



Hack Shack Challenge

Enhancing Network Resiliency on the Edge

Lianjie Cao, lianjie.cao@hpe.com, Hewlett Packard Labs,
Pramod Reddy Sareddy, pramod-reddy.sareddy@hpe.com, HPE CDO

June 18-20, 2019



- ✓ **Introduction to Kubernetes and Service Mesh**
- ✓ **Getting images from cameras on the edge**
- ✓ **Challenge task**
- ✓ **Enjoy!**



Introduction to Kubernetes (K8s)



Kubernetes (K8s)

– Container-centric management platform for containers, microservices, etc.

– K8s cluster

– Master: control plane

– kube-apiserver, kube-scheduler, etcd, ...

– Nodes:

– kubelet, kube-proxy, container runtime, ...

– K8s objects

– Pod: wrapper of one or more containers

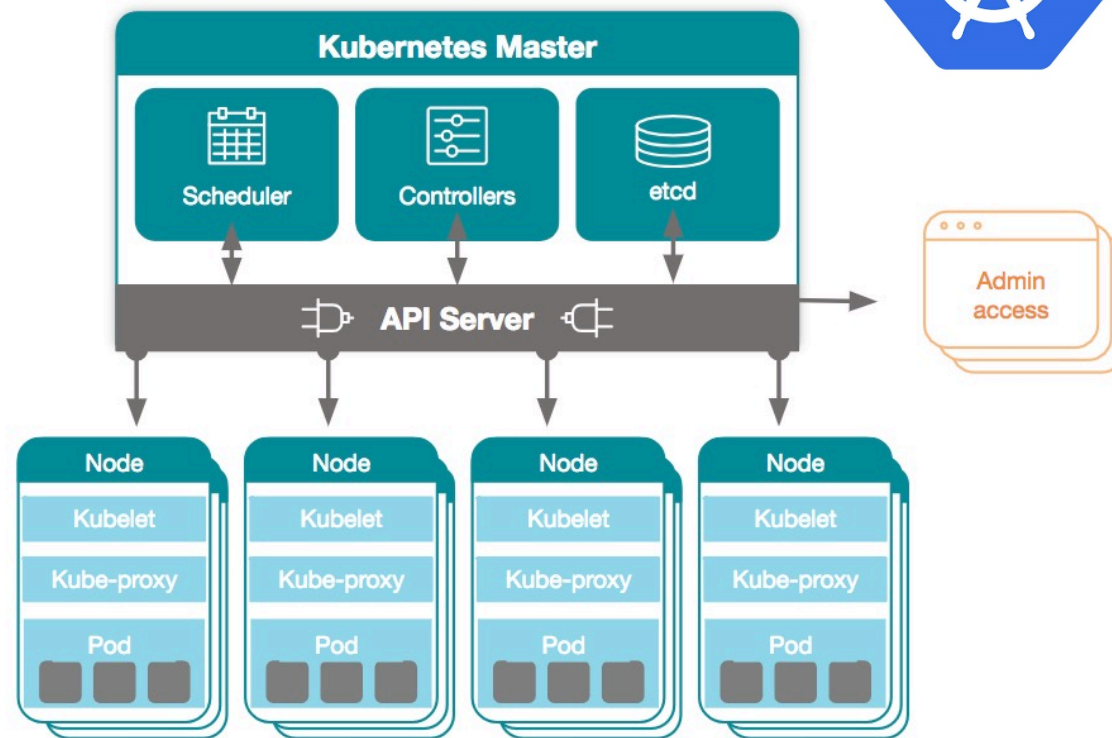
– Deployment: wrapper of pods

– Service: expose applications with network services

– K8s namespace

– Scope to have unique names for different objects

Kubernetes Architecture



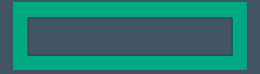
Examples

```
apiVersion: v1
kind: Service
metadata:
  name: webcam
spec:
  type: NodePort
  ports:
    - port: 8888
      name: http
  selector:
    app: webcam
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webcam
  labels:
    app: webcam
    tier: main
spec:
  replicas: 1
  selector: # Defines pods that will be used in this deployment
    matchLabels:
      app: webcam
      tier: main
  template: # Specify pod properties
    metadata:
      labels:
        app: webcam
        tier: main
    spec:
      containers:
        - name: webcam
          image: gongbaifei/webcam:v1
          imagePullPolicy: IfNotPresent
          ports:
            - containerPort: 8888
          env:
            - name: CAM_ADDR
              value: 192.168.1.11
            - name: CAM_USR
              value: "admin"
            - name: CAM_PASSWD
              value: "admin"
```



Useful K8s Commands



Create objects

```
kubectl create namespace webcam --label istio-injection=enabled  
kubectl apply -f webcam.yaml -n webcam
```

Show information of objects

```
kubectl get pods -o wide -all-namespaces  
kubectl get deployment -n webcam  
kubectl describe pods webcam-discover-2-7dfd96f95-kztnf -n webcam
```

Delete objects

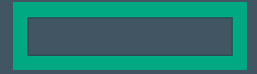
```
kubectl delete -f webcam.yaml -n webcam  
kubectl delete namespace webcam
```



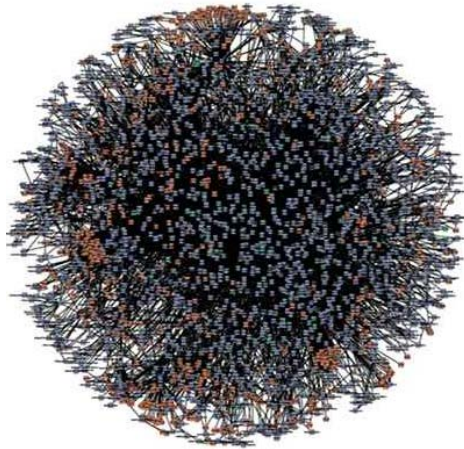
Introduction to Service Mesh (Istio)



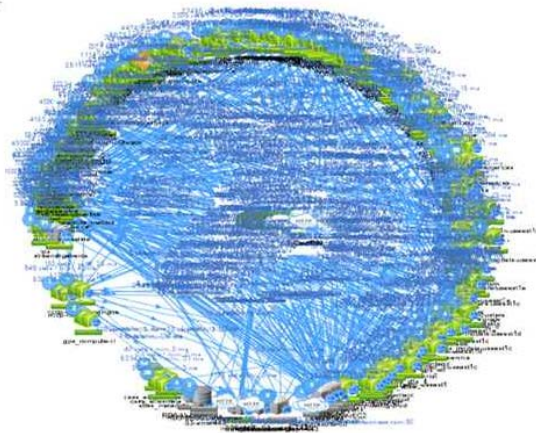
Service Mesh



servicemesh 



amazon.com



NETFLIX

Network of Microservices - Service-to-Service Communication

- Load balancing
- A/B testing
- Canary rollouts
- Rate limiting
- Access control
- End-to-end authentication

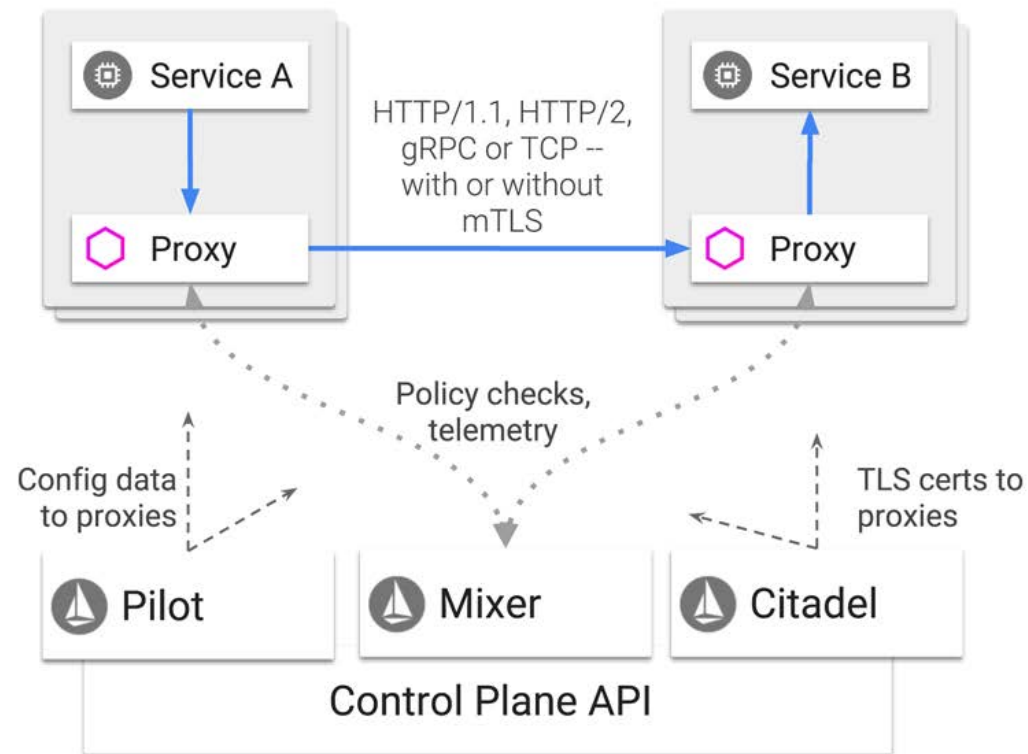


Service mesh is a communication paradigm that provides reconfigurable traffic management to microservices.

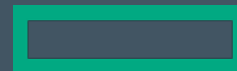
HPE DEV

Istio

- An implementation of service mesh (others include Linkerd, Consul Connect, etc)
- Developed and maintained by IBM, Google Cloud and the open community



Istio Service Examples



Gateway

Defines a load balancer at the edge of the mesh receiving incoming or outgoing connections

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: my-gateway
  namespace: some-config-namespace
spec:
  selector:
    app: my-gateway-controller
  servers:
  - port:
      number: 80
      name: http
      protocol: HTTP
    hosts:
    - "ns1/*"
    - "ns2/foo.bar.com"
```

Destination rules

Defines policies that apply to traffic intended for a service after routing has occurred

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: bookinfo-ratings
spec:
  host: ratings
  trafficPolicy:
    loadBalancer:
      simple: LEAST_CONN
  subsets:
  - name: testversion
    labels:
      version: v3
    trafficPolicy:
      loadBalancer:
        simple: ROUND_ROBIN
```

Virtual services

Defines a set of traffic routing rules to apply when a host is addressed

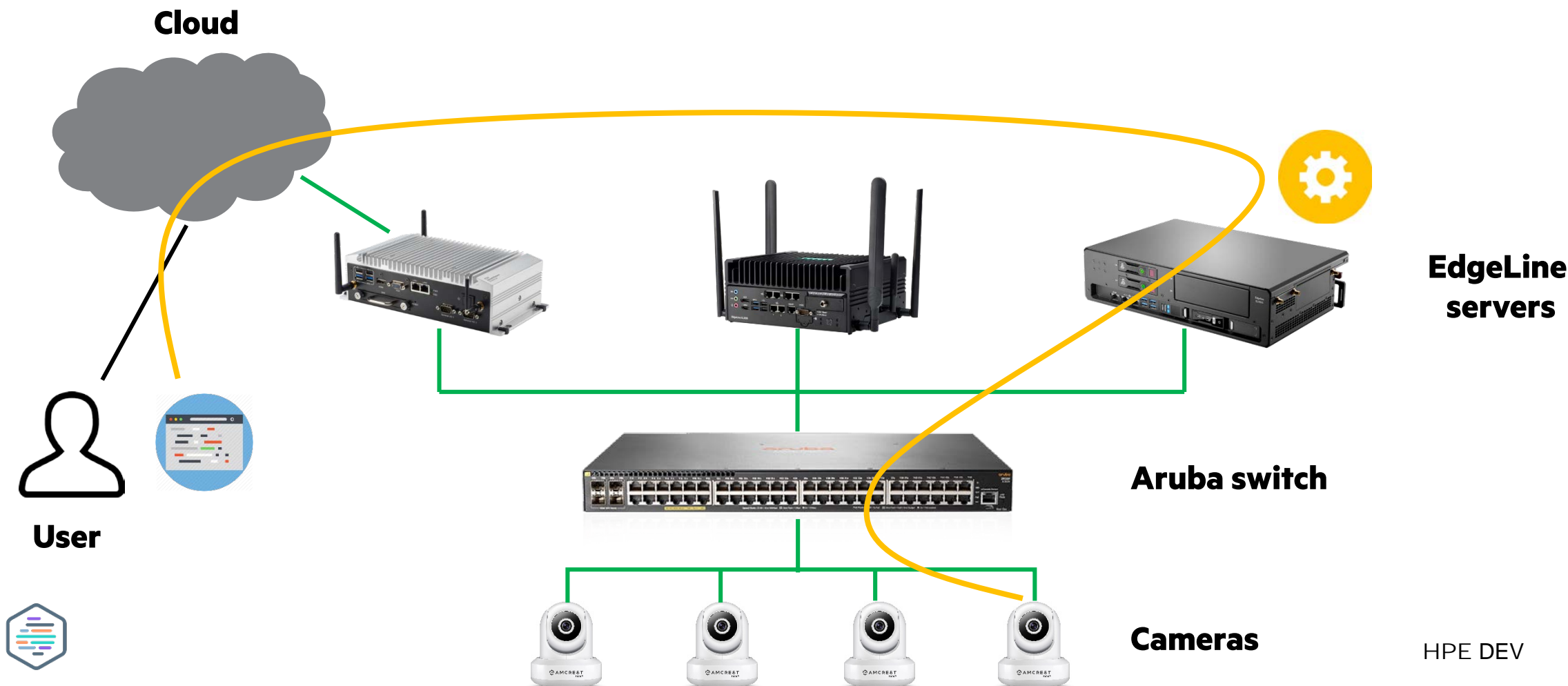
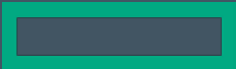
```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews-route
spec:
  hosts:
  - reviews
  http:
  - match:
    - uri:
        prefix: "/wpcatalog"
    - uri:
        prefix: "/consumercatalog"
    rewrite:
      uri: "/newcatalog"
    route:
    - destination:
        host: reviews
        subset: v2
  - route:
    - destination:
        host: reviews
        subset: v1
```



Cameras on the Edge



Challenge Setup



Challenge

Develop a solution to tolerate network variations (e.g., failures) for the application with or without Istio





Thank You!

#letshackshack

developer.hpe.com



HPE DEV