



Front-end Application Deployment Patterns

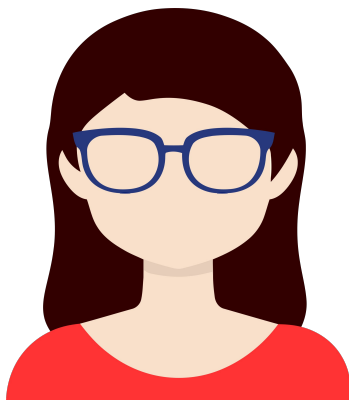
Ross Kukulinski - Sr. Product Manager

 @rosskukulinski

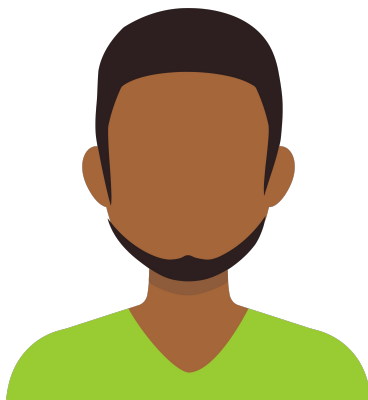
Kubernetes Personas



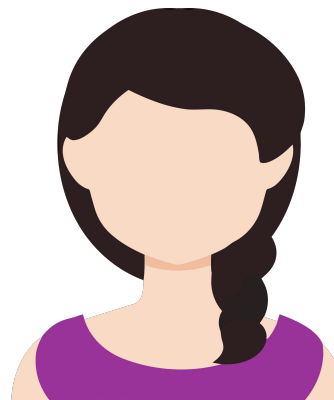
What about me?



Platform Engineer



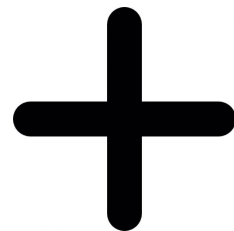
Backend Architect



CIO



Web Developer





Deployment Patterns

Deployment patterns



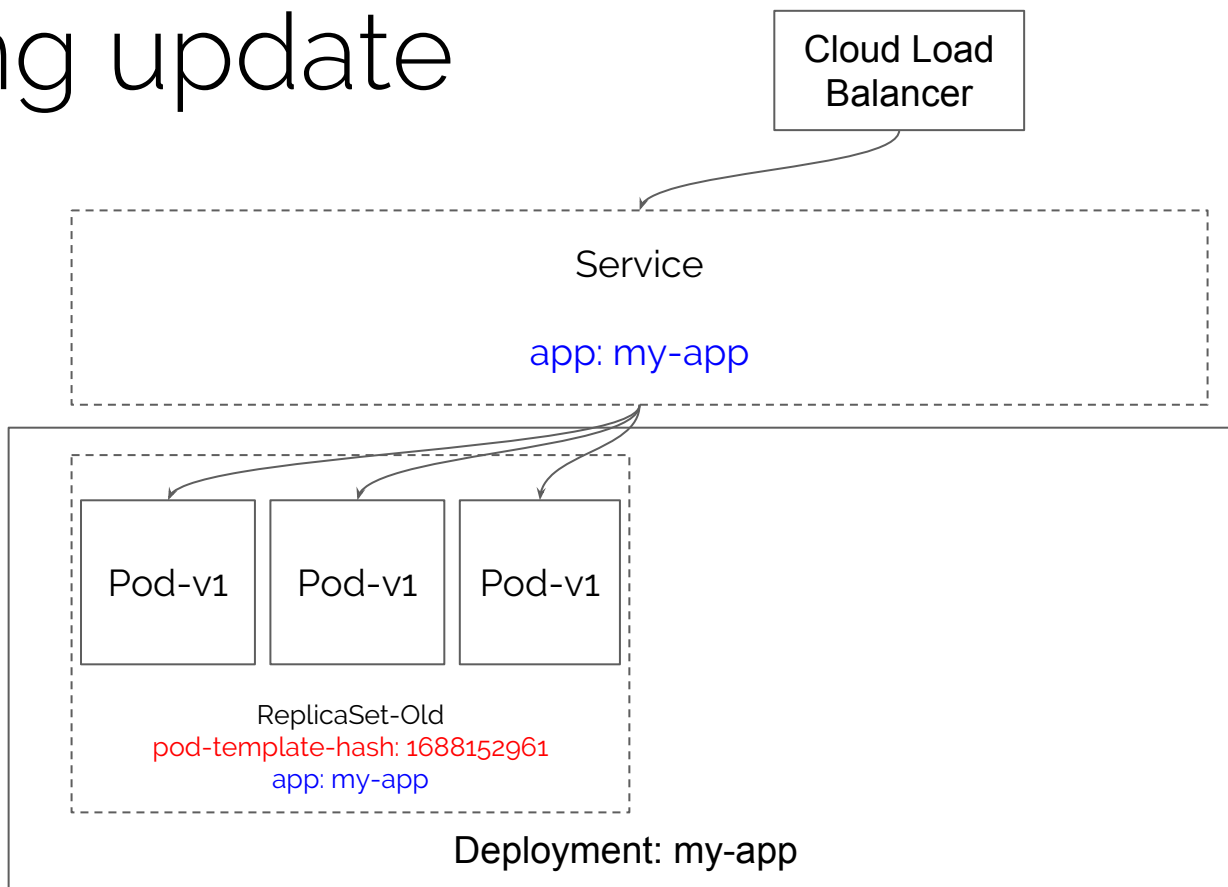
- Edit files in production
- Rolling update
- Blue-Green
- Canary

Deployment patterns

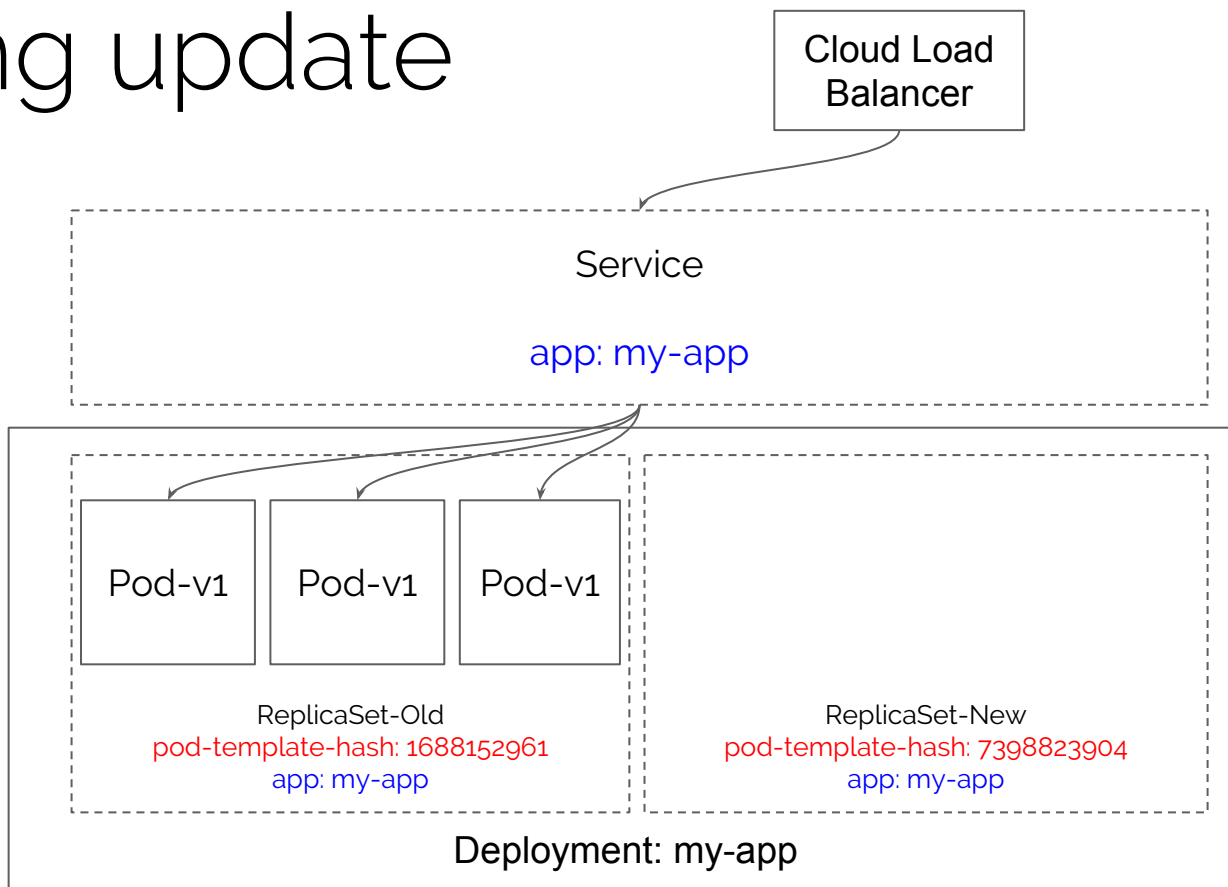


- ~~Edit files in production~~
- Rolling update
- Blue-Green
- Canary

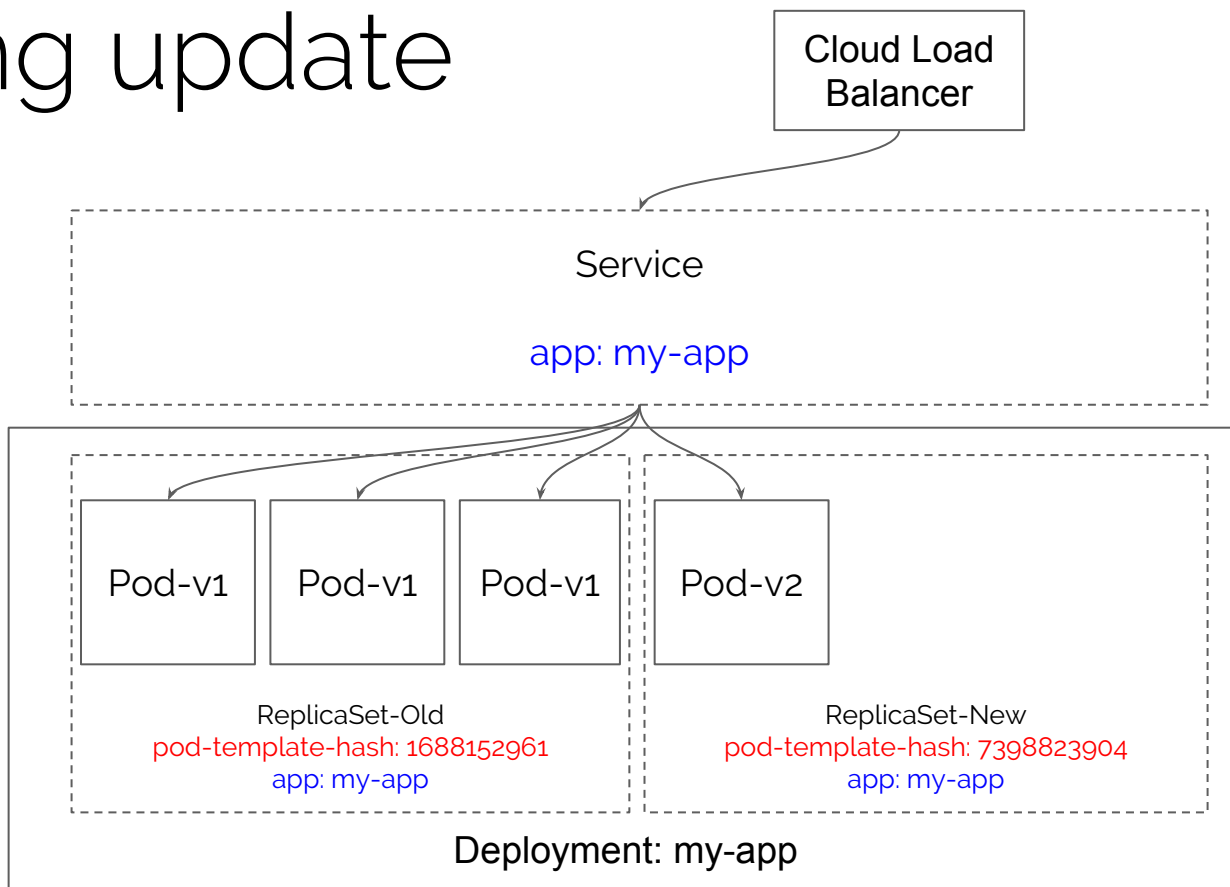
Rolling update



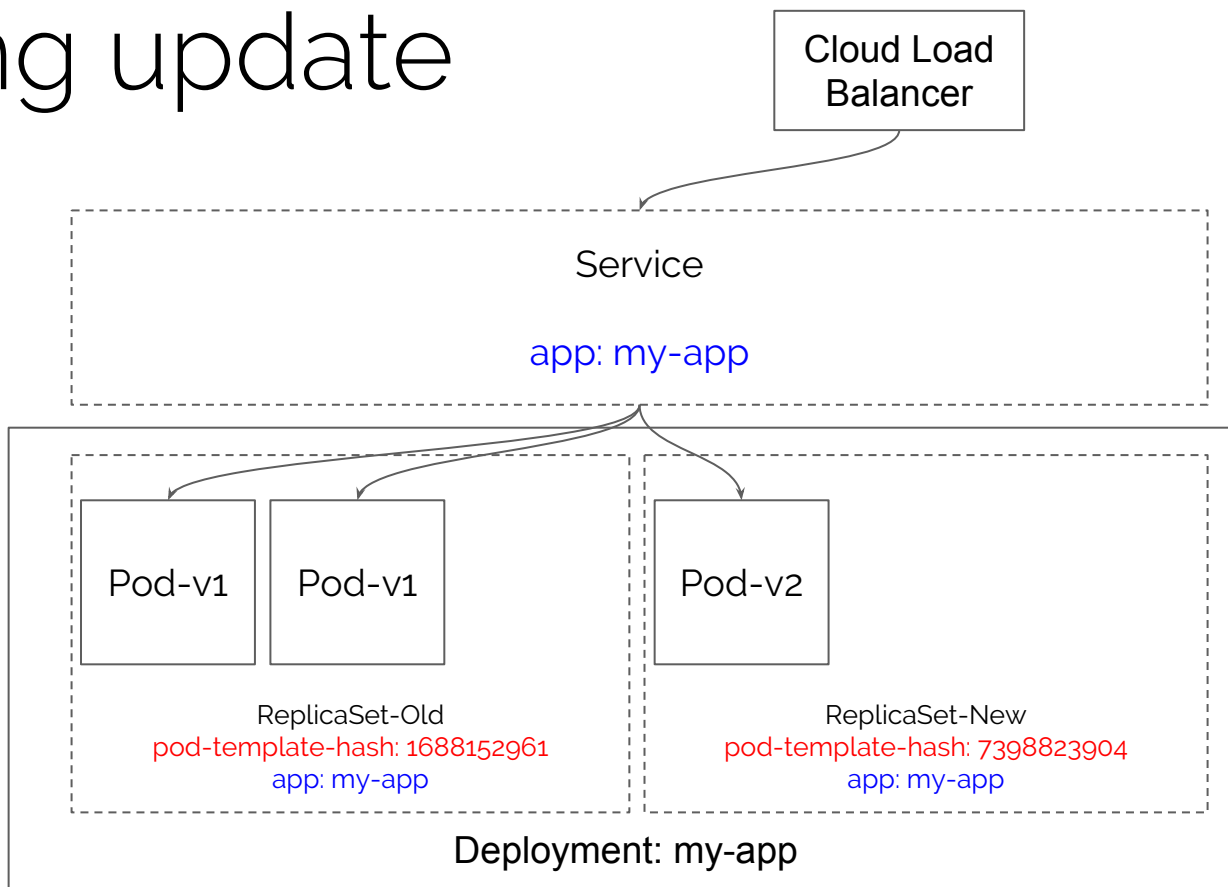
Rolling update



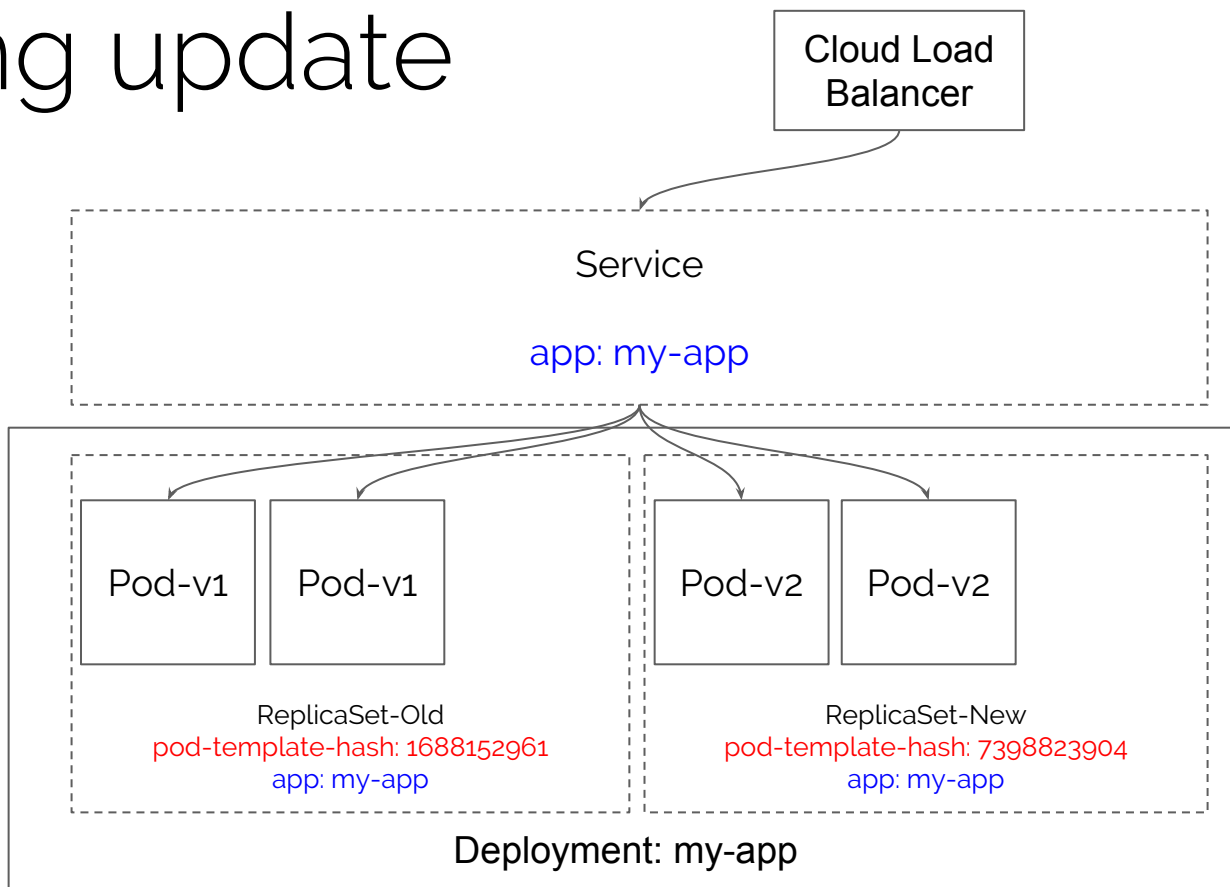
Rolling update



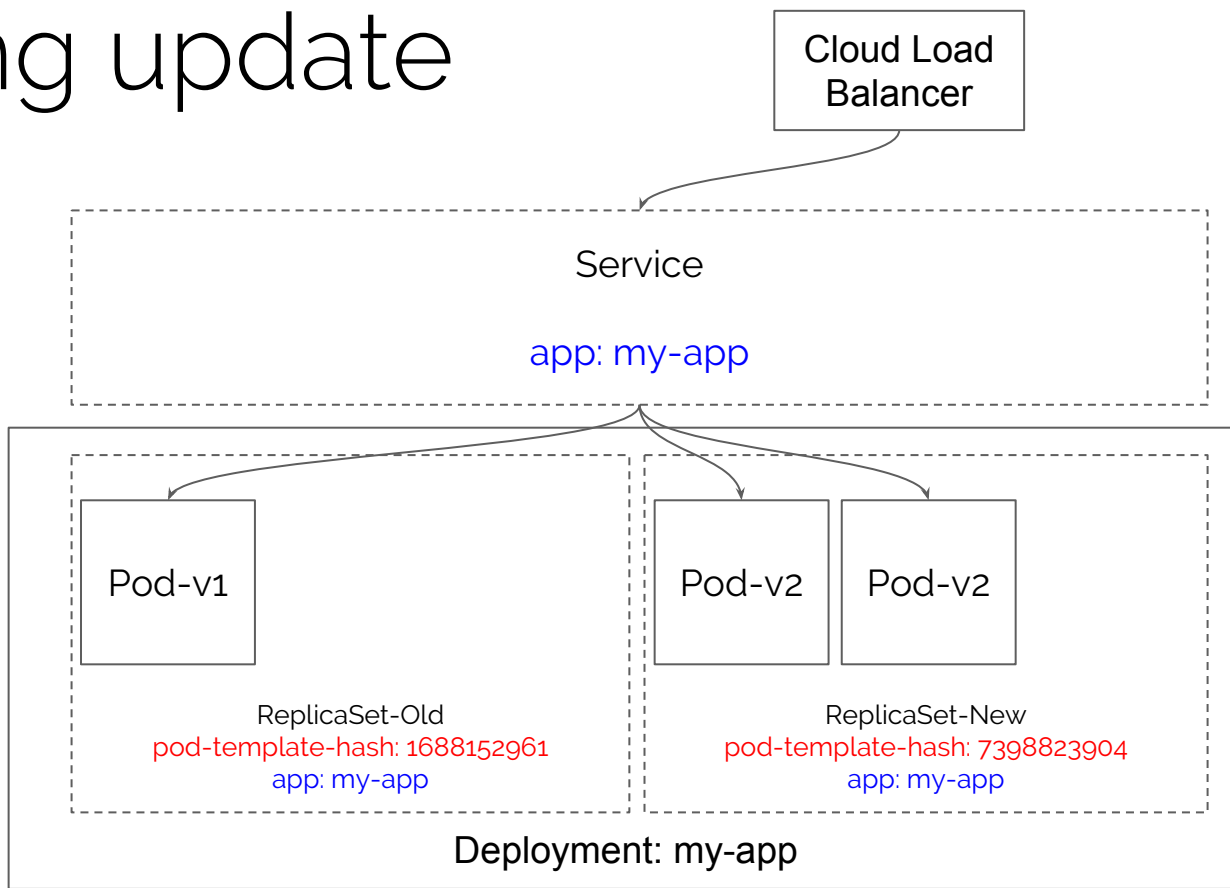
Rolling update



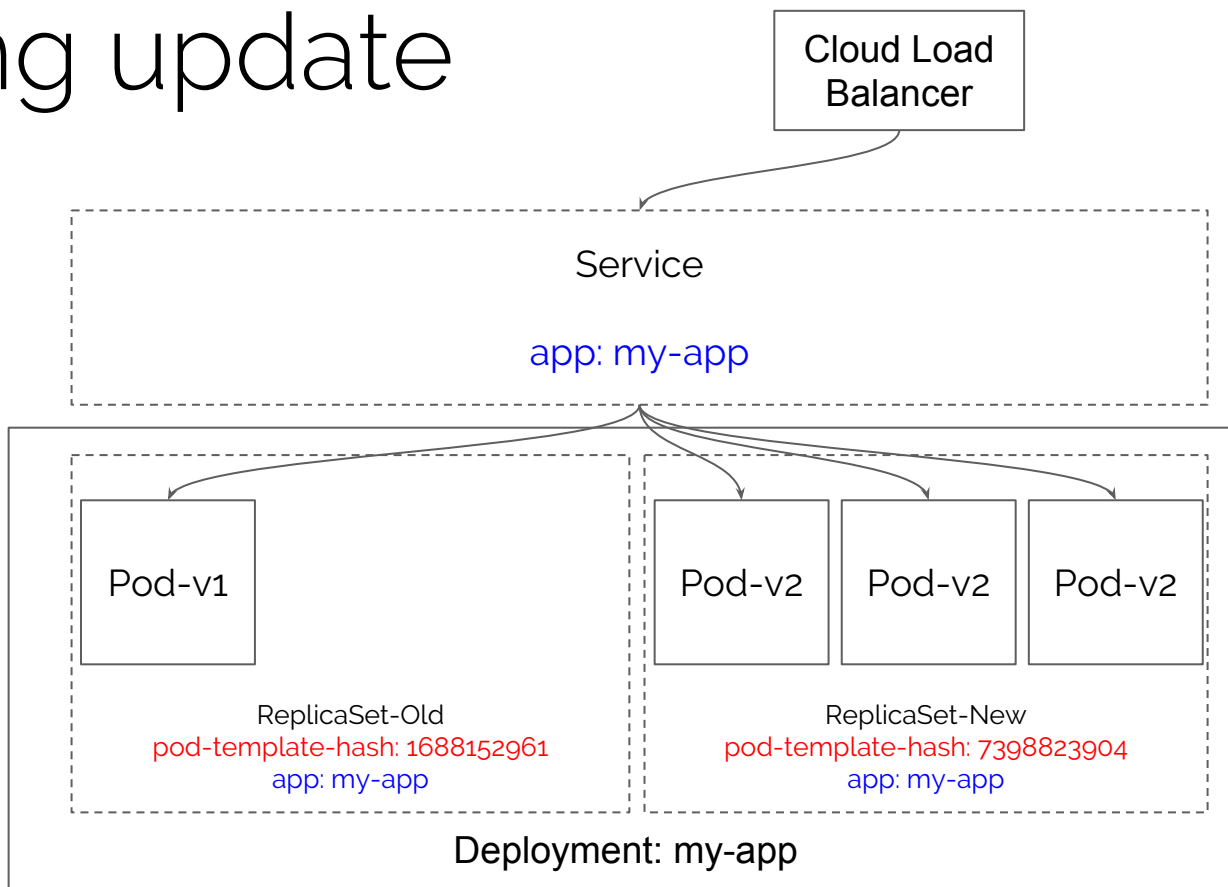
Rolling update



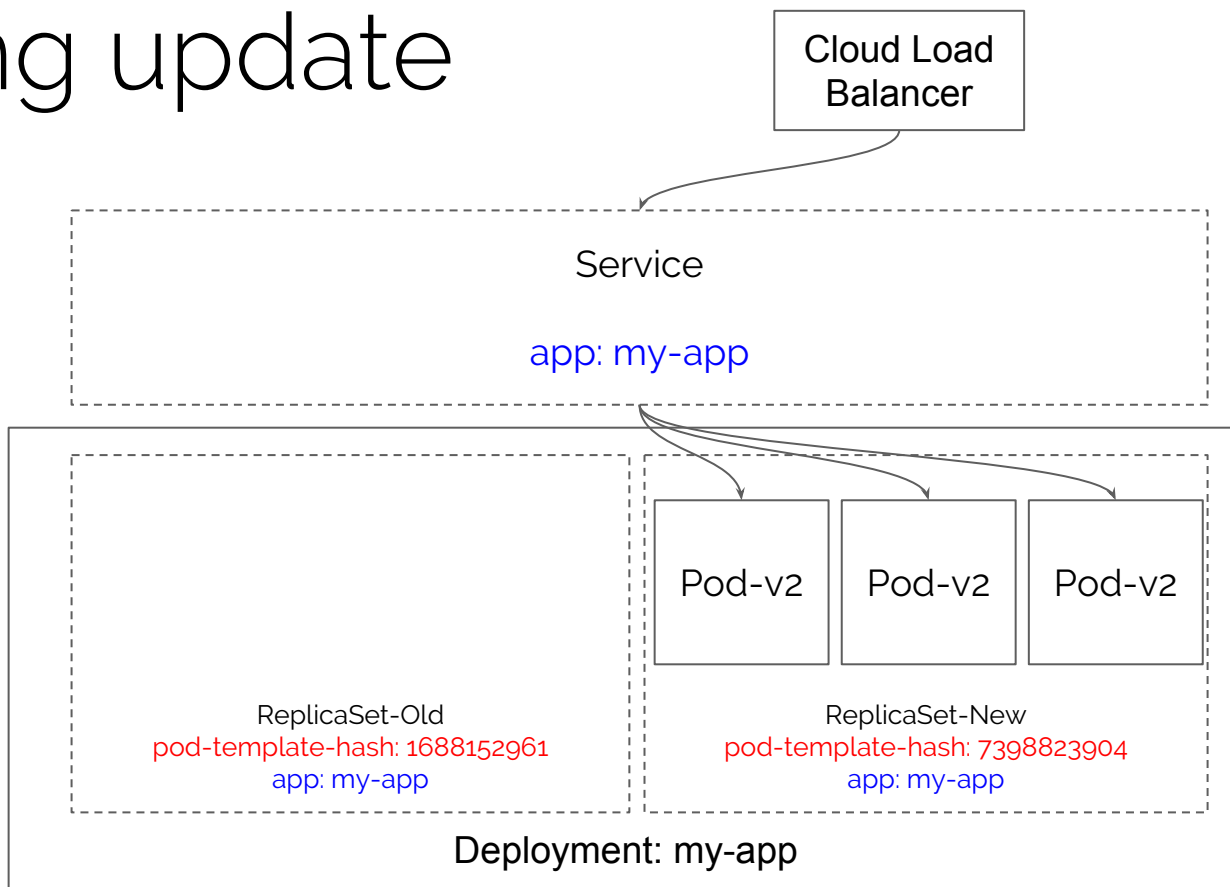
Rolling update



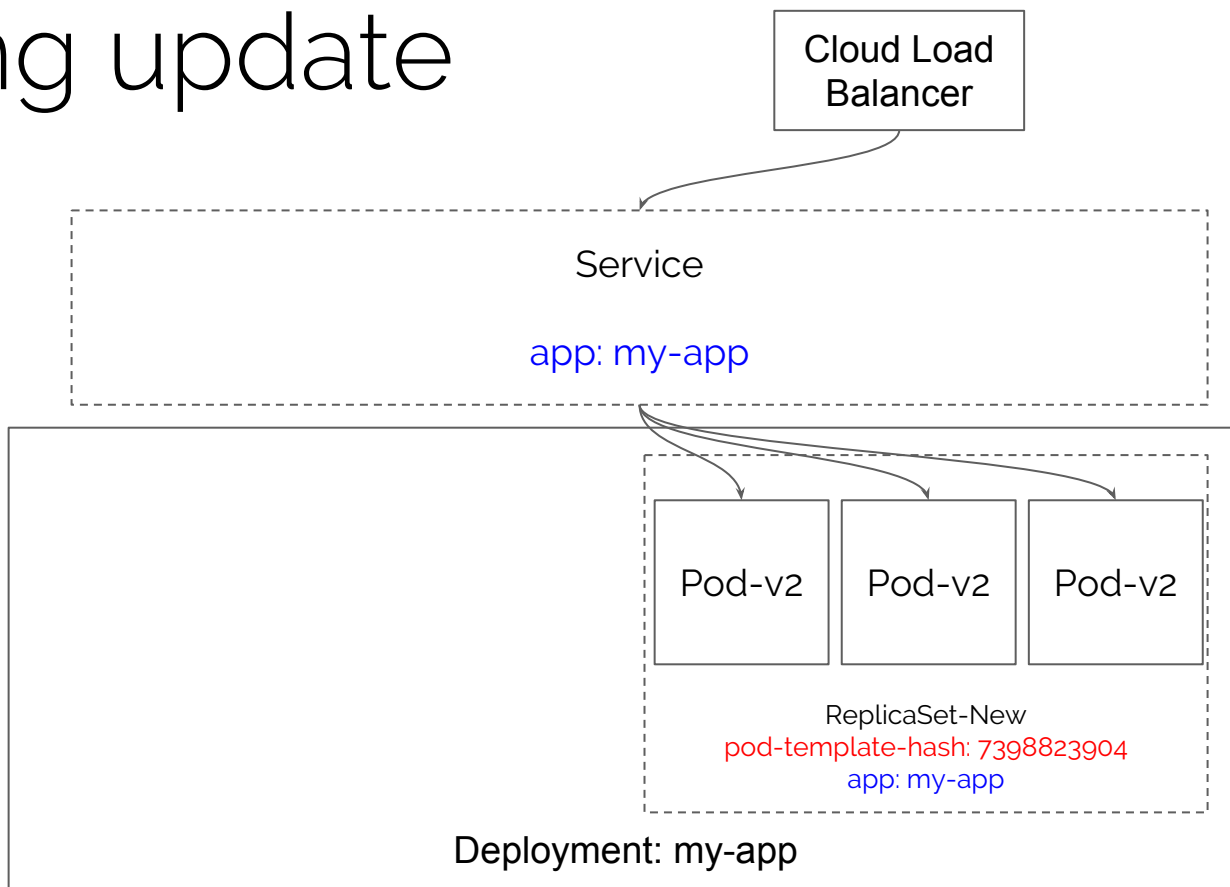
Rolling update



Rolling update



Rolling update



```
$ kubectl explain \  
deployment.spec.strategy.rollingUpdate
```


HTML



CSS

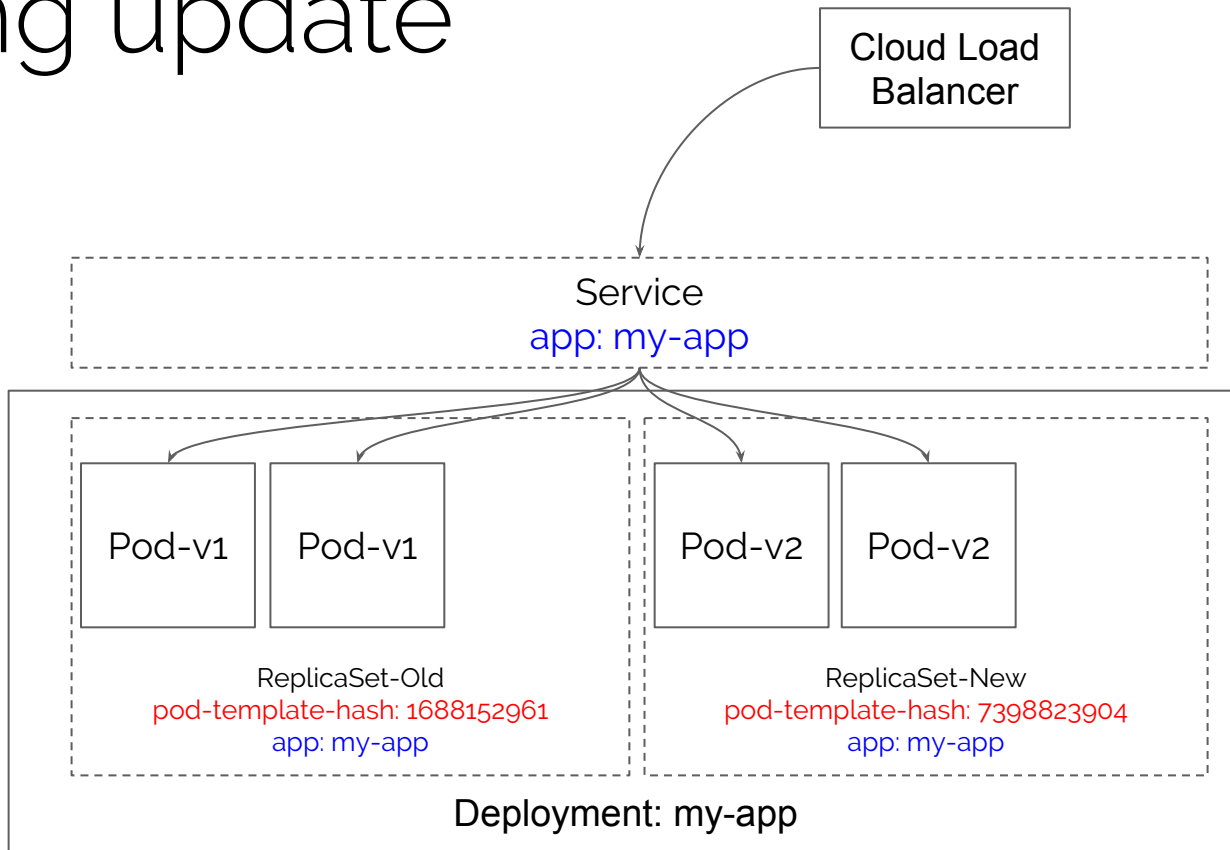


JS

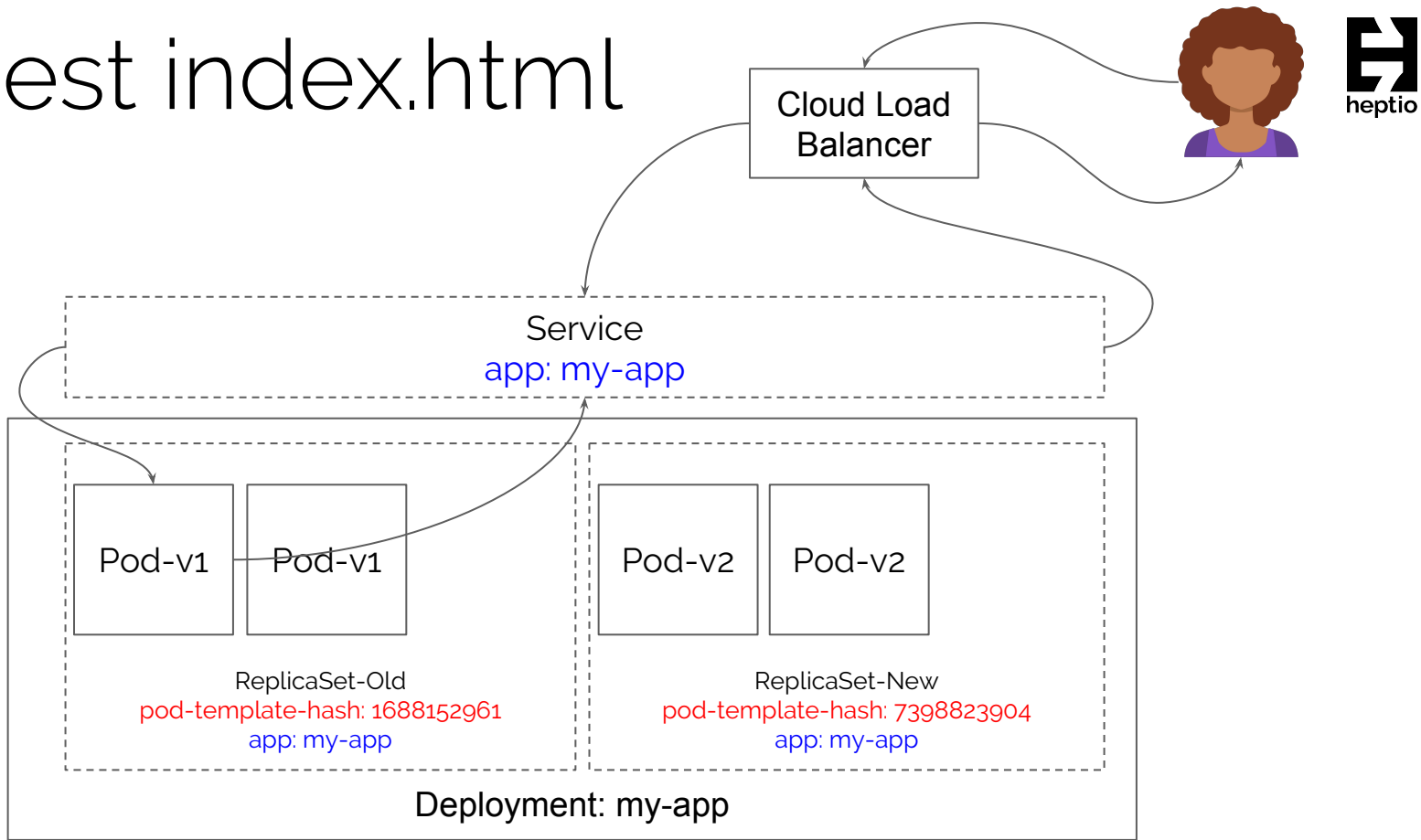


?

Rolling update



Request index.html

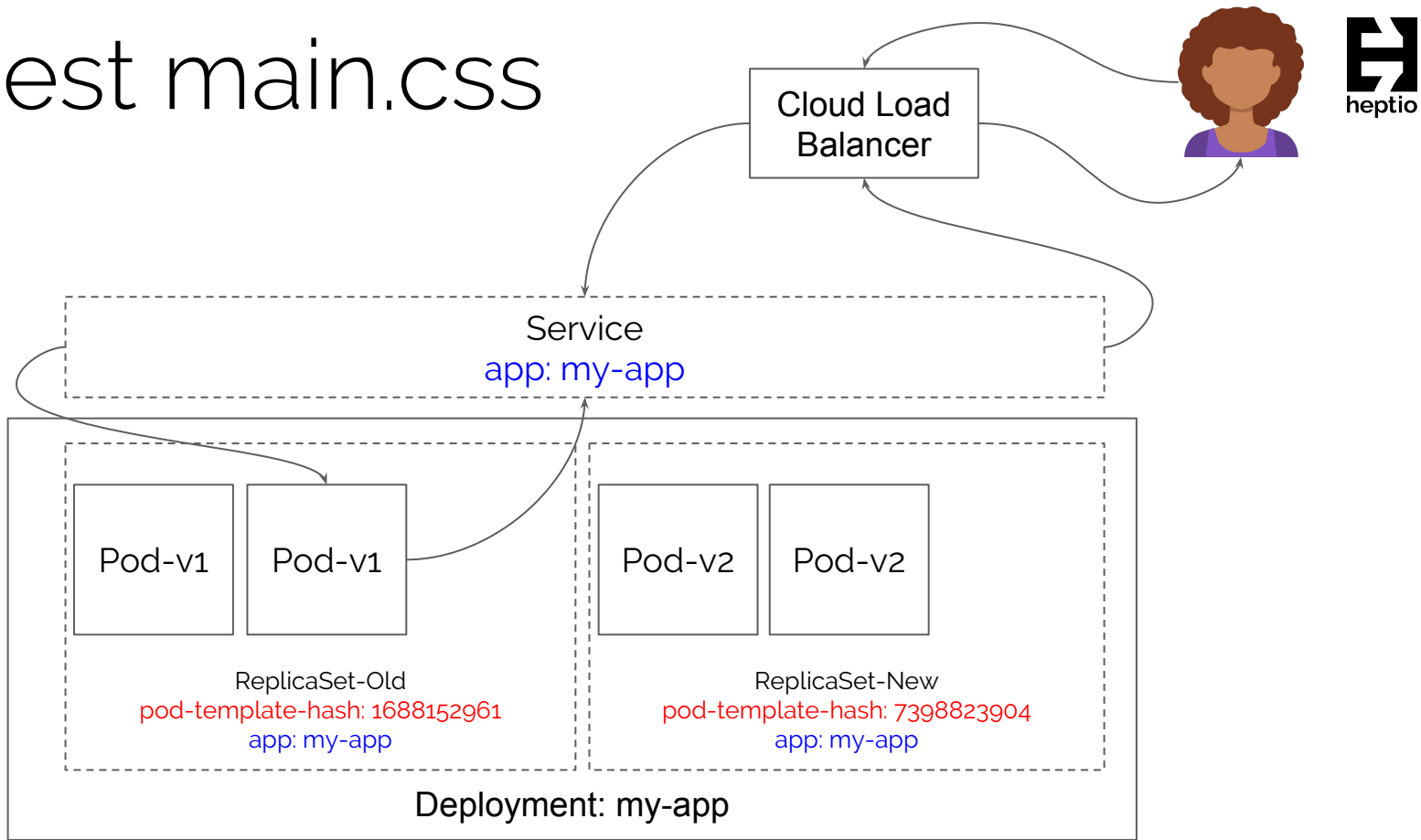


index.html

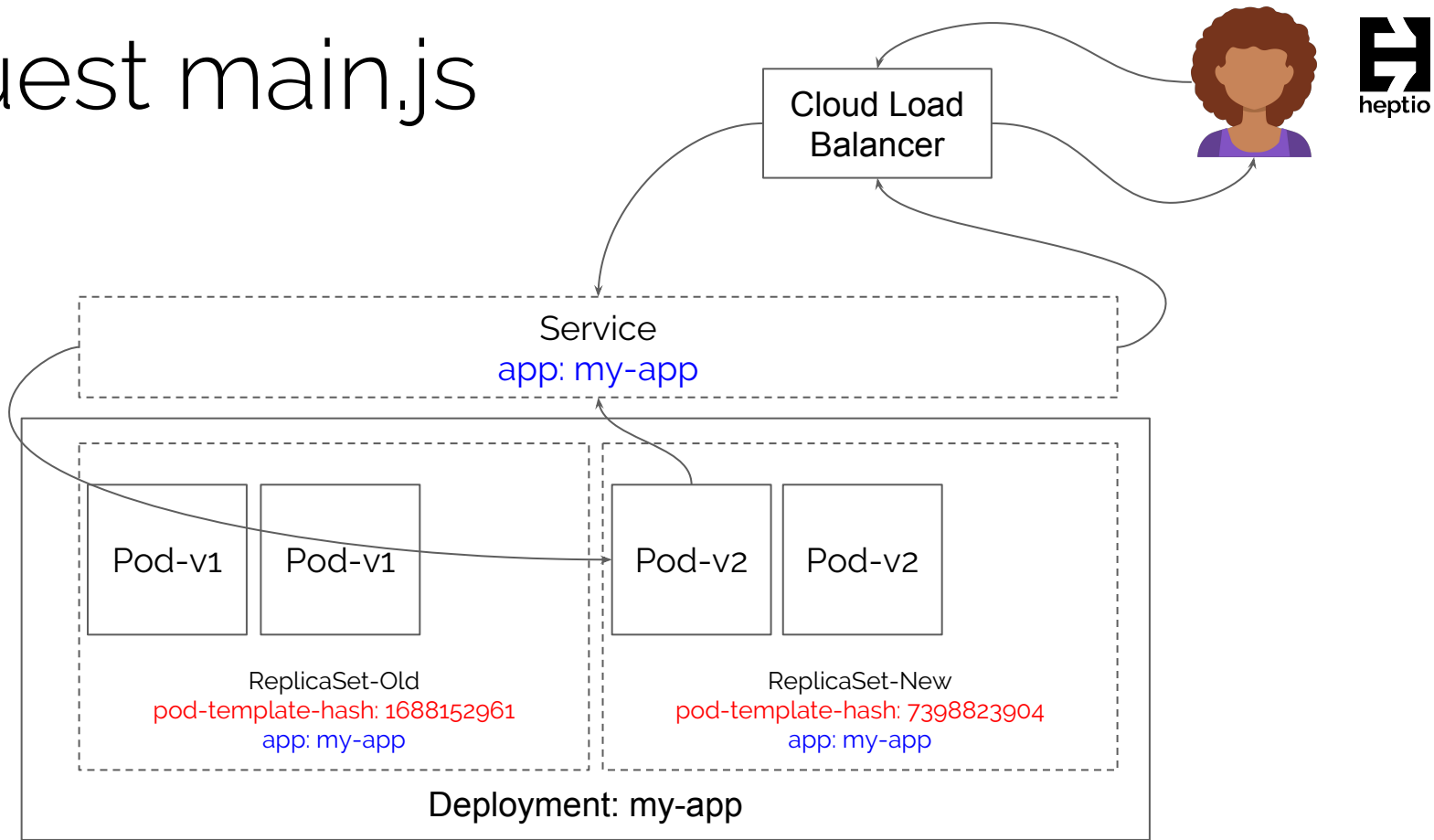


```
<html>  
  <head>  
    <link href="main.css" rel="stylesheet"></link>  
    <script src="main.js"></script>  
  </head>
```

Request main.css

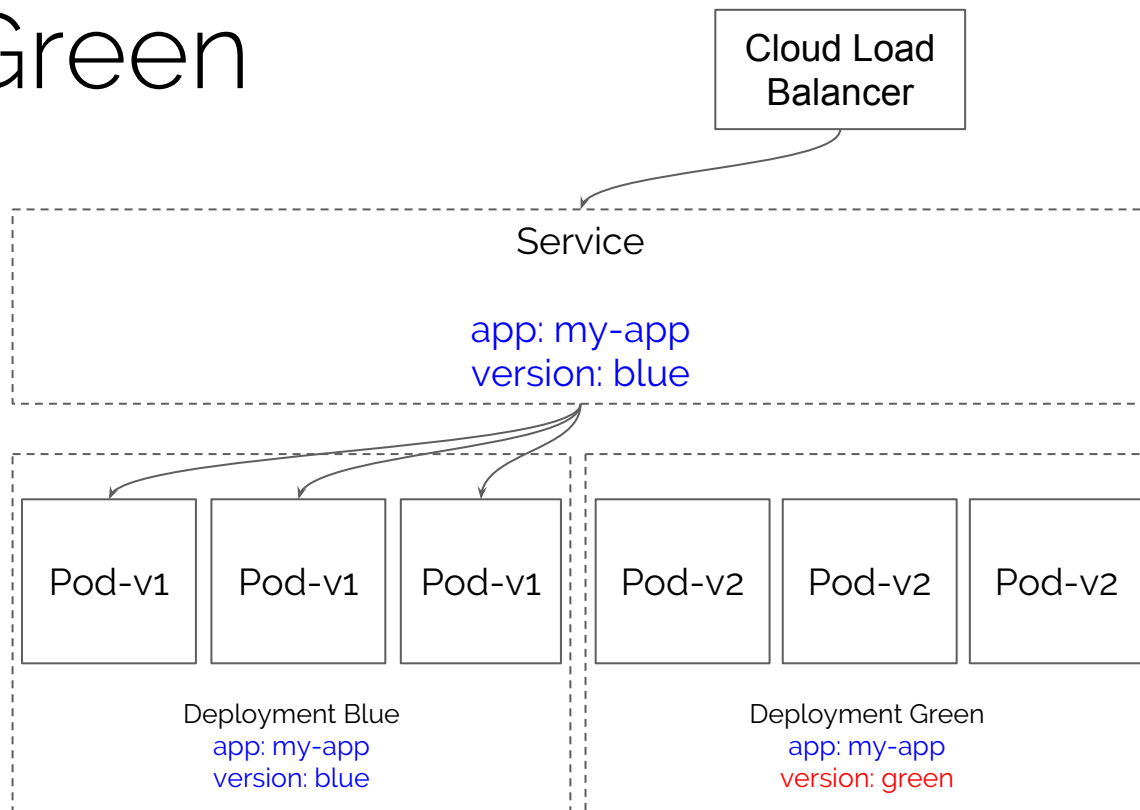


Request main.js

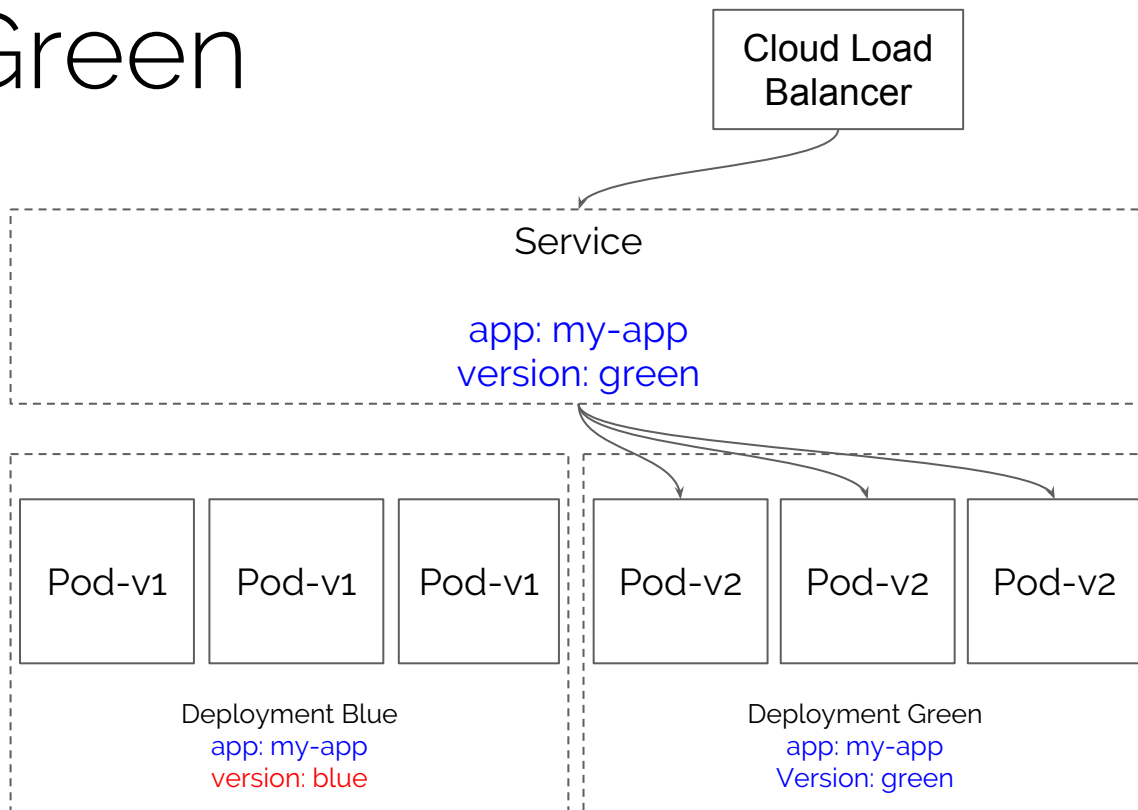




Blue-Green



Blue-Green



HTML



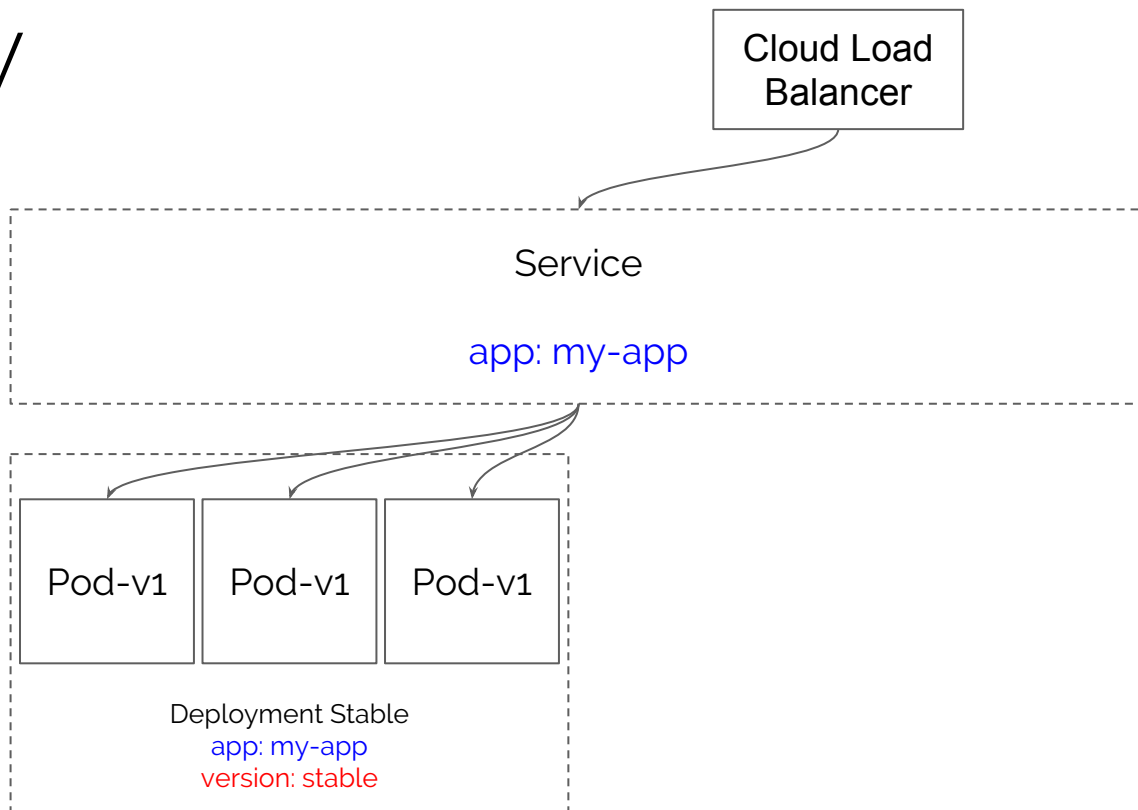
CSS



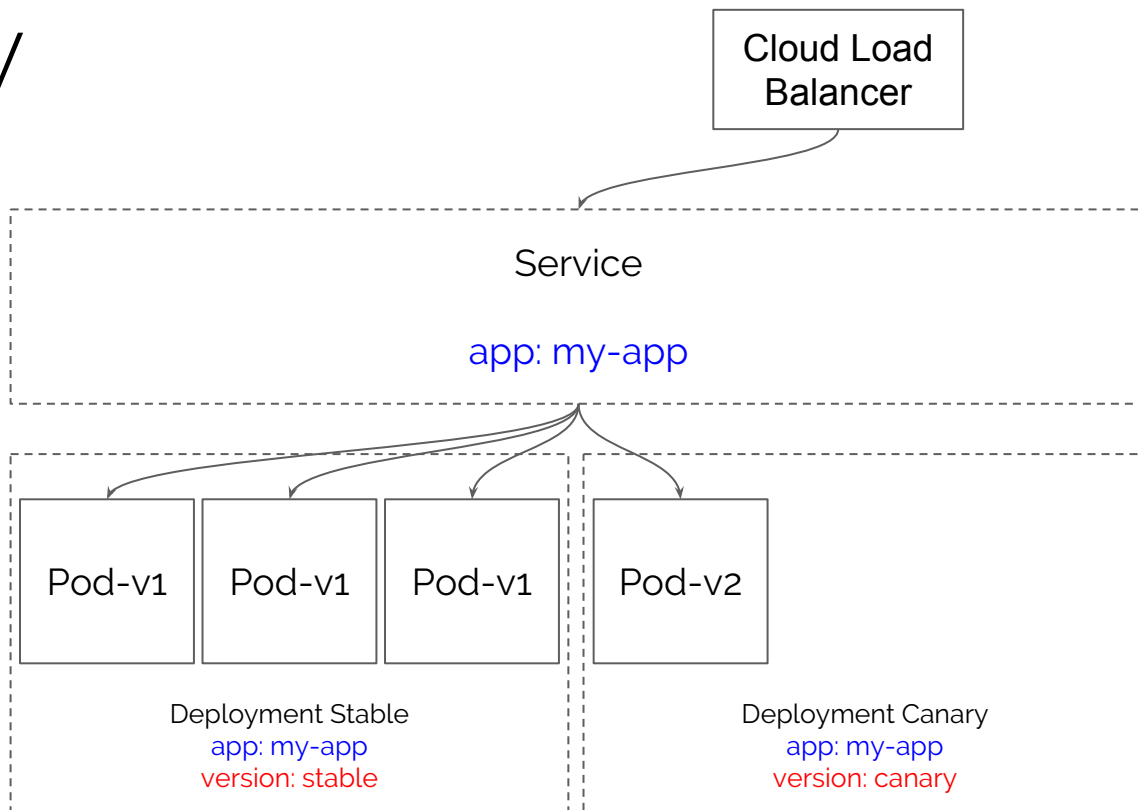
JS



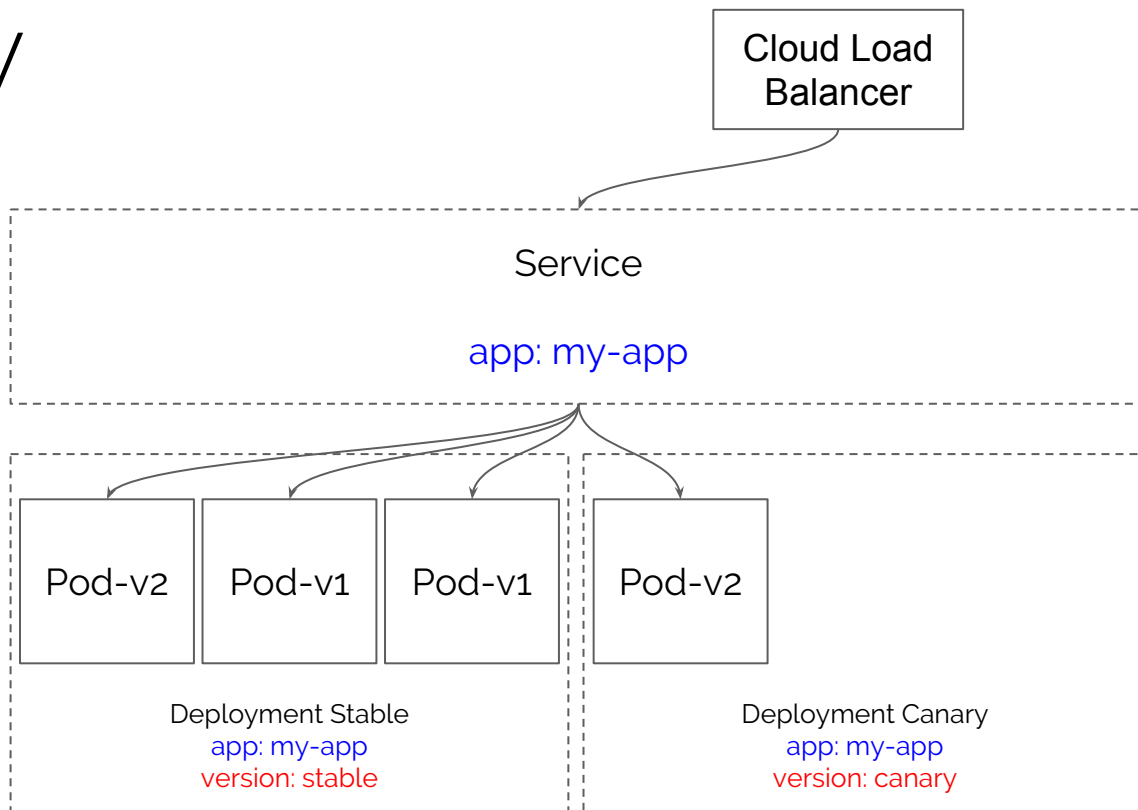
Canary



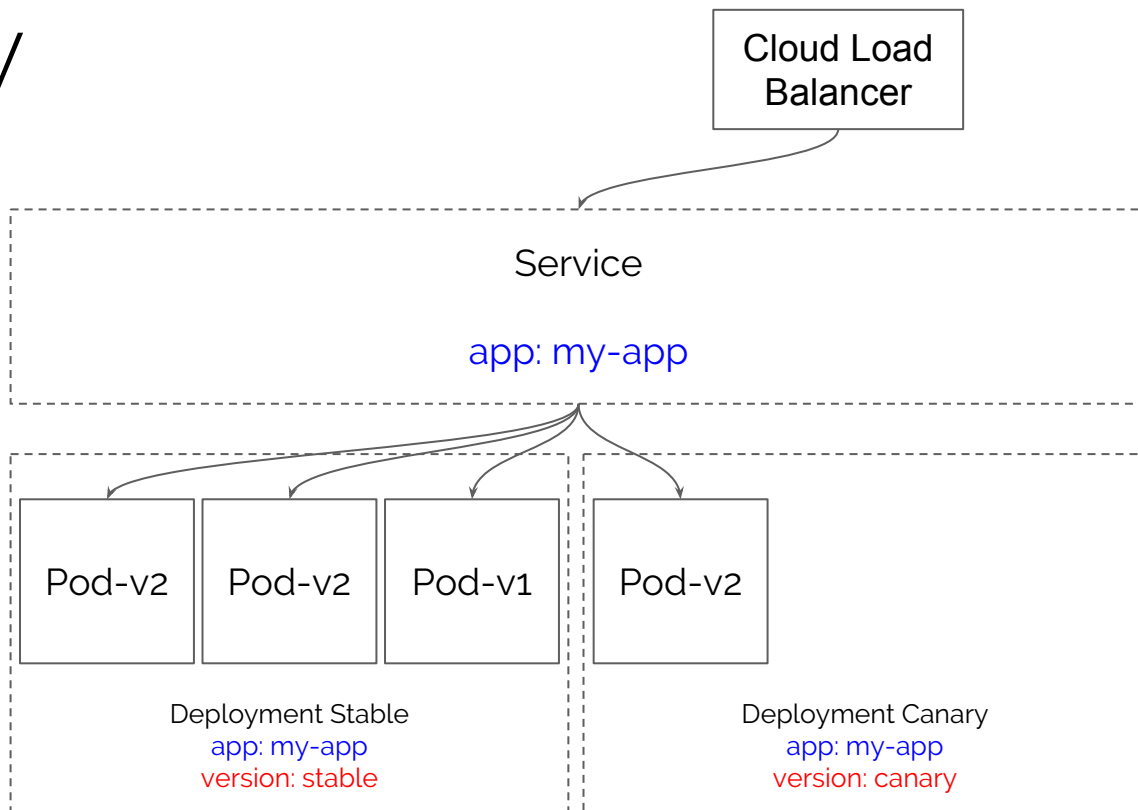
Canary



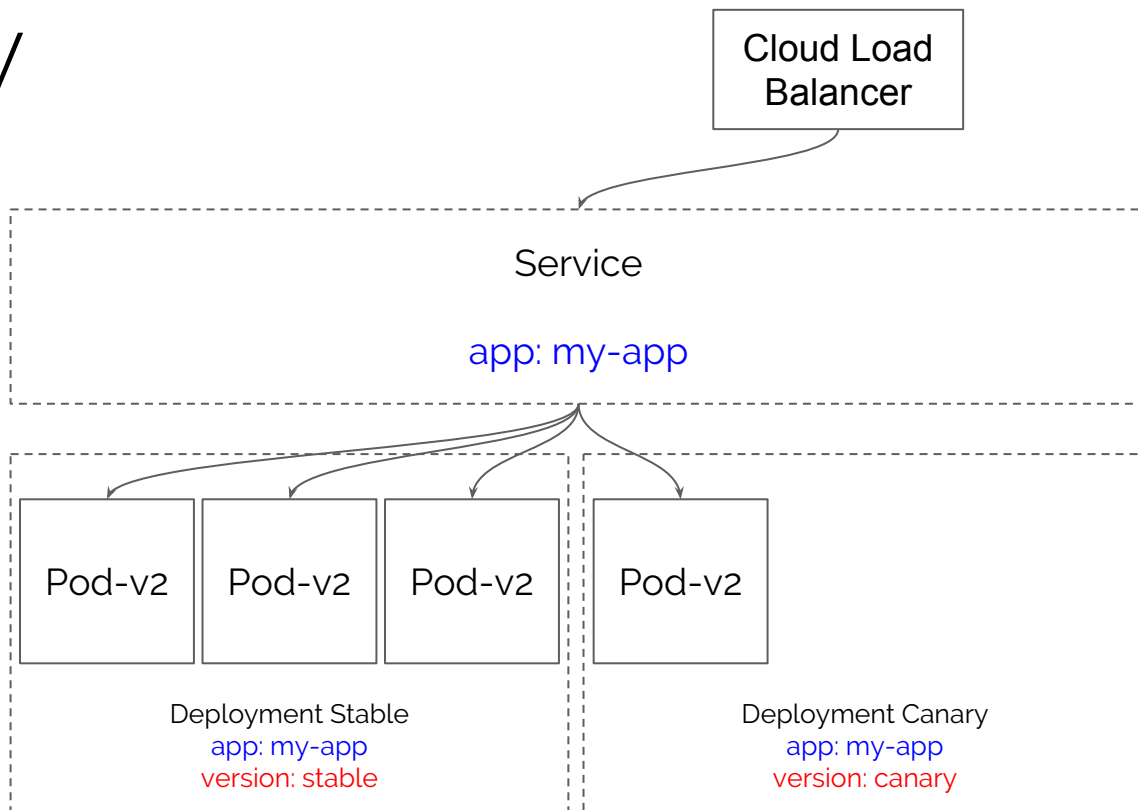
Canary



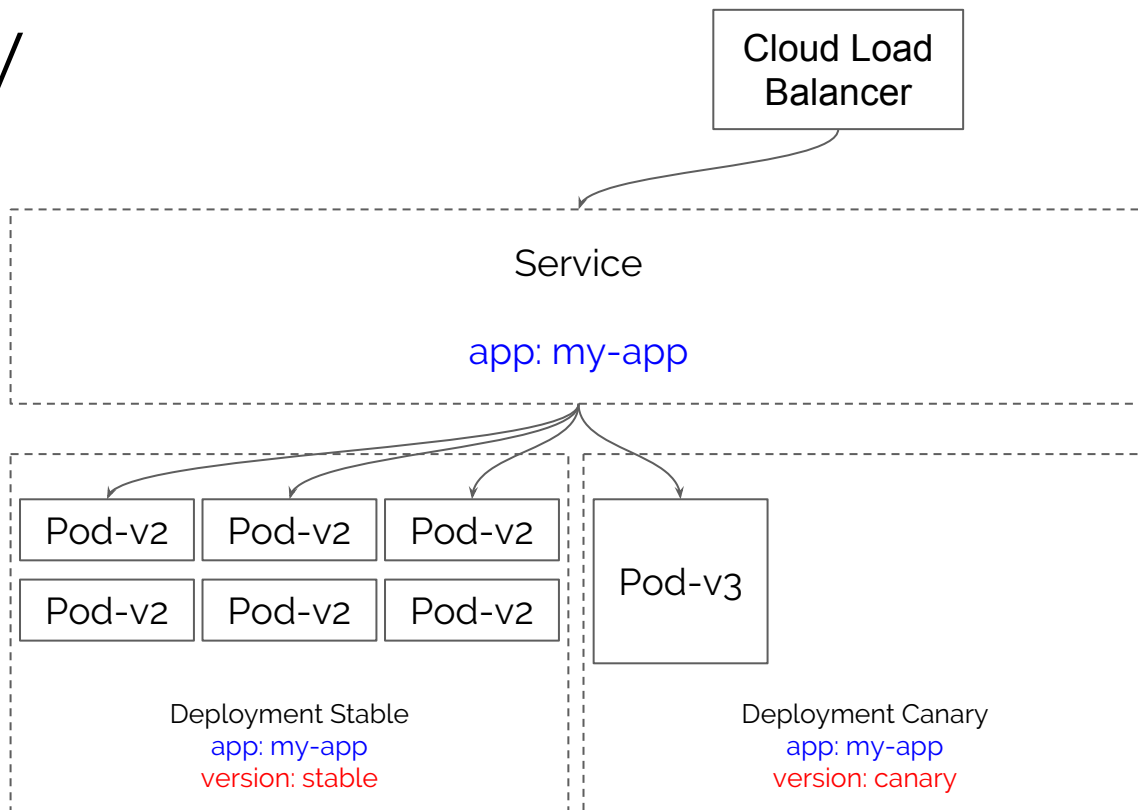
Canary



Canary



Canary



HTML



CSS



JS



Deployment Recap

- Rolling update
 - Requires fewer computing resources
 - Broken page loads and/or missing static assets
- Blue-Green
 - Near-instantaneous traffic switch
 - Requires double compute resources
- Canary
 - Great for validating changes on smaller audience
 - Broken page loads and/or missing static assets
 - Limited weighting control via Kubernetes Deployments



Workarounds

Ignore your problems

Scheduled maintenance window

Versioned assets via CDN

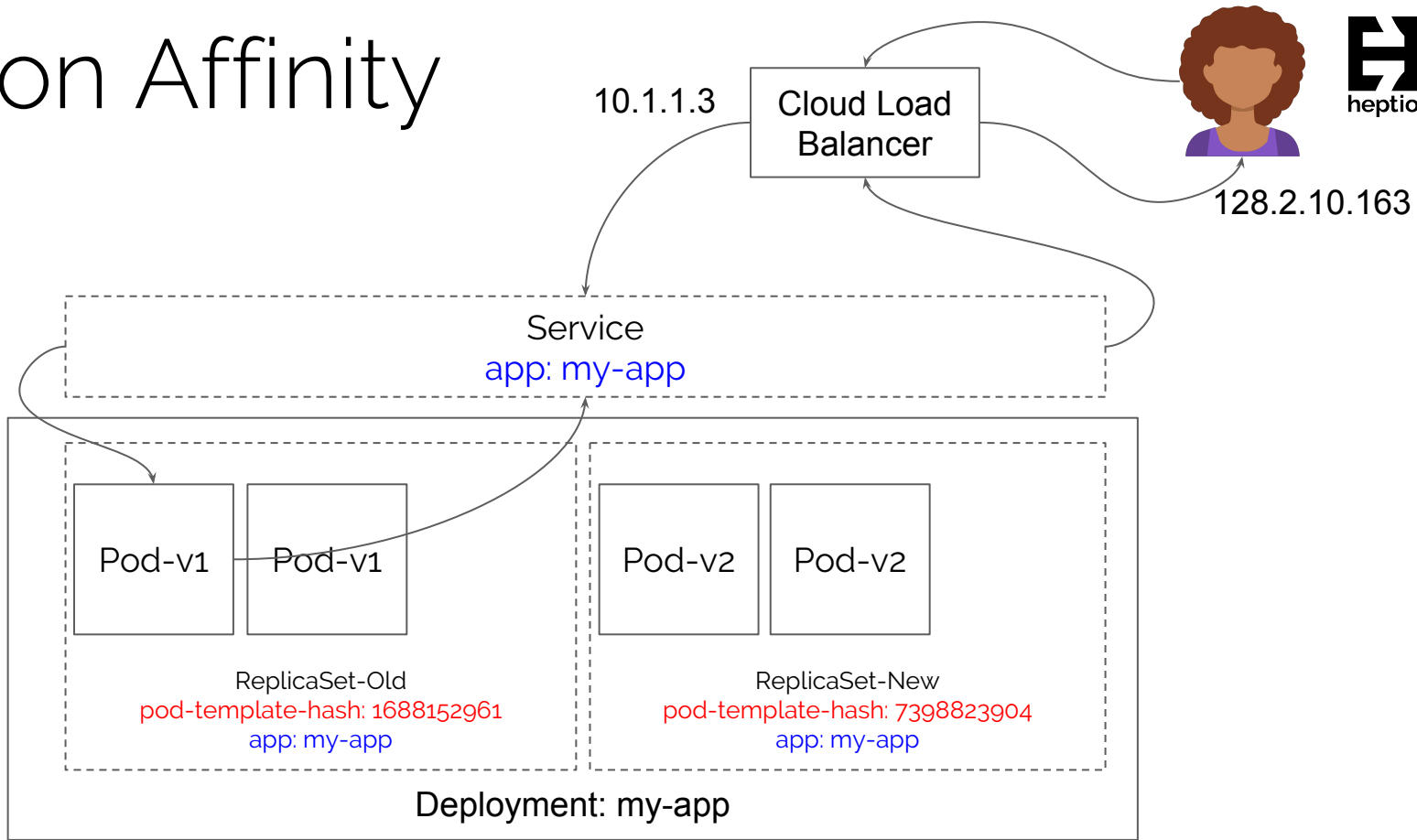


```
<link
  href="https://app.s3.amazonaws.com/main.v1.css"
  rel="stylesheet">
</link>
<script
  src="https://app.s3.amazonaws.com/main.v1.js">
</script>
```

Session affinity

```
kind: Service
apiVersion: v1
spec:
  selector:
    app: my-app
  sessionAffinity: ClientIP
...
```

Session Affinity



HTTP/2



- Single multiplexed connection
- Binary protocol, instead of text based
- Reduced additional RTT

Load balancers



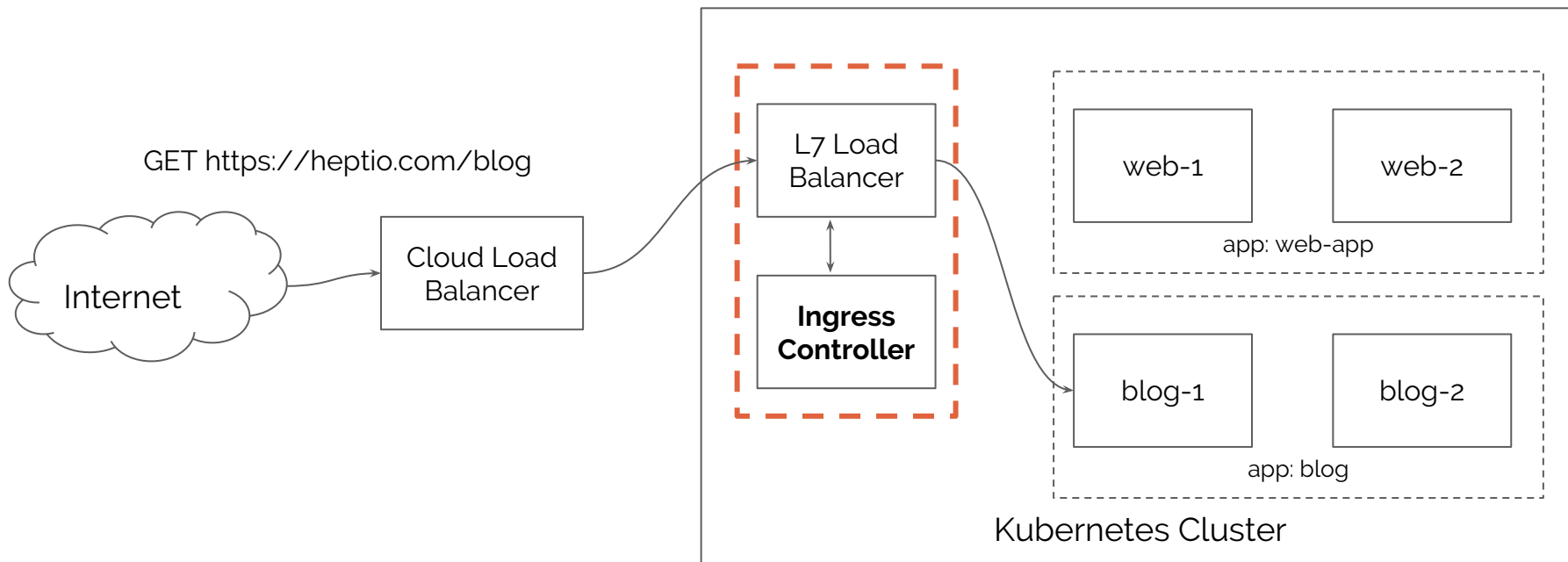


Kubernetes Ingress

Ingress provides a way to route requests to Services based on the request host or path

Layer 7 Load Balancing

Kubernetes Ingress Controller



Nginx, Contour, Traefik, GCE,
F5 bigip, Haproxy, and more!



Heptio Contour

Deploy and manage Envoy as an incoming load balancer.



- Envoy as the data plane
- Co-development project Yahoo Japan Corporation subsidiary, Actapio
- Dynamically updates without dropped connections
- Performance tested to millions of concurrent connections

Ingress API is limited

- Only one Service per path
- Can't weight shift traffic across multiple Services
- Can't force http -> https redirect
- Not safe for multi-team use
- And much more...

IngressRoute CRD



```
# ingressroute.yaml
# https://github.com/heptio/contour/blob/master/docs/ingressroute.md
apiVersion: contour.heptio.com/v1beta1
kind: IngressRoute
metadata:
  name: basic
spec:
  virtualhost:
    fqdn: www.heptio.com
  routes:
    - match: /
      services:
        - name: heptio-website
          port: 80
```

IngressRoute CRD



```
# ingressroute.yaml
# https://github.com/heptio/contour/blob/master/docs/ingressroute.md
apiVersion: contour.heptio.com/v1beta1
kind: IngressRoute
metadata:
  name: basic
spec:
  virtualhost:
    fqdn: www.heptio.com
  routes:
    - match: /
      services:
        - name: heptio-website
          port: 80
```

IngressRoute CRD



```
# ingressroute.yaml
# https://github.com/heptio/contour/blob/master/docs/ingressroute.md
apiVersion: contour.heptio.com/v1beta1
kind: IngressRoute
metadata:
  name: basic
spec:
  virtualhost:
    fqdn: www.heptio.com
  routes:
    - match: /
      services:
        - name: heptio-website
          port: 80
```

Blue-Green Deploys



```
# ingressroute.yaml
# https://github.com/heptio/contour/blob/master/docs/ingressroute.md
apiVersion: contour.heptio.com/v1beta1
kind: IngressRoute
metadata:
  name: basic
spec:
  virtualhost:
    fqdn: www.heptio.com
  routes:
    - match: /
      services:
        - name: heptio-website-green
          port: 80
```

Blue-Green Deploys



```
# ingressroute.yaml
# https://github.com/heptio/contour/blob/master/docs/ingressroute.md
apiVersion: contour.heptio.com/v1beta1
kind: IngressRoute
metadata:
  name: basic
spec:
  virtualhost:
    fqdn: www.heptio.com
  routes:
    - match: /
      services:
        - name: heptio-website-blue
          port: 80
```

Multiple Upstreams

```
# ingressroute.yaml
# https://github.com/heptio/contour/blob/master/docs/ingressroute.md
apiVersion: contour.heptio.com/v1beta1
kind: IngressRoute
metadata:
  name: multiple-upstreams
spec:
  virtualhost:
    fqdn: www.heptio.com
  routes:
    - match: /
      services:
        - name: heptio-website-v1
          port: 80
        - name: heptio-website-v2
          port: 80
```

Flexible weighting rules



```
apiVersion: contour.heptio.com/v1beta1
kind: IngressRoute
metadata:
  name: multiple-upstreams
  namespace: default
spec:
  virtualhost:
    fqdn: www.heptio.com
  routes:
    - match: /
      services:
        - name: heptio-website-v1
          port: 80
          weight: 90
        - name: heptio-website-v2
          port: 80
          weight: 10
```

Versioned routes



```
apiVersion: contour.heptio.com/v1beta1
kind: IngressRoute
metadata:
  name: multiple-upstreams
  namespace: default
spec:
  virtualhost:
    fqdn: canary.bar.com
  routes:
    - match: /static/v1/
      services:
        - name: my-app-v1
          port: 80
    - match: /static/v2/
      services:
        - name: my-app-v2
          port: 80
```


Ingress Session Affinity

- Cookie
- Header
- Source-IP Hash
- IngressRoute support not implemented yet ([#361](#))
- NGINX through cookie annotations
 - `nginx.ingress.kubernetes.io/affinity: cookie`

Pulling it all together

- Rolling vs Blue-Green vs Canary
- HTTP/2 and/or Session Affinity
- Learn what your cloud load provider supports
- Versioned static assets via CDN
- Use Kubernetes Ingress
 - Preferably Heptio Contour!



Thank you!

 @rosskukulinski

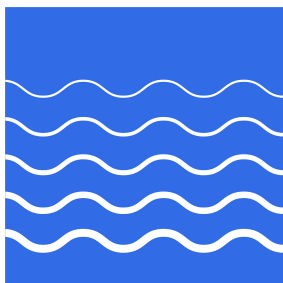
Heptio Booth P8



Heptio
Sonobuoy



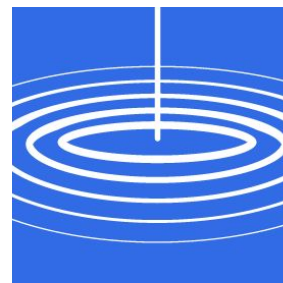
Heptio
Ark



Heptio
Contour



Heptio
Gimbal



Kubernetes
Kubeadm
ClusterAPI AWS



P.S. We're hiring!