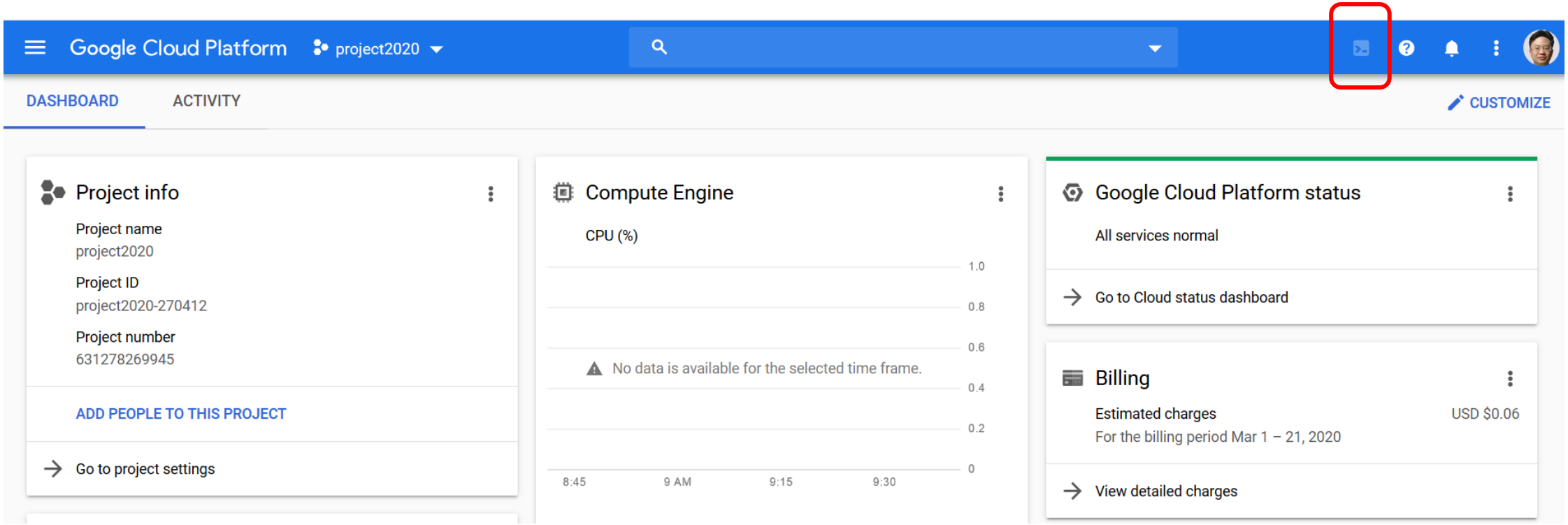
쿠버네티스를 활용한 멀티클라우드 도입과 운영전략
- AWS, Azure, GCP 비교와 실습

2020년 3월
아인인텔리전스
권재원, Ph.D.

Lab: Introduction to Kubernetes

- Using GCP CloudShell

실습 : CloudShell 시작하기



실습 : Hello Node Kubernetes

- **Setup and Requirements (GCP CloudShell)**

```
$  gcloud auth list
$  gcloud config set project {project_name}
$  gcloud config list project
$ gcloud config set compute/zone asia-northeast1-c
$ export PROJECT_ID={project_name}
$ echo $PROJECT_ID
```

** Google Cloud gcloud Overview - https://cloud.google.com/sdk/gcloud*

- **Cluster Creation & Get-Credentials**

```
$ gcloud container clusters create [CLUSTER-NAME]
$ gcloud container clusters get-credentials [CLUSTER-NAME]
```

** https://cloud.google.com/kubernetes-engine/docs/how-to/cluster-access-for-kubectl*

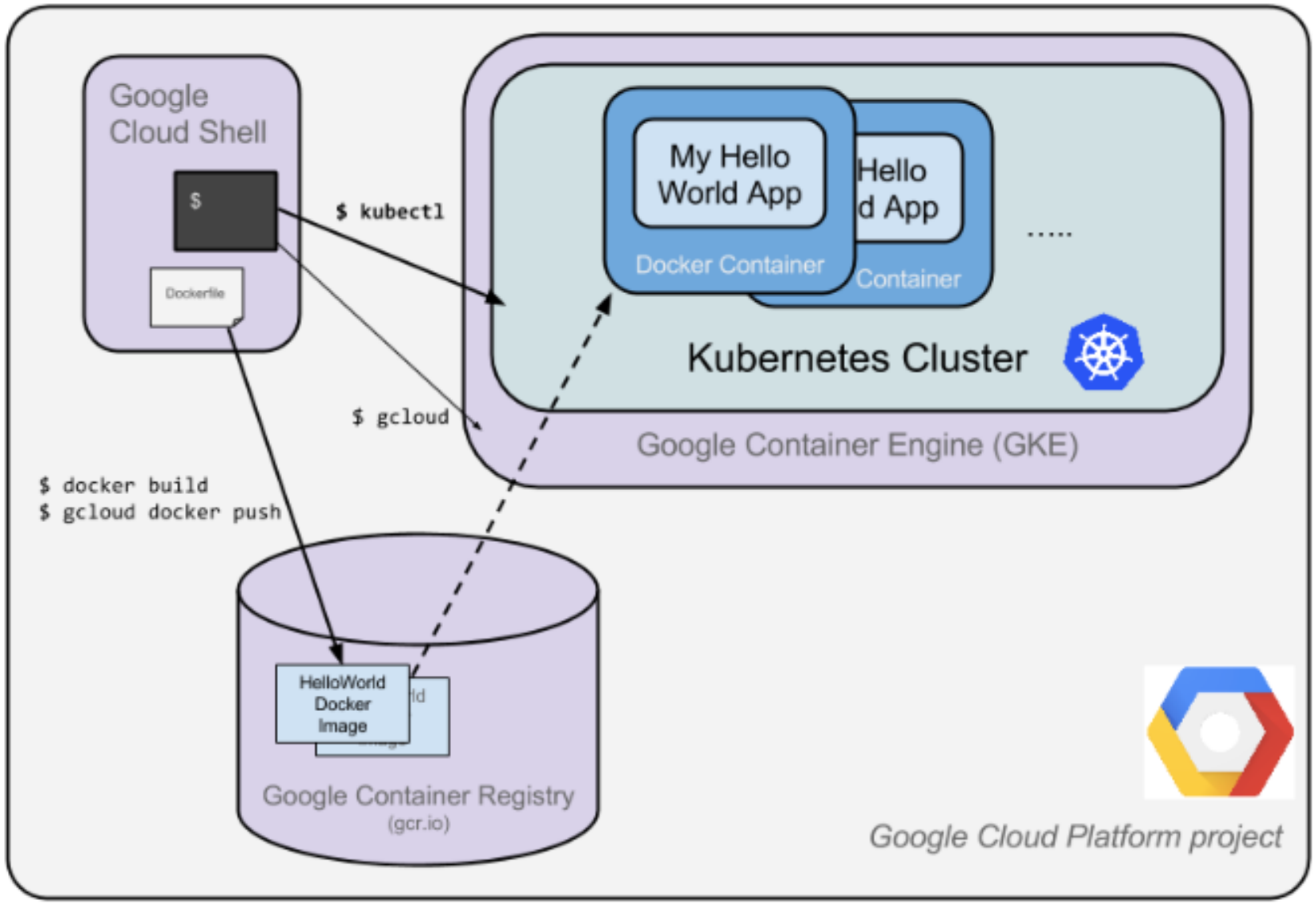
Kubernetes Lab

- **Hello Node Kubernetes**
- **Container Cluster / K8S Pod / Service**

- Create a Node.js server.
- Create a Docker container image.
- Create a container cluster.
- Create a Kubernetes pod.
- Scale up your services.

QwikLabs - https://www.qwiklabs.com/focuses/564?catalog_rank=%7B%22rank%22%3A1%2C%22num_filters%22%3A0%2C%22has_search%22%3Atrue%7D&parent=catalog&search_id=4900160

실습 : Hello Node Kubernetes



Container Image Push to Registry

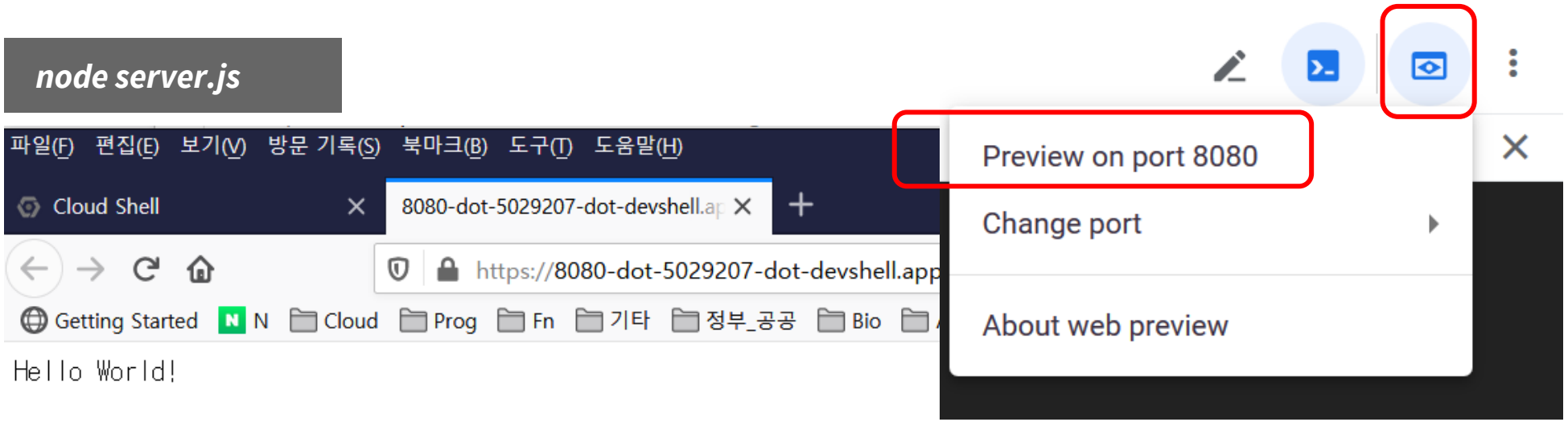
실습 : Hello Node Kubernetes

- Create your Node.js application

nano server.js

```
var http = require('http');
var handleRequest = function(request, response) {
 response.writeHead(200);
 response.end("Hello World!");
}
var www = http.createServer(handleRequest);
www.listen(8080);
```

node server.js



실습 : Hello Node Kubernetes

- Create a Docker container image

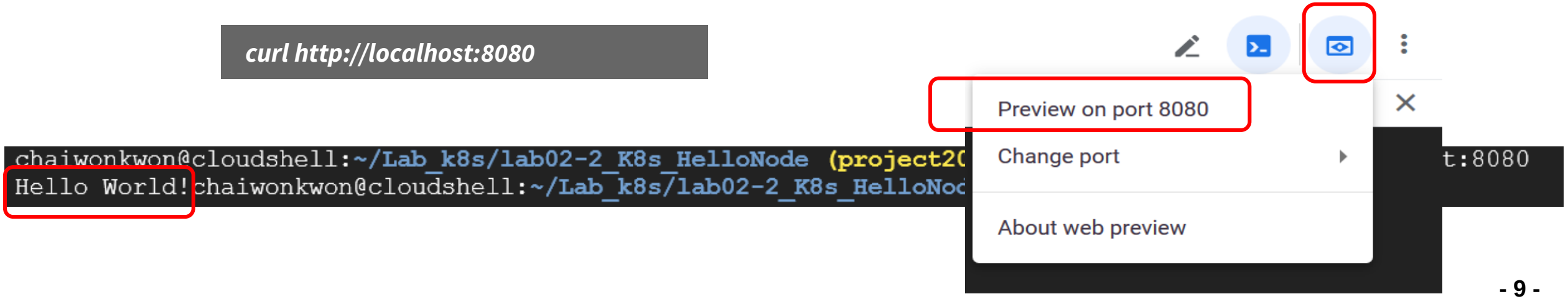
```
nano Dockerfile
```

```
FROM node:6.9.2
EXPOSE 8080
COPY server.js .
CMD node server.js
```

```
docker build -t gcr.io/$PROJECT_ID/hello-node:v1 .
docker run -d -p 8080:8080 gcr.io/$PROJECT_ID/hello-node:v1
```

```
325301e6b2bffd1d0049c621866831316d653c0b25a496d04ce0ec6854cb7998
```

```
curl http://localhost:8080
```



실습 : Hello Node Kubernetes

docker ps

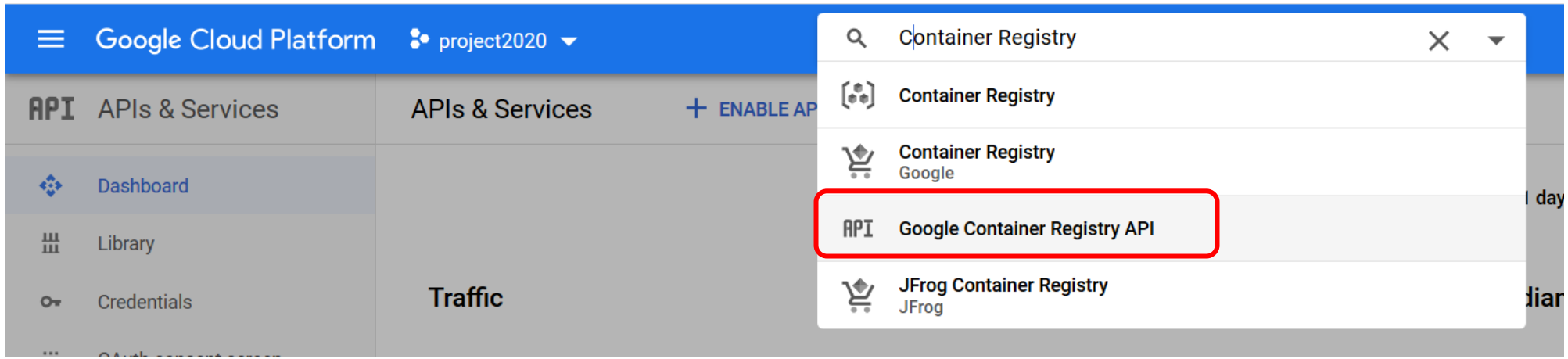
```
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ docker ps
CONTAINER ID        IMAGE                                          COMMAND                  CREATED
     STATUS              PORTS                    NAMES
55edf3e13a53        gcr.io/project2020-270412/hello-node:v1   "/bin/sh -c 'node se…"   3 minutes ago
     Up 3 minutes        0.0.0.0:8080->8080/tcp   condescending_zhukovsky
```

docker stop CONTAINER_ID

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ docker stop 55edf3e13a53
55edf3e13a53
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                  PO
RTS                 NAMES
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ docker ps -a
CONTAINER ID        IMAGE                                          COMMAND                  CREATED
     STATUS                      PORTS                    NAMES
55edf3e13a53        gcr.io/project2020-270412/hello-node:v1   "/bin/sh -c 'node se…"   8 minutes ago
     Exited (137) 8 seconds ago                           condescending_zhukovsky
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$
```

실습 : Hello Node Kubernetes

- Container Registry API Activation



실습 : Hello Node Kubernetes

gcloud auth configure-docker

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ gcloud auth configure-doc
ker
WARNING: Your config file at [/home/chaiwonkwon/.docker/config.json] contains these credential helper
entries:

{
  "credHelpers": {
    "us.gcr.io": "gcloud",
    "staging-k8s.gcr.io": "gcloud",
    "asia.gcr.io": "gcloud",
    "gcr.io": "gcloud",
    "marketplace.gcr.io": "gcloud",
    "eu.gcr.io": "gcloud"
  }
}
Adding credentials for all GCR repositories.
WARNING: A long list of credential helpers may cause delays running 'docker build'. We recommend passi
ng the registry name to configure only the registry you are using.
gcloud credential helpers already registered correctly.
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$
```

실습 : Hello Node Kubernetes

export PROJECT_ID=project2020-270412

echo $PROJECT_ID

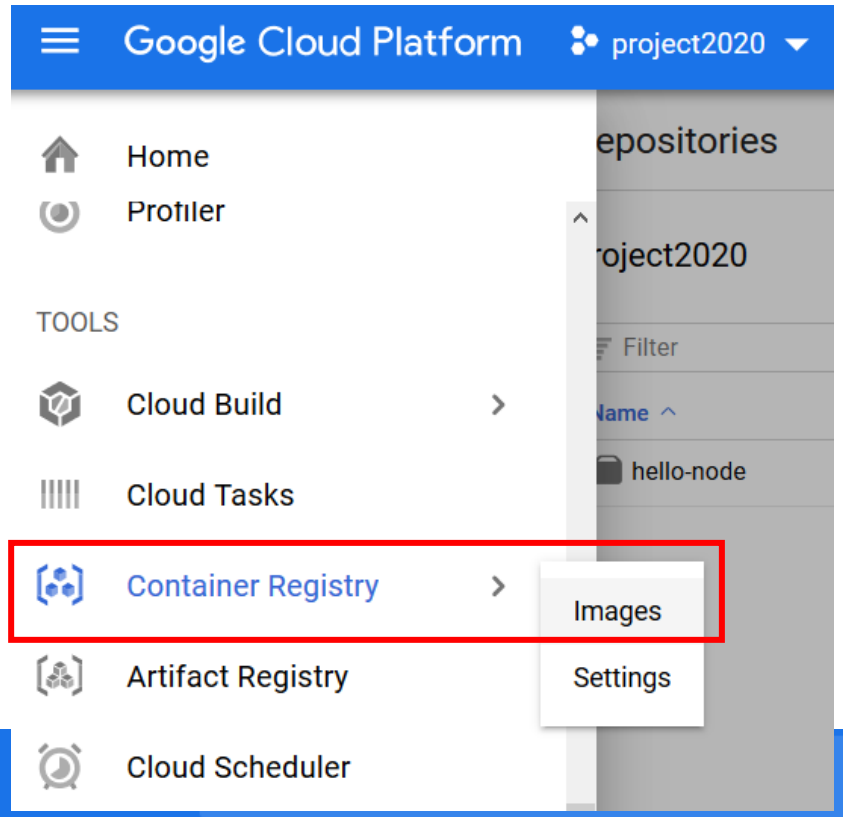
```
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ export PROJECT_ID=project2020-270412
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ echo $PROJECT_ID
project2020-270412
```

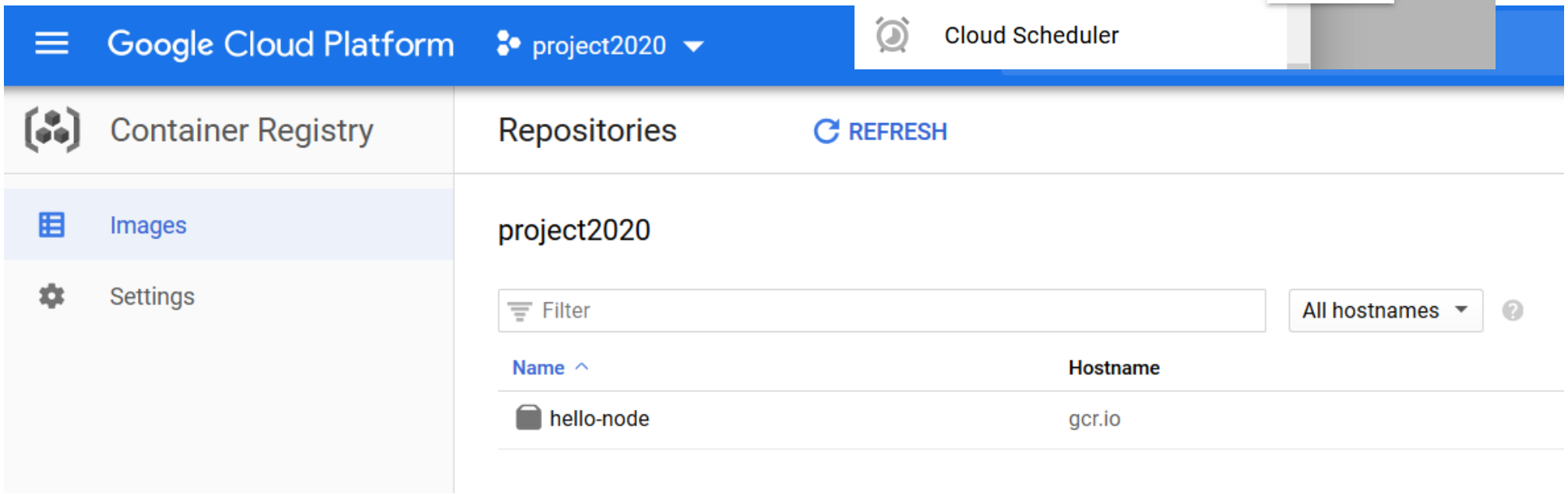
docker push gcr.io/$PROJECT_ID/hello-node:v1

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ docker push gcr.io/$PROJECT_ID/hello-node:v1
The push refers to repository [gcr.io/project2020-270412/hello-node]
6d5c86f6b65c: Pushed
381c97ba7dc3: Pushed
604c78617f34: Pushed
fa18e5ffd316: Pushed
0a5e2b2ddeaa: Pushed
53c779688d06: Pushed
60a0858edcd5: Pushed
b6ca02dfe5e6: Pushed
v1: digest: sha256:f742434ae4bf26777b097a804c0dd12ce54aae27350efef4625703c15e1c32fa size: 2002
```

실습 : Hello Node Kubernetes

- Container Registry Images



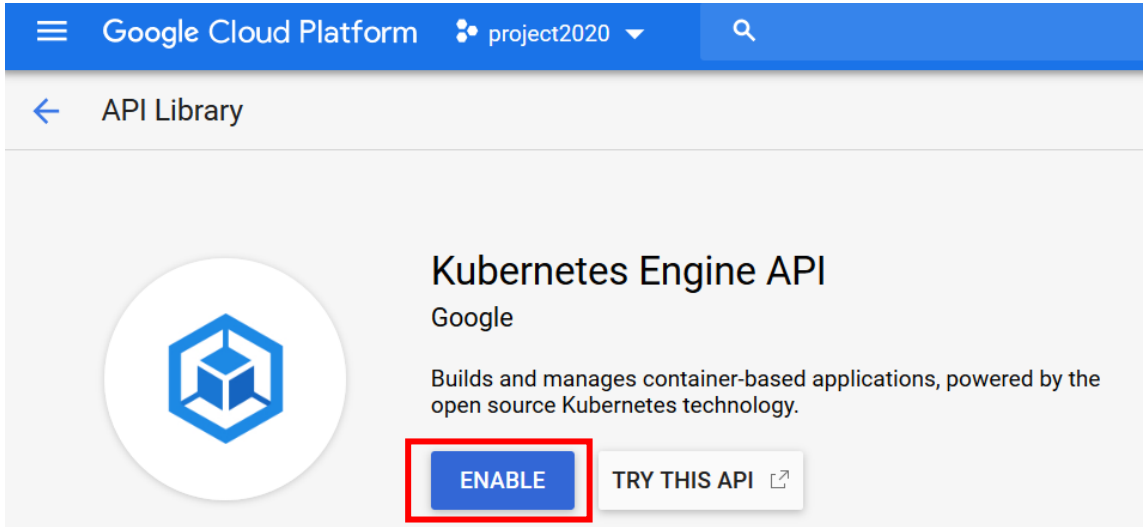


Kubernetes : Container Cluster & Pods

실습 : Hello Node Kubernetes

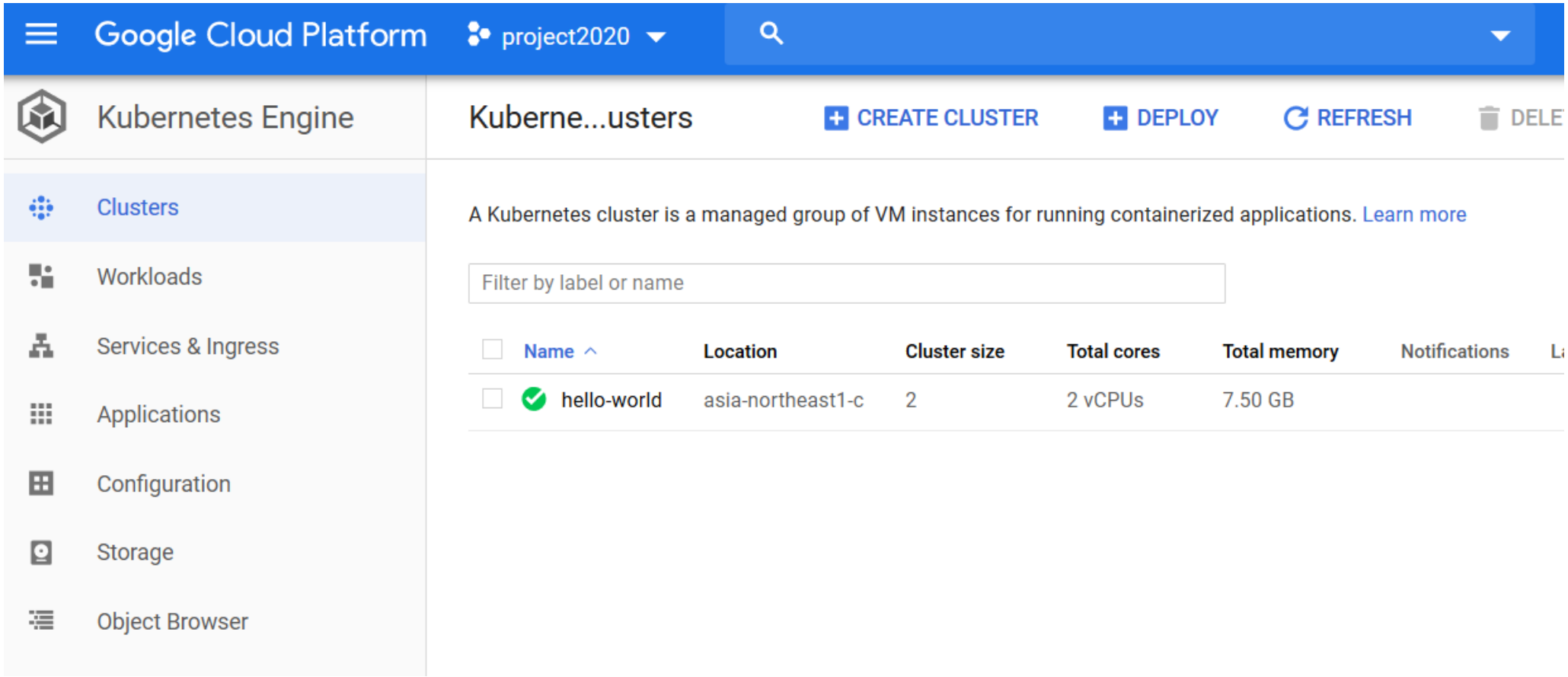
- Create your cluster

```
gcloud config set project $PROJECT_ID
gcloud container clusters create hello-world \
        --num-nodes 2 \
        --machine-type n1-standard-1 \
        --zone asia-northeast1-c
```




```
Creating cluster hello-world in asia-northeast1-c... Cluster is being health-checked (master is healthy)...done.
Created [https://container.googleapis.com/v1/projects/project2020-270412/zones/asia-northeast1-c/clusters/hello-world].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gcloud/asia-northeast1-c/he
llo-world?project=project2020-270412
kubeconfig entry generated for hello-world.
NAME         LOCATION           MASTER_VERSION  MASTER_IP      MACHINE_TYPE   NODE_VERSION    NUM_NODES  STATUS
hello-world  asia-northeast1-c  1.14.10-gke.24  35.243.81.252  n1-standard-1  1.14.10-gke.24  2          RUNNING
```

실습 : Hello Node Kubernetes



실습 : Hello Node Kubernetes

- Create your cluster

```
kubectl run hello-node \
  --image=gcr.io/$PROJECT_ID/hello-node:v1 \
  --port=8080
```

```
deployment.apps/hello-node created
```

```
kubectl get deployments
kubectl get pods
```

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
hello-node   1/1     1            1           106s
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
hello-node-566cc58495-8mzgh   1/1     Running   0          119s
```

실습 : Hello Node Kubernetes

kubectl config view

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://35.243.81.252
  name: gke_project2020-270412_asia-northeast1-c_hello-world
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://35.239.254.253
  name: gke_test20191004_us-central1-c_mnist-deployment
contexts:
- context:
    cluster: gke_project2020-270412_asia-northeast1-c_hello-world
    user: gke_project2020-270412_asia-northeast1-c_hello-world
  name: gke_project2020-270412_asia-northeast1-c_hello-world
- context:
    cluster: gke_test20191004_us-central1-c_mnist-deployment
    namespace: kubeflow
    user: gke_test20191004_us-central1-c_mnist-deployment
  name: gke_test20191004_us-central1-c_mnist-deployment
current-context: gke_project2020-270412_asia-northeast1-c_hello-world
kind: Config
preferences: {}
users:
- name: gke_project2020-270412_asia-northeast1-c_hello-world
  user:
    auth-provider:
      config:
        access-token: ya29.a0Adw1xeWbS1pmaKeMJ85t1OtF73INKK0T61RTBUmVW34MQAU7IH31EHn8UzdGj-J3vPspITbQWfBNytjMzvKVHJTNkB1XzNZizcTbCLJ3mzlpQpq07n27qMK9exA5ZnDRkMiLd9TmmYAgkI0YT89cSD
8tLxCAHk8xZDqkMEejeKbwAMfD2JEkjw2z9YBJnaAeqlVkhiHIAz7qtDBTg2GP_424i_OPhqlbMBwuxul-L-goP3iLO4jAtiuCSQbm1mEx9cwhOXcTcf8jSA
        cmd-args: config config-helper --format=json
        cmd-path: /google/google-cloud-sdk/bin/gcloud
        expiry: "2020-03-22T03:36:08Z"
        expiry-key: '{.credential.token_expiry}'
        token-key: '{.credential.access_token}'
      name: gcp
- name: gke_test20191004_us-central1-c_mnist-deployment
  user:
    auth-provider:
      config:
        access-token: ya29.Iq4BogeDRHeDfm5npdX4n0x_GgPFBrFPRf_YLwyuAnUXeFqG5W8lMHLglFHvMhNYhHTG8kO2Ht-ej3Gj6PrmXK7YKFur3QHD6rzk2UscQMuAf0Mcy_vNgeAGJ9Xcr8Z6RvGAPaMsRDt8lsRm3hhObpiA
wiBoT2swa8Oo107X2JtmHkelzJPgQD9aHc853YfFHBt7UZ_7VpNRLNJ_nhO1-rNunoxTcHwqNahPJKQkAao7
        cmd-args: config config-helper --format=json
        cmd-path: /google/google-cloud-sdk/bin/gcloud
        expiry: "2019-10-21T03:16:13Z"
        expiry-key: '{.credential.token_expiry}'
        token-key: '{.credential.access_token}'
      name: gcp
```

실습 : Hello Node Kubernetes

kubectl cluster-info

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ kubectl cluster-info
Kubernetes master is running at https://35.243.81.252
GLBCDefaultBackend is running at https://35.243.81.252/api/v1/namespaces/kube-system/services/default-http-backend:http/proxy
Heapster is running at https://35.243.81.252/api/v1/namespaces/kube-system/services/heapster/proxy
KubeDNS is running at https://35.243.81.252/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Metrics-server is running at https://35.243.81.252/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy
```

kubectl get nodes

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ kubectl get nodes
NAME                                          STATUS   ROLES    AGE   VERSION
gke-hello-world-default-pool-c41c2400-1qsp    Ready    <none>   16m   v1.14.10-gke.24
gke-hello-world-default-pool-c41c2400-651m    Ready    <none>   16m   v1.14.10-gke.24
```

kubectl get events

kubectl logs <pod-name>

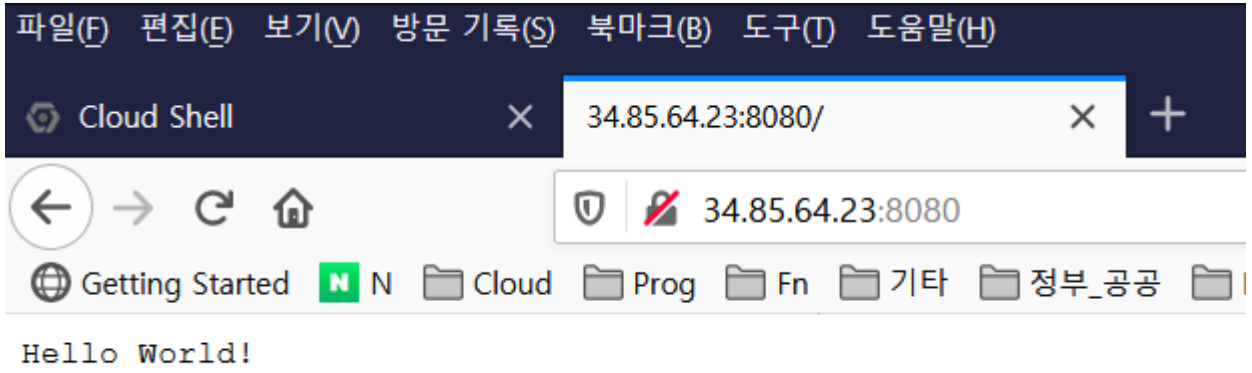
Kubernetes : External Traffic & Load-Balancing

실습 : Hello Node Kubernetes

kubectl expose deployment hello-node --type="LoadBalancer“
kubectl get services

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ kubectl expose deployment hello-node --type="LoadBalancer"
service/hello-node exposed
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ kubectl get services
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
hello-node   LoadBalancer   10.35.240.230   <pending>     8080:32647/TCP   41s
kubernetes   ClusterIP      10.35.240.1     <none>        443/TCP          22m
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ kubectl get services
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
hello-node   LoadBalancer   10.35.240.230   34.85.64.23   8080:32647/TCP   106s
kubernetes   ClusterIP      10.35.240.1     <none>        443/TCP          23m
```

http://<EXTERNAL_IP>:8080



Kubernetes : Scale-Up

실습 : Hello Node Kubernetes

- Scale-Up

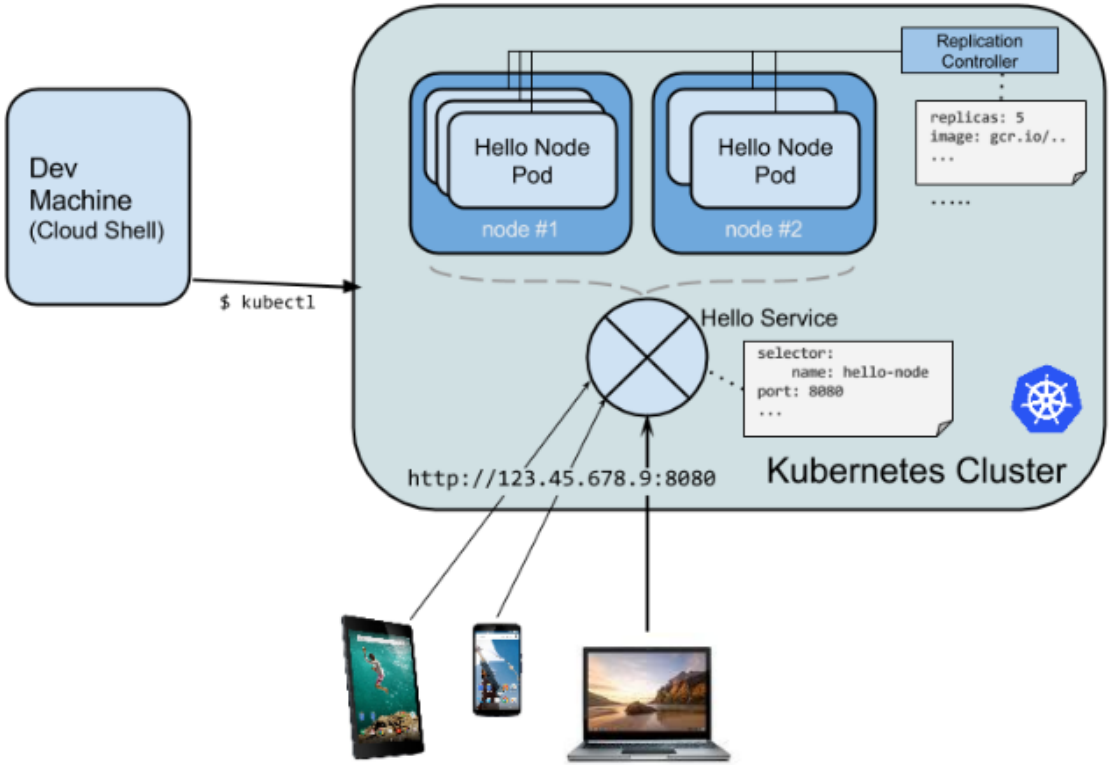
```
kubectl scale deployment hello-node --replicas=4
kubectl get deployment
kubectl get pods
```

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ kubectl scale deployment hello-node --replicas=4
deployment.extensions/hello-node scaled
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ kubectl get deployment
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
hello-node   1/4     4            1           23m
chaiwonkwon@cloudshell:~/Lab_k8s/lab02-2_K8s_HelloNode (project2020-270412)$ kubectl get pods
NAME                          READY   STATUS              RESTARTS   AGE
hello-node-566cc58495-8mzgh   1/1     Running             0          23m
hello-node-566cc58495-hm96p   0/1     ContainerCreating   0          2s
hello-node-566cc58495-hxrtg   0/1     ContainerCreating   0          2s
hello-node-566cc58495-pdglr   0/1     ContainerCreating   0          2s
```

실습 : Hello Node Kubernetes

- Scale-Up

```
kubectl scale deployment hello-node --replicas=4
kubectl get deployment
kubectl get pods
```



Kubernetes : Rolling Out an Upgrade

실습 : Hello Node Kubernetes

- Roll out an upgrade to your service

```
nano server.js
```

```
var http = require('http');
var handleRequest = function(request, response) {
 response.writeHead(200);
 response.end("Hello Kubernetes World!");
}
var www = http.createServer(handleRequest);
www.listen(8080);
```

```
docker build -t gcr.io/$PROJECT_ID/hello-node:v2 .
docker push gcr.io/$PROJECT_ID/hello-node:v2
```

실습 : Hello Node Kubernetes

```
export KUBE_EDITOR=nano
kubectl edit deployment hello-node
```

Look for Spec > containers > image and change the version number to v2:

```
      run: hello-node
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        run: hello-node
    spec:
      containers:
      - image: gcr.io/project2020-270412/hello-node:v1
        imagePullPolicy: IfNotPresent
        name: hello-node
        ports:
        - containerPort: 8080
          protocol: TCP
        resources: {}
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
```

:v2

```
chaiwonkwon@cloudshell:~ (project2020-270412)$ kubectl edit deployment hello-node
deployment.extensions/hello-node edited
```

실습 : Hello Node Kubernetes

kubectl get deployments

```
chaiwonkwon@cloudshell:~ (project2020-270412)$ kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
hello-node   4/4     4            4           44m
chaiwonkwon@cloudshell:~ (project2020-270412)$ kubectl get services
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
hello-node   LoadBalancer   10.35.240.230   34.85.64.23   8080:32647/TCP   32m
kubernetes   ClusterIP      10.35.240.1     <none>        443/TCP          54m
chaiwonkwon@cloudshell:~ (project2020-270412)$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
hello-node-6f64c677d8-82lfs   1/1     Running   0          3m35s
hello-node-6f64c677d8-8qjlh   1/1     Running   0          3m32s
hello-node-6f64c677d8-jn9lp   1/1     Running   0          3m31s
hello-node-6f64c677d8-pcjb7   1/1     Running   0          3m35s
```

Kubernetes : Deletion

실습 : Hello Node Kubernetes

gcloud container clusters list

gcloud container clusters delete hello-world --zone=asia-northeast1-c

```
chaiwonkwon@cloudshell:~ (project2020-270412)$ gcloud container clusters list
NAME         LOCATION           MASTER_VERSION  MASTER_IP      MACHINE_TYPE   NODE_VERSION    NUM_NODES  STATUS
hello-world  asia-northeast1-c  1.14.10-gke.24  35.243.81.252  n1-standard-1  1.14.10-gke.24  2          RUNNING
```

```
chaiwonkwon@cloudshell:~ (project2020-270412)$ gcloud container clusters delete hello-world --zone=asia-northeast1-c
The following clusters will be deleted.
 - [hello-world] in [asia-northeast1-c]

Do you want to continue (Y/n)?  Y

Deleting cluster hello-world...done.
Deleted [https://container.googleapis.com/v1/projects/project2020-270412/zones/asia-northeast1-c/clusters/hello-world].
chaiwonkwon@cloudshell:~ (project2020-270412)$ gcloud container clusters list
chaiwonkwon@cloudshell:~ (project2020-270412)$ gcloud compute instances list
```