

1. 도커 HUB(도커 사이트와 동일) 가입 정보

dwoseven@daum.net

kim19ht69*

2. 도커 컴포넌트

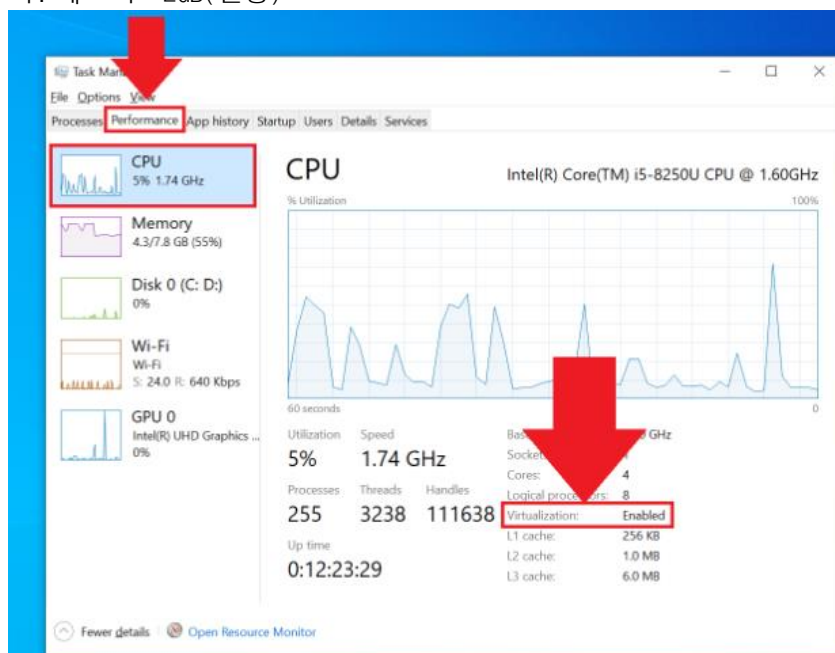
- 가. Docker for Mac - It allows one to run Docker containers on the Mac OS.
- 나. Docker for Linux - It allows one to run Docker containers on the Linux OS.
- 다. Docker for Windows - It allows one to run Docker containers on the Windows OS.
- 라. **Docker Engine** - It is used for building Docker images and creating Docker containers.
- 마. **Docker Hub** - This is the registry which is used to host various Docker images.
- 바. **Docker Compose** - This is used to define applications using multiple Docker containers.

3. 공식 도커 사이트: <https://www.docker.com>

4. 윈도우용 도커: 윈도우를 지원하기 위한 환경

가. 운영체제: 윈도우 10 64비트

나. 메모리: 2GB(권장)

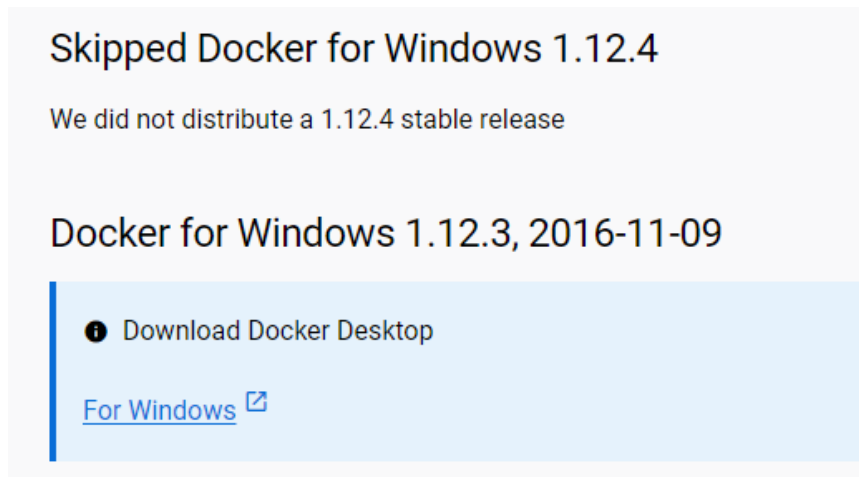


5. 도커 툴박스(ToolBox)

- 가. Docker ToolBox has been designed for older versions of Windows, such as Windows 8.1 and Windows 7. You need to have the following configuration in order to install Docker for Windows.
- 나. 운영체제: 윈도우 7, 8, 8.1
- 다. 메모리: 2GB(권장)
- 라. 가상화: 활성화되어야 함
- 마. 툴박스: D:\다운로드\0.에이콘\홍대강의\99.별도교재\02.홍대제외강좌\The Ultimate Docker Course\수업환경\ DockerToolbox-19.03.1.exe

6. 도커 인스톨

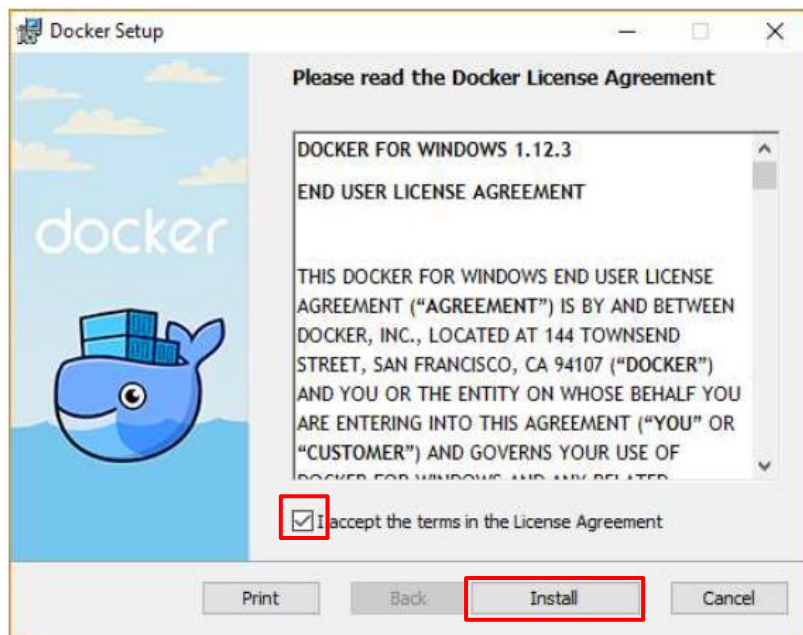
가. 사이트: <https://docs.docker.com/desktop/previous-versions/archive-windows/>



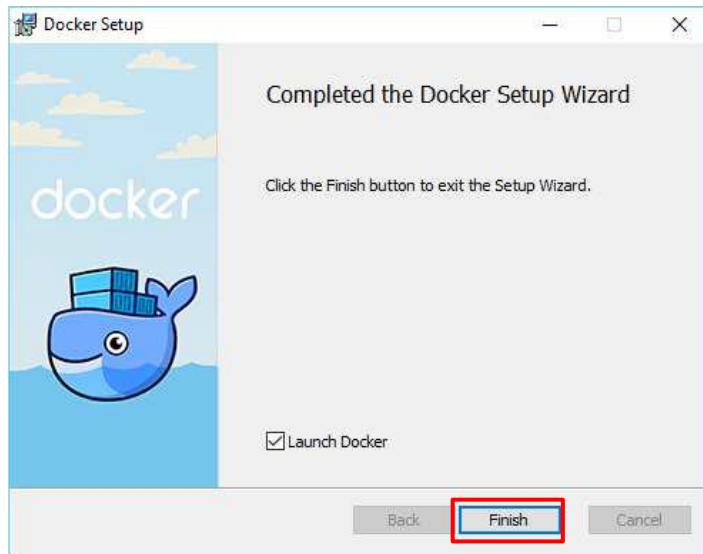
나. 다운받은 위치: D:\다운로드\0.에이콘\홍대강의\09. 별도교재\02. 홍대제외강좌\The Ultimate Docker Course\교재\InstallDocker.msi

7. 도커 설치 단계

가. 1단계



나. 2단계

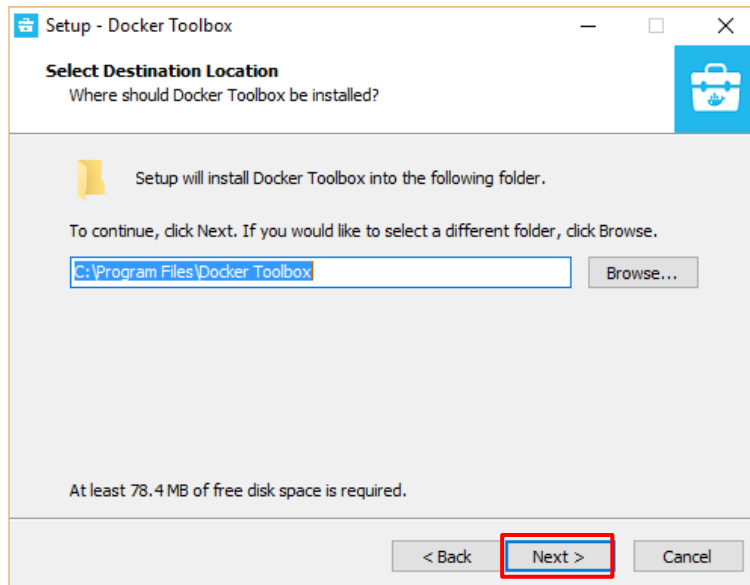


8. 도커 툴박스

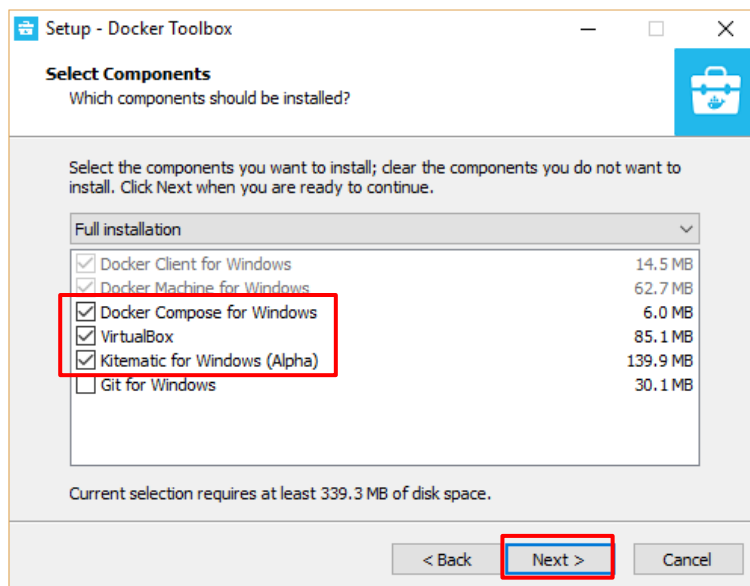
가. 1단계



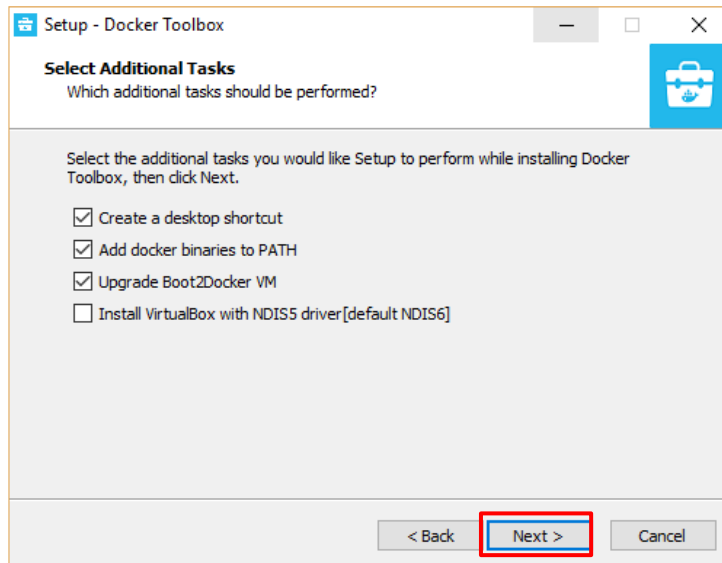
나. 2단계



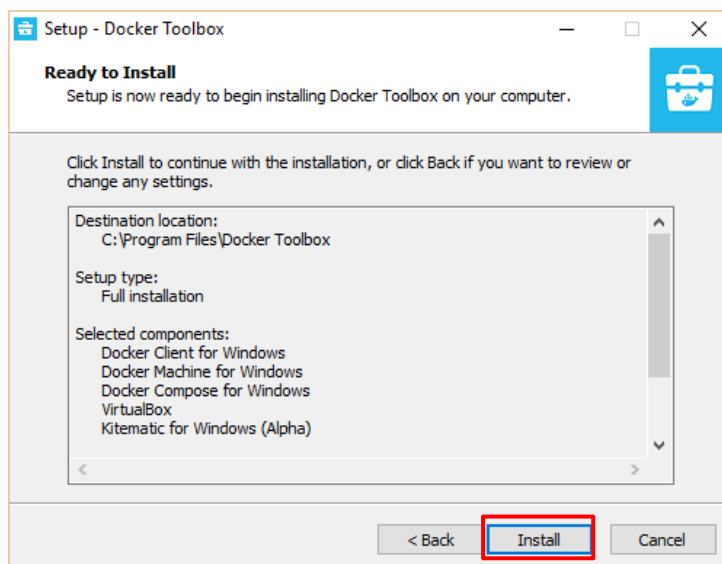
다. 3단계



라. 4단계

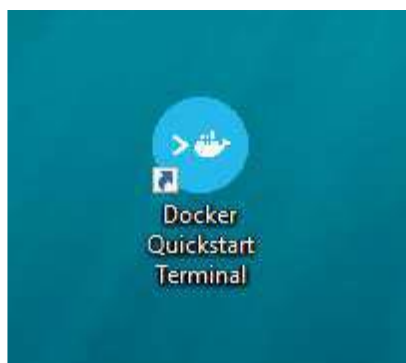


마. 5단계



9. 도커 툴박스 작업하기

가. 실행



다. 우분투 이미지 실행

- docker run -it(interactive mode) Ubuntu bash

```
root@3bba5a5155b8: /
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

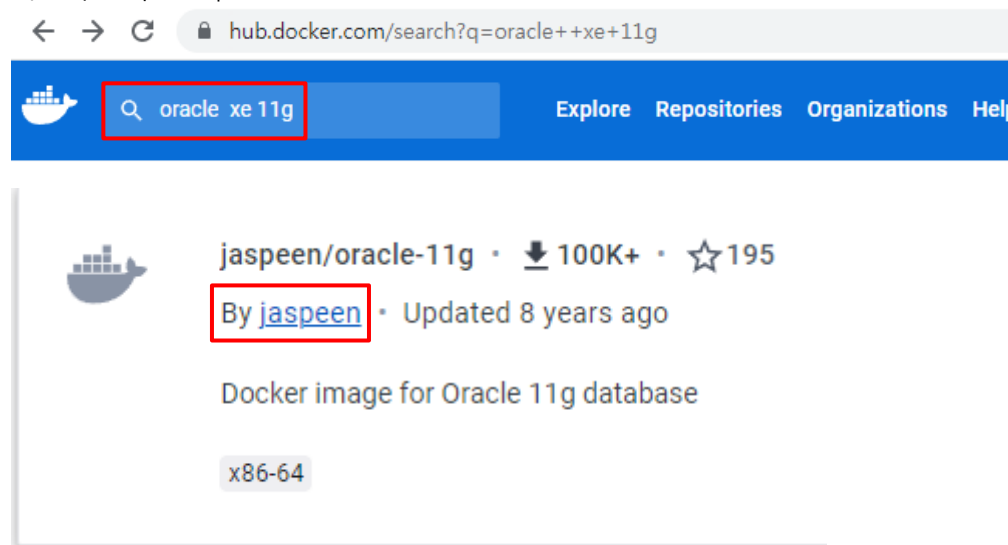
Share images, automate workflows, and more with a free Docker Hub account:
https://hub.docker.com

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/

s362692@DESKTOP-OMT61NR MINGW64 ~
$ docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
af49a5ceb2a5: Pull complete
8f9757b472e7: Pull complete
e931b117db38: Pull complete
47b5e16c0811: Pull complete
9332eaf1a55b: Pull complete
Digest: sha256:3b64c309deae7ab0f7dbdd42b6b326261ccd6261da5d88396439353162703fb5
Status: Downloaded newer image for ubuntu:latest
root@3bba5a5155b8: /#
```

10. 도커 - 도커 허브

- 1) 클라우드 상에 있는 누구나 다운로드 할 수 있는 다른 커뮤니티에 의해 사전 빌트된 도커 이미지 등록 서비스
 - 2) 도커 허브 사이트: <https://www.docker.com/products/docker-hub>
 - 3) 도커 허브로부터 오라클 xe 11g 도커 이미지를 다운로드하는 방법
- 가. 도커 허브 접속
- 나. 싸인 후 도커 로그인





jaspeen

Community User

Joined July 1, 2014



jaspeen/oracle-xe-11g · 50K+ · 8

By jaspeen · Updated 8 years ago

Image

Docker Pull Command

```
docker pull jaspeen/oracle-xe-11g
```



See description on <https://hub.docker.com/r/sath89/oracle-xe-11g/> Removed VOLUME with database files from Dockerfile to be able easily make database snapshots without using impdp/expdp.

설치 방법 인터넷 파일

<https://fomaios.tistory.com/entry/Oracle-Mac-OS%EC%97%90-Oracle-%EC%84%A4%EC%B9%98%ED%95%98%EB%8A%94-%EB%B0%A9%EB%B2%95>

다. 이미지 다운로드

```
:~ mac$ docker pull jaspeen/oracle-xe-11g
```

라. 이미지 확인

```
macui-MacBookPro:~ mac$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED      SIZE
jaspeen/oracle-xe-11g latest      52fbd1fe2d7a  5 years ago  792MB
```

마. 컨테이너를 만들고 실행

```
macui-MacBookPro:~ mac$ docker run --name fomagran -d -p 8080:8080 -p 1521:1521
jaspeen/oracle-xe-11g
955418e39a33 0.0.0.0:1521->1521/tcp, 0.0.0.0:8080->8080/tcp fomagran
```

바. 컨테이너 프로세스 확인

```
macui-MacBookPro:~ mac$ docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED        STATUS        PORTS
955418e39a33   jaspeen/oracle-xe-11g "/entrypoint.sh "       17 seconds ago Up 1
6 seconds     0.0.0.0:1521->1521/tcp, 0.0.0.0:8080->8080/tcp fomagran
```


사. sqlplus 실행: system/oracle(?)

docker exec -it 컨테이너아이디 또는 이름 /bin/bash

Sqlplus sys as sysdba

사. -p: 적절하게 컨테이너에 접속할 수 있도록 내부 도커 이미지와 포트 번호를 우분투 서버에 매핑

11.도커 - 이미지

1) 도커에서 모든 것은 이미지에 기반을 둠

2) 이미지는 파일 시스템과 파라미터 조합

3) 예: docker run hello-world

4) 도커 명령

가. 구체적이고, 운영체제 상의 도커 프로그램에게 해야 할 것을 알려줌

나. run 명령은 이미지의 인스턴스를 만들려고 한다는 것을 알리는데 사용하며, 그 시점에서 컨테이너가 호출됨

다. 예: docker run centos(이미지 이름) -it(대화모드) /bin/bash(bash shell 사용)

5) 도커 이미지 보기

가. docker images

```
demo@ubuntuserver:~$ sudo docker images
[sudo] password for demo:
REPOSITORY          TAG                 IMAGE ID            CREATED
VIRTUAL SIZE
newcentos            latest             7a86f8ffcb25       9 days ago
196.5 MB
jenkins              latest             998d1854867e       2 weeks ago
214.1 MB
centos               latest             97cad5e16cb6       4 weeks ago
196.5 MB
demo@ubuntuserver:~$ _
```

나. TAG: 논리적으로 이미지들 태그하는데 사용

다. Image ID: 유니크하게 이미지를 식별하는데 사용

라. Created: 이미지가 생성된 날로부터 흘러온 날

마. Virtual Size: 이미지 크기

6) 도커 이미지 다운로드

가. docker run image

나. 결과

```
demo@ubuntuserver:~$ sudo docker run centos
Unable to find image 'centos:latest' locally
latest: Pulling from centos

3690474eb5b4: Pull complete
af0819ed1fac: Pull complete
05fe84bf6d3f: Pull complete
97cad5e16cb6: Pull complete
Digest: sha256:934ff980b04db1b7484595bac0c8e6f838e1917ad3a38f904ece64f70bbca040
Status: Downloaded newer image for centos:latest
demo@ubuntuserver:~$ _
```

7) 도커 이미지 삭제

가. docker rmi

나. docker rmi 이미지 아이디

다. 예: docker rmi 7a86f8ffcb25

```
demo@ubuntu:~$ sudo docker rmi 7a86f8ffcb25
Untagged: newcentos:latest
Deleted: 7a86f8ffcb258e42c11d971a04b1145151b80122e566bc2b544f8fc3f94caf1e
demo@ubuntu:~$
```

라. 이미지 아이디만 리턴: docker images -q

```
demo@ubuntu:~$ sudo docker images -q
998d1854867e
97cad5e16cb6
demo@ubuntu:~$ _
```

8) 이미지와 컨테이너의 상세한 사항을 보는데 사용

가. Docker inspect repository(이미지 이름)

나. 예: docker inspect Jenkins

```
{
  "Hostname": "6b3797ab1e90",
  "Image": "sha256:532b1ef702484a402708f3b65a61e6ddf307bbf2fdfa01be553bc6c
a7678ce6c",
  "Labels": {},
  "MacAddress": "",
  "Memory": 0,
  "MemorySwap": 0,
  "NetworkDisabled": false,
  "OnBuild": [],
  "OpenStdin": false,
  "PortSpecs": null,
  "StdinOnce": false,
  "Tty": false,
  "User": "jenkins",
  "Volumes": {
    "/var/jenkins_home": {}
  },
  "WorkingDir": ""
},
{
  "Created": "2016-11-16T20:52:37.568557509Z",
  "DockerVersion": "1.12.3",
  "Id": "998d1854867eb7873a9f45ff4c3ab25bcf5378c77fc955d344e47cb27e5df723",
  "Os": "linux",
  "Parent": "983246da862f43a967b36cc2fc1af580df3f79760dfd841c1954e73253014ece"
},
{
  "Size": 5960,
  "VirtualSize": 714121162
}
]
demo@ubuntu:~$ _
```

12.도커 - 컨테이너

- 1) Run 명령어를 사용해서 실행할 수 있는 도커 이미지의 인스턴스
- 2) 컨테이너를 실행
- 3) 컨테이너 실행시키기

가. docker run -it centos /bin/bash

```
demo@ubuntu:~$ sudo docker run -it centos /bin/bash
[root@9f215ed0b0d3 /]#
```

- 4) 현재 실행중인 컨테이너 리스팅

가. docker ps

```
demo@ubuntu:~$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED
STATUS        PORTS      NAMES                   Up About a minute
9f215ed0b0d3   centos:latest   "/bin/bash"             cocky_colden
```

- 5) 모든 컨테이너 리스팅

가. docker ps -a

```
demo@ubuntu:~$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED
STATUS        PORTS      NAMES                   Up 4 minutes
9f215ed0b0d3   centos:latest   "/bin/bash"             cocky_colden
e5a02936065a   centos:latest   "/bin/bash"            Exited (0) 39 minutes ago
ecstatic_hodgkin
9b286dd1f16a   jenkins:latest  "/bin/tini -- /usr/l    18 hours ago
Exited (0) About an hour ago  0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000
cp_jolly_wright
3646aa260a2d   jenkins:latest  "/bin/tini -- /usr/l    9 days ago
Exited (0) 9 days ago        0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000
cp_reverent_morse
demo@ubuntu:~$ _
```

- 6) 컨테이너를 통해 이미지를 실행한 모든 명령어 보여주기

가. docker history 이미지 아이디

나. 예: docker history centos

```
demo@ubuntu:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED
VIRTUAL SIZE
jenkins       latest   998d1854867e   2 weeks ago
714.1 MB
centos        latest   97cad5e16cb6   4 weeks ago
196.5 MB
demo@ubuntu:~$ sudo docker history centos
IMAGE        SIZE      CREATED        CREATED BY
97cad5e16cb6  0 B       4 weeks ago    /bin/sh -c #(nop) CMD ["/bin/bash"]
05fe84bf6d3f  0 B       4 weeks ago    /bin/sh -c #(nop) LABEL name=CentOS B
e Ima  0 B
af0819ed1fac  0 B       4 weeks ago    /bin/sh -c #(nop) ADD file:54df3580ac9
66389  196.5 MB  3 months ago   /bin/sh -c #(nop) MAINTAINER https://
thub.  0 B
demo@ubuntu:~$ _
```

13.도커 – 컨테이너와 작업하기

1) docker top: 컨테이너 내에 존재하는 최상위 프로세스들 보여주기

가. docker top 컨테이너 아이디

나. 예: docker top 9f215ed0b0d3

```
demo@ubuntuserver:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
9f215ed0b0d3       centos:latest      "/bin/bash"        12 minutes ago
Up 12 minutes
cocky_colden
demo@ubuntuserver:~$ sudo docker top 9f215ed0b0d3
PID                PPID              C
TIME              CMD
root              678              0
18:13             pts/0            00:00:00           /bin/bash
demo@ubuntuserver:~$
```

2) docker stop: 실행중인 컨테이너 중지

가. docker stop 컨테이너아이디

나. 예: docker stop 9f215ed0b0d3

```
demo@ubuntuserver:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
9f215ed0b0d3       centos:latest      "/bin/bash"        22 minutes ago
Up 22 minutes
cocky_colden
demo@ubuntuserver:~$ sudo docker stop 9f215ed0b0d3
9f215ed0b0d3
demo@ubuntuserver:~$ sudo docker rm 9f215ed0b0d3
9f215ed0b0d3
demo@ubuntuserver:~$ _
```

3) docker rm: 컨테이너 삭제

가. docker rm 컨테이너 아이디

나. docker rm 9f215ed0b0d3

```
demo@ubuntuserver:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
9f215ed0b0d3       centos:latest      "/bin/bash"        22 minutes ago
Up 22 minutes
cocky_colden
demo@ubuntuserver:~$ sudo docker stop 9f215ed0b0d3
9f215ed0b0d3
demo@ubuntuserver:~$ sudo docker rm 9f215ed0b0d3
9f215ed0b0d3
demo@ubuntuserver:~$ _
```

4) docker stats: 실행중인 컨테이너에 대한 통계 정보 제공

가. docker stats 컨테이너아이디

나. docker rm 9f215ed0b0d3

CONTAINER	CPU %	MEM USAGE/LIMIT	MEM %
NET I/O			
07b0b6f434fe	0.00%	416 KiB/1.416 GiB	0.03%
648 B/648 B			

5) docker attach: 실행중인 컨테이너를 attach

가. docker attach 컨테이너 아이디

나. docker attach 07b0b6f434fe

```
demo@ubuntuserver:~$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED
STATUS        PORTS     NAMES
07b0b6f434fe   centos:latest  "/bin/bash"            3 minutes ago
Up 3 minutes
demo@ubuntuserver:~$ sudo docker attach 07b0b6f434fe

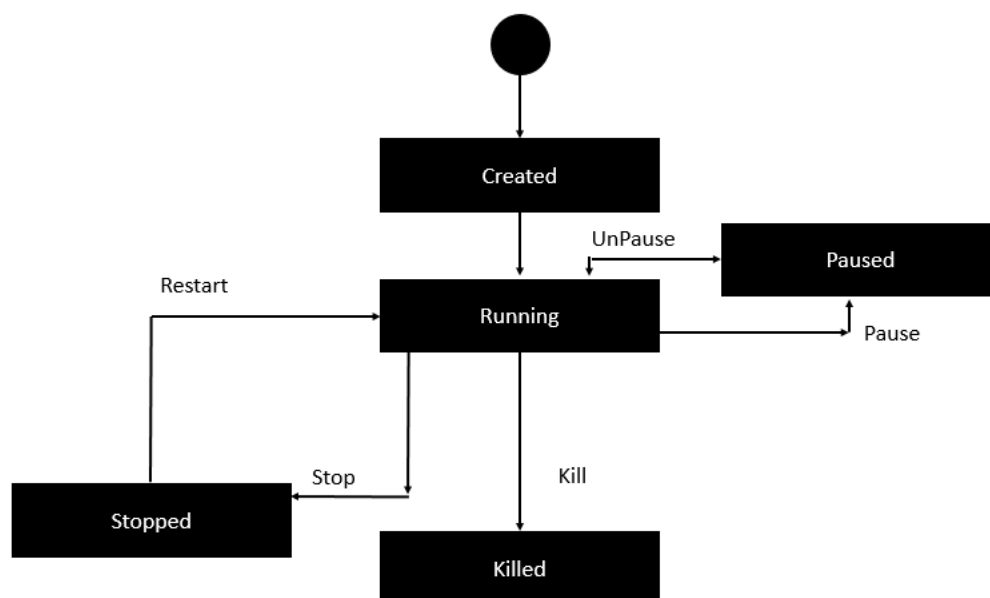
[root@07b0b6f434fe /]# _
```

6) docker pause: 실행중인 컨테이너내의 프로세스들을 중지: docker pause
컨테이너아이디

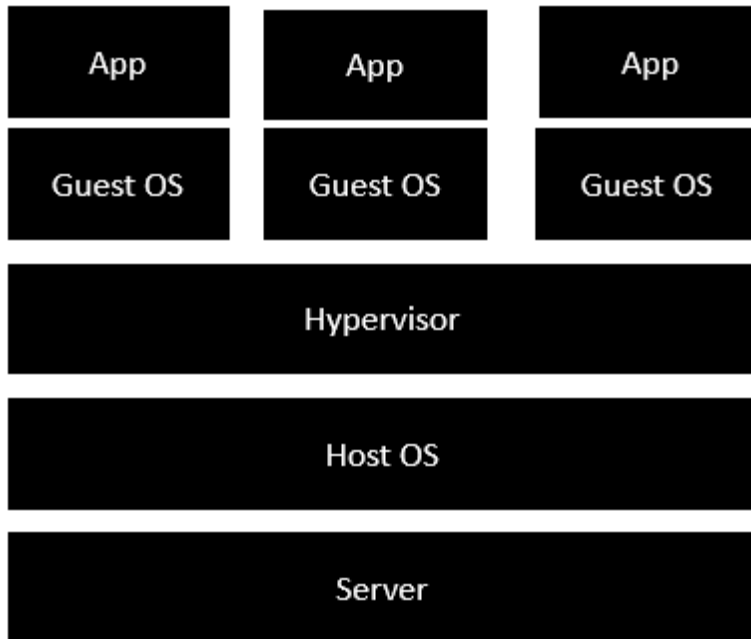
7) docker unpause: 실행중인 컨테이너내의 프로세스들을 unpause: docker unpause
컨테이너아이디

8) docker kill: 실행중인 컨테이너내의 프로세스들 죽이기: docker kill 컨테이너아이디

14. 컨테이너 라이프사이클



15.도커 - 도커 아키텍처



가. 서버: 여러 개의 가상머신을 호스트 하는데 사용하는 물리적 서버

나. Host OS: 리눅스나 윈도우 같은 기본 머신

다. Hypervisor: 가상머신을 호스트 하는데 사용하는 VMWare 또는 Windows Hyper V

16.도커 - 도커 설정

1) 도커 서비스 중지: 도커 데몬 프로세스 중지

가. service docker stop

```
demo@ubuntuserver:~$ sudo service docker stop
[sudo] password for demo:
docker stop/waiting
demo@ubuntuserver:~$
```

2) 도커 서비스 시작: 도커 데몬 프로세스 시작

가. service docker start

```
demo@ubuntuserver:~$ sudo service docker stop
[sudo] password for demo:
docker stop/waiting
demo@ubuntuserver:~$ sudo service docker start
docker start/running, process 1942
demo@ubuntuserver:~$ _
```