

Advanced Ingress for Your Kubernetes Apps with NGINX



by Michael Pleshakov
Software Engineer, F5 Networks

Agenda

Ingress load balancing on Kubernetes

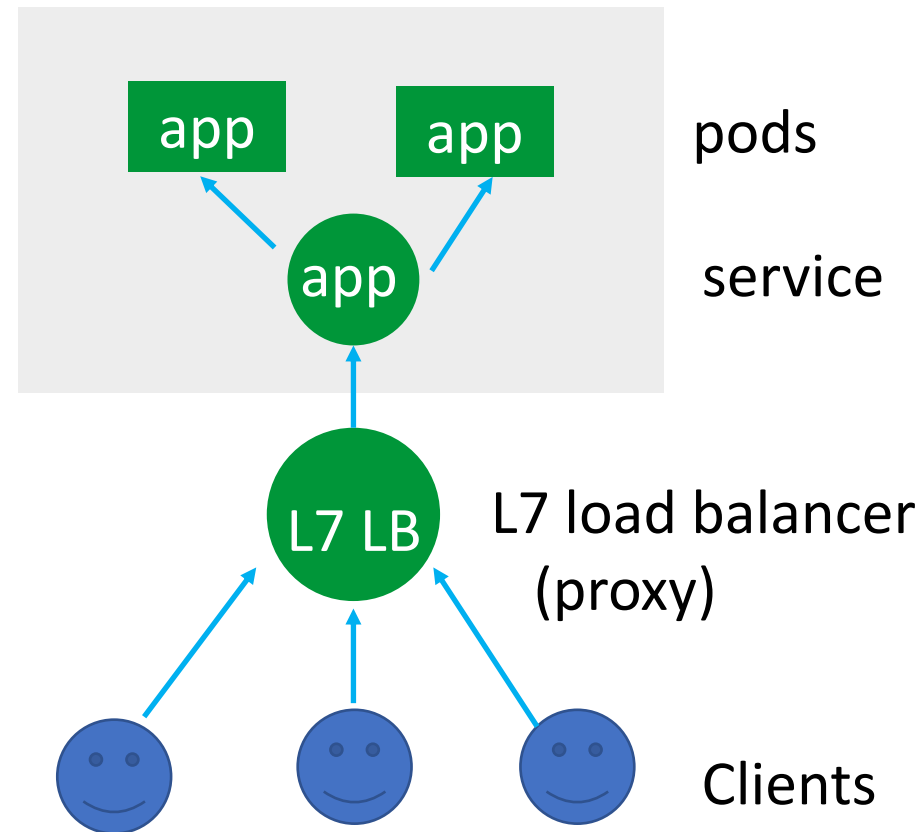
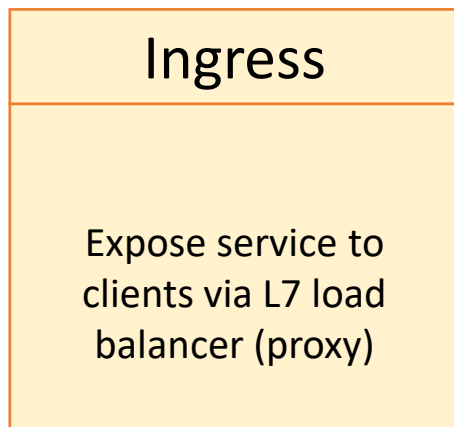
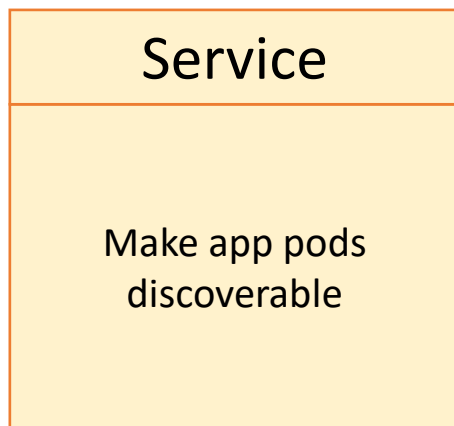
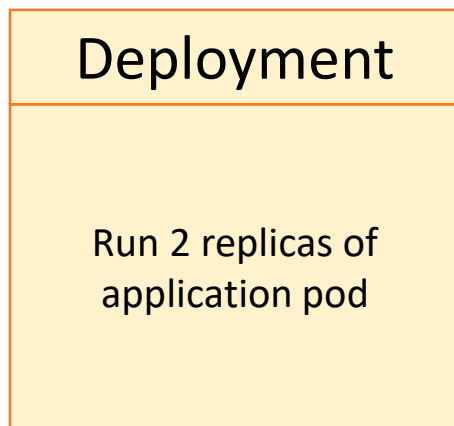
NGINX Ingress Controller

Kubernetes Ingress resource

NGINX Ingress resources

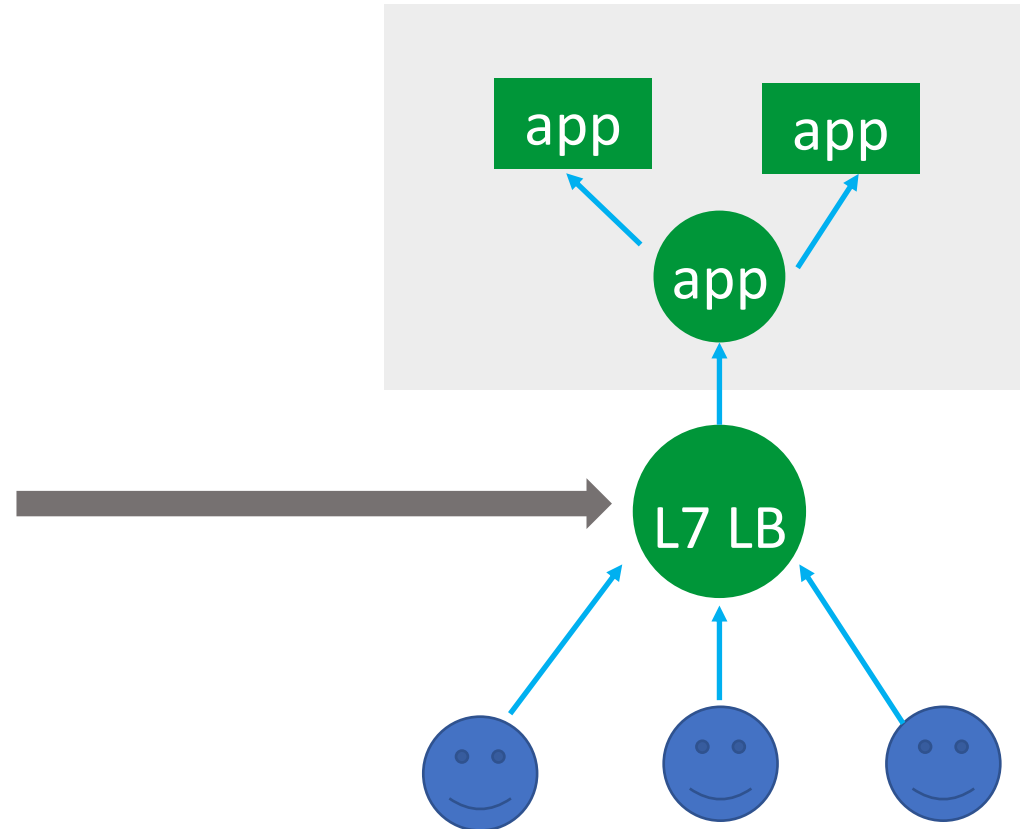
Demos

Running a Web Application on Kubernetes



Ingress Resource

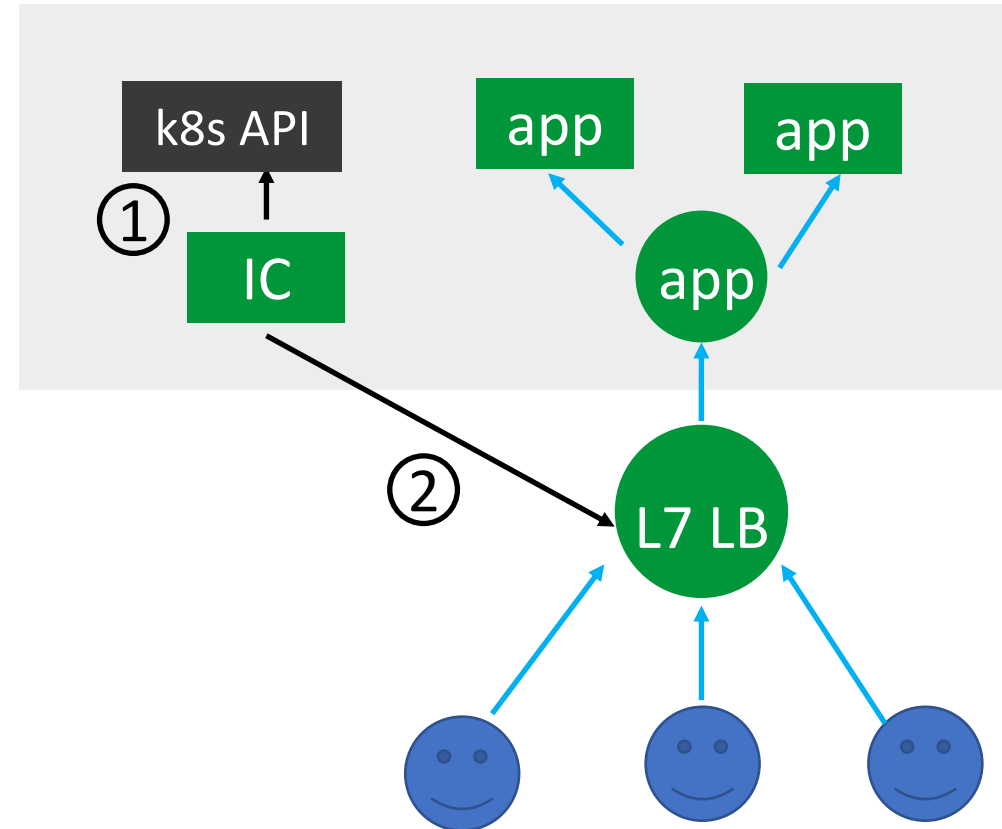
```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cafe-ingress
spec:
  rules:
  - host: webapp.example.com
    http:
      paths:
      - path: /
        backend:
          serviceName: webapp
          servicePort: 80
```



Ingress Controller

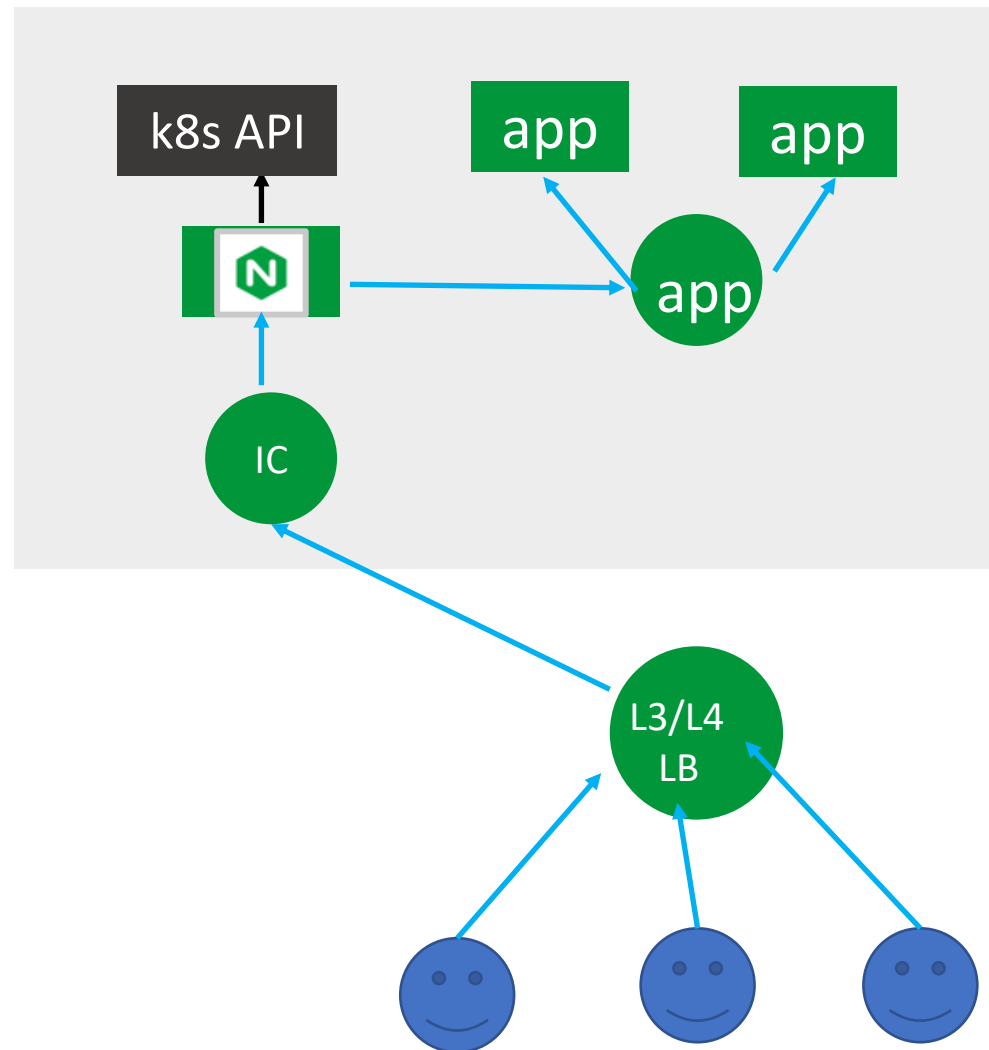
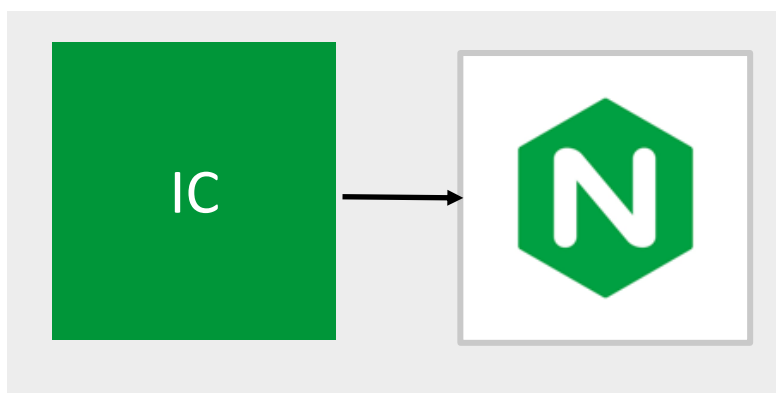
```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cafe-ingress
. . .
```

1. Get Ingress from Kubernetes API
2. Configure L7 Load Balancer
3. Repeat (go to Step 1)



NGINX Ingress Controller

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cafe-ingress
  . . .
```



Ingress Controller Purpose

Ingress Controller General Purpose

Provide external HTTP load balancing for applications running in a cluster

NGINX Ingress Controller Purpose

Provide external HTTP load balancing for applications running in a cluster...

...securely, reliably and with high performance through NGINX/NGINX Plus

To be 'k8s-native', integrating well with the ecosystem and using familiar configuration and processes

NGINX Ingress Controllers

 **nginxinc / kubernetes-ingress**

<https://github.com/nginxinc/kubernetes-ingress/>

 **kubernetes / ingress-nginx**

<https://github.com/kubernetes/ingress-nginx>

Read about the differences:

<https://github.com/nginxinc/kubernetes-ingress/blob/master/docs/nginx-ingress-controllers.md>

Demo 1

Ingress resource

Limitations of Ingress

- **Ingress features**
- Cross-namespace configuration

Limitations of Ingress – Features

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: webapp
spec:
  tls:
    - hosts:
      - webapp.example.com
      secretName: webapp-secret
  rules:
    - host: webapp.example.com
      http:
        paths:
          - path: /
            backend:
              serviceName: webapp
              servicePort: 80
```

Features

- TLS termination
- host-based routing
- path-based routing

Ingress Annotations

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: webapp
  annotations:
    nginx.org/lb-method: "ip_hash"
spec:
  rules:
  - host: webapp.example.com
    http:
      paths:
      - path: /
        backend:
          serviceName: webapp
          servicePort: 80
```

Good/Bad Things about Annotations

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: webapp
  annotations:
    nginx.org/lb-method: "ip_hash"
spec:
  rules:
  - host: webapp.example.com
    http:
      paths:
      - path: /
        backend:
          serviceName: webapp
          servicePort: 80
```

Good Things:

Implement features missing in the Ingress spec

Good/Bad Things about Annotations

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: webapp
  annotations:
    nginx.org/lb-method: "ip_hash"
    nginx.org/ssl-services: "webapp"
    nginx.org/proxy-connect-timeout: "10s"
    nginx.org/proxy-read-timeout: "10s"
    nginx.org/proxy-send-timeout: "10s"
    nginx.org/rewrites: "serviceName=webapp rewrite=/v1"
    nginx.com/jwt-key: "webapp-jwk"
    nginx.com/jwt-realm: "Webb App"
    nginx.com/jwt-token: "$cookie_auth_token"
    nginx.com/jwt-login-url: "https://login.example.com"
spec:
  rules:
  - host: webapp.example.com
    . . .
```

Good Things:

Implement features missing in Ingress spec

Bad Things:

Can quickly grow bigger than the spec

Lack validation

Not suited for granular configuration

Not portable

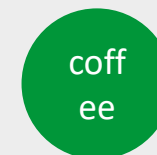
Limitations of Ingress

- Ingress features
- **Cross-namespace configuration**

Limitations of Ingress – Cross-namespace Configuration

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cafe-ingress
spec:
  rules:
  - host: cafe.example.com
    http:
      paths:
      - path: /coffee
        backend:
          serviceName: coffee
          servicePort: 80
      - path: /tea
        backend:
          serviceName: tea
          servicePort: 80
```

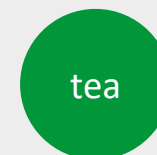
Namespace coffee



coffee

coffee

Namespace tea



tea

tea

tea

Mergeable Ingresses for Cross-namespace Configuration

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cafe-ingress
spec:
  rules:
    - host: cafe.example.com
      http:
        paths:
          - path: /coffee
            backend:
              serviceName: coffee
              servicePort: 80
          - path: /tea
            backend:
              serviceName: tea
              servicePort: 80
```

Namespace coffee



coffee

coffee

Namespace tea



tea

tea

tea

Mergeable Master

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cafe-ingress
spec:
  rules:
    - host: cafe.example.com
      http:
        paths:
          - path: /coffee
            backend:
              serviceName: coffee
              servicePort: 80
          - path: /tea
            backend:
              serviceName: tea
              servicePort: 80
```



```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cafe-ingress-master
  namespace: cafe
  annotations:
    nginx.org/mergeable-ingress-type: "master"
spec:
  rules:
    - host: cafe.example.com
```

Mergeable Minion

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cafe-ingress
spec:
  rules:
  - host: cafe.example.com
    http:
      paths:
        - path: /coffee
          backend:
            serviceName: coffee
            servicePort: 80
        - path: /tea
          backend:
            serviceName: tea
            servicePort: 80
```



```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cafe-ingress-coffee-minion
  namespace: coffee
  annotations:
    nginx.org/mergeable-ingress-type: "minion"
spec:
  rules:
  - host: cafe.example.com
    http:
      paths:
        - path: /coffee
          backend:
            serviceName: coffee-svc
            servicePort: 80
```

Mergeable Minion

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cafe-ingress
spec:
  rules:
  - host: cafe.example.com
    http:
      paths:
      - path: /coffee
        backend:
          serviceName: coffee
          servicePort: 80
      - path: /tea
        backend:
          serviceName: tea
          servicePort: 80
```



```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cafe-ingress-tea-minion
  namespace: tea
  annotations:
    nginx.org/mergeable-ingress-type: "minion"
spec:
  rules:
  - host: cafe.example.com
    http:
      paths:
      - path: /tea
        backend:
          serviceName: tea-svc
          servicePort: 80
```

Mergeable Ingresses for Cross-namespace Configuration

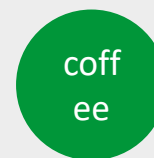
Master ingress for `cafe.example.com` in ns `cafe`

Minion ingress for `cafe.example.com/coffee`
in ns `coffee`

Minion ingress for `cafe.example.com/tea`
in ns `tea`

Namespace `cafe`

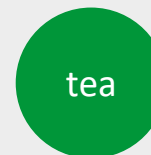
Namespace
`coffee`



coffee

coffee

Namespace
`tea`



tea

tea

tea

Limitations of Ingress

- Ingress features
- Cross-namespace configuration

Can We do Better?

Then Ingress resource with a lot of annotations

Provide external HTTP load balancing for applications running in a cluster...

...securely, reliably and with high performance through NGINX/NGINX Plus

To be 'k8s-native', integrating well with the ecosystem and using familiar configuration and processes

Goals for the NGINX Ingress Resources

Provide features without relying on annotations

Support cross-namespace configuration

VirtualServer Resource

```
apiVersion: k8s.nginx.org/v1
kind: VirtualServer
```

```
metadata:
```

```
  name: webapp
```

```
spec:
```

```
  host: webapp.example.com
```

```
  tls:
```

```
    secret: webapp-secret
```

```
  upstreams:
```

```
    - name: webapp
```

```
      service: webapp
```

```
      port: 80
```

```
  routes:
```

```
    - path: /
```

```
      action:
```

```
        pass: webapp
```

VirtualServer / Ingress Resources

```
apiVersion: k8s.nginx.org/v1
kind: VirtualServer
```

```
metadata:
  name: webapp
```

```
spec:
```

```
  host: webapp.example.com
```

```
  tls:
```

```
    secret: webapp-secret
```

```
  upstreams:
```

```
    - name: webapp
      service: webapp
      port: 80
```

```
  routes:
```

```
    - path: /
      action:
        path: webapp
```

```
apiVersion: extensions/v1beta1
kind: Ingress
```

```
metadata:
  name: webapp
```

```
spec:
```

```
  tls:
```

```
    - hosts:
      - webapp.example.com
      secretName: webapp-secret
```

```
  rules:
```

```
    - host: webapp.example.com
```

```
      http:
```

```
        paths:
```

```
          - path: /
            backend:
              serviceName: webapp
              servicePort: 80
```

Demo 3

VirtualServer Resource

Load Balancing

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: webapp
  annotations:
    nginx.org/lb-method: "ip_hash"
spec:
  rules:
  - host: webapp.example.com
    http:
      paths:
      - path: /
        backend:
          serviceName: webapp
          servicePort: 80
```

```
upstreams:
```

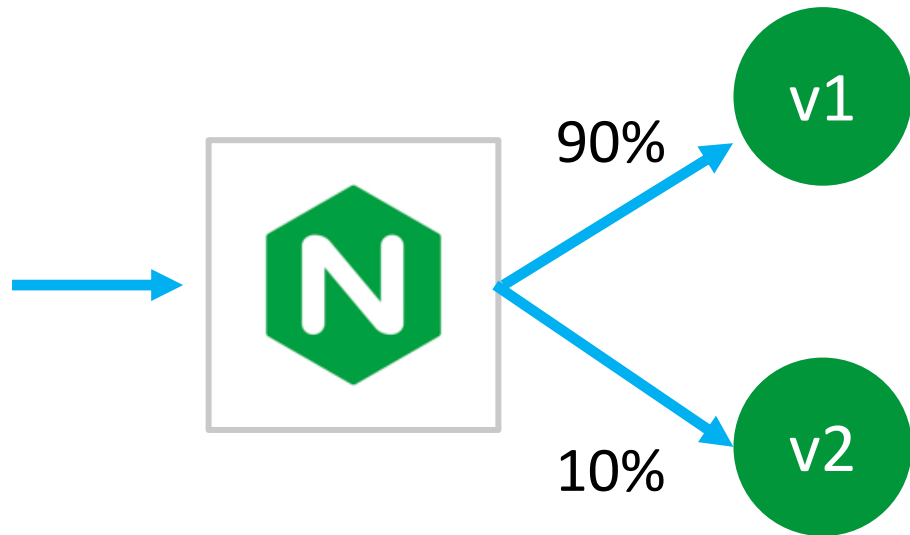
```
- name: webapp
  service: webapp
  port: 80
```

```
  lb-method: ip_hash
```

```
- name: coffee
  service: coffee-svc
  port: 80
```

```
  lb-method: least_conn
```

Traffic Splitting



```
upstreams:
```

- name: webapp-v1
service: webapp-v1
port: 80
- name: webapp-v2
service: webapp-v2
port: 80

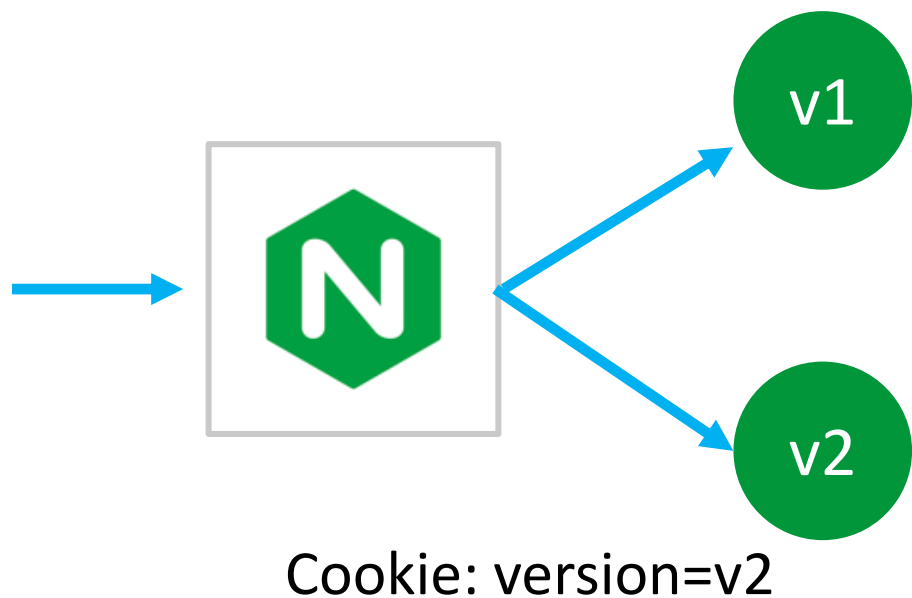
```
routes:
```

- path: /

```
splits:
```

- weight: 90
action:
pass: webapp-v1
- weight: 10
action:
pass: webapp-v2

Advanced Routing



```
upstreams:
```

- name: webapp-v1
service: webapp-v1
port: 80
- name: webapp-v2
service: webapp-v2
port: 80

```
routes:
```

- path: /

```
matches:
```

- conditions:
 - cookie: version
value: v2
- action:
 - pass: webapp-v2

```
action:
```

- pass: webapp-v1

Demo 4

VirtualServer with traffic shifting and advanced request routing

Cross-namespace Configuration

```
apiVersion: k8s.nginx.org/v1
kind: VirtualServer
metadata:
  name: cafe
  namespace: cafe
spec:
  host: cafe.example.com
  tls:
    secret: cafe-secret
  routes:
    - path: /coffee
      route: coffee/coffee
    - path: /tea
      route: tea/tea
```

Namespace cafe

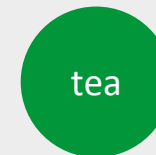
Namespace
coffee



coffee

coffee

Namespace
tea



tea

tea

tea

Cross-namespace Configuration - VirtualServerRoute

```
apiVersion: k8s.nginx.org/v1
```

```
kind: VirtualServerRoute
```

```
metadata:
```

```
  name: coffee
```

```
  namespace: coffee
```

```
spec:
```

```
  host: cafe.example.com
```

```
  upstreams:
```

```
  - name: coffee
```

```
    service: coffee-svc
```

```
    port: 80
```

```
  subroutes:
```

```
  - path: /coffee
```

```
    action:
```

```
      pass: coffee
```

Namespace
coffee



coffee

coffee

Cross-namespace Configuration - VirtualServerRoute

```
apiVersion: k8s.nginx.org/v1
```

```
kind: VirtualServerRoute
```

```
metadata:
```

```
  name: tea
```

```
  namespace: tea
```

```
spec:
```

```
  host: cafe.example.com
```

```
  upstreams:
```

```
  - name: tea
```

```
    service: tea-svc
```

```
    port: 80
```

```
  subroutes:
```

```
  - path: /tea
```

```
    action:
```

```
      pass: tea
```

Namespace

tea



tea

tea

tea

Cross-namespace Configuration

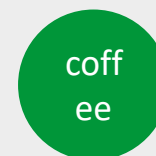
```
VirtualServer for cafe.example.com in ns café,  
/coffee -> VSR coffee in ns coffee  
/tea -> VSR tea in ns tea
```

```
VirtualServerRoute for cafe.example.com/coffee  
in ns tea
```

```
VirtualServerRoute for cafe.example.com/tea  
in ns tea
```

Namespace cafe

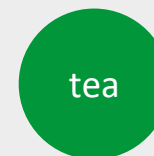
Namespace
coffee



coffee

coffee

Namespace
tea



tea

tea

tea

Summary of NGINX Ingress Resources

Based on Custom resources

Supports advanced features missing in the Kubernetes Ingress resource

Supports cross-namespace configuration missing in the Kubernetes Ingress resource

Multiple Protocols

```
apiVersion: k8s.nginx.org/v1alpha1
kind: TCPServer
metadata:
  name: simple-tcp
spec:
  port: 1234
  upstreams:
  - name: tcp-app
    service: tcp-app-svc
    port: 2001
  action:
    pass: tcp-app
```

TCP

```
apiVersion: k8s.nginx.org/v1alpha1
kind: TLSPassthroughServer
metadata:
  name: example
spec:
  host: myapp.example.com
  upstreams:
  - name: my-tls-app
    service: my-tls-app-svc
    port: 443
  action:
    pass: my-tls-app
```

TLS Passthrough

Next for NGINX Ingress Resources

- Support TCP/UDP and TLS Passthrough
- Support Polices – request/response manipulations, auth, rate-limiting, WAF

The End

Try out NGINX Ingress Controller

NGINX Ingress Controller:

<https://github.com/nginxinc/kubernetes-ingress>

Examples:

<https://github.com/nginxinc/kubernetes-ingress/tree/master/examples-of-custom-resources>