[kubernetes 설치]

설치에는 root권한이 필요하므로 sudo su -로 root권한을 얻은 후 설치를 진행하면 됩니다.

설치방법은 master와 node 동일합니다.

아래 명령을 kube-master-1, kube-node-1, kube-node-2등 모든 VM 인스턴스에서 실행합니다.

1. package 설치 [ master, node 시행]

**apt-get install openssh-server curl apt-transport-https**

**curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -**

**cat <<EOF > /etc/apt/sources.list.d/kubernetes.list**

**>deb http://apt.kubernetes.io/ kubernetes-xenial main**

**>EOF**

**apt-get update**

**apt-get install -y docker-engine**

**apt-get install -y kubelet kubeadm kubectl kubernetes-cni**

2 . master 초기화 [master만 시행]

root@ubuntu:~# **kubeadm init**

[kubeadm] WARNING: kubeadm is in beta, please do not use it for production clust ers.

[init] Using Kubernetes version: v1.6.0

[init] Using Authorization mode: RBAC

[preflight] Running pre-flight checks

[preflight] WARNING: hostname "ubuntu" could not be reached

[preflight] WARNING: hostname "ubuntu" lookup ubuntu on 192.168.17.2:53: no such host

[certificates] Generated CA certificate and key.

[certificates] Generated API server certificate and key.

[certificates] API Server serving cert is signed for DNS names [ubuntu kubernete s kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local ] and IPs [10.96.0.1 192.168.17.186]

[certificates] Generated API server kubelet client certificate and key.

[certificates] Generated service account token signing key and public key.

[certificates] Generated front-proxy CA certificate and key.

[certificates] Generated front-proxy client certificate and key.

[certificates] Valid certificates and keys now exist in "/etc/kubernetes/pki"

[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/admin.conf"

[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/kubelet.conf"

[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/controller-manager. conf"

[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/scheduler.conf"

[apiclient] Created API client, waiting for the control plane to become ready

[apiclient] All control plane components are healthy after 156.820507 seconds

[apiclient] Waiting for at least one node to register

[apiclient] First node has registered after 9.509898 seconds

[token] Using token: 3f538d.a0ad9a1a7be5f11c

[apiconfig] Created RBAC rules

[addons] Created essential addon: kube-proxy

[addons] Created essential addon: kube-dns

Your Kubernetes master has initialized successfully!

To start using your cluster, you need to run **(as a regular user): su – choga88**

**sudo cp /etc/kubernetes/admin.conf $HOME/**

**sudo chown $(id -u):$(id -g) $HOME/admin.conf**

**export KUBECONFIG=$HOME/admin.conf**

You should now deploy a pod network to the cluster.

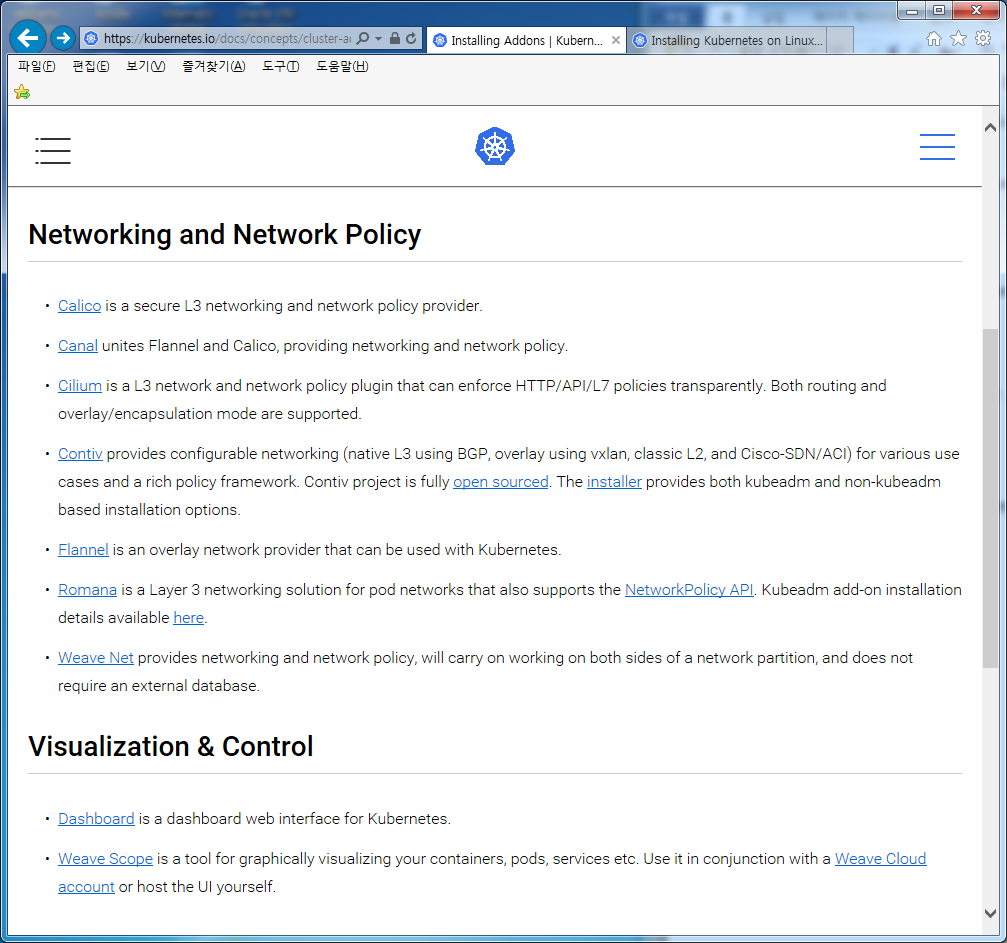
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

http://kubernetes.io/docs/admin/addons/

You can now join any number of machines by running the following on each node

as root:

2. Installing a pod network [master 노드]



Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

http://kubernetes.io/docs/admin/addons/

**choga88@192:~$ kubectl apply -f https://git.io/weave-kube-1.6**

clusterrole "weave-net" created

serviceaccount "weave-net" created

clusterrolebinding "weave-net" created

daemonset "weave-net" created

**choga88@192:~$ kubectl get pods --all-namespaces**

NAMESPACE NAME READY STATUS RESTARTS AGE

kube-system etcd-192 1/1 Running 0 35m

kube-system kube-apiserver-192 1/1 Running 0 36m

kube-system kube-controller-manager-192 1/1 Running 0 36m

**kube-system kube-dns-3913472980-shrmh 3/3 Running 0 36m**

kube-system kube-proxy-3l5jp 1/1 Running 0 36m

kube-system kube-scheduler-192 1/1 Running 0 36m

kube-system weave-net-6x4kg 2/2 Running 0 4m

choga88@192:~$ **/sbin/ifconfig**

weave Link encap:Ethernet HWaddr 66:f5:61:e4:10:22

**inet addr:10.32.0.1** Bcast:0.0.0.0 Mask:255.240.0.0

inet6 addr: fe80::64f5:61ff:fee4:1022/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1376 Metric:1

RX packets:1455 errors:0 dropped:0 overruns:0 frame:0

TX packets:1747 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:310535 (310.5 KB) TX bytes:376114 (376.1 KB)

**kubeadm join --token 3f538d.a0ad9a1a7be5f11c 192.168.17.186:6443**

3. node가 master조인 [node만 시행]

root@ubuntu2:~# **kubeadm join --token ef43c5.2dabff2357fd6fce 192.168.17.194:6443**

[kubeadm] WARNING: kubeadm is in beta, please do not use it for production clusters.

[preflight] Running pre-flight checks

[preflight] WARNING: hostname "ubuntu2" could not be reached

[preflight] WARNING: hostname "ubuntu2" lookup ubuntu2 on 192.168.17.2:53: no such host

[discovery] Trying to connect to API Server "192.168.17.194:6443"

[discovery] Created cluster-info discovery client, requesting info from "https://192.168.17.194:6443"

[discovery] Cluster info signature and contents are valid, will use API Server "https://192.168.17.194:6443"

[discovery] Successfully established connection with API Server "192.168.17.194:6443"

[bootstrap] Detected server version: v1.6.2

[bootstrap] The server supports the Certificates API (certificates.k8s.io/v1beta1)

[csr] Created API client to obtain unique certificate for this node, generating keys and certificate signing request

[csr] Received signed certificate from the API server, generating KubeConfig...

[kubeconfig] Wrote KubeConfig file to disk: "/etc/kubernetes/kubelet.conf"

Node join complete:

\* Certificate signing request sent to master and response

received.

\* Kubelet informed of new secure connection details.

Run 'kubectl get nodes' on the master to see this machine join.

choga88@ubuntu:~$ **kubectl get nodes**

NAME STATUS AGE VERSION

ubuntu **Ready**  25m v1.6.2

ubuntu2 **Ready** 2m v1.6.2

4. sample app 설치

choga88@ubuntu:~$ **kubectl create namespace sock-shop**

namespace "sock-shop" created

choga88@ubuntu:~$ **kubectl apply -n sock-shop -f "https://github.com/microservices-demo/microservices-demo/blob/master/deploy/kubernetes/complete-demo.yaml?raw=true"**

Warning: kubectl apply should be used on resource created by either kubectl create --save-config or kubectl apply

namespace "sock-shop" configured

namespace "zipkin" created

deployment "carts-db" created

service "carts-db" created

deployment "carts" created

service "carts" created

deployment "catalogue-db" created

service "catalogue-db" created

deployment "catalogue" created

service "catalogue" created

deployment "front-end" created

service "front-end" created

deployment "orders-db" created

service "orders-db" created

deployment "orders" created

service "orders" created

deployment "payment" created

service "payment" created

deployment "queue-master" created

service "queue-master" created

deployment "rabbitmq" created

service "rabbitmq" created

deployment "shipping" created

service "shipping" created

deployment "user-db" created

service "user-db" created

deployment "user" created

service "user" created

the namespace from the provided object "zipkin" does not match the namespace "sock-shop". You must pass '--namespace=zipkin' to perform this operation.

the namespace from the provided object "zipkin" does not match the namespace "sock-shop". You must pass '--namespace=zipkin' to perform this operation.

the namespace from the provided object "zipkin" does not match the namespace "sock-shop". You must pass '--namespace=zipkin' to perform this operation.

the namespace from the provided object "zipkin" does not match the namespace "sock-shop". You must pass '--namespace=zipkin' to perform this operation.

the namespace from the provided object "zipkin" does not match the namespace "sock-shop". You must pass '--namespace=zipkin' to perform this operation.

choga88@ubuntu:~$ **kubectl describe svc front-end -n sock-shop**

Name: front-end

Namespace: sock-shop

Labels: name=front-end

Annotations: kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"name":"front-end"},"name":"front-end","namespace":"sock-shop"},"spec":{"por...

Selector: name=front-end

Type: NodePort

IP: 10.107.210.27

Port: <unset> 80/TCP

NodePort: <unset> **30001/TCP**

Endpoints: <none>

Session Affinity: None

Events: <none>

choga88@ubuntu:~$ **kubectl -n sock-shop get svc front-end**

NAME CLUSTER-IP EXTERNAL-IP PORT(S) AGE

front-end **10.107.210.27 <nodes> 80:30001/TCP** 2m

choga88@ubuntu:~$ **kubectl get pods --all-namespaces**

NAMESPACE NAME READY STATUS RESTARTS AGE

kube-system etcd-ubuntu 1/1 Running 0 41m

kube-system kube-apiserver-ubuntu 1/1 Running 0 40m

kube-system kube-controller-manager-ubuntu 1/1 Running 0 41m

kube-system kube-dns-3913472980-l1p9c 3/3 Running 0 41m

kube-system kube-proxy-53822 1/1 Running 0 19m

kube-system kube-proxy-d5csn 1/1 Running 0 41m

kube-system kube-scheduler-ubuntu 1/1 Running 0 41m

kube-system weave-net-7bcfl 2/2 Running 0 40m

kube-system weave-net-jfz2j 2/2 Running 1 19m

sock-shop carts-153328538-qb403 1/1 Running 0 12m

sock-shop carts-db-4256839670-1n268 1/1 Running 0 12m

sock-shop catalogue-114596073-w5vqd 1/1 Running 0 12m

sock-shop catalogue-db-1956862931-xctth 1/1 Running 0 12m

sock-shop front-end-3570328172-f7jr2 1/1 Running 0 11m

sock-shop orders-2365168879-1kllj 1/1 Running 0 11m

sock-shop orders-db-836712666-vq5cc 1/1 Running 0 11m

sock-shop payment-1968871107-1nth7 1/1 Running 0 11m

sock-shop queue-master-2798459664-t2zl6 1/1 Running 0 11m

sock-shop rabbitmq-3429198581-pjw1w 1/1 Running 0 11m

sock-shop shipping-2899287913-d3fg4 1/1 Running 0 11m

sock-shop user-468431046-f20xc 1/1 Running 0 11m

sock-shop user-db-1166754267-5rzpd 1/1 Running 0 11m

[**http://192.168.17.194:30001--**](http://192.168.17.194:30001--)**> (host 접속 ip: 내부 port로 접속)**

