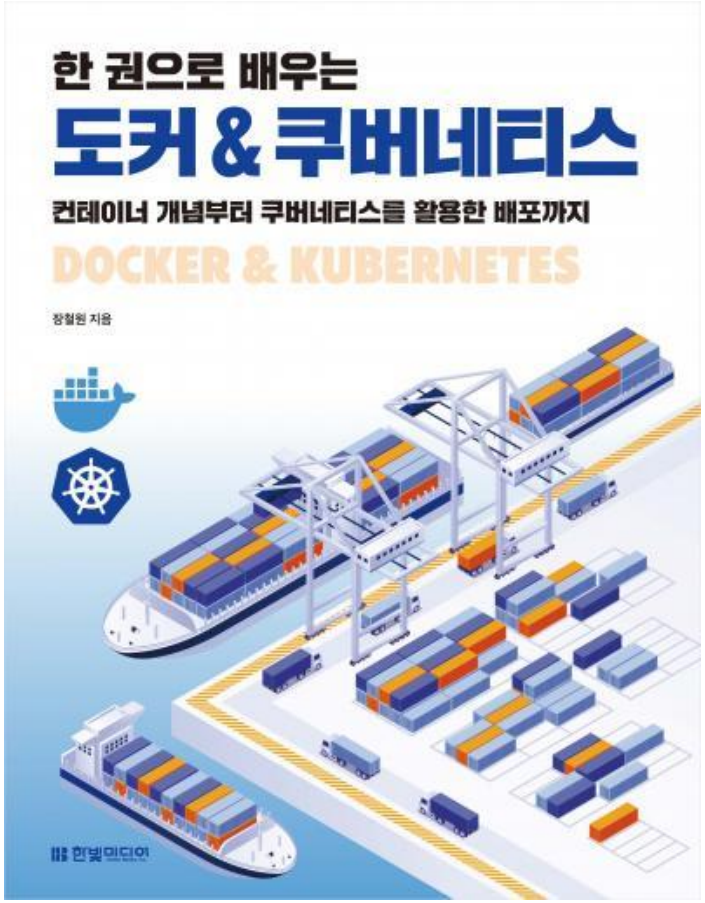
쿠버네티스 설치

Autumn 2024

AI융합학과
Seongbok Baik
sbbaik@dju.ac.kr

00 Textbook



장철원 소프트웨어공학자

01 쿠버네티스(Kubernetes) 설치 순서

1. **시스템 요구 사항 확인**
    - CPU와 메모리: 최소 2 vCPU, 2GB RAM 이상
    - 네트워크 설정: 방화벽과 포트 열기 (6443, 10250 등)
2. **Docker 또는 컨테이너 런타임 설치**
3. **쿠버네티스 패키지 설치**
4. **클러스터 초기화**
5. **네트워크 플러그인 설치**
6. **Worker Node 에서Master Node로 연결**
7. **설치 확인**

02 쿠버네티스 설치 준비

```
eevee@myserver01:~$ sudo apt-get update                                    ❶
eevee@myserver01:~$ sudo apt-get install -y apt-transport-https ca-certificates
curl                                                                       ❷
```

- apt 패키지 목록 업데이트 후 쿠버네티스 설치에 필요한 프로그램 설치

```
eevee@myserver01:~$ sudo mkdir -p /etc/apt/keyrings                        ❶
eevee@myserver01:~$ sudo curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/
deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-
keyring.gpg                                                                ❷

eevee@myserver01:~$ echo "deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /" | sudo tee /etc/apt/sources.list.d/
kubernetes.list                                                            ❸

deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/
core:/stable:/v1.29/deb/ /
```

- 다운로드에 필요한 서명 키 준비 및 k8s 패키지 저장소(repository) 추가

02 쿠버네티스 설치 준비

- apt 패키지 목록 업데이트 후 쿠버네티스 설치에 필요한 프로그램 설치

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

```
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

- 다운로드에 필요한 서명 키 준비 및 k8s 패키지 저장소(repository) 추가

03 쿠버네티스 설치

```
eevee@myserver01:~$ sudo apt-get update                                    ❺
Hit:4 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:1 http://mirror.kakao.com/ubuntu jammy InRelease
...(생략)

eevee@myserver01:~$ sudo apt install -y kubelet kubeadm kubectl
kubectl=1.26.5-00                                                          ❻
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
eevee@myserver01:~$ sudo apt-mark hold kubelet kubeadm kubectl             ❼
kubelet set on hold.
kubeadm set on hold.
kubectl set on hold.
```

버전 자동 지정

- 쿠버네티스 행심 모듈인 kubelet, kubeadm, kubectl을 설치하고 버전 고정

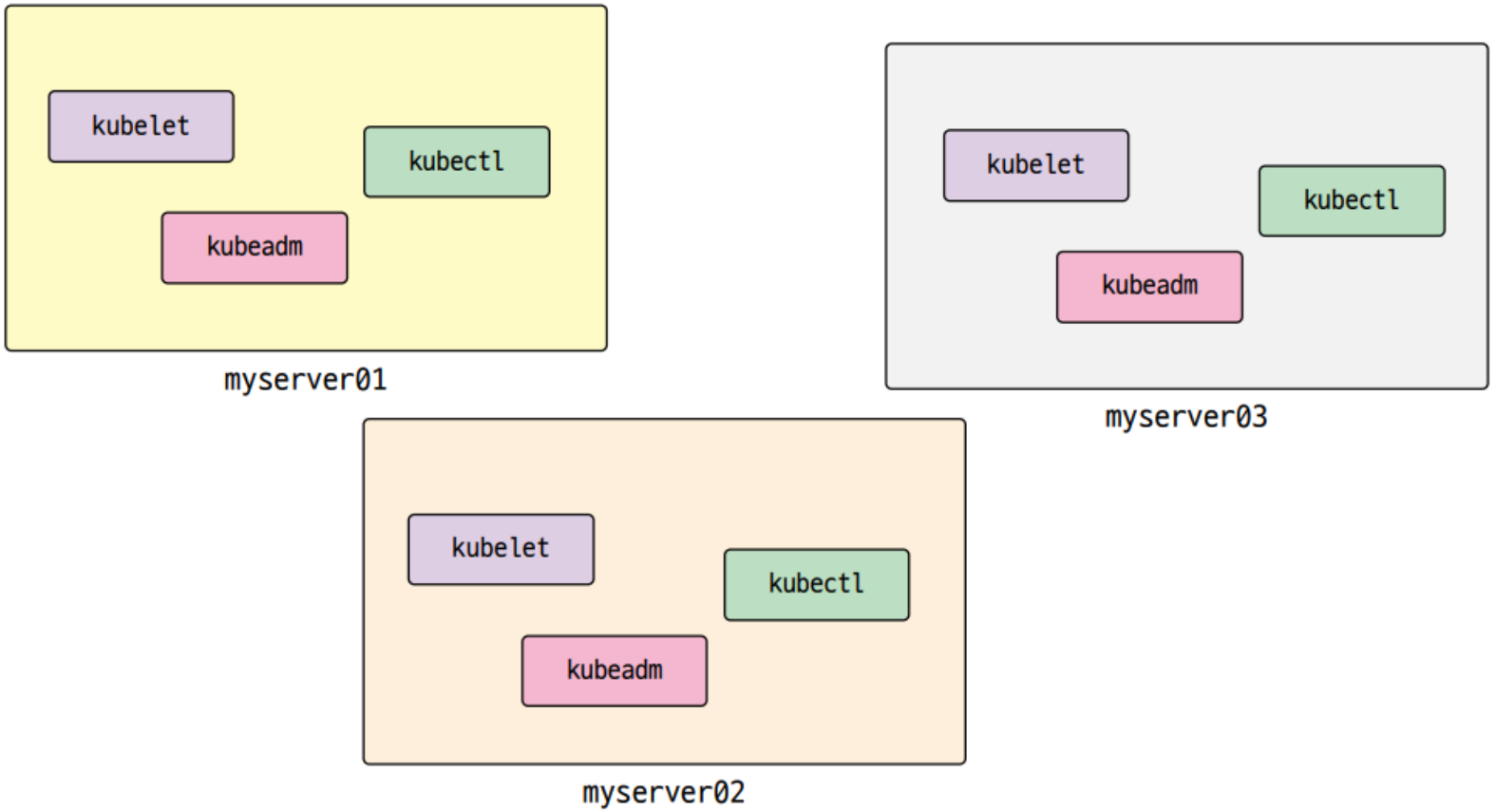
04 쿠버네티스 설치

```
eevee@myserver01:~$ sudo -i                                    ❶
root@myserver01:~# kubelet --version                           ❷
Kubernetes v1.29.5
root@myserver01:~# kubeadm version                             ❸
kubeadm version: &version.Info{Major:"1", Minor:"26", GitVersion:"v1.29.5", GitCommi
t:"890a139214b4de1f01543d15003b5bda71aae9c7", GitTreeState:"clean", BuildDate:"2023-
05-17T14:13:34Z", GoVersion:"go1.19.9", Compiler:"gc", Platform:"linux/amd64"}

root@myserver01:~# kubectl version --output=yaml               ❹
clientVersion:
  buildDate: "2023-05-17T14:14:46Z"
  compiler: gc
  gitCommit: 890a139214b4de1f01543d15003b5bda71aae9c7
  gitTreeState: clean
  gitVersion: v1.29.5
  goVersion: go1.19.9
  major: "1"
  minor: "26"
  platform: linux/amd64
kustomizeVersion: v4.5.7
The connection to the server localhost:8080 was refused - did you specify the right
host or port?
```

- Kubelet, kubeadm, kubectl 설치 확인

05 쿠버네티스 반복 설치



- 동일한 방식으로 myserver01을 포함한 모든 가상머신에 대해 반복 작업
- 마스터 노드: myserver01, 워커 노드: myserver02

06 쿠버네티스 인증서 확인

```
eevee@myserver01:~$ kubeadm certs check-expiration      ❶
CERTIFICATE   EXPIRES   RESIDUAL TIME   CERTIFICATE AUTHORITY   EXTERNALLY MANAGED
!MISSING! admin.conf
!MISSING! apiserver
!MISSING! apiserver-etcd-client
!MISSING! apiserver-kubelet-client
!MISSING! controller-manager.conf
```

```
CERTIFICATE AUTHORITY      EXPIRES   RESIDUAL TIME   EXTERNALLY MANAGED
!MISSING! ca
!MISSING! etcd-ca
!MISSING! front-proxy-ca
```

- 초기에는 해당 노드의 쿠버네티스 관련 인증 처리가 전혀 되어 있지 않음

07 쿠버네티스 인증서 확인

```
eevee@myserver01:~$ kubeadm config images list          ❶

I1105 06:13:59.280667   14008 version.go:256] remote version is much newer: v1.28.3;
falling back to: stable-1.26
registry.k8s.io/kube-apiserver:v1.26.10
registry.k8s.io/kube-controller-manager:v1.26.10
registry.k8s.io/kube-scheduler:v1.26.10
registry.k8s.io/kube-proxy:v1.26.10
registry.k8s.io/pause:3.9
registry.k8s.io/etcd:3.5.6-0
registry.k8s.io/coredns/coredns:v1.9.3
```

- Kubeadm 명령으로 사용 가능한 이미지 목록 출력

08 쿠버네티스 이미지 다운로딩

```
eevee@myserver01:~$ sudo -i                                          ❶
root@myserver01:~# kubeadm config images pull                        ❷
...(중략)
Found multiple CRI endpoints on the host. Please define which one do you wish to use
by setting the 'criSocket' field in the kubeadm configuration file: unix:///var/run/
containerd/containerd.sock, unix:///var/run/cri-dockerd.sock         ❸
To see the stack trace of this error execute with --v=5 or higher
```

- 설치에 필요한 이미지 다운로드 -> 에러 발생
- 에러 내용: 시스템 내에 설치된 CRI (Container Runtime Interface)가 여러 개 존재한다는 의미

09 쿠버네티스 이미지 다운로딩

```
root@myserver01:~# kubeadm config images pull --cri-socket /run/containerd/containerd.
sock                                                                    ❶

W1105 06:16:51.121202   14538 initconfiguration.go:119] Usage of CRI endpoints without
URL scheme is deprecated and can cause kubelet errors in the future. Automatically
prepending scheme "unix" to the "criSocket" with value "/run/containerd/containerd.
sock". Please update your configuration!
I1105 06:16:51.643212   14538 version.go:256] remote version is much newer: v1.28.3;
falling back to: stable-1.26
[config/images] Pulled registry.k8s.io/kube-apiserver:v1.26.10
[config/images] Pulled registry.k8s.io/kube-controller-manager:v1.26.10
[config/images] Pulled registry.k8s.io/kube-scheduler:v1.26.10
[config/images] Pulled registry.k8s.io/kube-proxy:v1.26.10
[config/images] Pulled registry.k8s.io/pause:3.9
[config/images] Pulled registry.k8s.io/etcd:3.5.6-0
[config/images] Pulled registry.k8s.io/coredns/coredns:v1.9.3
```

- --cri-socket 옵션: CRI를 containerd로 고정하여 이미지 pulling

10 쿠버네티스 마스터 노드 초기화

```
root@myserver01:~# kubeadm init --apiserver-advertise-address=10.0.2.4 --pod-network-
cidr=192.168.0.0/16 --cri-socket /run/containerd/containerd.sock        ❶

W0520 03:29:54.665421   53714 initconfiguration.go:119] Usage of CRI endpoints without
URL scheme is deprecated and can cause kubelet errors in the future. Automatically
prepending scheme "unix" to the "criSocket" with value "/var/run/cri-dockerd.sock".
Please update your configuration!
```

... 중략 ...

```
Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 10.0.2.4:6443 --token q56pek.f16j77pe8zpiicke \          ❷
        --discovery-token-ca-cert-hash sha256:8c2fa5dd3d2bcdbe91b5d8d5c3cdb74ff4831f1
c62094a9e9582327fde3a0271
```

- Kubeadm init 명령: 마스터 노드 초기화
  - --apiserver-advertise-address 옵션: 마스터 노드 IP주소 입력
  - --pod-network-cidr: 네트워크 대역 설정
- (2)번 내용: **워커-마스터 연결시 재사용 => 메모장에 저장**

10 Kubeadm init 명령 오류시 조치

```
sudo kubeadm reset
```

```
sudo rm -rf /etc/kubernetes /var/lib/etcd /var/lib/kubelet /etc/cni/net.d
```

```
sudo reboot now
```


Master node IP

```
sudo kubeadm init --apiserver-advertise-address=10.0.2.15 ₩
  --pod-network-cidr=192.168.0.0/16 ₩
  --cri-socket=unix:///run/containerd/containerd.sock
```

11 쿠버네티스 인증 확인

```
root@myserver01:~# kubeadm certs check-expiration      ❶

[check-expiration] Reading configuration from the cluster...
[check-expiration] FYI: You can look at this config file with 'kubectl -n kube-system
get cm kubeadm-config -o yaml'

CERTIFICATE                EXPIRES                  RESIDUAL TIME   CERTIFICATE
AUTHORITY   EXTERNALLY MANAGED
admin.conf                 Nov 04, 2024 06:18 UTC   364d            ca
no
```

… 중략 …

```
 CERTIFICATE AUTHORITY   EXPIRES                  RESIDUAL TIME   EXTERNALLY MANAGED
 ca                      Nov 02, 2033 06:18 UTC   9y              no
 etcd-ca                 Nov 02, 2033 06:18 UTC   9y              no
 front-proxy-ca          Nov 02, 2033 06:18 UTC   9y              no
```

- Kubeadm certs 명령: 쿠버네티스 인증서 상태 확인

12 쿠버네티스 사용자화

```
root@myserver01:~# exit                                                    ❶
logout
eevee@myserver01:~$ mkdir -p $HOME/.kube                                   ❷
eevee@myserver01:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config   ❸
eevee@myserver01:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config        ❹
```

- 일반 사용자 권한으로도 사용할 수 있도록 설정
  - 일반 사용자 $HOME 디렉토리 밑에 쿠버네티스 설정을 저장할 .kube 디렉토리 생성
  - 시스템의 쿠버네티스 설정 파일을 생성된 신규 디렉토리로 복사
  - 설정 디렉토리 소유자와 그룹을 변경하여 현재 사용자가 사용할 수 있도록 변경

13 쿠버네티스 네트워크 설정

```
eevee@myserver01:~$ kubectl create -f https://raw.githubusercontent.com/
projectcalico/calico/v3.26.3/manifests/tigera-operator.yaml          ❶

namespace/tigera-operator created
```

- Calico 설치를 위해 지정된 URL에 위치한 yaml 파일 실행

(*) 우분투가 noble 버전인 경우 :
kubectl apply -f
https://raw.githubusercontent.com/projectcalico/calico/**v3.27.2**/manifests/calico.yaml

14 쿠버네티스 네트워크 설정

```
eevee@myserver01:~$ curl https://raw.githubusercontent.com/projectcalico/calico/
v3.26.3/manifests/custom-resources.yaml -O                          ❷

% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
824  100   824    0     0   2571      0 --:--:-- --:--:-- --:--:--  2575


eevee@myserver01:~$ ls                                              ❸
custom-resources.yaml  work

eevee@myserver01:~$ kubectl create -f custom-resources.yaml         ❹
installation.operator.tigera.io/default created
apiserver.operator.tigera.io/default created
```

- Calico 설치를 위해 커스텀 리소스 설치
- 다운로드한 파일 확인
- 해당 yaml을 활용해 calico 설치

(*) noble버전: curl https://raw.githubusercontent.com/projectcalico/calico/v3.27.2/manifests/custom-resources.yaml -O

15 쿠버네티스 네트워크 설정

```
eevee@myserver01:~$ watch kubectl get pods -n calico-system          ❶
NAME                                       READY   STATUS    RESTARTS   AGE
calico-kube-controllers-7996d5fd5d-fgnbd   1/1     Running   0          108s
calico-node-p27hj                          1/1     Running   0          108s
```

- Calico 설치 완료 후 calico에 대한 파드가 실행 중인지 확인

16 쿠버네티스 네트워크 설정

```
eevee@myserver01:~$ watch kubectl get pods -n calico-system                ❶
NAME                                       READY   STATUS    RESTARTS   AGE
calico-kube-controllers-7996d5fd5d-fgnbd   1/1     Running   0          108s
calico-node-p27hj                          1/1     Running   0          108s
```

- Calico 설치 완료 후 calico에 대한 파드가 실행 중인지 확인

```
eevee@myserver01:~$ kubectl get node -o wide                                ❶
NAME         STATUS   ROLES           AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-
IMAGE             KERNEL-VERSION      CONTAINER-RUNTIME
myserver01   Ready    control-plane   10m   v1.29.5   10.0.2.4      <none>
Ubuntu 22.04.3 LTS   5.15.0-88-generic   containerd://1.6.24              ❷
```

- 쿠버네티스 클러스터 노드 확인

(현재 myserver01로만 구성됨 -> myserver02, myserver03로 점차 확장)

17 쿠버네티스 워커노드 설정

```
eevee@myserver02:~$ mkdir -p $HOME/.kube                                    ❶
eevee@myserver02:~$ scp -p eevee@10.0.2.4:~/.kube/config ~/.kube/config     ❷
eevee@10.0.2.4's password:
config                                        100% 5632     2.2MB/s   00:00
eevee@myserver02:~$ cd .kube/                                               ❸
eevee@myserver02:~/.kube$ ls                                                ❹
config
```

- Myserver02에 설정 파일 복사
  - 쿠버네티스 설정 디렉토리 생성
  - Scp 명령으로 myserver01의 설정 내용 복사
  - .kube 디렉토리에서 config 파일 확인

(*) Scp 명령 형식: scp –p {사용자명@IP주소}:{복사할파일} {붙여넣을파일}

18 쿠버네티스 워커노드-마스터 노드 연결 설정

```
eevee@myserver02:~$ sudo -i                                          ❶

root@myserver02:~# kubeadm join 10.0.2.4:6443 --token q56pek.f16j77pe8zpiicke
--discovery-token-ca-cert-hash sha256:8c2fa5dd3d2bcdbe91b5d8d5c3cdb74ff4831f1c62094a9
e9582327fde3a0271 --cri-socket /run/containerd/containerd.sock       ❷

W1105 06:35:58.487680    6589 initconfiguration.go:119] Usage of CRI endpoints without
URL scheme is deprecated and can cause kubelet errors in the future. Automatically
prepending scheme "unix" to the "criSocket" with value "/run/containerd/containerd.
sock". Please update your configuration!
```

- 워커노드 – 마스터 노드 연계
  - 루트 권한으로 실행
  - 본 강의노트 **"12"번 슬라이드 (2)번 (빨강색 코멘트 참조)** 명령 입력

  단, 맨 끝에 --cri-socket=unix:///run/containerd/containerd.sock 추가

(*) kubeadm join 10.0.2.15:6443 --token gk6pvw.gsfjzetu30aifmqt --discovery-token-ca-cert-hash ₩
sha256:1e906793ac0bfa2b8223076cf1ed43c102de9cdeb540880f45d22b8b930e3000 ₩
--cri-socket=unix:///run/containerd/containerd.sock

19 쿠버네티스 워커노드-마스터 노드 연결 설정

```
eevee@myserver01:~$ kubectl get node
NAME         STATUS   ROLES           AGE     VERSION
myserver01   Ready    control-plane   19m     v1.29.5
myserver02   Ready    <none>          2m37s   v1.29.5
```

- Myserver01에 접속하여 노드 확인 -> myserver02 생성 확인

```
eevee@myserver01:~$ kubectl get nodes
NAME         STATUS   ROLES           AGE     VERSION
myserver01   Ready    control-plane   23m     v1.29.5
myserver02   Ready    <none>          5m58s   v1.29.5
myserver03   Ready    <none>          72s     v1.29.5
```

- Myserver03에 접속하여 myserver02에서 수행한 작업 (슬라이드 “18”, “19”) 반복
- 최종 myserver02와 myserver03 생성 확인

20 쿠버네티스 실행 확인

```
eevee@myserver01:~$ kubectl run hello-world --image=hello-world --restart=Never
pod/hello-world created                          ❶

eevee@myserver01:~$ kubectl get pod              ❷
NAME          READY   STATUS      RESTARTS   AGE
hello-world   0/1     Completed   0          38s
```

- Hello-world 파드 생성 및 실행
  - Restart=Never: 파드가 한번 종료되면 재시작하지 않음

21 쿠버네티스 삭제

```
root@myserver01:~# sudo apt-get purge kubeadm kubectl kubelet   ❶
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  conntrack cri-tools ebtables kubernetes-cni socat
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  kubeadm* kubectl* kubelet*
```

- Purge 명령으로 kube* 패키지 삭제

```
root@myserver01:~# sudo apt-get autoremove                    ❷
```

- Autoremove: 시스템에서 사용하지 않는 패키지 자동 삭제

```
root@myserver01:~# sudo rm -rf ~/.kube                        ❸
```

- 쿠버네티스 설정 관련 .kube 디렉토리 전체 삭제

서로에게, 자신에게 친절합시다 - 허준이

DAEJEON UNIVERSITY
SINCE 1980
대전대학교