Kubernetes (GKE)

Kubernetes (GKE)

Course Outline

Day 1

- Google Container Registry
- Kubernetes Architecture
- Nodes
- Pods
- Services
- Replica Sets
- Deployments
- Daemon Sets
- Resource Quotas
- Health Check
- Horizontal Pod Autoscaler
- Labels
- Pod Disruption Budgets
- GCE PD

Day 2

- Disks
- Stateful Sets
- Config Maps
- Secrets
- Ingresses
- GCP L7 Ingress
- NGINX Ingress
- Automated https let's encrypt
- Jobs & CronJobs
- Automated Deploy

acoshift

acoshift/
course-kubernetes

DAY 1

YAML

- stands for “YAML Ain’t Markup Language”
- is a human friendly data serialization standard for all programming language

YAML

```yaml
name: Courses
list:
- name: Go for Beginner
  price: 600
- name: Redis Fundamental
  price: 300
- name: RxJS for Beginner
  price: 500
```

JSON

```json
{
  "name": "Courses",
  "list": [
    {
      "name": "Go for Beginner",
      "price": 600
    },
    {
      "name": "Redis Fundamental",
      "price": 300
    },
    {
      "name": "RxJS for Beginner",
      "price": 500
    }
  ]
}
```

Google Container Registry

https://cloud.google.com/container-registry/

Container Registry

REFRESH | SHOW PULL COMMAND | DELETE

gcr.io / acoshift-1362 / acourse

Filter by name or tag

Name	Tags	Virtual size	Uploaded
1f5261270fb3	7d9feb45ba5038b84856f51142e730a23fe9a3b9 latest	3.9 MB	2 days ago
6b81b1966f31	38f520dbf60aae34bb5528fd7559666e2c5f3eaf	3.9 MB	2 days ago
4dd203deaac1	09f311488b5756981ed1f8089b225032c98d1d1d	3.9 MB	2 days ago
b6bb495787e3	049cc2b0944aba25cf2b71a7b051cc5d8a807d5c	3.9 MB	2 days ago
a1e2abf97afa	09695bdb99fcbe4114e9e120e36b3a35881c2228	3.9 MB	4 days ago
c81615a32bfb	6444528f0898dc7074b03b2155702178e0cee3a4	3.9 MB	6 days ago
340072282343	390c17ac043b2f9881496b00430b451b49cb773a	3.9 MB	7 days ago
bf6d895e3d4f	b0b70dd5745c99969205a087b9b745c1fbdb6560	3.9 MB	8 days ago
d2d2aaa26269	5917485deaf70248110e2da0e6296e014967e1f5	3.9 MB	8 days ago
ef65e4cce15c	fe5253cdfa5a59e861fd2e61e7f77b7b7d97f1c5	3.9 MB	9 days ago
dfbc13dac286	977399a9770be959ae830da88933cd9e4884ad3f	3.9 MB	12 days ago

```
$ docker push acoshift/backend:1.0.0
$ gcloud docker -- push gcr.io/myproject/backend:1.0.0

$ docker pull acoshift/backend:1.0.0
$ gcloud docker -- pull gcr.io/myproject/backend:1.0.0
```

```
$ docker login -u _json_key -p "$(cat keyfile.json)" https://gcr.io
$ docker push gcr.io/myproject/backend:1.0.0
$ docker pull gcr.io/myproject/backend:1.0.0
```

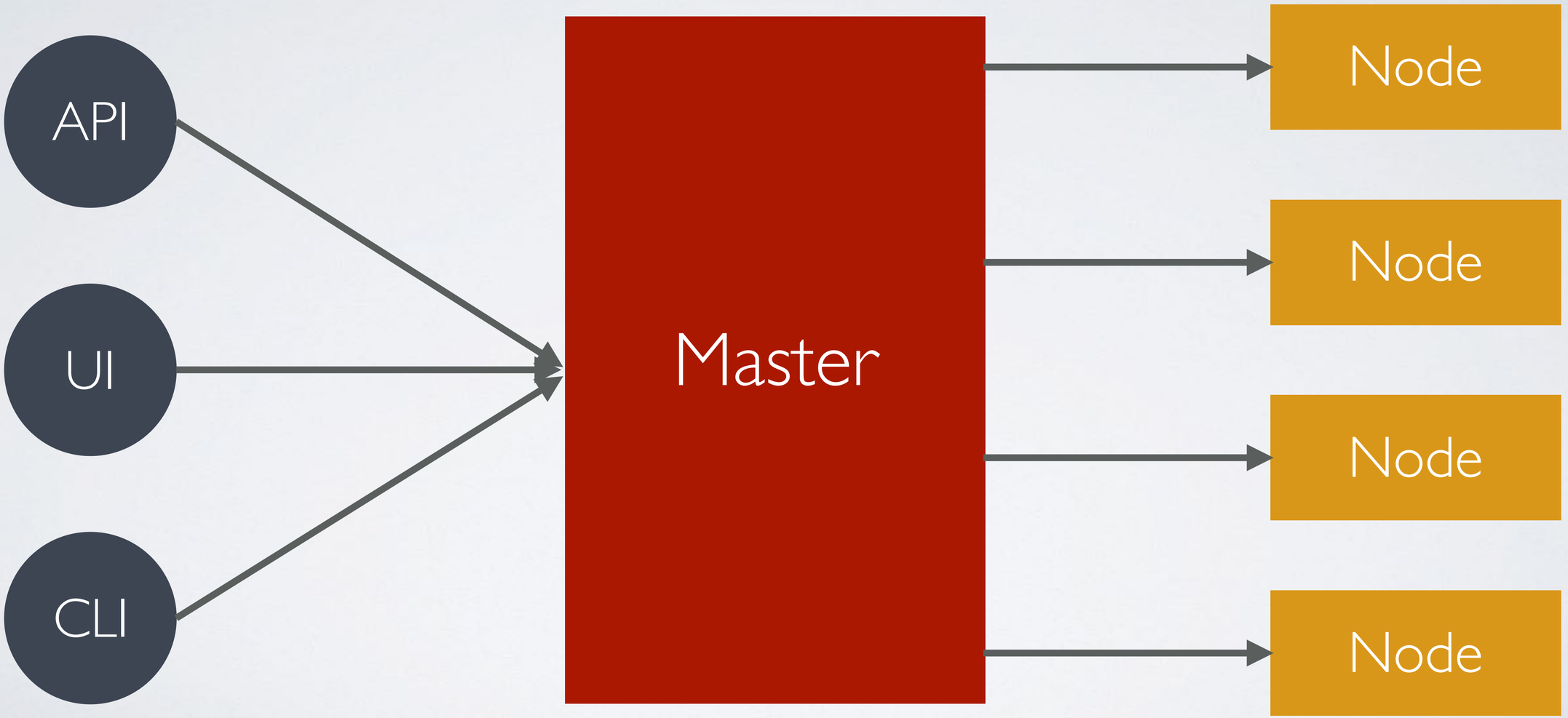
docker-credential-gcr

```
$ gcloud components install docker-credential-gcr
$ docker-credential-gcr configure-docker
$ docker-credential-gcr gcr-login
```

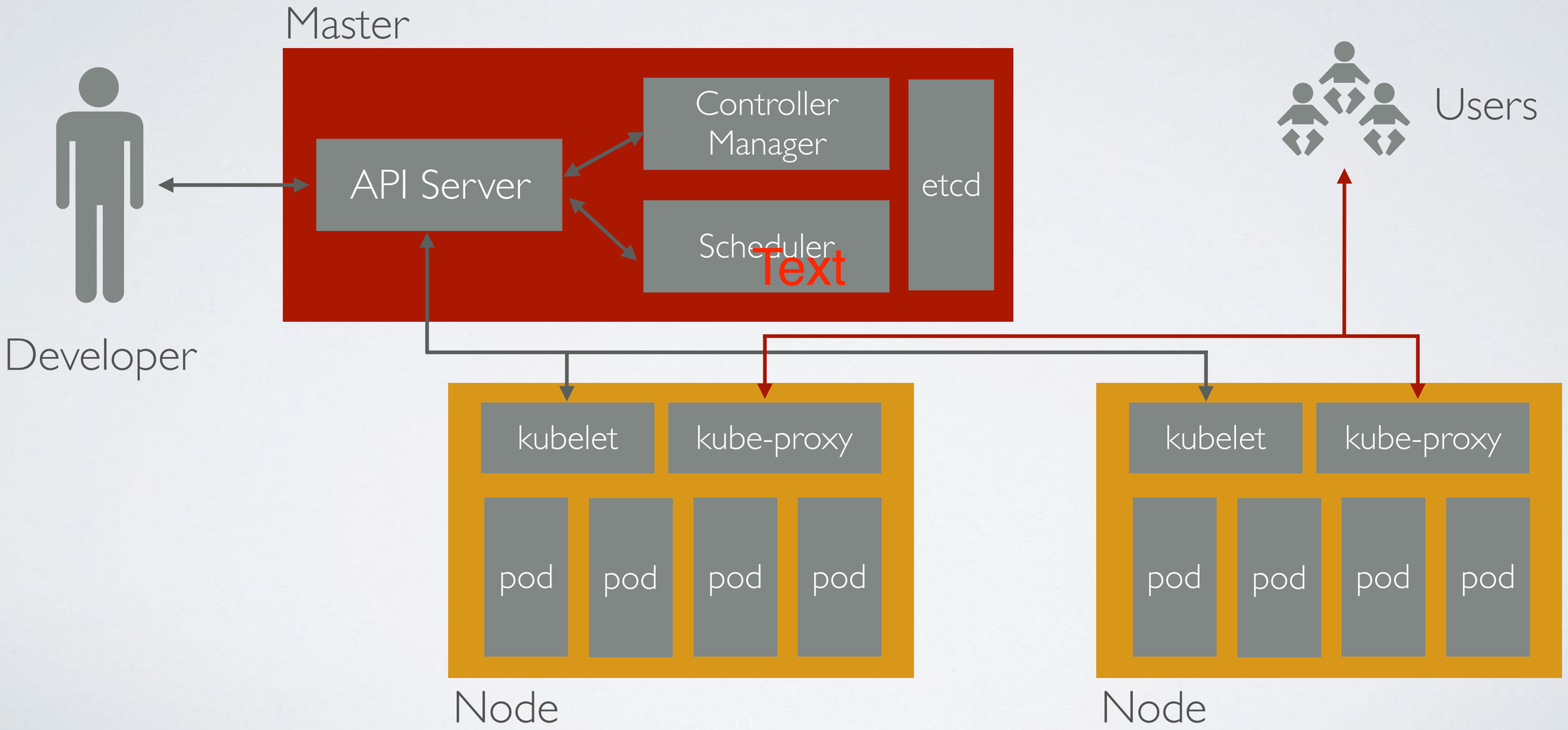
```
$ docker push gcr.io/myproject/backend:1.0.0
$ docker pull gcr.io/myproject/backend:1.0.0
```

https://gcr.io/google-containers/global

Kubernetes Architecture



Kubernetes Architecture



พิมพ์ในคำสั่งแบบ เอกพจน์ พหูพจน์ แล้วก็ย่อได้

Nodes (no)

a worker machine in Kubernetes

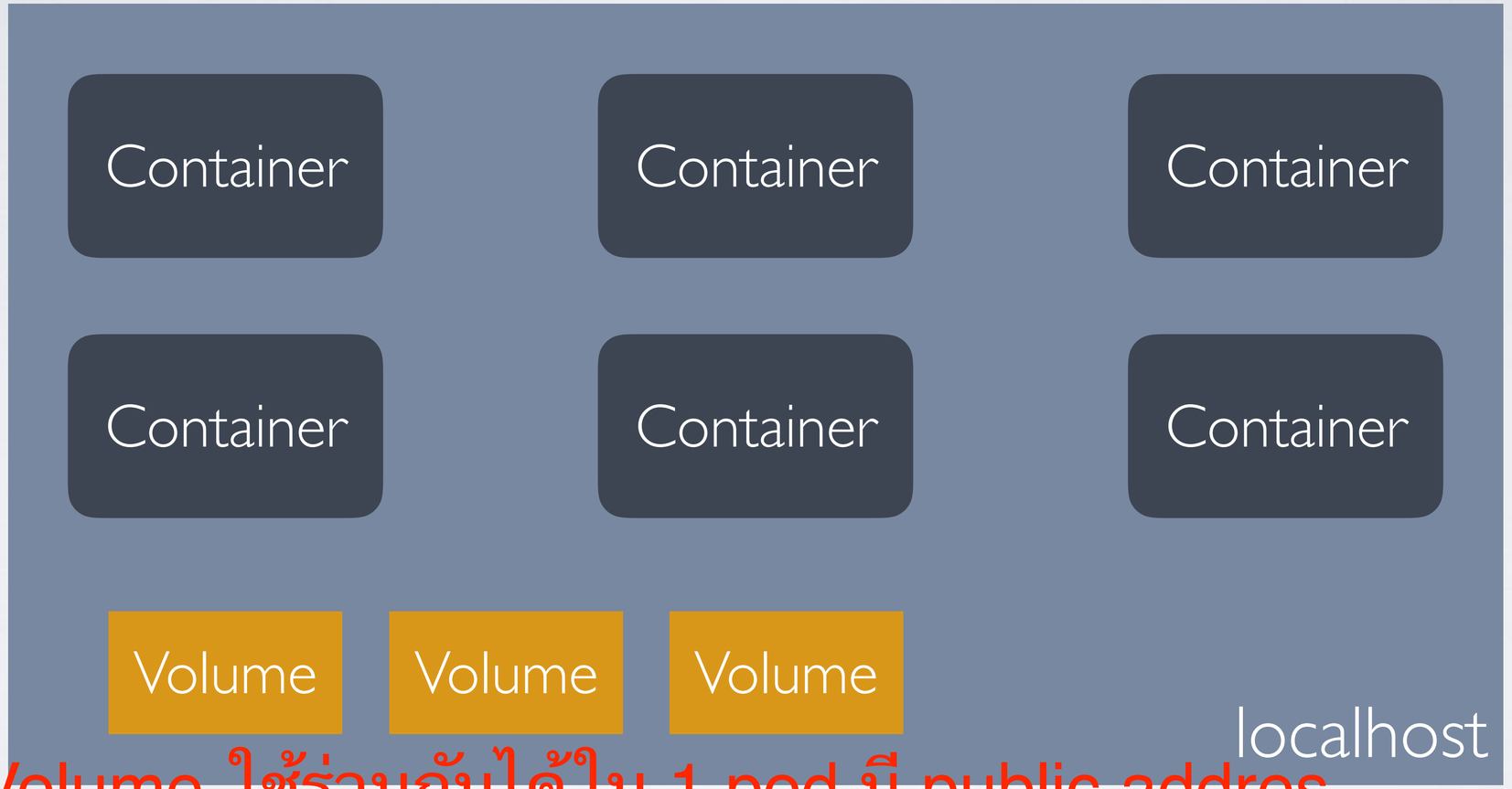
```
$ kubectl get nodes
NAME                                        STATUS    AGE       VERSION
gke-cluster-sg-1-pool-1-3fada004-n6gj   Ready     4d        v1.7.0
gke-cluster-sg-1-pool-1-3fada004-pglr   Ready     4d        v1.7.0

$ kubectl describe nodes gke-cluster-sg-1-pool-1-3fada004-n6gj
```

กลุ่มของ Container 1 Pods มีได้หลาย container

Pods (po)

a group of one or more containers

ส่วนใหญ่มีอันเดียว
แต่ละ Container ต้องแตกต่างกันเพราะคุย



Volume ใช้ร่วมกันได้ใน 1 pod มี public addres

```yaml
kind: Pod
apiVersion: v1
metadata:
  name: echoserver
spec:
  containers:
  - name: echoserver
    image: gcr.io/google-containers/echoserver:1.6
    ports:
    - containerPort: 8080
```

ชื่อ

ส่วนใหญ่ตั้งชื่อเดียวกับพอร์ต

array

ไม่ต้องใส่ก็ได้ ก็เข้าได้

just additional information

all ports listening on 0.0.0.0 will be accessible from network

-f = find

```
$ kubectl create -f pod.yaml
pod "echoserver" created
```

```
$ kubectl get pods
NAME                READY     STATUS    RESTARTS   AGE
echoserver          1/1       Running   0          4m
```

-o=wide -o json

เครื่องเรา: container ลอง http ได้เลย

```
$ kubectl port-forward echoserver 9000:8080
Forwarding from 127.0.0.1:9000 -> 8080
Forwarding from [::1]:9000 -> 8080
```

```
$ curl localhost:9000
Hostname: echoserver

Pod Information:
  -no pod information available-

Server values:
  server_version=nginx: 1.13.1 - lua: 10008

Request Information:
  client_address=127.0.0.1
  method=GET
  real path=/
  query=
  request_version=1.1
  request_uri=http://localhost:8080/

Request Headers:
  accept=*/*
  host=localhost:9000
  user-agent=curl/7.51.0

Request Body:
  -no body in request-
```

Text

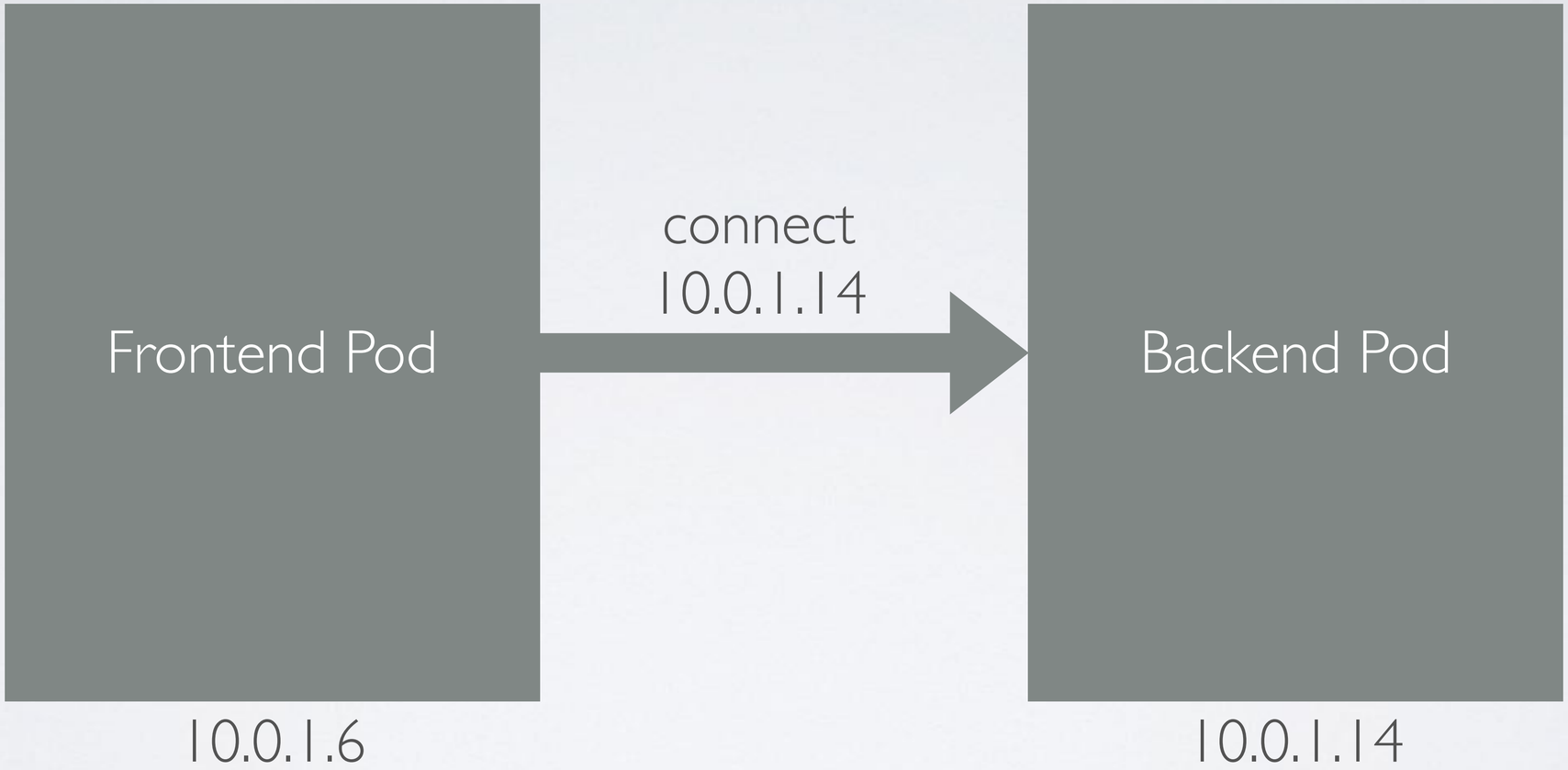
```
$ kubectl delete pod echoserver
pod "echoserver" deleted
```

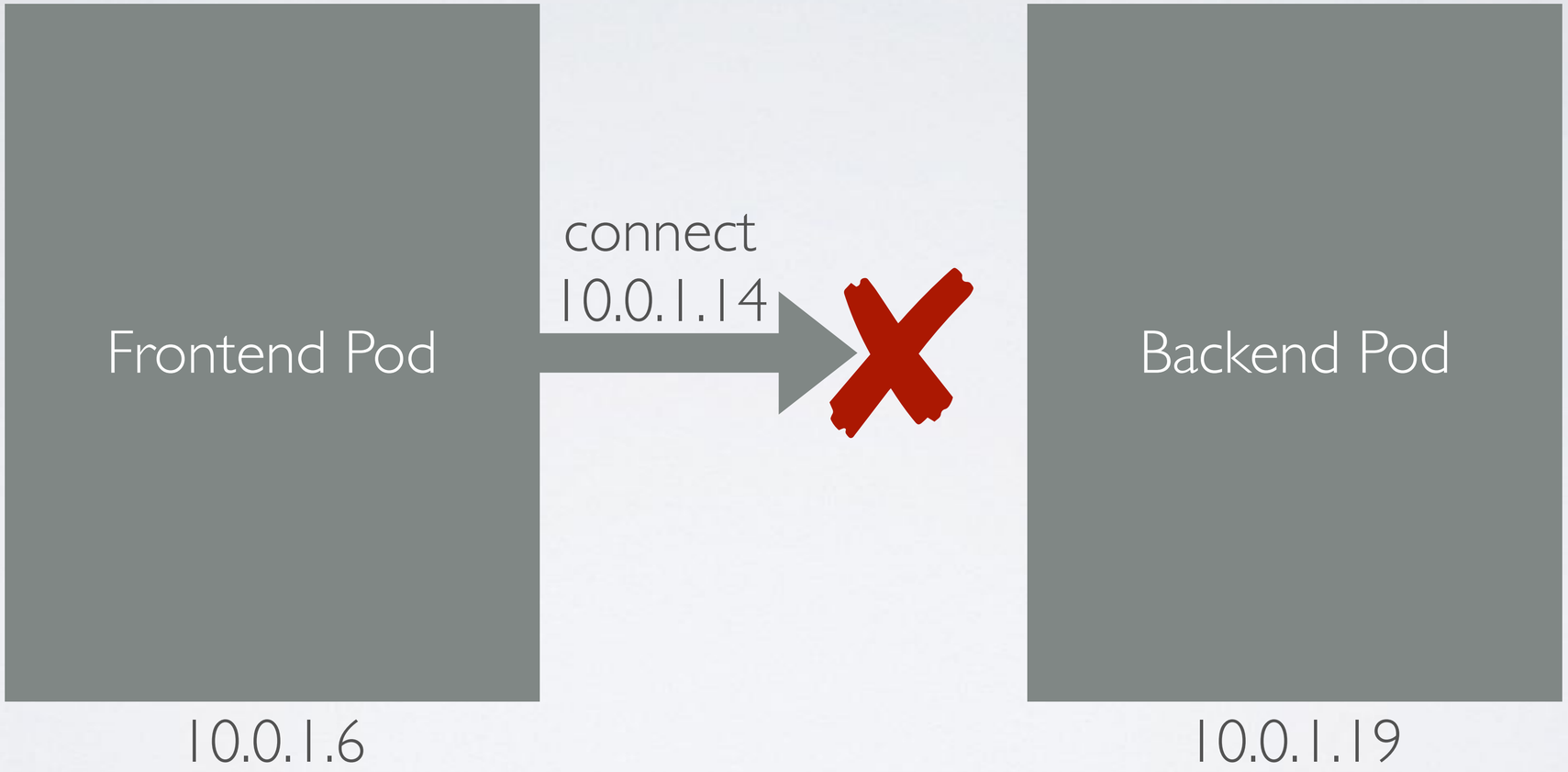
```yaml
kind: Pod
apiVersion: v1
metadata:
  name: web
spec:
  volumes:
  - name: www
    emptyDir: {}
  containers:
  - name: nginx
    image: gcr.io/google-containers/nginx-slim:0.8
    ports:
    - containerPort: 80
    volumeMounts:
    - name: www
      mountPath: /usr/share/nginx/html
  - name: ubuntu
    image: ubuntu
    volumeMounts:
    - name: www
      mountPath: /data
    command:
    - /bin/sh
    args:
    - -c
    - while true; do dd if=/dev/urandom bs=32 count=1 | base64 > /data/index.html; sleep 1; done
```

```
$ kubectl create -f multi-container.yaml
pod "web" created

$ kubectl port-forward web 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80

$ curl localhost:8080
```

```
$ kubectl exec web -itc ubuntu -- bash

root@web:/# apt-get update
root@web:/# apt-get install curl

root@web:/# curl localhost
s4zX9vA0juonZhYlCjfRiXUpIV54EsAfz+UwAgnrWhA=

root@web:/# curl localhost
n7dhG+ZDK//+vQm/M6upoA55JqK5lQ96tYsiDdGj+7M=

root@web:/# curl localhost -I
HTTP/1.1 200 OK
Server: nginx/1.11.1
Date: Fri, 21 Jul 2017 12:26:55 GMT
Content-Type: text/html
Content-Length: 45
Last-Modified: Fri, 21 Jul 2017 12:26:54 GMT
Connection: keep-alive
ETag: "5971f30e-2d"
Accept-Ranges: bytes
```

```
$ kubectl delete -f multi-container.yaml
pod "web" deleted
```


Frontend Pod
connect
10.0.1.14
Backend Pod
10.0.1.6
10.0.1.14


Frontend Pod
connect
10.0.1.14
Backend Pod
10.0.1.6
10.0.1.19

Services (svc)

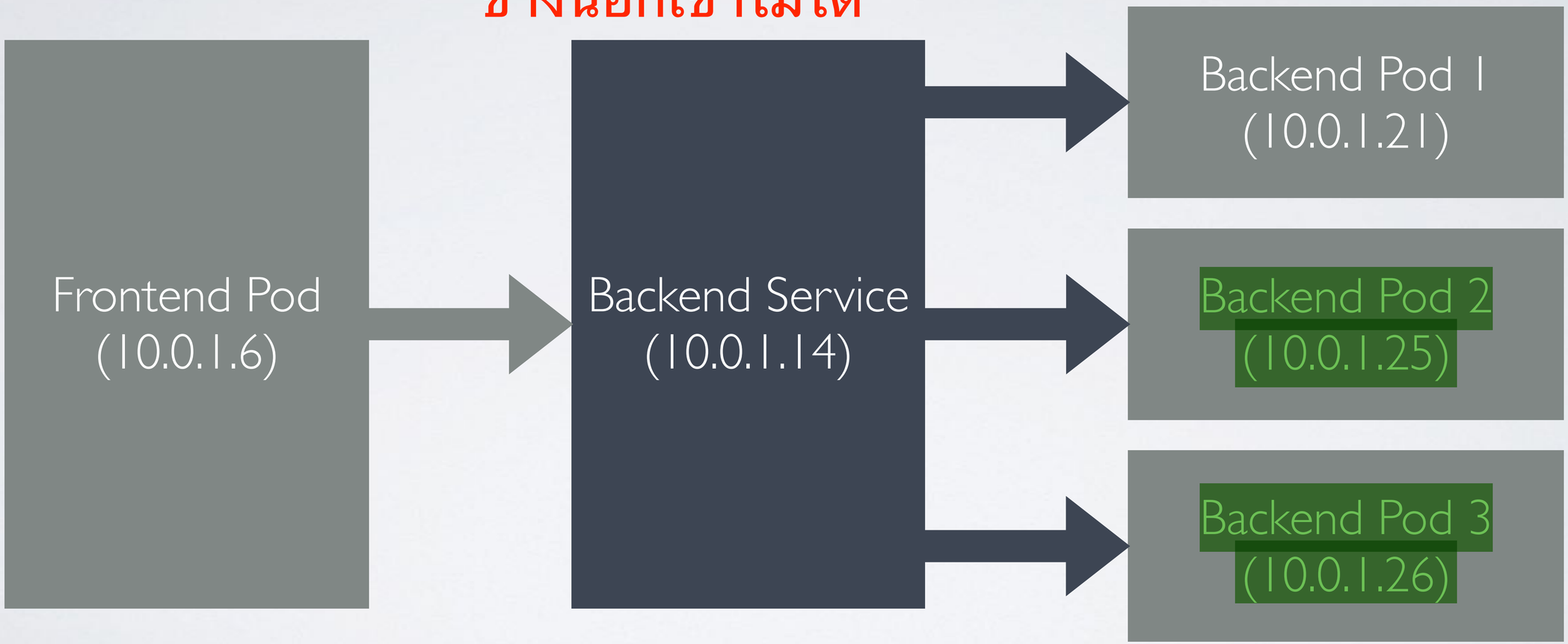
an abstraction which defines a logical set of Pods and a policy by which to access them

Service Types

- ClusterIP
- NodePort
- LoadBalancer
- ExternalName

ClusterIP

Internal Load Balancer มี DNS
ข้างนอกเข้าไม่ได้



พิมพ์ http://backend ได้เลย

```yaml
kind: Pod
apiVersion: v1
metadata:
  name: echoserver
  labels:
    app: echoserver
spec:
  containers:
  - name: echoserver
    image: gcr.io/google-containers/echoserver:1.6
    ports:
    - containerPort: 8080
```

ไม่สามารถตั้งพอร์ทสองอันชื่อ
เดียวกันได้
เลยต้องตั้ง labels -> key - values
app: echoserver
Ser

```
kind: Service
apiVersion: v1
metadata:
  name: echoserver
spec:
  selector:
    app: echoserver
  ports:
  - port: 80
    targetPort: 8080
```

Service ดูจาก labels

Service เปิดพอร์ท 80 แต่ยิงไปที่ 8080 ของ echoserver

```
$ kubectl create -f clusterIp.yaml
pod "echoserver" created
service "echoserver" created
```

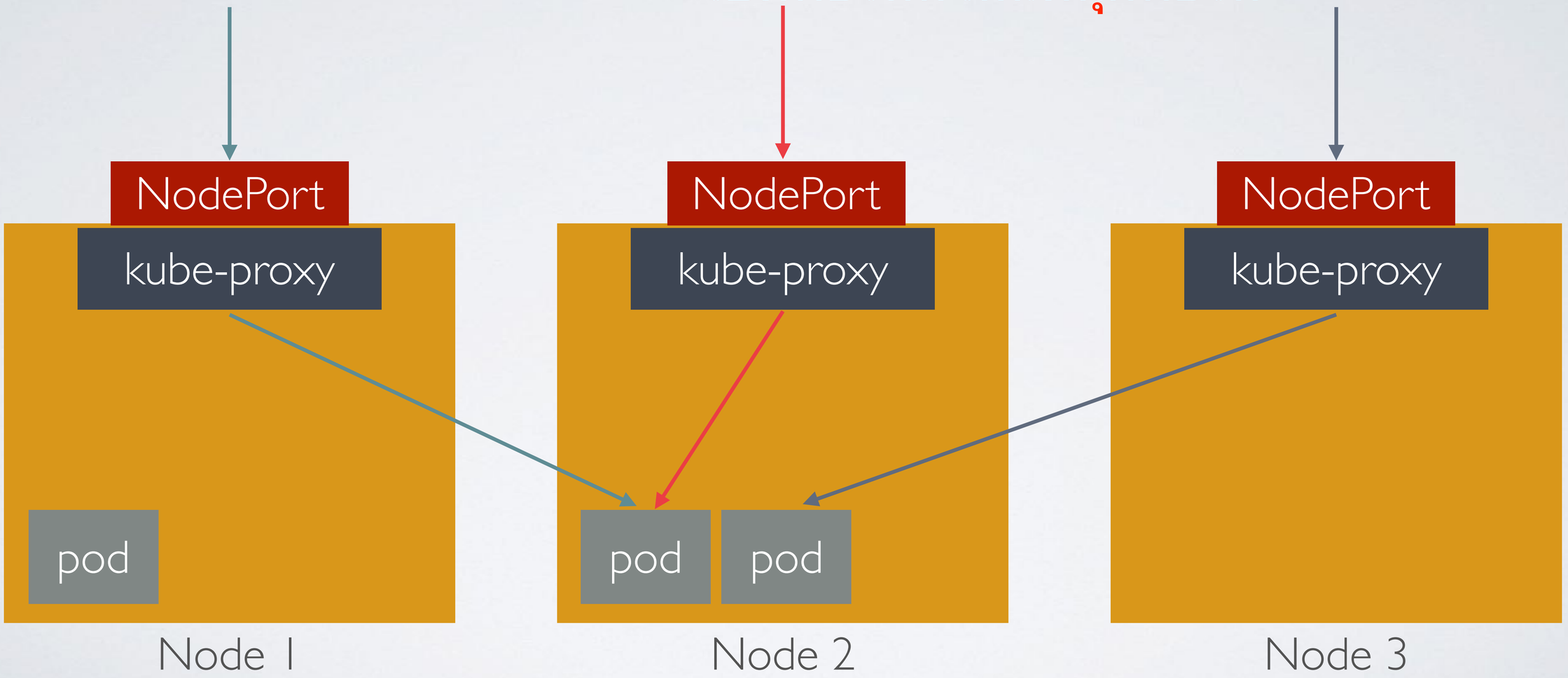
```
$ kubectl get services
NAME          CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
echoserver    10.3.248.15  <none>        80/TCP    11s
```

```
$ kubectl run -it --rm busybox --image=busybox
$ wget -O- http://echoserver
```

```
$ kubectl delete -f clusterIp.yaml
```

NodePort

เปิดพอร์ทที่โหนดทุกพอร์ท



```yaml
kind: Service
apiVersion: v1
metadata:
  name: echoserver
spec:
  type: NodePort
  selector:
    app: echoserver
  ports:
  - port: 80
    targetPort: 8080
    nodePort: 31000
```

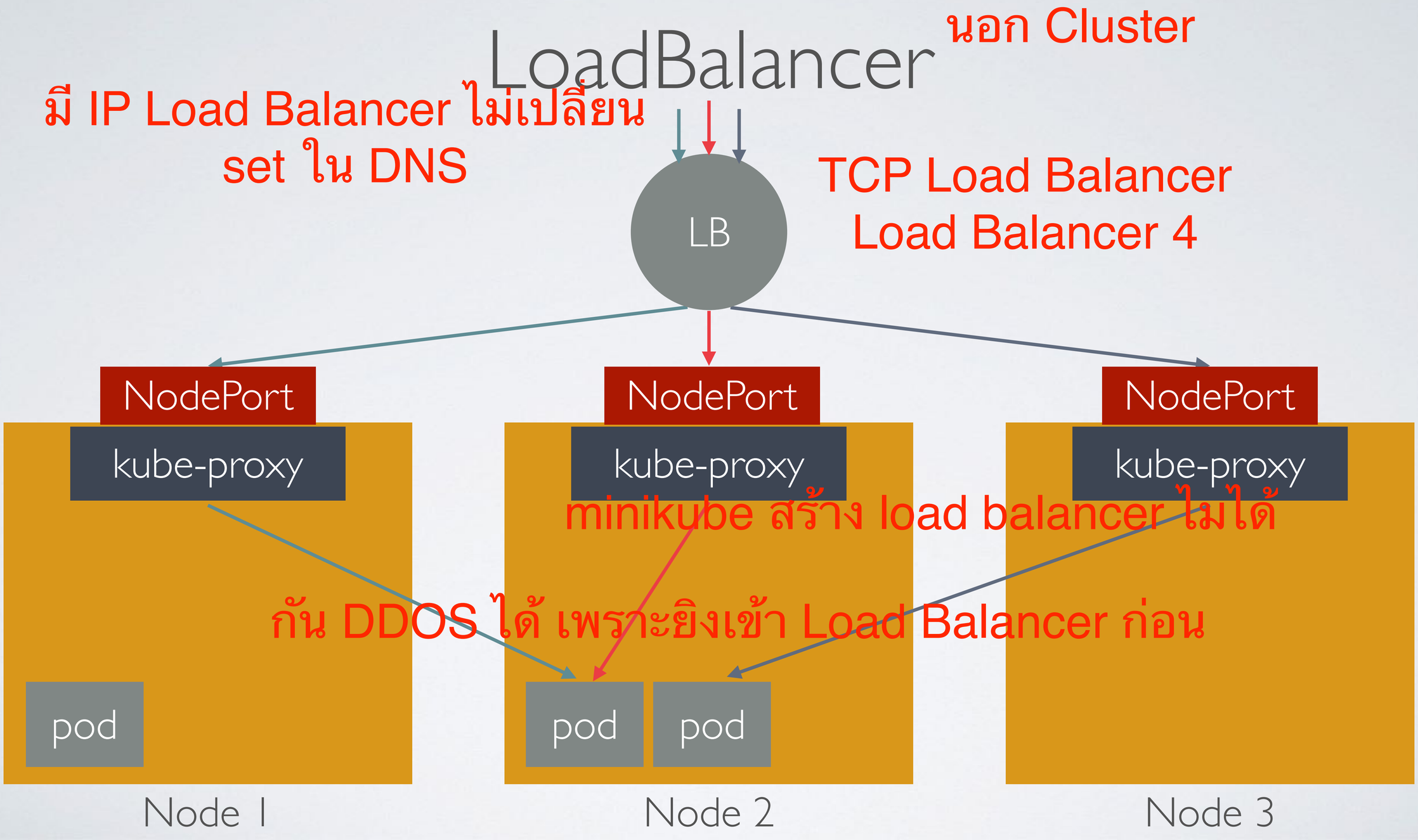
มีพอร์ทเปิดพอ

55.107.244

ร์ทได้แค่นี้

valid port:
30000-32767

```
$ curl http://serverIP:31000
```


LoadBalancer
นอก Cluster
มี IP Load Balancer ไม่เปลี่ยน
set ใน DNS
LB
TCP Load Balancer
Load Balancer 4
NodePort
kube-proxy
NodePort
kube-proxy
NodePort
kube-proxy
minikube สร้าง load balancer ไม่ได้
กัน DDOS ได้ เพราะยิงเข้า Load Balancer ก่อน
pod
pod
pod
Node 1
Node 2
Node 3

```
kind: Service
apiVersion: v1
metadata:
  name: echoserver
spec:
  type: LoadBalancer
  selector:
    app: echoserver
  ports:
  - port: 80
    targetPort: 8080
  loadBalancerIP: 35.185.1.1
```

Text

optional static ip

```
$ curl http://loadbalcnerIP
```

ExternalName

อยากยิงข้างนอกแล้วย้ายเข้ามาทีหลัง

```
kind: Service
apiVersion: v1
metadata:
  name: google
spec:
  type: ExternalName
  externalName: google.com
```

Text

```
$ kubectl run -it --rm busybox --image=busybox
$ wget -O- --header="Host: www.google.com" http://google
```

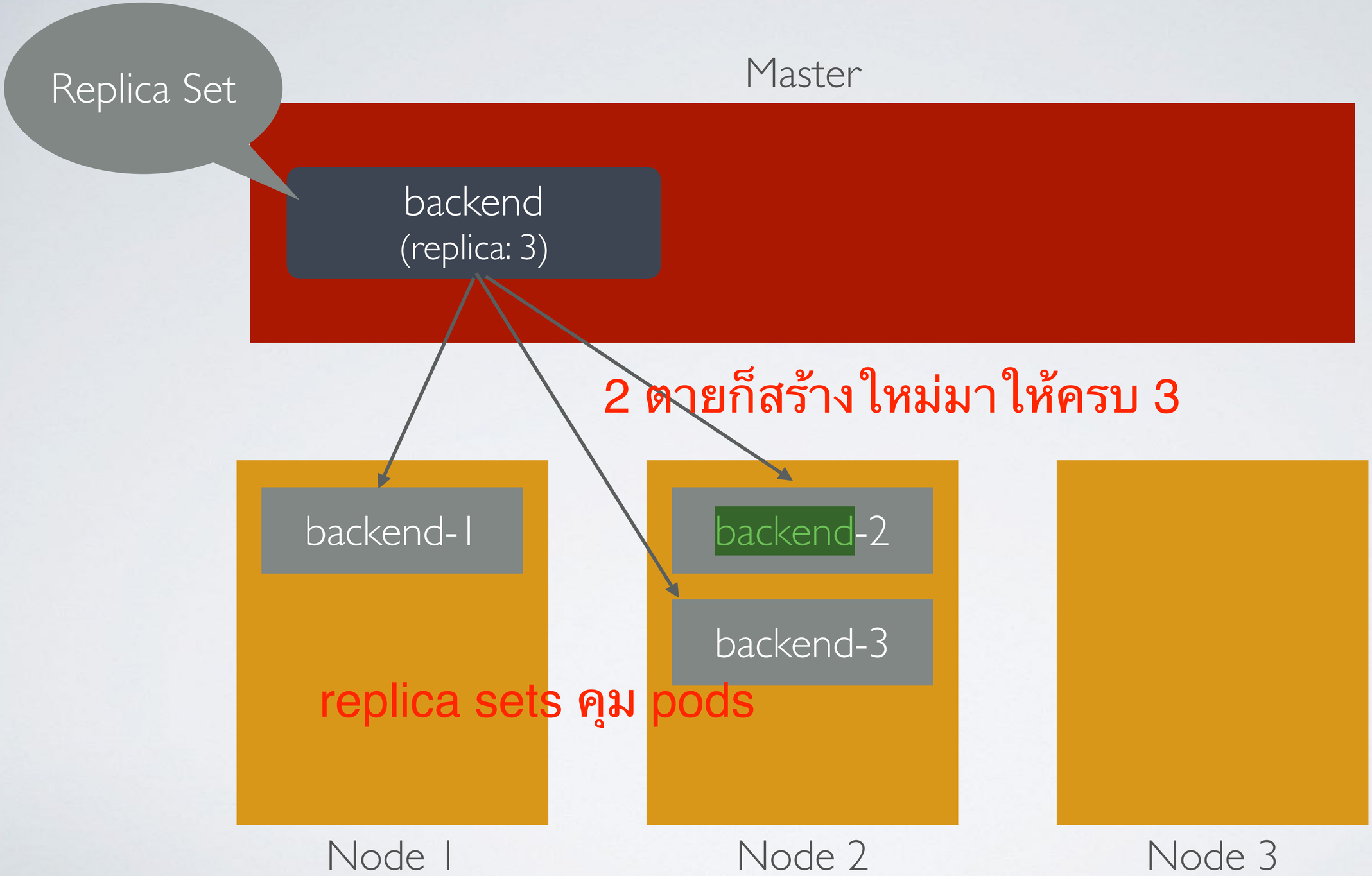
~~Replication Controller (rc)~~

Replica Sets (rs)

the next-generation Replication Controller

ensures that a specified number of pod “replicas” are running at any given time

replica sets คุม pods


Replica Set
Master
backend
(replica: 3)
2 ตายก็สร้างใหม่มาให้ครบ 3
backend-1
backend-2
backend-3
replica sets คุม pods
Node 1
Node 2
Node 3

```
kind: ReplicaSet
apiVersion: apps/v1beta1
metadata:
  name: echoserver
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: echoserver
    spec:
      containers:
      - name: echoserver
        image: gcr.io/google-containers/echoserver:1.6
        ports:
        - containerPort: 8080
```

kubectl scale rs/echoserver --replicsa=2

apps/v1 selector -> replicas take sskubover existing pod

Text

Pod

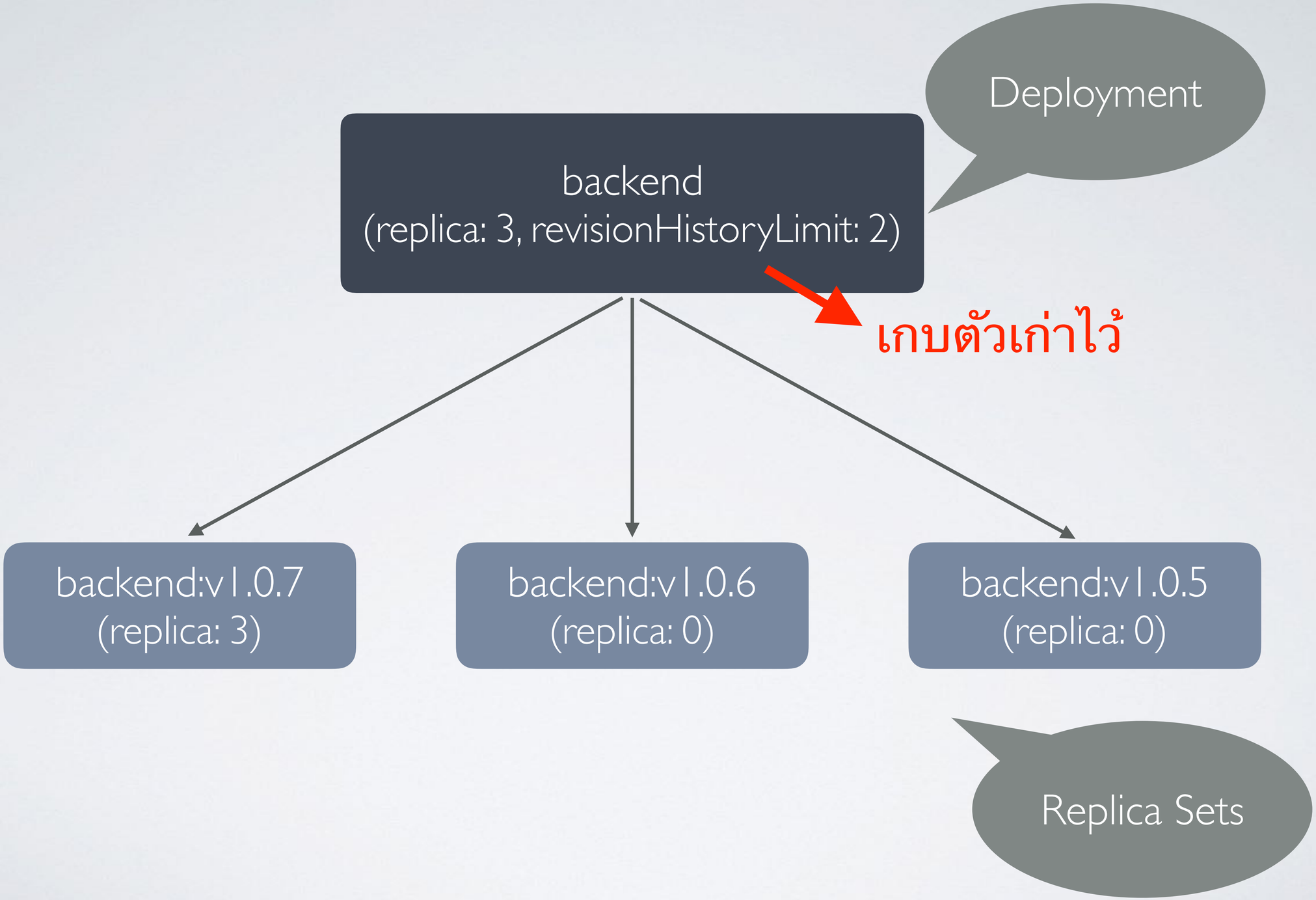
replica sets คุม pods

replica sets คุม pods

คุม replica set อีกทีนึง
ตัว manage version

Deployments (deploy)

provides declarative updates for Pods and ReplicaSets


Deployment
backend
(replica: 3, revisionHistoryLimit: 2)
เกบตัวเก่าไว้
backend:v1.0.7
(replica: 3)
backend:v1.0.6
(replica: 0)
backend:v1.0.5
(replica: 0)
Replica Sets

Update Strategy

Default

kubectl scale rs/echoserver --replicsa=2

- RollingUpdate — updates one pod at a time
  - Max Unavailable — maximum number of Pods that can be unavailable during the update process

ลดพอร์ทต่ำกว่า replica ที่เราตั้งไว้

default 1

  - Max Surge — maximum number of Pods that can be created above the desired number of Pods

default 1

- Recreate — All existing Pods are killed before new ones are created

ปิดพอร์ทแล้วค่อยเปิดใหม่

```yaml
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: echoserver
spec:
  replicas: 3
  revisionHistoryLimit: 2
  template:
    metadata:
      labels:
        app: echoserver
    spec:
      containers:
      - name: echoserver
        image: gcr.io/google-containers/echoserver:1.1
        ports:
        - containerPort: 8080
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
```

```
$ kubectl create -f deployment.yaml --record=true
deployment "echoserver" created

$ kubectl set image deployment/echoserver echoserver=gcr.io/google-containers/echoserver:1.2
deployment "echoserver" image updated

$ kubectl rollout status deployment/echoserver
Waiting for rollout to finish: 1 out of 3 new replicas have been updated...
Waiting for rollout to finish: 1 out of 3 new replicas have been updated...
Waiting for rollout to finish: 2 out of 3 new replicas have been updated...
Waiting for rollout to finish: 2 out of 3 new replicas have been updated...
Waiting for rollout to finish: 2 out of 3 new replicas have been updated...
Waiting for rollout to finish: 1 old replicas are pending termination...
Waiting for rollout to finish: 1 old replicas are pending termination...
deployment "echoserver" successfully rolled out
```

```
$ kubectl set image deployment/echoserver echoserver=gcr.io/google-containers/echoserver:1.3
deployment "echoserver" image updated

$ kubectl rollout history deployment/echoserver
deployments "echoserver"
REVISION        CHANGE-CAUSE
1               kubectl create -filename=deployment.yaml --record=true
2               kubectl set image deployment/echoserver echoserver=gcr.io/google-containers/echoserver:1.2
3               kubectl set image deployment/echoserver echoserver=gcr.io/google-containers/echoserver:1.3
```

```
$ kubectl rollout history deployment/echoserver --revision=2
deployments "echoserver" with revision #2
Pod Template:
  Labels:       app=echoserver
        pod-template-hash=1885346732
  Annotations:  kubernetes.io/change-cause=kubectl set image deployment/echoserver echoserver=gcr.io/google-containers/echoserver:1.2
  Containers:
   echoserver:
    Image:      gcr.io/google-containers/echoserver:1.2
    Port:       8080/TCP
    Environment:        <none>
    Mounts:     <none>
  Volumes:      <none>
```

```
$ kubectl rollout undo deployment/echoserver
deployment "echoserver" rolled back

$ kubectl rollout history deployment/echoserver
deployments "echoserver"
REVISION        CHANGE-CAUSE
1               kubectl create --filename=deployment.yaml --record=true
3               kubectl set image deployment/echoserver echoserver=gcr.io/google-containers/echoserver:1.3
4               kubectl set image deployment/echoserver echoserver=gcr.io/google-containers/echoserver:1.2
```

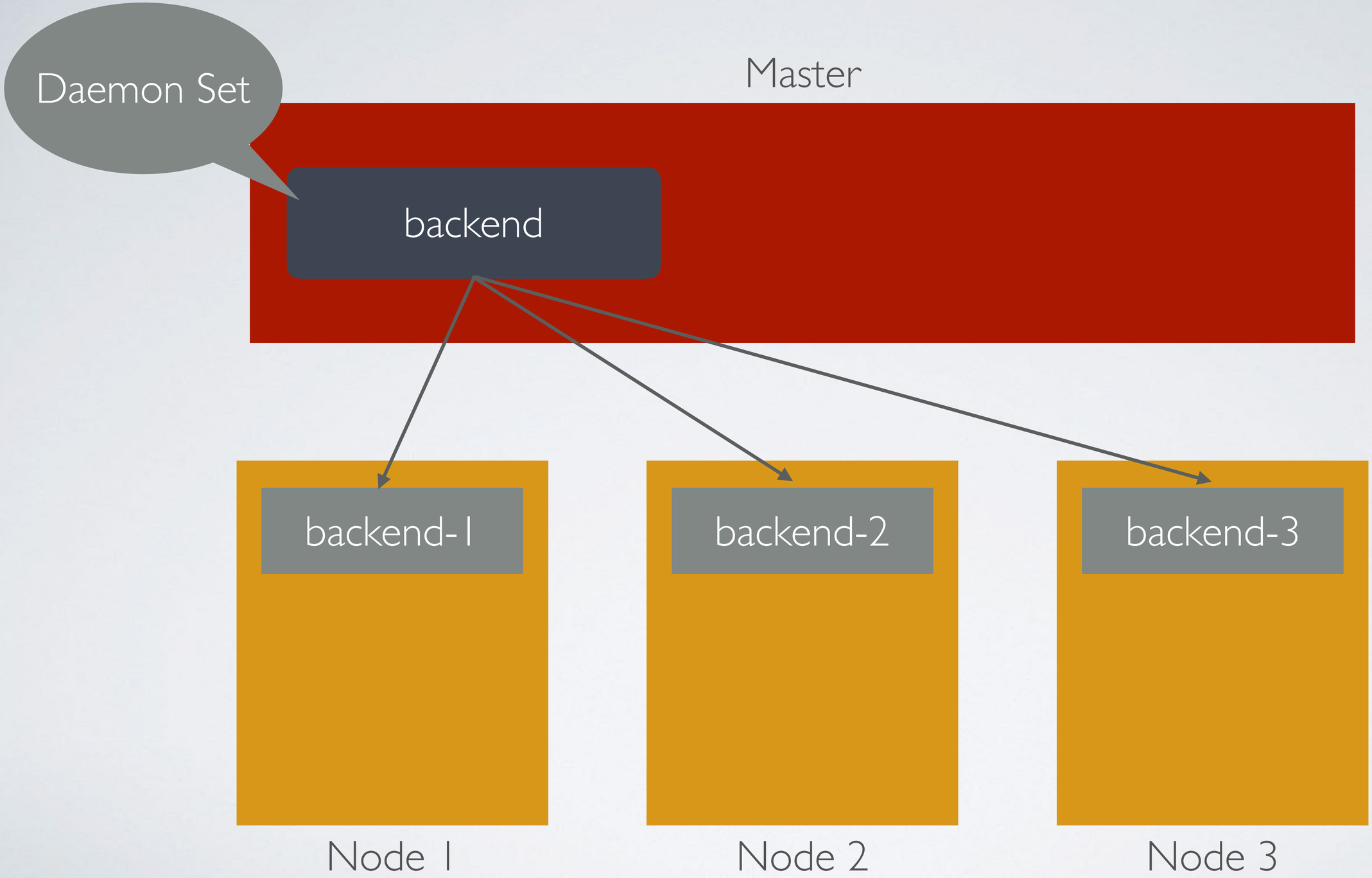
```
$ kubectl rollout undo deployment/echoserver --to-revision=1
deployment "echoserver" rolled back

$ kubectl rollout history deployment/echoserver
deployments "echoserver"
REVISION        CHANGE-CAUSE
3               kubectl set image deployment/echoserver echoserver=gcr.io/google-containers/echoserver:1.3
4               kubectl set image deployment/echoserver echoserver=gcr.io/google-containers/echoserver:1.2
5               kubectl create --filename=deployment.yaml --record=true
```

```
$ kubectl scale deployment/echoserver --replicas 6
deployment "echoserver" scaled

$ kubectl get deployment/echoserver
NAME         DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
echoserver   6         6         6            6           11m
```

รันพอร์ทตัวนี้ทุก node เพิ่มไม่ได้ เช่น watch google ใช้ log event

Daemon Sets (ds)

ensures that all (or some) nodes run a copy of a pod


Daemon Set
Master
backend
backend-1
backend-2
backend-3
Node 1
Node 2
Node 3

```yaml
kind: DaemonSet
apiVersion: extensions/v1beta1
metadata:
  name: echoserver
spec:
  template:
    metadata:
      labels:
        app: echoserver
    spec:
      containers:
      - name: echoserver
        image: gcr.io/google-containers/echoserver:1.6
        ports:
        - containerPort: 8080
  updateStrategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
```

Strategy

Default

- OnDelete — new DaemonSet pods will only be created when you manually delete old DaemonSet pods
- RollingUpdate

Resource Quotas (quota)

limit aggregate resource consumption

```yaml
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: echoserver
spec:
  replicas: 3
  revisionHistoryLimit: 2
  template:
    metadata:
      labels:
        app: echoserver
    spec:
      containers:
      - name: echoserver
        image: gcr.io/google-containers/echoserver:1.6
        ports:
        - containerPort: 8080
        resources:
          requests:
            cpu: 200m
            memory: 300Mi
          limits:
            cpu: 1
            memory: 1Gi
```

1 CPU = 1000m

จองล่วงหน้าได้แน่ๆ ต้องได้ไม่พอ pending

ใช้ห้ามกินเท่าไหร่ จากที่เหลือ เกินรีสตาร์ท

ต้องนิยามคำว่า health ของเราของตัวโปรแกรมเรา

Health Check

becrypt, argon

Health Check

ยังอยู่ไหม kub ยิง ว่าอยู่ไหม ไม่ตอบ restart

- Liveness Probes — know when to restart a Container
- Readiness Probes — don't send requests until application started

สำหรับ rollout

```yaml
kind: Deployment
apiVersion: app/v1beta1
metadata:
  name: default-http-backend
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: default-http-backend
    spec:
      containers:
      - name: default-http-backend
        image: gcr.io/google-containers/defaultbackend:1.3
        ports:
        - containerPort: 8080
        readinessProbe:
          httpGet:
            path: /healthz
            port: 8080
            scheme: HTTP
          initialDelaySeconds: 30
          timeoutSeconds: 5
          periodSeconds: 10
          successThreshold: 1
          failureThreshold: 3
        livenessProbe:
          httpGet:
            path: /healthz
            port: 8080
            scheme: HTTP
          initialDelaySeconds: 30
          timeoutSeconds: 5
          periodSeconds: 10
          successThreshold: 1
          failureThreshold: 3
```

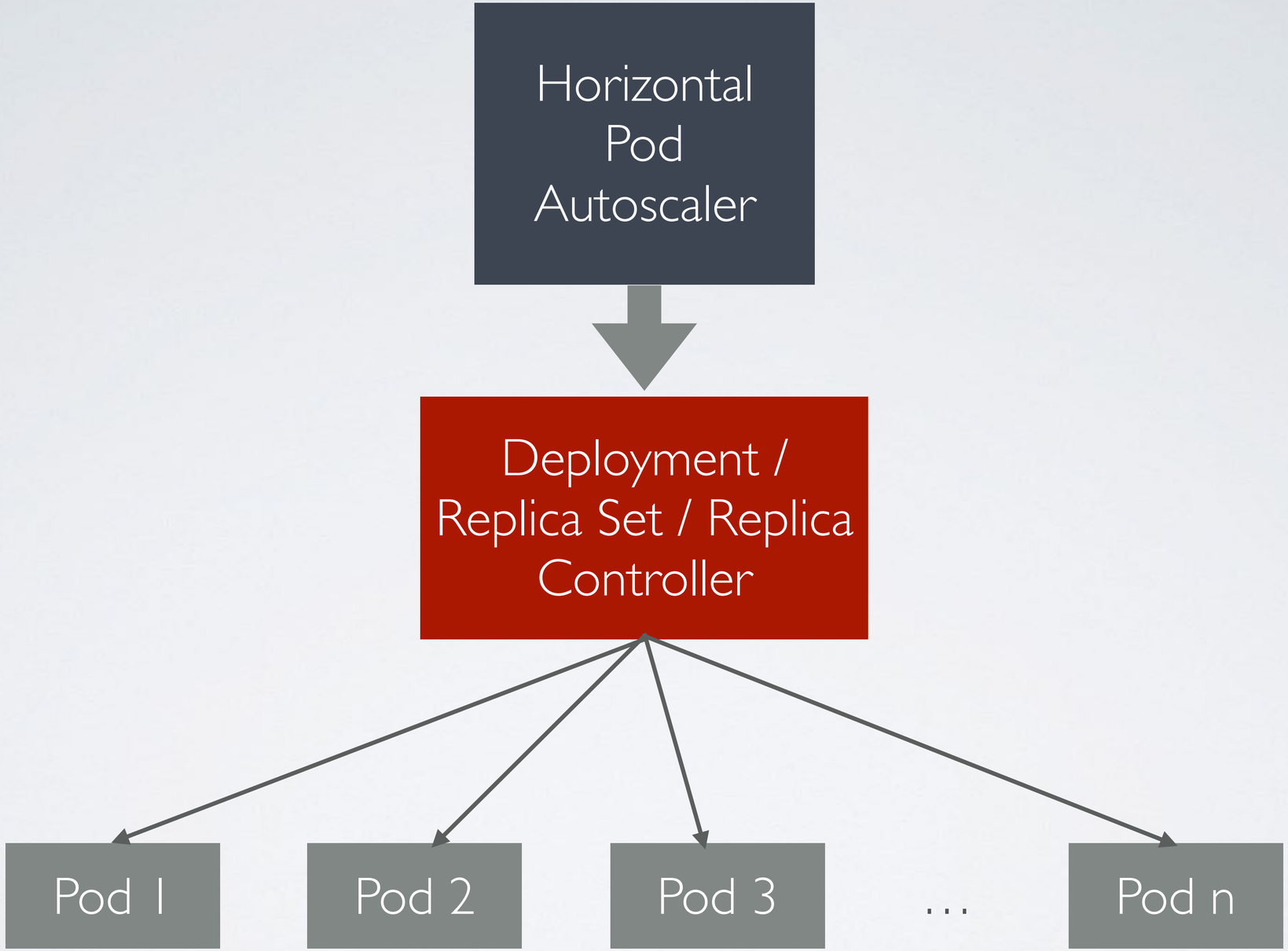
liveness นานกว่า readiness

จองล่วงหน้าได้แน่ๆ ต้องได้ไม่พอ pending

ต้อง 200 body ไม่สน

Horizontal Pod Autoscaler (hpa)

automatically scales the number of pods in
a replication controller, deployment or replica set


Horizontal Pod Autoscaler
Deployment / Replica Set / Replica Controller
Pod 1
Pod 2
Pod 3
…
Pod n

```
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: hpa-example
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: hpa-example
    spec:
      containers:
      - name: hpa-example
        image: gcr.io/google-containers/hpa-example
        ports:
        - containerPort: 80
        resources:
          requests:
            cpu: 100m
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: hpa-example
spec:
  selector:
    app: hpa-example
  ports:
  - port: 80
```

```yaml
kind: HorizontalPodAutoscaler
apiVersion: autoscaling/v1
metadata:
  name: hpa-example
spec:
  scaleTargetRef:
    apiVersion: apps/v1beta1
    kind: Deployment
    name: hpa-example
  minReplicas: 1
  maxReplicas: 6
  targetCPUUtilizationPercentage: 50
```

```
$ kubectl get hpa --watch
NAME          REFERENCE                     TARGETS     MINPODS   MAXPODS   REPLICAS   AGE
hpa-example   Deployment/hpa-example     0% / 50%   1         6         1          12m
hpa-example   Deployment/hpa-example   522% / 50%   1         6         1          12m
hpa-example   Deployment/hpa-example   522% / 50%   1         6         1          12m
hpa-example   Deployment/hpa-example   941% / 50%   1         6         1          13m
hpa-example   Deployment/hpa-example   941% / 50%   1         6         4          13m
hpa-example   Deployment/hpa-example   362% / 50%   1         6         4          14m
hpa-example   Deployment/hpa-example   362% / 50%   1         6         4          14m
hpa-example   Deployment/hpa-example    12% / 50%   1         6         4          15m
hpa-example   Deployment/hpa-example    12% / 50%   1         6         4          15m
hpa-example   Deployment/hpa-example     0% / 50%   1         6         4          16m
```

Auto-scale Node
on Container Engine

```
$ gcloud alpha container clusters update cluster-1 \
  --enable-autoscaling \
  --min-nodes=2 \
  --max-nodes=6 \
  --zone=asia-southeast1-b \
  --node-pool=default-pool \
  --project=acoshift-k8s
```

Name	default-pool
Current size	1
Node version	1.7.0
Node image	Container-Optimized OS (cos)
Machine type	n1-standard-1 (1 vCPU, 3.75 GB memory)
Total cores	1 vCPU
Total memory	3.75 GB
Automatic node upgrades	Disabled
Automatic node repair	Disabled

Automatic node upgrades (beta)

Disabled

Automatic node repair (beta)

Disabled

Autoscaling (beta)

On

Minimal size

1

Maximal size

6

Preemptible nodes	Disabled
Boot disk size in GB (per node)	100
Local SSD disks (per node)	0
Instance groups	gke-cluster-1-default-pool-73cdab92-grp

Labels

key/value pairs that are attached to objects

```yaml
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: echoserver
  labels:
    app: echoserver
spec:
  template:
    metadata:
      labels:
        app: echoserver
    spec:
      containers:
      - name: echoserver
        image: gcr.io/google-containers/echoserver:1.6
        ports:
        - containerPort: 8080
---
kind: Service
apiVersion: v1
metadata:
  name: echoserver
  labels:
    app: echoserver
spec:
  selector:
    app: echoserver
  ports:
  - port: 80
    targetPort: 8080
```

```
$ kubectl get all -l app=echoserver
NAME                                 READY     STATUS    RESTARTS   AGE
po/echoserver-3345770719-c5q61   1/1       Running   0          10s

NAME              CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
svc/echoserver   10.3.240.126   <none>        80/TCP    9s

NAME                DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
deploy/echoserver   1         1         1            1           10s

NAME                           DESIRED   CURRENT   READY     AGE
rs/echoserver-3345770719   1         1         1         10s
```

Node Selector

```
$ kubectl label no node-3 nodeType=highmem

$ kubectl get no --show-labels
```

```
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: redis
spec:
  template:
    metadata:
      labels:
        app: redis
    spec:
      containers:
      - name: redis
        image: redis:3.2.9
        ports:
        - containerPort: 6379
      nodeSelector:
        nodeType: highmem
```

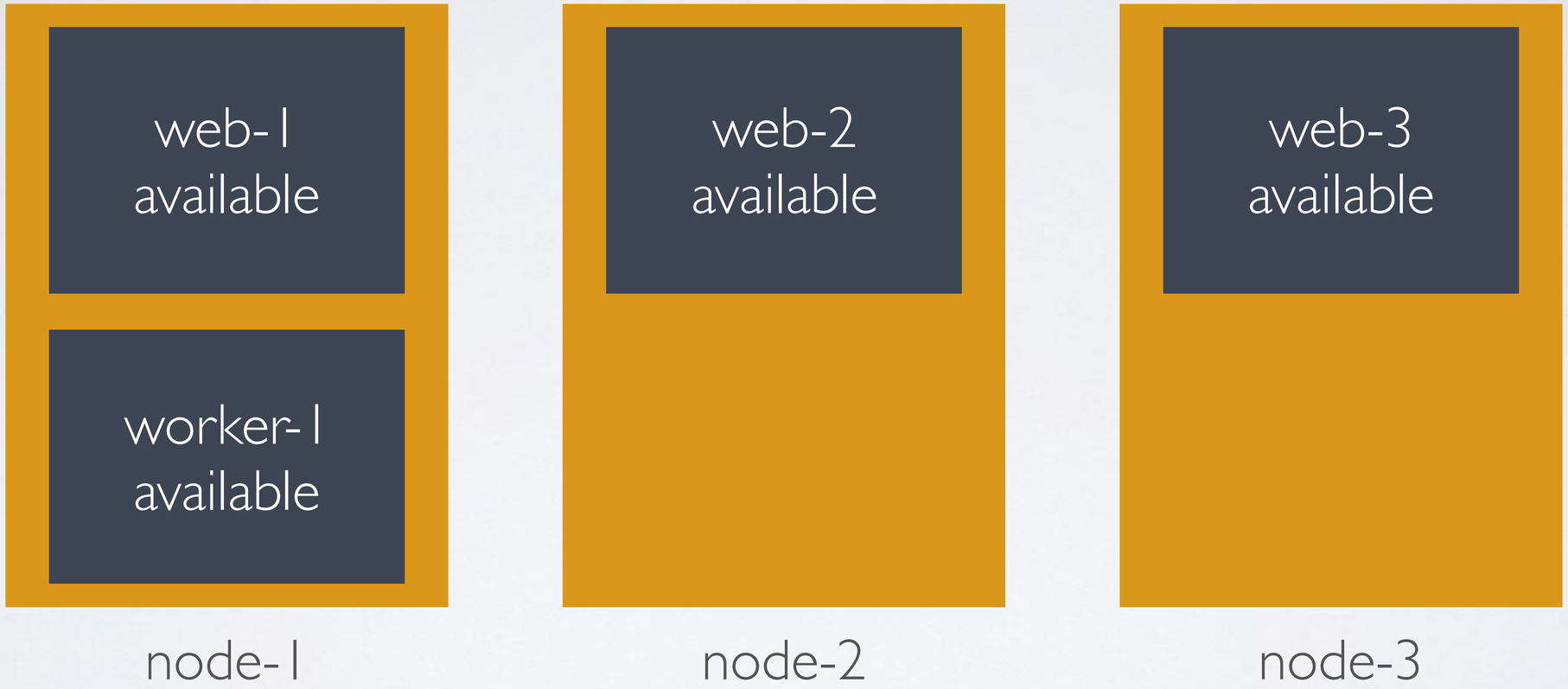
```
$ kubectl label no node-3 nodeType-
```

โดน Disrupt ได้เท่าไหร่ มันดาวน์ได้เท่าไหร่ รู้ก่อนร่วงหน้าเช่น เราจะอัพเดทโโหนด

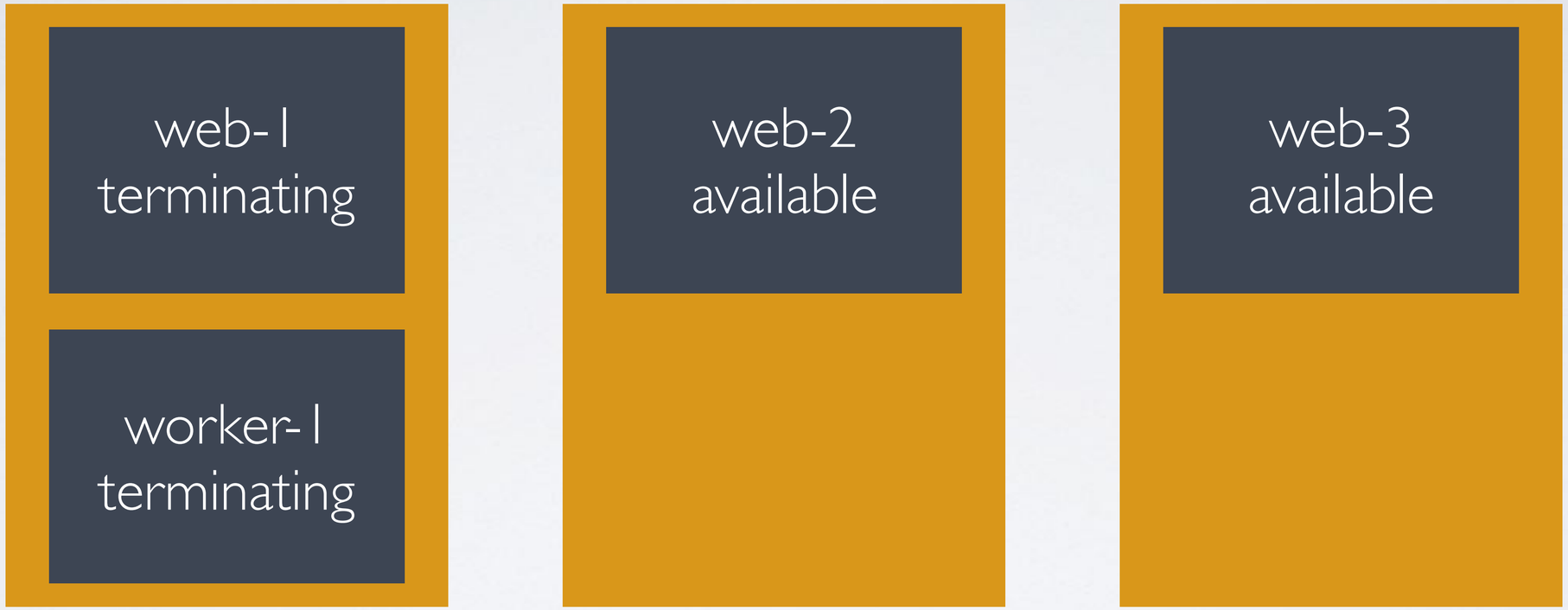
Pod Disruption Budgets (pdb)

limits the number pods of a replicated application
that are down simultaneously from voluntary disruptions

web: replicas=3, minAvailable=2



web: replicas=3, minAvailable=2



web: replicas=3, minAvailable=2

node-1

draining

node-2

node-3

web: replicas=3, minAvailable=2

web: replicas=3, minAvailable=2

web: replicas=3, minAvailable=2

node-1

drained

node-2

draining

node-3

web: replicas=3, minAvailable=2

web-5
pending

node-1

drained

node-2

drain blocked

node-3

PodDisruptionBudget

มีไว้เพื่อช่วยตอน update maintain ไม่ให้ล่ม drain เปิดปิดใหม่

- minAvailable
- maxUnavailable

config ได้อันใดอันนึง

```yaml
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: echoserver
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: echoserver
    spec:
      containers:
      - name: echoserver
        image: gcr.io/google-containers/echoserver:1.6
        ports:
        - containerPort: 8080
        resources:
          requests:
            cpu: 250m
```

```yaml
kind: PodDisruptionBudget
apiVersion: policy/v1beta1
metadata:
  name: echoserver
spec:
  minAvailable: 2
  # minAvailable: 67%
  selector:
    matchLabels:
      app: echoserver
```

```
$ kubectl get no
NAME                                     STATUS    AGE       VERSION
gke-cluster-2-default-pool-75546f17-39fq   Ready     10m       v1.7.0
gke-cluster-2-default-pool-75546f17-9dc5   Ready     10m       v1.7.0
gke-cluster-2-default-pool-75546f17-xr57   Ready     10m       v1.7.0

$ kubectl get po
NAME                           READY     STATUS    RESTARTS   AGE
echoserver-1994896057-4lk6s   1/1       Running   0          50s
echoserver-1994896057-f0rft   1/1       Running   0          51s
echoserver-1994896057-p0lpr   1/1       Running   0          42s

$ kubectl drain gke-cluster-2-default-pool-75546f17-39fq --force --ignore-daemonsets
node "gke-cluster-2-default-pool-75546f17-39fq" already cordoned
pod "kube-dns-autoscaler-3880103346-4qj3t" evicted
pod "kube-dns-1413379277-h96k0" evicted
node "gke-cluster-2-default-pool-75546f17-39fq" drained

$ kubectl get no
NAME                                     STATUS                     AGE       VERSION
gke-cluster-2-default-pool-75546f17-39fq   Ready,SchedulingDisabled   16m       v1.7.0
gke-cluster-2-default-pool-75546f17-9dc5   Ready                      16m       v1.7.0
gke-cluster-2-default-pool-75546f17-xr57   Ready                      16m       v1.7.0
```

```
$ kubectl drain gke-cluster-2-default-pool-75546f17-9dc5 --force --ignore-daemonsets
node "gke-cluster-2-default-pool-75546f17-9dc5" already cordoned
pod "heapster-v1.4.0-2764992688-5xp57" evicted
pod "kubernetes-dashboard-1962351010-2xtq1" evicted
pod "echoserver-1994896057-f0rft" evicted
pod "l7-default-backend-2954409777-d4x3g" evicted
pod "event-exporter-v0.1.4-1771975458-s86dg" evicted
# hang
pod "echoserver-1994896057-p0lpr" evicted
node "gke-cluster-2-default-pool-75546f17-9dc5" drained
$ kubectl get no
NAME                                       STATUS                     AGE       VERSION
gke-cluster-2-default-pool-75546f17-39fq   Ready,SchedulingDisabled   18m       v1.7.0
gke-cluster-2-default-pool-75546f17-9dc5   Ready,SchedulingDisabled   18m       v1.7.0
gke-cluster-2-default-pool-75546f17-xr57   Ready                      18m       v1.7.0

$ kubectl get po
NAME                           READY     STATUS    RESTARTS   AGE
echoserver-1994896057-1tg93   0/1       Pending   0          1m
echoserver-1994896057-1w5s7   1/1       Running   0          1m
echoserver-1994896057-4lk6s   1/1       Running   0          7m

$ kubectl describe po/echoserver-1994896057-1tg93
...
Events:
  FirstSeen LastSeen  Count From      SubObjectPath Type    Reason      Message
  --------- --------  ----- ----      ------------- --------  ------      -------
  1m    1m    5 default-scheduler     Warning   FailedScheduling  No nodes are available that match all of the
following predicates:: Insufficient cpu (1), PodToleratesNodeTaints (1).
...

$ kubectl uncordon gke-cluster-2-default-pool-75546f17-39fq
node "gke-cluster-2-default-pool-75546f17-39fq" uncordoned
```

```
$ kubectl get no
NAME                                     STATUS                        AGE       VERSION
gke-cluster-2-default-pool-75546f17-39fq   Ready                         22m       v1.7.0
gke-cluster-2-default-pool-75546f17-9dc5   Ready,SchedulingDisabled   22m       v1.7.0
gke-cluster-2-default-pool-75546f17-xr57   Ready                         22m       v1.7.0

$ kubectl get po
NAME                           READY     STATUS    RESTARTS   AGE
echoserver-1994896057-1tg93   1/1       Running   0          4m
echoserver-1994896057-1w5s7   1/1       Running   0          4m
echoserver-1994896057-4lk6s   1/1       Running   0          10m

$ kubectl uncordon gke-cluster-2-default-pool-75546f17-9dc5
node "gke-cluster-2-default-pool-75546f17-9dc5" uncordoned

$ kubectl get no
NAME                                     STATUS    AGE       VERSION
gke-cluster-2-default-pool-75546f17-39fq   Ready     25m       v1.7.0
gke-cluster-2-default-pool-75546f17-9dc5   Ready     25m       v1.7.0
gke-cluster-2-default-pool-75546f17-xr57   Ready     25m       v1.7.0
```

GCE Persistent Disks

Create Persistent Disk (pd) on GCP

```
$ gcloud compute disks create --size=20GB --zone=asia-southeast1-b --project=acoshift-k8s mysql-disk
Created [https://www.googleapis.com/compute/v1/projects/acoshift-k8s/zones/asia-southeast1-b/disks/mysql-disk].
NAME        ZONE               SIZE_GB  TYPE         STATUS
mysql-disk  asia-southeast1-b  20       pd-standard  READY
```

```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: mysql
spec:
  replicas: 1
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
      - name: mysql
        env:
        - name: MYSQL_ROOT_PASSWORD
          value: mysqlpassword1234
        image: mysql:5.6.36
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 3306
        volumeMounts:
        - mountPath: /var/lib/mysql
          name: mysql-disk
      volumes:
      - name: mysql-disk
        gcePersistentDisk:
          pdName: mysql-disk
          fsType: ext4
```

```
$ kubectl create -f pd.yaml
deployment "mysql" created

$ kubectl get po
NAME                     READY     STATUS    RESTARTS   AGE
mysql-1398320157-mgf6c   1/1       Running   0          3m

$ kubectl port-forward mysql-1398320157-mgf6c 3306:3306
```

```
$ mysql -u root -p -h 127.0.0.1

mysql> create database db1;
Query OK, 1 row affected (0.05 sec)

mysql> use db1;
Database changed

mysql> create table users (
    -> id int auto_increment,
    -> name varchar(255) not null,
    -> created_at timestamp not null default now(),
    -> primary key (id)
    -> );
Query OK, 0 rows affected (0.08 sec)

mysql> insert into users (name) values ('acoshift'), ('user1'), ('user2');
Query OK, 3 rows affected (0.08 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from users;
+----+----------+---------------------+
| id | name     | created_at          |
+----+----------+---------------------+
|  1 | acoshift | 2017-07-15 14:46:04 |
|  2 | user1    | 2017-07-15 14:46:04 |
|  3 | user2    | 2017-07-15 14:46:04 |
+----+----------+---------------------+
3 rows in set (0.03 sec)

mysql> exit
Bye
```

```
$ kubectl get po
NAME                        READY     STATUS    RESTARTS   AGE
mysql-1398320157-mgf6c      1/1       Running   0          19m

$ kubectl delete po/mysql-1398320157-mgf6c
pod "mysql-1398320157-mgf6c" deleted

$ kubectl get po
NAME                        READY     STATUS              RESTARTS   AGE
mysql-1398320157-d0scs      0/1       ContainerCreating   0          30s
```

Pods

Name	Status	Restarts	Age	CPU (cores)	Memory (bytes)
mysql-1398320157-d0scs	Waiting: Containe…	0	41 seconds	-	-

AttachVolume.Attach failed for volume "mysql-disk" : googleapi: Error 400: The disk resource 'projects/acoshift-k8s/zones/asia-southeast1-b/disks/mysql-disk' is already being used by 'projects/acoshift-k8s/zones/asia-southeast1-b/instances/gke-cluster-1-default-pool-73cdab92-hhk2'

```
$ kubectl get po
NAME                        READY     STATUS    RESTARTS   AGE
mysql-1398320157-d0scs      1/1       Running   0          6m

$ kubectl port-forward mysql-1398320157-d0scs 3306:3306
Forwarding from 127.0.0.1:3306 -> 3306
Forwarding from [::1]:3306 -> 3306

$ mysql -u root -p -h 127.0.0.1

mysql> use db1;
Database changed

mysql> select * from users;
+----+----------+---------------------+
| id | name     | created_at          |
+----+----------+---------------------+
|  1 | acoshift | 2017-07-15 14:46:04 |
|  2 | user1    | 2017-07-15 14:46:04 |
|  3 | user2    | 2017-07-15 14:46:04 |
+----+----------+---------------------+
3 rows in set (0.04 sec)

mysql> exit
Bye
```

```
$ kubectl delete -f pd.yaml
deployment "mysql" deleted

$ gcloud compute disks delete --zone=asia-southeast1-b --project=acoshift-k8s mysql-disk
The following disks will be deleted:
 - [mysql-disk] in [asia-southeast1-b]

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

Deleted [https://www.googleapis.com/compute/v1/projects/acoshift-k8s/zones/asia-southeast1-b/disks/mysql-disk].
```

Q&A

DAY 2

ปกติเราจะไม่สร้าง PD

Persistent Volumes (pv)

a piece of storage in the cluster that has been provisioned by an administrator

admin ของ cluster kubernates บอกมีเท่านี่

Persistent Volume Claims (pvc)

a request for storage by a user

คำขอ User อยากใช้ Disk นี่

Storage Classes

a way for administrators to describe the "classes" of storage they offer

Clsss ชื่อเราตั้งชื่อกลุ่ม

Provisioning

- Static เราขอมาก่อน แล้วบอก kubernates
- Dynamic

user ขอมาจะไปสร้างให้

เขียน plugin เข้าไปได้

```
$ kubectl get storageclass
NAME                 TYPE
standard (default)   kubernetes.io/gce-pd

$ kubectl describe storageclass standard
Name:            standard
IsDefaultClass: Yes
Annotations:     storageclass.beta.kubernetes.io/is-default-class=true
Provisioner:     kubernetes.io/gce-pd
Parameters:      type=pd-standard
Events:          <none>
```

ถ้าเราขอแล้วไม่ได้ใส่ชื่อก็จะเอา class นี้

```yaml
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: ssd
provisioner: kubernetes.io/gce-pd
parameters:
  # type: pd-standard
  type: pd-ssd
  zone: asia-southeast1-b
```

parameter ของ google cloud

```
$ kubectl create -f 01-storageclass.yaml
storageclass "ssd" created

$ kubectl get storageclass
NAME                 TYPE
ssd                  kubernetes.io/gce-pd
standard (default)   kubernetes.io/gce-pd
```

Access Modes

- ReadWriteOnce — the volume can be mounted as read-write by a single node

node,pod mount ได้ node เดียว

- ReadOnlyMany — the volume can be mounted read-only by many nodes

- ReadWriteMany — the volume can be mounted as read-write by many nodes

storage ของเราต้อง support NFS, หลายๆที่ Network Disk

Volume Plugin	ReadWriteOnce	ReadOnlyMany	ReadWriteMany
AWSElasticBlockStore	✓	-	-
AzureFile	✓	✓	✓
AzureDisk	✓	-	-
CephFS	✓	✓	✓
Cinder	✓	-	-
FC	✓	✓	-
FlexVolume	✓	✓	-
Flocker	✓	-	-
GCEPersistentDisk	✓	✓	-
Glusterfs	✓	✓	✓
HostPath	✓	-	-
iSCSI	✓	✓	-
PhotonPersistentDisk	✓	-	-
Quobyte	✓	✓	✓
NFS	✓	✓	✓
RBD	✓	✓	-
VsphereVolume	✓	-	-
PortworxVolume	✓	-	✓
ScaleIO	✓	✓	-
StorageOS	✓	-	-

network disk scale disk พวก stateful (wordpress)

https://kubernetes.io/docs/concepts/storage/persistent-volumes/

Reclaim Policy

Text

Default for static provisioning

- Retain
- Recycle
- Delete

Network Disk

Default for dynamic provisioning

ลบ PVC ปีบหายเลย

```yaml
kind: PersistentVolume
apiVersion: v1
metadata:
  name: disk-1
  annotations:
    volume.beta.kubernetes.io/mount-options: discard
spec:
  storageClassName: standard
  capacity:
    storage: 10Gi
  accessModes:
  - ReadWriteOnce
  gcePersistentDisk:
    fsType: ext4
    pdName: disk-1
```

สร้าง Disk ก่อน ขนาด 10 gb ไปที่เว็ปสร้าง

```yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: mysql-pvc
spec:
  storageClassName: standard
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
```

ขอ 5 มี 10 ได้ 10 ถ้าไม่พอสร้างใหม่เลย

```
$ gcloud compute disks create --size=10GB --zone=asia-southeast1-b --project=acoshift-k8s disk-1
Created [https://www.googleapis.com/compute/v1/projects/acoshift-k8s/zones/asia-southeast1-b/disks/disk-1].
NAME    ZONE               SIZE_GB  TYPE         STATUS
disk-1  asia-southeast1-b  10       pd-standard  READY

$ kubectl create -f 02-pv.yaml
persistentvolume "disk-1" created

$ kubectl get pv
NAME      CAPACITY   ACCESSMODES   RECLAIMPOLICY   STATUS      CLAIM     STORAGECLASS   REASON    AGE
disk-1    10Gi       RWO           Retain          Available                                      12s

$ kubectl create -f 02-pvc.yaml
persistentvolumeclaim "mysql-pvc" created

$ kubectl get pv
NAME      CAPACITY   ACCESSMODES   RECLAIMPOLICY   STATUS    CLAIM                 STORAGECLASS   REASON    AGE
disk-1    10Gi       RWO           Retain          Bound     default/mysql-pvc     standard                 1m

$ kubectl get pvc
NAME        STATUS    VOLUME    CAPACITY   ACCESSMODES   STORAGECLASS   AGE
mysql-pvc   Bound     disk-1    10Gi       RWO           standard       34s
```

```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: mysql
spec:
  replicas: 1
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
      - name: mysql
        env:
        - name: MYSQL_ROOT_PASSWORD
          value: mysqlpassword1234
        image: mysql:5.6.36
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 3306
        volumeMounts:
        - mountPath: /var/lib/mysql
          name: mysql-disk
      volumes:
      - name: mysql-disk
        persistentVolumeClaim:
          claimName: mysql-pvc
```

```
$ kubectl create -f 02-mysql.yaml
deployment "mysql" created

$ kubectl get po
NAME                     READY     STATUS    RESTARTS   AGE
mysql-68058648-d3m8l     1/1       Running   0          48s

$ kubectl delete po/mysql-68058648-d3m8l
pod "mysql-68058648-d3m8l" deleted

$ kubectl get po
NAME                     READY     STATUS    RESTARTS   AGE
mysql-68058648-rsqk1     1/1       Running   0          4s
```

```
$ kubectl delete deploy/mysql pvc/mysql-pvc pv/disk-1
deployment "mysql" deleted
persistentvolumeclaim "mysql-pvc" deleted
persistentvolume "disk-1" deleted

$ gcloud compute disks delete --zone=asia-southeast1-b --project=acoshift-k8s disk-1
The following disks will be deleted:
 - [disk-1] in [asia-southeast1-b]

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

Deleted [https://www.googleapis.com/compute/v1/projects/acoshift-k8s/zones/asia-
southeast1-b/disks/disk-1].
```

ใช้สำหรับโปรแกรมสำหรับ stateful ไม่มีใครคุมมัน

Stateful Sets

provides guarantees about the ordering of deployment and scaling

Stateful Sets

- Stable, unique network identifier รู้ชื่อ pod เลย
- Stable, persistent storage
  ถ้าต้องการ Disk
  เข้าข้อใดข้อนี้
  Deployment ไม่ควรมี disk
- Ordered, graceful deployment and scaling
  scale เปิดทีละตัว
- Ordered, graceful deletion and termination
  ลบตาม Order
- Ordered, automated rolling updates
  Rolling Update เป็น Order

terminationGracePeriodSeconds: 60 เกิน 60 ไม่ปิด kill ทิ้งเลย ค่าต้องมาก

gratful ชัดดาว

```
$ kubectl create -f https://raw.githubusercontent.com/cockroachdb/cockroach/master/cloud/kubernetes/
cockroachdb-statefulset.yaml
service "cockroachdb-public" created
service "cockroachdb" created
poddisruptionbudget "cockroachdb-budget" created
statefulset "cockroachdb" created

$ kubectl get po
NAME             READY     STATUS    RESTARTS   AGE
cockroachdb-0    1/1       Running   0          10m
cockroachdb-1    1/1       Running   0          9m
cockroachdb-2    1/1       Running   0          8m

$ kubectl port-forward cockroachdb-0 8080
```

RBAC => Role-Based

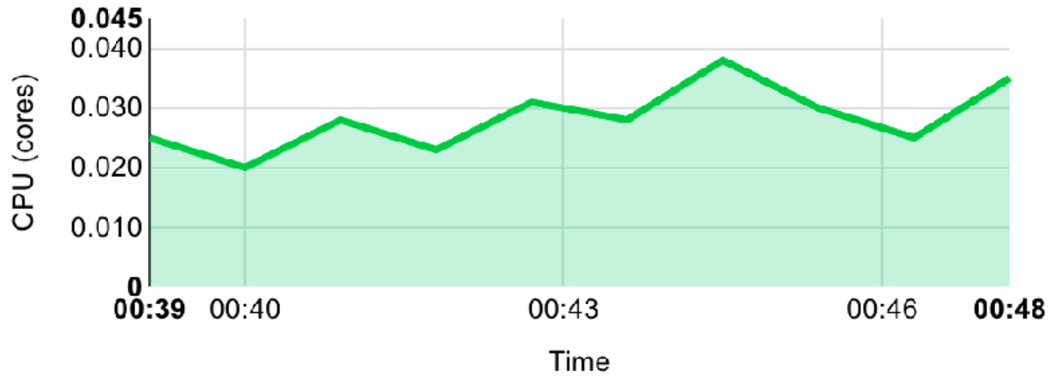
imagePullPolicy: IfNotPresent, Away (ดึงใหม่ตลอด)

Affinity -> Pod ต้องให้ลง node เดียวกัน

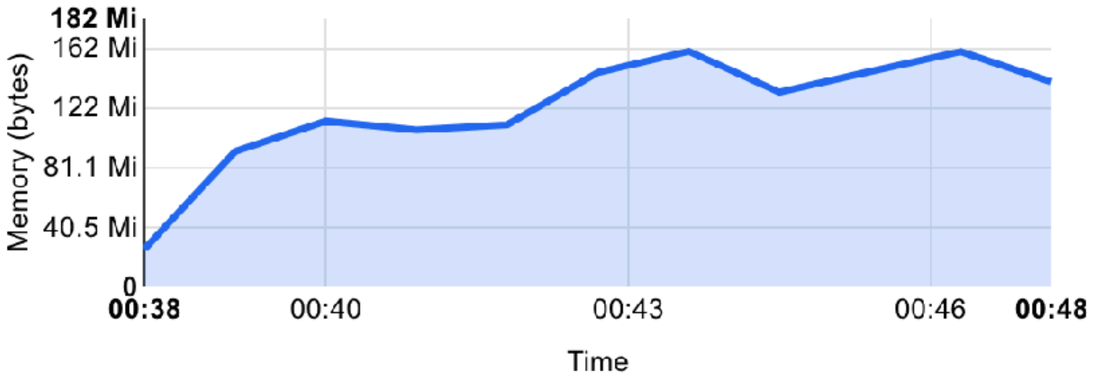
Live Nodes

ID ▾	ADDRESS ▾	UPTIME ▾	BYTES ▾	REPLICAS ▾	MEM USAGE ▾	LOGS
1	• cockroachdb-0.cockroachdb.default.svc.cluster.local:26257	10 minutes	3.2 MiB	10	95.6 MiB	Logs
2	• cockroachdb-1.cockroachdb.default.svc.cluster.local:26257	9 minutes	3.3 MiB	10	71.5 MiB	Logs
3	• cockroachdb-2.cockroachdb.default.svc.cluster.local:26257	9 minutes	3.3 MiB	10	70.4 MiB	Logs

CPU usage



Memory usage



Pods

Name	Status	Restarts	Age	CPU (cores)	Memory (bytes)
cockroachdb-4	Running	0	52 seconds	-	-
cockroachdb-3	Running	0	a minute	-	-
cockroachdb-2	Running	0	11 minutes	0.009	35.863 Mi
cockroachdb-1	Running	0	11 minutes	0.011	37.871 Mi
cockroachdb-0	Running	0	13 minutes	0.015	65.488 Mi

Live Nodes

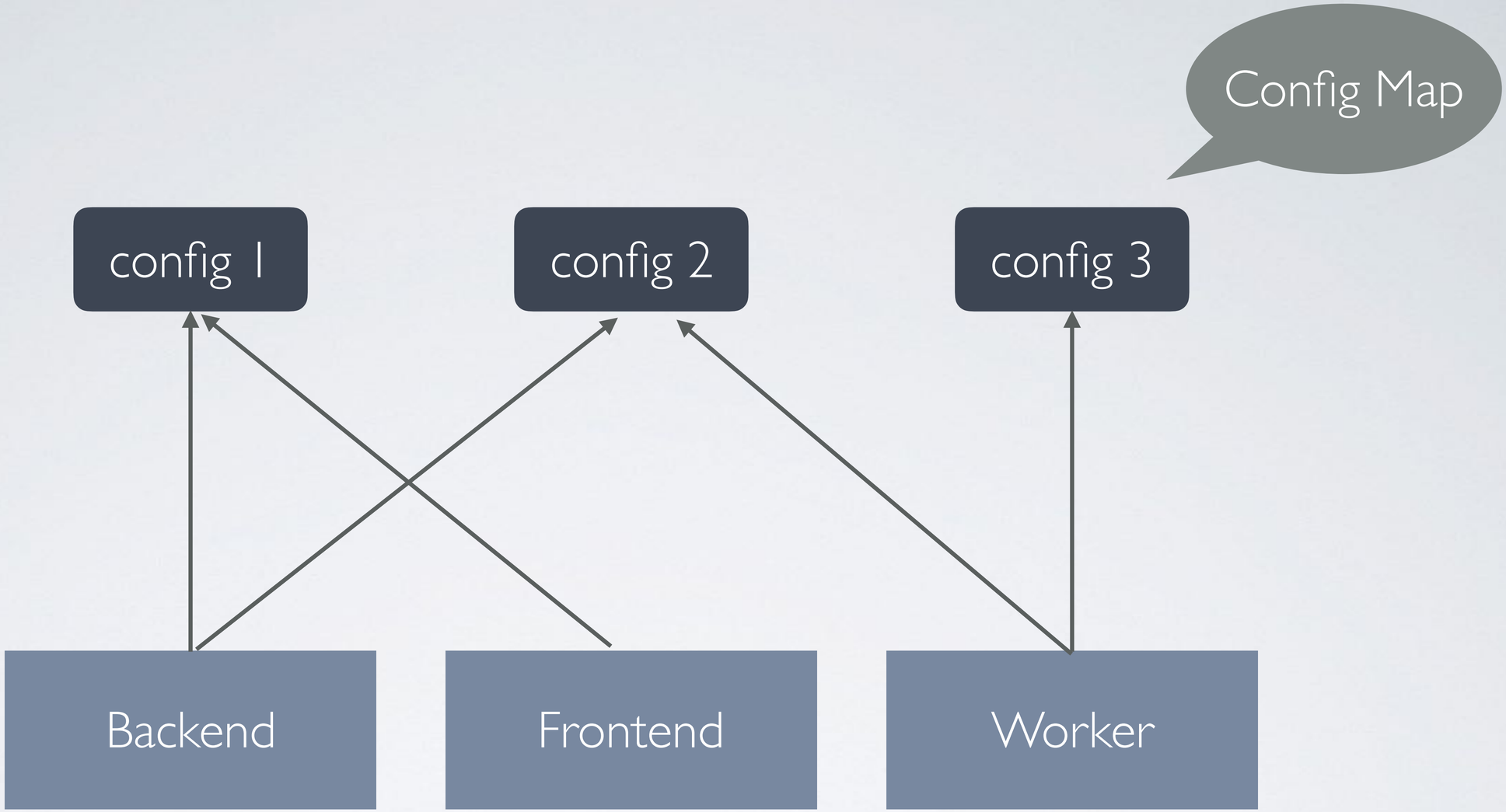
ID ▾	ADDRESS ▾	UPTIME ▾	BYTES ▾	REPLICAS ▾	MEM USAGE ▾	LOGS
1	• cockroachdb-0.cockroachdb.default.svc.cluster.local:26257	12 minutes	4.2 MiB	6	101.6 MiB	Logs
2	• cockroachdb-1.cockroachdb.default.svc.cluster.local:26257	11 minutes	84.4 KiB	7	73.5 MiB	Logs
3	• cockroachdb-2.cockroachdb.default.svc.cluster.local:26257	11 minutes	111.2 KiB	6	73.6 MiB	Logs
4	• cockroachdb-3.cockroachdb.default.svc.cluster.local:26257	a minute	4.1 MiB	5	60.4 MiB	Logs
5	• cockroachdb-4.cockroachdb.default.svc.cluster.local:26257	a minute	4.1 MiB	6	60.5 MiB	Logs

```
$ kubectl run -it --rm cockroach-client --image=cockroachdb/cockroach --restart=Never --command -- ./cockroach sql --host
cockroachdb-public --insecure

root@cockroachdb-public:26257/> create database db1;
CREATE DATABASE

root@cockroachdb-public:26257/> set database = db1;
SET

root@cockroachdb-public:26257/db1> create table users (
                                -> id serial,
                                -> name string not null default '',
                                -> created_at timestamp not null default now(),
                                -> primary key (id)
                                -> );
CREATE TABLE

root@cockroachdb-public:26257/db1> insert into users (name) values ('acoshift'), ('user1'), ('user2');
INSERT 3

root@cockroachdb-public:26257/db1> select * from users;
+------------------+----------+----------------------------------+
|        id        |   name   |            created_at            |
+------------------+----------+----------------------------------+
| 262376372306051076 | acoshift | 2017-07-15 17:55:14.366042+00:00 |
| 262376372306247684 | user1    | 2017-07-15 17:55:14.366042+00:00 |
| 262376372306345988 | user2    | 2017-07-15 17:55:14.366042+00:00 |
+------------------+----------+----------------------------------+
(3 rows)
```

Config Maps (cm)

decouple configuration artifacts from image content
to keep containerized applications portable


Backend
config 1
config 2
Frontend
config 1
Worker
config 2
config 3


Config Map
config 1
config 2
config 3
Backend
Frontend
Worker

```yaml
kind: ConfigMap
apiVersion: v1
metadata:
  name: redis-config
data:
  redis.conf: |
    databases 1
    save ""
    appendonly no
    maxmemory 2mb
    maxmemory-policy allkeys-lru
---
kind: Service
apiVersion: v1
metadata:
  name: redis
spec:
  selector:
    app: redis
  ports:
  - port: 6379
```

z

```yaml
kind: StatefulSet
apiVersion: apps/v1beta1
metadata:
  name: redis
spec:
  serviceName: redis
  replicas: 1
  template:
    metadata:
      labels:
        app: redis
    spec:
      containers:
      - name: redis
        image: redis:3.2.9
        ports:
        - containerPort: 6379
        volumeMounts:
        - mountPath: /usr/local/etc/redis
          name: config
        command:
        - redis-server
        - /usr/local/etc/redis/redis.conf
      volumes:
      - name: config
        configMap:
          name: redis-config
          items:
          - key: redis.conf
            path: redis.conf
```

ท่า Volume

```
$ kubectl create -f cm.yaml
configmap "redis-config" created
service "redis" created
statefulset "redis" created

$ kubectl run -it --rm redis-client --image=redis --restart=Never --command -- bash
root@redis-client:/data# redis-cli -h redis

redis:6379> config get save
1) "save"
2) ""
```

ConfigMaps ที่อ่านได้แค่บางคน

respect access control and are not visible to users without read permissions

Secrets

hold sensitive information

แปลงเป็น base64

```
$ echo -n "mysqlpassword" | base64
bXlzcWxwYXNzd29yZA==
```

```yaml
kind: Secret
apiVersion: v1
metadata:
  name: mysql
data:
  password: bXlzcWxwYXNzd29yZA==
```

base64 ถอดเองตอน mount

```yaml
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: mysql
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
      - name: mysql
        env:
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql
              key: password
        image: mysql:5.6.36
        ports:
        - containerPort: 3306
```

ท่า set environment

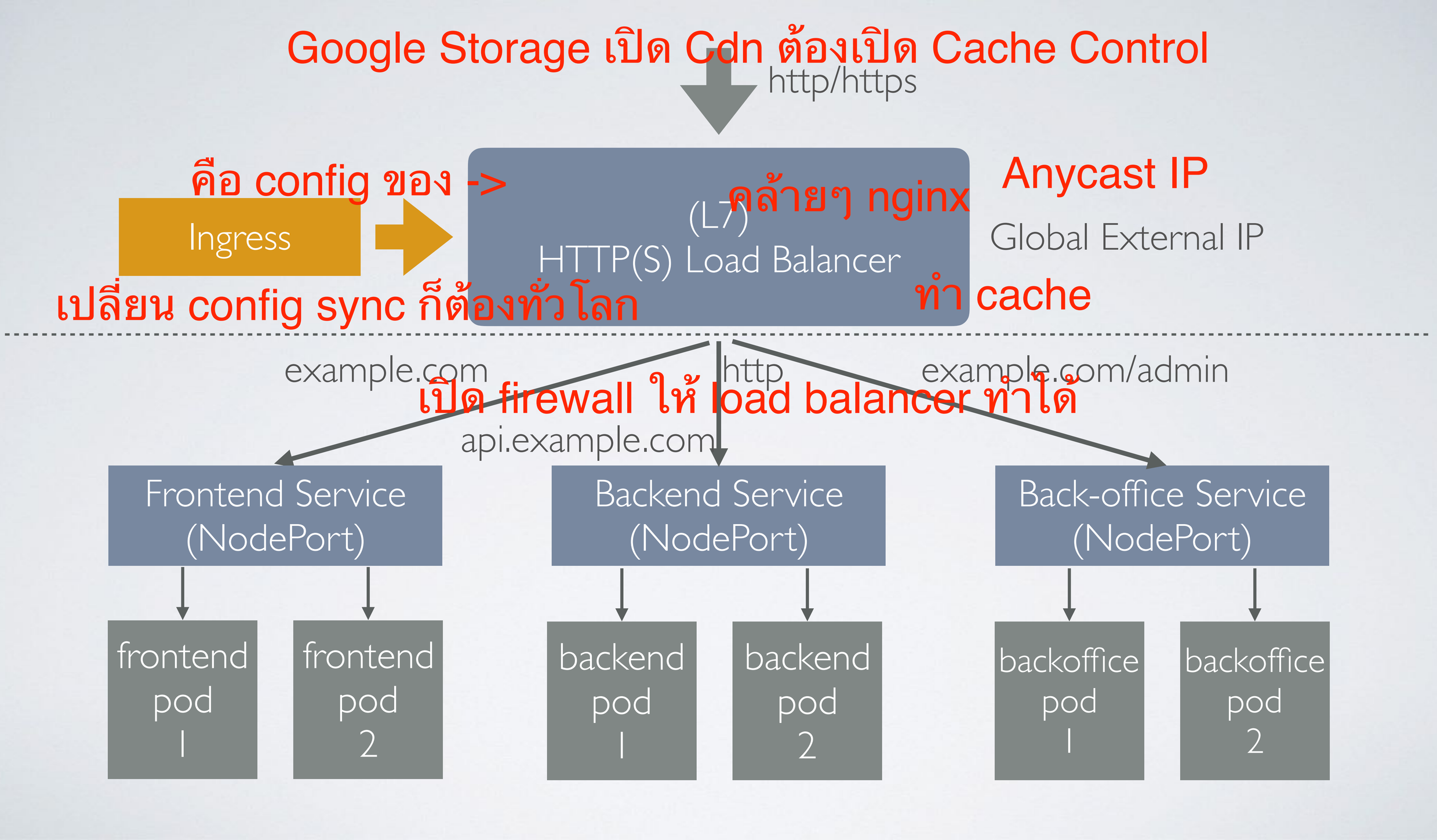
```
$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out tls.crt -subj "/CN=echoserver-secure.acoshift.me"
Generating a 2048 bit RSA private key
.....................................+++
...+++
writing new private key to 'tls.key'
-----

$ kubectl create secret tls echoserver-secure-acoshift-me-tls --key=tls.key --cert=tls.crt
secret "echoserver-secure-acoshift-me-tls" created
```

load balance level 7

Ingresses (ing)

a collection of rules that allow inbound connections to reach the cluster services

Google Cloud
HTTP(S) Load Balancer


Google Storage เปิด Cdn ต้องเปิด Cache Control
http/https
คือ config ของ ->
Ingress
(L7)
HTTP(S) Load Balancer
คล้ายๆ nginx
Anycast IP
Global External IP
เปลี่ยน config sync ก็ต้องทั่วโลก
ทำ cache
example.com
http
example.com/admin
เปิด firewall ให้ load balancer ทำได้
api.example.com
Frontend Service
(NodePort)
Backend Service
(NodePort)
Back-office Service
(NodePort)
frontend pod 1
frontend pod 2
backend pod 1
backend pod 2
backoffice pod 1
backoffice pod 2

```yaml
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: echoserver
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: echoserver
    spec:
      containers:
      - name: echoserver
        image: gcr.io/google-containers/echoserver:1.6
        ports:
        - containerPort: 8080
        readinessProbe:
          httpGet:
            path: /
            port: 8080
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: echoserver
spec:
  type: NodePort
  selector:
    app: echoserver
  ports:
  - port: 8080
```

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: gce
  name: gce-ingress
spec:
  rules:
  - host: echoserver-secure.acoshift.me
    http:
      paths:
      - backend:
          serviceName: echoserver
          servicePort: 8080
        path: /*
  tls:
  - secretName: echoserver-secure-acoshift-me-tls
    hosts:
    - echoserver-secure.acoshift.me
```

สำหรับ google cloud compute engine

ไม่ต้องใส่ถ้ามี ingress config ตัวเดียว

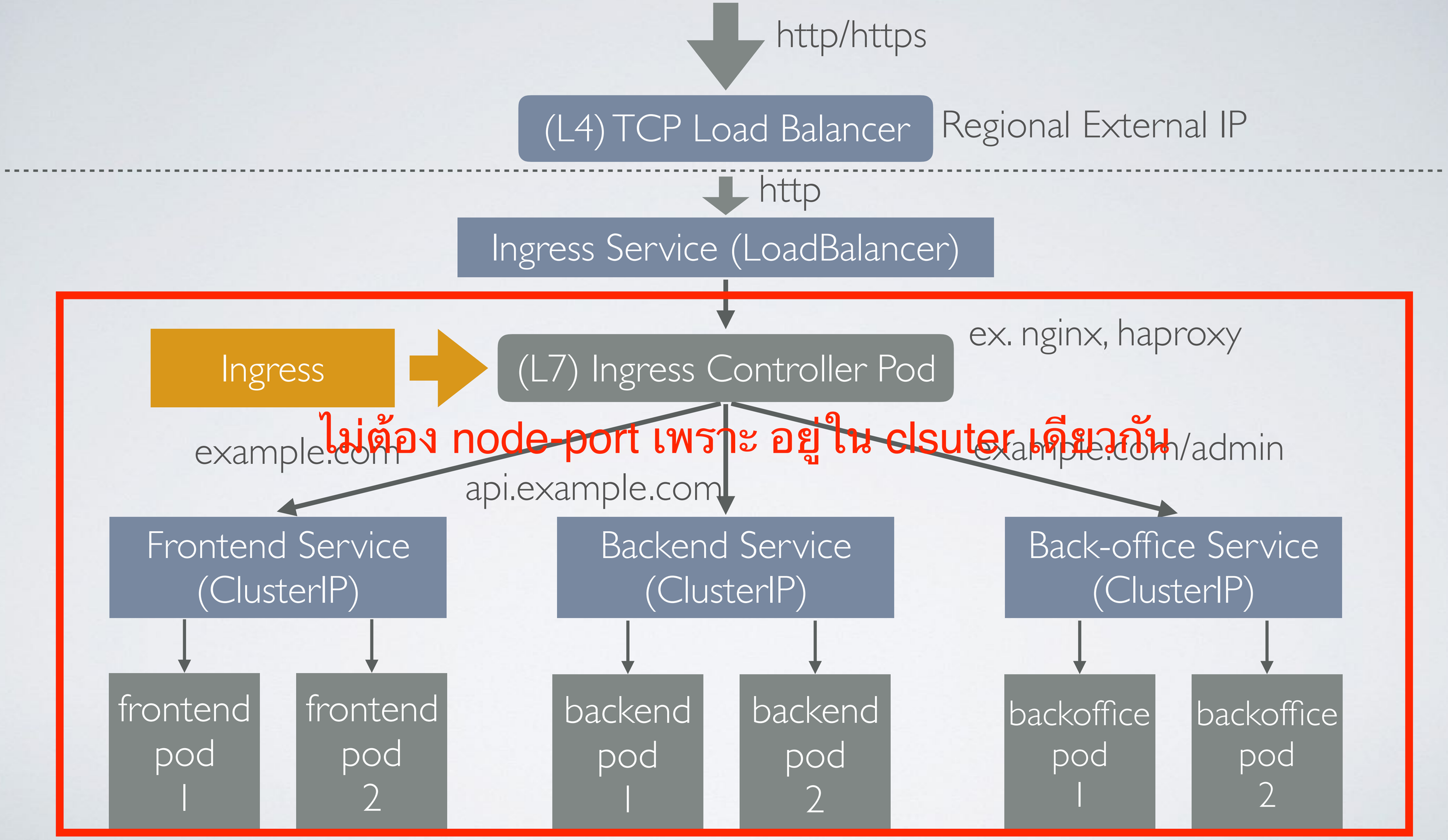
key value ใส่ comments ได้

* เฉพาะ gce

google อ่านแค่ secret ตัวเดียว

nginx config ด้วย yaml ยิงเทสก่อน ยิง reload ให้ด้วย แค่ edit config map

Nginx Ingress Controller


http/https
(L4) TCP Load Balancer
Regional External IP
http
Ingress Service (LoadBalancer)
Ingress
(L7) Ingress Controller Pod
ex. nginx, haproxy
ไม่ต้อง node-port เพราะ อยู่ใน clsuter เดียวกัน
example.com
api.example.com
example.com/admin
Frontend Service (ClusterIP)
Backend Service (ClusterIP)
Back-office Service (ClusterIP)
frontend pod 1
frontend pod 2
backend pod 1
backend pod 2
backoffice pod 1
backoffice pod 2

```yaml
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: echoserver
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: echoserver
    spec:
      containers:
      - name: echoserver
        image: gcr.io/google-containers/echoserver:1.6
        ports:
        - containerPort: 8080
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: echoserver
spec:
  ports:
  - port: 80
    targetPort: 8080
  selector:
    app: echoserver
```

```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: default-http-backend
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: default-http-backend
    spec:
      containers:
      - name: default-http-backend
        image: gcr.io/google-containers/defaultbackend:1.0
        livenessProbe:
          httpGet:
            path: /healthz
            port: 8080
            scheme: HTTP
          initialDelaySeconds: 30
          timeoutSeconds: 5
        ports:
        - containerPort: 8080
        resources:
          limits:
            cpu: 10m
            memory: 20Mi
          requests:
            cpu: 10m
            memory: 20Mi
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: default-http-backend
spec:
  ports:
  - port: 80
    targetPort: 8080
  selector:
    app: default-http-backend
```

```yaml
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-config
data:
  client-max-body-size: 20m
  hsts: "false"
  keep-alive: "30"
  proxy-body-size: 20m
  server-tokens: "false"
  use-gzip: "true"
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-ingress
spec:
  type: LoadBalancer
  selector:
    app: nginx-ingress
  ports:
  - name: http
    port: 80
  - name: https
    port: 443
```

```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: nginx-ingress
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: nginx-ingress
    spec:
      containers:
      - name: nginx-ingress-controller
        image: gcr.io/google-containers/nginx-ingress-controller:0.9.0-beta.10
        imagePullPolicy: Always
        ports:
        - containerPort: 80
        - containerPort: 443
        env:
        - name: POD_NAME
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: metadata.name
        - name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: metadata.namespace
        args:
        - /nginx-ingress-controller
        - --default-backend-service=$(POD_NAMESPACE)/default-http-backend
        - --configmap=$(POD_NAMESPACE)/nginx-config
        - --publish-service=$(POD_NAMESPACE)/nginx-ingress
        livenessProbe:
          failureThreshold: 3
          httpGet:
            path: /healthz
            port: 10254
            scheme: HTTP
          initialDelaySeconds: 10
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 5
        readinessProbe:
          failureThreshold: 3
          httpGet:
            path: /healthz
            port: 10254
            scheme: HTTP
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 1
```

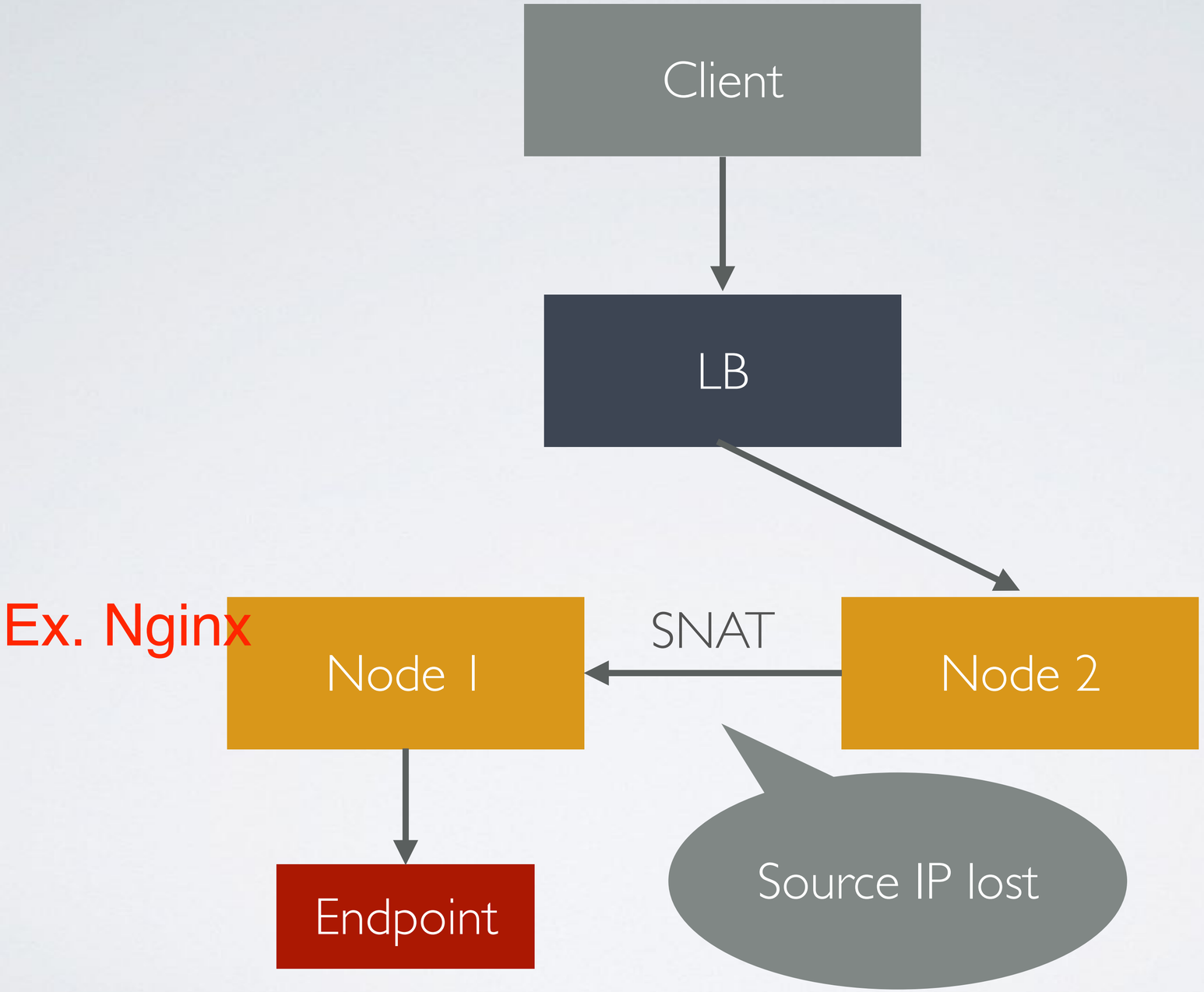
```yaml
kind: Ingress
apiVersion: extensions/v1beta1
metadata:
  name: nginx-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  rules:
  - host: echoserver-secure.acoshift.me
    http:
      paths:
      - path: /
        backend:
          serviceName: echoserver
          servicePort: 80
  - host: echoserver.acoshift.me
    http:
      paths:
      - path: /
        backend:
          serviceName: echoserver
          servicePort: 80
  tls:
  - secretName: echoserver-secure-acoshift-me-tls
    hosts:
    - echoserver-secure.acoshift.me
```

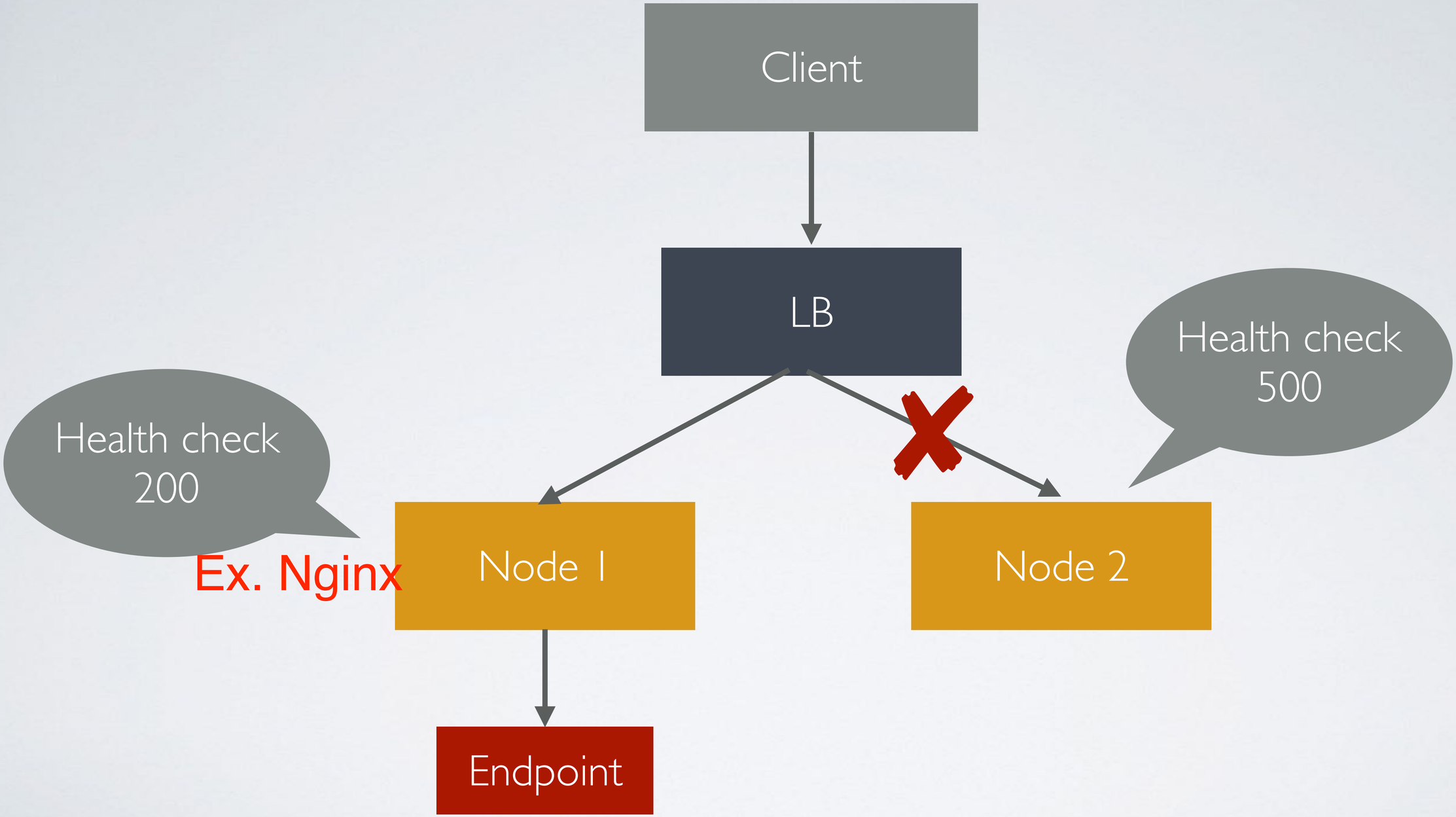
External Traffic Policy

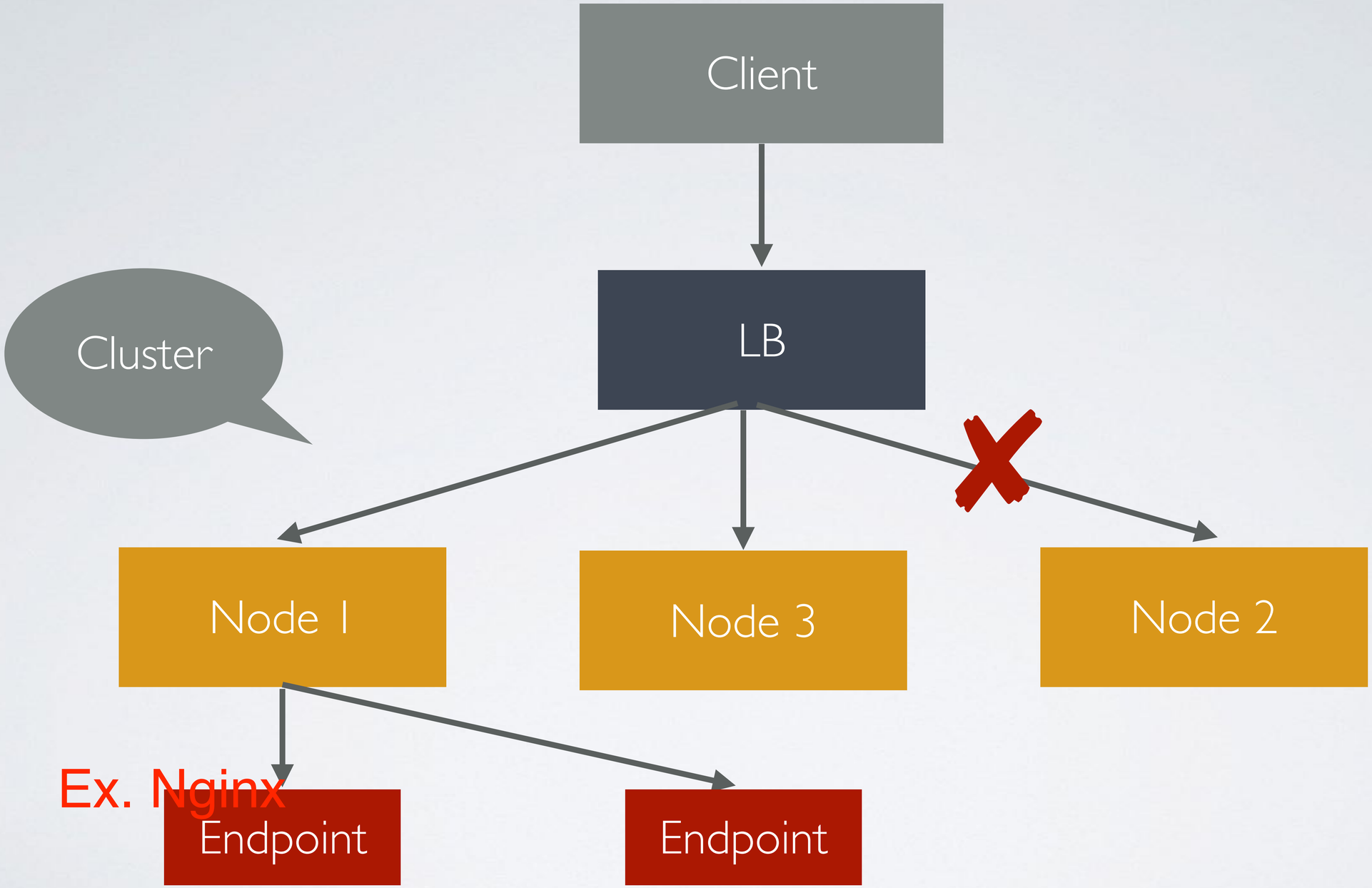
- Cluster

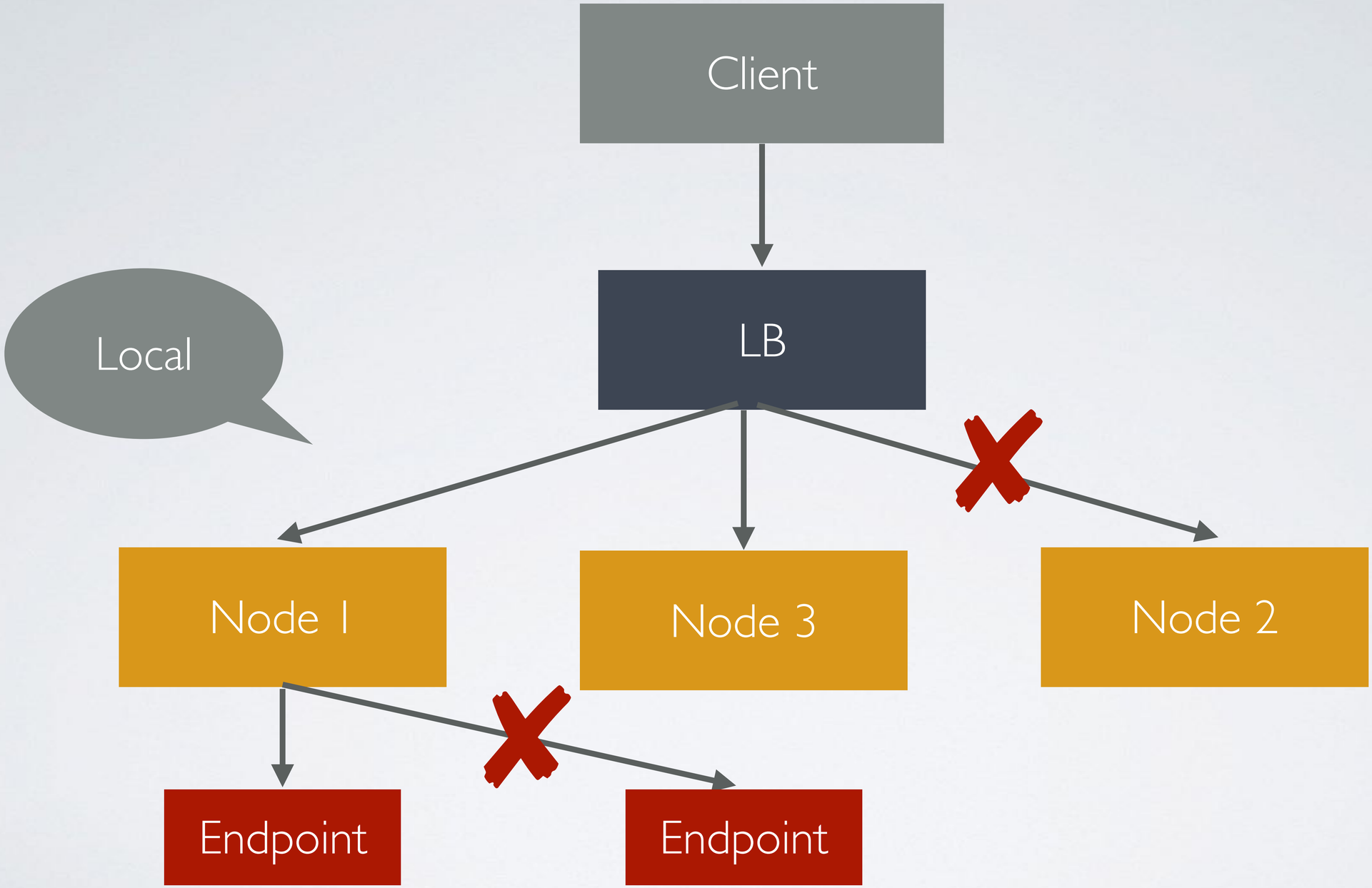
Default

- Local ถ้า nginx เปล่าๆ


Client
LB
Ex. Nginx
Node 1
SNAT
Node 2
Endpoint
Source IP lost


Client
LB
Health check
500
Health check
200
Ex. Nginx
Node 1
Node 2
Endpoint


Client
LB
Cluster
Node 1
Node 3
Node 2
Ex. Nginx
Endpoint
Endpoint


Client
LB
Local
Node 1
Node 3
Node 2
Endpoint
Endpoint

```
kind: Service
apiVersion: v1
metadata:
  name: nginx-ingress
spec:
  type: LoadBalancer
  externalTrafficPolicy: Local
  selector:
    app: nginx-ingress
...
```

a46023bb4699311e7a0bf42010a94002

Frontend

Protocol	IP:Port
TCP	35.187.231.192:80-443

Backend

Name: **a46023bb4699311e7a0bf42010a94002** Region: **asia-southeast1** Session affinity: **None** Health check: **a46023bb4699311e7a0bf42010a94002**

Instances	35.187.231.192
gke-cluster-1-default-pool-73cdab92-hhk2	
gke-cluster-1-default-pool-73cdab92-k7np	

ใช้ง่ายมาก
k8s<=1.8

kube-lego

automatically requests certificates for Kubernetes Ingress resources from Let's Encrypt

https://github.com/jetstack/kube-lego

1.8 = คือความฉิบหาย

cert-manager

Automatically provision and manage TLS certificates in Kubernetes

Auto ยิงไปขอต่อ Certificate จากเช่น Let’s Encrypt ใช้ได้หลาย https, http

https://github.com/jetstack/cert-manager

helm

The Kubernetes Package Manager

https://github.com/kubernetes/helm

Jobs

creates one or more pods and ensures that
a specified number of them successfully terminate

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: backup-postgres
spec:
  template:
    spec:
      restartPolicy: OnFailure
      volumes:
      - name: data
        gcePersistentDisk:
          pdName: backup-disk
          fsType: ext4
      containers:
      - name: postgres
        image: postgres:9
        imagePullPolicy: Always
        env:
        - name: PGPASSWORD
          valueFrom:
            secretKeyRef:
              name: postgres
              key: backup
        command:
        - /bin/sh
        - -c
        args:
        - pg_dumpall -U backup -h postgres > /data/$(date +"%Y%m%d%H%M%S")-postgres
        volumeMounts:
        - name: data
          mountPath: /data
```

Cron Jobs

manages time based Jobs

```yaml
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: backup-cronjob
spec:
  schedule: "0 21 * * *"
  successfulJobsHistoryLimit: 7
  failedJobsHistoryLimit: 7
  jobTemplate:
    spec:
      template:
        spec:
          restartPolicy: OnFailure
          volumes:
          - name: data
            gcePersistentDisk:
              pdName: backup-disk
              fsType: ext4
          containers:
          - name: postgres
            image: postgres:9
            imagePullPolicy: Always
            env:
            - name: PGPASSWORD
              valueFrom:
                secretKeyRef:
                  name: postgres
                  key: backup
            command:
            - /bin/sh
            - -c
            args:
            - pg_dumpall -U backup -h postgres > /data/$(date +"%Y%m%d%H%M%S")-postgres
            volumeMounts:
            - name: data
              mountPath: /data
```

Google Container Builder

Fast, consistent, reliable builds on Google Cloud Platform

Ex. push git auto deploy เข้า K8s เลย

Source: GitHub Repository: https://github.com/acoshift/acourse

View triggered builds

Name (Optional)

My trigger

Trigger type

- (•) Branch
- ( ) Tag

Branch (regex)

Matches 2 branches: master, staging

master|staging

Build configuration

- ( ) Dockerfile
  Specify the path within the Git repo
- (•) cloudbuild.yaml
  Specify the path to a Cloud Build configuration file in the Git repo Learn more

Cloud Build บอกเป็นสเตปได้

cloudbuild.yaml location

/ cloudbuild.yaml

Substitution variables (Optional)

Substitutions allow to re-use a cloudbuild.yaml file with different variable values Learn more

+ Add item

Summary

Changes pushed to master|staging branch will trigger a build defined by the "cloudbuild.yaml" file.

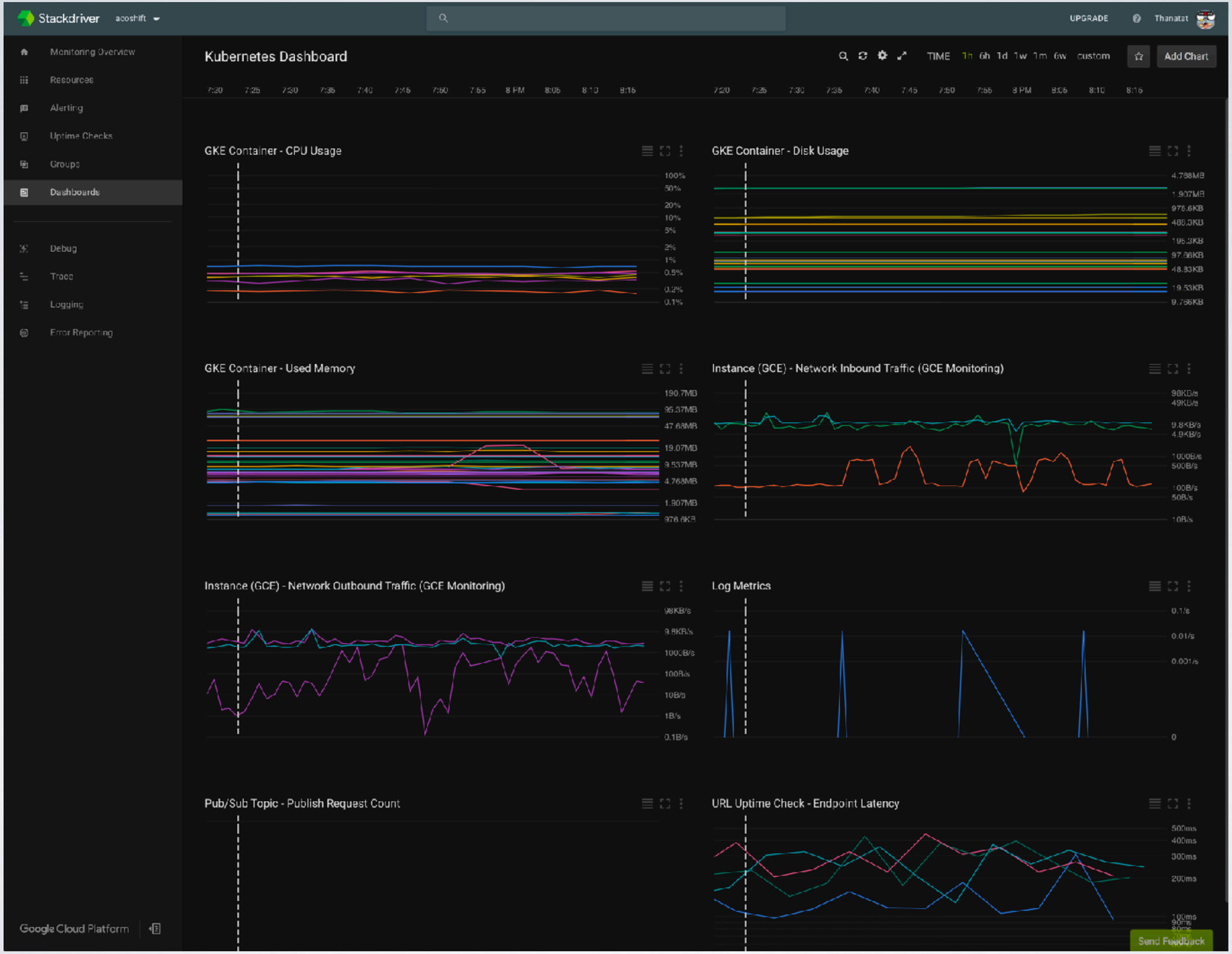
Save Cancel

```yaml
steps:
- name: 'gcr.io/cloud-builders/docker'
  args: ['build', '-t', 'gcr.io/$PROJECT_ID/myapp:$COMMIT_SHA', '.']
- name: 'gcr.io/cloud-builders/docker'
  args: ['push', 'gcr.io/$PROJECT_ID/myapp:$COMMIT_SHA']
- name: 'gcr.io/cloud-builders/kubectl'
  args: ['set', 'image', 'deploy/myapp', 'myapp=gcr.io/$PROJECT_ID/myapp:$COMMIT_SHA']
  env:
  - 'CLOUDSDK_COMPUTE_ZONE=asia-southeast1-b'
  - 'CLOUDSDK_CONTAINER_CLUSTER=cluster-1'
images:
  - 'gcr.io/$PROJECT_ID/myapp:$COMMIT_SHA'
```

```yaml
steps:
- name: 'gcr.io/cloud-builders/npm'
  args: ['install']
- name: 'gcr.io/cloud-builders/npm'
  args: ['run', 'build']
- name: 'gcr.io/cloud-builders/go'
  args: ['build', '-o', 'entrypoint', '-a', '-ldflags', '-w -s', 'cmd/acourse/main.go']
  env:
  - 'PROJECT_ROOT=github.com/acoshift/acourse'
  - 'GOOS=linux'
  - 'GOARCH=amd64'
  - 'CGO_ENABLED=0'
- name: 'gcr.io/cloud-builders/docker'
  args: ['build', '-t', 'gcr.io/$PROJECT_ID/acourse:$COMMIT_SHA', '.']
- name: 'gcr.io/cloud-builders/docker'
  args: ['push', 'gcr.io/$PROJECT_ID/acourse:$COMMIT_SHA']
- name: 'gcr.io/cloud-builders/kubectl'
  args: ['set', 'image', 'deploy/acourse', 'acourse=gcr.io/$PROJECT_ID/acourse:$COMMIT_SHA']
  env:
  - 'CLOUDSDK_COMPUTE_ZONE=asia-southeast1-b'
  - 'CLOUDSDK_CONTAINER_CLUSTER=cluster-sg-1'
images:
- 'gcr.io/$PROJECT_ID/acourse:$COMMIT_SHA'
```

Google Stackdriver

Monitoring, logging, and diagnostics for applications on Cloud Platform and AWS


Stackdriver
acoshift
UPGRADE
Thanatat
Monitoring Overview
Resources
Alerting
Uptime Checks
Groups
Dashboards
Debug
Trace
Logging
Error Reporting
Kubernetes Dashboard
TIME 1h 6h 1d 1w 1m 6w custom
Add Chart
GKE Container - CPU Usage
GKE Container - Disk Usage
GKE Container - Used Memory
Instance (GCE) - Network Inbound Traffic (GCE Monitoring)
Instance (GCE) - Network Outbound Traffic (GCE Monitoring)
Log Metrics
Pub/Sub Topic - Publish Request Count
URL Uptime Check - Endpoint Latency
Google Cloud Platform
Send Feedback

Q&A