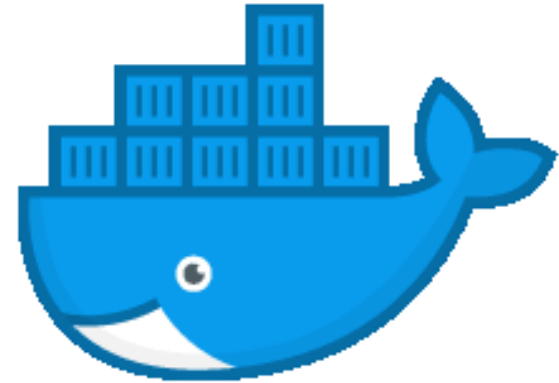
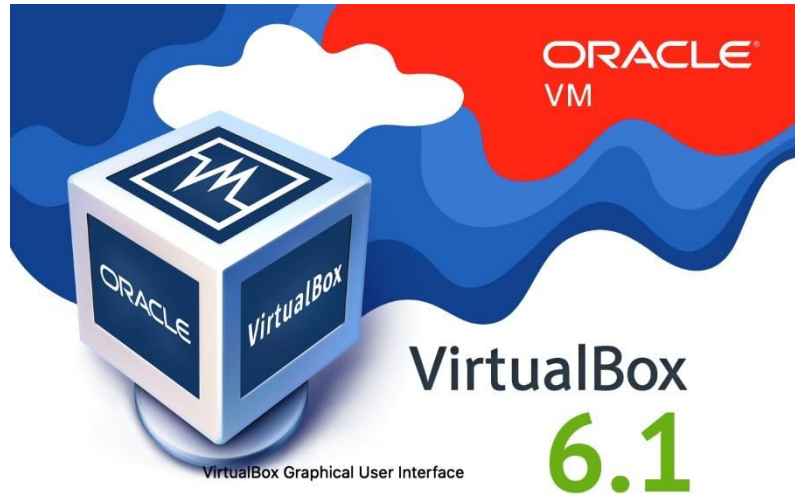


실습 환경 소개



docker



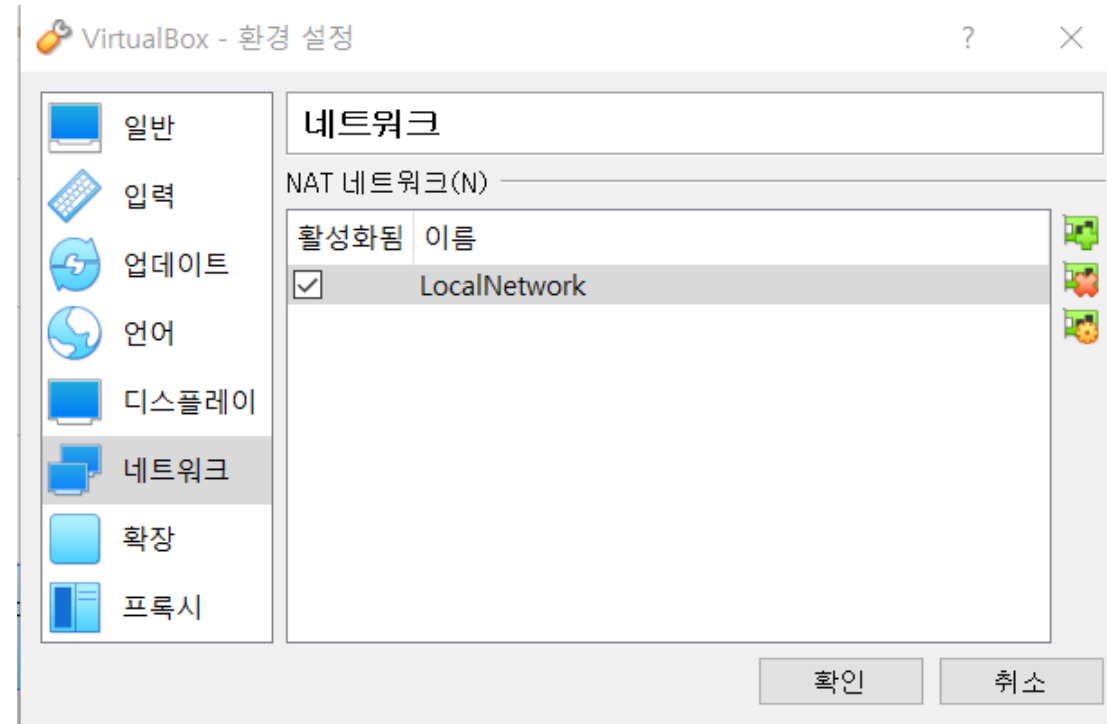
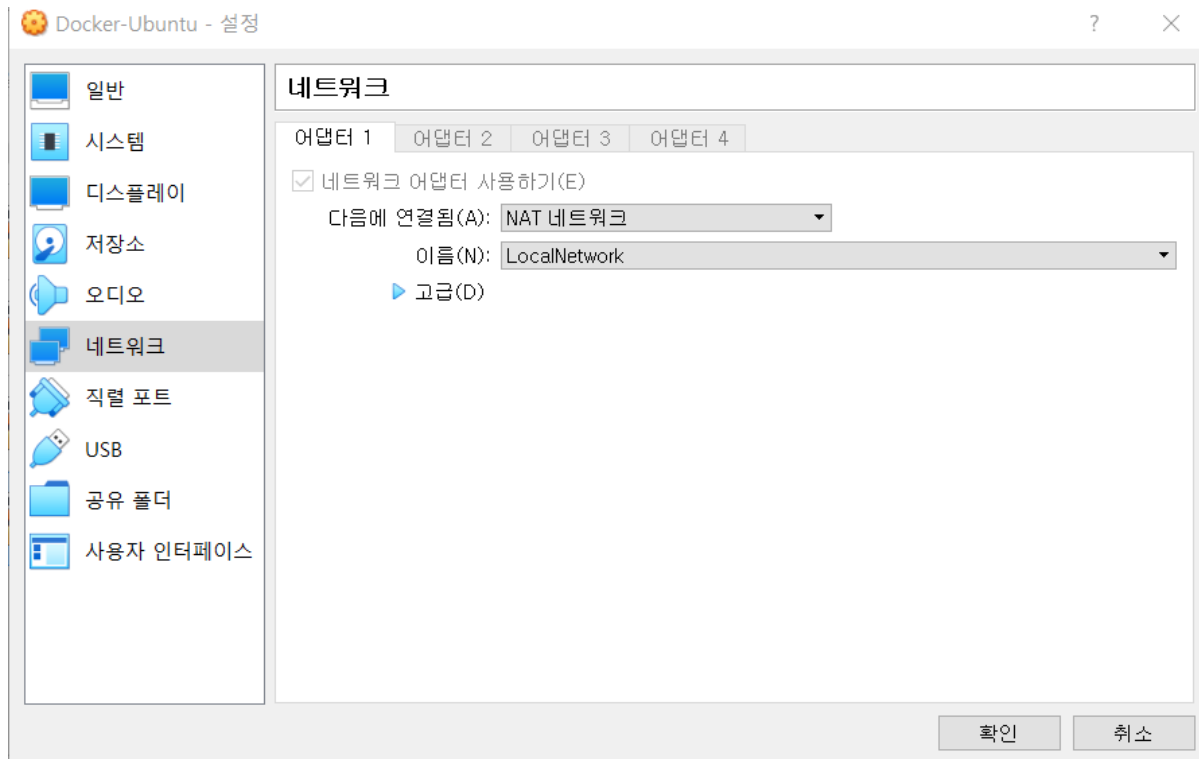
kubernetes

1. 가상 머신 생성 및 운영 환경 구성

▪ 접근 계정

ID : guru / password : work

ID : root / password : password



Docker 설치_Ubuntu #1

■ 패키지 사용 설치 전 확인 사항

- 최신 버전의 커널을 사용하고 있는지 확인
- 호스트 운영체제가 최소한 3.10 버전 이상을 사용해야 도커 컨테이너를 정상적으로 사용 가능

```
$ uname -r  
4.15.0-117-generic
```

- 지원기간 내에 있는 배포판 인지 확인
- 64bit 리눅스 인지 확인
- sudo 명령어를 통해서 설치하거나 root 권한을 소유한 계정으로 설치를 진행

■ 패키지 사용 설치 (ctrl + insert 로 copy , shift + insert로 복사, 단 root 로 로그인)

1) **sudo apt-get update**

2) **sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common**

https를 사용해서 레포지토리를 사용할 수 있도록 필요한 패키지를 설치한다.

3) **curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -**

Docker 공식 리포지토리에서 패키지를 다운로드 받았을 때 위변조 확인을 위한 GPG 키를 추가한다.

Docker 설치_Ubuntu #2

4) apt-key fingerprint

```
/etc/apt/trusted.gpg
-----
pub  rsa4096 2017-02-22 [SCEA]
     9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
uid          [ unknown] Docker Release (CE deb) <docker@docker.com>
sub  rsa4096 2017-02-22 [S]
```

Docker.com 의 GPG 키가 등록됐는지 확인한다.

5) add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu \$(lsb_release -cs) stable"

Docker 공식 저장소를 리포지토리로 등록한다.

6) grep docker /etc/apt/sources.list

```
deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable
```

저장소 등록정보에 기록됐는지 확인한다.

7) apt-get update

리포지토리 정보를 갱신

8) apt-get install -y docker-ce

docker container engine 을 설치한다.

Docker 설치_Ubuntu #3

9) Docker 서비스 상태 확인

```
# systemctl status docker
# docker version
```

root@master:~# systemctl status docker

● docker.service - Docker Application Container Engine
Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
Active: active (running) since Tue 2021-09-21 01:20:52 KST; 1h 55min ago
TriggeredBy: ● docker.socket
Docs: <https://docs.docker.com>
Main PID: 752 (dockerd)
Tasks: 27
Memory: 148.9M
CGroup: /system.slice/docker.service
└─752 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

```
9월 21 01:20:52 master dockerd[752]: time="2021-09-21T01:20:52.147869552+09:00" level=info msg="Daemon has completed initialization"
9월 21 01:20:52 master systemd[1]: Started Docker Application Container Engine.
9월 21 01:20:52 master dockerd[752]: time="2021-09-21T01:20:52.304265097+09:00" level=info msg="API listen on /run/docker.sock"
9월 21 01:22:59 master dockerd[752]: time="2021-09-21T01:22:59.975383400+09:00" level=info msg="ignoring event" container=5b19bbdaa10fdeebb4d1ea38abd18c11d>
9월 21 01:23:46 master dockerd[752]: time="2021-09-21T01:23:46.768029930+09:00" level=info msg="ignoring event" container=0395cb4c1e26309161b1df2d89e4c89e3>
9월 21 01:24:13 master dockerd[752]: time="2021-09-21T01:24:13.129800955+09:00" level=error msg="Not continuing with pull after error: context canceled"
9월 21 02:00:55 master dockerd[752]: time="2021-09-21T02:00:55.178263470+09:00" level=info msg="ignoring event" container=0f7d56a3df64df9fd20c697719374e2d2>
9월 21 02:01:20 master dockerd[752]: time="2021-09-21T02:01:20.765285494+09:00" level=info msg="ignoring event" container=3a7e0eec1a28d282b41f7474f84a6445a>
```

```
root@docker-ubuntu:~# docker version
Client: Docker Engine - Community
Version:      20.10.8
API version:  1.41
Go version:   go1.16.6
Git commit:   3967b7d
Built:        Fri Jul 30 19:54:27 2021
OS/Arch:      linux/amd64
Context:      default
Experimental: true

Server: Docker Engine - Community
Engine:
Version:      20.10.8
API version:  1.41 (minimum version 1.12)
Go version:   go1.16.6
Git commit:   75249d8
Built:        Fri Jul 30 19:52:33 2021
OS/Arch:      linux/amd64
Experimental: false
containerd:
Version:      1.4.10
GitCommit:    8848fdb7c4ae3815afcc990a8a99d663dda1b590
runc:
Version:      1.0.2
GitCommit:    v1.0.2-0-g52b36a2
docker-init:
Version:      0.19.0
GitCommit:    de40ad0
root@docker-ubuntu:~#
```