

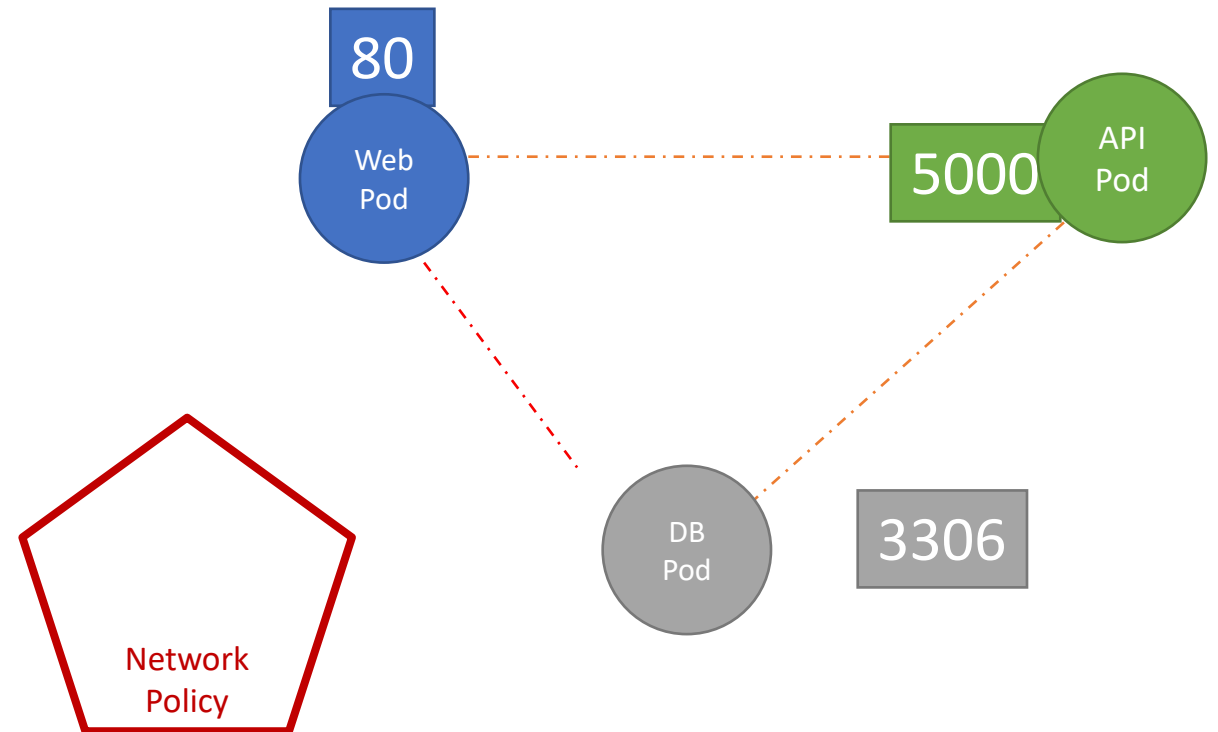


{KODE}{KLOUD

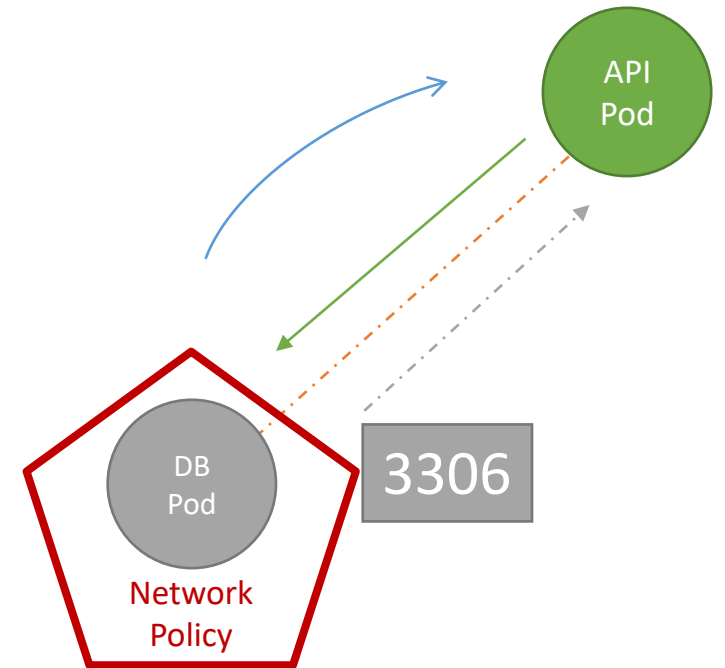
# Network Policies



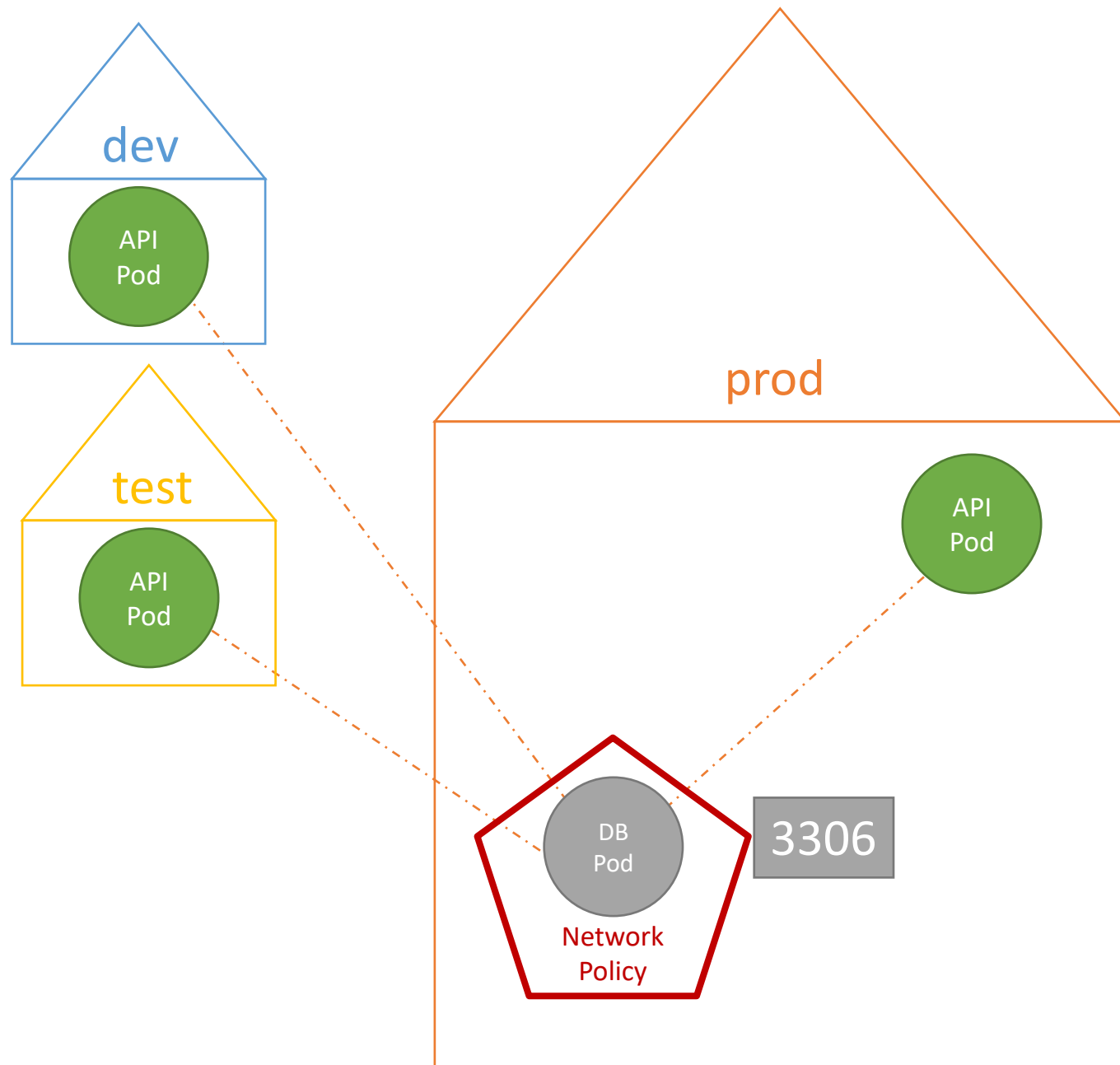
```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db
```



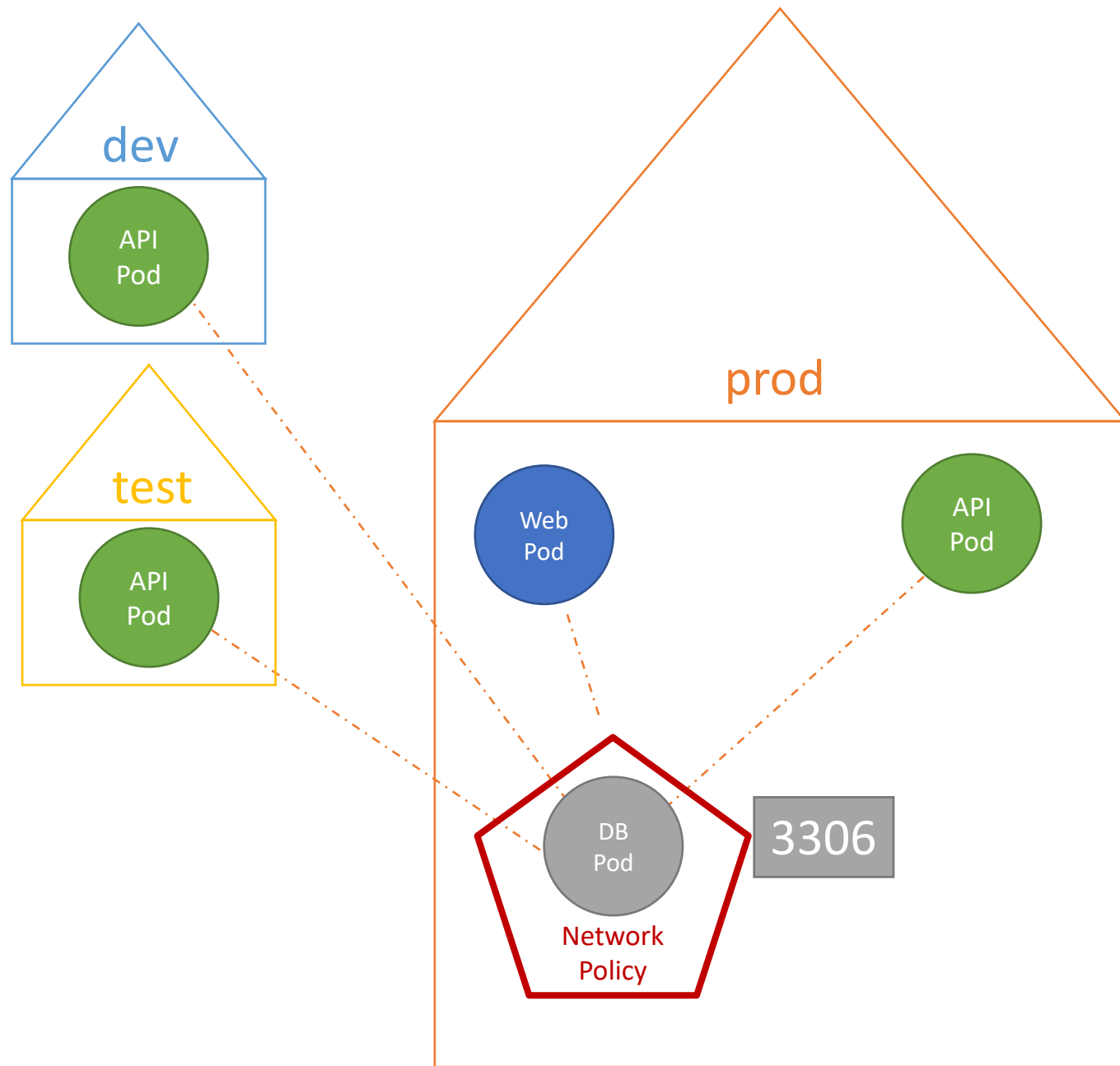
```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
    - Ingress
```



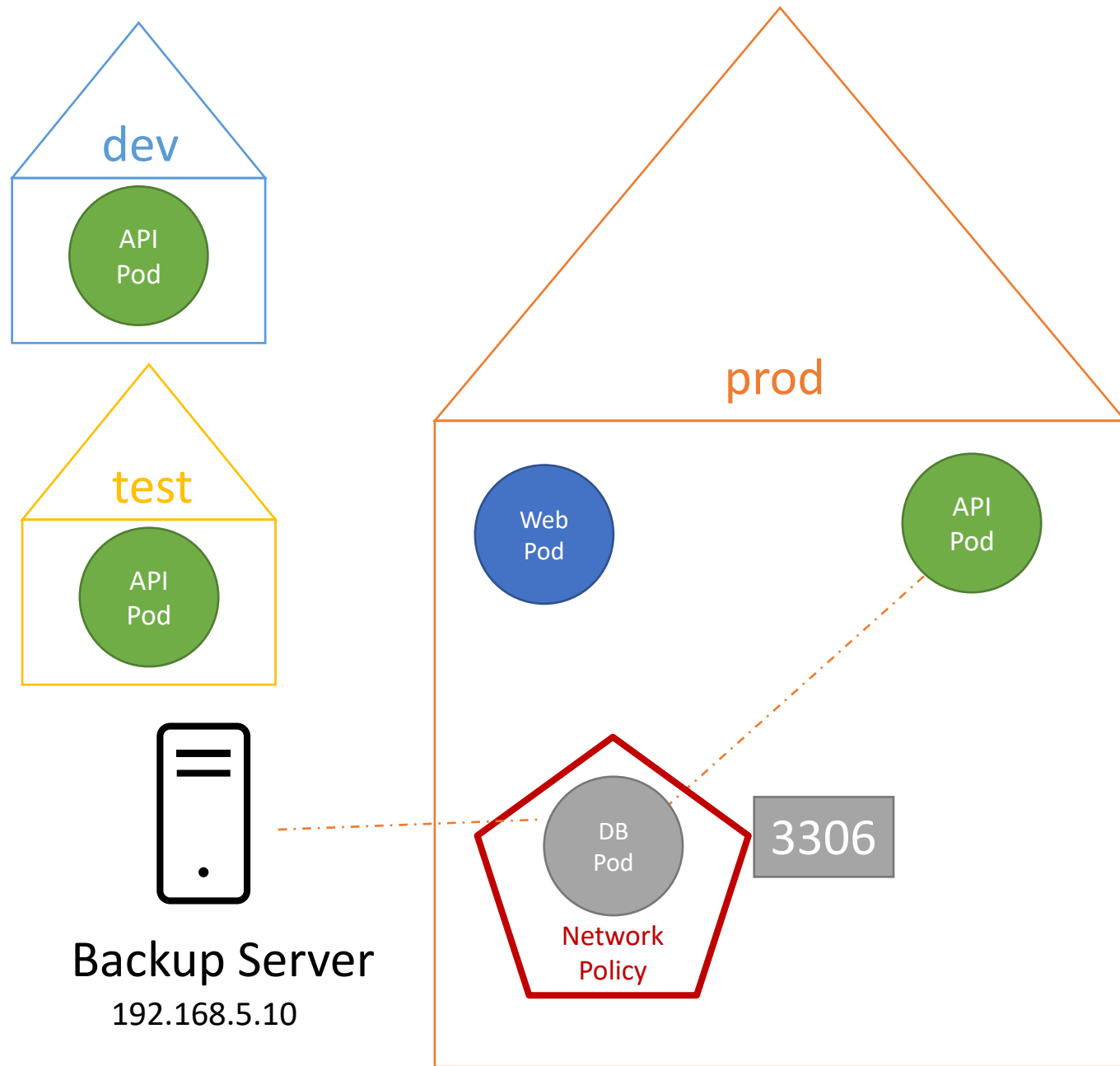
```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          name: api-pod
      ports:
      - protocol: TCP
        port: 3306
```



```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          name: api-pod
      namespaceSelector:
        matchLabels:
          name: prod
    ports:
    - protocol: TCP
      port: 3306
```



```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          name: api-pod
      namespaceSelector:
        matchLabels:
          name: prod
    ports:
    - protocol: TCP
      port: 3306
```



spec:

```
podSelector:  
  matchLabels:  
    role: db
```

```
policyTypes:  
- Ingress
```

```
ingress:
```

```
- from:
```

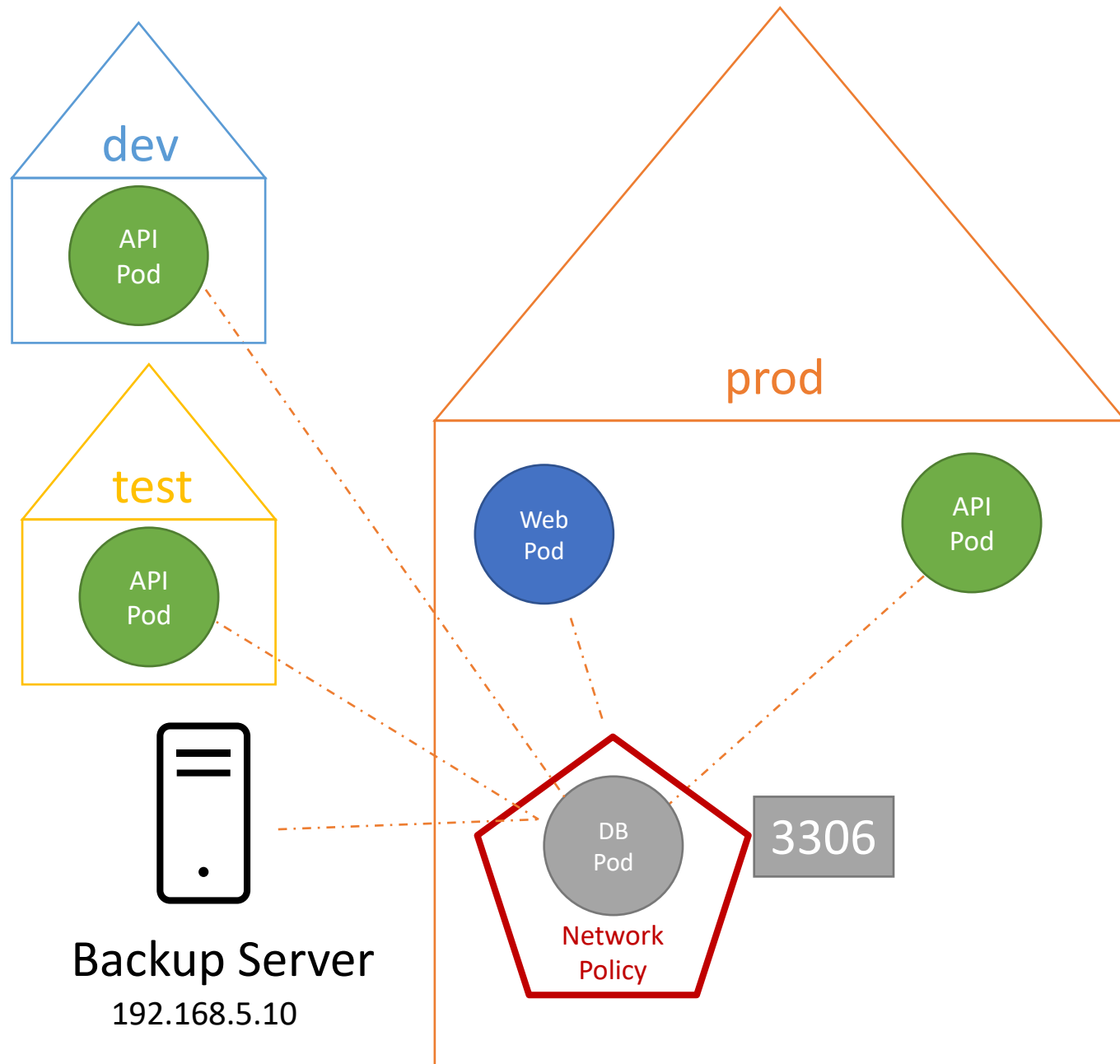
```
- podSelector:  
  matchLabels:  
    name: api-pod
```

```
- namespaceSelector:  
  matchLabels:  
    name: prod
```

```
- ipBlock:  
  cidr: 192.168.5.10/32
```

```
ports:
```

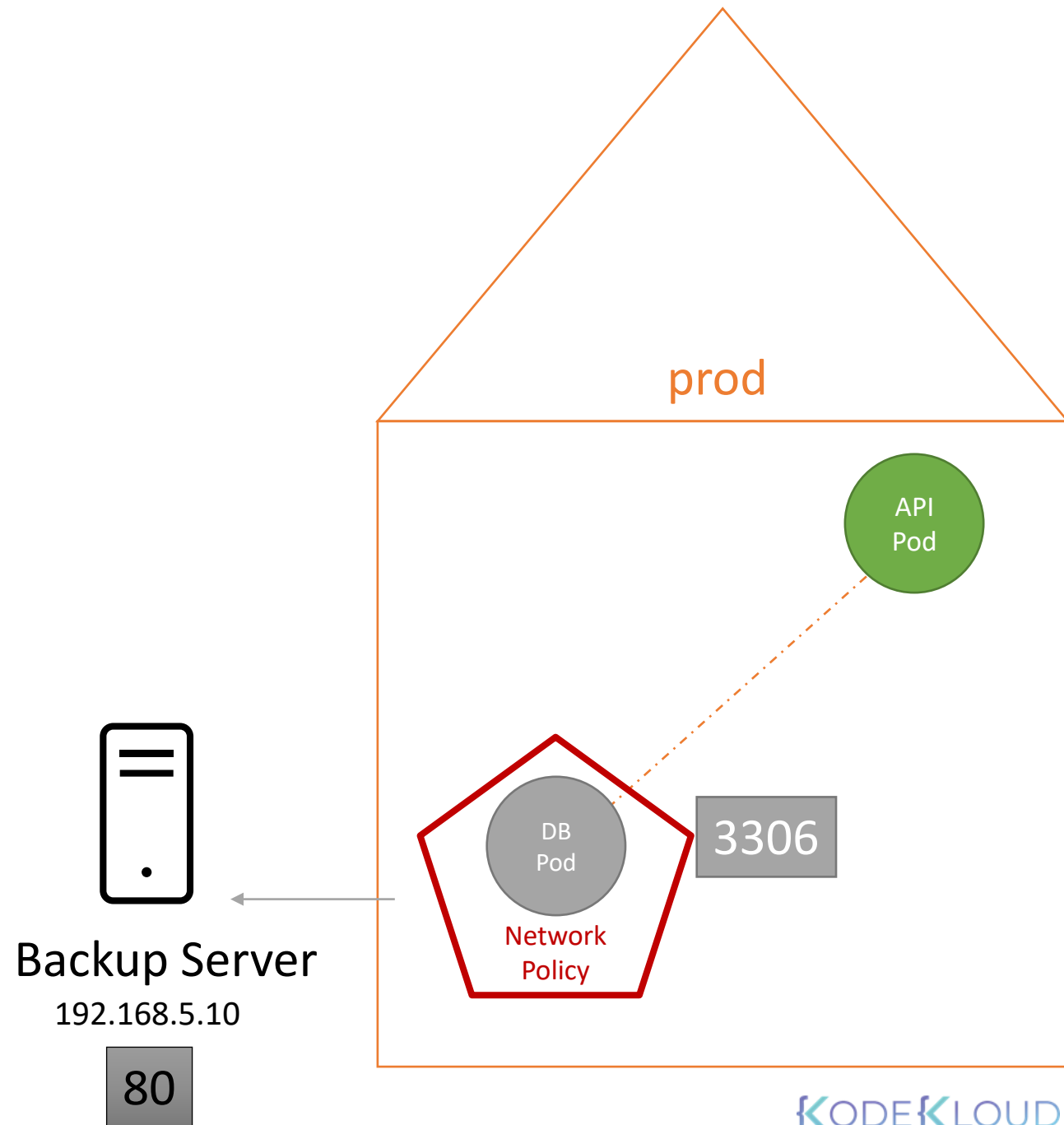
```
- protocol: TCP  
  port: 3306
```





spec:

```
podSelector:  
  matchLabels:  
    role: db  
  
policyTypes:  
- Ingress  
- Egress  
  
ingress:  
- from:  
  - podSelector:  
    matchLabels:  
      name: api-pod  
  
  ports:  
  - protocol: TCP  
    port: 3306  
  
egress:  
- to:  
  - ipBlock:  
    cidr: 192.168.5.10/32  
  
  ports:  
  - protocol: TCP  
    port: 80
```



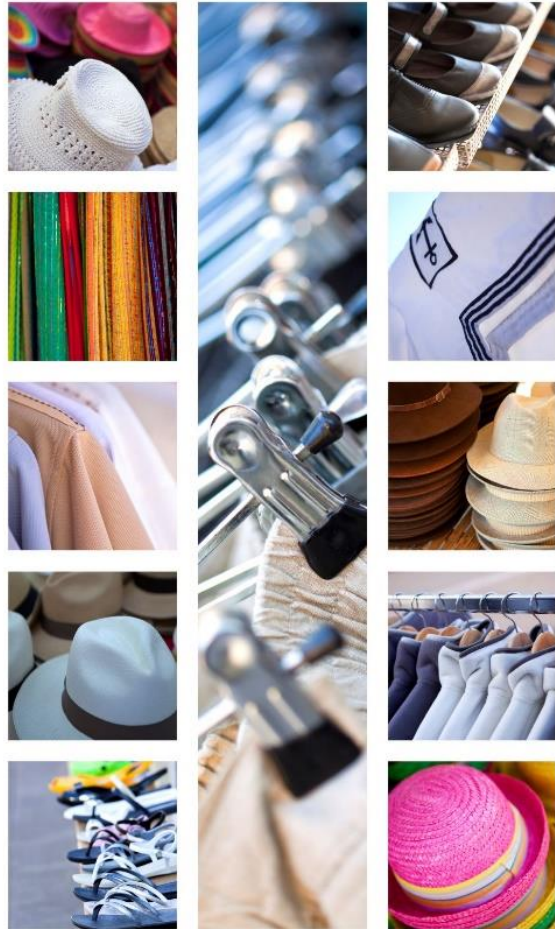


{KODE}{KLOUD

# INGRESS



[www.my-online-store.com](http://www.my-online-store.com)



www.my-online-store.com

http://<node-ip>:38080

38080

wear-service (NodePort)

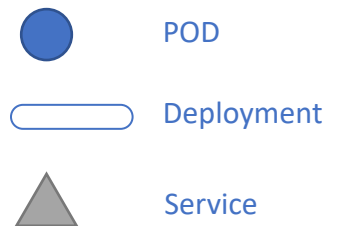
wear

wear

wear

mysql-service (ClusterIP)

MySQL



www.my-online-store.com



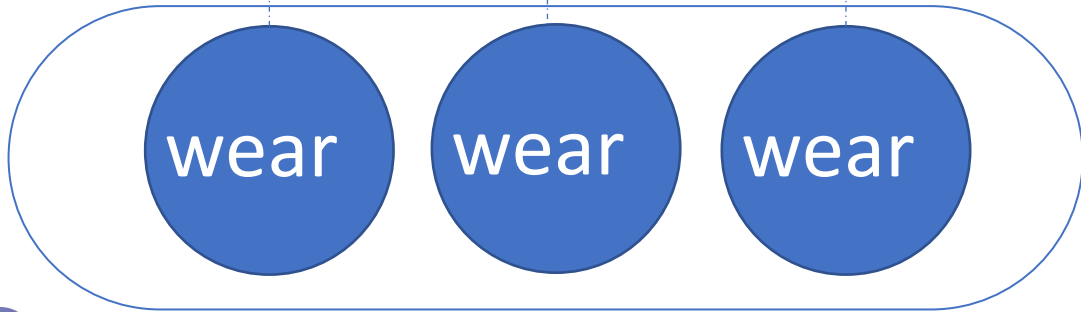
<node-ip>

http://my-online-store.com:38080

38080



wear-service (NodePort)



www.my-online-store.com



proxy-server

http://my-online-store.com

80

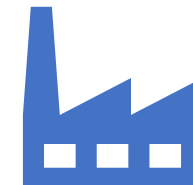


proxy-server

38080



wear-service (NodePort)



www.my-online-store.com



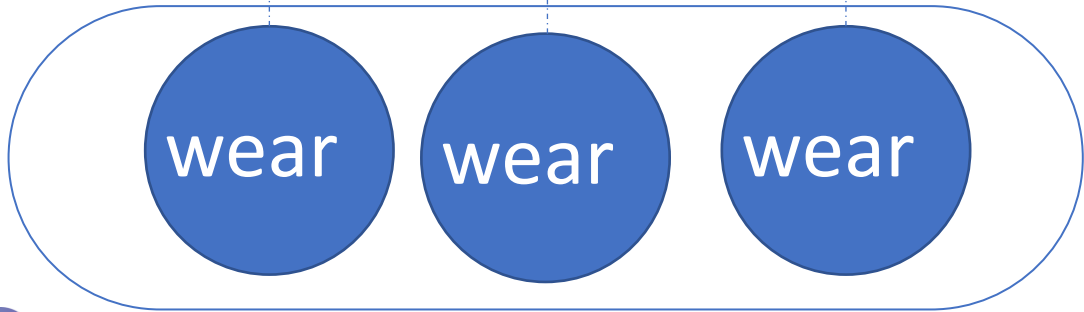
<node-ip>

http://my-online-store.com:38080

38080



wear-service (Node.js)



Google  
Cloud Platform





www.my-online-store.com → gcp load-balancer

http://my-online-store.com

80

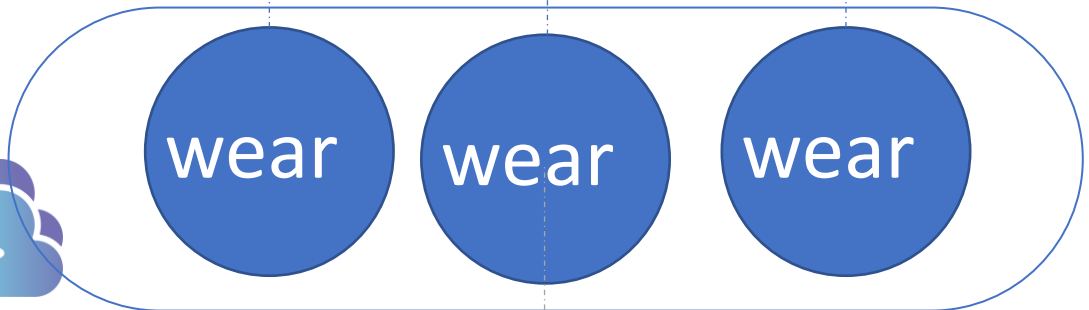
gcp load-balancer

38080

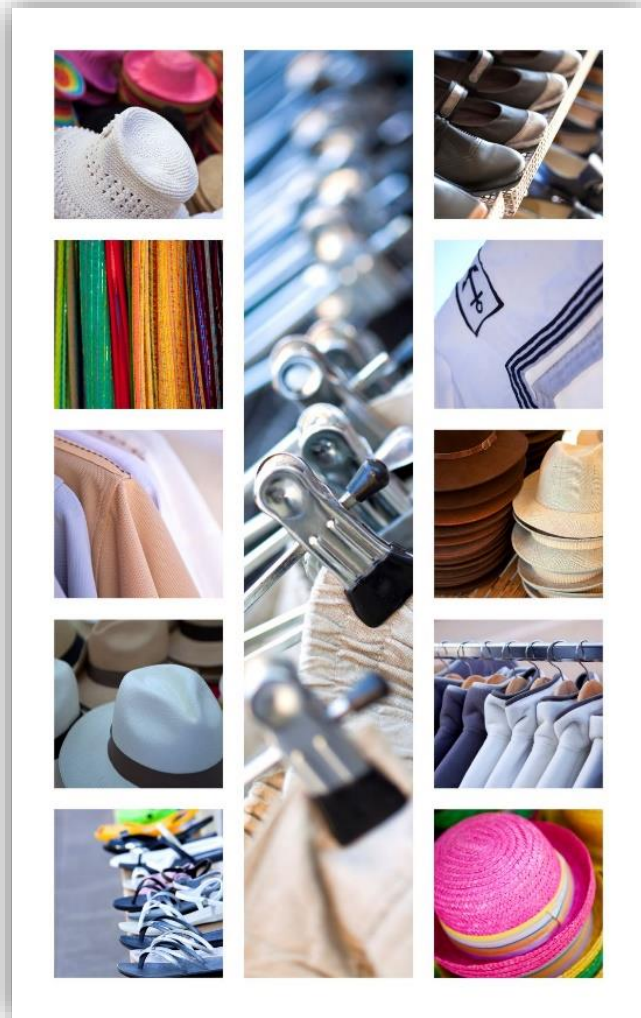


Google Cloud Platform

wear-service (LoadBalancer)

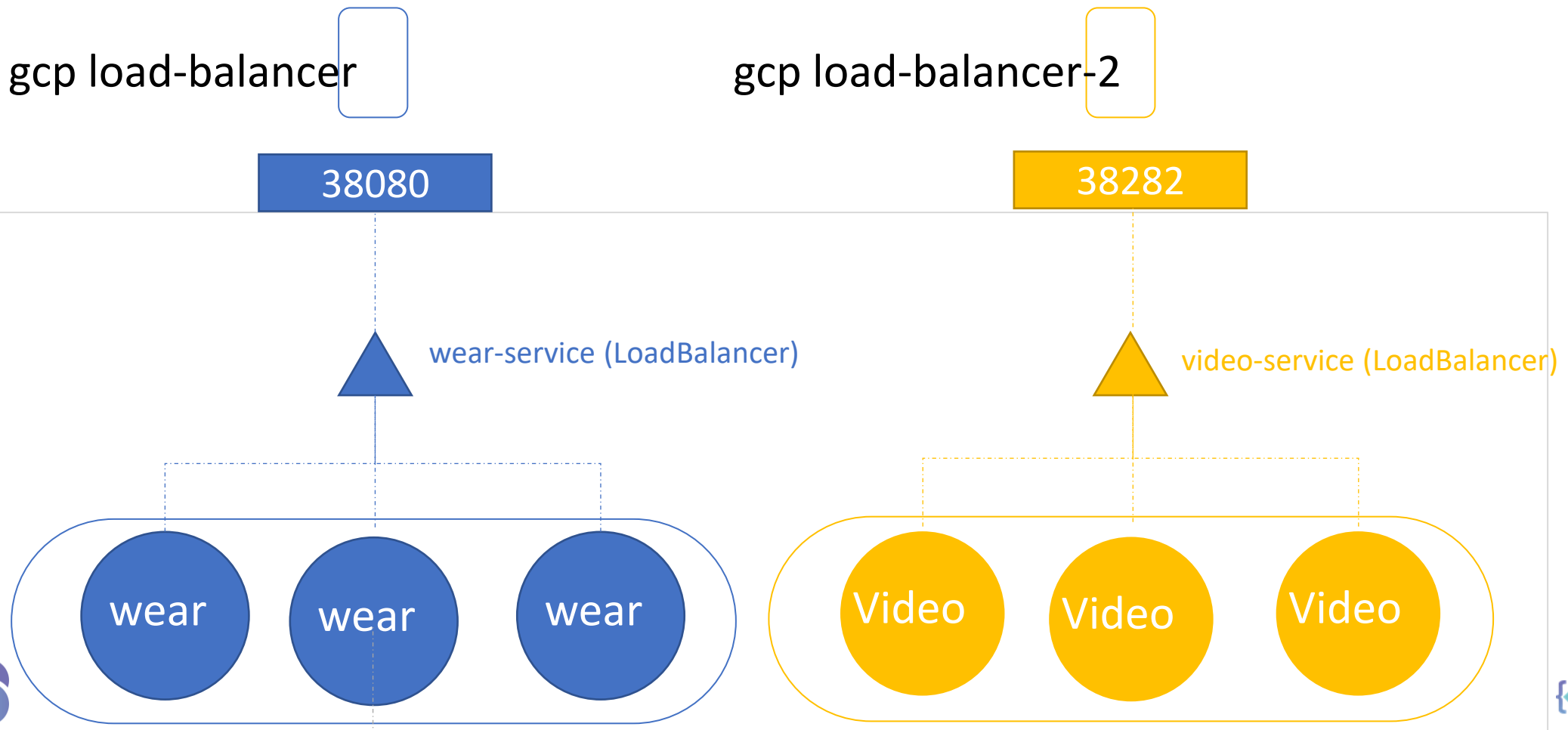


[www.my-online-store.com/wear](http://www.my-online-store.com/wear) / [www.my-online-store.com/watch](http://www.my-online-store.com/watch)



www.my-online-store.com → gcp load-balancer

http://my-online-store.com



www.my-online-store.com



yet another load-balancer

https://my-online-store.com



yet another load-balancer



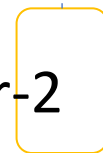
/apparel -> gcp load-balancer

/video -> gcp load-balancer-2

gcp load-balancer



gcp load-balancer-2



38080

38282



wear-service (LoadBalancer)



video-service (LoadBalancer)

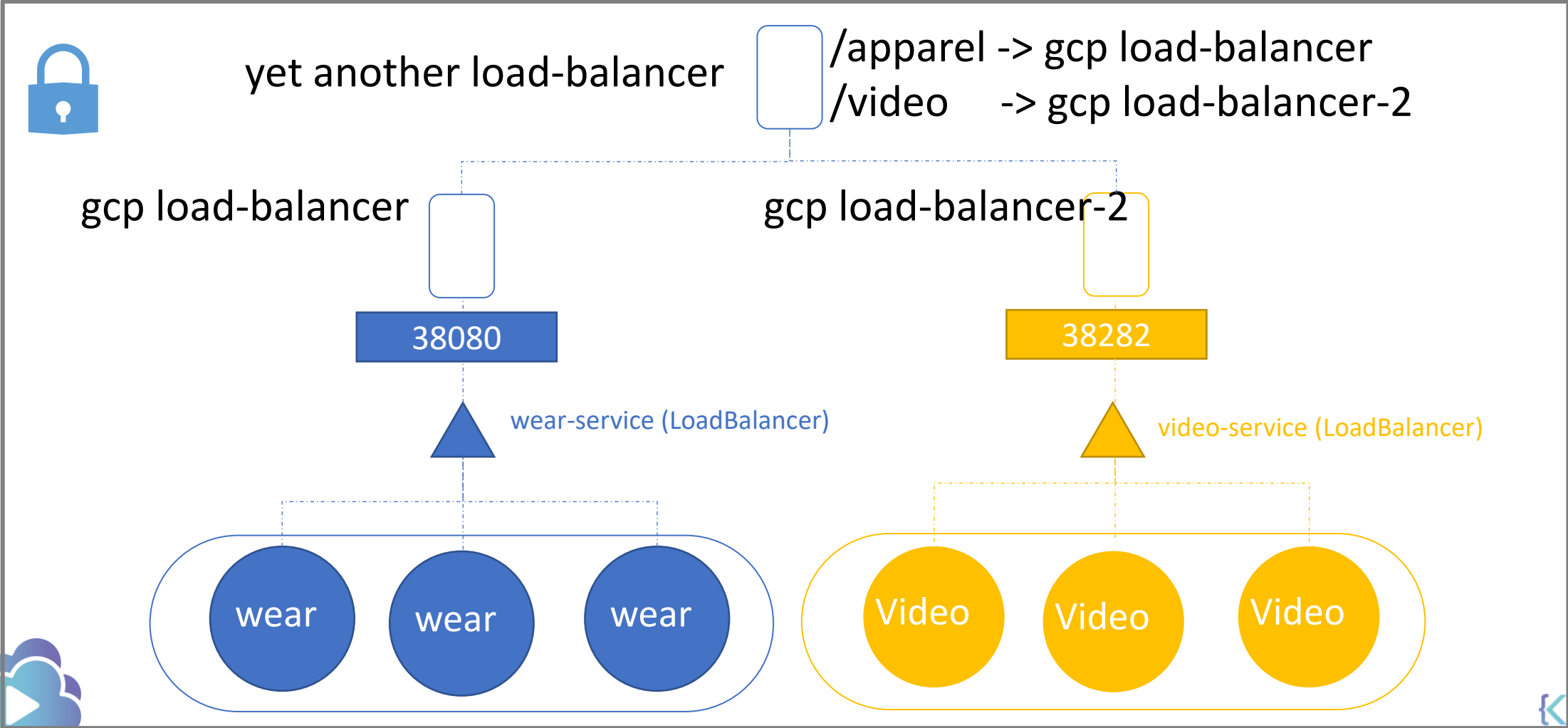


www.my-online-store.com

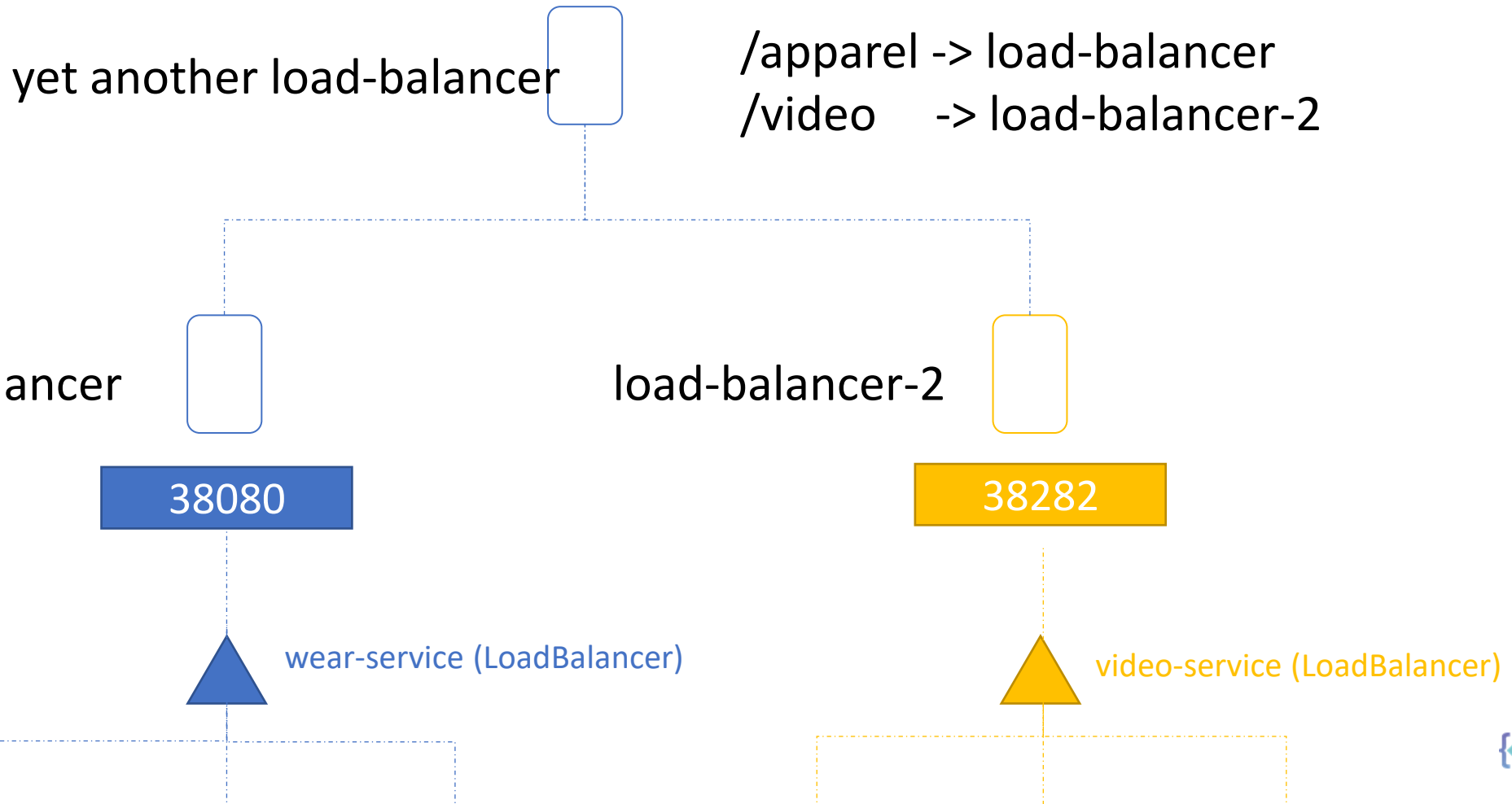


yet another load-balancer

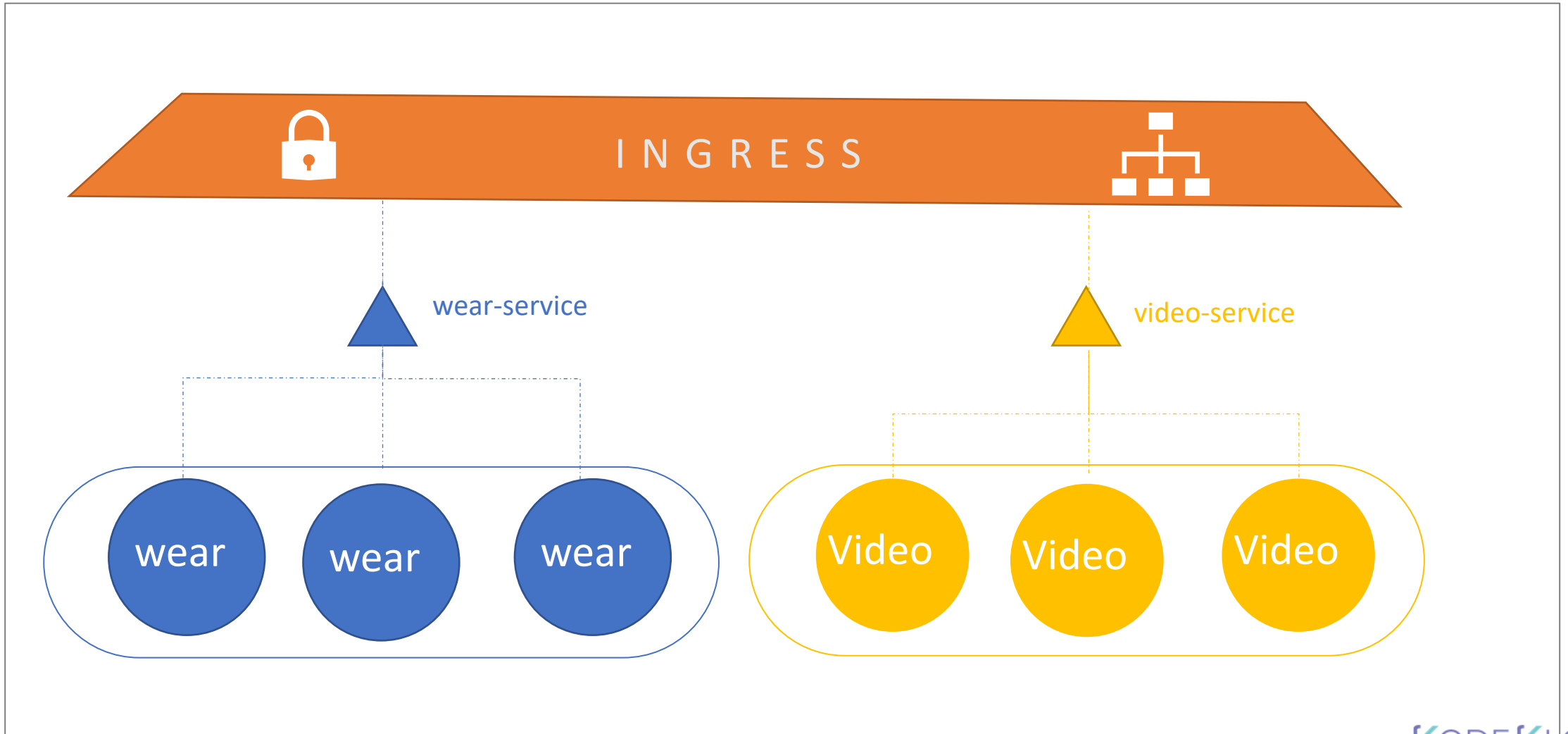
https://my-online-store.com



# Ingress



# Ingress



# Ingress



38080

ingress-service (NodePort)



wear-service

video-service





# Ingress

1. Deploy



HAPROXY



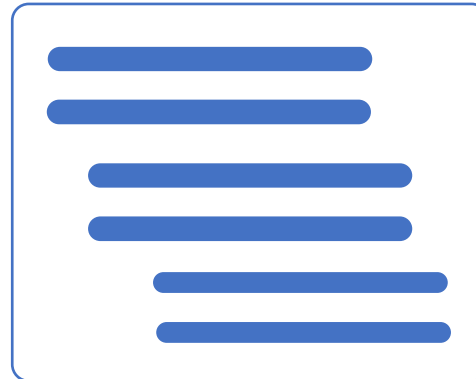
træfik

INGRESS CONTROLLER



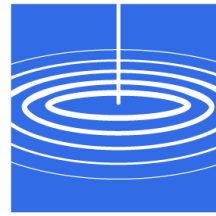
2. Configure

INGRESS RESOURCES

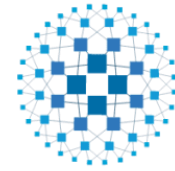


# INGRESS CONTROLLER

**GCP HTTP(S)  
Load Balancer (GCE)**



**Contour**



**HAPROXY**



**traefik**



**Istio**



# INGRESS CONTROLLER



ConfigMap

nginx-configuration

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
```

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-ingress-controller
spec:
  replicas: 1
  selector:
    matchLabels:
      name: nginx-ingress
  template:
    metadata:
      labels:
        name: nginx-ingress
    spec:
      containers:
        - name: nginx-ingress-controller
          image: quay.io/kubernetes-ingress-controller/nginx-ingress-controller:0.21.0
          args:
            - /nginx-ingress-controller
            - --configmap=$(POD_NAMESPACE)/nginx-configuration
```



# INGRESS CONTROL



**ConfigMap**

nginx-configuration

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
```

```
name: nginx-ingress-controller
spec:
  replicas: 1
  selector:
    matchLabels:
      name: nginx-ingress
  template:
    metadata:
      labels:
        name: nginx-ingress
    spec:
      containers:
        - name: nginx-ingress-controller
          image: quay.io/kubernetes-ingress-
            controller/nginx-ingress-controller:0.21.0
      args:
        - /nginx-ingress-controller
        - --configmap=$(POD_NAMESPACE)/nginx-configuration
      env:
        - name: POD_NAME
          valueFrom:
            fieldRef:
              fieldPath: metadata.name
        - name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
```

# INGRESS CONTROL



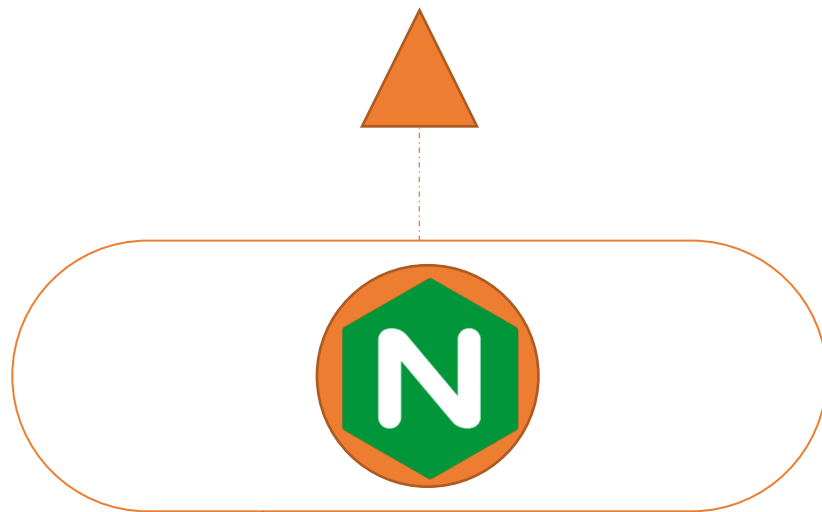
**ConfigMap**  
nginx-configuration

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
```

```
- name: nginx-ingress-controller
  image: quay.io/kubernetes-ingress-
    controller/nginx-ingress-controller:0.21.0
  args:
    - /nginx-ingress-controller
    - --configmap=$(POD_NAMESPACE)/nginx-configuration
  env:
    - name: POD_NAME
      valueFrom:
        fieldRef:
          fieldPath: metadata.name
    - name: POD_NAMESPACE
      valueFrom:
        fieldRef:
          fieldPath: metadata.namespace
  ports:
    - name: http
      containerPort: 80
    - name: https
      containerPort: 443
```



# INGRESS CONTROL



ConfigMap

nginx-configuration

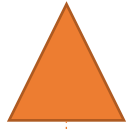
```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
```

```
fieldPath: metadata.namespace

ports:
  - name: http
    containerPort: 80
  - name: https
    containerPort: 443
```

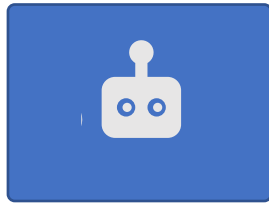
```
apiVersion: v1
kind: Service
metadata:
  name: nginx-ingress
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: 80
      protocol: TCP
      name: http
    - port: 443
      targetPort: 443
      protocol: TCP
      name: https
  selector:
    name: nginx-ingress
```

# INGRESS CONTROL



ConfigMap

nginx-configuration



ServiceAccount

nginx-ingress-serviceaccount

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
```

```
protocol: TCP
- port: 443
  targetPort: 443
  protocol: TCP
  name: https
  selector:
    name: nginx-ingress
```

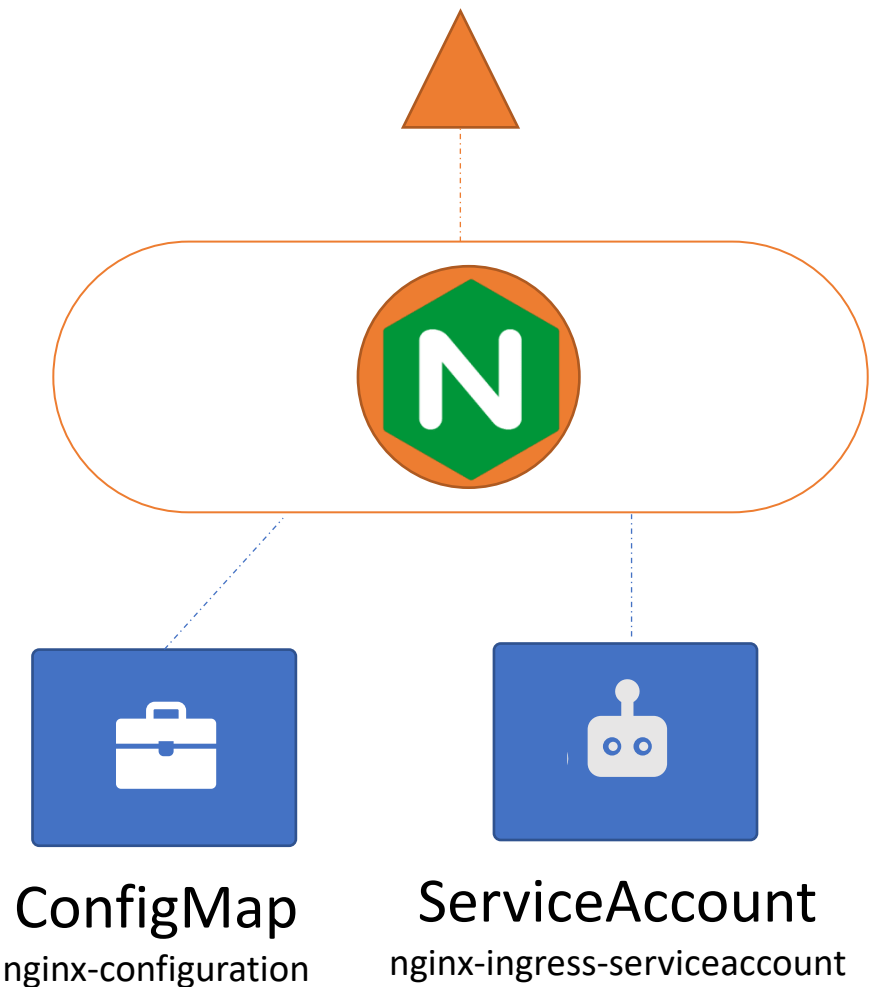
```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: nginx-ingress-serviceaccount
```

Roles

ClusterRole

RoleBinding

# INGRESS CONTROLLER



## Deployment

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-ingress-controller
spec:
  replicas: 1
  selector:
    matchLabels:
      name: nginx-ingress
  template:
    metadata:
      labels:
        name: nginx-ingress
    spec:
      containers:
        - name: nginx-ingress-controller
          image: quay.io/kubernetes-ingress-controller/nginx-ingress-controller:0.21.0
      args:
        - /nginx-ingress-controller
        - --configmap=$(POD_NAMESPACE)/nginx-configuration
      env:
        - name: POD_NAME
          valueFrom:
            fieldRef:
              fieldPath: metadata.name
        - name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
      ports:
        - name: http
          containerPort: 80
        - name: https
          containerPort: 443
```

## Service

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-ingress
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: 80
      protocol: TCP
      name: http
    - port: 443
      targetPort: 443
      protocol: TCP
      name: https
  selector:
    name: nginx-ingress
```

## ConfigMap

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
```

## Auth

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: nginx-ingress-serviceaccount
```

Roles

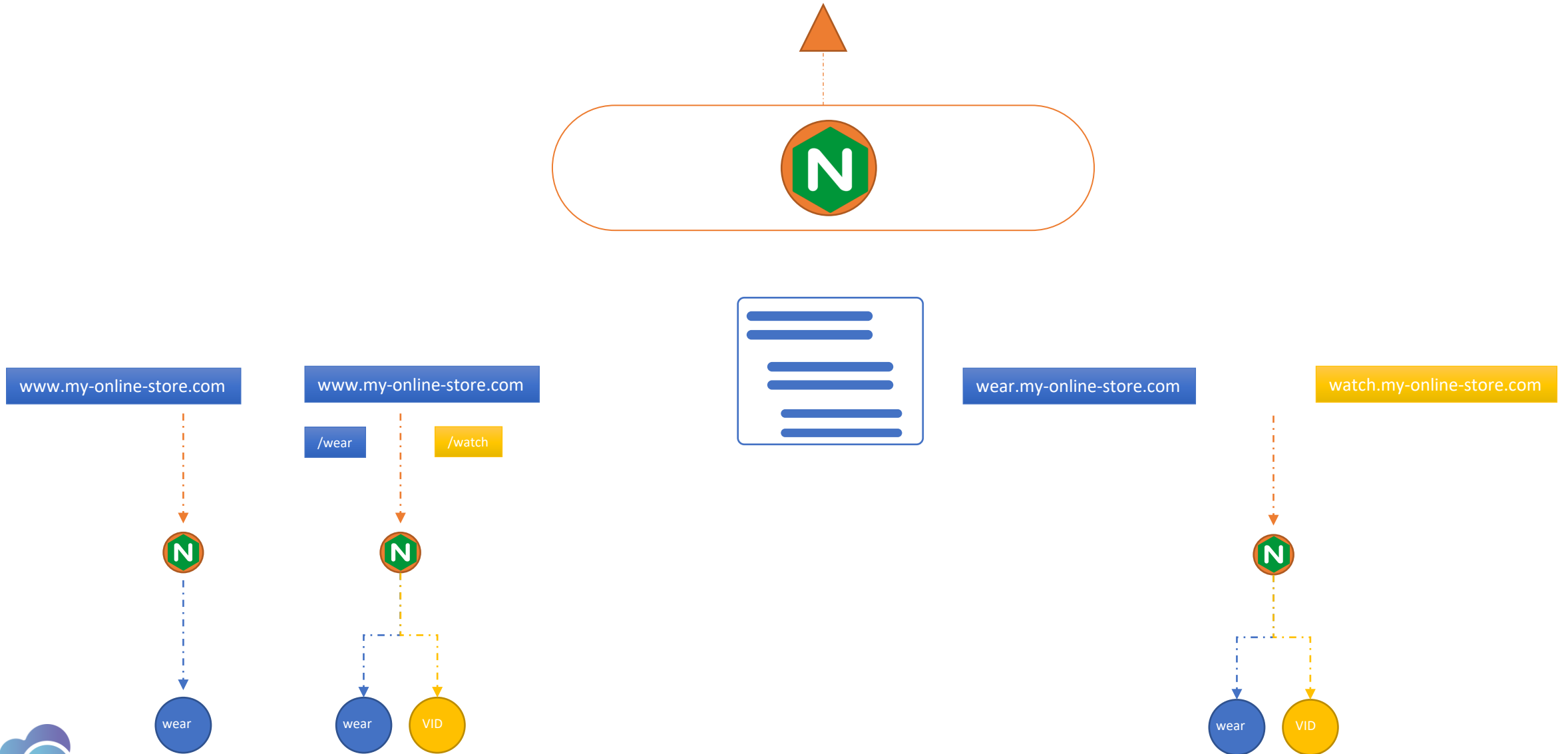
ClusterRoles

RoleBindings

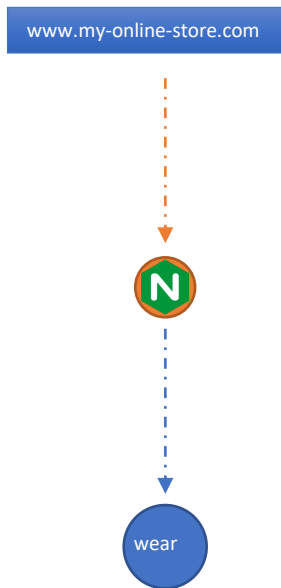




# INGRESS RESOURCE



# INGRESS RESOURCE

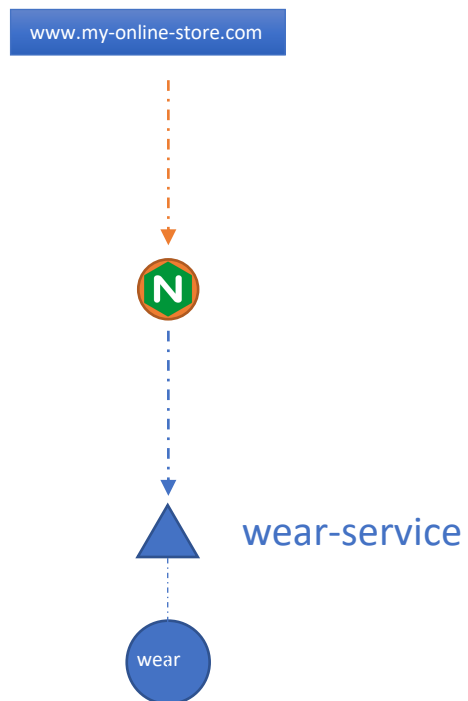


Ingress-wear.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear
spec:
```



# INGRESS RESOURCE



Ingress-wear.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear
spec:
  backend:
    serviceName: wear-service
    servicePort: 80
```

```
▶ kubectl create -f Ingress-wear.yaml
```

```
ingress.extensions/ingress-wear created
```

```
▶ kubectl get ingress
```

NAME	HOSTS	ADDRESS	PORTS
ingress-wear	*	80	2s



# INGRESS RESOURCE - RULES

www.my-online-store.com

Rule 1

www.wear.my-online-store.com

Rule 2

www.watch.my-online-store.com

Rule 3

Everything Else

Rule 4



# INGRESS RESOURCE - RULES

DNS Name	Forward IP
<a href="http://www.my-online-store.com">www.my-online-store.com</a>	10.123.23.12 (INGRESS SERVICE)
<a href="http://www.wear.my-online-store.com">www.wear.my-online-store.com</a>	10.123.23.12
<a href="http://www.watch.my-online.store.com">www.watch.my-online.store.com</a>	10.123.23.12
<a href="http://www.my-wear-store.com">www.my-wear-store.com</a>	10.123.23.12
<a href="http://www.my-online-store.com">www.my-online-store.com</a>	10.123.23.12
<a href="http://www.my-watch-store.com">www.my-watch-store.com</a>	10.123.23.12

Rule 1

Rule 4



# INGRESS RESOURCE - RULES

www.my-online-store.com

www.wear.my-online-store.com

www.watch.my-online-store.com

Everything Else

http://www.my-online-store.com/wear

http://www.my-online-store.com/watch

http://www.my-online-store.com/listen

Rule 1

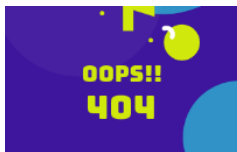
Path /wear



Path /watch



Path /



Rule 2

Rule 3

Rule 4



# INGRESS RESOURCE - RULES

www.my-online-store.com

www.wear.my-online-store.com

www.watch.my-online-store.com

Everything Else

http://www.my-online-store.com/wear

http://www.wear.my-online-store.com/

http://www.my-online-store.com/watch

http://www.wear.my-online-store.com/returns

http://www.my-online-store.com/listen

http://www.wear.my-online-store.com/support

Rule 1

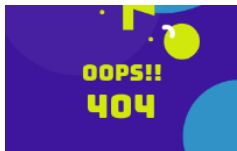
Path /wear



Path /watch



Path /



Rule 2

Path /



Path /returns



Path /support



Rule 3

Rule 4



# INGRESS RESOURCE - RULES

www.my-online-store.com

www.wear.my-online-store.com

www.watch.my-online-store.com

Everything Else

http://www.my-online-store.com/wear

http://www.wear.my-online-store.com/

http://www.watch.my-online-store.com/

http://www.my-online-store.com/watch

http://www.wear.my-online-store.com/returns

http://www.watch.my-online-store.com/movies

http://www.my-online-store.com/listen

http://www.wear.my-online-store.com/support

http://www.watch.my-online-store.com/tv

Rule 1

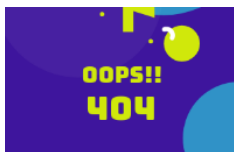
Path /wear



Path /watch



Path /



Rule 2

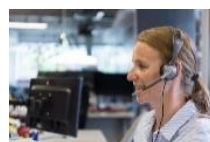
Path /



Path /returns



Path /support



Rule 3

Path /



Path /movies



Path /tv



Rule 4





# INGRESS RESOURCE - RULES

www.my-online-store.com

www.wear.my-online-store.com

www.watch.my-online-store.com

Everything Else

http://www.my-online-store.com/wear

http://www.wear.my-online-store.com/

http://www.watch.my-online-store.com/

http://www.listen.my-online-store.com/

http://www.my-online-store.com/watch

http://www.wear.my-online-store.com/returns

http://www.watch.my-online-store.com/movies

http://www.eat.my-online-store.com/

http://www.my-online-store.com/listen

http://www.wear.my-online-store.com/support

http://www.watch.my-online-store.com/tv

http://www.drink.my-online-store.com/tv

Rule 1

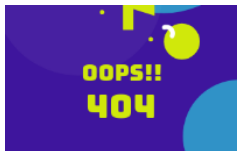
Path /wear



Path /watch



Path /



Rule 2

Path /



Path /returns



Path /support



Rule 3

Path /



Path /movies



Path /tv

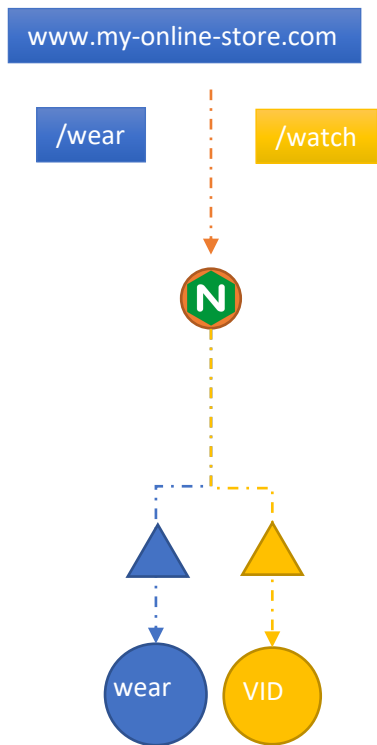


Rule 4

Path /



# INGRESS RESOURCE



Ingress-wear.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear
spec:
  backend:
    serviceName: wear-service
    servicePort: 80
```

Ingress-wear-watch.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear-watch
spec:
  rules:
    - http:
        paths:
          - path: /wear

          - path: /watch
        backend:
          serviceName: watch-service
          servicePort: 80
```



# INGRESS RESOURCE

```
▶ kubectl describe ingress ingress-wear-watch
```

```
Name:          ingress-wear-watch
```

```
Namespace:     default
```

```
Address:
```

```
Default backend: default-http-backend:80 (<none>)
```

```
Rules:
```

```
Host Path Backends
```

```
----
```

```
*
```

```
    /wear wear-service:80 (<none>)
```

```
    /watch watch-service:80 (<none>)
```

```
Annotations:
```

```
Events:
```

```
Type Reason Age From Message
```

```
----
```

```
Normal CREATE 14s nginx-ingress-controller Ingress default/ingress-wear-watch
```

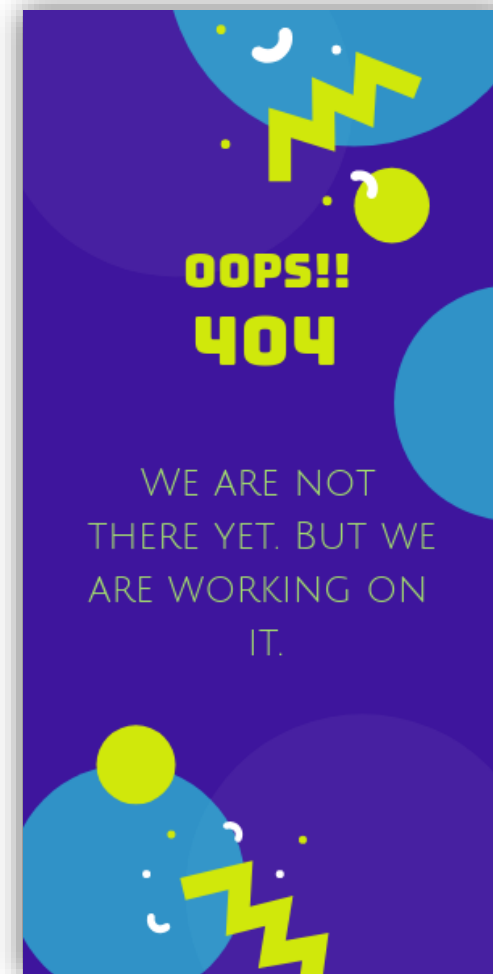
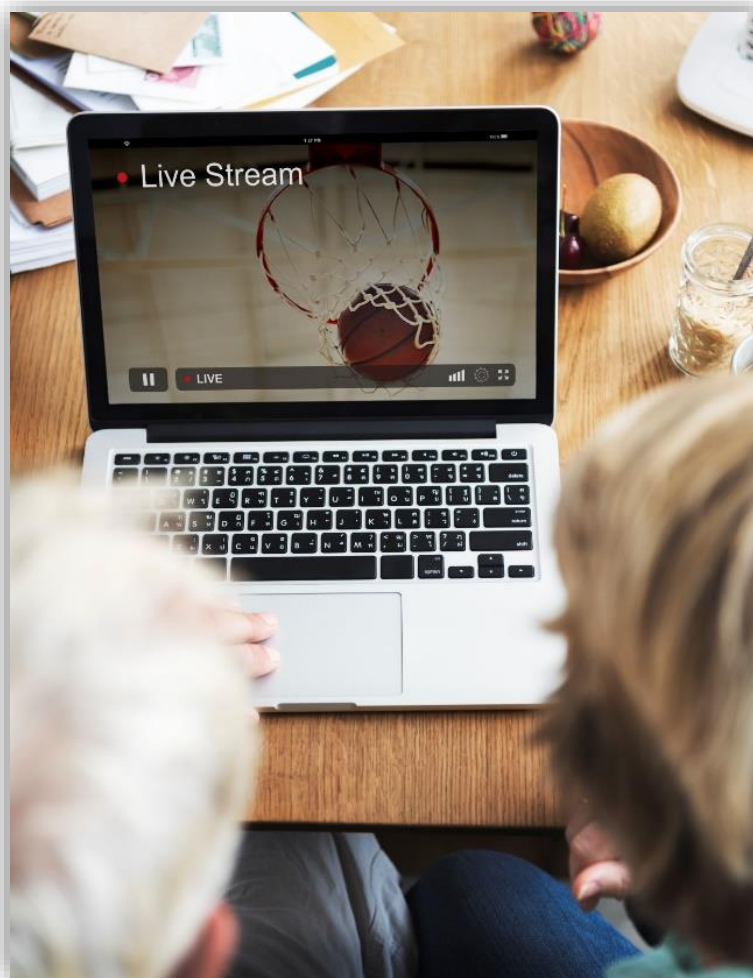
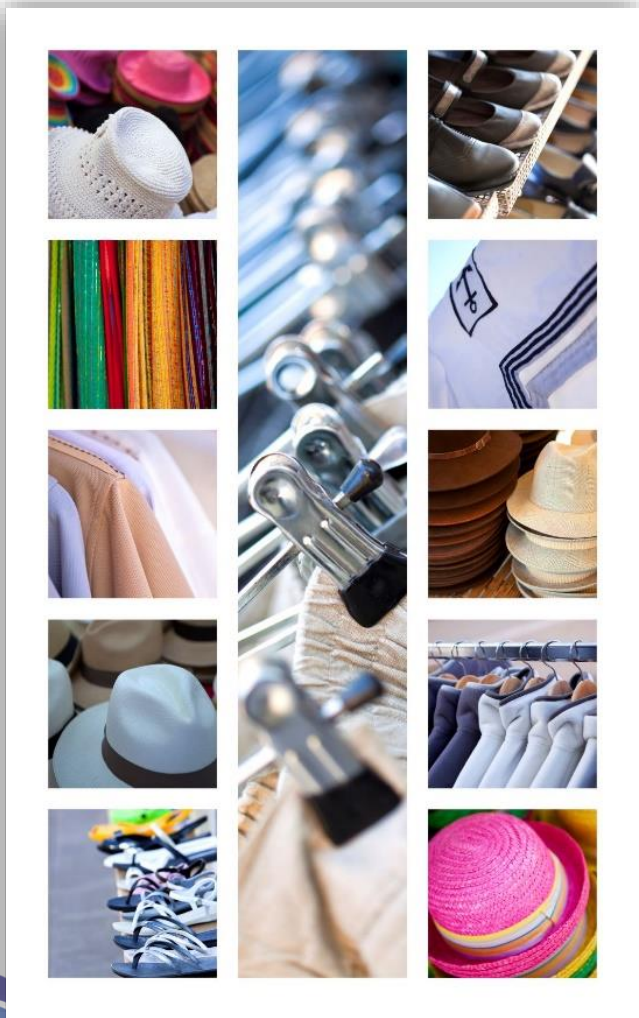


# INGRESS RESOURCE

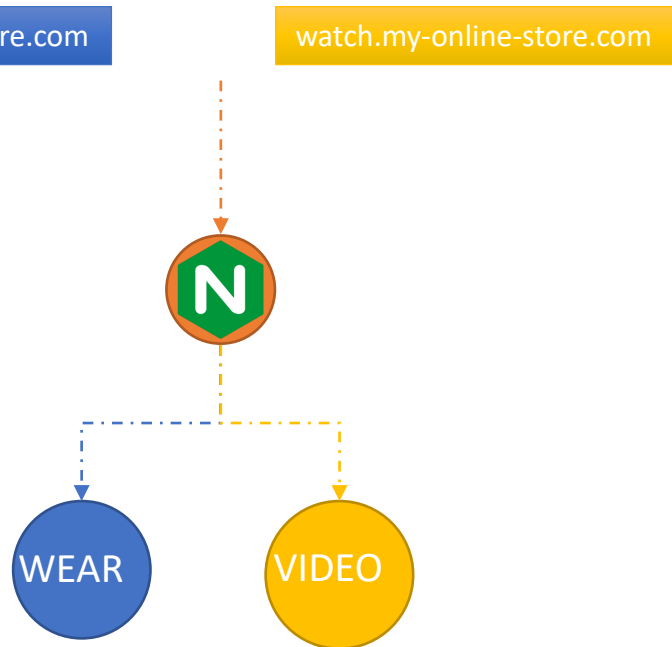
[www.my-online-store.com/eat](http://www.my-online-store.com/eat)

[www.my-online-store.com/wear](http://www.my-online-store.com/wear) [www.my-online-store.com/watch](http://www.my-online-store.com/watch)

[www.my-online-store.com/liste](http://www.my-online-store.com/liste)



# INGRESS RESOURCE



Ingress-wear-watch.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear-watch
spec:
  rules:
  - host: wear.my-online-store.com
    http:
      paths:
      - backend:
          serviceName: wear-service
          servicePort: 80
  - host: watch.my-online-store.com
    http:
      paths:
      - backend:
          serviceName: watch-service
          servicePort: 80
```



# INGRESS RESOURCE

Ingress-wear-watch.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear-watch
spec:
  rules:
  - http:
      paths:
      - path: /wear
        backend:
          serviceName: wear-service
          servicePort: 80
      - path: /watch
        backend:
          serviceName: watch-service
          servicePort: 80
```

Ingress-wear-watch.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-wear-watch
spec:
  rules:
  - host: wear.my-online-store.com
    http:
      paths:
      - backend:
          serviceName: wear-service
          servicePort: 80
  - host: watch.my-online-store.com
    http:
      paths:
      - backend:
          serviceName: watch-service
          servicePort: 80
```





