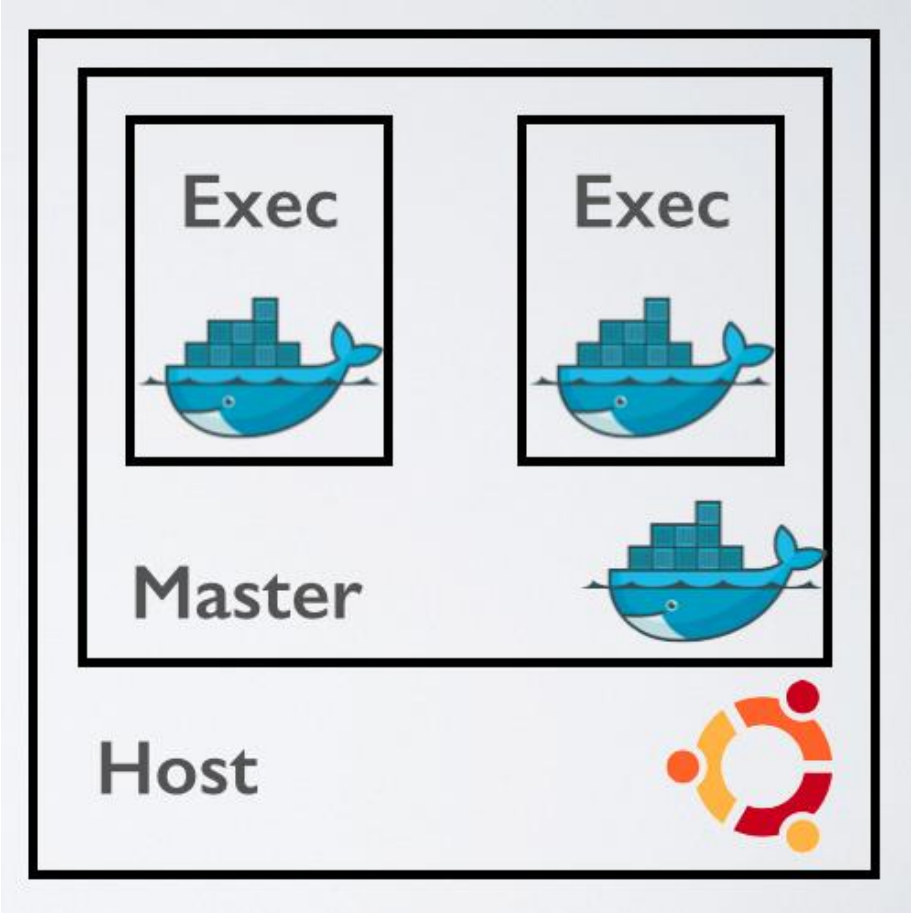
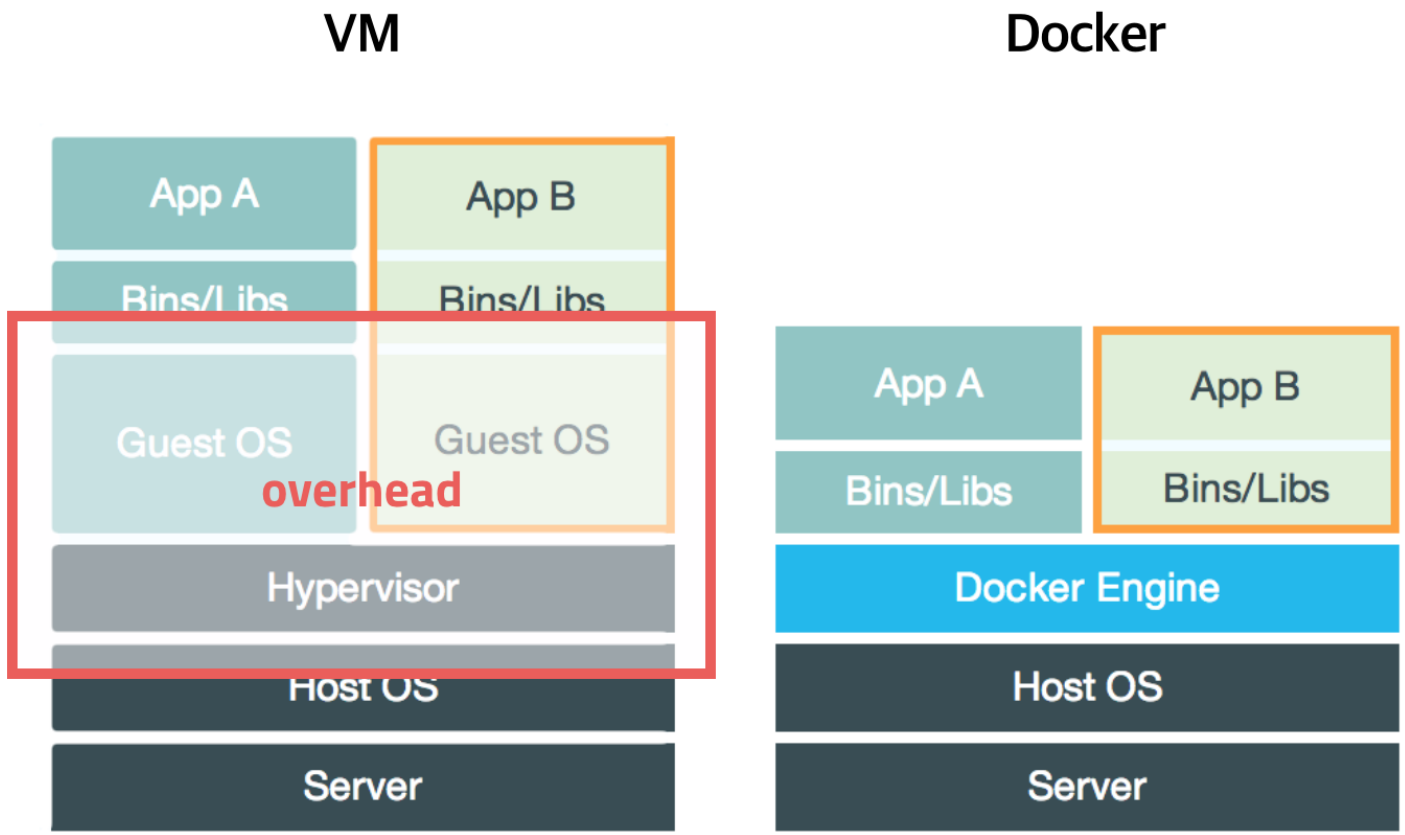
Docker 기초

목차

도커란 무엇인가?



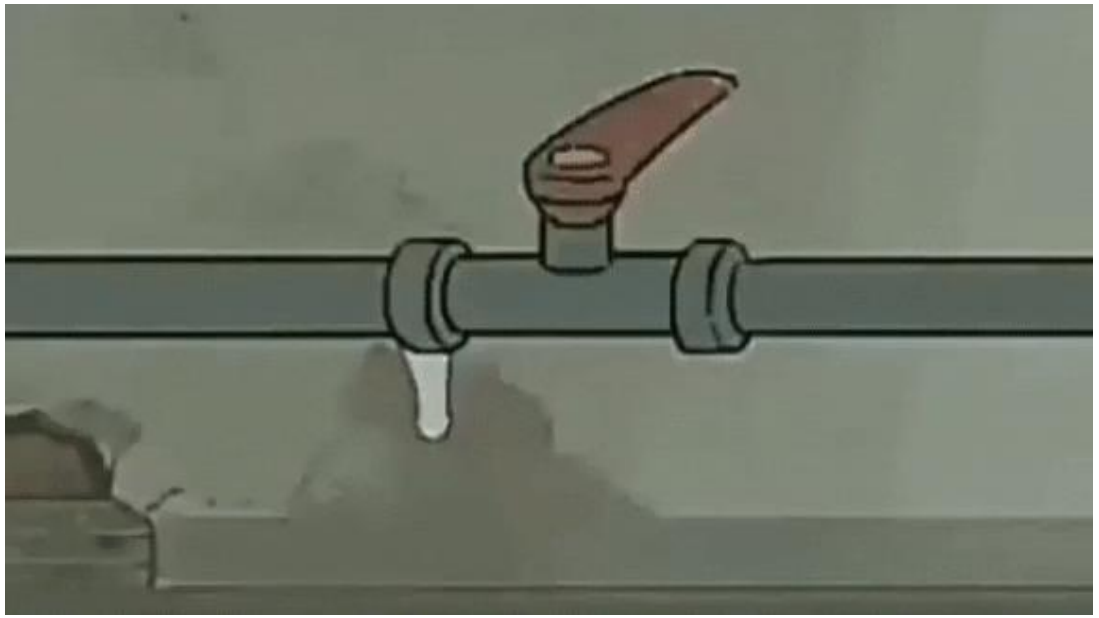
도커란 무엇인가? - VM vs Docker



도커란 무엇인가? - 성능 측정

	Benchmark tool	Host	Docker
CPU	sysbench	1	0.9945
메모리 쓰기	sysbench	1	0.9826
메모리 읽기	sysbench	1	1.0025
디스크 I/O	dd	1	0.9811
네트워크	iperf	1	0.9626

왜 사용해야 되는가?
도커가 없을 때 인프라 담당자가 겪는 일



왜 사용해야 되는가? - 사례1

```
[minds@182.162.19.23 /]$tree -L 1 -d /home
/home
├── bhlim
├── changho
├── dh7
├── hcshin
├── hee
├── isaac
├── juho
├── mcjoe
├── minds
├── ShynarT
└── swpark
```

```
[minds@182.162.19.23 /]$tree -L 1 -d /
/
├── bin -> usr/bin
├── boot
├── data1
├── DATA1
├── dev
├── diarl_data
├── etc
├── home
├── lib -> usr/lib
├── lib64 -> usr/lib64
├── lost+found
├── media
├── mnt
├── opt
├── proc
├── root
├── run
├── sbin -> usr/sbin
├── srv
├── sys
├── tmp
├── usr
└── var
```

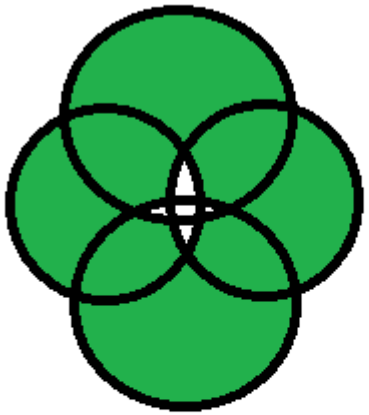
왜 사용해야 되는가? - 사례1

```
[minds@182.162.19.23 local]$ll
total 52
drwxr-xr-x.  2 root root 4096 Apr  7 05:34 bin
lrwxrwxrwx.  1 root root   20 May 21  2019 cuda -> /usr/local/cuda-10.0
drwxr-xr-x. 19 root root 4096 May 21  2019 cuda-10.0
drwxr-xr-x.  4 root root 4096 Oct 21  2019 cuda-9.0
drwxr-xr-x.  4 root root 4096 Oct 21  2019 cuda-9.2
drwxr-xr-x.  2 root root 4096 Apr 11  2018 etc
drwxr-xr-x.  2 root root 4096 Apr 11  2018 games
drwxr-xr-x.  4 root root 4096 Oct 24  2019 include
drwxr-xr-x.  5 root root 4096 Oct 25  2019 lib
drwxr-xr-x.  3 root root 4096 Apr  7 05:34 lib64
drwxr-xr-x.  2 root root 4096 Apr 11  2018 libexec
drwxr-xr-x.  2 root root 4096 Apr 11  2018 sbin
drwxr-xr-x.  5 root root 4096 May 20  2019 share
drwxr-xr-x.  5 root root 4096 Nov 15 05:44 src
[minds@182.162.19.23 local]$echo $PATH
/usr/local/cuda/bin:/usr/local/cuda/bin:/usr/local/cuda/bin:/usr/local/cu
```

왜 사용해야 되는가? - 사례2

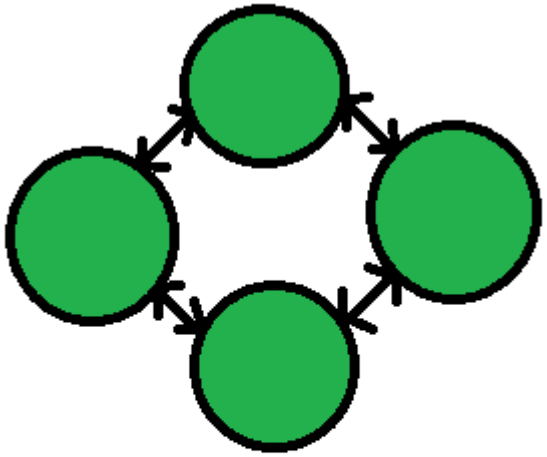
Time	Client	Command
29/04/20 11:10:46	A	python -m multiproc -d 4 -d 5 trainerd.py -c 2020_03_19-132051/checkpoint_13000 --hparams=distributed_run=True
29/04/20 11:20:36	B	python -m multiproc -d 4 -d 5 trainerd.py -c 2019_09_02_fp16_22k_kor_base_2_178000 --warm_start --hparams=distributed_run=True

왜 사용해야 되는가? - Coupling



Tight coupling:

1. More Interdependency
2. More coordination
3. More information flow



Loose coupling:

1. Less Interdependency
2. Less coordination
3. Less information flow

공부할 때 참고할 만한 자료

- https://jungwoon.github.io/docker/2019/01/11/Docker-1/
- http://pyrasis.com/docker.html
- https://subicura.com/2017/01/19/docker-guide-for-beginners-1.html
- https://docs.docker.com/reference/

용어 정의 - Image? Container?

- Image: 컨테이너 실행에 필요한 파일과 설정값등을 포함하고 있는 것으로 상태값을 가지지 않고 변하지 않습니다.

- Container: 이미지를 실행한 상태라고 볼 수 있고 추가되거나 변하는 값은 컨테이너에 저장됩니다.

용어 정의

재료

주입

붕어빵 틀

생성

붕어빵

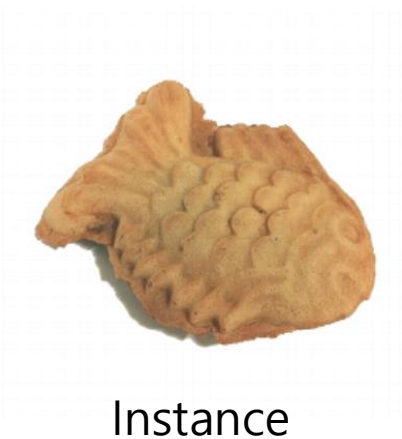
용어 정의

Member Variable

주입

Class

생성



Instance

용어 정의

Request

주입

JSP

생성

Html

용어 정의

Config

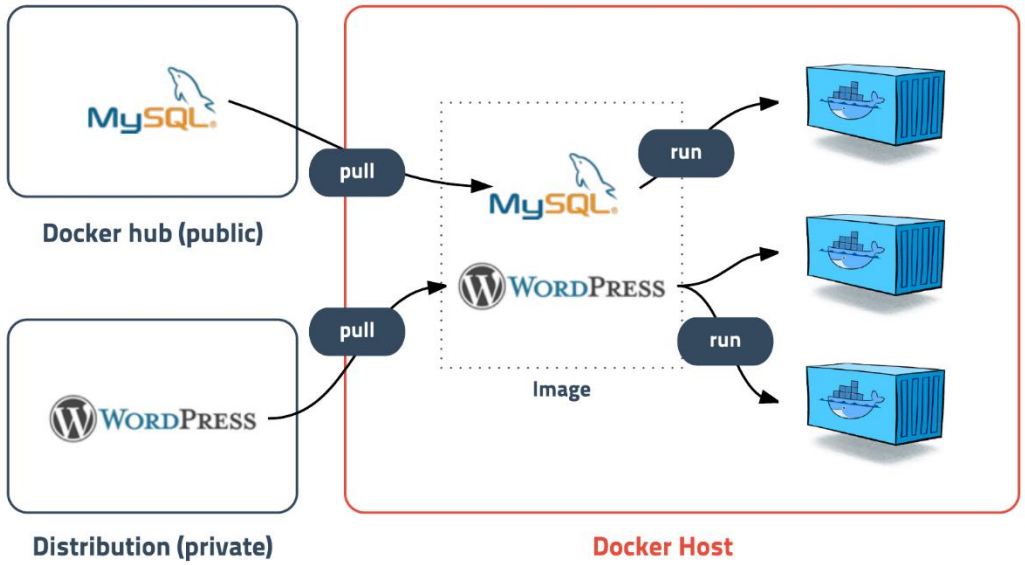
주입

Image

생성

Container

도커를 사용하는 과정



1. Build or pull
2. Run
3. Attach or exec or ssh

명령어

- docker app
- docker attach
- docker build
- docker builder
- docker buildx
- docker checkpoint
- docker cluster
- docker commit
- docker config
- docker container
- docker context
- docker cp
- docker create
- docker diff
- docker events
- docker exec
- docker export
- docker history
- docker image
- docker images
- docker import
- docker info
- docker inspect
- docker kill
- docker load
- docker login
- docker logout
- docker logs
- docker manifest
- docker network
- docker node
- docker pause
- docker plugin
- docker port
- docker ps
- docker pull
- docker push
- docker registry
- docker rename
- docker restart
- docker rm
- docker rmi
- docker run
- docker save
- docker search
- docker secret
- docker service
- docker stack
- docker start
- docker stats
- docker stop
- docker swarm
- docker system
- docker tag
- docker top
- docker trust
- docker unpause
- docker update
- docker version
- docker volume
- docker wait

명령어 - Image

- docker images
- docker rmi
- docker build

- docker search
- docker pull

명령어 - docker images

- 도커 이미지의 목록을 출력하는 명령어

```
[minds@182.162.19.23 ~]$docker images
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
g2p                         test                60cbb98b765e        9 days ago          10.8GB
tts/server/base             v1.2.2              8547a437e8f5        2 weeks ago         1.05GB
tts/base                    v1.2.2              268ba296d5a1        2 weeks ago         1.05GB
g2p/eng/server/base         v1                  0dabb562c54e        2 weeks ago         8.24GB
g2p/eng/base                v1                  a84e2076dd31        2 weeks ago         8.19GB
tts/server/base             v1.2.1              d37ce1a98635        3 weeks ago         1.05GB
tts/base                    v1.2.1              1b39d35275be        3 weeks ago         1.05GB
tts/server/base             v1.2                2b73bf3f3e35        4 weeks ago         1.05GB
tts/base                    v1.2                bb01d0aacd93        4 weeks ago         1.05GB
waveglow/server/base        v1.2                9275dfff50c7        4 weeks ago         9.79GB
waveglow/base               v1.2                df3fca09dcee        4 weeks ago         9.79GB
tacotron2/server/base       v1.2                7903a8f04531        4 weeks ago         9.78GB
tacotron2/base              v1.2                44ba5acb4542        4 weeks ago         9.78GB
openjdk                     8                   0d54b885dc70        4 weeks ago         510MB
waveglow/server/base        v1.1.1              aa4d6d65f395        5 weeks ago         8.24GB
nvcr.io/nvidia/pytorch      20.03-py3           16c4987611fa        5 weeks ago         9.39GB
tacotron2/server/base       v1.1                650243ae9214        2 months ago        8.1GB
wav2letter                  cuda-latest         59a0e919a00b        2 months ago        12.9GB
konglish/server/base        v1                  236a9ea4f194        3 months ago        8.13GB
nvcr.io/nvidia/pytorch      19.06-py3           fad001550f7a        10 months ago       7.71GB
```

명령어 - docker rmi

- 도커 이미지를 삭제하는 명령어
- docker rmi <image id>
- docker rmi <repository>:<tag>
- 해당 이미지로 만든 컨테이너가 존재하거나, 해당 이미지를 base로 삼은 이미지로 만든 컨테이너가 존재하면 삭제가 되지 않음

```
[minds@182.162.19.23 ~]$docker rmi tts/server/base:v1.2
Untagged: tts/server/base:v1.2
Deleted: sha256:2b73bf3f3e35208731900fff35643c84cdf659fbf1f4c44972242816ea813f39
Deleted: sha256:1956b8027721d767304cd88ec3fab56a544493912fde60aec695cd545847c0a5
Deleted: sha256:a7bded5751264455726419e6db9a6e740c164c1f66456ec4dd4e67e3698598fa
Deleted: sha256:45dc8e230d2fb0555052acfb9cb5372a0deac1fe955ffef6fb94914f2f3a2ea3
Deleted: sha256:0bfbdd310f521485dc7fb196b582d0bd4d3df8136db5f1ffec19085e8c4daaa3
Deleted: sha256:0bc561e8d465215c2bf03b4d582b0071e196f58df11cdb971f979675ecfad8dd
Deleted: sha256:3d2b8fb4bffdf193cca14380930024552db6767ab89d1e07e895a17562cdf7bf
```

명령어 - docker build

- 도커 이미지를 생성하는 명령어
- 읽을 Dockerfile와 <repository>:<tag>를 설정하여 사용
- docker build -f <Dockerfile 경로> -t <repository>:<tag> .

```
[minds@182.162.19.23 g2p_kor]$docker build -f ./Dockerfile -t g2p:test2 .
Sending build context to Docker daemon  27.24MB
Step 1/5 : FROM nvcr.io/nvidia/pytorch:20.03-py3
 ---> 16c4987611fa
Step 2/5 : RUN mkdir /root/g2p_kor
 ---> Using cache
 ---> 6c3b76e537af
Step 3/5 : COPY . /root/g2p_kor
 ---> 51edc2d3dba8
Step 4/5 : RUN APT_INSTALL="apt-get install -y --no-install-recommends" &&     cd /tmp && wget https://apt.repos.intel.com/intel-gpg-keys/GPG-PUB-KEY
019.PUB &&     apt-key add GPG-PUB-KEY-INTEL-SW-PRODUCTS-2019.PUB &&     apt-get update &&     DEBIAN_FRONTEND=noninteractive $APT_INSTALL         op
IP_INSTALL="python3 -m pip --no-cache-dir install --upgrade" &&     $PIP_INSTALL         konlpy &&     cd /tmp &&     curl -LO https://bitbucket.org/
nloads/mecab-0.996-ko-0.9.2.tar.gz &&     tar zxfv mecab-0.996-ko-0.9.2.tar.gz &&     cd mecab-0.996-ko-0.9.2 &&     ./configure &&     make &&     m
e install &&     cd /tmp &&     curl -LO https://bitbucket.org/eunjeon/mecab-ko-dic/downloads/mecab-ko-dic-2.1.1-20180720.tar.gz &&     tar -zxvf mec
0720.tar.gz &&     cd mecab-ko-dic-2.1.1-20180720 &&     ./autogen.sh &&     ./configure &&     ldconfig &&     make &&     sh -c 'echo "dicdir=/usr/
mecab-ko-dic" > /usr/local/etc/mecabrc' &&     make install &&     cd /tmp &&     git clone https://bitbucket.org/eunjeon/mecab-python-0.996.git  &&
     /tmp/mecab-python-0.996
 ---> Running in d1389e5a5b12
```

Dockerfile

- Base image에서 추가로 설치해야되는 것들을 작성함
- Example
  - base image가 nvcr.io/nvidia/pytorch:20.03-py3
  - /root/tacotron2 디렉토리 생성
  - Dockerfile과 동일한 위치에 있는 파일을 /root/tacotron2로 복사
  - 추가로 필요한 python package를 설치
  - Proto 파일을 컴파일
  - 임시 파일 정리

```
FROM nvcr.io/nvidia/pytorch:20.03-py3

RUN mkdir /root/tacotron2

COPY . /root/tacotron2

RUN python3 -m pip --no-cache-dir install --upgrade \
        tensorflow==1.9.0 \
        tensorboardX==1.2 \
        grpcio==1.13.0 \
        grpcio-tools==1.13.0 \
        protobuf==3.5.1 \
        xpinyin==0.5.6 && \
    cd /root/tacotron2 &&\
    python -m grpc.tools.protoc \
        --proto_path=tts/proto/src/main/proto/maum/brain/tts \
        --python_out=. \
        --grpc_python_out=. \
        tts/proto/src/main/proto/maum/brain/tts/tts.proto && \
# ================================================================
# config & cleanup
# ----------------------------------------------------------------
    ldconfig && \
    apt-get clean && \
    apt-get autoremove && \
    rm -rf /var/lib/apt/lists/* /tmp/* /workspace/*
```

명령어 - docker search

- 저장소에 있는 도커 이미지를 검색하는 명령어
- docker search <검색어>

```
[minds@182.162.19.23 ~]$docker search pytorch
NAME                                              DESCRIPTION                                     STARS               OFFICIAL            AUTOMATED
pytorch/pytorch                                   PyTorch is a deep learning framework that pu…   194
floydhub/pytorch                                  pytorch                                         43                                      [OK]
anibali/pytorch                                   Docker images for the PyTorch deep learning …   16                                      [OK]
stepankuzmin/pytorch-notebook                     Jupyter Notebook Pytorch Stack                  6                                       [OK]
thechaos16/pytorch_tf                             ubuntu docker with pytorch, tensorflow          4                                       [OK]
bitnami/pytorch                                    Bitnami PyTorch Docker Image                   3
spellrun/pytorch-jupyter                                                                          3
spellrun/pytorch                                                                                  2
chaneyk/pytorch                                   Pytorch releases with GPU support               2
shatu/pytorch                                     Collection of Docker Images for PyTorch         1
mapler/pytorch-cpu                                Dockerfile of PyTorch on CPU                    1                                       [OK]
pytorch/pytorch-binary-docker-image-ubuntu16.04                                                   1
dsksd/pytorch                                     0.4 : python3.6, cuda8, cudnn6, pytorch==0.4    1
jetware/pytorch                                   PyTorch Production                              1
xiaoxiaoxh/pytorch                                pytorch-based repo                              0
clipper/pytorch-container                         Model container for PyTorch models              0
spellrun/pytorch0.2.0                                                                             0
clipper/pytorch36-container                       Python 3.6 PyTorch container                    0
spellrun/pytorch0.4.0                                                                             0
lucidfrontier45/pytorch                           Minimal PyTorch Image                           0                                       [OK]
nakosung/pytorch_dev                              pytorch                                         0                                       [OK]
```

명령어 - docker pull

- 저장소에 있는 도커 이미지를 받아오는 명령어
- docker pull <repository>:<tag>

```
[minds@182.162.19.23 ~]$docker pull nvcr.io/nvidia/pytorch:20.02-py3
20.02-py3: Pulling from nvidia/pytorch
5c939e3a4d10: Pulling fs layer
c63719cdbe7a: Pulling fs layer
19a861ea6baf: Pulling fs layer
651c9d2d6c4f: Waiting
ec1ff02470e4: Waiting
b0cf94422843: Waiting
24634a544ff3: Waiting
08eeefc4f275: Waiting
8bd0daba03d4: Waiting
0de885550a64: Waiting
0c7d3841bbb7: Waiting
13713568c4cc: Waiting
f36b0f5c6c1c: Waiting
08da6a17fb56: Waiting
1ec1ed31e758: Waiting
29169720b35e: Waiting
36a4f3849bb4: Waiting
88dcd4bbe90f: Waiting
1f108f08952d: Waiting
```

명령어 - Container

- docker ps
- docker start
- docker restart
- docker stop
- docker rm

- docker create
- docker run

명령어 - docker ps

- 도커 컨테이너의 목록을 출력하는 명령어
- docker ps
- -a를 붙여서 실행을 해야 모든 컨테이너가 나옴
- -f name=<검색어>와 같은 식으로 name 컬럼에 조건을 걸어서 조회할 수도 있으며, 저 검색어 부분은 정규표현식으로 입력할 수 있음
- --no-trunc를 붙여서 실행하면 생략된 부분이 다 보임

```
[minds@182.162.19.23 g2p_kor]$docker ps -a
CONTAINER ID        IMAGE                          COMMAND                  CREATED             STATUS                      PORTS                    NAMES
0c85d10574f4        waveglow/server/base:v1.2      "/bin/sh -c 'python …"   5 days ago          Exited (137) 4 days ago                              wg_server
4aa060d1abb2        tacotron2/server/base:v1.2     "/bin/sh -c 'python …"   5 days ago          Exited (137) 4 days ago                              taco_server
c1a1c7131ce5        g2p:test                       "/usr/local/bin/nvid…"   9 days ago          Up 9 days                   6006/tcp, 8888/tcp       g2p_test
643491a623b8        wav2letter:cuda-latest         "/bin/bash"              11 days ago         Up 4 days                                            w2l_train
```

명령어 - docker start/restart/stop/rm

- 도커 컨테이너를 시작/재시작/정지/삭제하는 명령어
- docker start/restart/stop/rm <name>

명령어 - docker create/run

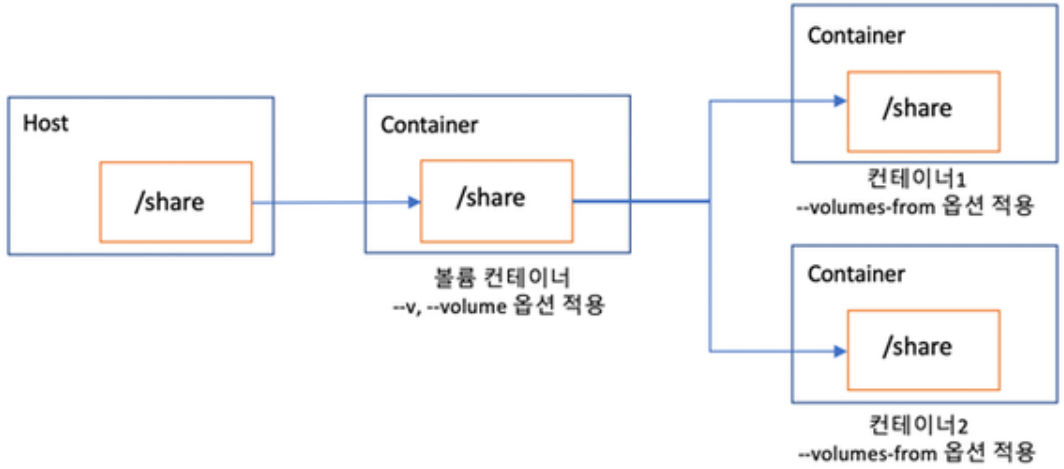
- 도커 컨테이너를 생성하는 명령어
- docker run --name <name> <repository>:<tag>
- 추가로 붙일 수 있는 옵션이 상당히 많음
  - --cpuset-cpus "0-3"
  - --gpus '"device=1"'
  - -v /home/minds/maum/trained/tts/waveglow:/model
  - --net host
  - -p 35001:35001
  - --ipc host
  - -itd
  - -e LC_ALL C.UTF-8

옵션 - 자원 할당

- 메모리
  - --memory <용량. ex) 1g, 1024m>
  - --memory-swap <용량. ex) 1g, 1024m>
- CPU
  - --cpu-shares <숫자. 기본값: 1024>
  - --cpuset-cpus <설정. ex) 2, "0,2", "0-2">
- GPU
  - --gpus <설정. ex) all, 2, '"device=1,2"'>

옵션 - volume

- -v <host>:<container>
  - host의 디렉토리를 container에 mount 시키는 옵션
- --volumes-from <name>
  - 다른 container의 volume을 연결하는 옵션



옵션 - network

- --net <mode>
  - bridge - 기본값. host 안에 격리된 내부망으로 이해하면 됨
  - host - host의 네트워크를 사용
  - none - 네트워크를 사용하지 않음
  - container:<name, id> - 다른 컨테이너의 네트워크를 사용
- -p <host>:<container>
  - host의 port를 container의 port에 연결하는 옵션
  - network mode가 bridge일 때 작동하며, port forwarding으로 이해하면 됨

옵션 - etc

- --ipc host
  - InterProcess Communication. 프로세스 사이에 접근할 필요가 있을 때 설정
- -it
  - Bash를 사용할 때 설정
- -d
  - Detached 모드, 혹은 demon 모드. 백그라운드로 실행
- -e <name> <value>
  - 환경변수를 설정

명령어 - Use

- docker attach
- docker exec
- docker logs

명령어 - docker attach

- 실행 중인 도커 컨테이너에 접속하는 명령어
- 여러 사람이 동시에 접속해도 세션이 하나만 있어서 한 사람이 실행한 명령이 다른 사람 화면에도 보임
- docker attach <name>

명령어 - docker exec

- 외부에서 컨테이너 안의 명령어를 실행하는 명령어
- 아래와 같은 명령어로 한 컨테이너에 여러 사람이 접속할 목적으로도 사용
- docker exec -it <name> /bin/bash

명령어 - docker logs

- 도커 컨테이너의 log를 조회하는 명령어
- docker logs <name>
- -f를 붙이면 tail -f 명령어처럼 실시간으로 log를 볼 수 있음
- 다만, exec로 실행한 건 안 보임

```
[minds@182.162.19.23 g2p_kor]$docker logs -f wg_server
[waveglow|INFO|runner.py:273] 2020-04-29_06:53:44 Asia >>> Save Config File: server.yaml
[waveglow|INFO|server.py:101] 2020-04-29_06:53:51 Asia >>> waveglow starting at 0.0.0.0:35001
Selected optimization level O3:  Pure FP16 training.
Defaults for this optimization level are:
enabled                : True
opt_level              : O3
cast_model_type        : torch.float16
patch_torch_functions  : False
keep_batchnorm_fp32    : False
master_weights         : False
loss_scale             : 1.0
```