[실습 2] 클러스터 배포 및 안정성 테스트

시작하기 전에

- kubectl 명령어 치트 시트
- https://kubernetes.io/docs/reference/kubectl/cheatsheet/

LAB

Kubernets Cluster

실습을 위한 쿠버네티스 클러스터 구성 정보 확인

```
# LAB002 디렉토리로 이동
$ cd ~/labhome/lab002/

$ labctl --help
Please use corret option [restore|rebuild]

   labctl restore: Quick lab restore
   labctl rebuild: Complete lab rebuild

$ labctl restore
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Restoring snapshot 'init-status' (83d22d48-eac8-49a9-812d-750af4412469)
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Waiting for VM "minikube" to power on...
VM "minikube" has been successfully started.
Switched to context "minikube".
minikube is ready!!

$ minikube ssh
                         _             _
            _         _ ( )           ( )
  ___ ___  (_)  ___  (_)| |/')  _   _ | |_      __
/' _ ` _ `\| |/' _ `\| || , <  ( ) ( )| '_`\  /'__`\
```

```
| ( ) ( ) || || ( ) || || |\`\ | (_) || |_) )(  ___/
(_) (_) (_)(_)(_) (_)(_)(_) (_)`\___/'(_,__/'`\____)

$ hostname
minikube
$ df -hT
Filesystem     Type      Size  Used Avail Use% Mounted on
devtmpfs       devtmpfs  3.9G     0  3.9G   0% /dev
tmpfs          tmpfs     3.9G     0  3.9G   0% /dev/shm
tmpfs          tmpfs     3.9G   17M  3.9G   1% /run
tmpfs          tmpfs     3.9G     0  3.9G   0% /sys/fs/cgroup
tmpfs          tmpfs     3.9G  8.0K  3.9G   1% /tmp
/dev/sda1      ext4       17G  1.3G   14G   9% /mnt/sda1
/hosthome      vboxsf    234G   25G  210G  11% /hosthome
$ exit
logout


$ kubectl cluster-info
Kubernetes master is running at https://192.168.99.100:8443
KubeDNS is running at https://192.168.99.100:8443/api/v1/namespaces/kube-
system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

$ kubectl get nodes
NAME       STATUS    ROLES     AGE       VERSION
minikube   Ready     master    21m       v1.10.0

$ kubectl describe nodes
Name:               minikube
Roles:              master
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/hostname=minikube
                    node-role.kubernetes.io/master=
Annotations:        node.alpha.kubernetes.io/ttl=0
                    volumes.kubernetes.io/controller-managed-attach-detach=true
CreationTimestamp:  Mon, 20 Aug 2018 03:15:00 +0900
Taints:             <none>
Unschedulable:      false

{{ 이하 출력 생략 }}

$ kubectl get namespaces
NAME          STATUS    AGE
default       Active    5m
kube-public   Active    5m
kube-system   Active    5m

$ kubectl get pod --all-namespaces
NAMESPACE     NAME                                    READY     STATUS    RESTARTS
AGE
```

```
kube-system   etcd-minikube                          1/1       Running   0          6m
kube-system   kube-addon-manager-minikube            1/1       Running   0          6m
kube-system   kube-apiserver-minikube                1/1       Running   0          6m
kube-system   kube-controller-manager-minikube       1/1       Running   0          6m
kube-system   kube-dns-86f4d74b45-grwt9              3/3       Running   0          7m
kube-system   kube-proxy-hklhf                       1/1       Running   0          7m
kube-system   kube-scheduler-minikube                1/1       Running   0          6m
kube-system   kubernetes-dashboard-5498ccf677-4qnnw  1/1       Running   0          7m
kube-system   metrics-server-85c979995f-tm4r8        1/1       Running   0          7m
kube-system   storage-provisioner                    1/1       Running   0          7m

$ kubectl get pod -o wide --all-namespaces
NAMESPACE     NAME                                   READY     STATUS    RESTARTS
AGE       IP           NODE
kube-system   etcd-minikube                          1/1       Running   0          7m
       10.0.2.15    minikube
kube-system   kube-addon-manager-minikube            1/1       Running   0          7m
       10.0.2.15    minikube
kube-system   kube-apiserver-minikube                1/1       Running   0          7m
       10.0.2.15    minikube
kube-system   kube-controller-manager-minikube       1/1       Running   0          7m
       10.0.2.15    minikube
kube-system   kube-dns-86f4d74b45-grwt9              3/3       Running   0          8m
       172.17.0.2   minikube
kube-system   kube-proxy-hklhf                       1/1       Running   0          8m
       10.0.2.15    minikube
kube-system   kube-scheduler-minikube                1/1       Running   0          7m
       10.0.2.15    minikube
kube-system   kubernetes-dashboard-5498ccf677-4qnnw  1/1       Running   0          8m
       172.17.0.3   minikube
kube-system   metrics-server-85c979995f-tm4r8        1/1       Running   0          8m
       172.17.0.4   minikube
kube-system   storage-provisioner                    1/1       Running   0          8m
       10.0.2.15    minikube

$ kubectl describe pod kube-apiserver-minikube -n kube-system
Name:         kube-apiserver-minikube
Namespace:    kube-system
Node:         minikube/10.0.2.15
Start Time:   Mon, 20 Aug 2018 03:14:18 +0900
Labels:       component=kube-apiserver
              tier=control-plane
Annotations:  kubernetes.io/config.hash=d6ed90b5a86db1591da65c1dfb8bdfc7
              kubernetes.io/config.mirror=d6ed90b5a86db1591da65c1dfb8bdfc7
              kubernetes.io/config.seen=2018-08-19T18:14:15.677572664Z
              kubernetes.io/config.source=file
              scheduler.alpha.kubernetes.io/critical-pod=
Status:       Running
IP:           10.0.2.15
Containers:
  kube-apiserver:
    Container ID:
docker://f2f05ffa9f73070558bf6951dbc62cc3b2c41625c6aaacb8ba55e68081b44843
```

```
    Image:         k8s.gcr.io/kube-apiserver-amd64:v1.10.0
```

Pod

Pod 명세서 예제 내용 확인 및 배포 연습

```
# LAB002 디렉토리로 이동
$ cd ~/labhome/lab002/

$ cat nginx-pod.yml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: web
spec:
  containers:
  - name: nginx
    image: nginx
    ports:
    - containerPort: 80

$ kubectl create -f nginx-pod.yml
pod/nginx created

$ kubectl get pod
NAME        READY     STATUS              RESTARTS   AGE
nginx-pod   0/1       ContainerCreating   0          6s

$ kubectl get pod
NAME        READY     STATUS    RESTARTS   AGE
nginx-pod   1/1       Running   0          7s

$ kubectl describe pod nginx-pod
Name:         nginx-pod
Namespace:    default
Node:         minikube/10.0.2.15
Start Time:   Mon, 20 Aug 2018 03:42:54 +0900
Labels:       app=web
Annotations:  <none>
Status:       Running
IP:           172.17.0.5
Containers:
  nginx:
    Container ID:
docker://0b55ab4d2f4899de6b47beed1a58f9b18b4c8baf5527fded49b2bf96c6fb02e1
    Image:          nginx

{{ 이하 출력 생략 }}
```

```
# minikube 외부 호스트에서는 Pod 에 접근할 수 없음
$ curl 172.17.0.5
curl: (7) Failed to connect to 172.17.0.5 port 80: 호스트로 갈 루트가 없음

# minikube 내부에서는 Pod 에 접근 가능
$ minikube ssh
                         _             _
            _         _ ( )           ( )
  ___ ___  (_)  ___  (_)| |/')  _   _ | |_      __
/' _ ` _ `\| |/' _ `\| || , <  ( ) ( )| '_`\  /'__`\
| ( ) ( ) || || ( ) || || |\`\ | (_) || |_) )(  ___/
(_) (_) (_)(_)(_) (_)(_)(_) (_)`\___/'(_,__/''`\____)

$ curl 172.17.0.5
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>

$ cat nginx-hostport-pod.yml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod-hostport
  labels:
    app: web
spec:
  containers:
  - name: nginx
    image: nginx
    ports:
    - containerPort: 80
```

```
      hostPort: 8080

$ kubectl create -f nginx-hostport-pod.yml
pod/nginx-pod-hostport created

$ kubectl get pod
NAME                 READY     STATUS              RESTARTS   AGE
nginx-pod            1/1       Running             0          5m
nginx-pod-hostport   0/1       ContainerCreating   0          5s

$ kubectl get pod
NAME                 READY     STATUS    RESTARTS   AGE
nginx-pod            1/1       Running   0          5m
nginx-pod-hostport   1/1       Running   0          6s

$ kubectl describe pod nginx-pod-hostport
Name:         nginx-pod-hostport
Namespace:    default
Node:         minikube/10.0.2.15
Start Time:   Mon, 20 Aug 2018 03:47:57 +0900
Labels:       app=web
Annotations:  <none>
Status:       Running
IP:           172.17.0.6
Containers:
  nginx:
    Container ID:
docker://cf5ba55df5e2b6fe87c7b590d9bdf4b80523be159713b8e762573548c8bc6027
    Image:          nginx
    Image ID:       docker-
pullable://nginx@sha256:d85914d547a6c92faa39ce7058bd7529baacab7e0cd4255442b04577c4d1f42
4
    Port:           80/TCP
    Host Port:      8080/TCP

{{ 이하 출력 생략 }}

$ minikube ip
192.168.99.100

$ curl $(minikube ip):8080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
```

```
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>

$ kubectl get pod --show-labels=true
NAME                READY     STATUS    RESTARTS   AGE       LABELS
nginx-pod           1/1       Running   0          27m       app=web
nginx-pod-hostport   1/1       Running   0          22m       app=web

$ kubectl delete pod nginx-pod nginx-pod-hostport
pod "nginx-pod" deleted
pod "nginx-pod-hostport" deleted
```

ReplicaSet

ReplicaSet 명세서 예제 내용 확인 및 배포 연습 Pod 장애시 ReplicaSet 복구 과정 확인

```
# LAB002 디렉토리로 이동
$ cd ~/labhome/lab002/

$ cat frontend-rs.yml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: frontend
  labels:
    app: guestbook
    tier: frontend
spec:
  # modify replicas according to your case
  replicas: 3
  selector:
    matchLabels:
      tier: frontend
    matchExpressions:
      - {key: tier, operator: In, values: [frontend]}
  template:
    metadata:
      labels:
        app: guestbook
        tier: frontend
```

```
    spec:
      containers:
      - name: php-redis
        image: gcr.io/google_samples/gb-frontend:v3
        resources:
          requests:
            cpu: 100m
            memory: 100Mi
        env:
        - name: GET_HOSTS_FROM
          value: dns
          # If your cluster config does not include a dns service, then to
          # instead access environment variables to find service host
          # info, comment out the 'value: dns' line above, and uncomment the
          # line below.
          # value: env
        ports:
        - containerPort: 80

$ kubectl create -f frontend-rs.yml
replicaset.apps/frontend created

$ kubectl get rs
NAME       DESIRED   CURRENT   READY     AGE
frontend   3         3         3         1m

$ kubectl describe rs frontend
Name:         frontend
Namespace:    default
Selector:     tier=frontend,tier in (frontend)
Labels:       app=guestbook
              tier=frontend
Annotations:  <none>
Replicas:     3 current / 3 desired
Pods Status:  3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=guestbook
           tier=frontend
  Containers:
   php-redis:
    Image:      gcr.io/google_samples/gb-frontend:v3
    Port:       80/TCP
    Host Port:  0/TCP
    Requests:
      cpu:     100m
      memory:  100Mi
    Environment:
      GET_HOSTS_FROM:  dns
    Mounts:            <none>
  Volumes:             <none>
Events:
  Type    Reason            Age   From                   Message
  ----    ------            ----  ----                   -------
```

```
  Normal  SuccessfulCreate  1m    replicaset-controller  Created pod: frontend-wskjg
  Normal  SuccessfulCreate  1m    replicaset-controller  Created pod: frontend-k24t7
  Normal  SuccessfulCreate  1m    replicaset-controller  Created pod: frontend-xx5h8

$ cp frontend-rs.yml frontend-rs.yml.orig
$ sed -i 's/replicas:.*/replicas: 5/' frontend-rs.yml
$ grep replicas frontend-rs.yml
  # modify replicas according to your case
  replicas: 5

$ kubectl apply -f frontend-rs.yml
Warning: kubectl apply should be used on resource created by either kubectl create --
save-config or kubectl apply
replicaset.apps/frontend configured

$ kubectl get rs
NAME       DESIRED   CURRENT   READY     AGE
frontend   5         5         5         5m

$ kubectl get pod
NAME             READY     STATUS    RESTARTS   AGE
frontend-nk2hn   1/1       Running   0          13s
frontend-pk7s8   1/1       Running   0          13s
frontend-vjdtn   1/1       Running   0          13s
frontend-wskjg   1/1       Running   0          14m
frontend-xx5h8   1/1       Running   0          14m

$ kubectl delete pod frontend-wskjg frontend-xx5h8 frontend-vjdtn
pod "frontend-wskjg" deleted
pod "frontend-xx5h8" deleted
pod "frontend-vjdtn" deleted

$ kubectl get rs
NAME       DESIRED   CURRENT   READY     AGE
frontend   5         5         2         15m
$ kubectl get rs
NAME       DESIRED   CURRENT   READY     AGE
frontend   5         5         3         15m
$ kubectl get rs
NAME       DESIRED   CURRENT   READY     AGE
frontend   5         5         4         15m
$ kubectl get rs
NAME       DESIRED   CURRENT   READY     AGE
frontend   5         5         5         15m

$ kubectl get pods
NAME             READY     STATUS    RESTARTS   AGE
frontend-njhvq   1/1       Running   0          1m
frontend-nk2hn   1/1       Running   0          1m
frontend-pk7s8   1/1       Running   0          1m
frontend-trzrk   1/1       Running   0          1m
frontend-xdmjp   1/1       Running   0          1m
```

```
$ kubectl describe rs frontend
Events:
  Type    Reason            Age              From                   Message
  ----    ------            ----             ----                   -------
  Normal  SuccessfulCreate  17m              replicaset-controller  Created pod:
frontend-wskjg
  Normal  SuccessfulCreate  17m              replicaset-controller  Created pod:
frontend-k24t7
  Normal  SuccessfulCreate  17m              replicaset-controller  Created pod:
frontend-xx5h8
  Normal  SuccessfulCreate  12m              replicaset-controller  Created pod:
frontend-kpwzf
  Normal  SuccessfulCreate  12m              replicaset-controller  Created pod:
frontend-ktsn6
  Normal  SuccessfulCreate  6m               replicaset-controller  Created pod:
frontend-6pjk8
  Normal  SuccessfulCreate  3m               replicaset-controller  Created pod:
frontend-cpj8p
  Normal  SuccessfulCreate  3m               replicaset-controller  Created pod:
frontend-hswrf
  Normal  SuccessfulCreate  3m               replicaset-controller  Created pod:
frontend-gbh77
  Normal  SuccessfulCreate  2m (x6 over 2m)  replicaset-controller  (combined from
similar events): Created pod: frontend-njhvq

$ kubectl delete rs frontend
replicaset.extensions "frontend" deleted
```

Serivce

Service 명세서 예제 내용 확인 및 배포 연습 앞서 배포한 ReplicaSet 과 Service 연결

```
# LAB002 디렉토리로 이동
$ cd ~/labhome/lab002/

$ cat hello-app-rs.yml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: hello-app-rs
  labels:
    app: hello-app
    tier: frontend
spec:
  replicas: 3
  selector:
    matchLabels:
      app: hello-app
      tier: frontend
  template:
    metadata:
```

```
      labels:
        app: hello-app
        tier: frontend
    spec:
      containers:
      - name: hello-app
        image: gcr.io/google-samples/hello-app:2.0
        ports:
          - containerPort: 8080

$ kubectl create -f hello-app-rs.yml
replicaset.apps/hello-app-rs created

$ kubectl get pods
NAME                 READY     STATUS    RESTARTS   AGE
hello-app-rs-5jlpp   1/1       Running   0          4s
hello-app-rs-dp2kn   1/1       Running   0          4s
hello-app-rs-wz4bw   1/1       Running   0          4s

$ kubectl describe pod hello-app-rs-5jlpp | grep IP
IP:             172.17.0.7

#minikube 에 SSH 로 접근하는 대신에 busybox 이미지로 curl 실행
$ kubectl run busyboxplus --image=radial/busyboxplus:curl -i --tty --rm
If you don't see a command prompt, try pressing enter.
[ root@busyboxplus-5697648fcc-vgkhj:/ ]$ curl 172.17.0.7:8080
Hello, world!
Version: 2.0.0
Hostname: hello-app-rs-5jlpp
[ root@busyboxplus-5697648fcc-vgkhj:/ ]$ exit
Session ended, resume using 'kubectl attach busyboxplus-5697648fcc-vgkhj -c busyboxplus
-i -t' command when the pod is running
deployment.apps "busyboxplus" deleted

$ cat hello-app-svc.yml
apiVersion: v1
kind: Service
metadata:
  name: hello-app-svc
  labels:
    app: hello-app
spec:
  selector:
    app: hello-app
    tier: frontend
  ports:
  - port: 80
    targetPort: 8080

$ kubectl create -f hello-app-svc.yml
service/hello-app-svc created

$ kubectl get svc
```

```
NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
hello-app-svc   ClusterIP   10.99.224.94   <none>        80/TCP    5s
kubernetes      ClusterIP   10.96.0.1      <none>        443/TCP   1h

$ kubectl get svc -o wide
NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE       SELECTOR
hello-app-svc   ClusterIP   10.99.224.94   <none>        80/TCP    1m        app=hello-
app,tier=frontend
kubernetes      ClusterIP   10.96.0.1      <none>        443/TCP   1h        <none>

$ kubectl get pod --show-labels=true
NAME                        READY     STATUS    RESTARTS   AGE       LABELS
busybox-5858cc4697-bbrhr    1/1       Running   0          10m       pod-template-
hash=1414770253,run=busybox
hello-app-rs-5jlpp          1/1       Running   0          13m       app=hello-
app,tier=frontend
hello-app-rs-dp2kn          1/1       Running   0          13m       app=hello-
app,tier=frontend
hello-app-rs-wz4bw          1/1       Running   0          13m       app=hello-
app,tier=frontend

$ kubectl run busyboxplus --image=radial/busyboxplus:curl -i --tty --rm
If you don't see a command prompt, try pressing enter.
[ root@busyboxplus-5697648fcc-8ljqp:/ ]$ curl 10.99.224.94
Hello, world!
Version: 2.0.0
Hostname: hello-app-rs-dp2kn
[ root@busyboxplus-5697648fcc-8ljqp:/ ]$ curl 10.99.224.94
Hello, world!
Version: 2.0.0
Hostname: hello-app-rs-5jlpp
[ root@busyboxplus-5697648fcc-8ljqp:/ ]$ curl 10.99.224.94
Hello, world!
Version: 2.0.0
Hostname: hello-app-rs-wz4bw
[ root@busyboxplus-5697648fcc-8ljqp:/ ]$ exit
Session ended, resume using 'kubectl attach busyboxplus-5697648fcc-8ljqp -c busyboxplus
-i -t' command when the pod is running
deployment.apps "busyboxplus" deleted

$ kubectl create -f hello-app-svc-nodeport.yml
service/hello-app-svc-nodeport created

$ kubectl get svc
NAME                    TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)         AGE
hello-app-svc           ClusterIP   10.99.224.94     <none>        80/TCP          6m
hello-app-svc-nodeport  NodePort    10.109.169.162   <none>        80:32045/TCP    5s
kubernetes              ClusterIP   10.96.0.1        <none>        443/TCP         1h

$ cat hello-app-svc-nodeport.yml
apiVersion: v1
kind: Service
metadata:
```

```
  name: hello-app-svc-nodeport
  labels:
    app: hello-app
spec:
  type: NodePort
  selector:
    app: hello-app
    tier: frontend
  ports:
  - port: 80
    targetPort: 8080

$ kubectl create -f hello-app-svc-nodeport.yml
service/hello-app-svc-nodeport created
$ kubectl get svc
NAME                    TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
hello-app-svc           ClusterIP   10.99.224.94     <none>        80/TCP         6m
hello-app-svc-nodeport  NodePort    10.109.169.162   <none>        80:32045/TCP   5s
kubernetes              ClusterIP   10.96.0.1        <none>        443/TCP        1h

$ minikube service list
|-------------|-------------------------|-----------------------------|
|  NAMESPACE  |          NAME           |             URL             |
|-------------|-------------------------|-----------------------------|
| default     | hello-app-svc           | No node port                |
| default     | hello-app-svc-nodeport | http://192.168.99.100:32045 |
| default     | kubernetes              | No node port                |
| kube-system | kube-dns                | No node port                |
| kube-system | kubernetes-dashboard    | http://192.168.99.100:30000 |
| kube-system | metrics-server          | No node port                |
|-------------|-------------------------|-----------------------------|

$ minikube service hello-app-svc-nodeport
Opening kubernetes service default/hello-app-svc-nodeport in default browser...

$ minikube service hello-app-svc-nodeport --url
http://192.168.99.100:32045

$ curl $(minikube service hello-app-svc-nodeport --url)
Hello, world!
Version: 2.0.0
Hostname: hello-app-rs-dp2kn
$ curl $(minikube service hello-app-svc-nodeport --url)
Hello, world!
Version: 2.0.0
Hostname: hello-app-rs-5jlpp
$ curl $(minikube service hello-app-svc-nodeport --url)
Hello, world!
Version: 2.0.0
Hostname: hello-app-rs-wz4bw

$ labctl restore
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
```

```
Restoring snapshot 'init-status' (01419346-a9c2-4ca6-8375-2e8f12c6762f)
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Waiting for VM "minikube" to power on...
VM "minikube" has been successfully started.
Switched to context "minikube".
minikube is ready!!
```

References

- https://kubernetes.io/docs/concepts/workloads/pods/pod-overview/
- https://kubernetes.io/docs/concepts/workloads/pods/pod/
- https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/
- https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/
- https://kubernetes.io/docs/concepts/services-networking/service/
- https://kubernetes.io/docs/reference/kubectl/cheatsheet/