Kubernetes hello-app

https://cloud.google.com/kubernetes-engine/docs/tutorials/hello-app

Deploying a containerized web application

This tutorial shows you how to package a web application in a Docker container image, and run that container image on a Google Kubernetes Engine cluster as a load-balanced set of replicas that can scale to the needs of your users.

Objectives

To package and deploy your application on GKE, you must:

1. Package your app into a Docker image
2. Run the container locally on your machine (optional)
3. Upload the image to a registry
4. Create a container cluster
5. Deploy your app to the cluster
6. Expose your app to the Internet
7. Scale up your deployment
8. Deploy a new version of your app

Build the container image

gcloud components install kubectl

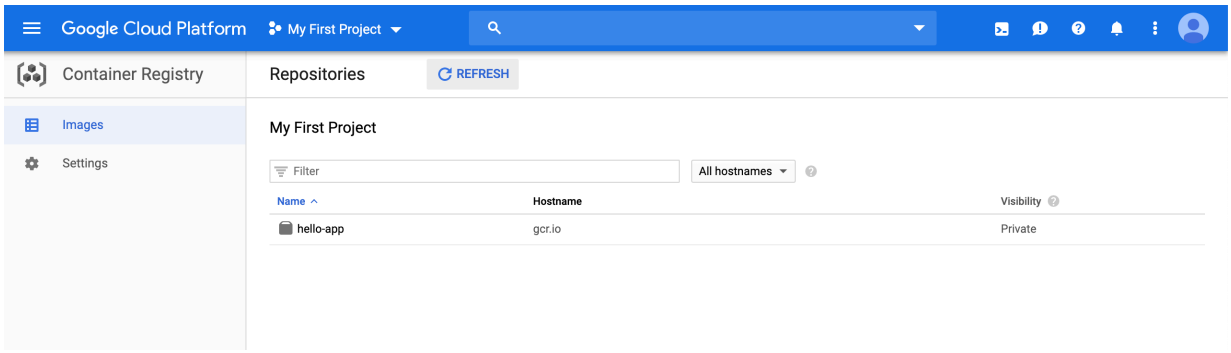
gcloud config set project avian-compiler-238122
gcloud config set compute/zone us-central1-f

git clone https://github.com/GoogleCloudPlatform/kubernetes-engine-samples
cd kubernetes-engine-samples/hello-app

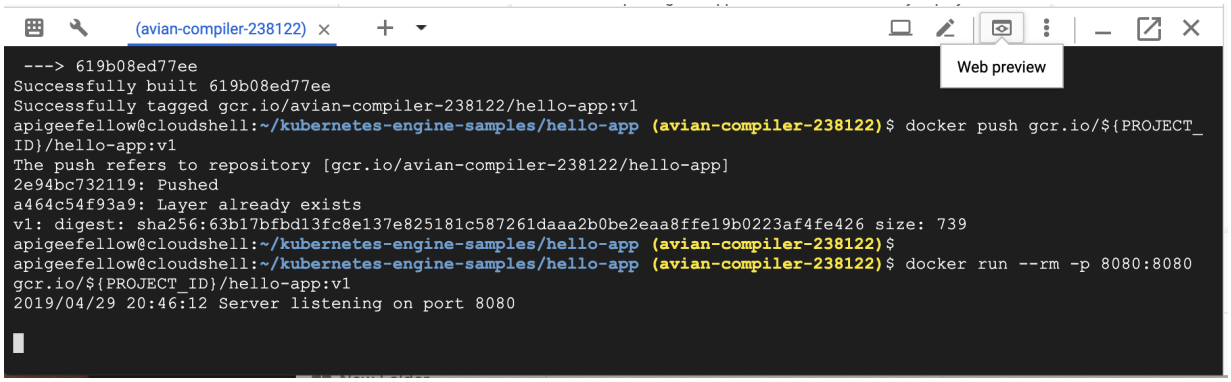
export PROJECT_ID="$(gcloud config get-value project -q)"

Upload the container image

docker build -t gcr.io/${PROJECT_ID}/hello-app:v1 .
gcloud auth configure-docker
docker push gcr.io/${PROJECT_ID}/hello-app:v1



docker run --rm -p 8080:8080 gcr.io/${PROJECT_ID}/hello-app:v1



Run your container locally (optional)

https://8080-dot-7139517-dot-devshell.appspot.com/?authuser=0

curl http://localhost:8080

Create a container cluster

gcloud container clusters create hello-cluster --num-nodes=2

```
Creating cluster hello-cluster in us-central1-f... Cluster is being deployed...⠹
Creating cluster hello-cluster in us-central1-f... Cluster is being health-checked (master is healthy)...done.
Created [https://container.googleapis.com/v1/projects/avian-compiler-238122/zones/us-central1-f/clusters/hello-clust
er].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gcloud/us-cent
ral1-f/hello-cluster?project=avian-compiler-238122
kubeconfig entry generated for hello-cluster.
NAME           LOCATION       MASTER_VERSION  MASTER_IP        MACHINE_TYPE   NODE_VERSION  NUM_NODES  STATUS
hello-cluster  us-central1-f  1.11.8-gke.6    104.154.178.255  n1-standard-1  1.11.8-gke.6  2          RUNNING
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

Kubernetes clusters

CREATE CLUSTER DEPLOY REFRESH DELETE

A Kubernetes cluster is a managed group of VM instances for running containerized applications. Learn more

Filter by label or name

Name	Location	Cluster size	Total cores	Total memory	Notifications	Labels	
hello-cluster	us-central1-f	2	2 vCPUs	7.50 GB			Connect

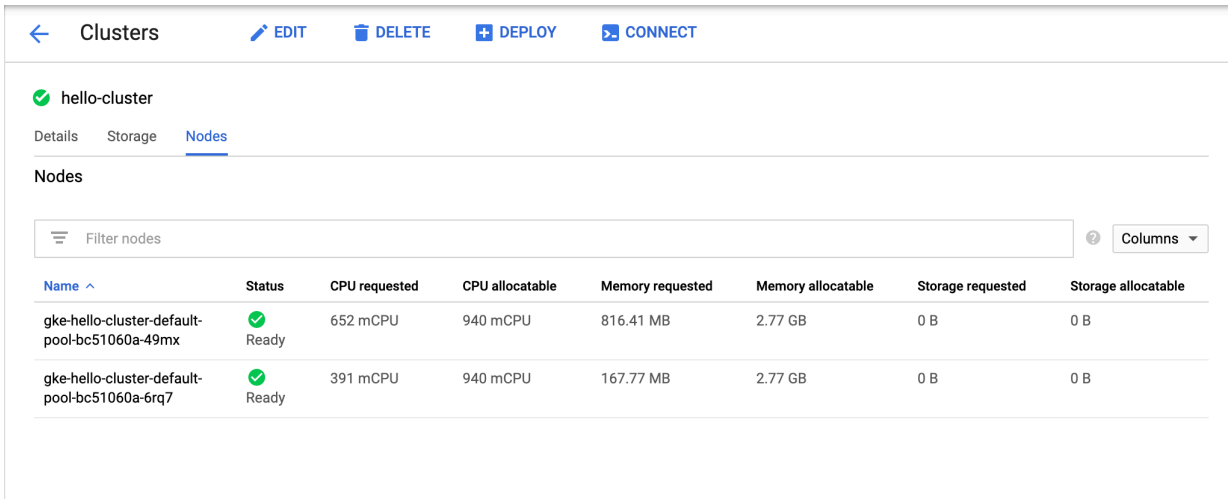
Clusters EDIT DELETE DEPLOY CONNECT

hello-cluster

Details Storage Nodes

Cluster

Master version	1.11.8-gke.6	Upgrade available
Endpoint	104.154.178.255	Show credentials
Client certificate	Enabled	
Binary Authorization	Disabled	
Kubernetes alpha features	Disabled	
Total size	2	
Master zone	us-central1-f	
Node zones	us-central1-f	
Network	default	
Subnet	default	
VPC-native (alias IP)	Disabled	



gcloud compute instances list

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ gcloud compute instances list
NAME                                        ZONE           MACHINE_TYPE   PREEMPTIBLE  INTERNAL_IP   EXTERNAL_IP
  STATUS
gke-hello-cluster-default-pool-bc51060a-49mx  us-central1-f  n1-standard-1               10.128.0.17   35.238.173.95
  RUNNING
gke-hello-cluster-default-pool-bc51060a-6rq7  us-central1-f  n1-standard-1               10.128.0.16   35.222.173.136
  RUNNING
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

gcloud container clusters get-credentials hello-cluster

> **Note:** If you are using an existing Google Kubernetes Engine cluster or if you have created a cluster through Google Cloud Platform Console, you need to run the following command to retrieve cluster credentials and configure **`kubectl`** command-line tool with them:
>
> ```
> gcloud container clusters get-credentials hello-cluster
> ```
>
> If you have already created a cluster with the **`gcloud container clusters create`** command listed above, this step is not necessary.

Deploy your application

kubectl get pods

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get pods
No resources found.
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

kubectl run hello-web --image=gcr.io/${PROJECT_ID}/hello-app:v1 --port 8080

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get pods
NAME                         READY     STATUS    RESTARTS   AGE
hello-web-6854fd845b-q4lps   1/1       Running   0          29s
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

Expose your application to the Internet

kubectl get service

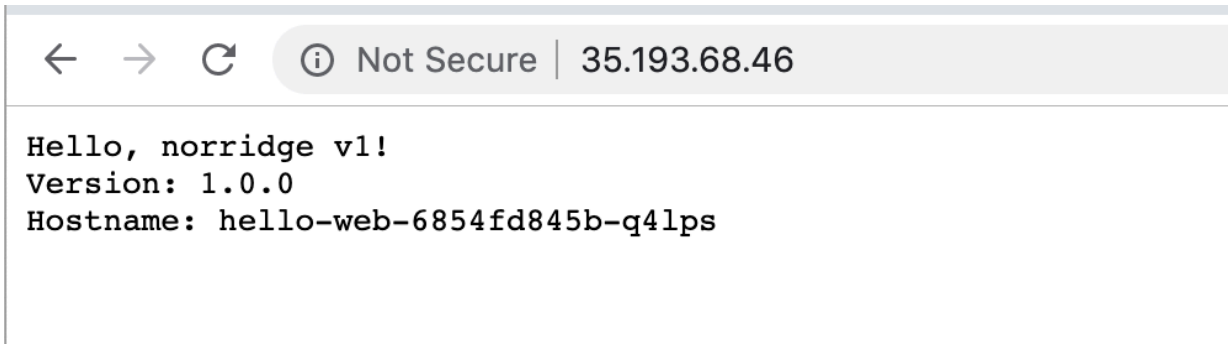
```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get service
NAME         TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.19.240.1   <none>        443/TCP   10m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

kubectl expose deployment hello-web --type=LoadBalancer --port 80 --target-port 8080

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl expose deployment hel
lo-web --type=LoadBalancer --port 80 --target-port 8080
service/hello-web exposed
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get service
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
hello-web    LoadBalancer   10.19.246.157   <pending>     80:31982/TCP   11s
kubernetes   ClusterIP      10.19.240.1     <none>        443/TCP        10m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get service
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
hello-web    LoadBalancer   10.19.246.157   <pending>     80:31982/TCP   22s
kubernetes   ClusterIP      10.19.240.1     <none>        443/TCP        11m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get service
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
hello-web    LoadBalancer   10.19.246.157   <pending>     80:31982/TCP   25s
kubernetes   ClusterIP      10.19.240.1     <none>        443/TCP        11m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get service
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
hello-web    LoadBalancer   10.19.246.157   <pending>     80:31982/TCP   34s
kubernetes   ClusterIP      10.19.240.1     <none>        443/TCP        11m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get service
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)        AGE
hello-web    LoadBalancer   10.19.246.157   35.193.68.46   80:31982/TCP   56s
kubernetes   ClusterIP      10.19.240.1     <none>         443/TCP        11m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

http://35.193.68.46/



Scale up your application

kubectl scale deployment hello-web --replicas=3

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl scale deployment hell
o-web --replicas=3
deployment.extensions/hello-web scaled
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

kubectl get deployment hello-web

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get deployment hello-
web
NAME        DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
hello-web   3         3         3            3           7m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

kubectl get pods

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get pods
NAME                         READY     STATUS    RESTARTS   AGE
hello-web-6854fd845b-q4lps   1/1       Running   0          8m
hello-web-6854fd845b-rcqs8   1/1       Running   0          1m
hello-web-6854fd845b-xd779   1/1       Running   0          1m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get pods
NAME                     READY     STATUS    RESTARTS   AGE
hello-web-6854fd845b-q4lps   1/1       Running   0          8m
hello-web-6854fd845b-rcqs8   1/1       Running   0          1m
hello-web-6854fd845b-xd779   1/1       Running   0          1m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ curl http://35.193.68.46/
Hello, norridge v1!
Version: 1.0.0
Hostname: hello-web-6854fd845b-q4lps
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ curl http://35.193.68.46/
Hello, norridge v1!
Version: 1.0.0
Hostname: hello-web-6854fd845b-q4lps
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ curl http://35.193.68.46/
Hello, norridge v1!
Version: 1.0.0
Hostname: hello-web-6854fd845b-rcqs8
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ curl http://35.193.68.46/
Hello, norridge v1!
Version: 1.0.0
Hostname: hello-web-6854fd845b-xd779
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ curl http://35.193.68.46/
Hello, norridge v1!
Version: 1.0.0
Hostname: hello-web-6854fd845b-xd779
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ curl http://35.193.68.46/
Hello, norridge v1!
Version: 1.0.0
Hostname: hello-web-6854fd845b-q4lps
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

Deploy a new version of your app

apigeefellow@cloudshell:**~/kubernetes-engine-samples/hello-app (avian-compiler-238122)**$ vi main.go

```
// [START all]
package main

import (
        "fmt"
        "log"
        "net/http"
        "os"
)

func main() {
        // use PORT environment variable, or default to 8080
        port := "8080"
        if fromEnv := os.Getenv("PORT"); fromEnv != "" {
                port = fromEnv
        }

        // register hello function to handle all requests
        server := http.NewServeMux()
        server.HandleFunc("/", hello)

        // start the web server on port and accept requests
        log.Printf("Server listening on port %s", port)
        err := http.ListenAndServe(":"+port, server)
        log.Fatal(err)
}

// hello responds to the request with a plain-text "Hello, world" message.
func hello(w http.ResponseWriter, r *http.Request) {
        log.Printf("Serving request: %s", r.URL.Path)
        host, _ := os.Hostname()
        fmt.Fprintf(w, "Hello, hoffman v2!\n")
        fmt.Fprintf(w, "Version: 1.0.0\n")
        fmt.Fprintf(w, "Hostname: %s\n", host)
}
// [END all]
"main.go" 52L, 1435C
```

docker build -t gcr.io/${PROJECT_ID}/hello-app:v2 .

docker push gcr.io/${PROJECT_ID}/hello-app:v2

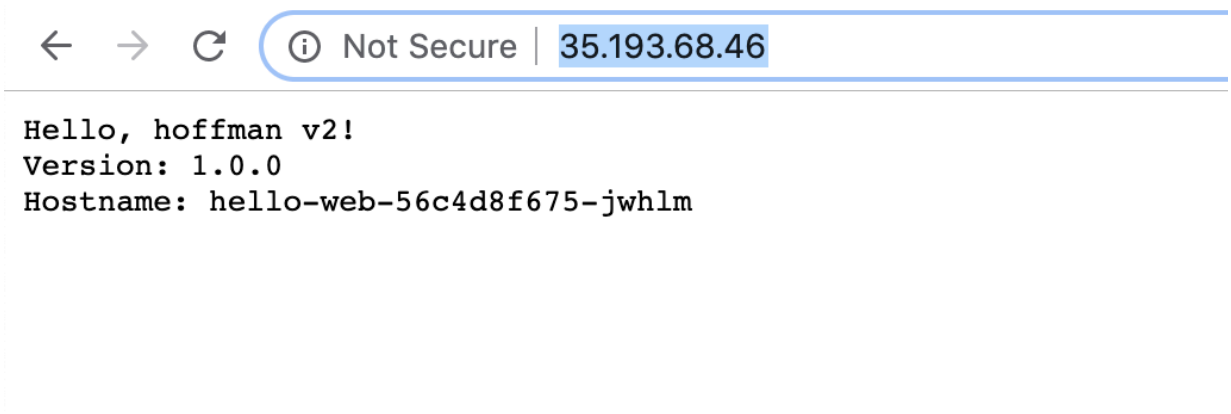
```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ docker build -t gcr.io/${PROJECT_ID}/hello-app:v2 .

Sending build context to Docker daemon  9.728kB
Step 1/7 : FROM golang:1.8-alpine
 ---> 4cb86d3661bf
Step 2/7 : ADD . /go/src/hello-app
 ---> 68aa0a3ccca7
Step 3/7 : RUN go install hello-app
 ---> Running in 62bb1ecf34b5
Removing intermediate container 62bb1ecf34b5
 ---> 0dbc7b5e6637
Step 4/7 : FROM alpine:latest
 ---> cdf98d1859c1
Step 5/7 : COPY --from=0 /go/bin/hello-app .
 ---> 8e1b4e1a1244
Step 6/7 : ENV PORT 8080
 ---> Running in de7f80e5ae56
Removing intermediate container de7f80e5ae56
 ---> 25e92a982982
Step 7/7 : CMD ["./hello-app"]
 ---> Running in 7f2d3bf2d1c4
Removing intermediate container 7f2d3bf2d1c4
 ---> d6d551b4cfc3
Successfully built d6d551b4cfc3
Successfully tagged gcr.io/avian-compiler-238122/hello-app:v2
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ docker push gcr.io/${PROJECT_ID}/hello-app:v2
The push refers to repository [gcr.io/avian-compiler-238122/hello-app]
d7ce40555324: Pushed
a464c54f93a9: Layer already exists
v2: digest: sha256:f1ec51a539e2cb814deda5b6ea5fc115ad9bb71d3a2261a8ae663fe046e68a2a size: 739
```

kubectl get deployment hello-web
kubectl get pods

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl set image deployment/hello-web hello-web=gcr.io/${PROJECT_ID}/hello
-app:v2
deployment.extensions/hello-web image updated
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get deployment hello-web
NAME        DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
hello-web   3         3         3            3           19m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get pods
NAME                         READY     STATUS    RESTARTS   AGE
hello-web-56c4d8f675-jwhlm   1/1       Running   0          24s
hello-web-56c4d8f675-wqr27   1/1       Running   0          29s
hello-web-56c4d8f675-xhbkz   1/1       Running   0          26s
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

http://35.193.68.46/



```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get pods
NAME                         READY     STATUS    RESTARTS   AGE
hello-web-56c4d8f675-jwhlm   1/1       Running   0          2m
hello-web-56c4d8f675-wqr27   1/1       Running   0          2m
hello-web-56c4d8f675-xhbkz   1/1       Running   0          2m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ curl http://35.193.68.46/
Hello, hoffman v2!
Version: 1.0.0
Hostname: hello-web-56c4d8f675-wqr27
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ curl http://35.193.68.46/
Hello, hoffman v2!
Version: 1.0.0
Hostname: hello-web-56c4d8f675-xhbkz
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ curl http://35.193.68.46/
Hello, hoffman v2!
Version: 1.0.0
Hostname: hello-web-56c4d8f675-jwhlm
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ curl http://35.193.68.46/
Hello, hoffman v2!
Version: 1.0.0
Hostname: hello-web-56c4d8f675-jwhlm
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ curl http://35.193.68.46/
Hello, hoffman v2!
Version: 1.0.0
Hostname: hello-web-56c4d8f675-wqr27
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

Setting up HTTP Load Balancing with Ingress (This works as Load Balancer takes 10-15 minutes to propagate)

https://cloud.google.com/kubernetes-engine/docs/tutorials/http-balancer

kubectl get pod

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get pod
NAME                         READY     STATUS    RESTARTS   AGE
hello-web-56c4d8f675-jwhlm   1/1       Running   0          11m
hello-web-56c4d8f675-wqr27   1/1       Running   0          11m
hello-web-56c4d8f675-xhbkz   1/1       Running   0          11m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

Deploy a web application

kubectl run web1 --image=gcr.io/${PROJECT_ID}/hello-app:v1 --port 8080
kubectl run web2 --image=gcr.io/${PROJECT_ID}/hello-app:v2 --port 8080

kubectl get pod

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get pod
NAME                         READY     STATUS    RESTARTS   AGE
hello-web-56c4d8f675-jwhlm   1/1       Running   0          12m
hello-web-56c4d8f675-wqr27   1/1       Running   0          12m
hello-web-56c4d8f675-xhbkz   1/1       Running   0          12m
web1-5ffc78664b-hkphs        1/1       Running   0          18s
web2-976d7685c-jscst         1/1       Running   0          5s
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

Expose your Deployment as a Service internally

kubectl expose deployment web1 --target-port=8080 --type=NodePort
kubectl expose deployment web2 —target-port=8080 --type=NodePort

kubectl get service web1

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get service web
No resources found.
Error from server (NotFound): services "web" not found
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get service web1
No resources found.
Error from server (NotFound): services "web1" not found
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl expose deployment web1 --target-port=8080 --type=NodePort
service/web1 exposed
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl expose deployment web2 —target-port=8080 --type=NodePort
service/web2 exposed
Error from server (NotFound): deployments.extensions "—target-port=8080" not found
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get service web
No resources found.
Error from server (NotFound): services "web" not found
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get service web1
NAME      TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
web1      NodePort   10.19.246.86   <none>        8080:32332/TCP   1m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

kubectl get service web1

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get service web1
NAME      TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
web1      NodePort   10.19.246.86   <none>        8080:32332/TCP   2m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

vi basic-ingress.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: basic-ingress
spec:
  backend:
    serviceName: web1
    servicePort: 8080
~
~
```

kubectl apply -f basic-ingress.yaml

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl apply -f basic-ingress.yaml
ingress.extensions/basic-ingress created
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

kubectl get ingress basic-ingress

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get ingress basic-ingress
NAME            HOSTS     ADDRESS   PORTS     AGE
basic-ingress   *                   80        34s
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get ingress basic-ingress
NAME            HOSTS     ADDRESS         PORTS     AGE
basic-ingress   *         34.95.110.35    80        1m
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

Setting up HTTP Load Balancing with Ingress - Part 2

Makes use of yaml

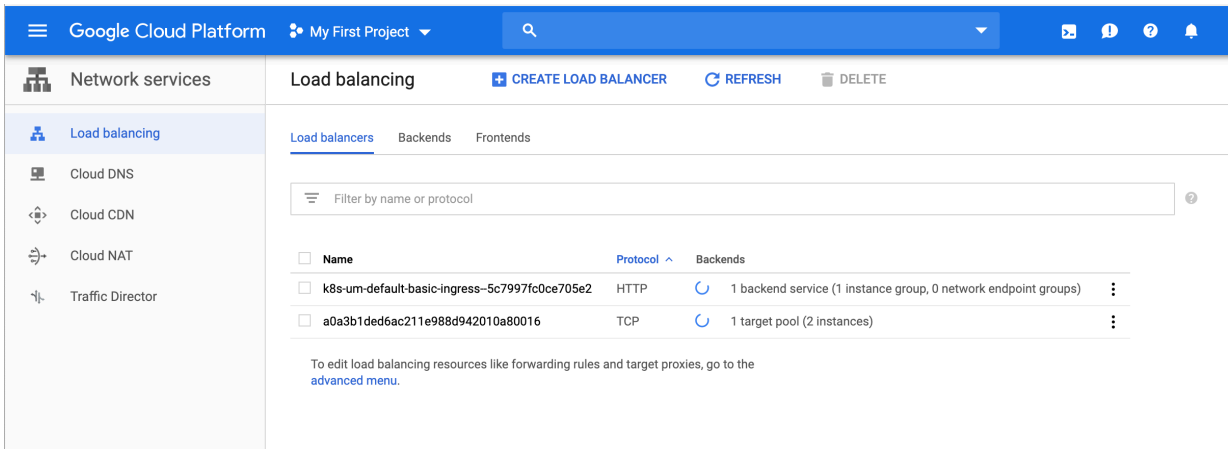
gcloud container clusters create loadbalancedcluster

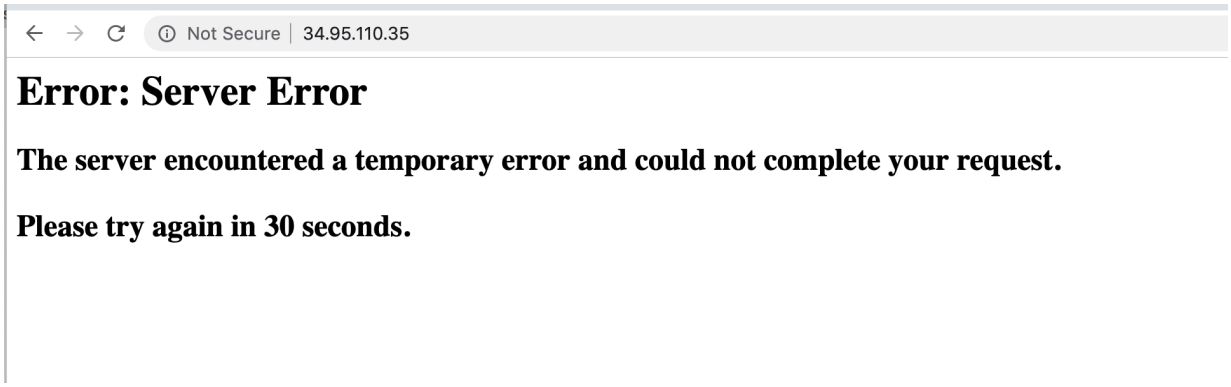
https://cloud.google.com/kubernetes-engine/docs/tutorials/http-balancer

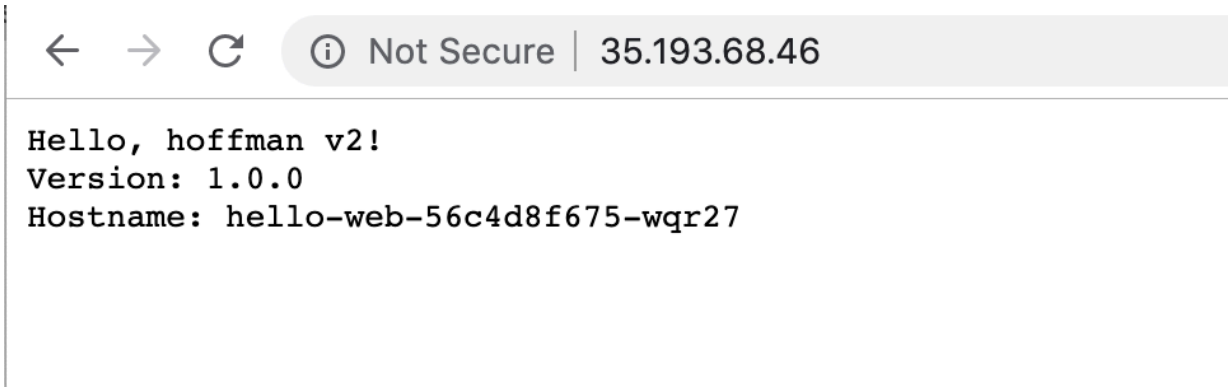
```
kubeconfig entry generated for loadbalancedcluster.
NAME                LOCATION       MASTER_VERSION  MASTER_IP       MACHINE_TYPE   NODE_VERSION  NUM_NODES  STATUS
loadbalancedcluster  us-central1-f  1.11.8-gke.6    104.154.73.174  n1-standard-1  1.11.8-gke.6  3          RUNNING
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl run web --image=gcr.io/google-samples/hello-app:1.0 --port=8080
deployment.apps/web created
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl expose deployment web --target-port=8080 --type=NodePort
service/web exposed
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get service web
NAME      TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
web       NodePort   10.43.250.32   <none>        8080:31650/TCP   10s
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ vi basic-ingress.yaml
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl apply -f basic-ingress.yaml
ingress.extensions/basic-ingress created
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get ingress basic-ingress
NAME            HOSTS     ADDRESS   PORTS     AGE
basic-ingress   *                   80        9s
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

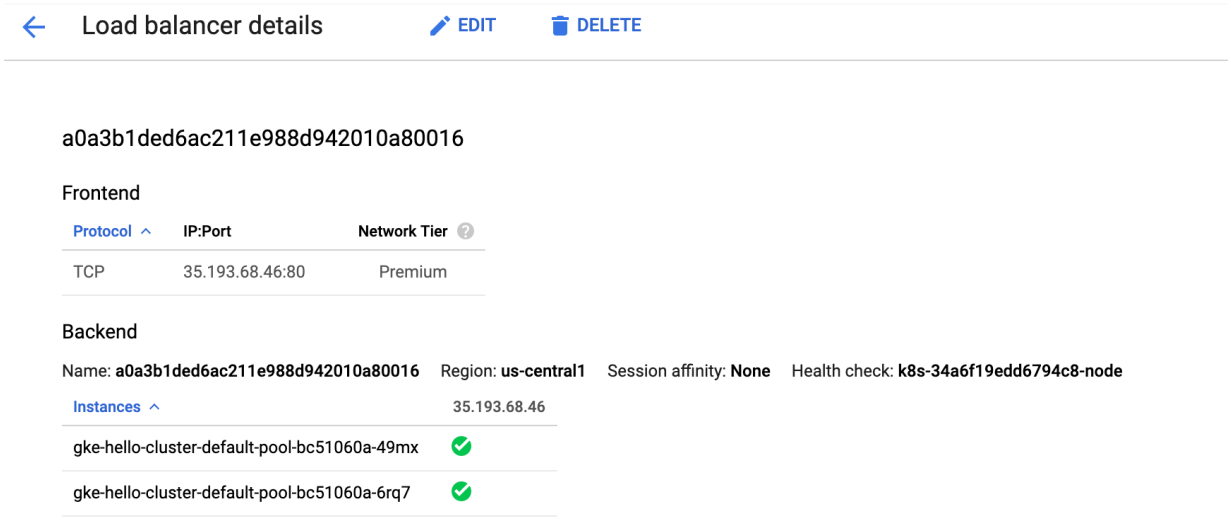
```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get ingress basic-ingress
NAME            HOSTS     ADDRESS         PORTS     AGE
basic-ingress   *         34.95.110.35    80        46s
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

★ **Note:** It may take a few minutes for GKE to allocate an external IP address and set up forwarding rules until the load balancer is ready to serve your application. In the meanwhile, you may get errors such as HTTP 404 or HTTP 500 until the load balancer configuration is propagated across the globe.

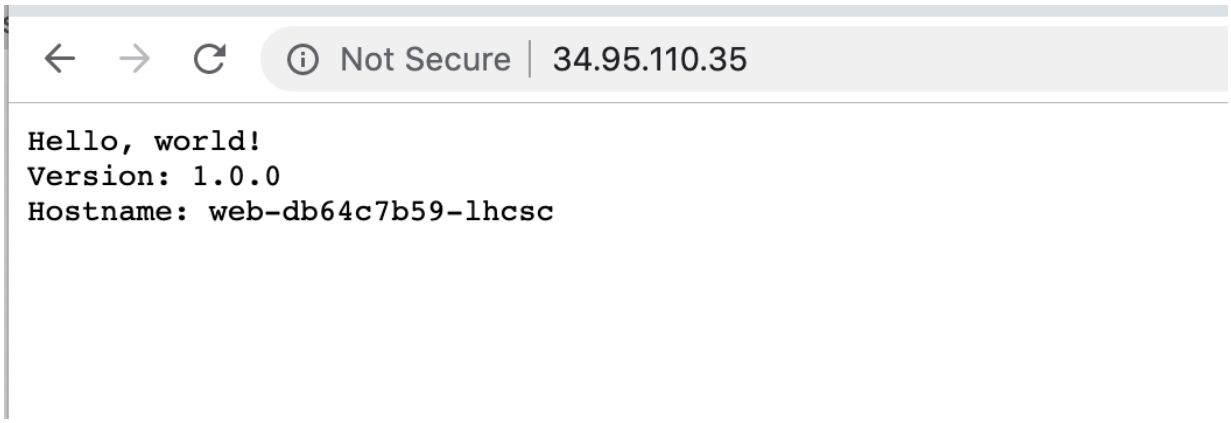

Google Cloud Platform
My First Project
Network services
Load balancing
Cloud DNS
Cloud CDN
Cloud NAT
Traffic Director
Load balancing
CREATE LOAD BALANCER
REFRESH
DELETE
Load balancers
Backends
Frontends
Filter by name or protocol
Name
Protocol
Backends
k8s-um-default-basic-ingress--5c7997fc0ce705e2
HTTP
1 backend service (1 instance group, 0 network endpoint groups)
a0a3b1ded6ac211e988d942010a80016
TCP
1 target pool (2 instances)
To edit load balancing resources like forwarding rules and target proxies, go to the advanced menu.


Not Secure
34.95.110.35
Error: Server Error
The server encountered a temporary error and could not complete your request.
Please try again in 30 seconds.


Not Secure
35.193.68.46
Hello, hoffman v2!
Version: 1.0.0
Hostname: hello-web-56c4d8f675-wqr27



Ingress http://34.95.110.35/



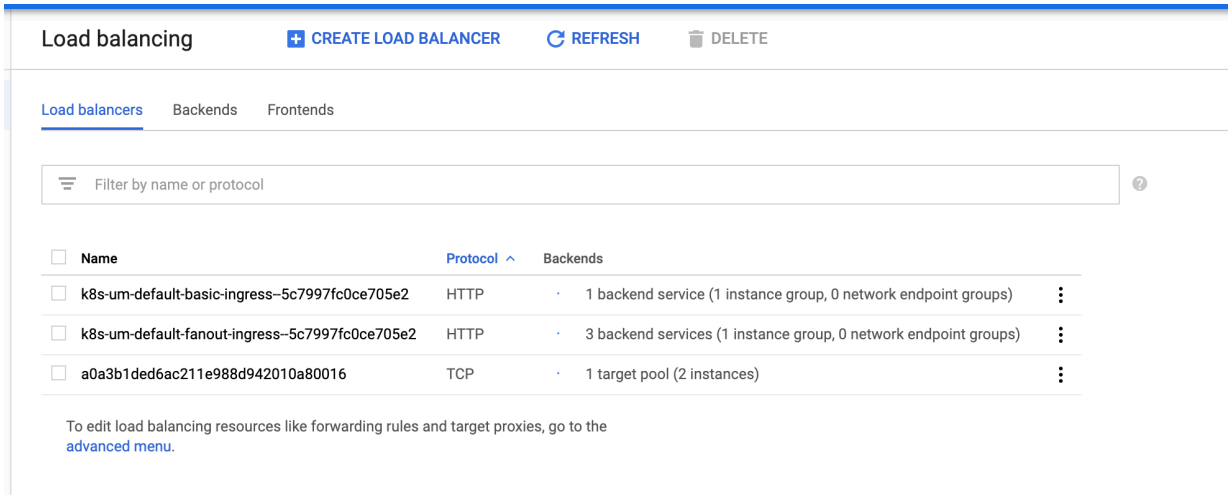
kubectl run web2 --image=gcr.io/google-samples/hello-app:2.0 --port=8080

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: fanout-ingress
spec:
  rules:
  - http:
      paths:
      - path: /*
        backend:
          serviceName: web
          servicePort: 8080
      - path: /v2/*
        backend:
          serviceName: web2
          servicePort: 8080
~
```

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl run web2 --image=gcr.io/google-samples/hello-app:2.0 --port=8080
deployment.apps/web2 created
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl expose deployment web2 --target-port=8080 --type=NodePort
service/web2 exposed
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ vi fanout-ingress.yaml
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ vi fanout-ingress.yaml
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl create -f fanout-ingress.yaml
ingress.extensions/fanout-ingress created
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```

kubectl get ingress fanout-ingress

```
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$ kubectl get ingress fanout-ingress
NAME            HOSTS     ADDRESS         PORTS     AGE
fanout-ingress  *         34.95.99.222    80        47s
apigeefellow@cloudshell:~/kubernetes-engine-samples/hello-app (avian-compiler-238122)$
```



k8s-um-default-fanout-ingress--5c7997fc0ce705e2

Details Monitoring Caching

Frontend

Protocol	IP:Port	Network Tier
HTTP	34.95.99.222:80	Premium

Host and path rules

Hosts	Paths	Backend
All unmatched (default)	All unmatched (default)	k8s-be-31148--5c7997fc0ce705e2
*	/*	k8s-be-31148--5c7997fc0ce705e2
*	/*	k8s-be-31650--5c7997fc0ce705e2
*	/v2/*	k8s-be-31345--5c7997fc0ce705e2

Backend

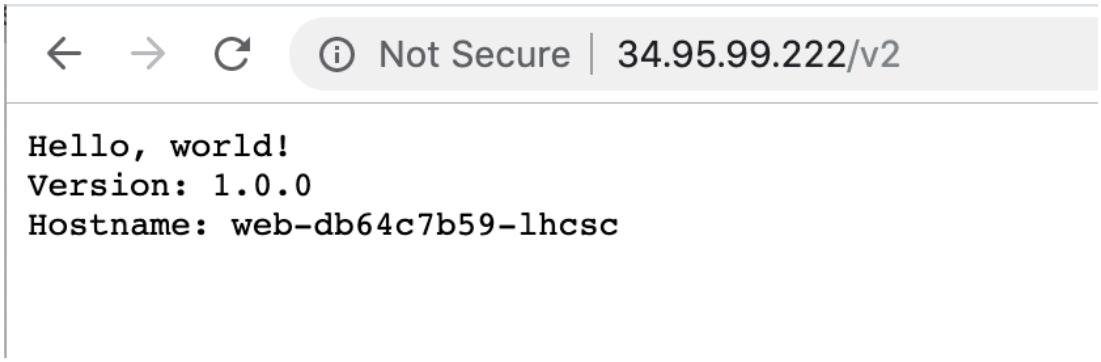
Backend services

1. k8s-be-31148--5c7997fc0ce705e2

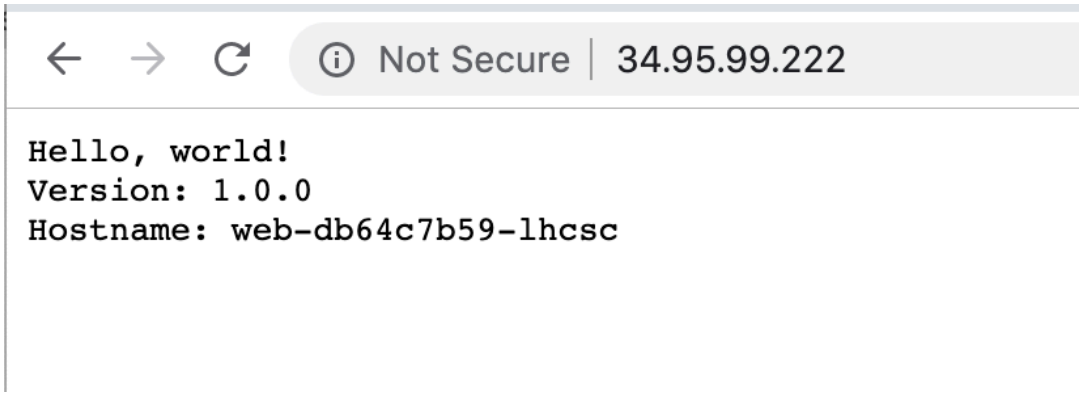
Endpoint protocol: **HTTP** Named port: **port31148** Timeout: **30 seconds** Cloud CDN: **disabled** Health check: k8s-be-31148--5c7997fc0ce705e2

V2

http://34.95.99.222/v2



http://34.95.99.222/


Not Secure
34.95.99.222
Hello, world!
Version: 1.0.0
Hostname: web-db64c7b59-lhcsc