GKE integration --version1.3 2022.03.31

Prerequisites

The installation assumes the following:

- There is a pre-existing Kubernetes (K8s) cluster with an ingress controller configured
- The cluster is managed by GKE(Google Kubernetes Engine)
- There is a kubeconfig file available with adequate permissions on the K8s cluster to: Manage namespaces
- Helm 3 is available and configured to use the kubeconfig.

Ingress Management

HO GKE project is using gce ingress not nginx ingress. The direct difference is gce ingress is built-in GKE, it can work with Google HTTPS load balancers well. Kubernetes including many kinds of manifests, helm is the package manager for Kubernetes. we can easily manage ingress by helm in HO project as we need.

Tree Structure

```
oss-hogke
└── gke-lumos
    ├── Chart.yaml
    ├── templates
    |   ├── _helpers.tpl
    |   ├── ingress.yaml
    |   └── NOTES.txt
    └── values.yaml
```

`oss-hogke` is the parent directory. There is a chart named `gke-lumos` is used for managing ingress rules.

If you need to add an ingress rule applying `demo2`, update `values.yaml` file:

Copy the following part as another parameter.

Before Updating:

```
ingress:
  enabled: true
  host: oss-gke.lumos.hyperoptic.com
  apps:
  - name: oss-od-web-rc
    port: 8080
    path:
      - externalPath: /oss/bc/od_web/appInfo/status
      - externalPath: /oss/static/oss_core/od
      - externalPath: /oss/oss_core/od
    pathType: Prefix
```

After Updating:

```
ingress:
  enabled: true
  host: oss-gke.lumos.hyperoptic.com
  apps:
  - name: oss-od-web-rc
    port: 8080
    path:
      - externalPath: /oss/bc/od_web/appInfo/status
      - externalPath: /oss/static/oss_core/od
      - externalPath: /oss/oss_core/od
    pathType: Prefix
  - name: demo2
    port: 8080
    path:
      - externalPath: /aaa/bbb/ccc
      - externalPath: /ddd/eee/fff
      - externalPath: /ggg/hhh/iii
    pathType: Prefix
```

Description for each parameter:
`ingress.enabled`: default value: `true`.
`ingress.host`: external domain host, here is `oss-gke.lumos.hyperoptic.com`.
`ingress.apps`: Display the forwarding rules of each micro service in the form of array.
`ingress.apps.path`: ingress path, format is `- externalPath: matchPath`. If there are multiple rules, configure them in sequence. Here is

```
- externalPath: /oss/edesign
- externalPath: /oss/bc/im_edesign/appInfo/status
- externalPath: /oss/auth
```

`ingress.pathType`: path match type, apply `Prefix` as default value.

Operation

1. List Release

```
helm ls -n hogke
```

2. Add Release

```
helm install gke-lumos oss-hogke/gke-lumos -n hogke
```

3. Upgrade Release

```
helm upgrade gke-lumos oss-hogke/gke-lumos -n hogke
```

4. Remove Release

```
helm uninstall gke-lumos -n hogke
```

Service Management

```
https://cloud.google.com/kubernetes-
engine/docs/concepts/ingress#multiple_backend_services
```

In the Service manifest, you must use `type: NodePort` unless you're using container native load balancing. If using container native load balancing, use the `type: ClusterIP`.

Create several yaml files by templates file `service-oss-pot-ci.yaml`

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    cloud.google.com/neg: '{"ingress":true}'
    service.alpha.kubernetes.io/app-protocols: '{"http-8080":"HTTP"}'
  name: oss-pot-ci
  namespace: hogke
spec:
  ports:
    - name: http-8080
      port: 8080
      protocol: TCP
      targetPort: 8080
  selector:
    zcm-app: oss-pot-ci
  type: NodePort
```

execute command to replace ClusterIP type service to NodePort type service.

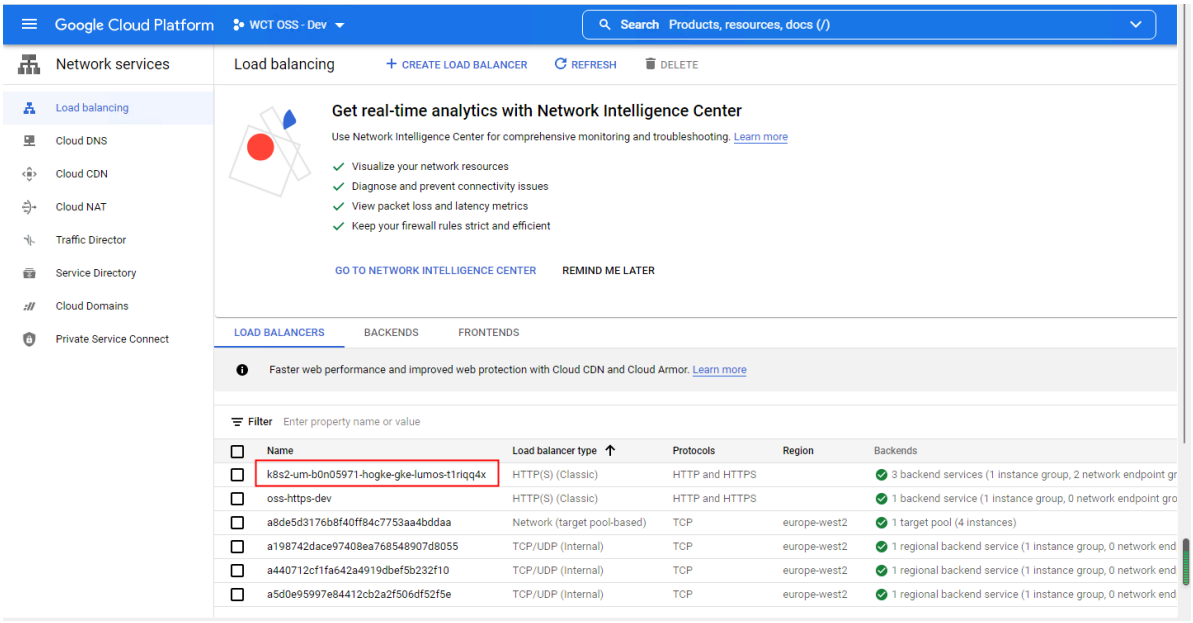
```
kubectl delete -f service-appname.yaml
kubectl apply -f service-appname.yaml
```

Load Balancing Rules Management

Because of gce ingress cannot fulfill rewrite-target function, we find a second solution, that is Load Balancing Rules Configuration.

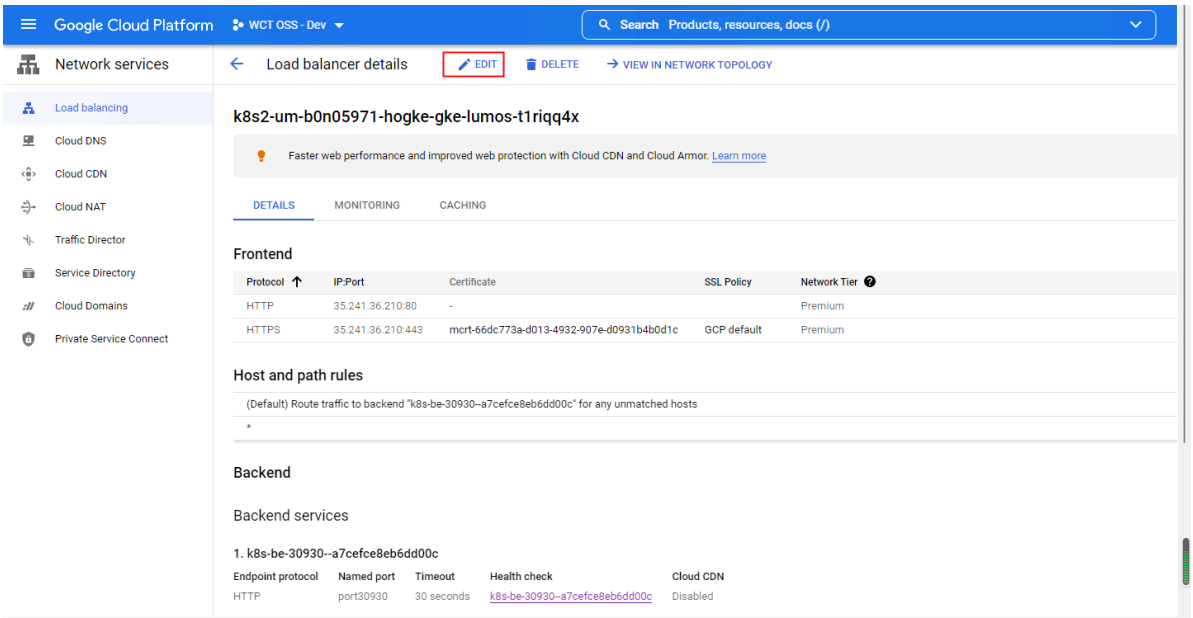
Click Load balancing

Click `LOAD BALANCERS` k8s2-um-b0n05971-hogke-gke-lumos-t1riqq4x



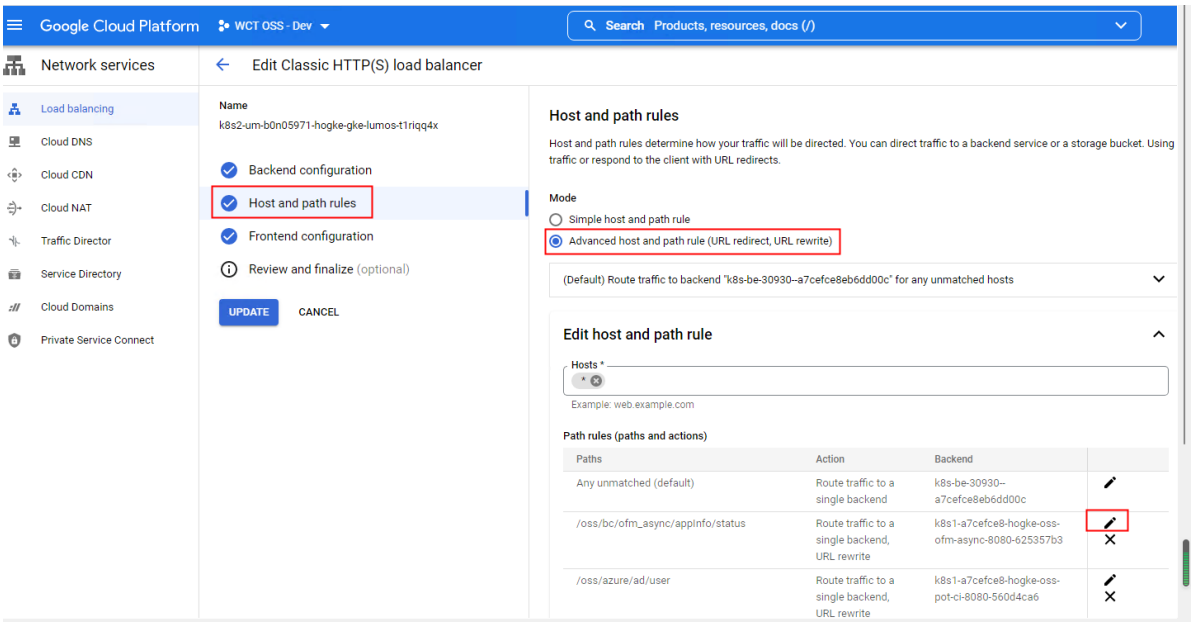
Click `EDIT`

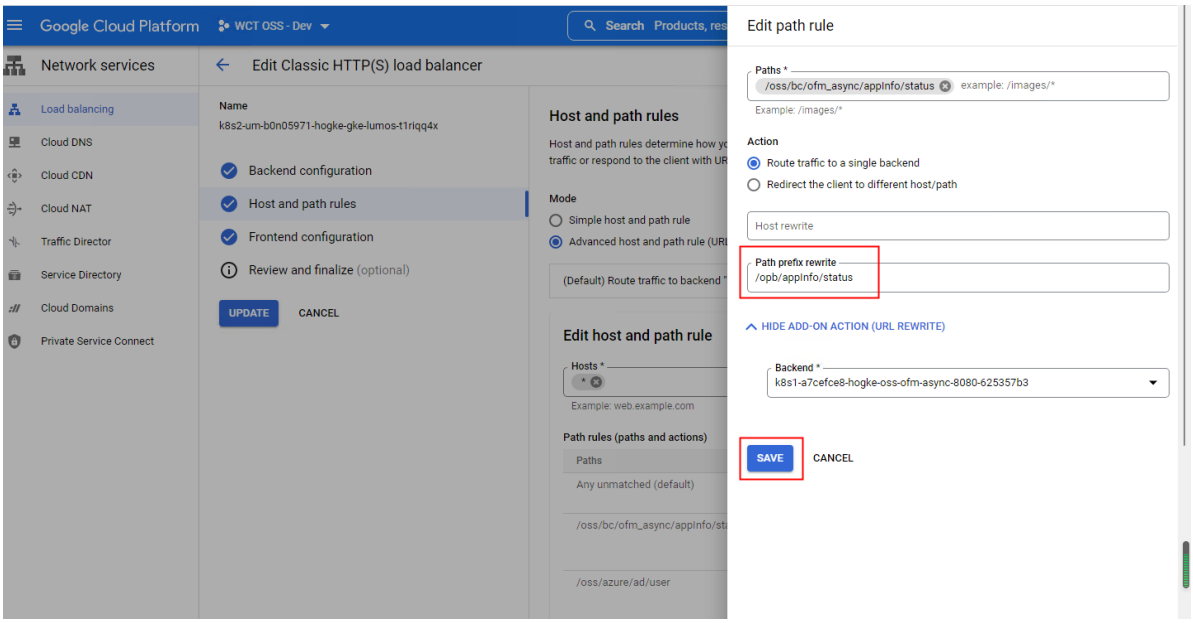
https://console.cloud.google.com/net-services/loadbalancing/details/http/k8s2-um-b0n05971-hogke-gke-lumos-t1riqq4x?project=wct-oss-dev



1. In the left column of the screen, click **Host and path rules**.
2. Select **Advanced host and path rule (URL redirect, URL rewrite)**.
3. Click the row that contains the non-default path rule, in this case, the row that has an asterisk (`*`) for all hosts.
4. Click the pencil icon edit for the specified row of apps.
5. Under **Action**, select default **Route traffic to a single backend**.
6. Click **Add-on action (URL rewrite)**.
7. Leave **Host rewrite** blank.
8. Under **Path prefix rewrite**, enter rewrite path prefix.
9. Under **Backend**, select default backend.
10. Click **Save**.
11. Click **Done**.
12. If everything looks correct, click **Update** to update your HTTP load balancer. It will cost several minutes to finish updating.

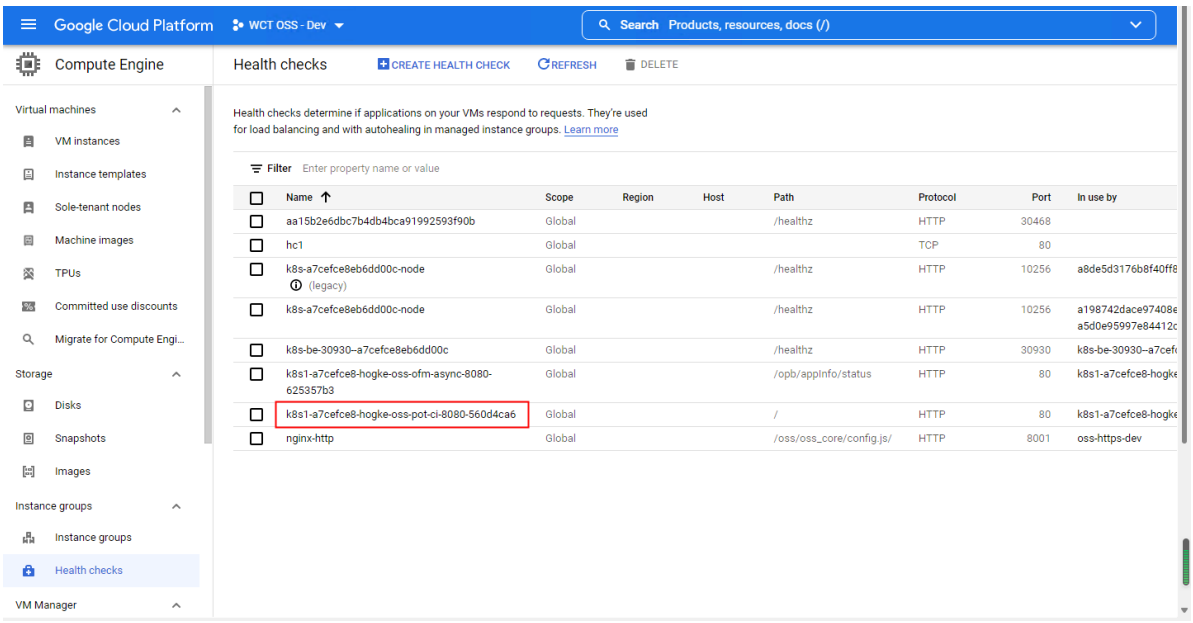
https://console.cloud.google.com/net-services/loadbalancing/edit/http/k8s2-um-b0n05971-hogke-gke-lumos-t1riqq4x?project=wct-oss-dev



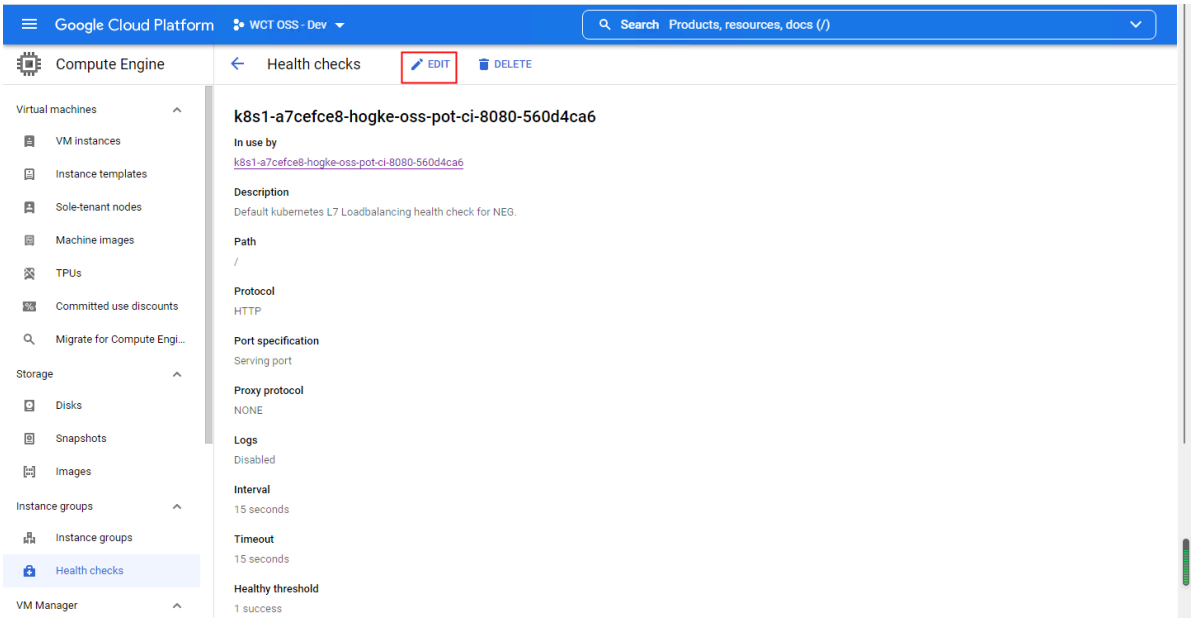


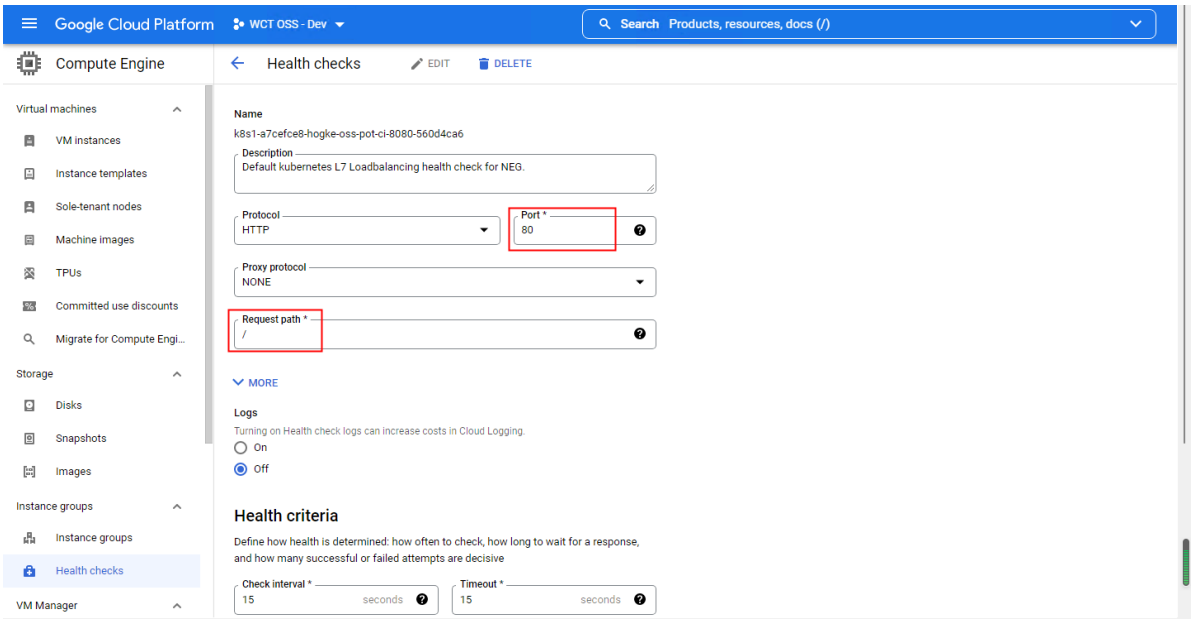
Health Checks Management

Click Health checks



[Learn more]



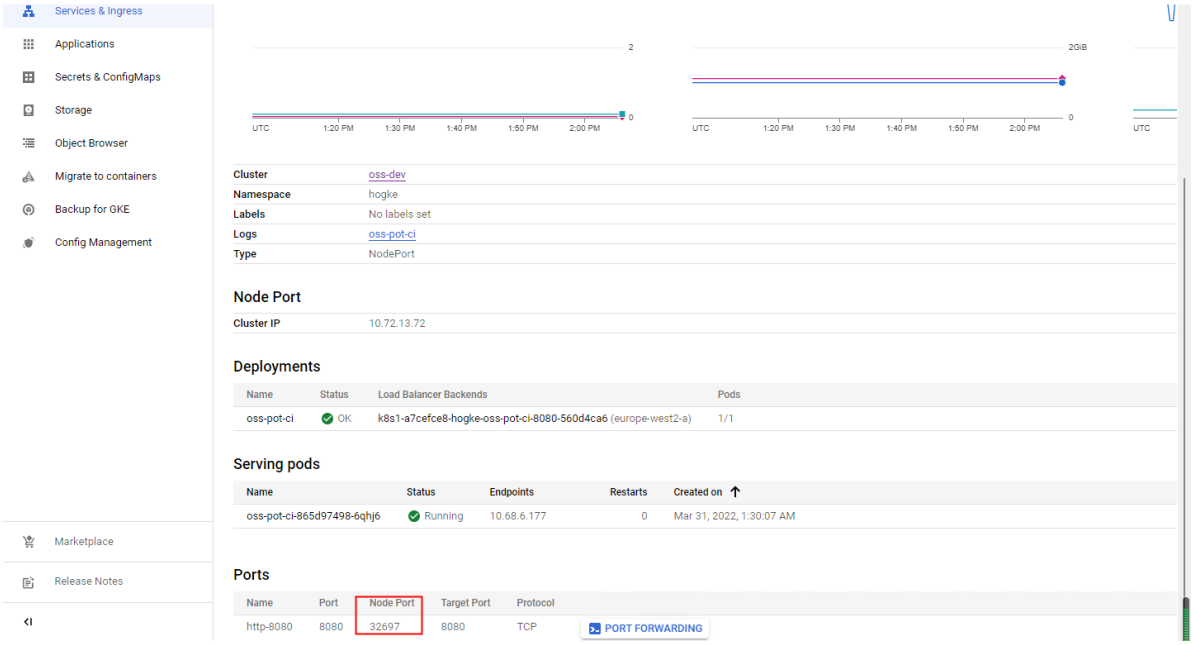


Port: get NodePort from service.

Request path: get path from gateway rules.

The method to test `Port` and `Request path`: execute below commands in container, test response code is 200 or not.

```
#curl -v 127.0.0.1:8080/opb/appInfo/status
* About to connect() to 127.0.0.1 port 8080 (#0)
*   Trying 127.0.0.1...
* Connected to 127.0.0.1 (127.0.0.1) port 8080 (#0)
> GET /opb/appInfo/status HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 127.0.0.1:8080
> Accept: */*
>
< HTTP/1.1 200
< ITRACING_TRACE_ID: 172024019230_42_1^1648701872628^11969
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
< Cache-Control: no-cache, no-store, max-age=0, must-revalidate
< Pragma: no-cache
< Expires: 0
< X-Frame-Options: SAMEORIGIN
< Content-Type: application/json;charset=UTF-8
< Transfer-Encoding: chunked
< Date: Thu, 31 Mar 2022 14:13:32 GMT
<
* Connection #0 to host 127.0.0.1 left intact
{"resultCode":"0","resultDesc":"success"}
```



To Test

https://oss-gke.lumos.hyperoptic.com/oss/bc/ofm_async/appInfo/status

or

https://35.241.36.210/oss/bc/ofm_async/appInfo/status

http or https both can work.