

Ubuntu에서 쿠버네티스(k8s) 설치 가이드

이 문서는 쿠버네티스(k8s) 구축에 대한 정보를 공유하기 위해 작성되었다.

- [Docker 설치](#)
 - 저장소 설정
 - Docker 엔진 설치
- [Kubernetes 설치](#)
 - 설치 전 확인 사항 (master, node)
 - kubelet, kubeadm, kubectl 설치 (master, node)
 - Control-plane 구성 (master only)
 - Pod network 애드온 설치 (master only)
 - Worker node 구성 (node only)
 - Master에서 노드 확인하기 (master)
 - 자동 완성 설정 (master)

Docker 설치

Docker Engine 설치를 위해 다음 절차를 따라 진행한다. 세부적인 가이드는 아래 링크를 참고한다.

- docs.docker.com 가이드 참조

저장소 설정

Docker 설치를 위해 먼저 apt 패키지를 업데이트한다.

```
sudo apt-get update

sudo apt-get install \
  ca-certificates \
  curl \
  gnupg \
  lsb-release
```

Docker's 공식 GPG key 추가하기 위해 다음 절차를 진행한다.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

Stable repository 설정을 위해 다음 명령을 수행한다.

```
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] \
  https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Docker 엔진 설치

최신 버전의 Docker Engine 및 containerd를 설치하기 위해 다음 명령을 수행한다.

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

설치가 완료되면 Docker 버전 확인한다.

```
sudo docker version
```

설치된 버전

- Client: Docker Engine - Community
 - Version: 20.10.13

- Server: Docker Engine - Community
 - Version: 20.10.13

Docker를 서비스 등록 및 실행하기 위해 다음을 수행한다.

```
sudo systemctl enable docker
sudo systemctl start docker
```

Kubernetes 설치

Docker 설치가 완료되었다면 kubernetes 설치를 진행한다. k8s는 kubeadmin을 통해 설치를 진행할 수 있다.

설치 전 확인 사항 (master, node)

각 노드들은 Swap Disable 해야하기 때문에 각 노드별로 다음 명령을 통해 설정한다.

```
swapoff -a && sed -i '/swap/s/^/#/' /etc/fstab

sudo swapoff -a && sudo sed -i '/swap/s/^/#/' /etc/fstab
```

그리고 iptable 설정하기 위해 다음 명령을 수행한다.

```
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
br_netfilter
EOF

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
sudo sysctl --system
```

통신을 위해 방화벽 예외 설정을 수행한다.

```
sudo systemctl stop firewalld
sudo systemctl disable firewalld
```

다음과 같이 방화벽 설정 이후 포트가 오픈되어 있는지 확인한다.

```
telnet 127.0.0.1 6443
```

kubelet, kubeadm, kubectl 설치 (master, node)

쿠버네티스 설치를 진행하기위해 저장소 업데이트 및 필수 패키지 추가한다.

```
sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl
```

구글 클라우드 퍼블릭 키 다운로드를 수행한다.

```
sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
```

쿠버네티스를 설치하기 위해 Kubernetes 저장소 추가한다.

```
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

저장소 업데이트 후 kubelet, kubeadm, kubectl 설치를 순차적으로 진행한다.

```
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
```

쿠버네티스를 서비스 등록 및 재시작을 수행한다.

```
sudo systemctl daemon-reload
sudo systemctl restart kubelet
```

Control-plane 구성 (master only)

control-plane 노드 초기화 - 해당 작업은 반드시 master에서만 진행한다.

```
kubeadm init
```

만약 초기화 진행중에 다음과 같은 에러가 발생가 발생한다면

```
failed with error: Get "http://localhost:10248/healthz"
```

해당 문제 해결책하기 위해 아래와 같이 daemon.json 생성 후에 kubeadm을 reset 후에 다시 init을 수행한다.

하위 작업은 node에도 수행 해준다.

```
sudo mkdir /etc/docker

cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF

sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker

kubeadm reset
```

최종적으로 아래와 같이 node가 접속할 때 사용하는 토큰이 발행된다.

```
kubeadm join 10.0.100.40:6443 --token zbgv72.v9ac8xhexl28xjwp \
--discovery-token-ca-cert-hash sha256:2193f25bad65918edbc41b543e22327741bdd99748b1
```

모든 사용자가 kube 명령어를 사용할 수 있게 하기 위해 다음을 설정한다.

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Pod network 애드온 설치 (master only)

Pod 간의 네트워크 통신 위해 다음 명령을 통해 써드파티 애드온인 Weave Net works 설치 수행한다.

```
kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"
```

Worker node 구성 (node only)

다음 명령을 통해 Worker node를 등록한다.

```
sudo kubeadm join 10.0.100.40:6443 --token zbgv72.v9ac8xhex128xjwp \
  --discovery-token-ca-cert-hash sha256:2193f25bad65918197d7b543e282327741bdd99748b1a6d879e1b4dc
```

```
curvc@k8s-node2:~$ sudo kubeadm join 10.0.100.40:6443 --token zbgv72.v9ac8xhex128xjwp \
  --discovery-token-ca-cert-hash sha256:2193f25bad65918197d7b543e282327741bdd99748b1a6d879e1b4dc
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
W0315 09:55:59.930860 12567 utils.go:69] The recommended value for "resolvConf" in "KubeletConfiguration" is: /run/systemd/resolve/resolv.conf; the provided value is: /run/systemd/resolve/resolv.conf
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

curvc@k8s-node2:~$
curvc@k8s-node2:~$
curvc@k8s-node2:~$
```

Master에서 노드 확인하기 (master)

다음 명령을 수행하여 모든 노드가 연결되었는지 확인한다.

```
kubectl get nodes -o wide
```

```

curvc@k8s-master:~$ kubectl get nodes
NAME          STATUS    ROLES          AGE      VERSION
k8s-master    Ready     control-plane,master  25m      v1.23.4
k8s-node1     Ready     <none>         6m56s    v1.23.4
k8s-node2     Ready     <none>         6m55s    v1.23.4
curvc@k8s-master:~$ kubectl get nodes -o wide
NAME          STATUS    ROLES          AGE      VERSION    INTERNAL-IP    EXTERNAL-IP    OS-IMAGE             KERNEL-VERSION    CONTAINER-RUNTIME
k8s-master    Ready     control-plane,master  25m      v1.23.4    10.0.100.40    <none>         Ubuntu 20.04.4 LTS   5.4.0-104-generic  docker://20.10.13
k8s-node1     Ready     <none>         6m59s    v1.23.4    10.0.100.41    <none>         Ubuntu 20.04.4 LTS   5.4.0-104-generic  docker://20.10.13
k8s-node2     Ready     <none>         6m8s     v1.23.4    10.0.100.42    <none>         Ubuntu 20.04.4 LTS   5.4.0-104-generic  docker://20.10.13
curvc@k8s-master:~$

```

자동 완성 설정 (master)

bash에서 kubectl 명령어를 tab 자동 완성하기 위해 다음을 수행한다.

```

source <(kubectl completion bash)
echo "source <(kubectl completion bash)" >> ~/.bashrc

```