

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/cc
        - 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](#)

Google Cloud Platform WCT OSS - Dev

Search Products, resources, docs (/)

Network services

Load balancing

+ CREATE LOAD BALANCER REFRESH DELETE

Get real-time analytics with Network Intelligence Center

Use Network Intelligence Center for comprehensive monitoring and troubleshooting. [Learn more](#)

- Visualize your network resources
- Diagnose and prevent connectivity issues
- View packet loss and latency metrics
- Keep your firewall rules strict and efficient

GO TO NETWORK INTELLIGENCE CENTER REMIND ME LATER

LOAD BALANCERS BACKENDS FRONTENDS

Faster web performance and improved web protection with Cloud CDN and Cloud Armor. [Learn more](#)

Filter Enter property name or value

Name	Load balancer type	Protocols	Region	Backends
k8s2-um-b0n05971-hogke-gke-lumos-t1riqq4x	HTTP(S) (Classic)	HTTP and HTTPS		3 backend services (1 instance group, 2 network endpoint gr
oss-https-dev	HTTP(S) (Classic)	HTTP and HTTPS		1 backend service (1 instance group, 0 network endpoint gro
a8de5d3176b8f40ff84c7753aa4bddaa	Network (target pool-based)	TCP	europa-west2	1 target pool (4 instances)
a198742dace97408ea768548907d8055	TCP/UDP (Internal)	TCP	europa-west2	1 regional backend service (1 instance group, 0 network end
a440712cf1fa642a4919dbef5b232f10	TCP/UDP (Internal)	TCP	europa-west2	1 regional backend service (1 instance group, 0 network end
a5d0e95997e84412cb2a2f506df52f5e	TCP/UDP (Internal)	TCP	europa-west2	1 regional backend service (1 instance group, 0 network end

Click **EDIT**

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

Google Cloud Platform WCT OSS - Dev

Search Products, resources, docs (/)

Network services

Load balancing

Load balancer details

EDIT DELETE VIEW IN NETWORK TOPOLOGY

k8s2-um-b0n05971-hogke-gke-lumos-t1riqq4x

Faster web performance and improved web protection with Cloud CDN and Cloud Armor. [Learn more](#)

DETAILS MONITORING CACHING

Frontend

Protocol	IP:Port	Certificate	SSL Policy	Network Tier
HTTP	35.241.36.210:80	-		Premium
HTTPS	35.241.36.210:443	mcr1-66dc773a-d013-4932-907e-d0931b4b0d1c	GCP default	Premium

Host and path rules

(Default) Route traffic to backend "k8s-be-30930-a7cefce8eb6dd00c" for any unmatched hosts

*

Backend

Backend services

1. k8s-be-30930-a7cefce8eb6dd00c

Endpoint protocol	Named port	Timeout	Health check	Cloud CDN
HTTP	port30930	30 seconds	k8s-be-30930-a7cefce8eb6dd00c	Disabled

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>

Google Cloud Platform WCT OSS - Dev

Network services

Load balancing

Cloud DNS

Cloud CDN

Cloud NAT

Traffic Director

Service Directory

Cloud Domains

Private Service Connect

Edit Classic HTTP(S) load balancer

Name: k8s2-um-b0n05971-hogke-gke-lumos-t1riqq4x

Backend configuration

Host and path rules

Frontend configuration

Review and finalize (optional)

UPDATE CANCEL

Host and path rules

Host and path rules determine how your traffic will be directed. You can direct traffic to a backend service or a storage bucket. Using traffic or respond to the client with URL redirects.

Mode

Simple host and path rule

Advanced host and path rule (URL redirect, URL rewrite)

(Default) Route traffic to backend "k8s-be-30930-a7cefce8eb6dd00c" for any unmatched hosts

Edit host and path rule

Hosts *

Example: web.example.com

Path rules (paths and actions)

Paths	Action	Backend	
Any unmatched (default)	Route traffic to a single backend	k8s-be-30930-a7cefce8eb6dd00c	
/oss/bc/ofm_async/appinfo/status	Route traffic to a single backend, URL rewrite	k8s1-a7cefce8-hogke-oss-ofm-async-8080-625357b3	
/oss/azure/ad/user	Route traffic to a single backend, URL rewrite	k8s1-a7cefce8-hogke-oss-pot-ci-8080-560d4ca6	

Google Cloud Platform WCT OSS - Dev

Network services

Load balancing

Cloud DNS

Cloud CDN

Cloud NAT

Traffic Director

Service Directory

Cloud Domains

Private Service Connect

Edit Classic HTTP(S) load balancer

Name: k8s2-um-b0n05971-hogke-gke-lumos-t1riqq4x

Backend configuration

Host and path rules

Frontend configuration

Review and finalize (optional)

UPDATE CANCEL

Host and path rules

Host and path rules determine how your traffic or respond to the client with URL

Mode

Simple host and path rule

Advanced host and path rule (URL redirect, URL rewrite)

(Default) Route traffic to backend "k8s-be-30930-a7cefce8eb6dd00c" for any unmatched hosts

Edit host and path rule

Hosts *

Example: web.example.com

Path rules (paths and actions)

Paths

Any unmatched (default)

/oss/bc/ofm_async/appinfo/status

/oss/azure/ad/user

Edit path rule

Paths *

/oss/bc/ofm_async/appinfo/status example: /images/*

Example: /images/*

Action

Route traffic to a single backend

Redirect the client to different host/path

Host rewrite

Path prefix rewrite

/opb/appinfo/status

HIDE ADD-ON ACTION (URL REWRITE)

Backend *

k8s1-a7cefce8-hogke-oss-ofm-async-8080-625357b3

SAVE CANCEL

Health Checks Management

Click [Health checks](#)

Google Cloud Platform WCT OSS - Dev

Search Products, resources, docs (/)

Compute Engine

Health checks CREATE HEALTH CHECK REFRESH DELETE

Health checks determine if applications on your VMs respond to requests. They're used for load balancing and with autohealing in managed instance groups. [Learn more](#)

Filter Enter property name or value

Name	Scope	Region	Host	Path	Protocol	Port	In use by
aa15b2e6dbc7b4db4bca91992593f90b	Global			/healthz	HTTP	30468	
hc1	Global				TCP	80	
k8s-a7cefce8eb6dd00c-node (legacy)	Global			/healthz	HTTP	10256	a8de5d3176b8f40ff8
k8s-a7cefce8eb6dd00c-node	Global			/healthz	HTTP	10256	a198742dace97408e5d0e95997e84412c
k8s-be-30930-a7cefce8eb6dd00c	Global			/healthz	HTTP	30930	k8s-be-30930-a7cef
k8s1-a7cefce8-hogke-oss-ofm-async-8080-625357b3	Global			/opb/appinfo/status	HTTP	80	k8s1-a7cefce8-hogke
k8s1-a7cefce8-hogke-oss-pot-ci-8080-560d4ca6	Global			/	HTTP	80	k8s1-a7cefce8-hogke
nginx-http	Global			/oss/oss_core/config.js/	HTTP	8001	oss-https-dev

[Learn more](#)

Google Cloud Platform WCT OSS - Dev

Search Products, resources, docs (/)

Compute Engine

Health checks EDIT DELETE

k8s1-a7cefce8-hogke-oss-pot-ci-8080-560d4ca6

In use by
[k8s1-a7cefce8-hogke-oss-pot-ci-8080-560d4ca6](#)

Description
Default kubernetes L7 Loadbalancing health check for NEG.

Path
/

Protocol
HTTP

Port specification
Serving port

Proxy protocol
NONE

Logs
Disabled

Interval
15 seconds

Timeout
15 seconds

Healthy threshold
1 success

Google Cloud Platform WCT OSS - Dev

Search Products, resources, docs (/)

Compute Engine

Health checks EDIT DELETE

Name
k8s1-a7cefce8-hogke-oss-pot-ci-8080-560d4ca6

Description
Default kubernetes L7 Loadbalancing health check for NEG.

Protocol
HTTP

Port *
80

Proxy protocol
NONE

Request path *
/

MORE

Logs
Turning on Health check logs can increase costs in Cloud Logging.
☐ On
☒ Off

Health criteria
Define how health is determined: how often to check, how long to wait for a response, and how many successful or failed attempts are decisive

Check interval *
15 seconds

Timeout *
15 seconds

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"}
```

The screenshot shows the Google Cloud Platform console interface. On the left is a sidebar with navigation options: Services & Ingress, Applications, Secrets & ConfigMaps, Storage, Object Browser, Migrate to containers, Backup for GKE, and Config Management. The main panel displays details for a service named 'oss-dev' in the 'hogke' namespace. It shows the cluster IP as 10.72.13.72 and the type as NodePort. Below this, there are sections for Deployments and Serving pods. The Deployments section shows a single deployment 'oss-pot-ci' with a status of 'OK'. The Serving pods section shows a single pod 'oss-pot-ci-865d97498-6qhj6' with a status of 'Running'. At the bottom, the 'Ports' section shows a table with columns: Name, Port, Node Port, Target Port, and Protocol. The table has one entry: 'http-8080' with Port '8080', Node Port '32697' (highlighted with a red box), Target Port '8080', and Protocol 'TCP'. A 'PORT FORWARDING' button is visible next to the table.

Name	Port	Node Port	Target Port	Protocol
http-8080	8080	32697	8080	TCP

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.

