



AWS
re:Invent

CON219-R

Achieving zero-downtime deployments with Amazon Elastic Kubernetes Service (EKS)

Yang Yang

Software Development Engineer
Amazon Web Services

Vijay Khanna

Solutions Architect
Amazon Web Services

Agenda

Kubernetes basics

Service / Ingress networking

Tips and tricks

Demonstration

Q&A

Related breakouts

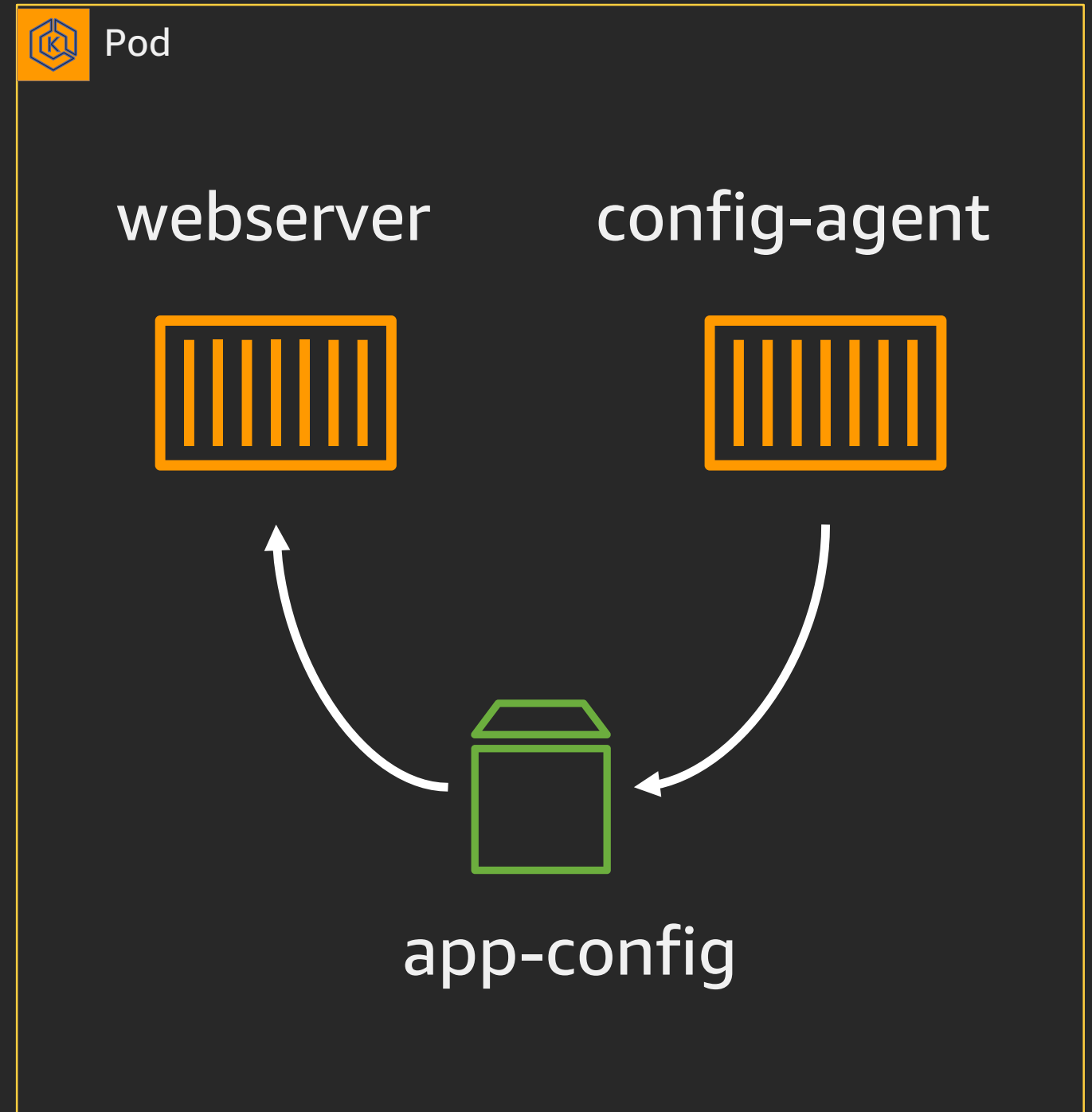
CON411-R: Advanced network resource management on Amazon EKS

CON315-R: Deep dive of observability of Kubernetes applications

Kubernetes basics

Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: awesome-a42wer-9uest
spec:
  containers:
  - name: webserver
    image: awesome-webserver: v1.0.0
    ports:
    - containerPort: 8080
    volumeMounts:
    - name: app-config
      mountPath: /etc/awesome/conf
  - name: config-agent
    image: awesome-config-agent: v1.0.0
    volumeMounts:
    - name: app-config
      mountPath: /etc/awesome/conf
  volumes:
  - name: app-config
    emptyDir: {}
```



ReplicaSet

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: awesome-a42wer
spec:
  replicas: 3
  selector:
    matchLabels:
      app: awesome
  template:
    metadata:
      labels:
        app: awesome
    spec:
      containers:
        - name: webserver
          image: awesome-webserver: v1.0.0
          ports:
            - containerPort: 8080
        - name: config-agent
          image: awesome-config-agent: v1.0.0
```



ReplicaSet

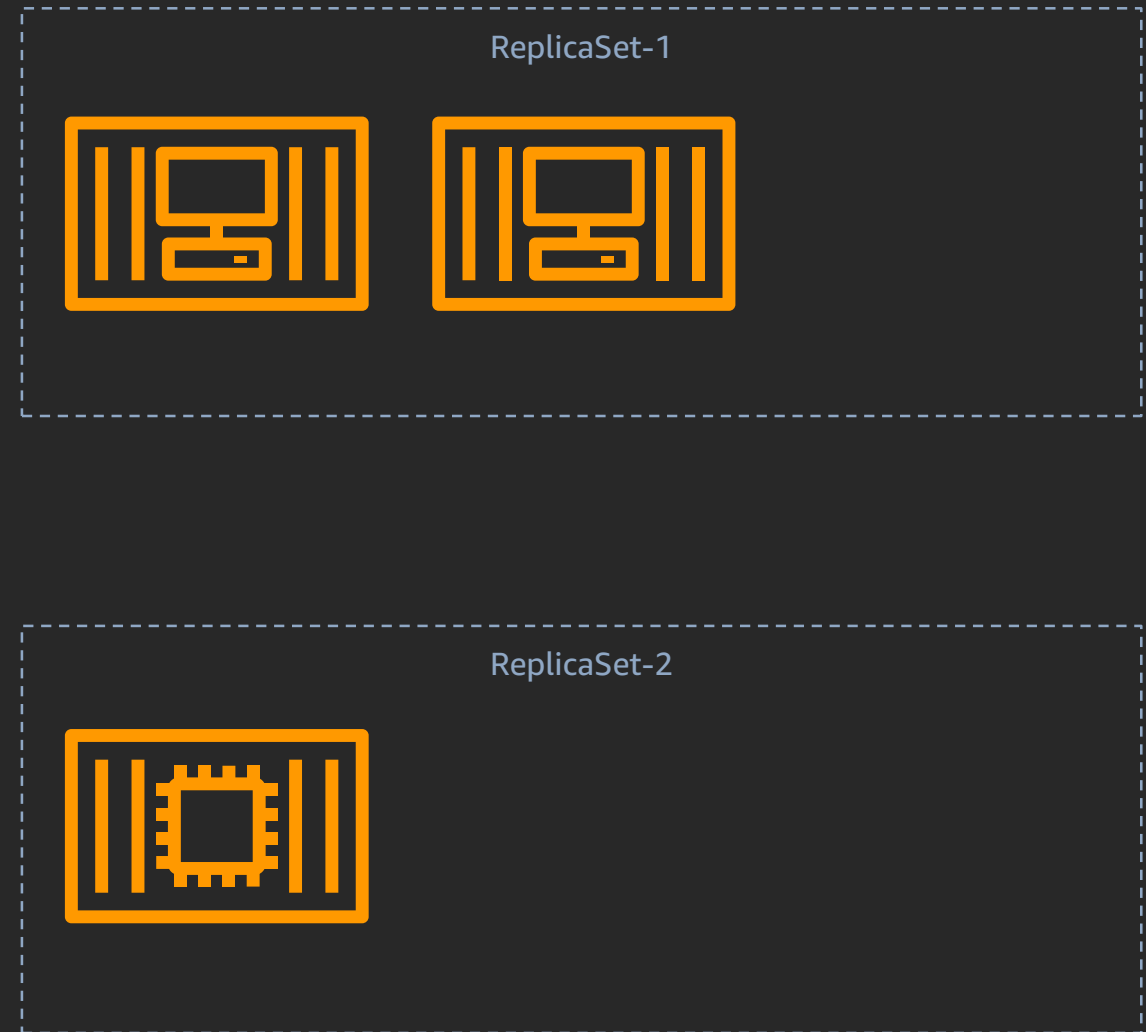


Deployment

```
apiVersion: v1
kind: Deployment
metadata:
  name: awesome
spec:
  replicas: 3
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 25%
      maxSurge: 25%
  selector:
    matchLabels:
      app: awesome
  template:
    ... pod template ...
```



Deployment



Service

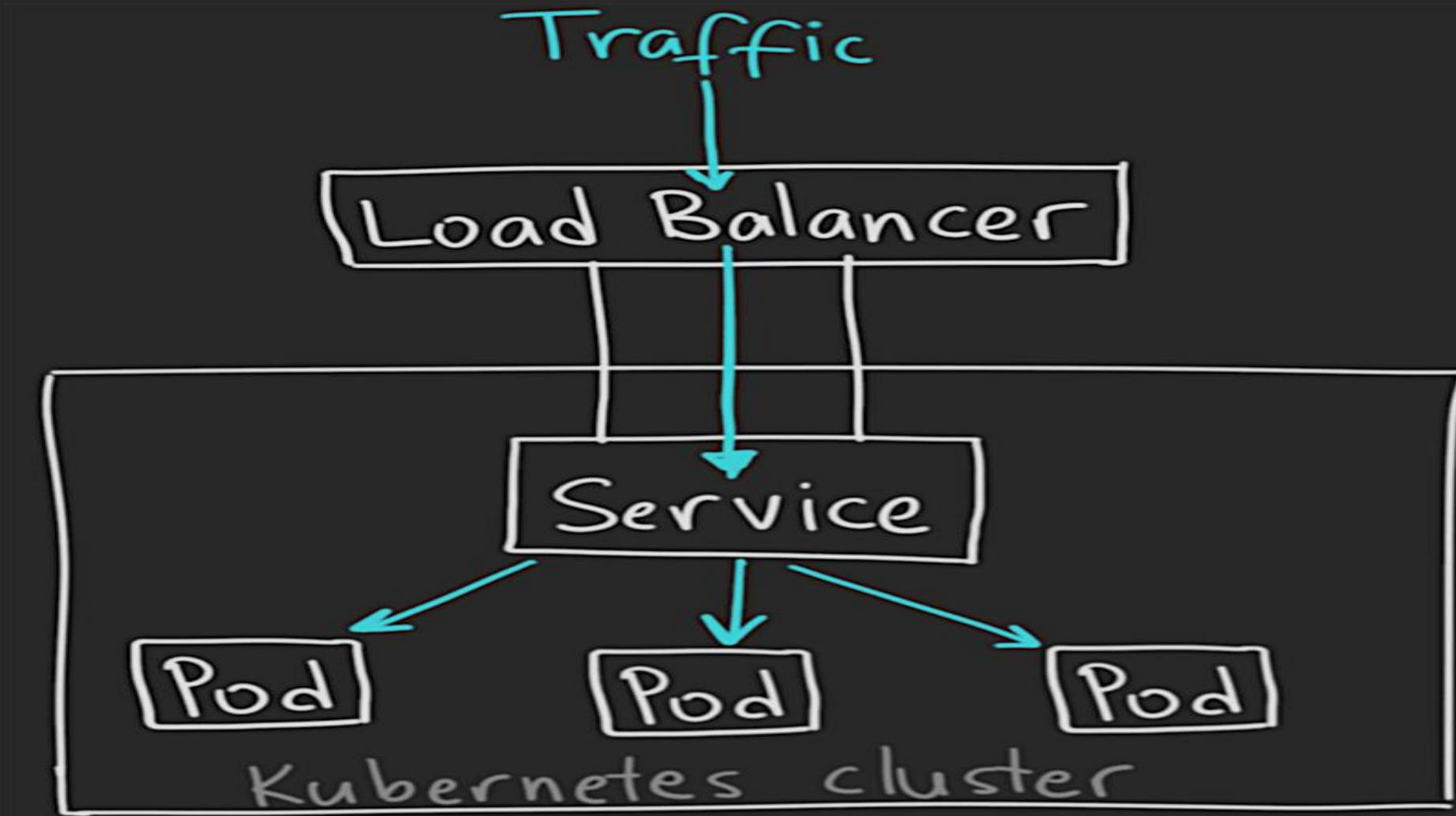
Service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: front-end-service-sample-app
spec:
  selector:
    app: sample-app
  type: LoadBalancer
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

Deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-sample-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: sample-app
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
  template:
    metadata:
      labels:
        app: sample-app
    spec:
      containers:
        - name: nginx-container
          image: nginx:1.7.9
          ports:
            - containerPort: 80
```

Kubernetes ServiceType: LoadBalancer



Tips and tricks

Kubernetes native - Rolling Update

Kubernetes supports the RollingUpdate strategy to **replace old pods with new ones gradually**, while continuing to serve clients without incurring downtime.

Set `.spec.strategy.type` to RollingUpdate (the default value).

Set `.spec.strategy.rollingUpdate.maxUnavailable` and `.spec.strategy.rollingUpdate.maxSurge`

maxUnavailable: the maximum number of pods that can be unavailable during the update process. This can be an absolute number or percentage of the replicas count; the default is 25%.

maxSurge: the maximum number of pods that can be created over the desired number of pods. Again this can be an absolute number or a percentage of the replicas count; the default is 25%.

Configure the **readinessProbe** for your service container to help Kubernetes determine the state of the pods.

Graceful shutdown – PreStop

PreStop hook is a great way to trigger a graceful shutdown without modifying the application.

spec:

containers:

- name: nginx

image: nginx:1.15

ports:

- containerPort: 80

lifecycle:

preStop:

exec:

command: [

Gracefully shutdown nginx

"/usr/sbin/nginx", "-s", "quit"

]

Graceful shutdown – termination grace period

If your pod usually takes longer than 30 seconds to shut down/drain gracefully, make sure you increase the grace period.

Kubernetes waits for a specified time called the termination grace period. By default, this is 30 seconds. It's important to note that this happens in **parallel** to the preStop hook and the SIGTERM signal. **Kubernetes does not wait for the preStop hook to finish.**

```
apiVersion: v1
kind: Pod
metadata:
  name: web-server-pod
spec:
  containers:
  - name: web-pod
    image: busybox
  terminationGracePeriodSeconds: 60
```

Graceful shutdown – connection draining

Service with Classic LoadBalancer:

service.beta.kubernetes.io/aws-load-balancer-connection-draining-enabled: "true"

service.beta.kubernetes.io/aws-load-balancer-connection-draining-timeout: "60"

Ingress with Application LoadBalancer:

alb.ingress.kubernetes.io/target-group-attributes: deregistration_delay.timeout_seconds=30

Graceful shutdown – signal 'connection close' to application load balancer (ALB)

Stop routing new requests to pod

Go away message to existing connections

For HTTP, connection:close header (e.g. ALB)

Graceful shutdown – handling SIGTERM

SIG_TERM Sent to main process (PID 1) in container

third-party application use different signal for graceful shutdown

NGINX: SIG_QUIT

Apache: SIG_WINCH

Docker ENTRYPOINT/CMD “exec form” vs “shell form”

CMD: my-app param1 param2

CMD: ["my-app", "param1", "param2"]

Shell “exec” without fork

/bin/sh -c my-app param1 param2 >log.txt 2>&1

/bin/sh -c exec my-app param1 param2 >log.txt 2>&1

Graceful startup: Probes

Readiness

readinessProbe:

exec:

command:

- cat
- /tmp/healthy

initialDelaySeconds: 5

periodSeconds: 5

spec:

containers:

- name: nginx

image: nginx:1.7.9

args:

- /server

readinessProbe:

httpGet:

path: /healthz

port: 8080

httpHeaders:

- name: Custom-Header
value: Awesome

initialDelaySeconds: 3

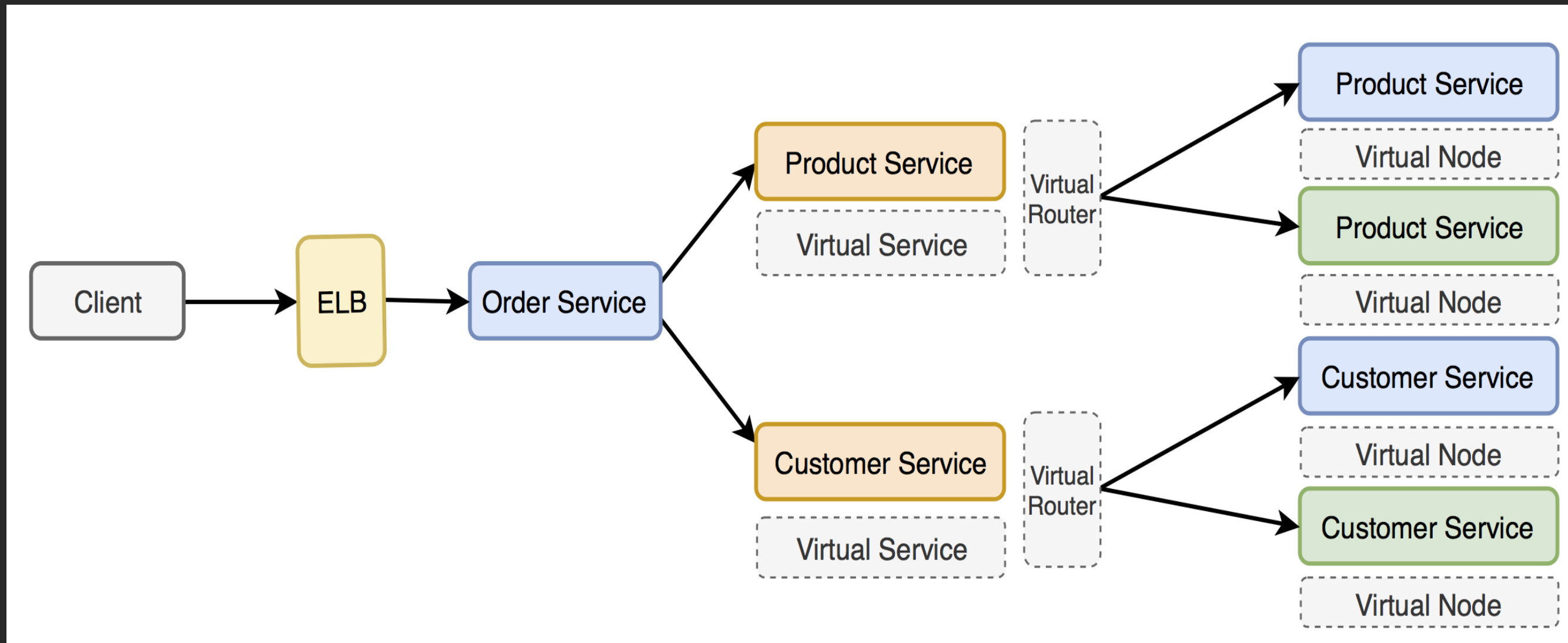
periodSeconds: 3

Demo

Building complex use cases on top of this

Canary Deployments with AWS App Mesh on Amazon EKS

<https://github.com/aws-samples/eks-canary-deployment-stepfunction>



Building complex use cases on top of this

Evaluate Flagger (<https://flagger.app>). It is a Kubernetes operator that automates the promotion of canary deployments using Istio, Linkerd, AWS App Mesh, NGINX or Gloo routing for traffic shifting and Prometheus metrics for canary analysis. The canary analysis can be extended with webhooks for running system integration/acceptance tests, load tests, or any other custom validation.

AWS Container Day - Weaveworks Partner Demo

https://www.youtube.com/watch?time_continue=26&v=IV5VXkAzPw0

Thank you!

Yang Yang

Vijay Khanna



Please complete the session
survey in the mobile app.