

## 02. Docker 명령어 [Ubuntu 20.04]

### 1. Docker 엔진 버전 확인

```
# docker -v
```

```
root@docker-01:~# docker -v
Docker version 20.10.5, build 55c4c88
root@docker-01:~#
```

### 2. Docker Run / 컨테이너 생성 및 실행

```
# docker run -i -t ubuntu:20.04
```

```
root@docker-01:~# docker run -i -t ubuntu:20.04
Unable to find image 'ubuntu:20.04' locally
20.04: Pulling from library/ubuntu
5d3b2c2d21bb: Pull complete
3fc2062ea667: Pull complete
75adf526d75b: Pull complete
Digest: sha256:b4f9e18267eb98998f6130342baacaeb9553f136142d40959a1b46d6401f0f2b
Status: Downloaded newer image for ubuntu:20.04
root@10029779321b:/#
```

- Docker run 명령어는 컨테이너 생성 및 실행

- ubuntu 20.04 이미지가 로컬 도커 엔진에 존재하지 않으므로 도커 중앙 이미지 저장소인 **도커 허브**에서 자동으로 이미지를 내려받음

- 현재 컨테이너 내부로 들어온 상태 (호스트 네임변경)

> 호스트네임은 무작위의 16 진수 해시값

> -i 옵션 : 상호 입출력 설정

> -t 옵션 : tty 를 활성화하여 배시(bash) 셸을 사용하도록 컨테이너 설정

> 위 두가지 옵션을 하나라도 사용하지 않으면 정상적으로 셀사용 불가

### 3. ls 명령어로 컨테이너 내부 확인

# ls

```
root@10029779321b:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@10029779321b:/#
```

### 4. Docker 컨테이너 빠져나가기

- Docker 컨테이너 나오는 방법 2 가지

#### 1) Shell 에 exit 입력 or Ctrl + D 입력

> 대신 위 방법은 컨테이너 내부에서 빠져 나오면서 동시에 컨테이너를 정지

#### 2) Ctrl + P,Q 입력

> 단순히 컨테이너의 셸에서만 빠져나오기 때문에 컨테이너 애플리케이션을 개발하는 목적으로 컨테이너를 사용할때 이 방법을 많이 씀

```
root@10029779321b:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@10029779321b:/# exit
exit
root@docker-01:~# docker
```

### 5. Docker pull 로 Image 내려받기

# docker pull centos:8

```

root@docker-01:~# docker pull centos:8
8: Pulling from library/centos
7a0437f04f83: Pull complete
Digest: sha256:5528e8b1b1719d34604c87e11dcd1c0a20bedf46e83b5632cdeac91b8c04efc1
Status: Downloaded newer image for centos:8
docker.io/library/centos:8
root@docker-01:~#

```

- **docker pull** 명령어는 도커 이미지만 다운로드 한다.

- **docker run** 은 이미지 다운로드 및 실행까지 동시에 한다.

# docker images

```

root@docker-01:~# docker images
REPOSITORY    TAG        IMAGE ID      CREATED       SIZE
ubuntu        20.04      4dd97cefde62  2 weeks ago   72.9MB
centos        8          300e315adb2f  3 months ago  209MB
root@docker-01:~#

```

- docker images 명령어를 통해 다운로드 한 docker image 들을 확인한다.

## 6. docker create 명령어로 컨테이너 생성

# docker create -i -t --name mycentos centos:8

```

root@docker-01:~# docker create -i -t --name mycentos centos:8
65734c1a35eac6aa2cd7a5ea95d2f30752e35db3f23352044709d8aed19424ab
root@docker-01:~#
root@docker-01:~#

```

- 컨테이너를 생성할 때는 run 명령어가 아닌 create 명령어를 사용할 수도 있다.

> **--name** 옵션에는 컨테이너의 이름을 설정

- create 명령어의 결과로 출력된 무작위의 16 진수 해시값은 컨테이너의 고유 ID

> 너무길어 일반적으로 12 자리만 사용

> **docker inspect** 명령어로 컨테이너의 ID 를 다시 확인가능

```
# docker inspect mycentos
```

```
# docker inspect mycentos | grep Id
```

```
root@docker-01:~# docker inspect mycentos
[
  {
    "Id": "65734c1a35eac6aa2cd7a5ea95d2f30752e35db3f23352044709d8aed19424ab",
    "Created": "2021-03-23T06:16:10.636869965Z",
    "Path": "/bin/bash",
    "Args": [],
    "State": {
      "Status": "created",
      "Running": false,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 0,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "0001-01-01T00:00:00Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:300e315adb2f96afe5f0b2780b87f28ae95231fe3bdd1e16b9ba606307728f55",
  },
]
root@docker-01:~# docker inspect mycentos | grep Id
    "Id": "65734c1a35eac6aa2cd7a5ea95d2f30752e35db3f23352044709d8aed19424ab",
root@docker-01:~#
```

## 7. docker start 컨테이너 시작

```
root@docker-01:~# docker start mycentos
mycentos
root@docker-01:~#
```

- docker start 명령어로 만들어 놓은 컨테이너 실행

## 8. docker attach 컨테이너 내부로 들어가기

```
root@docker-01:~# docker attach mycentos
[root@65734c1a35ea /]# ls
bin dev etc home lib lib64 lost+found media mnt opt proc root run sbin srv sys tmp usr var
[root@65734c1a35ea /]#
```

- docker attach 명령어로 실행중인 컨테이너 shell 접속

## 9. Docker 컨테이너 목록 확인

```
# docker ps
```

```
# docker ps -a
```

```

root@docker-01:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
65734c1a35ea   centos:8  "/bin/bash"             2 hours ago   Up 58 minutes                mycentos
root@docker-01:~# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
65734c1a35ea   centos:8  "/bin/bash"             2 hours ago   Up About an hour                mycentos
10029779321b   ubuntu:20.04  "/bin/bash"             3 hours ago   Exited (0) 2 hours ago        beautiful_tharp
root@docker-01:~#

```

- **docker ps** : 정지되지 않은 컨테이너 목록을 보여줌

- **docker ps -a** : 정지된 컨테이너를 포함한 모든 컨테이너 출력

- 컨테이너 상태는 STATUS 항목에서 확인

> Exited : 정지된 상태

> Up .. seconds : 실행 중인 상태

## 9.1 docker ps 출력 결과 설정

```

root@docker-01:~# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
d382291764e5   ubuntu:20.04  "echo hello world!"     4 minutes ago   Exited (0) 4 minutes ago        my_container
65734c1a35ea   centos:8     "/bin/bash"             20 hours ago   Up 19 hours                mycentos
10029779321b   ubuntu:20.04  "/bin/bash"             21 hours ago   Exited (0) 20 hours ago        beautiful_tharp
root@docker-01:~# docker ps --format "table {{.ID}}\t{{.Status}}\t{{.Image}}"
CONTAINER ID   STATUS        IMAGE
65734c1a35ea   Up 19 hours   centos:8
root@docker-01:~#

```

- **ps** 명령어의 출력 결과가 줄바꿈되어 알아보기 힘들 때는 --

**format** 옵션에 **{{.ID}}**와 **{{.Status}}**와 같은 Go 템플릿을 입력해서 원하는 정보만 출력 가능

## 10. Docker container 이름 바꾸기

### # docker rename 기존이름 바꿀이름

```

root@docker-01:~# docker rename magical_satoshi my_container
root@docker-01:~# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
d382291764e5   ubuntu:20.04  "echo hello world!"     About a minute ago   Exited (0) About a minute ago        my_container
65734c1a35ea   centos:8     "/bin/bash"             20 hours ago   Up 19 hours                mycentos
10029779321b   ubuntu:20.04  "/bin/bash"             21 hours ago   Exited (0) 20 hours ago        beautiful_tharp
root@docker-01:~#

```

- 컨테이너의 이름은 중복될 수는 없지만 **rename** 명령어로 바꿀 수 있다.

## 11. Docker container 삭제

# docker rm 컨테이너이름

```
root@docker-01:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
d382291764e5   ubuntu:20.04   "echo hello world!"     14 minutes ago Exited (0) 14 minutes ago
65734c1a35ea   centos:8       "/bin/bash"            20 hours ago   Up 19 hours
10029779321b   ubuntu:20.04   "/bin/bash"            21 hours ago   Exited (0) 20 hours ago
root@docker-01:~# docker rm my_container
my_container
root@docker-01:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
65734c1a35ea   centos:8       "/bin/bash"            20 hours ago   Up 19 hours
10029779321b   ubuntu:20.04   "/bin/bash"            21 hours ago   Exited (0) 20 hours ago
root@docker-01:~#
```

- 더 이상 사용하지 않는 컨테이너를 삭제할 때는 docker rm 명령어를 사용
- 한 번 삭제한 컨테이너는 복구할 수 없으므로 신중히 삭제

```
root@docker-01:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
65734c1a35ea   centos:8       "/bin/bash"            20 hours ago   Up 19 hours
10029779321b   ubuntu:20.04   "/bin/bash"            21 hours ago   Exited (0) 20 hours ago
root@docker-01:~# docker rm mycentos
Error response from daemon: You cannot remove a running container 65734c1a35eac6aa2cd7a5ea95d2f30752e35db3f23352044709d8aed19424ab. Stop the container before attempting removal or force remove
root@docker-01:~#
root@docker-01:~# docker stop mycentos
mycentos
root@docker-01:~# docker rm mycentos
mycentos
root@docker-01:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
10029779321b   ubuntu:20.04   "/bin/bash"            21 hours ago   Exited (0) 21 hours ago
root@docker-01:~#
```

- 실행중인 컨테이너는 삭제가 불가능
- 컨테이너 stop 후, 삭제

# docker rm -f 컨테이너이름

```
root@docker-01:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
1411f23560e3   ubuntu:20.04   "/bin/bash"            53 seconds ago Exited (0) 2 seconds ago
368fb8bdfcf2   ubuntu:20.04   "/bin/bash"            56 seconds ago Exited (0) 5 seconds ago
745e0671ed9f   ubuntu:20.04   "/bin/bash"            About a minute ago Up 37 seconds
10029779321b   ubuntu:20.04   "/bin/bash"            21 hours ago   Exited (0) 21 hours ago
root@docker-01:~# docker rm -f test1
test1
root@docker-01:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
1411f23560e3   ubuntu:20.04   "/bin/bash"            About a minute ago Exited (0) 12 seconds ago
368fb8bdfcf2   ubuntu:20.04   "/bin/bash"            About a minute ago Exited (0) 14 seconds ago
10029779321b   ubuntu:20.04   "/bin/bash"            21 hours ago   Exited (0) 21 hours ago
root@docker-01:~#
```

- 실행중인 컨테이너를 강제 삭제 하기 위해서, -f 옵션을 부여

# docker container prune

```
root@docker-01:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
1411f23560e3   ubuntu:20.04   "/bin/bash"             2 minutes ago Exited (0) About a minute ago test3
368fb8bdfcf2   ubuntu:20.04   "/bin/bash"             2 minutes ago Exited (0) About a minute ago test2
10029779321b   ubuntu:20.04   "/bin/bash"             21 hours ago  Exited (0) 21 hours ago beautiful_tharp
root@docker-01:~# docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
1411f23560e35731289cc546f2a179ffe8fefc6da3202242d2897f061b8b3688
368fb8bdfcf2f0ee0520bf096d9e40268d49358a781fc56ad5b1720b5c1c95fa
10029779321b98eacd491c595b63cb1f575d32a1cc693c30524b17c2585b7d95
Total reclaimed space: 14B
root@docker-01:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
root@docker-01:~#
```

- 실행중이지 않은 모든 컨테이너를 삭제하려면 **docker container prune** 명령어를 사용

# docker ps -a -q

# docker stop \$(docker ps -a -q)

# docker rm \$(docker ps -a -q)

```
root@docker-01:~# docker ps -a -q
3c20566cf2bc
6c68c0eb75e7
b03c7ac0a4e8
flc53677da73
root@docker-01:~# docker stop $(docker ps -a -q)
3c20566cf2bc
6c68c0eb75e7
b03c7ac0a4e8
flc53677da73
root@docker-01:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
3c20566cf2bc   centos:8       "/bin/bash"             About a minute ago Exited (0) 2 seconds ago test4
6c68c0eb75e7   centos:8       "/bin/bash"             About a minute ago Exited (0) 2 seconds ago test3
b03c7ac0a4e8   centos:8       "/bin/bash"             About a minute ago Created              test2
flc53677da73   centos:8       "/bin/bash"             About a minute ago Created              test1
root@docker-01:~# docker rm $(docker ps -a -q)
3c20566cf2bc
6c68c0eb75e7
b03c7ac0a4e8
flc53677da73
root@docker-01:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES
root@docker-01:~#
```

- docker ps

> -a 옵션은 컨테이너 상태와 관계 없이 모든 컨테이너를 출력

> -q 옵션은 컨테이너 ID 만 출력

- 이 두가지 옵션들을 사용해 출력된 컨테이너 리스트를 변수로 컨테이너를 삭제하면 모든 컨테이너를 간단히 삭제

### \*\*\*요약 정리\*\*\*

1. **docker run** 명령어는 **pull, create, start** 명령어를 일괄적으로 실행한 후 **attach** 가 가능한 컨테이너라면 컨테이너 내부까지 들어감.
2. **docker create** 명령어는 도커 이미지를 **pull** 한 뒤에 컨테이너를 생성만 할 뿐 **start, attach** 를 실행하지 않음.
3. 보통은 컨테이너를 생성함과 동시에 시작하기 때문에 **run** 명령어를 더 많이 사용
4. 실행중인 컨테이너는 별도의 옵션없이 삭제 불가능 ( **f 옵션**으로 가능 )