

## 4. Docker 기본 명령어

### 4-1. 도커 명령어 구조

- 도커의 명령어는 크게 이미지 관련, 컨테이너 관련, 축약 명령어, 운용 관련 명령어 등으로 나누어진다. 도커의 기본 명령어 구조는 아래와 같다.

```
$ docker [명령어] [옵션]
```

### 4-2. 도커 도움말

- 도커 도움말은 명령어 창에서 help로 확인할 수 있다.

```
$ docker help
```

- 도커 명령어 별 상세 사용 방법은 아래와 같이 확인할 수 있다.

```
$ docker [명령어] --help
```

```
$docker search --help
```



```
C:\WINDOWS\system32>docker search --help
Usage:  docker search [OPTIONS] TERM
Search the Docker Hub for images

Options:
  -f, --filter filter      Filter output based on conditions provided
  --format string          Pretty-print search using a Go template
  --limit int              Max number of search results (default 25)
  --no-trunc               Don't truncate output

C:\WINDOWS\system32>
```

### 4-3. 이미지 관련 명령어

#### 4-3-1. 도커 이미지 검색하기

- DockerHub 레지스트리에서 도커 이미지를 검색한다.

```
$ docker search [이미지이름]
```

- 우분투 이미지를 검색해보면 우분투 이미지 관련 이미지가 다수 검색된다. 검색 결과는 stars 순이다. 대부분 목록을 보면 xxxxx/ubuntu 같은 네임스페이스가 붙지만 첫 번째 아이템은 네임스페이스가 없이 ubuntu로만 되어 있다. OFFICIAL에 OK가 체크된 이미지는 공식 레포지토리인 경우인데, 이 경우는 네임스페이스가 생략된다.

```
$ docker search ubuntu
```

```
선택 관리자: 명령 프롬프트
C:\WINDOWS\system32>docker search ubuntu
NAME                                DESCRIPTION                                STARS    OFFICIAL    AUTOMATED
ubuntu                              Ubuntu is a Debian-based Linux operating sys... 12894    [OK]
dorowu/ubuntu-desktop-lxde-vnc     Docker image to provide HTML5 VNC interface ... 574      [OK]
websphere-liberty                  WebSphere Liberty multi-architecture images ... 280      [OK]
rastasherp/ubuntu-sshd             Dockerized SSH service, built on top of offi... 255      [OK]
consol/ubuntu-xfce-vnc             Ubuntu container with "headless" VNC session... 242      [OK]
ubuntu-upstart                     DEPRECATED: as is Upstart (find other proces... 113
ansible/ubuntu14.04-ansible        Ubuntu 14.04 LTS with ansible                98       [OK]
neurodebian                        NeuroDebian provides neuroscience research s... 85       [OK]
landinternet/ubuntu-16-nginx-php-phmyadmin-mysql-5
open-liberty                       Open Liberty multi-architecture images based... 48       [OK]
ubuntu-debootstrap                 DEPRECATED: use "ubuntu" instead              44       [OK]
i386/ubuntu                        Ubuntu is a Debian-based Linux operating sys... 25
solita/ubuntu-sysdend              Ubuntu + sysdend                             24
fnnos/ubuntu-python3              A slim Ubuntu-based Python3 image             24
nuagebec/ubuntu                   Simple always updated Ubuntu docker images w... 24
landinternet/ubuntu-16-apache-php-5.6
landinternet/ubuntu-16-apache-php-7.0
landinternet/ubuntu-16-nginx-php-phmyadmin-mariadb-10
landinternet/ubuntu-16-nginx-php-5.6-wordpress-4
landinternet/ubuntu-16-nginx-php-5.6
landinternet/ubuntu-16-apache-php-7.1
landinternet/ubuntu-16-nginx-php-7.0
smarterentry/ubuntu               ubuntu with smarterentry                       1
landinternet/ubuntu-16-php-7.1     ubuntu-16-php-7.1                             1
landinternet/ubuntu-16-rspec       ubuntu-16-rspec                               0
```

### 4-3-2. 이미지 다운로드 하기

- 도커 이미지를 도커 레지스트리에서 다운받으려면 `docker image pull` 명령어를 이용한다.

\$ `docker image pull [옵션] 리포지토리명[:태그명]`

- 우분투 이미지를 다운로드 받아 볼 것이다. 태그명을 생략하면 기본적으로 latest가 적용된다.

```
$ docker image pull ubuntu
```

```
관리자: 명령 프롬프트
C:\WINDOWS\system32>docker image pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
f3ef4ff62e0d: Pull complete
Digest: sha256:44ab2c3b2b3b3b23dcb965498ab06abf74a1e6af20a732902250743df0d4172d
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
C:\WINDOWS\system32>
```

### 4-3-3. 이미지 목록 보기

- 현재 보유하고 있는 이미지 목록을 보기 위해서는 `docker image ls` 명령어를 이용한다.

\$ `docker image ls`

```
$ docker image ls
```

```
선택 관리자: 명령 프롬프트
C:\WINDOWS\system32>docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    597ce1600cf4   3 days ago    72.8MB
C:\WINDOWS\system32>
```

## 4-4. 컨테이너 관련 명령어

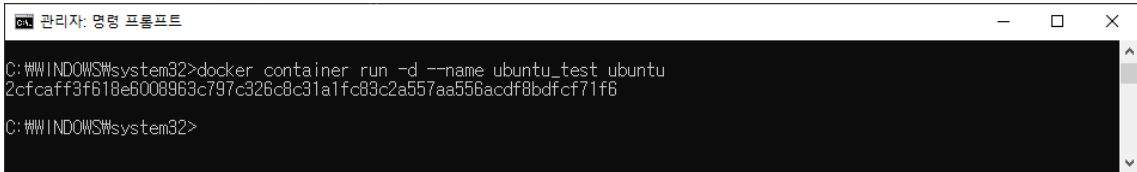
### 4-4-1. 컨테이너 실행하기

- 도커 컨테이너를 실행하기 위해서는 docker container run 명령어를 이용한다.

```
$ docker container run [옵션] 이미지명[:태그명]
```

- ubuntu 이미지를 백그라운드에서 실행하려면 -d 옵션을 주고 실행한다.

```
$ docker container run -d --name ubuntu_test ubuntu
```



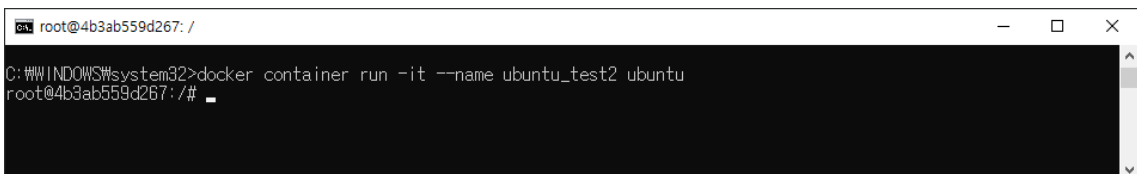
```
관리자: 명령 프롬프트
C:\WINDOWS\system32>docker container run -d --name ubuntu_test ubuntu
2cfcaff3f618e6008963c797c326c8c31a1fc83c2a557aa556acdf8bdfcf71f6
C:\WINDOWS\system32>
```

- 사용 가능한 컨테이너 옵션은 아래와 같다.

옵션	설명
-d	백그라운드로 실행한다.
-p	외부포트:컨테이너포트 포트를 지정하지 않는 경우 임의의 포트가 자동으로 할당된다.
-t	유닉스 터미널 연결 활성화를 시킨다. -i 옵션과 같이 많이 사용되며 -it 옵션으로 합쳐서 실행한다.
-i	컨테이너 쪽 표준 입력(stdout)과 연결을 유지한다. 컨테이너 쪽 셸에 들어가려면 이 옵션을 추가해야 한다.
-rm	컨테이너가 종료 시 컨테이너를 파기한다.
--name	컨테이너에 원하는 이름을 붙일 수 있다. 생성된 컨테이너는 이름으로 조회를 하거나 삭제할 수 있다. name을 부여하지 않고 컨테이너를 생성하면 랜덤한 이름이 생성된다.

- it 옵션을 주어 실행중인 컨테이너의 터미널을 이용할 수 있다.

```
$ docker container run -it --name ubuntu_test2 ubuntu
```



```
root@4b3ab559d267: /
C:\WINDOWS\system32>docker container run -it --name ubuntu_test2 ubuntu
root@4b3ab559d267: /#
```

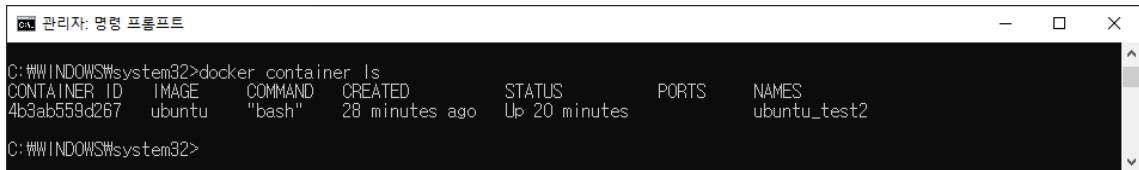
도커 컨테이너 내부 셸에서 나가기 위해서는 exit 명령어를 이용할 수 있다. (단축키 : Ctrl + D) 이 때 컨테이너를 종료시키지 않고 나오기 위해서는 Ctrl + P, Q를 누르면 컨테이너를 종료시키지 않게 된다.

다시 컨테이너로 들어가기 위해서는 docker attach 명령어를 이용할 수 있다.

## 4-4-2. 실행중인 컨테이너 목록 조회하기

- 실행중인 컨테이너를 조회하기 위해서는 docker container ls 명령어를 이용할 수 있다.

```
$ docker container ls
```



```
C:\WINDOWS\system32>docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
4b3ab559d267   ubuntu   "bash"    28 minutes ago    Up 20 minutes           ubuntu_test2
C:\WINDOWS\system32>
```

- 종료된 컨테이너의 목록을 조회하거나 컨테이너의 특정 조건을 필터링 해서 볼 수 있는 다양한 옵션을 제공하고 있다.

옵 션	설명
-a	모든 컨테이너 목록 조회 (종료된 컨테이너도 조회됨)
-f	컨테이너 목록 필터링 해서 보기 (예 : docker container ls -f "name=ubuntu")
-n	모든 컨테이너 목록들 중 n번째 까지의 최근 목록을 보여준다. (예: docker container ls -n 3)
-l	모든 컨테이너 목록 중 가장 최근의 컨테이너 목록을 보여준다.
-q	실행중인 컨테이너의 아이디를 보여준다.
-s	실행중인 컨테이너의 파일 사이즈를 보여준다.

## 4-4-3. 실행중인 컨테이너로 들어가기

- 실행중인 컨테이너 내부로 들어가기 위해 docker container attach 명령어를 이용할 수 있다.

```
$ docker container attach [컨테이너명]
```

```
$ docker attach ubuntu_test2
```

## 4-4-4. 실행중인 컨테이너 정지하기

- 실행중인 컨테이너를 종료하기 위해 docker container stop 명령어를 이용할 수 있다.

```
$ docker container stop [컨테이너아이디 | 컨테이너명]
```

```
$ docker container stop ubuntu_test2
```

```
관리자: 명령 프롬프트
C:\WINDOWS\system32>docker container stop ubuntu_test2
ubuntu_test2
C:\WINDOWS\system32>docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
4b3ab559d267   ubuntu   "bash"    43 minutes ago   Exited (0) 14 seconds ago           ubuntu_test2
2cfcaff3f618   ubuntu   "bash"    49 minutes ago   Exited (0) 49 minutes ago           ubuntu_test
C:\WINDOWS\system32>
```

## 4-4-5. 정지된 컨테이너 시작하기

- 정지된 컨테이너를 시작하기 위해 `docker container start` 명령어를 이용할 수 있다. (아이디를 입력하는 경우에는 컨테이너를 식별 가능한 정도로 일부만 작성해도 된다.)

```
$ docker container start [컨테이너아이디 | 컨테이너명]
```

```
$ docker container start 4b
```

```
관리자: 명령 프롬프트
C:\WINDOWS\system32>docker container start 4b
4b
C:\WINDOWS\system32>docker container ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
4b3ab559d267   ubuntu   "bash"    45 minutes ago   Up 13 seconds           ubuntu_test2
C:\WINDOWS\system32>
```

## 4-5-6. 컨테이너 재시작하기

- 컨테이너를 재시작 하는 경우 `docker container restart` 명령어를 이용할 수 있다.

```
$ docker container restart [컨테이너아이디 | 컨테이너명]
```

```
$ docker container restart ubuntu_test2
```

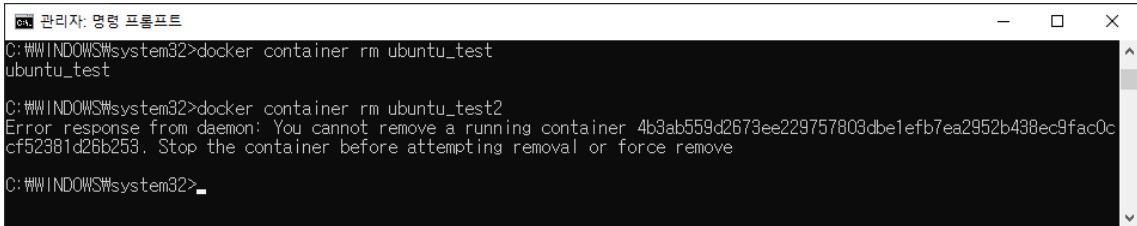
```
관리자: 명령 프롬프트
C:\WINDOWS\system32>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
4b3ab559d267   ubuntu   "bash"    48 minutes ago   Up 35 seconds           ubuntu_test2
C:\WINDOWS\system32>docker container restart ubuntu_test2
ubuntu_test2
C:\WINDOWS\system32>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
4b3ab559d267   ubuntu   "bash"    49 minutes ago   Up 2 seconds           ubuntu_test2
C:\WINDOWS\system32>
```

## 4-5-7. 컨테이너 삭제하기

- 컨테이너를 삭제할 시 `docker container rm` 명령어를 이용할 수 있다. (기본 옵션으로는 실행중인 컨테이너를 삭제할 수 없기 때문에 종료하고 삭제해야 함.)

```
$ docker container rm [컨테이너아이디 | 컨테이너명]
```

```
$ docker container rm ubuntu_test
$ docker container rm ubuntu_test2
```



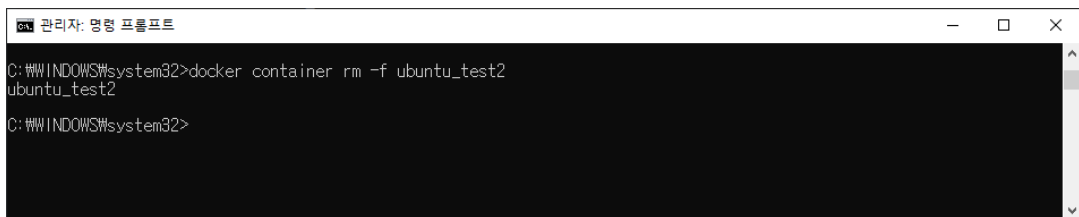
```
관리자: 명령 프롬프트
C:\WINDOWS\system32>docker container rm ubuntu_test
ubuntu_test

C:\WINDOWS\system32>docker container rm ubuntu_test2
Error response from daemon: You cannot remove a running container 4b3ab559d2673ee229757803dbe1efb7ea2952b438ec9fac0c
cf52381d26b253. Stop the container before attempting removal or force remove

C:\WINDOWS\system32>
```

- 실행중인 컨테이너를 삭제하기 위한 방법
  - 컨테이너 생성 시 `--rm` 옵션 부여하면 컨테이너 종료 시 자동 파기됨
  - 실행중인 컨테이너 종료 후 컨테이너 삭제
  - 컨테이너 삭제 시 `-f` 옵션 부여 후 삭제 (실행중인 컨테이너 강제 종료 후 삭제하는 옵션)

```
$ docker container rm -f ubuntu_test2
```



```
관리자: 명령 프롬프트
C:\WINDOWS\system32>docker container rm -f ubuntu_test2
ubuntu_test2

C:\WINDOWS\system32>
```

## 4-5. 축약 명령어

- 도커의 이미지, 컨테이너 관련 명령어는 축약 명령어도 제공하고 있다. 하지만 축약 명령어보다는 full 명령어를 사용하는 것을 권장한다.

도커 full 명령어	축약 명령어
docker image pull	docker pull
docker image ls	docker images
docker container run	docker run
docker container start	docker start
docker container ls	docker ps
docker container stop	docker stop

## 4-6. 운용 관련 명령어

### 4-6-1. 컨테이너 및 이미지 파기하기

- 현재 실행중이지 않은 모든 컨테이너를 삭제할 때 `docker container prune` 명령어를 사용한다.

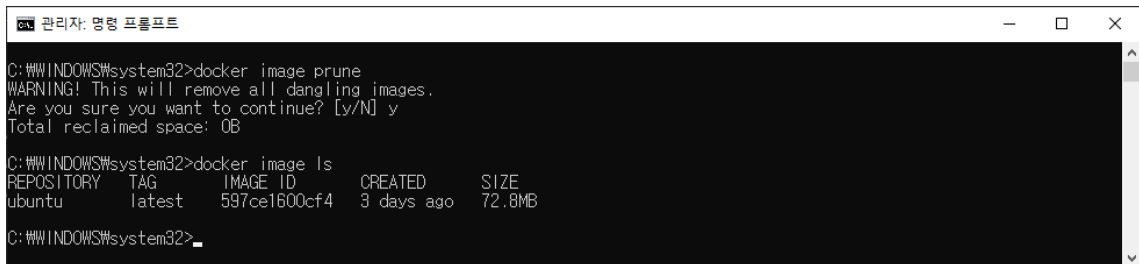
```
$ docker container prune
```



```
C:\WINDOWS\system32>docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
C:\WINDOWS\system32>
```

- 태그가 붙지 않은 모든 이미지를 삭제할 때 `docker image prune` 명령어를 사용한다.

```
$ docker image prune
```



```
C:\WINDOWS\system32>docker image prune
WARNING! This will remove all dangling images.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B

C:\WINDOWS\system32>docker image ls
REPOSITORY    TAG       IMAGE ID   CREATED   SIZE
ubuntu        latest    597ce160cf4 3 days ago 72.8MB
C:\WINDOWS\system32>
```

- 이미지(태그가 없는 이미지), 컨테이너, 볼륨, 네트워크 등 모든 리소스를 일괄 삭제할 때 `docker system prune` 명령어를 사용한다.

```
$ docker run -d --name ubuntu_test3 ubuntu
$ docker ps -a
$ docker system prune
$ docker ps -a
```

```
관리자: 명령 프롬프트
C:\WINDOWS\system32>docker run -d --name ubuntu_test3 ubuntu
5234dc14b77df1e91c9ed94ad72984766710f05ca9cba355204ddea438367e75

C:\WINDOWS\system32>docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
5234dc14b77d   ubuntu   "bash"    4 seconds ago    Exited (0) 5 seconds ago           ubuntu_test3

C:\WINDOWS\system32>docker system prune
WARNING! This will remove:
 - all stopped containers
 - all networks not used by at least one container
 - all dangling images
 - all dangling build cache

Are you sure you want to continue? [y/N] y
Deleted Containers:
5234dc14b77df1e91c9ed94ad72984766710f05ca9cba355204ddea438367e75

Total reclaimed space: 0B

C:\WINDOWS\system32>docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
C:\WINDOWS\system32>
```

## 4-6-2. 컨테이너 시스템 리소스 사용 현황 확인하기

- 현재 실행중인 컨테이너가 있다면 docker container stats명령어를 이용하면 실시간으로 리소스 사용 현황을 보여준다.

```
$ docker container run -it --name ubuntu_test3 ubuntu
ctrl + P,Q

$ docker container ls -a
$ docker container stats ubuntu_test3
ctrl + C
```

```
관리자: 명령 프롬프트
CONTAINER ID   NAME          CPU %     MEM USAGE / LIMIT   MEM %     NET I/O   BLOCK I/O  PIDS
2bbea0f44a93   ubuntu_test3  0.00%    840KiB / 25.01GiB   0.00%    976B / 0B  0B / 0B    1
^C
C:\WINDOWS\system32>^D
```