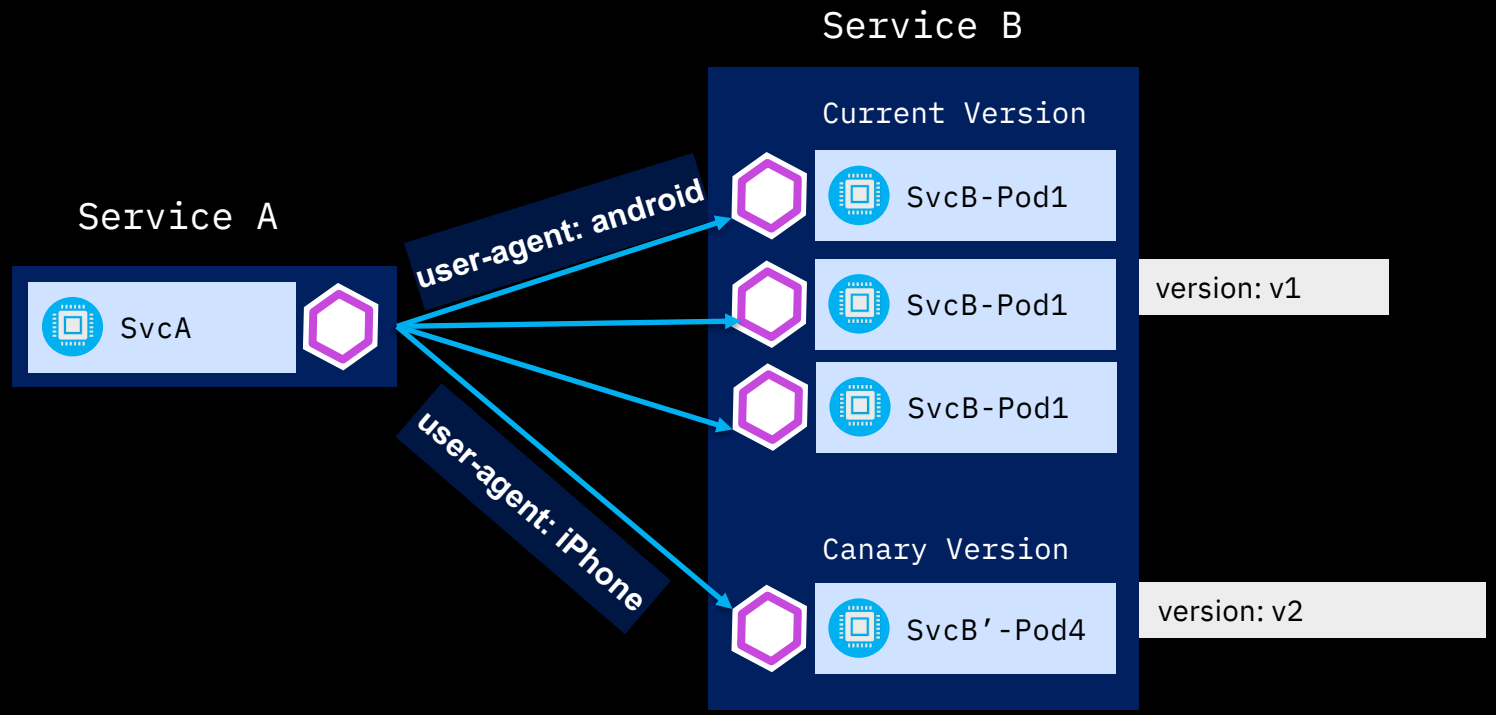



```
kind: VirtualService
destination:
  serviceB.cluster.local
match:
  source:
    serviceA.cluster.local
route:
  - labels:
    version: v1.5
    env: us-prod
  weight: 95
  - labels:
    version: v2.0-alpha
    env: us-staging
  weight: 5
```

CHALLENGE 2
HOW TO DO CANARY TESTING

Routing not based on the request content.
Staged rollouts with %-based traffic splits.

Traffic Steering

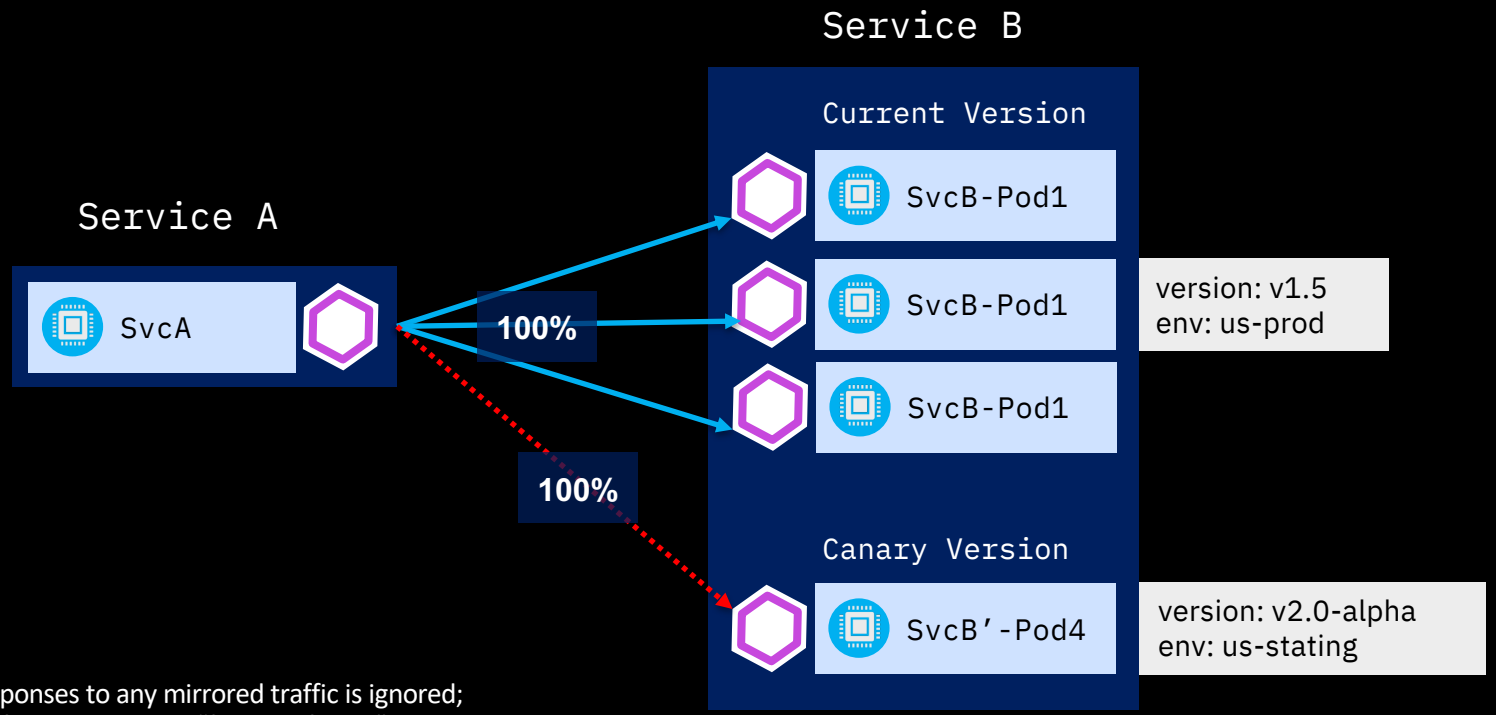



```
kind: VirtualService
destination:
 serviceB.cluster.local
match:
  httpHeaders:
    user-agent:
      regex:
        ^(.*?;)?(iPhone)(;.*)?$
 route:
- labels:
  version: v2
```

Routing based on the request content

CHALLENGE 3
HOW TO DO A/B TESTING

Traffic Mirroring



Responses to any mirrored traffic is ignored;
traffic is mirrored as "fire-and-forget"
You'll need to have the 0-weighted route to hint to
Istio to create the proper Envoy

```
kind: VirtualService
destination:
  serviceB.cluster.local
match:
  source:
    serviceA.cluster.local
route:
  - labels:
    version: v1.5
    env: us-prod
  weight: 100
  - labels:
    version: v2.0-alpha
    env: us-staging
  weight: 0
  mirror:
    name: httpbin
    labels:
      version: v2.0-alpha
      env: us-staging
```

CHALLENGE 4
THINGS DON'T ALWAYS GO CORRECTLY IN PRODUCTION...

Resiliency

Istio adds fault tolerance to your application without any changes to code

```
// Circuit breakers
destination: serviceB.example.cluster.local
policy:
- labels:
    version: v1
  circuitBreaker:
    simpleCb:
      maxConnections: 100
      httpMaxRequests: 1000
      httpMaxRequestsPerConnection: 10
      httpConsecutiveErrors: 7
      sleepWindow: 15m
      httpDetectionInterval: 5m
```

Resilience features

- Timeouts
- Retries with timeout budget
- Circuit breakers
- Health checks
- AZ-aware load balancing w/ automatic failover
- Control connection pool size and request load

CHALLENGE 4
THINGS DON'T ALWAYS GO CORRECTLY IN PRODUCTION...

Rate limiting

Istio protects your application from rogue actors by imposing ratelimits

```
Quotas:
- name: requestcount.quota.istio-system
  maxAmount: 5000
  validDuration: 1s
  overrides:
  - dimensions:
      destination: ratings
      source: reviews
      sourceVersion: v3
    maxAmount: 1
    validDuration: 1s
  - dimensions:
      destination: ratings
    maxAmount: 100
    validDuration: 1s
```

Rate limit

- Configurable limits with overrides
- Multiple rate limiting backends
- Conditional rate limiting

CHALLENGE 5
HOW CAN I LIMIT RATE FOR SOME OF MY SERVICES?

Telemetry

Monitoring & tracing should not be an afterthought in the infrastructure

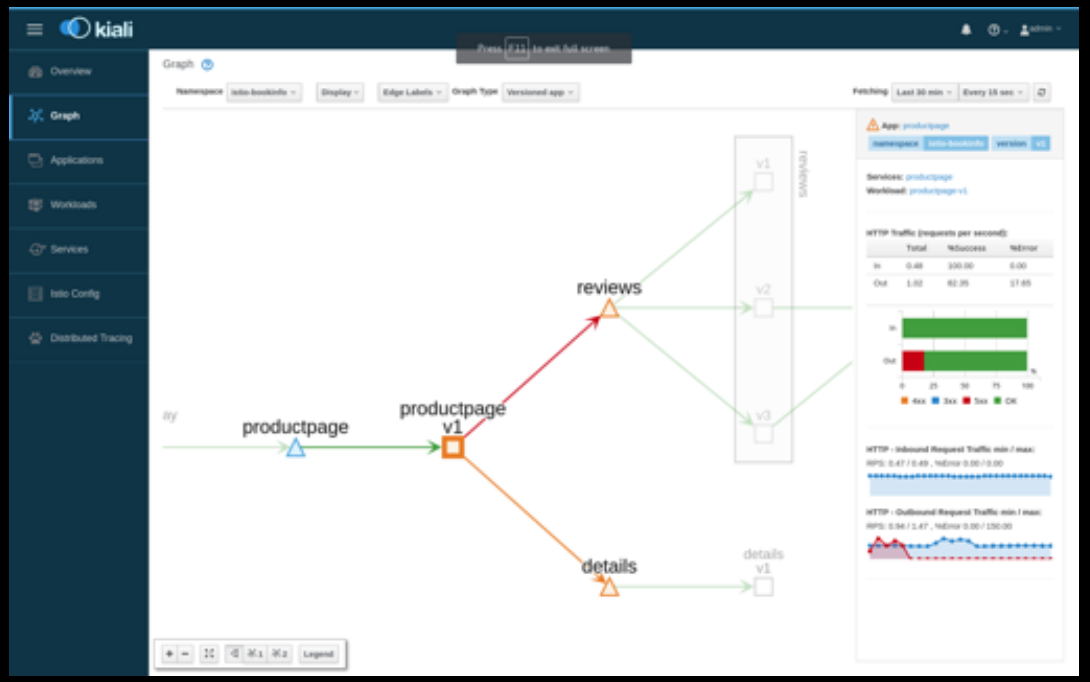
Goals

- Metrics without instrumenting apps
- Consistent metrics across fleet
- Trace flow of requests across services
- Portable across metric backend providers

CHALLENGE 6
I NEED TO VIEW WHAT IS GOING ON WHEN CRISIS ARISES

Kiali

Kiali (greek κιάλι)
monocular or *spyglass*
Visualise the service mesh topology, features like circuit breakers or request rates

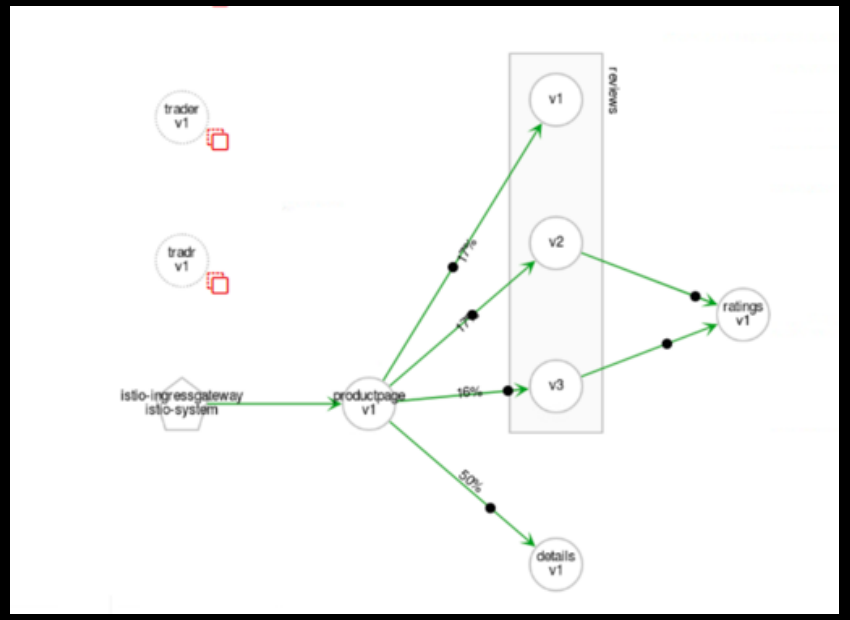


Features

Graph
- Health
- Types
- Side Panel
- Traffic Animation

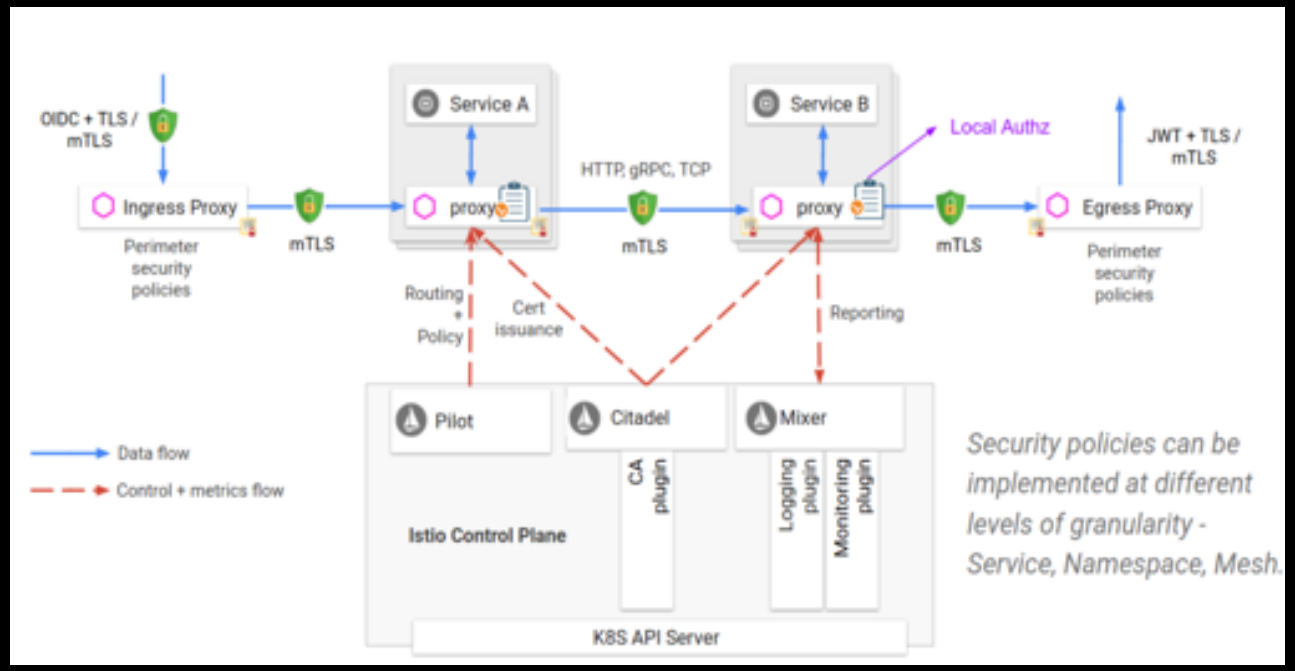
Applications, Workloads and Services
- Detailed Metrics

Traffic Routing
Istio compliance
Istio Configuration



CHALLENGE 6
I NEED TO VIEW WHAT IS GOING ON WHEN CRISIS ARISES

Security



Authentication

- Transport authentication, also known as service-to-service authentication
- Origin authentication, also known as end-user authentication

Authorization

- Based on RBAC
- Namespace-level, service-level and method-level access control for services

CHALLENGE 7
HOW CAN I SECURE MY SERVICES?

Service Mesh - Bad Idea ?

A Service Mesh is not always the right solution...

- **Service Meshes are Opinionated**
  They are a *platform* solution. “Work their way”
- **Service Meshes are Complex**
  Adds considerable complexity with sidecars and control plane
- **Service Meshes can be Slow**
  Routing traffic through a series of proxies can get painfully slow (about 700 nodes → reflector)
- **Service Meshes are for Developers**
  Focused primarily on Developer view.

https://glasnostic.com/blog/comparing-service-meshes-linkerd-vs-istio

Getting started

- Go to https://istio.io/

  Download ISTIO Release

With Kubectl

```
$ kubectl apply -f install/kubernetes/helm/istio/templates/crds.yaml

$ kubectl apply -f install/kubernetes/istio-demo.yaml
```

With HELM Templating

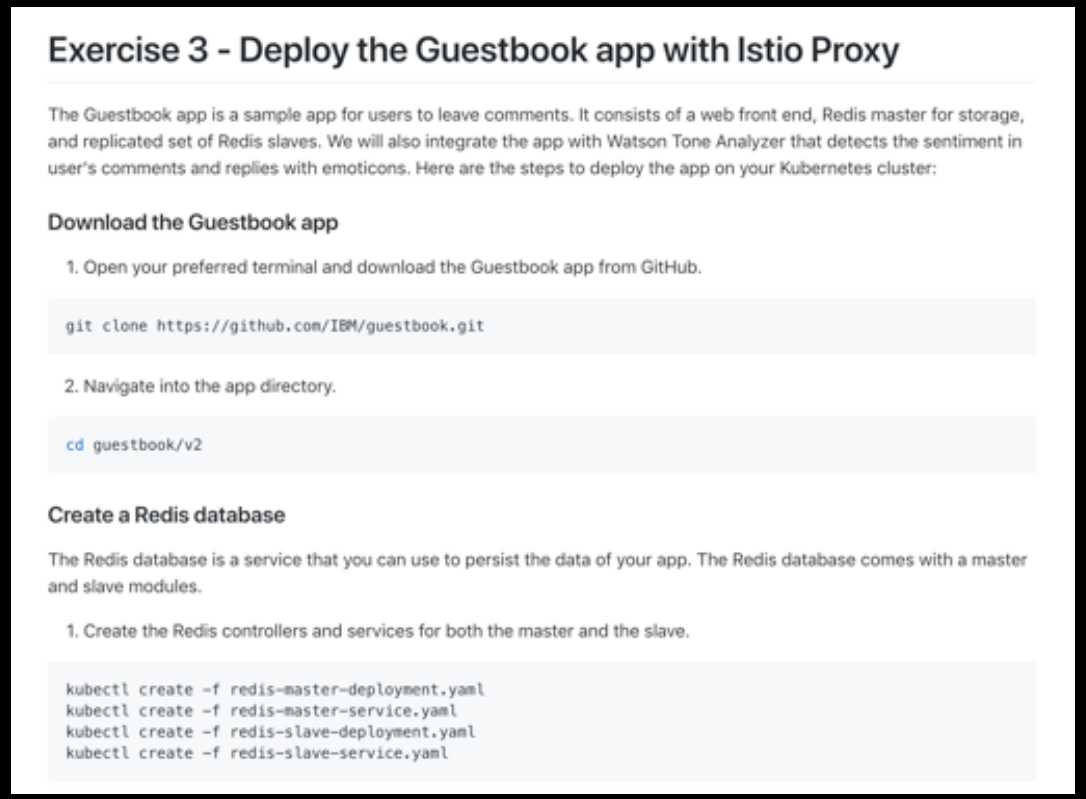
```
$ kubectl create namespace istio-system

$ helm template --name istio
      --namespace istio-system
      --set grafana.enabled=true
      --set servicegraph.enabled=true
      --set kiali.enabled=true > istio.yaml

$ kubectl apply -f istio.yaml
```

Getting started

- IBMs ISTIO 101 Hands On
- Go to https://github.com/IBM/istio101



Exercise 3 - Deploy the Guestbook app with Istio Proxy

The Guestbook app is a sample app for users to leave comments. It consists of a web front end, Redis master for storage, and replicated set of Redis slaves. We will also integrate the app with Watson Tone Analyzer that detects the sentiment in user's comments and replies with emoticons. Here are the steps to deploy the app on your Kubernetes cluster:

Download the Guestbook app

1. Open your preferred terminal and download the Guestbook app from GitHub.

```
git clone https://github.com/IBM/guestbook.git
```

2. Navigate into the app directory.

```
cd guestbook/v2
```

Create a Redis database

The Redis database is a service that you can use to persist the data of your app. The Redis database comes with a master and slave modules.

1. Create the Redis controllers and services for both the master and the slave.

```
kubectl create -f redis-master-deployment.yaml
kubectl create -f redis-master-service.yaml
kubectl create -f redis-slave-deployment.yaml
kubectl create -f redis-slave-service.yaml
```

Useful links

- Web istio.io
- Twitter: @Istiomesh
- Istio 101: https://github.com/IBM/istio101
- Traffic management using Istio: https://ibm.co/2F7xSnf
- Resiliency and fault-tolerance using Istio: https://bit.ly/2qStF2B
- Reliable application roll out and operations using Istio: https://bit.ly/2K9IRQX

IBM

QUESTIONS?

IBM **Cloud**

IBM

The Journey to Cloud

Serverless with Knative

03

IBM **Cloud**

Serverless paradigm

Source-to-image

Simply provide code to the platform and the platform manages all of the hosting aspects (e.g., building, hosting, scaling, etc.) for them.

Auto-scaling/scale-to-zero

Scales the application based on the load it is experiencing. Including scaling the application down to zero instances when it is not in use.

Short-lived functions

Splitting up the microservices into even smaller “functions” allows for a more fine-grained hosting model, meaning better resource utilization.

Event-driven

Optimized scaling by responding to events rather than simply always running and waiting for something to happen. This allows for a much more loosely-coupled architecture.

Knative

A platform for developers

to

build and run

serverless applications

atop Kubernetes

Open source project being developed by some of the key cloud innovators, including IBM, RedHat, Google, Pivotal, and SAP

Knative - building blocks

Knative solves these concerns through its three main components:

Build: Integrate building of container images into t lication configuration ~~DEPRECATED in 0.8~~ ge can **encapsulate the configuration** of their application n of how to build their application images all at once.

Serving: **Event-driven** hosting scheme to ensure that the **applications are scaled** based on actual need, including **scaling down to zero** when appropriate. Also automatically **manages the rolling-out** of newer versions of the code and allow for advanced traffic routing (such as A/B testing), relying on Istio.

Eventing: **Core eventing primitives** to allow for the specification of interest in events from event sources (both internal and external to the cluster), as well as simple orchestration.

Tekton Pipelines

Tekton Pipelines are **Cloud Native**

- Run on Kubernetes
- Have Kubernetes clusters as a first class type
- Use containers as their building blocks

Tekton Pipelines are **Decoupled**

- One Pipeline can be used to deploy to any k8s cluster
- The Tasks which make up a Pipeline can easily be run in isolation
- Resources such as git repos can easily be swapped between runs

Tekton Pipelines are **Typed**

- The concept of typed resources means that for a resource such as an Image, implementations can easily be swapped out (e.g. building with kaniko v.s. buildkit)

Tekton Pipelines

```yaml
apiVersion: tekton.dev/v1alpha1
kind: Pipeline
metadata:
  name: demo-pipeline
spec:
  tasks:
    - name: build-skaffold-web
      taskRef:
        name: build-docker-image-from-git-source
      resources:
        inputs:
          - name: docker-source
            resource: myproject-git
    - name: deploy-web
      taskRef:
        name: deploy-using-kubectl
    ...
```


```yaml
apiVersion: tekton.dev/v1alpha1
kind: Task
metadata:
  name: echo-hello-world
spec:
 inputs:
    resources:
      - name: docker-source
        type: git
  steps:
    - name: echo
      image: ubuntu
      command:
        - echo
      args:
        - "hello world"
```

```yaml
apiVersion: tekton.dev/v1alpha1
kind: PipelineResource
metadata:
  name: myproject-git
spec:
  type: git
  params:
    - name: revision
      value: master
    - name: url
      value: https://github.com/nick/myproject
```

GitHub

Knative – Serving

The Knative Serving project provides middleware primitives that enable:

- Automatically deploy containers and configure routing
- Automatically scale up and down, including scale-to-zero
- Point-in-time snapshots of deployments allows multiple versions of applications at once
- Easy rollbacks, blue-green deployments, partial load testing, etc.

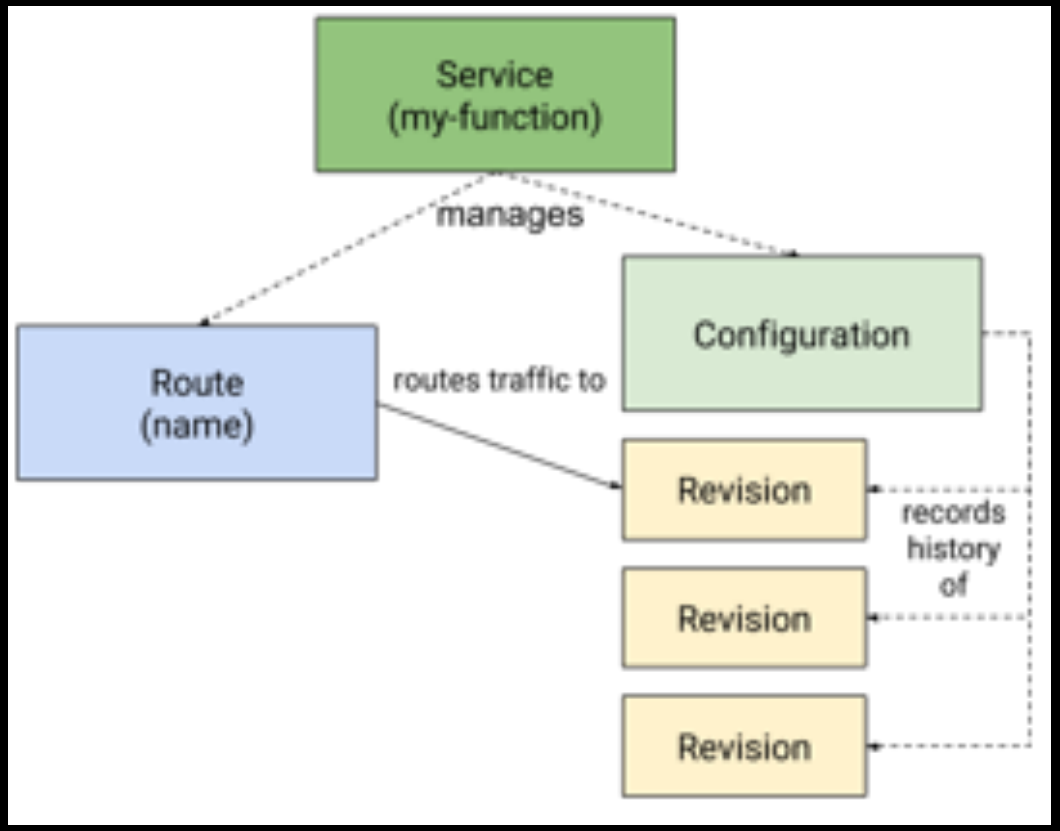
Knative – Serving

Revision: Single instance of your application. Contains a specific image and a specific set of configuration options - immutable.

Configuration: Responsible for defining the application image and its configuration, similar to a revision, except these values are mutable. Whenever these values are changed a new instance of the application (Revision) is created.

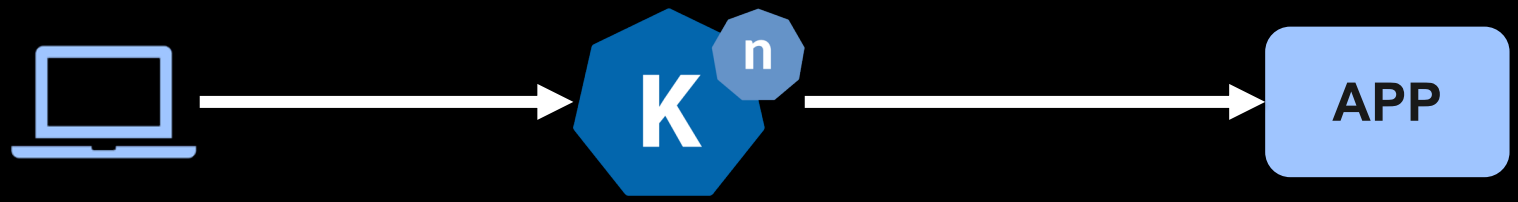
Route: Directing of traffic to a specific revision. By default sends all traffic to the latest revision. Used to specify traffic splits

(Knative) Service: Highest-level resource that ties together a complete serverless application. Only resource that users need to interact with to deploy their application.



Knative – Serving

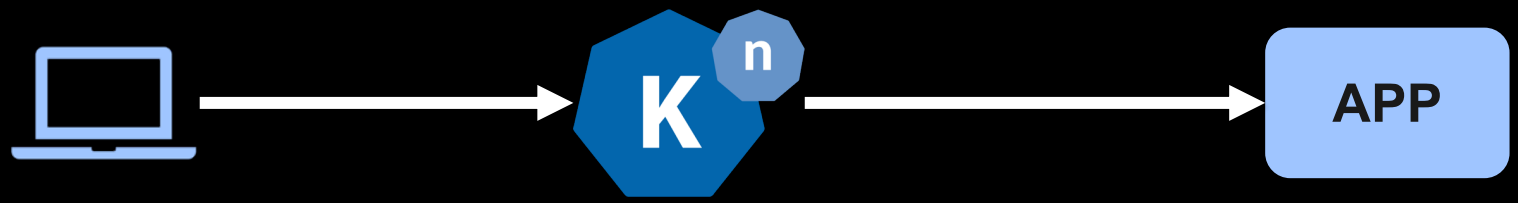
Deployment




```
apiVersion: serving.knative.dev/v1alpha1
kind: Service
metadata:
  name: knative-helloworld
  namespace: default
spec:
  runLatest:
    configuration:
      revisionTemplate:
        spec:
          container:
            image: docker.io/gswk/knative-helloworld:latest
```

Knative – Serving

Deployment



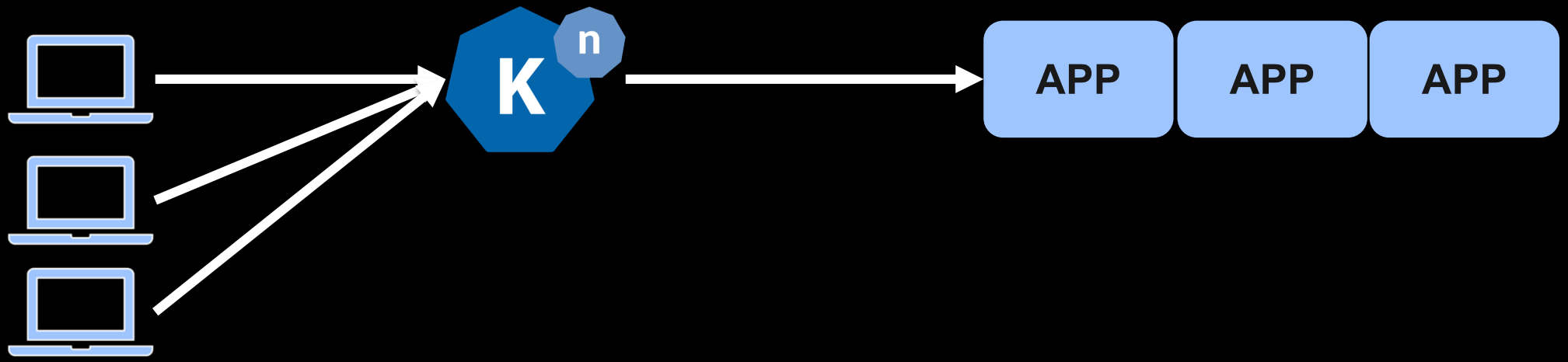
App reachable at:

{app-name}.{namespace}.{custom-domain}

knative-helloworld.default.example.com

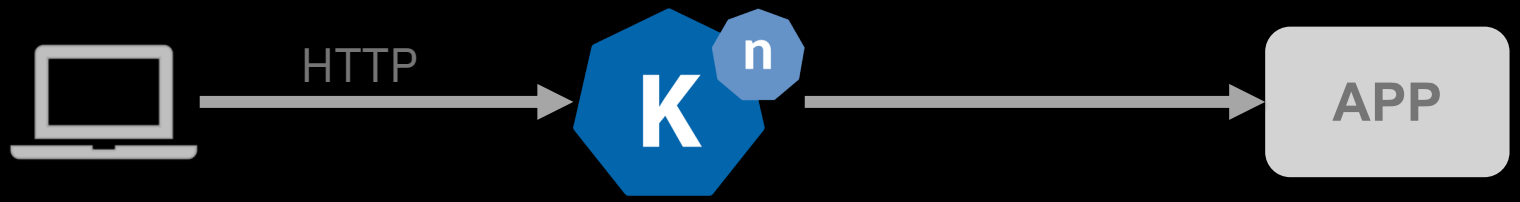
Knative – Serving

Scale up



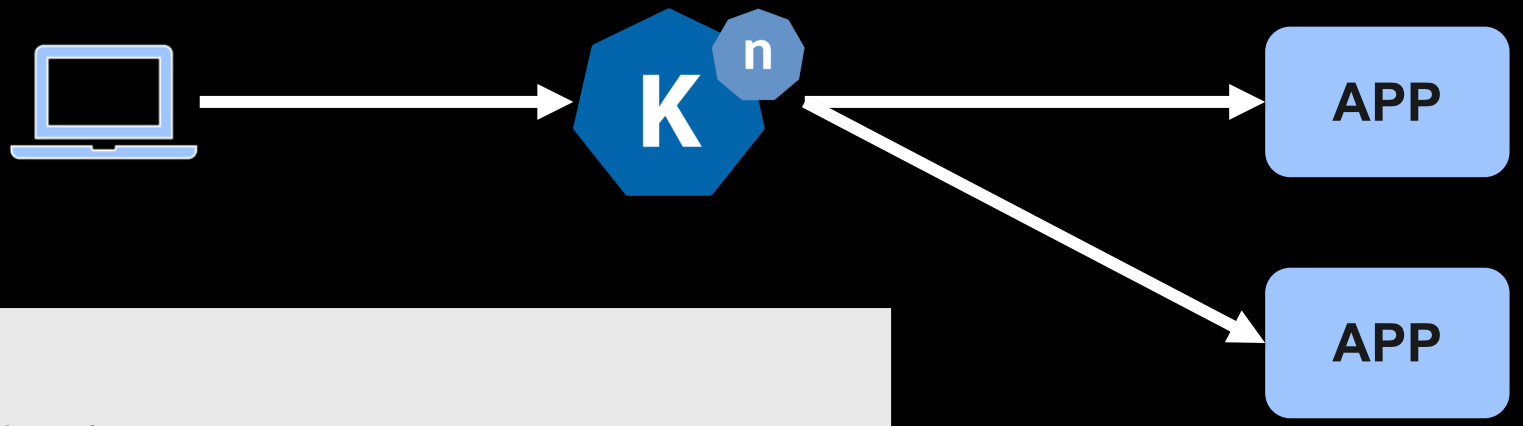
Knative – Serving

Scale down



Knative – Serving

Routing (Istio)



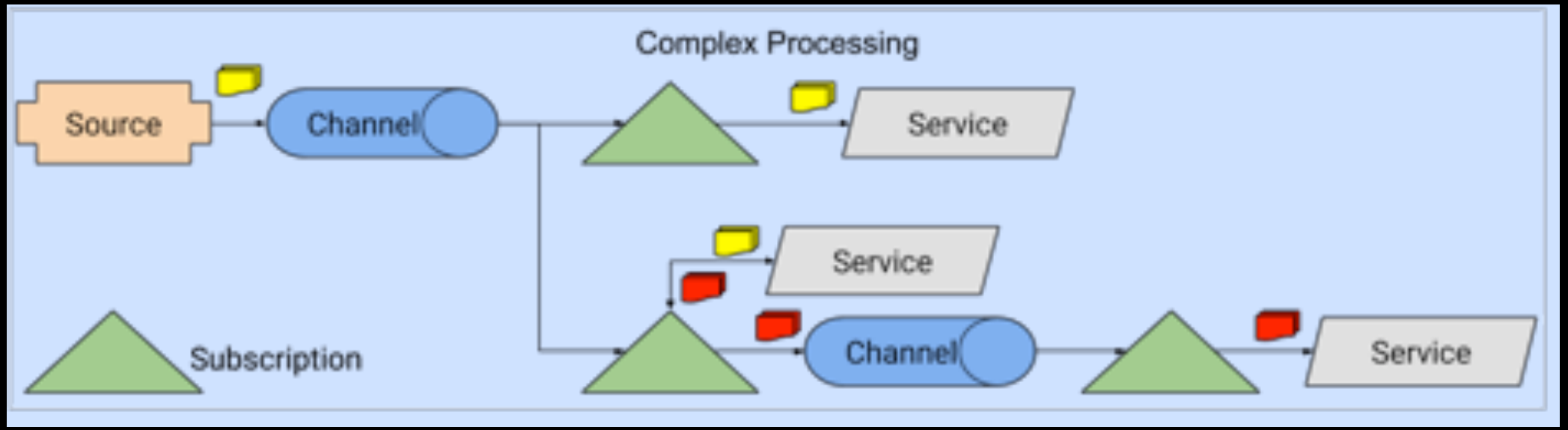

```
kind: Route
metadata:
  name: knative-routing-demo
  namespace: default
spec:
  traffic:
  - revisionName: knative-routing-demo-00001
    name: v1
    percent: 10
  - revisionName: knative-routing-demo-00002
    name: v2
    percent: 90
```

Knative – Eventing

Designed to address a common need for cloud native development, providing composable primitives to enable late-binding event sources and event consumers.

- Services are loosely coupled and can be developed and deployed independently on, and across a variety of platforms (for example Kubernetes, VMs, SaaS or FaaS).
- Event producers and event sources are independent.
- Other services can be connected to the Eventing system.
- Ensure cross-service interoperability.

Knative – Eventing



Some Event Sources

Kubernetes	Brings Kubernetes cluster events into Knative.
GitHub	Registers for events of the specified types on the specified GitHub organization/repository
Cron Job	Uses an in-memory timer to produce events on the specified Cron schedule.
....	

https://github.com/knative/docs/blob/master/eventing/sources/README.md

Getting started

- Go to https://github.com/knative/
- Istio and Knative: Extending Kubernetes for a New Developer Experience ibm.biz/Bd2Vet
- Knative: What is it and why should you care? ibm.biz/Bd2V8A
- Install Knative and Deploy an App on IBM Cloud ibm.biz/Bd2V8C
- IBMs Knative 101 Hands On https://github.com/IBM/knative101

IBM

QUESTIONS?

IBM **Cloud**

IBM

The Journey to Cloud

GitOps with ArgoCD

04

IBM **Cloud**

ArgoCD

A tool for developers

for

declarative, GitOps continuous delivery

atop Kubernetes

GitOps

At its core, GitOps refers to
a set of practices and tooling
that put
Git at the center of the DevOps toolchain
and as the
source of truth
for what should be deployed on the cluster.

With GitOps, developers and operators use familiar Git workflows to define, review, approve, and audit changes to their infrastructure and applications, whereas automated tools take care of synchronizing the live state of their cluster with the desired state described in Git.

ArgoCD

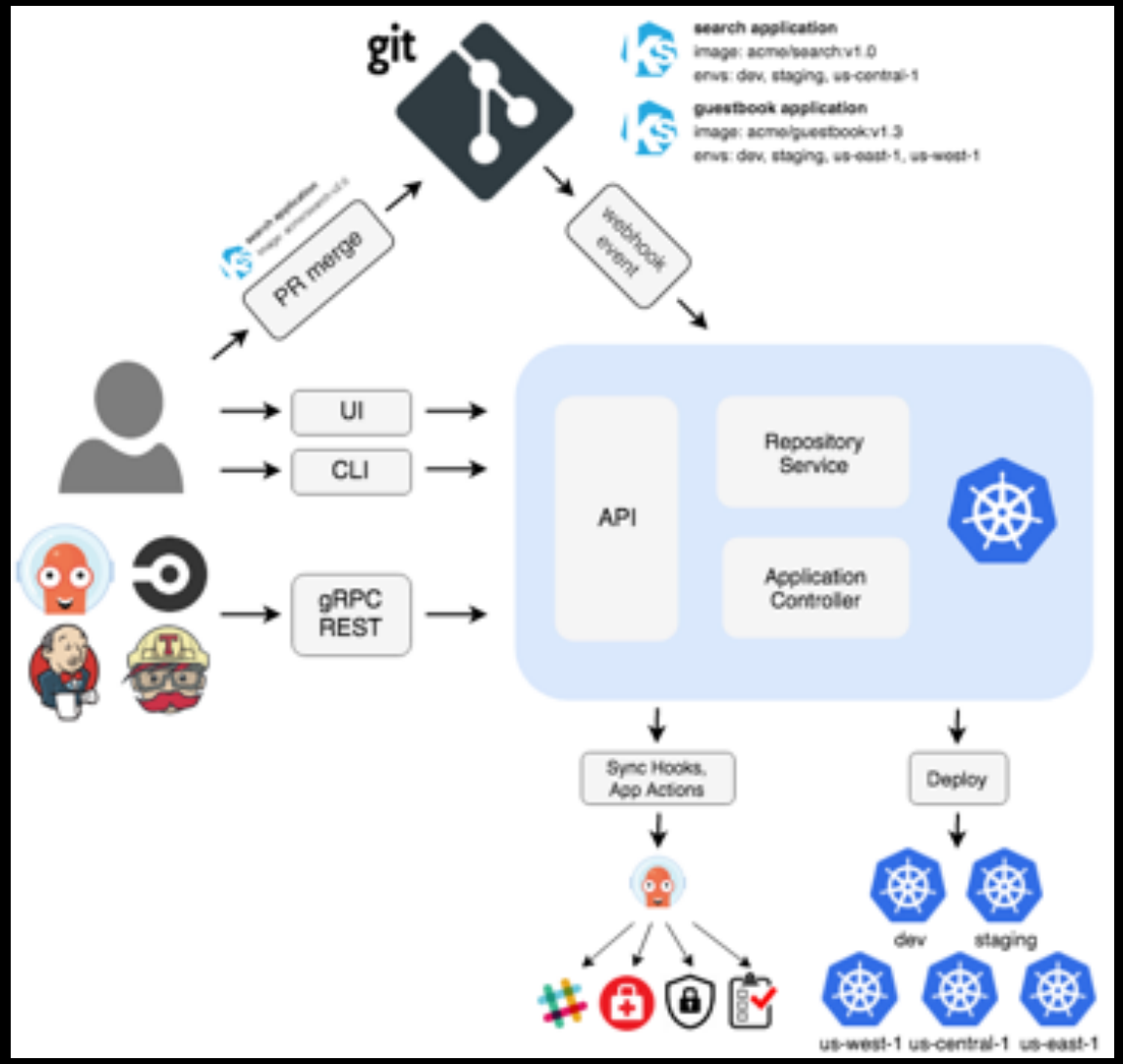
Argo CD is a declarative, GitOps continuous delivery tool for Kubernetes.

Application definitions, configurations, and environments should be declarative and version controlled. Application deployment and lifecycle management should be automated, auditable, and easy to understand.

- kustomize applications
- helm charts
- ksonnet applications
- jsonnet files
- Plain directory of YAML/json manifests

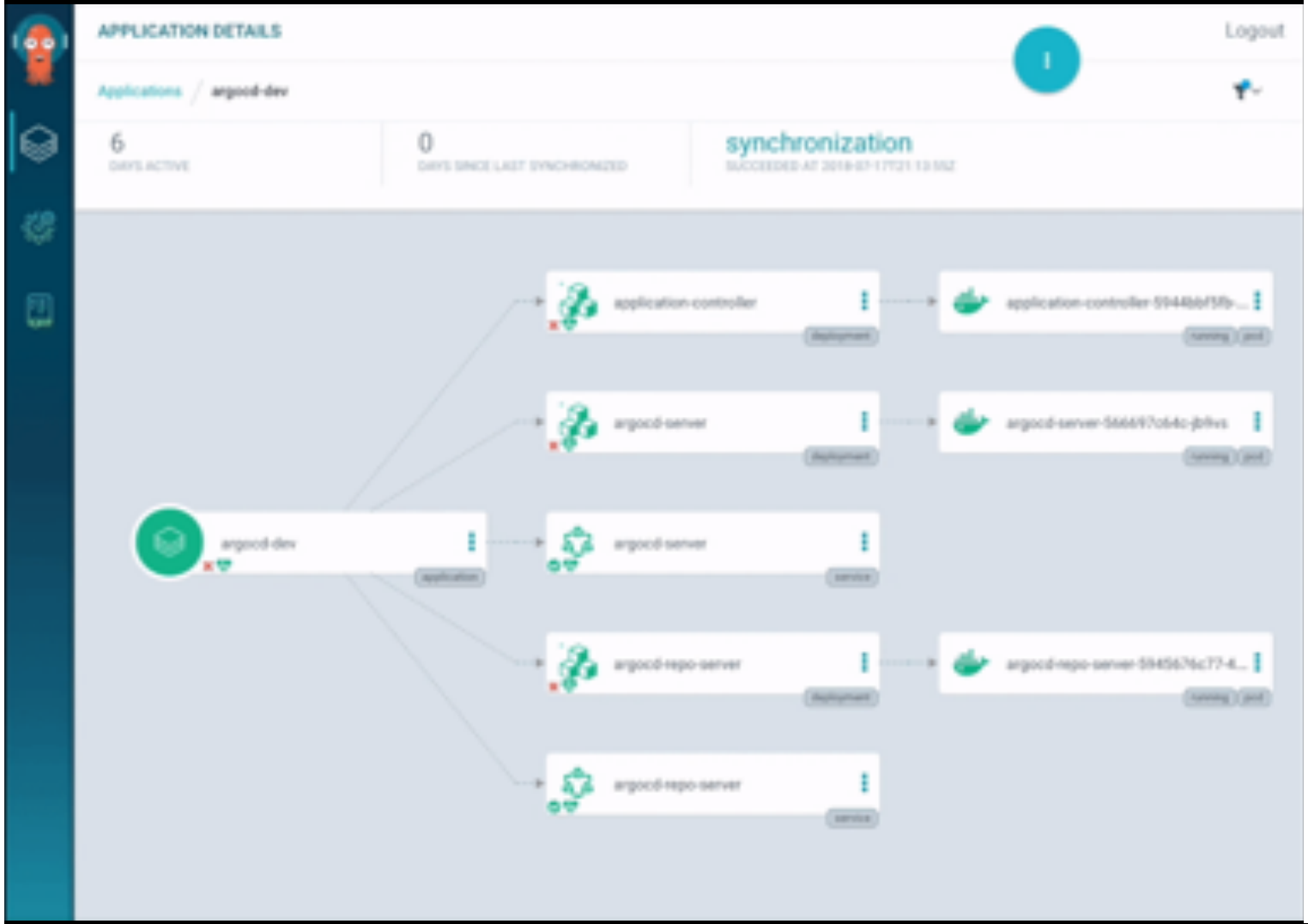
ArgoCD

Argo CD is a declarative, GitOps continuous delivery tool for Kubernetes.



ArgoCD

Argo CD is a declarative, GitOps continuous delivery tool for Kubernetes.



IBM

QUESTIONS?

IBM **Cloud**

The Journey to Cloud
Mesh Networking - *Hands On*

05

IBM **Cloud**

Remember your Team Color

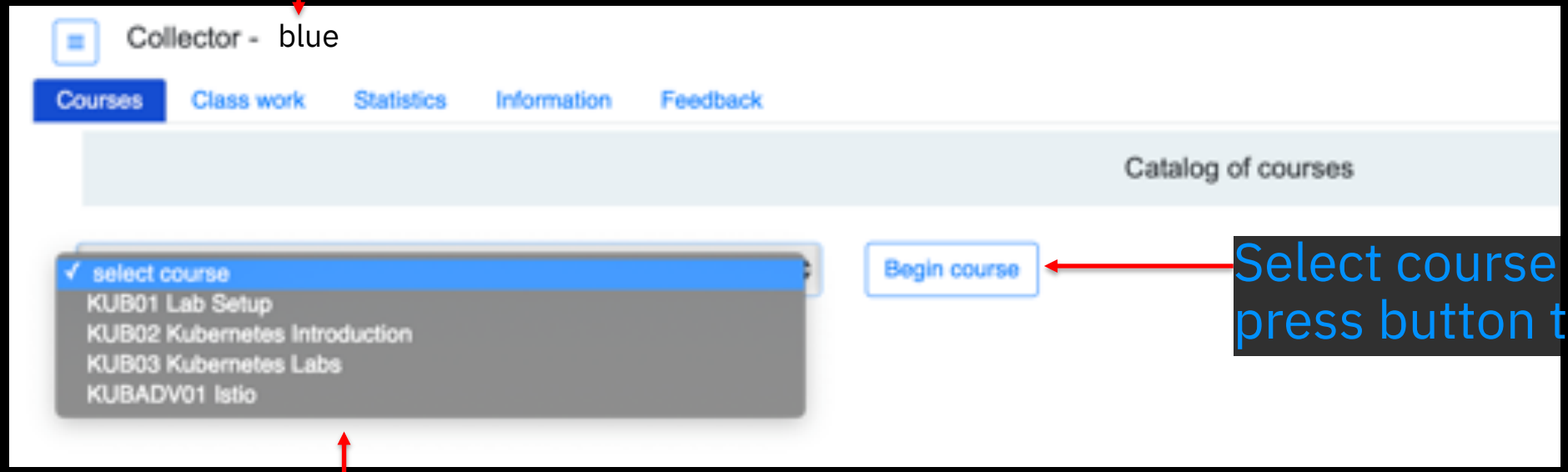
black 31701	olive 31711	peru 31715
white 31702	brown 31712	chocolate 31716
red 31703	lightblue 31713	orchid 31717
blue 31704	orange 31708	gold 31718
yellow 31705	purple 31709	pink 31719
lime 31706	maroon 31710	violet 31720
cyan 31707	firebrick 31714	

Collector - Accessing team web site

http://158.177.137.195:{port#}

blue 31704

Team name / color will be shown



JTC10 Istio

Lab 0 : Introduction

Lab 1 - Make sure minikube is running

Lab 2 - Installing Istio

Lab 3 - Deploy the Bookinfo App

Lab 4 - Monitoring with Kiali

Lab 5 - Traffic flow management

Lab 6 - Access policy enforcement

Lab 7 - Telemetry data aggregation

READY SET GO!!!!

http://158.177.137.195:{port#}

Duration: 60 mins

IBM

QUESTIONS?

IBM **Cloud**

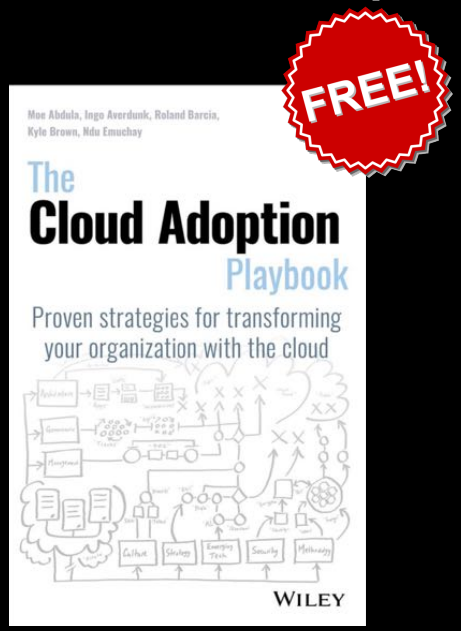
IBM

The Journey to Cloud
Wrap Up

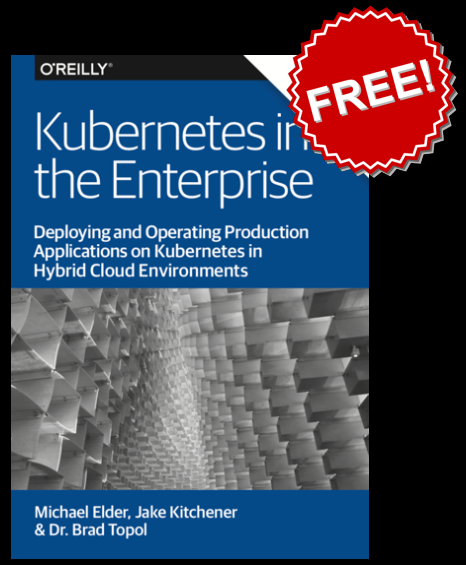
99

IBM **Cloud**

Cloud Adoption - Further reads



The de facto guide to improving your enterprise with the cloud, created by distinguished members of our Solution Engineering team
http://ibm.biz/playbook



Deploying and Operating Production Applications on Kubernetes in Hybrid Cloud Environments
http://ibm.biz/k8sintheenterprise

THANK YOU!!!!

Niklaus Hirt

nikh@ch.ibm.com

@nhirt