

NGINX Gateway Fabric (NGF)

Deploy Service, Gateway, and Route

Deploy the service:

```
kubectl create -f - <<EOF
apiVersion: apps/v1
kind: Deployment
metadata:
  name: coffee
spec:
  replicas: 1
  selector:
    matchLabels:
      app: coffee
  template:
    metadata:
      labels:
        app: coffee
    spec:
      containers:
      - name: coffee
        image: nginxdemos/nginx-hello:plain-text
        ports:
        - containerPort: 8080
---
apiVersion: v1
kind: Service
metadata:
  name: coffee
spec:
  ports:
  - port: 80
    targetPort: 8080
    protocol: TCP
    name: http
  selector:
    app: coffee
EOF
```

Check new pod and services:

```
kubectl get pod,svc -owide
```

Deploy gateway:

```
kubectl create -f - <<EOF
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: cafe-gateway
spec:
  gatewayClassName: nginx
  listeners:
  - name: http
    port: 80
    protocol: HTTP
    hostname: coffee.lab.f5npi.net
EOF
```

Check the new cafe-gateway health:

```
kubectl get gateways cafe-gateway
kubectl describe gateways cafe-gateway
```

Create HTTP Route:

```
kubectl create -f - <<EOF
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: coffee
spec:
  parentRefs:
  - name: cafe-gateway
    sectionName: http
  rules:
  - matches:
    - path:
        type: PathPrefix
        value: /coffee
    backendRefs:
    - name: coffee
      port: 80
EOF
```

Check the route:

```
kubectl describe httproutes coffee
kubectl get httproutes coffee
```

Test the route:

```
curl coffee.lab.f5npi.net/coffee
```

Clean up:

```
kubectl delete deployments.apps coffee
kubectl delete service coffee
kubectl delete gateways cafe-gateway
kubectl delete httproutes coffee
```

Cross Namespace Services

Create services:

```
kubectl create -f - <<EOF
apiVersion: v1
kind: Namespace
metadata:
  name: retail
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hats
  namespace: retail
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hats
  template:
    metadata:
      labels:
        app: hats
    spec:
      containers:
```

```
    - name: hats
      image: nginxdemos/nginx-hello:plain-text
      ports:
        - containerPort: 8080
---
apiVersion: v1
kind: Service
metadata:
  name: hats
  namespace: retail
spec:
  ports:
    - port: 80
      targetPort: 8080
      protocol: TCP
      name: http
  selector:
    app: hats
EOF
```

Check service:

```
kubectl -n retail get pod,svc -owide
```

Deploy Gateway:

```
kubectl create -f - <<EOF
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: retail-gateway
spec:
  gatewayClassName: nginx
  listeners:
    - name: http
      port: 80
      protocol: HTTP
      hostname: hats.lab.f5npi.net
EOF
```

Check gateway:

```
kubectl get gateways retail-gateway
```

```
kubectl describe gateways retail-gateway
```

Deploy route:

```
kubectl create -f - <<EOF
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: retail-httproute
spec:
  parentRefs:
  - name: retail-gateway
    sectionName: http
  rules:
  - matches:
    - path:
        type: PathPrefix
        value: /hats
    backendRefs:
    - name: hats
      namespace: retail
      port: 80
EOF
```

Test route:

```
kubectl get httproutes retail-httproute
kubectl describe httproutes retail-httproute
```

Test application:

```
curl hats.lab.f5npi.net/hats
```

Create ReferenceGrant:

```
kubectl create -f - <<EOF
apiVersion: gateway.networking.k8s.io/v1beta1
kind: ReferenceGrant
metadata:
  name: access-to-retail-services
  namespace: retail
spec:
  to:
```

```
- group: ""
  kind: Service
from:
- group: gateway.networking.k8s.io
  kind: HTTPRoute
  namespace: default
EOF
```

Check the ReferenceGrant:

```
kubectl get -n retail referencegrants access-to-retail-services
kubectl describe -n retail referencegrants access-to-retail-services
```

Test application:

```
curl hats.lab.f5npi.net/hats
```

Clean Up:

```
kubectl -n retail delete deployments.apps hats
kubectl -n retail delete service hats
kubectl delete namespace retail
kubectl delete gateways retail-gateway
kubectl delete httproutes retail-httproute
kubectl delete -n retail referencegrants access-to-retail-services
```

Request Header Modification

deploy service:

```
kubectl create -f - <<EOF
apiVersion: apps/v1
kind: Deployment
metadata:
  name: whatigot
spec:
  replicas: 1
  selector:
    matchLabels:
      app: whatigot
  template:
    metadata:
```

```
  labels:
    app: whatigot
spec:
  containers:
  - name: whatigot
    image: nginx
    ports:
    - containerPort: 8080
    volumeMounts:
    - name: config-volume
      mountPath: /etc/nginx
      readOnly: true
  volumes:
  - name: config-volume
    configMap:
      name: whatigot-config
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: whatigot-config
data:
  nginx.conf: |-
    user  nginx;
    worker_processes  1;
    pid          /var/run/nginx.pid;
    load_module  /usr/lib/nginx/modules/nginx_http_js_module.so;
    events {}
    http {
        default_type text/plain;
        js_import /etc/nginx/headers.js;
        js_set  \${headers} headers.getRequestHeaders;
        server {
            listen 8080;
            return 200 "\${headers}";
        }
    }
  headers.js: |-
    function getRequestHeaders(r) {
        let s = "Headers:\n";
        for (let h in r.headersIn) {
            s += ` header '\${h}' is '\${r.headersIn[h]}'\n`;
        }
        return s;
    }
    export default {getRequestHeaders};
```

```
---
apiVersion: v1
kind: Service
metadata:
  name: whatigot
spec:
  ports:
    - port: 80
      targetPort: 8080
      protocol: TCP
      name: http
  selector:
    app: whatigot
EOF
```

deploy gateway:

```
kubectl create -f - <<EOF
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: gateway
spec:
  gatewayClassName: nginx
  listeners:
    - name: http
      port: 80
      protocol: HTTP
EOF
```

deploy route:

```
kubectl create -f - <<EOF
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: whatigot
spec:
  parentRefs:
    - name: gateway
      sectionName: http
  hostnames:
    - "whatigot.lab.f5npi.net"
  rules:
    - matches:
```



```
- path:
  type: PathPrefix
  value: /whatigot
filters:
- type: RequestHeaderModifier
  requestHeaderModifier:
  set:
  - name: My-favorite-band-today
    value: The-Fratellis
  add:
  - name: My-list-of-bands
    value: Black-Sabbath
  remove:
  - User-Agent
backendRefs:
- name: whatigot
  port: 80
EOF
```

test app:

```
curl -v http://whatigot.lab.f5npi.net/whatigot -H 'My-list-of-bands:Led-Zeppelin,Rush' -H 'My-favorite-band-today:Muse'
```

clean up:

```
kubectl delete deployments whatigot
kubectl delete service whatigot
kubectl delete gateways gateway
kubectl delete httproutes whatigot
kubectl delete configmap whatigot-config
```