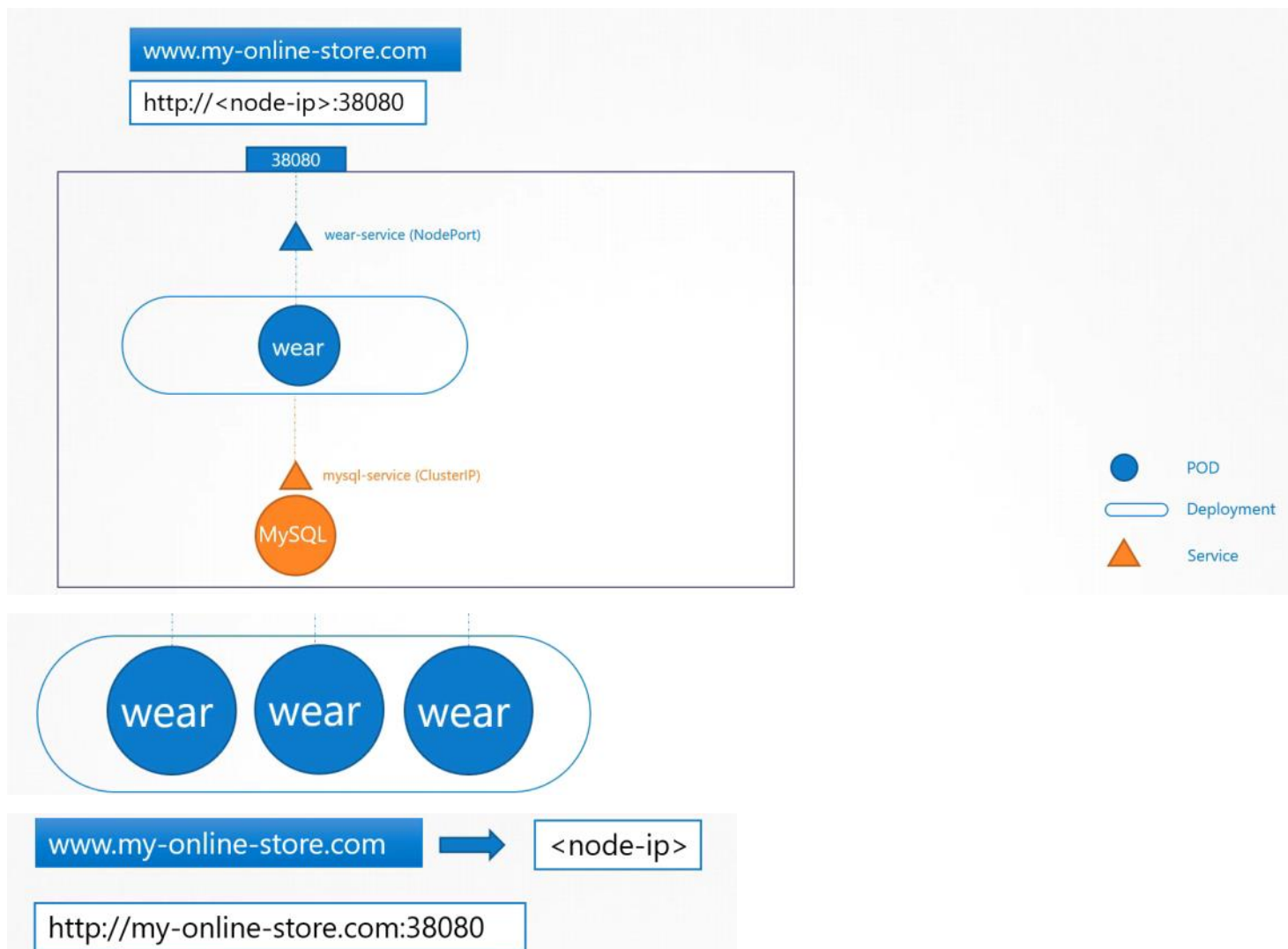
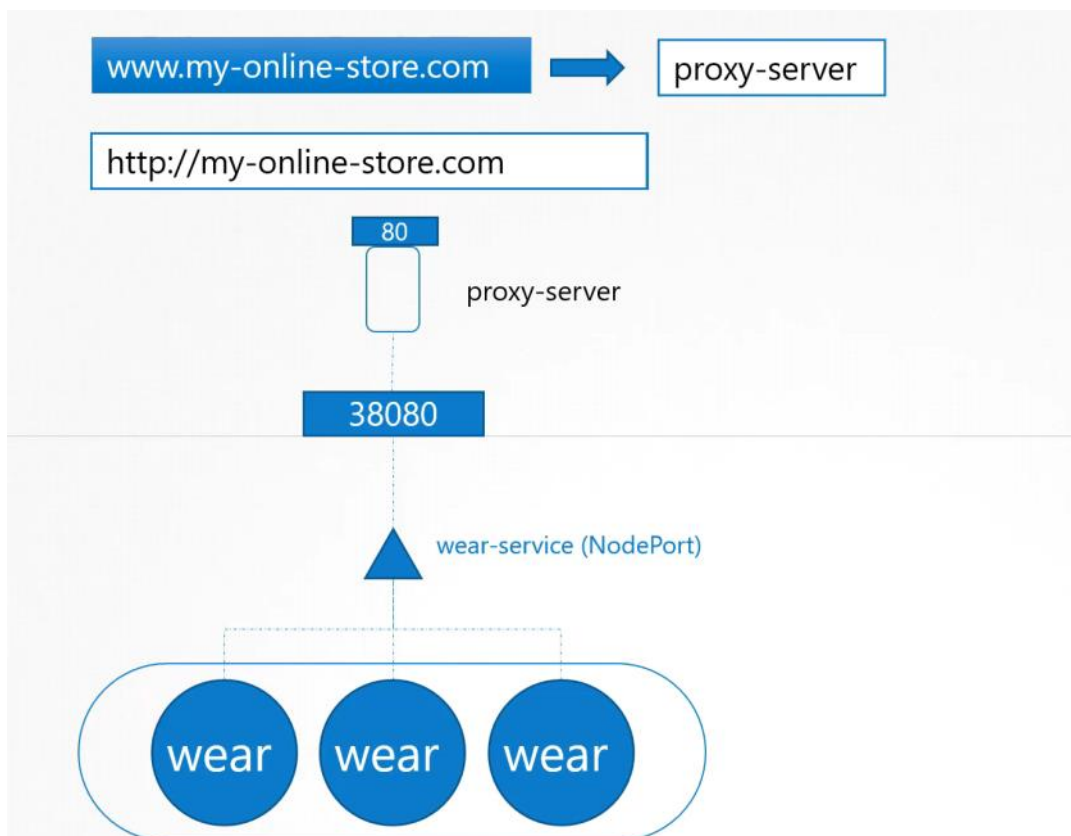
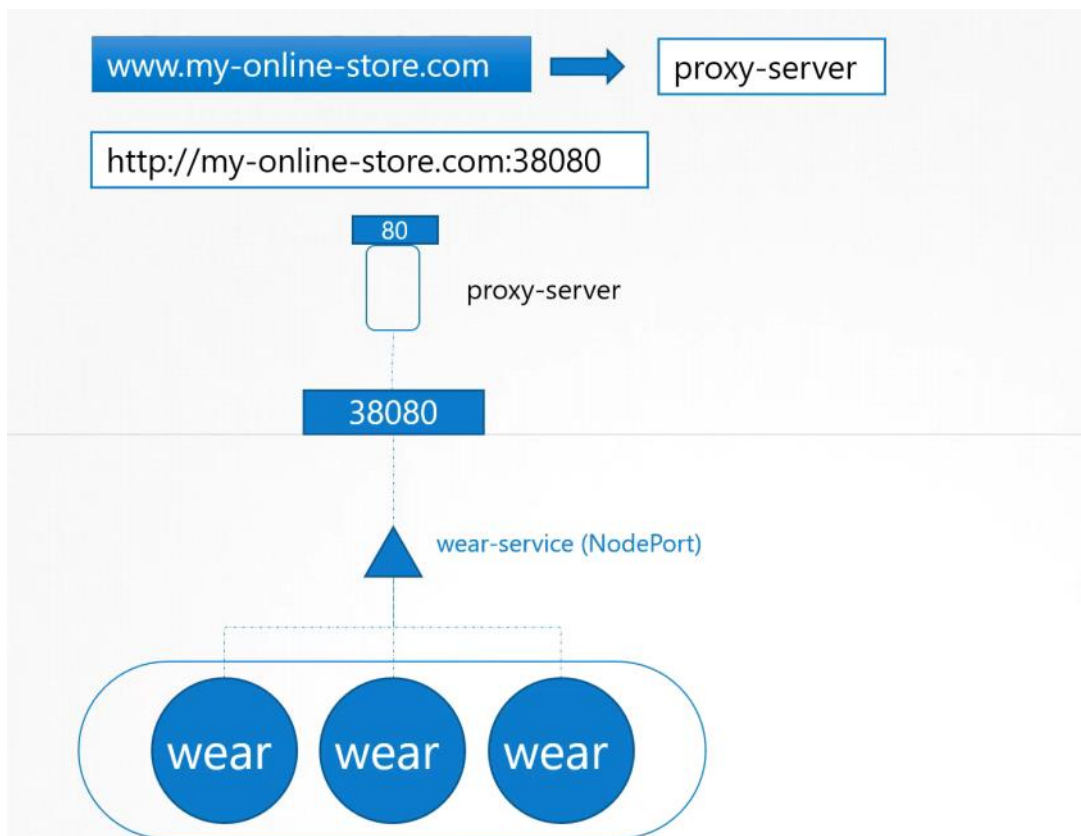


# INGRESS



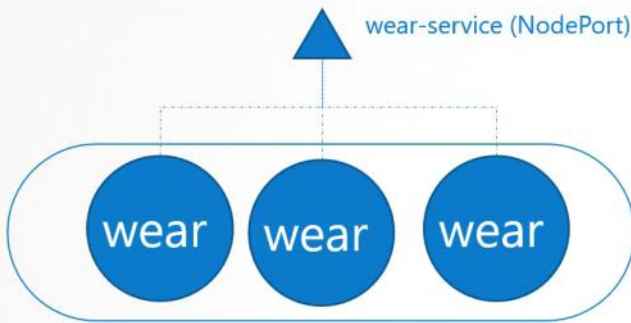


www.my-online-store.com



<node-ip>

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

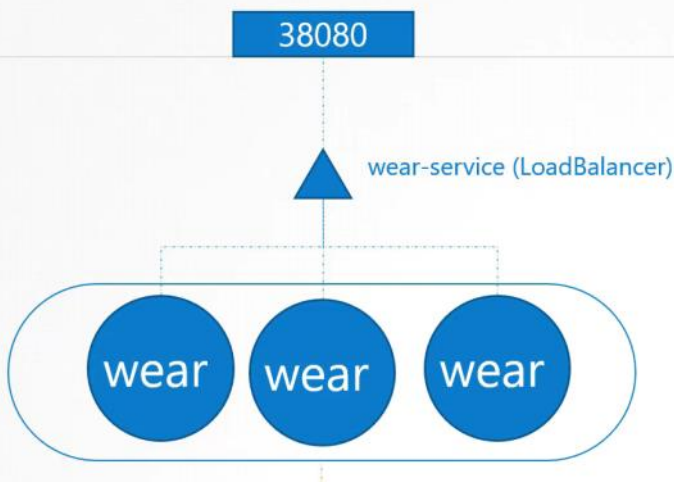


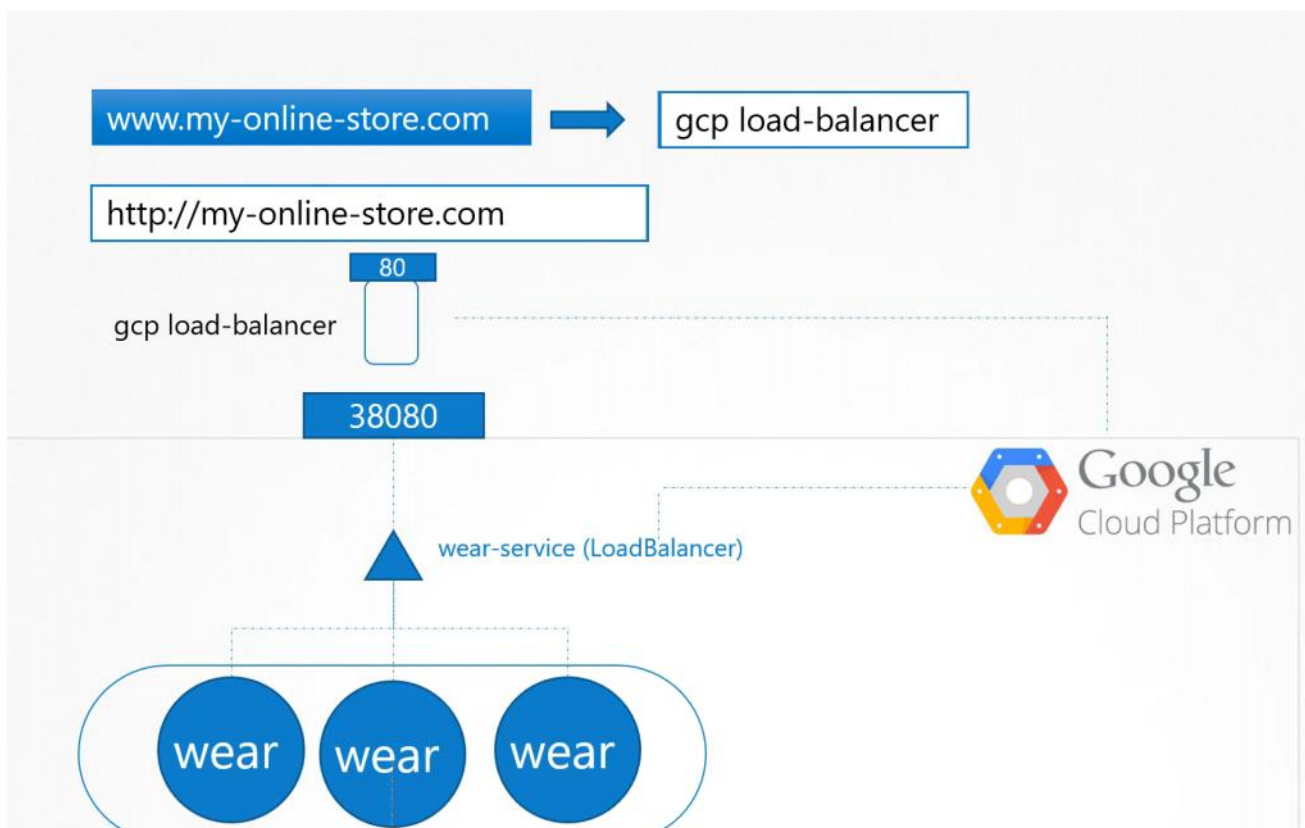
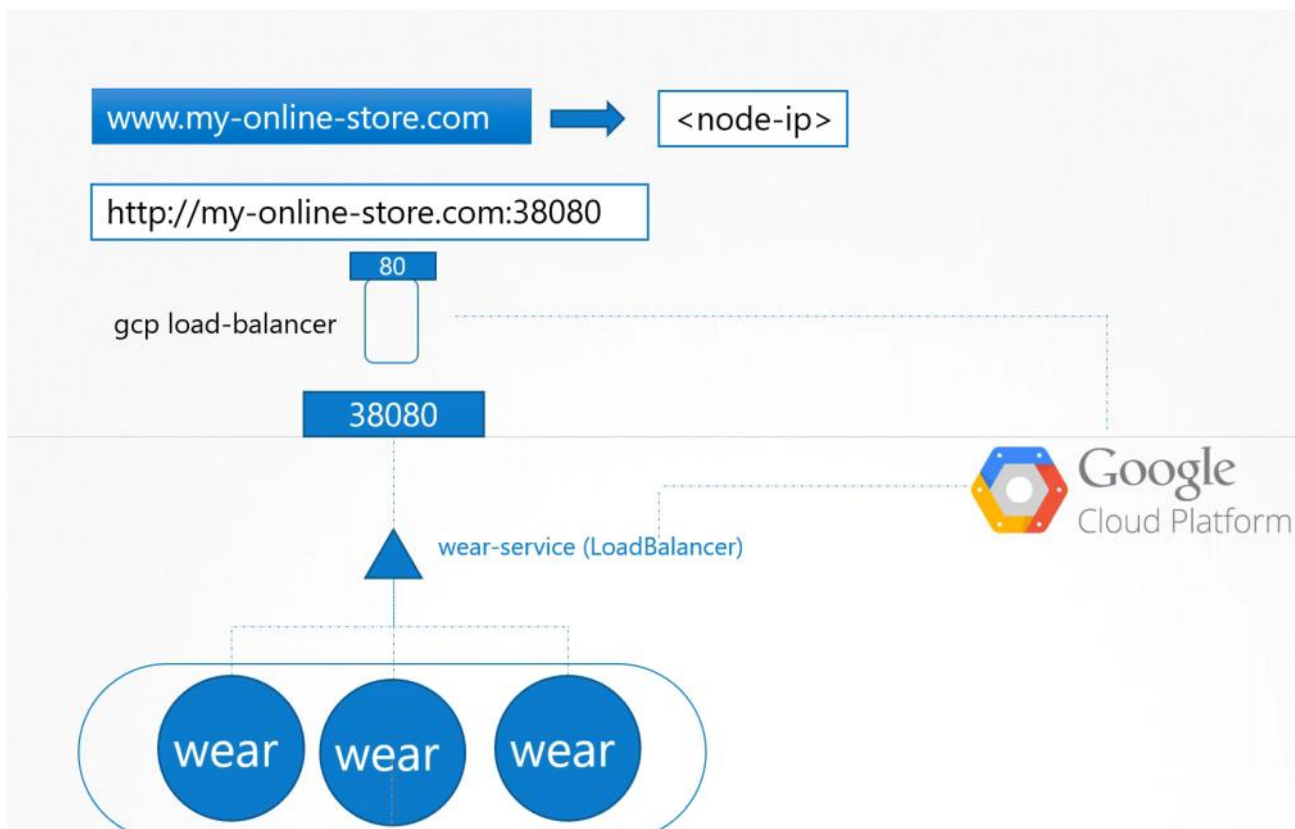
www.my-online-store.com



<node-ip>

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





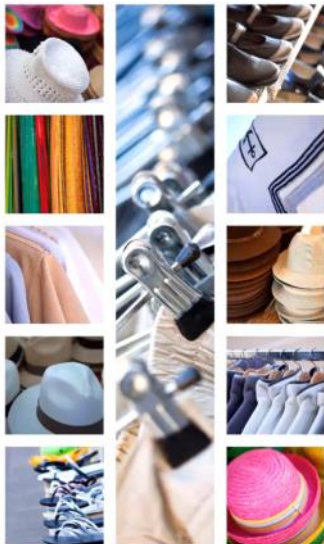
[www.my-online-store.com](http://www.my-online-store.com)



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



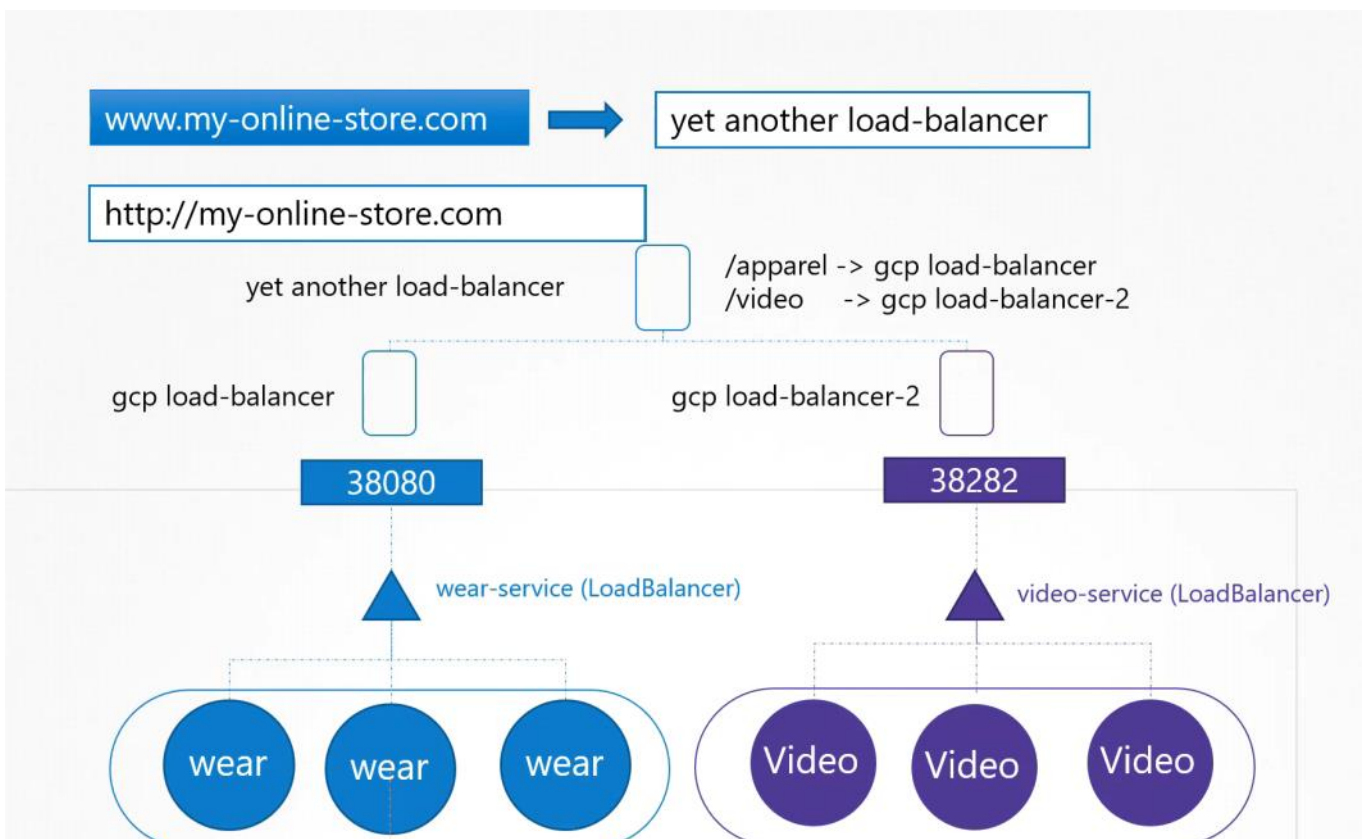
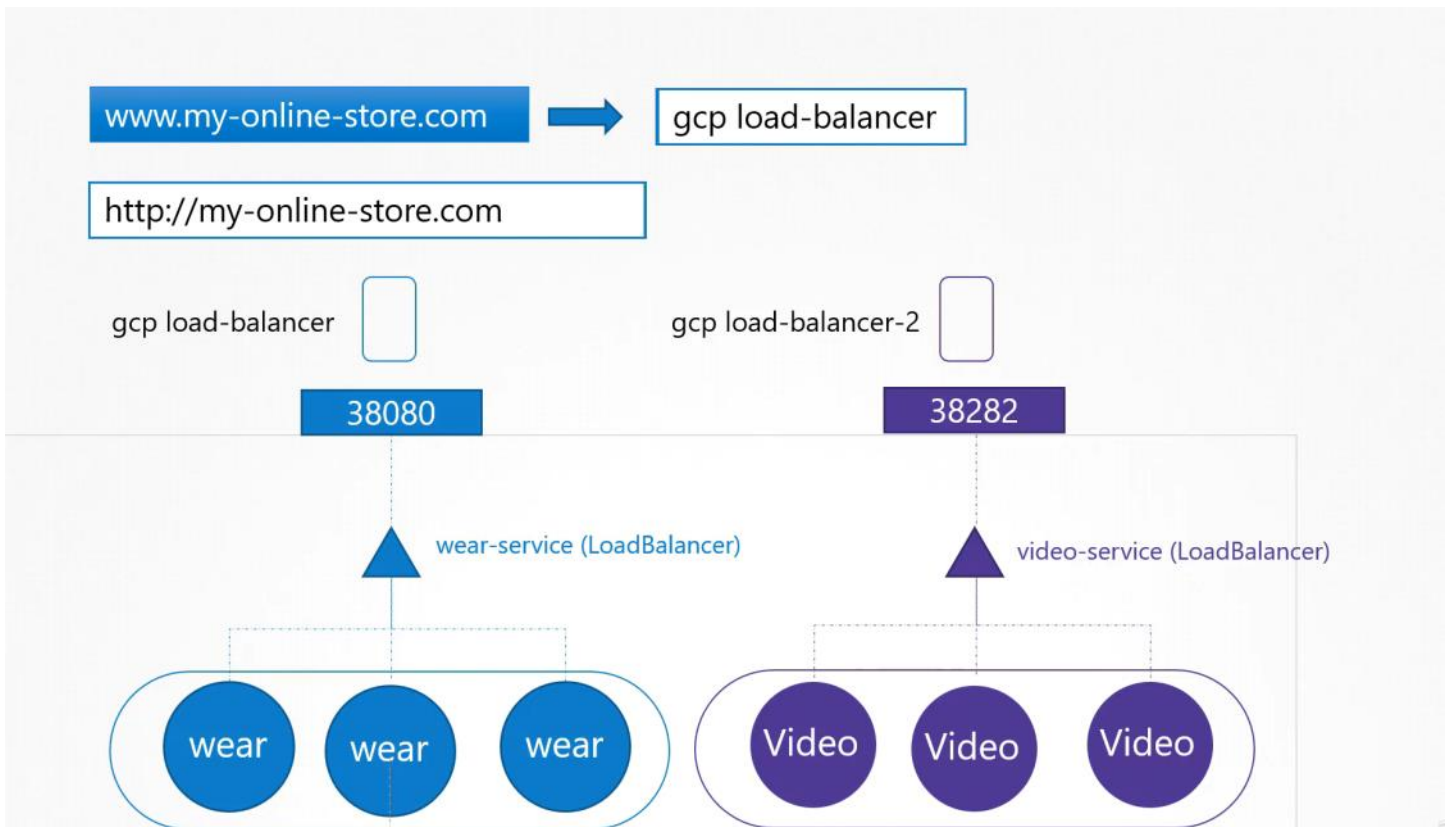
[www.my-online-store.com/wear](http://www.my-online-store.com/wear)

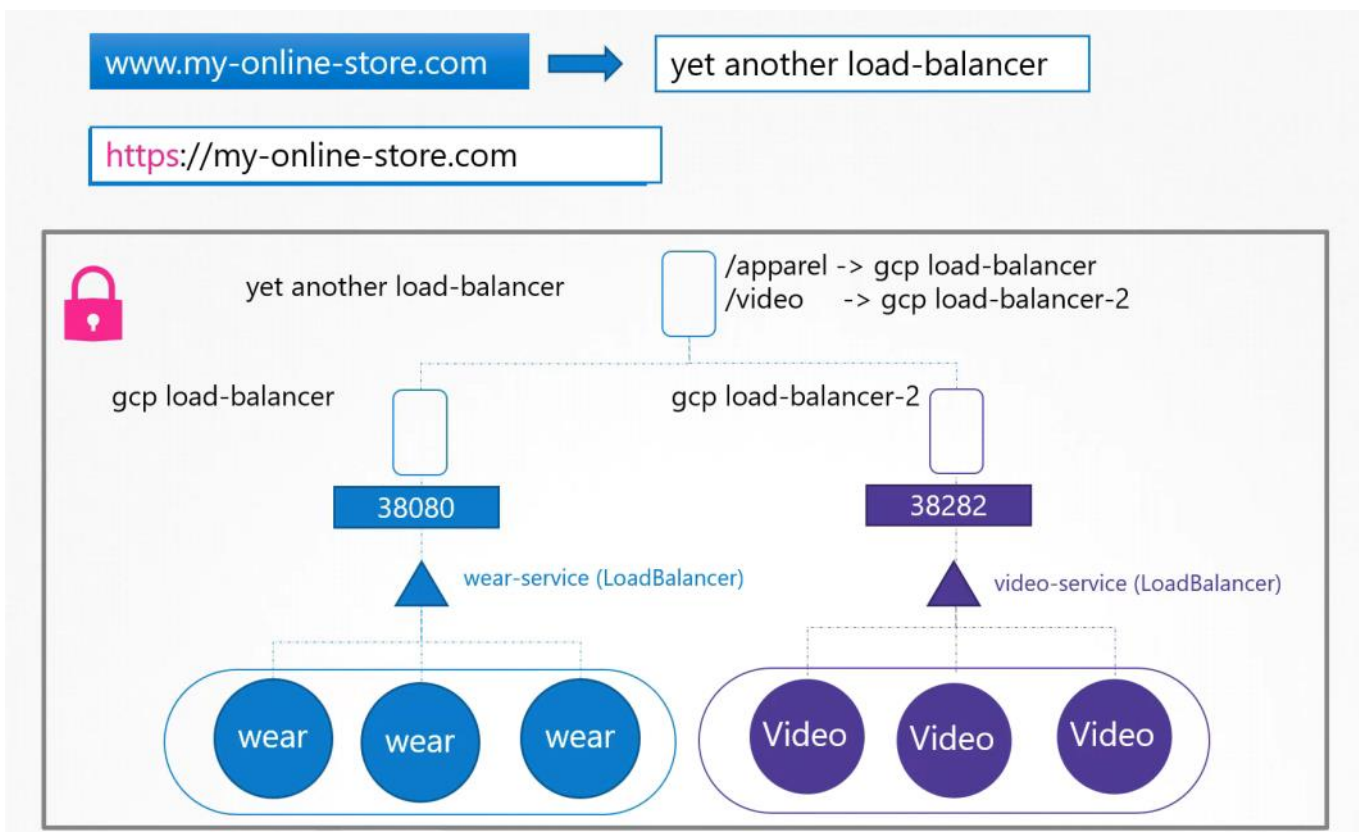
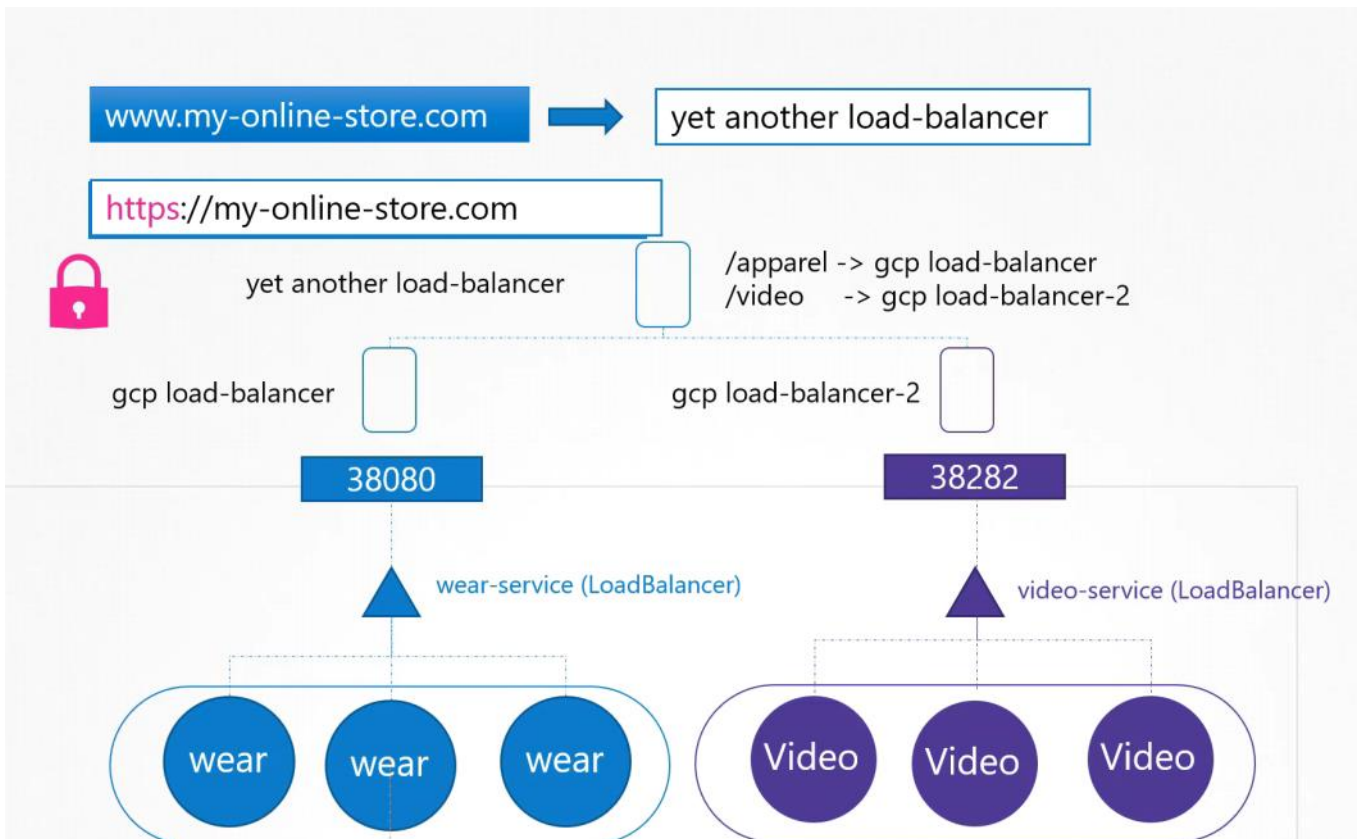


[www.my-online-store.com/watch](http://www.my-online-store.com/watch)

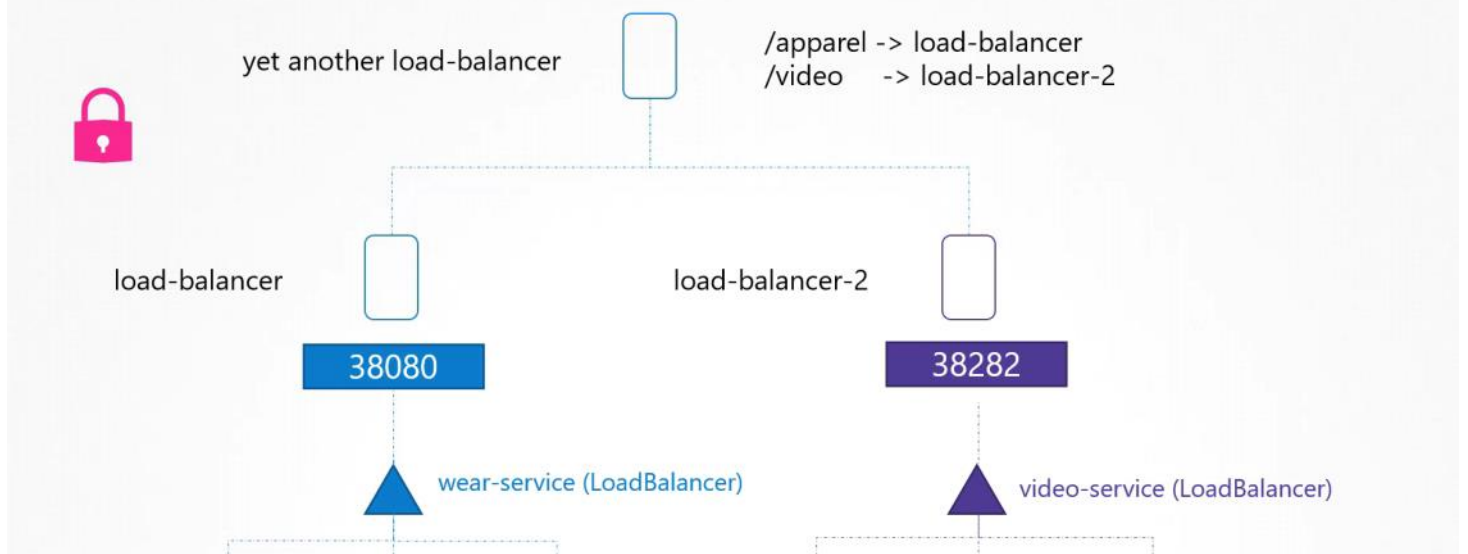




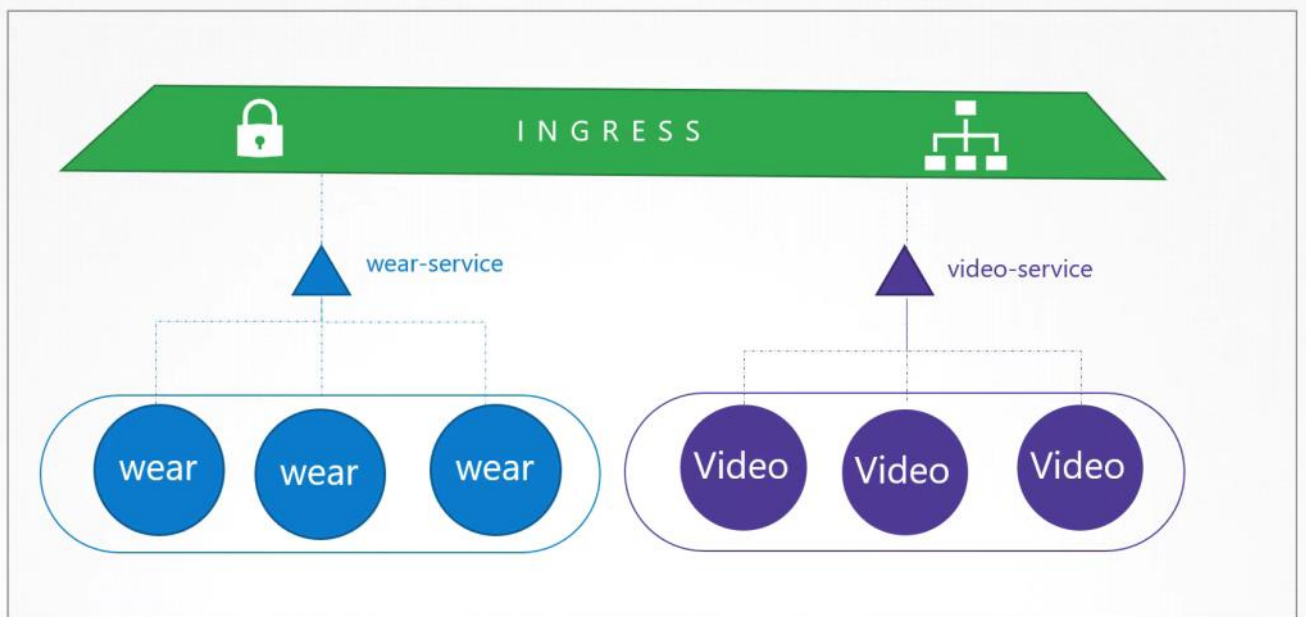




# Ingress

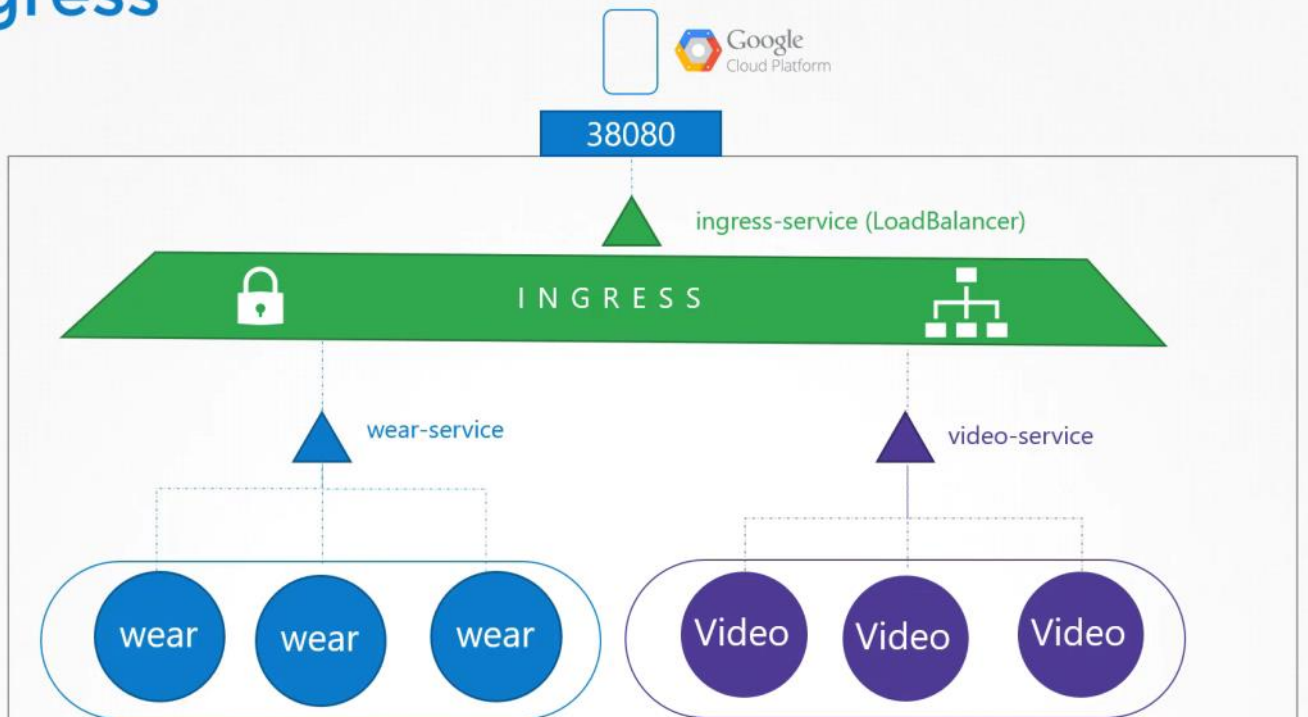


# Ingress





# Ingress



# Ingress

1. Deploy



2. Configure



# Ingress

1. Deploy

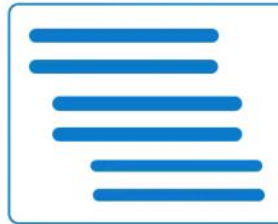


INGRESS CONTROLLER



2. Configure

INGRESS RESOURCES



# Ingress

1. Deploy



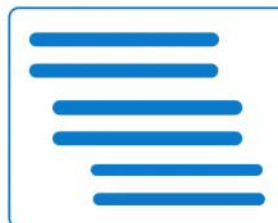
NOT deployed by default

INGRESS CONTROLLER



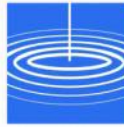
2. Configure

INGRESS RESOURCES



# INGRESS CONTROLLER

GCP HTTP(S)  
Load Balancer (GCE)



Contour



HAPROXY



træfik



Istio

# INGRESS CONTROLLER



```
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:
```

# INGRESS CONTROLLER



```
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
```

# INGRESS CONTROLLER



```
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
```

- err-log-path
- keep-alive
- ssl-protocols



# INGRESS CONTROLLER



- err-log-path
- keep-alive
- ssl-protocols

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
```

# 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
```



```

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

```

```

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

```



```

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

```

```

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

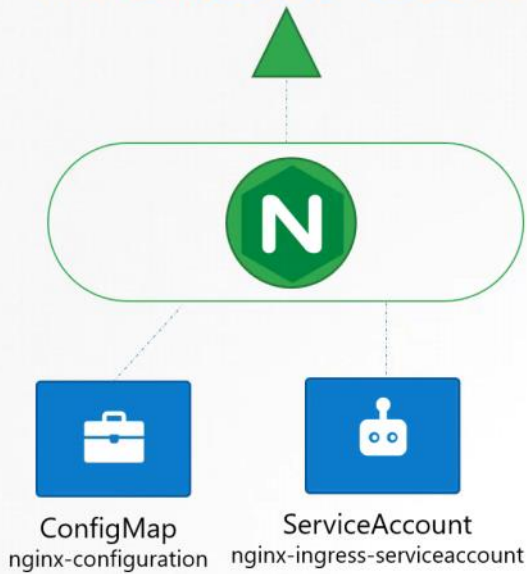
```

Roles

ClusterRoles

RoleBindings

# 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
          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

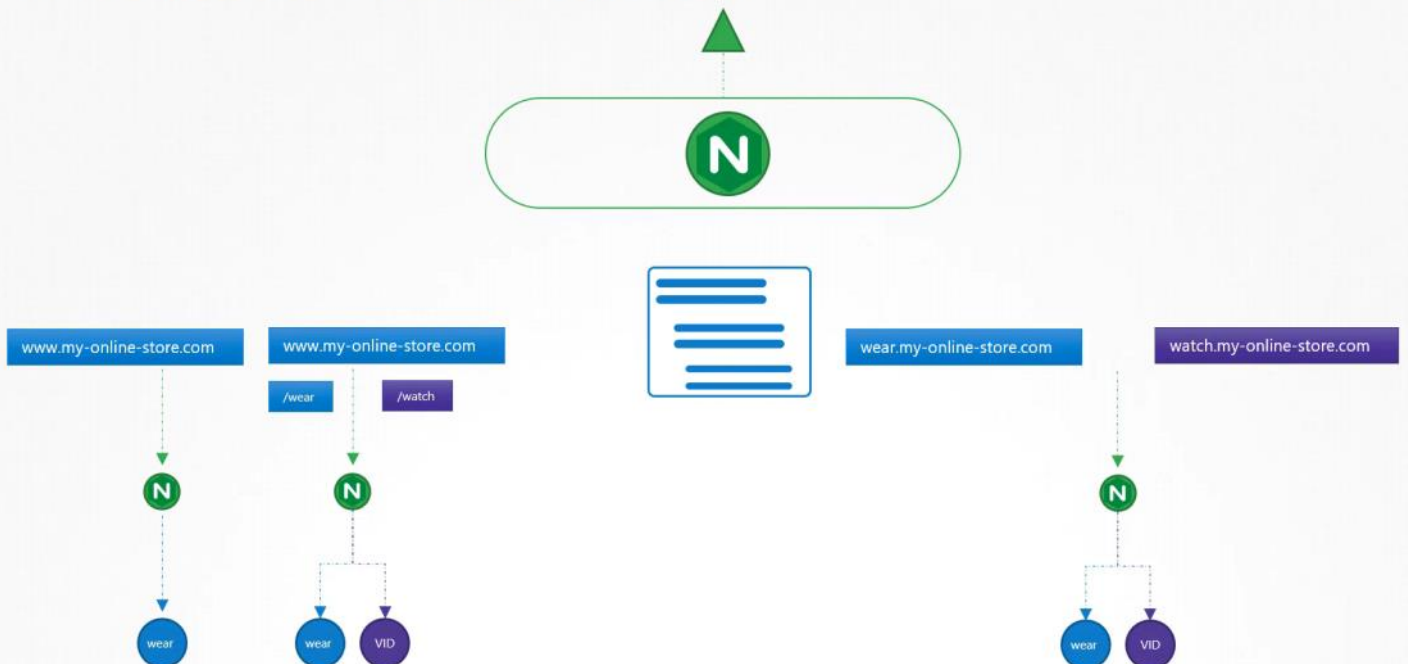
```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: nginx-ingress-serviceaccount
```

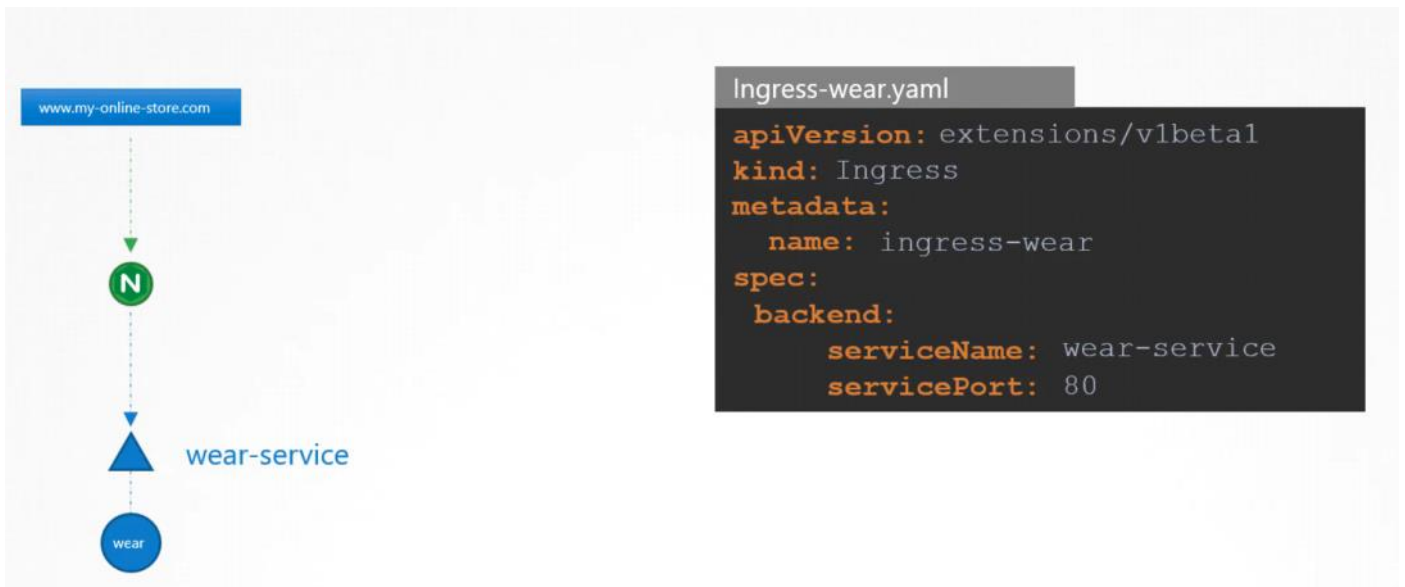
Roles

ClusterRoles

RoleBindings

# INGRESS RESOURCE





```

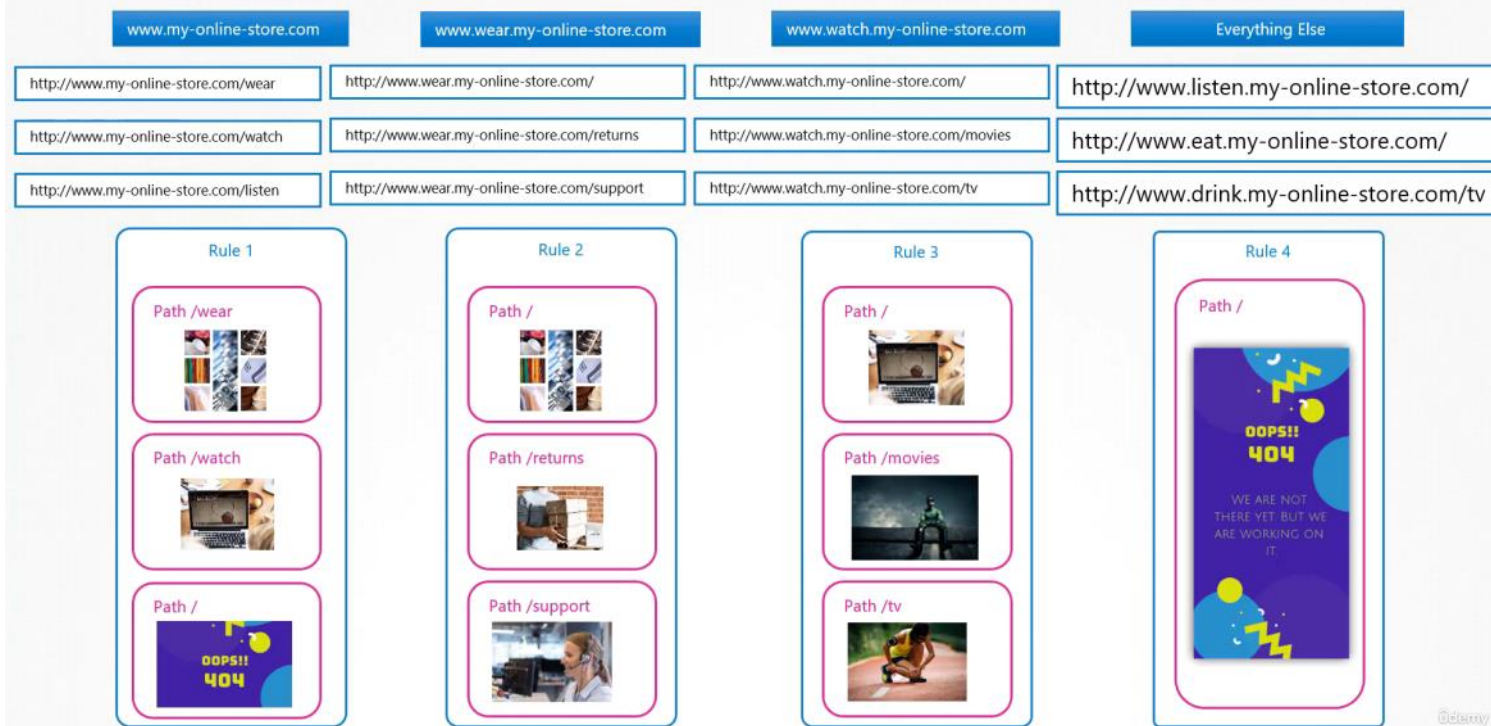
kubect1 create -f Ingress-wear.yaml
ingress.extensions/ingress-wear created
  
```

```

kubect1 get ingress
  
```

NAME	HOSTS	ADDRESS	PORTS
ingress-wear	*	80	2s

# INGRESS RESOURCE - RULES



```

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

```

```

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

```

```
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:

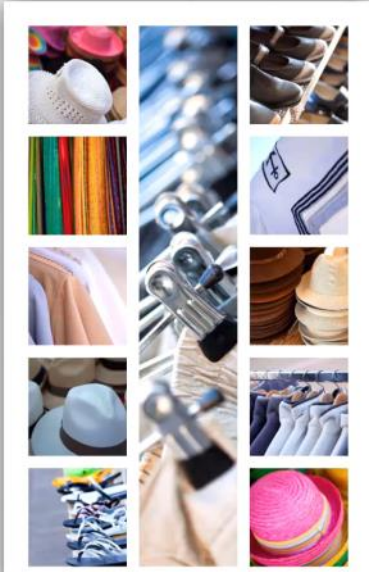
Events:

Type	Reason	Age	From	Message
Normal	CREATE	14s	nginx-ingress-controller	Ingress default/ingress-wear-watch



# INGRESS RESOURCE

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

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



# INGRESS RESOURCE

## 2 Rules

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

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

N

1 Path each

WEAR

VIDEO



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
```

### Article: Ingress

As we already discussed **Ingress** in our previous lecture. Here is an update.

In this article, we will see what changes have been made in previous and current versions in **Ingress**.

Like in **apiVersion**, **serviceName** and **servicePort** etc.

```
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
```

```
apiVersion: networking.k8s.io/v1
```

```
kind: Ingress
```

```
metadata:
```

```
  name: ingress-wear-watch
```

```
spec:
```

```
  rules:
```

```
  - http:
```

```
    paths:
```

```
    - path: /wear
```

```
      pathType: Prefix
```

```
      backend:
```

```
        service:
```

```
          name: wear-service
```

```
          port:
```

```
            number: 80
```

```
    - path: /watch
```

```
      pathType: Prefix
```

```
      backend:
```

```
        service:
```

```
          name: watch-service
```

```
          port:
```

```
            number: 80
```

Now, in k8s version 1.20+ we can create an Ingress resource from the imperative way like this:-

Format - `kubectl create ingress <ingress-name> --rule="host/path=service:port"`

Example - `kubectl create ingress ingress-test --rule="wear.my-online-store.com/wear*=wear-service:80"`

Find more information and examples in the below reference link:-

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#-em-ingress-em->

References:-

<https://kubernetes.io/docs/concepts/services-networking/ingress>

<https://kubernetes.io/docs/concepts/services-networking/ingress/#path-types>