쿠버네티스(k8s) 대시보드 설치 및 외부 접속 기능 추가하기

Cello플랫폼팀 정연호

대쉬보드는 웹 기반의 k8s UI입니다. 대쉬보드를 통해 컨테이너 배포, 트러블 슈팅, 클러스터 리소스 관리를 할수 있습니다. 자세한 내용은 k8s 공식 문서를 참고하세요.

Web UI (Dashboard) - https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/

클라우드 환경에서는 Ingress를 이용해서 L7 기반의 LB를 지원하나 Bare Metal환경에서는 정상적으로 동작하지 않는듯 합니다. 그래서, k8s 문서에서는 MetalLB를 사용하도록 가이드 하고 있습니다. https://kubernetes.github.io/ingress-nginx/deploy/baremetal/

1. MetalLB 설치

MetalLB를 아래와 같이 배포합니다.

```
$ kubectl apply -f
https://raw.githubusercontent.com/google/metallb/v0.7.3/manifests/metallb.yaml
$ kubectl get pods -n metallb-system
NAME                           READY   STATUS    RESTARTS   AGE
controller-7cc9c87cfb-vzpx6    1/1     Running   0          40m
speaker-v5bg7                  1/1     Running   0          38m
```

정상적으로 설치된 후 설정(ConfigMap)을 아래와 같이 작성합니다. addresses 항목은 호스트의 IP 대역중 일부를 지정합니다. 아래 예에서는 192.168.56.0/24 네트워크를 사용한 경우입니다.

```
$ vi layer2-config.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: metallb-system
  name: config
data:
  config: |
    address-pools:
    - name: default
      protocol: layer2
      addresses:
      - 192.168.56.240-192.168.56.250
```

설정 파일 적용 후 로그를 확인합니다.

```
$ kubectl apply -f layer2-config.yaml
$ kubectl logs -l component=speaker -n metallb-system
{"caller":"main.go:229","event":"serviceAnnounced","ip":"192.168.56.240","msg":"service has
IP, announcing","pool":"my-ip-space","protocol":"layer2","service":"kube-system/kubernetes-
dashboard","ts":"2019-03-10T15:43:13.074660522Z"}
{"caller":"main.go:231","event":"endUpdate","msg":"end of service update","service":"kube-
system/kubernetes-dashboard","ts":"2019-03-10T15:43:13.074679512Z"}
{"caller":"announcer.go:89","event":"createARPResponder","interface":"cali91edbbd7430","msg":
"created ARP responder for interface","ts":"2019-03-10T15:43:16.327434239Z"}
{"caller":"announcer.go:94","error":"creating NDP responder for \"cali91edbbd7430\": listen
ip6:ipv6-icmp fe80::ecee:eeff:feee:eeee%cali91edbbd7430: bind: invalid
argument","interface":"cali91edbbd7430","msg":"failed to create NDP
responder","op":"createNDPResponder","ts":"2019-03-10T15:43:16.327555722Z"}
{"caller":"announcer.go:94","error":"creating NDP responder for \"cali91edbbd7430\": listen
ip6:ipv6-icmp fe80::ecee:eeff:feee:eeee%cali91edbbd7430: bind: invalid
argument","interface":"cali91edbbd7430","msg":"failed to create NDP
responder","op":"createNDPResponder","ts":"2019-03-10T15:43:26.328656021Z"}
{"caller":"announcer.go:94","error":"creating NDP responder for \"cali91edbbd7430\": listen
ip6:ipv6-icmp fe80::ecee:eeff:feee:eeee%cali91edbbd7430: bind: invalid
argument","interface":"cali91edbbd7430","msg":"failed to create NDP
responder","op":"createNDPResponder","ts":"2019-03-10T15:43:36.335616967Z"}
{"caller":"announcer.go:94","error":"creating NDP responder for \"cali91edbbd7430\": listen
ip6:ipv6-icmp fe80::ecee:eeff:feee:eeee%cali91edbbd7430: bind: invalid
argument","interface":"cali91edbbd7430","msg":"failed to create NDP
responder","op":"createNDPResponder","ts":"2019-03-10T15:43:46.336750005Z"}
{"caller":"announcer.go:94","error":"creating NDP responder for \"cali91edbbd7430\": listen
ip6:ipv6-icmp fe80::ecee:eeff:feee:eeee%cali91edbbd7430: bind: invalid
argument","interface":"cali91edbbd7430","msg":"failed to create NDP
responder","op":"createNDPResponder","ts":"2019-03-10T15:43:56.339752158Z"}
{"caller":"announcer.go:98","event":"createNDPResponder","interface":"cali91edbbd7430","msg":
"created NDP responder for interface","ts":"2019-03-10T15:44:06.341274101Z"}
{"caller":"arp.go:102","interface":"enp0s8","ip":"192.168.56.240","msg":"got ARP request for
service IP, sending
response","responseMAC":"08:00:27:52:8c:8c","senderIP":"192.168.56.1","senderMAC":"0a:00:27:0
0:00:00","ts":"2019-03-10T16:03:27.247572222Z"}
```

2. Dashboard 설치

Dashboard는 HTTPS(SSL/TLS)로만 접속이 가능합니다. 그래서, Dashboard에서 사용할 Key, Cert를 아래와 같이 생성합니다.

- 개인키, CSR(Certificate Signing Request) 생성하기

```
$ mkdir certs; cd certs
$ openssl genrsa -des3 -passout pass:x -out dashboard.pass.key 2048
Generating RSA private key, 2048 bit long modulus
........................+++
.....................................................................+++
e is 65537 (0x10001)
$ openssl rsa -passin pass:x -in dashboard.pass.key -out dashboard.key
writing RSA key
$ rm dashboard.pass.key
$ openssl req -new -key dashboard.key -out dashboard.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
```

```
Country Name (2 letter code) [AU]:KR
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:Seoul
Organization Name (eg, company) [Internet Widgits Pty Ltd]:YH
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:dashboard.k8s.local
Email Address []:
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

- SSL Certificate 생성하기

```
$ openssl x509 -req -sha256 -days 365 -in dashboard.csr -signkey dashboard.key -out
dashboard.crt
```

- k8s secret 만들기

```
$ ls
dashboard.csr  dashboard.key  dashboard.pass.key
$ cd ..
$ kubectl create secret generic kubernetes-dashboard-certs --from-file=./certs -n kube-system
secret/kubernetes-dashboard-certs created
```

- Dashboard Yaml파일을 다운로드 한후 Service의 type을 LoadBalancer로 수정합니다.

```
$ wget
https://raw.githubusercontent.com/kubernetes/dashboard/v1.10.1/src/deploy/recommended/kuberne
tes-dashboard.yaml
$ vi kubernetes-dashboard.yaml
# ------------------- Dashboard Service ------------------- #
kind: Service
apiVersion: v1
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kube-system
spec:
  type: LoadBalancer # <-- 추가
  ports:
    - port: 443
      targetPort: 8443
  selector:
    k8s-app: kubernetes-dashboard
$ kubectl apply -f kubernetes-dashboard.yaml
secret/kubernetes-dashboard-certs created
serviceaccount/kubernetes-dashboard created
role.rbac.authorization.k8s.io/kubernetes-dashboard-minimal created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard-minimal created
deployment.apps/kubernetes-dashboard created
service/kubernetes-dashboard created

$ kubectl get svc --all-namespaces
NAMESPACE     NAME                      TYPE           CLUSTER-IP      EXTERNAL-IP      PORT(S)
        AGE
default       kubernetes                ClusterIP      10.96.0.1       <none>           443/TCP
        5m27s
kube-system   calico-etcd               ClusterIP      10.96.232.136   <none>           6666/TCP
       38m
```

```
kube-system   kube-dns               ClusterIP      10.96.0.10      <none>
53/UDP,53/TCP   39m
kube-system   kubernetes-dashboard   LoadBalancer   10.100.223.71   192.168.56.240
443:32393/TCP   16s
```

3. Dashboard Admin계정 생성

아래와 같이 Dashboard Amin Yaml파일을 생성합니다.

```
$ vi dashboard-admin.yaml
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kube-system
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kube-system
```

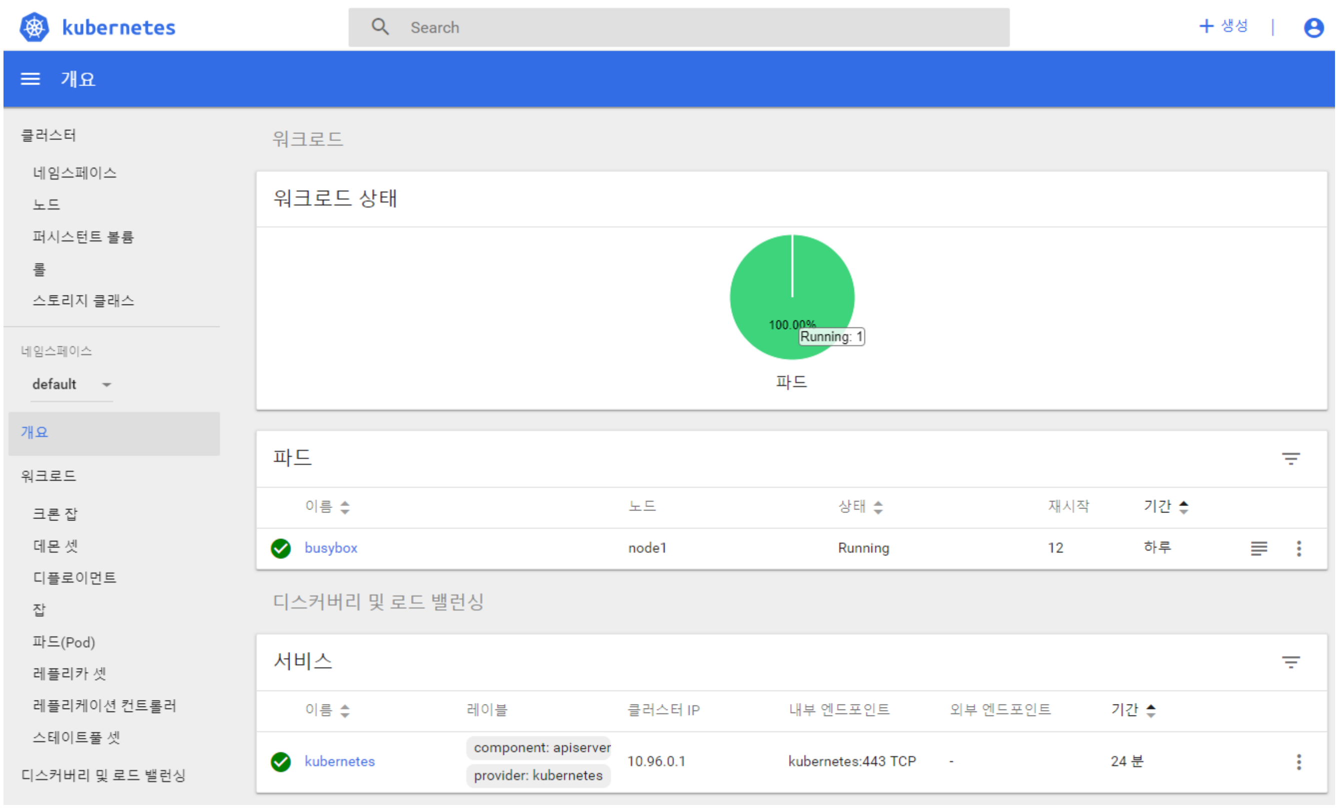
적용후 Token을 구합니다. Token값은 Dashboard 로그인 시에 사용됩니다.

```
$ kubectl apply -f dashboard-admin.yaml
clusterrolebinding.rbac.authorization.k8s.io/admin-user created
$ kubectl -n kube-system describe secret $(kubectl -n kube-system get secret | grep admin-user
| awk '{print $1}')
Name:         admin-user-token-44485
Namespace:    kube-system
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: admin-user
              kubernetes.io/service-account.uid: 8bebbff6-434b-11e9-95b4-080027129a38
Type:  kubernetes.io/service-account-token
Data
====
ca.crt:     1025 bytes
namespace:  11 bytes
token:
eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlc
y5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlLXN5c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY29
1bnQvc2VjcmV0Lm5hbWUiOiJhZG1pbi11c2VyLXRva2VuLTQ0NDg1Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3Vud
C9zZXJ2aWNlLWFjY291bnQubmFtZSI6ImFkbWluLXVzZXIiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZ
pY2UtYWNjb3VudC51aWQiOiI4YmViYmZmNi00MzRiLTExZTktOTViNC0wODAwMjcxMjlhMzgiLCJzdWIiOiJzeXN0ZW06c
2VydmljZWFjY291bnQ6a3ViZS1zeXN0ZW06YWRtaW4tdXNlciJ9.m24R-
c_ThYoeYR92s9KiNr4AFYyNOUOpN95XOGCGcjPP4o9ieQbTrpu8OYmIRa1uL8bspNqkraO2jMK1G_V2XOPpd_G19yeoZSV
qp0_FDDniPnO0pGDg0jyQH1T5xF2jy3lElOveBHXOT1CirR1NOQ791PvqZ5NA7xrHucCSXBP9V7eUjdwKbvwsyIQdufOLY
EUygQ3mj_jkXjmaN31e2qUZYRhZfeog54r0qj_g7WOGtFmoy06B3ifgLlbrX4eT_eOZfrhma67KWXGor1KTKfuy25Pv8Pb
d7kcrdafCam6YdTyTaehskMycNBo--m-hteYcxiYa-QfjfXH1w4bcwQ
```

아래 명령어로 확인된 Dashboard의 LoadBalancer IP를 통해 접속 및 로그인 합니다.

```
$ kubectl get svc kubernetes-dashboard -n kube-system
NAME                   TYPE           CLUSTER-IP      EXTERNAL-IP      PORT(S)         AGE
kubernetes-dashboard   LoadBalancer   10.100.223.71   192.168.56.240   443:32393/TCP   38m
```

브라우저를 통해 https://192.168.56.240 사이트를 접속한 후 Token값을 입력하면 아래와 같은 화면을 보실 수 있습니다.



파드

이름	노드	상태	재시작	기간
busybox	node1	Running	12	하루

서비스

이름	레이블	클러스터 IP	내부 엔드포인트	외부 엔드포인트	기간
kubernetes	component: apiserver provider: kubernetes	10.96.0.1	kubernetes:443 TCP	-	24 분