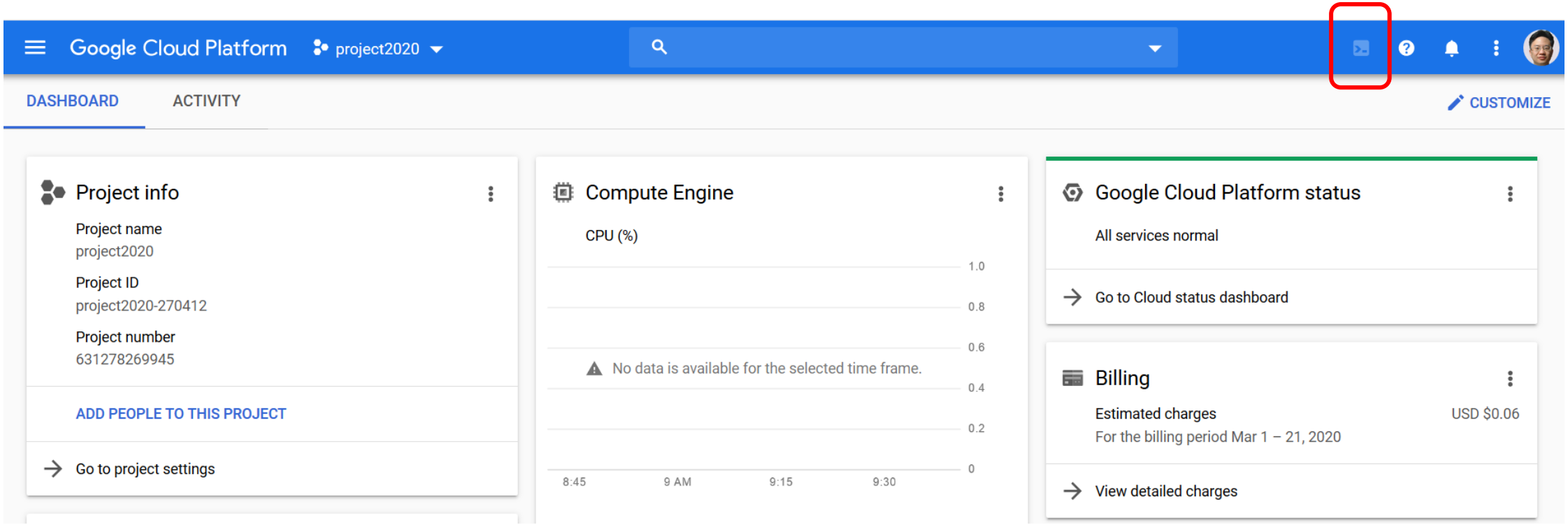
쿠버네티스를 활용한 멀티클라우드 도입과 운영전략
- AWS, Azure, GCP 비교와 실습

2020년 3월
아인인텔리전스
권재원, Ph.D.

Lab: Introduction to Kubernetes

- Using GCP CloudShell

실습 : CloudShell 시작하기



실습 : Introduction to Docker

- **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

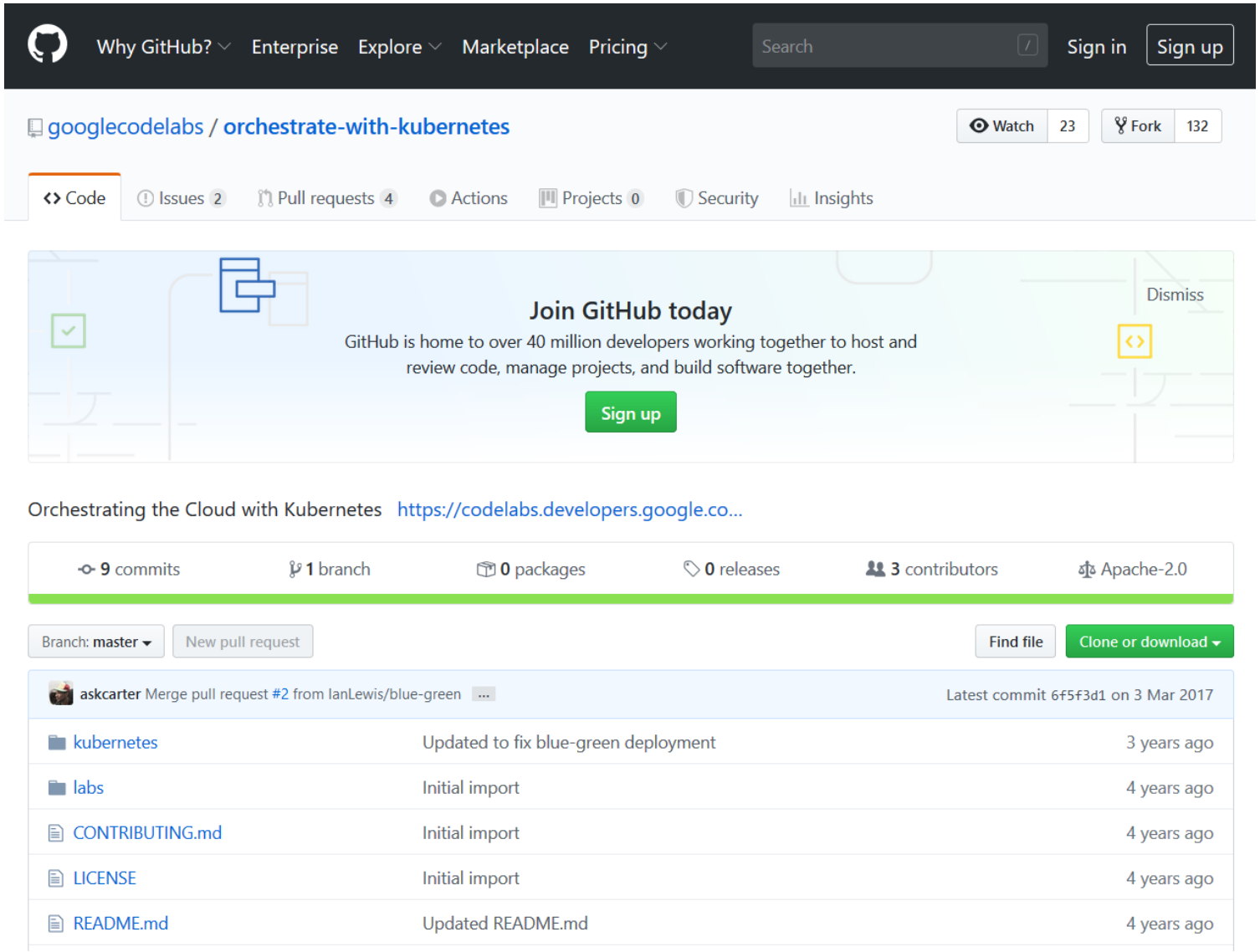
- Orchestrating the Cloud with Kubernetes

- Provision a complete Kubernetes cluster using Kubernetes Engine.
- Deploy and manage Docker containers using kubectl.
- Break an application into microservices using Kubernetes' Deployments and Services.

QwikLabs - https://www.qwiklabs.com/focuses/557?parent=catalog

실습 : Orchestrating the Cloud with Kubernetes

- https://github.com/googlecodelabs/orchestrate-with-kubernetes



실습 : Orchestrating the Cloud with Kubernetes

- App is hosted on GitHub and provides an example 12-Factor application. During this lab you will be working with the following Docker images:
- kelseyhightower/monolith - Monolith includes auth and hello services.
- kelseyhightower/auth - Auth microservice. Generates JWT tokens for authenticated users.
- kelseyhightower/hello - Hello microservice. Greets authenticated users.
- ngnix - Frontend to the auth and hello services.

Get the sample code

실습 : Orchestrating the Cloud with Kubernetes

```
$ gcloud config set compute/zone asia-northeast1-c
$ gcloud container clusters create io
```

```
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gcloud
/asia-northeast1-c/io?project=project2020-270412
kubeconfig entry generated for io.
NAME  LOCATION           MASTER_VERSION  MASTER_IP       MACHINE_TYPE   NODE_VERSION    NUM_NODES  STATUS
io    asia-northeast1-c  1.14.10-gke.24  35.194.111.122  n1-standard-1  1.14.10-gke.24  3          RUNNING
```

```
$ git clone https://github.com/googlecodelabs/orchestrate-with-kubernetes.git
$ cd orchestrate-with-kubernetes/kubernetes
$ ls -la
```

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ ls -la
total 32
drwxr-xr-x 7 chaiwonkwon chaiwonkwon 4096 Mar 22 16:26 .
drwxr-xr-x 5 chaiwonkwon chaiwonkwon 4096 Mar 22 16:26 ..
-rw-r--r-- 1 chaiwonkwon chaiwonkwon  283 Mar 22 16:26 cleanup.sh
drwxr-xr-x 2 chaiwonkwon chaiwonkwon 4096 Mar 22 16:26 deployments
drwxr-xr-x 2 chaiwonkwon chaiwonkwon 4096 Mar 22 16:26 nginx
drwxr-xr-x 2 chaiwonkwon chaiwonkwon 4096 Mar 22 16:26 pods
drwxr-xr-x 2 chaiwonkwon chaiwonkwon 4096 Mar 22 16:26 services
drwxr-xr-x 2 chaiwonkwon chaiwonkwon 4096 Mar 22 16:26 tls
```

Quick Kubernetes Demo

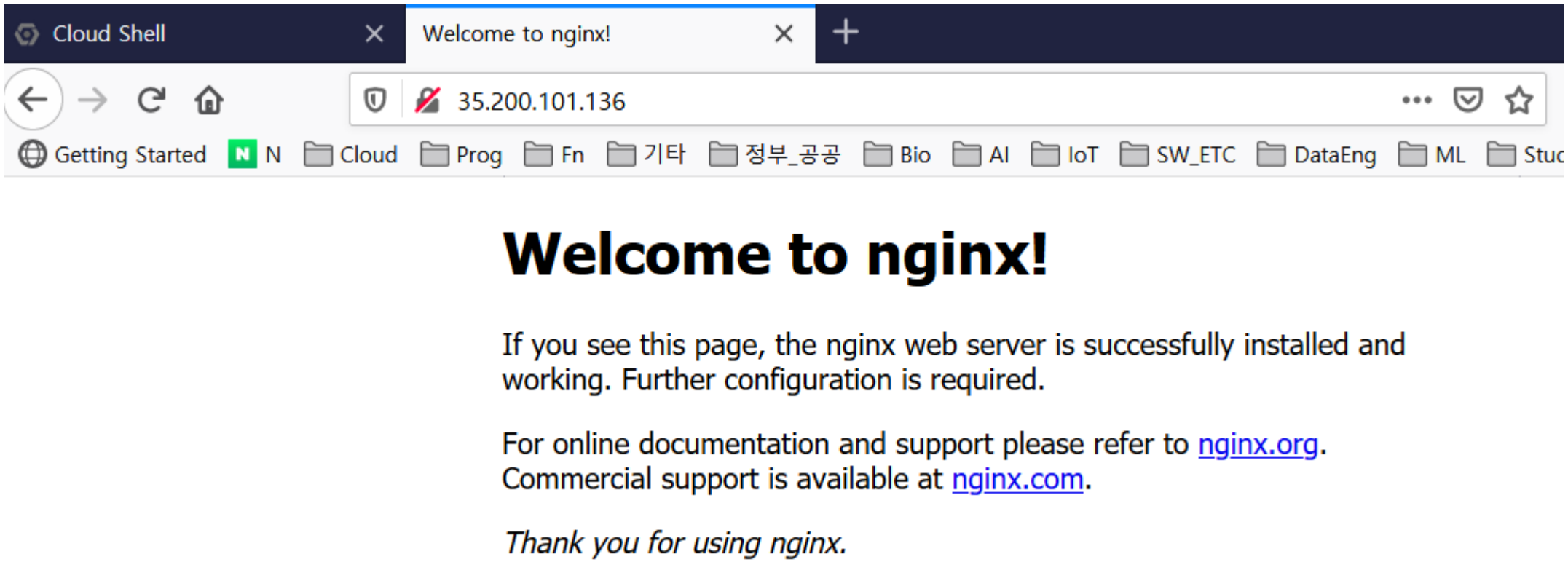
실습 : Orchestrating the Cloud with Kubernetes

$ kubectl create deployment nginx --image=nginx:1.10.0
$ kubectl get pods
$ kubectl expose deployment nginx --port 80 --type LoadBalancer
$ kubectl get services

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes (project2020-270412)$ kubectl create deployment
nginx --image=nginx:1.10.0
deployment.apps/nginx created
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes (project2020-270412)$ kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
nginx-574c99d7c8-slrh5   1/1     Running   0          38s
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes (project2020-270412)$ kubectl expose deployment
nginx --port 80 --type LoadBalancer
service/nginx exposed
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes (project2020-270412)$ kubectl get services
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP      10.51.240.1     <none>        443/TCP        8m1s
nginx        LoadBalancer   10.51.245.155   <pending>     80:30137/TCP   8s
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes (project2020-270412)$ kubectl get services
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP      10.51.240.1     <none>        443/TCP        8m27s
nginx        LoadBalancer   10.51.245.155   <pending>     80:30137/TCP   34s
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes (project2020-270412)$ kubectl get services
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
kubernetes   ClusterIP      10.51.240.1     <none>           443/TCP        8m50s
nginx        LoadBalancer   10.51.245.155   35.200.101.136   80:30137/TCP   57s
```

실습 : Orchestrating the Cloud with Kubernetes

http::/<External_IP>:80



Pods

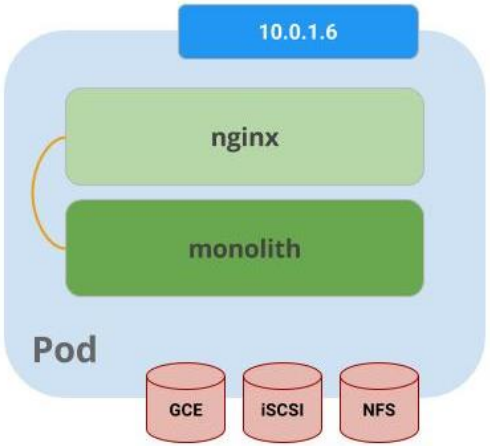
실습 : Orchestrating the Cloud with Kubernetes

```
$ cat pods/monolith.yaml
$ kubectl create -f pods/monolith.yaml
$ kubectl get pods
```

Pods

Logical Application

- One or more containers and volumes
- Shared namespaces
- One IP per pod




```
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl create
-f pods/monolith.yaml
pod/monolith created
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl get pod
s
NAME                     READY   STATUS    RESTARTS   AGE
monolith                 1/1     Running   0          33s
nginx-574c99d7c8-slrh5   1/1     Running   0          12m
```

```
$ kubectl describe pods monolith
```

실습 : Orchestrating the Cloud with Kubernetes

- In the 2nd terminal, run this command to set up port-forwarding:

```
$ kubectl port-forward monolith 10080:80
```

- Now in the 1st terminal start talking to your pod using curl:

```
$ curl http://127.0.0.1:10080
$ curl http://127.0.0.1:10080/secure
$ curl -u user http://127.0.0.1:10080/login
```

Password = password

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ curl http://127
.0.0.1:10080
{"message":"Hello"}
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ curl http://127
.0.0.1:10080/secure
authorization failed
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ curl -u user ht
tp://127.0.0.1:10080/login
Enter host password for user 'user':
{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6InVzZXJAZXhhbXBsZS5jb20iLCJleHAiOjE1ODUxMjI1NjUsImlhdCI6MT
U4NDg2MzM2NSwiaXNzIjoiYXV0aC5zZXJ2aWNlIiwic3ViIjoidXNlciJ9.Mgio2jue0S6kfRRuoqmvbrTH38aMiR-7A-EMFjS8WXc"}
```

```
$ TOKEN=$(curl http://127.0.0.1:10080/login -u user|jq -r '.token')
```

실습 : Orchestrating the Cloud with Kubernetes

- In the 2nd terminal, run this command to set up port-forwarding:

```
$ curl -H "Authorization: Bearer $TOKEN" http://127.0.0.1:10080/secure
$ kubectl logs monolith
```

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ curl -H "Author
ization: Bearer $TOKEN" http://127.0.0.1:10080/secure
{"message":"Hello"}
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl logs mo
nolith

2020/03/22 07:41:20 Starting server...
2020/03/22 07:41:20 Health service listening on 0.0.0.0:81
2020/03/22 07:41:20 HTTP service listening on 0.0.0.0:80
127.0.0.1:59282 - - [Sun, 22 Mar 2020 07:47:49 UTC] "GET / HTTP/1.1" curl/7.52.1
127.0.0.1:59338 - - [Sun, 22 Mar 2020 07:48:33 UTC] "GET /secure HTTP/1.1" curl/7.52.1
127.0.0.1:59412 - - [Sun, 22 Mar 2020 07:49:25 UTC] "GET /login HTTP/1.1" curl/7.52.1
127.0.0.1:59696 - - [Sun, 22 Mar 2020 07:53:03 UTC] "GET /login HTTP/1.1" curl/7.52.1
127.0.0.1:59802 - - [Sun, 22 Mar 2020 07:54:24 UTC] "GET /secure HTTP/1.1" curl/7.52.1
```

실습 : Orchestrating the Cloud with Kubernetes

- Open a 3rd terminal and use the -f flag to get a stream of the logs happening in real-time:

```
$ kubectl logs -f monolith
```

- Now if you use curl in the 1st terminal to interact with the monolith, you can see the logs updating (in the 3rd terminal):

```
$ curl http://127.0.0.1:10080
$ kubectl exec monolith --stdin --tty -c monolith /bin/sh
/ # ping -c 3 google.com
/ # exit
```

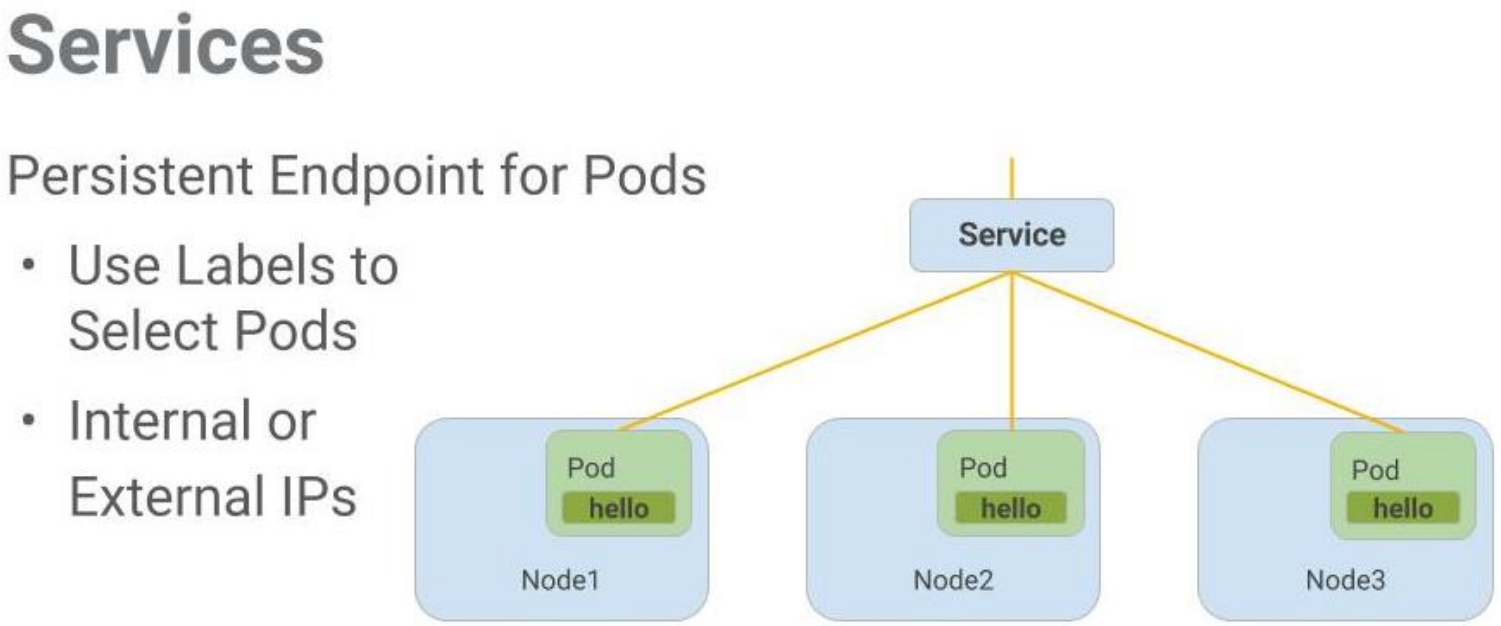
```
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl logs monolith

2020/03/22 07:41:20 Starting server...
2020/03/22 07:41:20 Health service listening on 0.0.0.0:81
2020/03/22 07:41:20 HTTP service listening on 0.0.0.0:80
127.0.0.1:59282 - - [Sun, 22 Mar 2020 07:47:49 UTC] "GET / HTTP/1.1" curl/7.52.1
127.0.0.1:59338 - - [Sun, 22 Mar 2020 07:48:33 UTC] "GET /secure HTTP/1.1" curl/7.52.1
127.0.0.1:59412 - - [Sun, 22 Mar 2020 07:49:25 UTC] "GET /login HTTP/1.1" curl/7.52.1
127.0.0.1:59696 - - [Sun, 22 Mar 2020 07:53:03 UTC] "GET /login HTTP/1.1" curl/7.52.1
127.0.0.1:59802 - - [Sun, 22 Mar 2020 07:54:24 UTC] "GET /secure HTTP/1.1" curl/7.52.1
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ curl http://127.0.0.1:10
080
{"message":"Hello"}
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ curl http://127.0.0.1:10
080
{"message":"Hello"}
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl exec monolith --
stdin --tty -c monolith /bin/sh
/ # ping -c 3 google.com
PING google.com (172.217.175.110): 56 data bytes
64 bytes from 172.217.175.110: seq=0 ttl=62 time=1.170 ms
64 bytes from 172.217.175.110: seq=1 ttl=62 time=1.337 ms
64 bytes from 172.217.175.110: seq=2 ttl=62 time=1.196 ms

--- google.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1.170/1.234/1.337 ms
/ # exit
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$
```

Services

실습：Orchestrating the Cloud with Kubernetes

Three Service's type

- ClusterIP (internal) -- the default type means that this Service is only visible inside of the cluster
- NodePort gives each node in the cluster an externally accessible IP
- LoadBalancer adds a load balancer from the cloud provider which forwards traffic from the service to Nodes within it.



실습 : Orchestrating the Cloud with Kubernetes

$ cd ~/orchestrate-with-kubernetes/kubernetes
$ cat pods/secure-monolith.yaml
$ kubectl create secret generic tls-certs --from-file tls/
$ kubectl create configmap nginx-proxy-conf --from-file nginx/proxy.conf
$ kubectl create -f pods/secure-monolith.yaml

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$  kubectl create secret g
eneric tls-certs --from-file tls/
secret/tls-certs created
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl create configmap
 nginx-proxy-conf --from-file nginx/proxy.conf
configmap/nginx-proxy-conf created
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl create -f pods/s
ecure-monolith.yaml
pod/secure-monolith created
```

실습 : Orchestrating the Cloud with Kubernetes

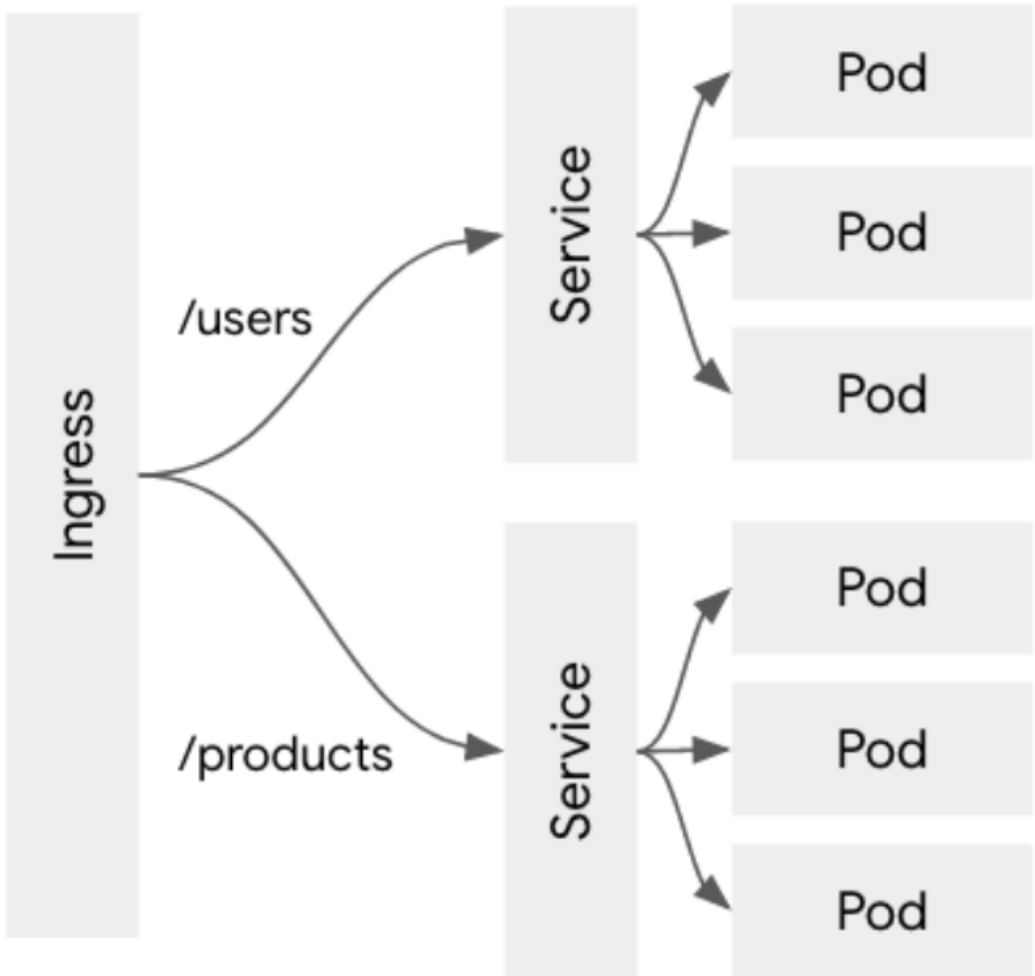
$ cat services/monolith.yaml

$ kubectl create -f services/monolith.yaml

$ gcloud compute firewall-rules create allow-monolith-nodeport --allow=tcp:31000

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ cat services/monolith
ml
kind: Service
apiVersion: v1
metadata:
  name: "monolith"
spec:
  selector:
    app: "monolith"
    secure: "enabled"
  ports:
    - protocol: "TCP"
      port: 443
      targetPort: 443
      nodePort: 31000
  type: NodePort
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl create -f ser
es/monolith.yaml
service/monolith created
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ gcloud compute firewa
rules create allow-monolith-nodeport \
>   --allow=tcp:31000

Creating firewall...⠹Created [https://www.googleapis.com/compute/v1/projects/project2020-270412/global/firewalls/allow-mono
h-nodeport].
Creating firewall...done.
NAME                    NETWORK  DIRECTION  PRIORITY  ALLOW      DENY  DISABLED
allow-monolith-nodeport  default  INGRESS    1000      tcp:31000        False
```

Adding Labels to Pods


Ingress
/users
/products
Service
Service
Pod
Pod
Pod
Pod
Pod
Pod

실습 : Orchestrating the Cloud with Kubernetes

$ kubectl get pods -l "app=monolith"
$ kubectl get pods -l "app=monolith,secure=enabled"

$ kubectl label pods secure-monolith 'secure=enabled'
$ kubectl get pods secure-monolith --show-labels

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl get pods
 -l "app=monolith"
NAME             READY   STATUS    RESTARTS   AGE
monolith         1/1     Running   0          4h35m
secure-monolith  2/2     Running   0          27m
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl get pods
 -l "app=monolith,secure=enabled"
No resources found.
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl label po
ds secure-monolith 'secure=enabled'
pod/secure-monolith labeled
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl get pods
 secure-monolith --show-labels
NAME             READY   STATUS    RESTARTS   AGE   LABELS
secure-monolith  2/2     Running   0          29m   app=monolith,secure=enabled
```

실습 : Orchestrating the Cloud with Kubernetes

$ kubectl describe services monolith | grep Endpoints

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl describe
 services monolith | grep Endpoints
Endpoints:                10.48.0.11:443
```

$ kubectl get nodes -o wide

$ curl -k https://<External_IP>:31000

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl get node
s -o wide
NAME                                STATUS   ROLES    AGE     VERSION           INTERNAL-IP   EXTERNAL-IP      OS-IMAG
E                           KERNEL-VERSION   CONTAINER-RUNTIME
gke-io-default-pool-a3b23fa7-29ll   Ready    <none>   5h58m   v1.14.10-gke.24   10.146.0.8    35.194.105.203   Contain
er-Optimized OS from Google   4.14.138+        docker://18.9.7
gke-io-default-pool-a3b23fa7-847k   Ready    <none>   5h58m   v1.14.10-gke.24   10.146.0.7    34.84.216.30     Contain
er-Optimized OS from Google   4.14.138+        docker://18.9.7
gke-io-default-pool-a3b23fa7-njw8   Ready    <none>   5h58m   v1.14.10-gke.24   10.146.0.9    34.84.51.138     Contain
er-Optimized OS from Google   4.14.138+        docker://18.9.7
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ curl -k https://
35.194.105.203:31000
{"message":"Hello"}
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ curl -k https://
34.84.216.30:31000
{"message":"Hello"}
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ curl -k https://
34.84.51.138:31000
{"message":"Hello"}
```

Deploying Applications with Kubernetes

Deployments

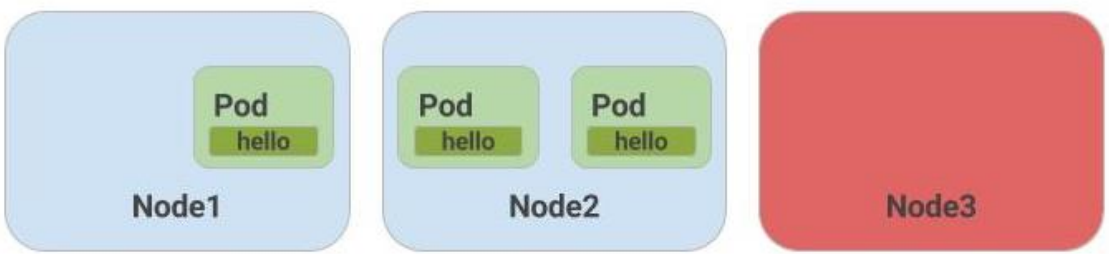
Drive current state towards desired state

app: hello
replicas: 3

Deployments

Drive current state towards desired state

app: hello
replicas: 3



실습 : Orchestrating the Cloud with Kubernetes

We're going to break the monolith app into three separate pieces:

- auth - Generates JWT tokens for authenticated users.
- hello - Greet authenticated users.
- frontend - Routes traffic to the auth and hello services.

```
$ cat deployments/auth.yaml
$ kubectl create -f deployments/auth.yaml
$ kubectl create -f services/auth.yaml
$ kubectl create -f deployments/hello.yaml
$ kubectl create -f services/hello.yaml
```

```
$ kubectl create configmap nginx-frontend-conf --from-file=nginx/frontend.conf
$ kubectl create -f deployments/frontend.yaml
$ kubectl create -f services/frontend.yaml
```

```
$ kubectl get services frontend
$ curl -k https://<EXTERNAL-IP>
```

실습 : Orchestrating the Cloud with Kubernetes

$ kubectl describe services monolith | grep Endpoints

```
chaiwonkwon@cloudshell:~/Lab_k8s/lab03-1_orchestrate-with-kubernetes/kubernetes (project2020-270412)$ kubectl describe
 services monolith | grep Endpoints
Endpoints:                10.48.0.11:443
```

$ kubectl get nodes -o wide

$ curl -k https://<External_IP>:31000