DGX-1 DOCKER USER GUIDE 17.08

Josh Park| Senior Solutions Architect

Contents created by Jack Han | Solutions Architect

AGENDA

Introduction to Docker & DGX-1 SW Stack

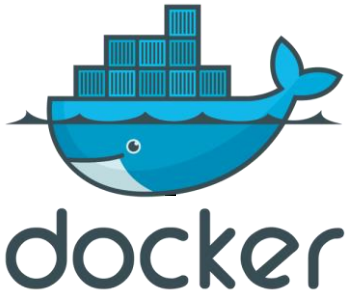
Docker basic & nvidia-docker

Docker image management

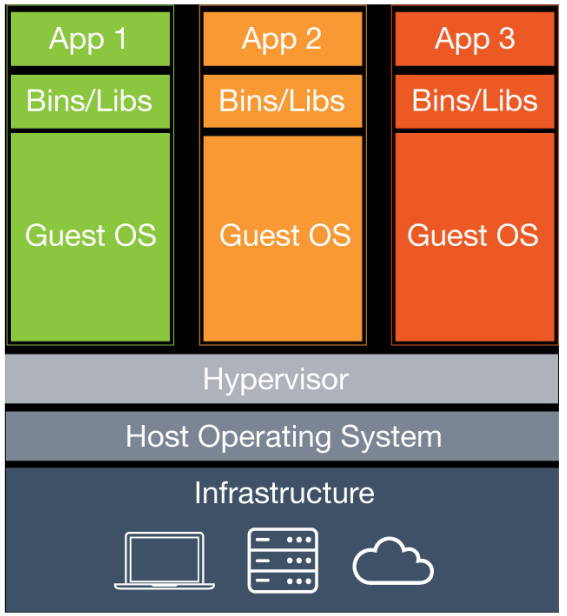
Local registry

DGX-1 & DOCKER

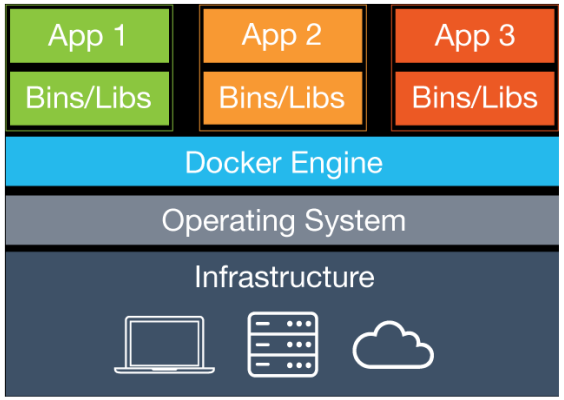
INTRODUCTION TO



- Docker
  - Container level virtualization
  - No machine virtualization resource
  - Shares Host OS's Kernel & Resources
  - Lightweight
  - Free to build & deploy application
  - Enables virtual control



Hypervisor based Virtualization



Container Virtualization

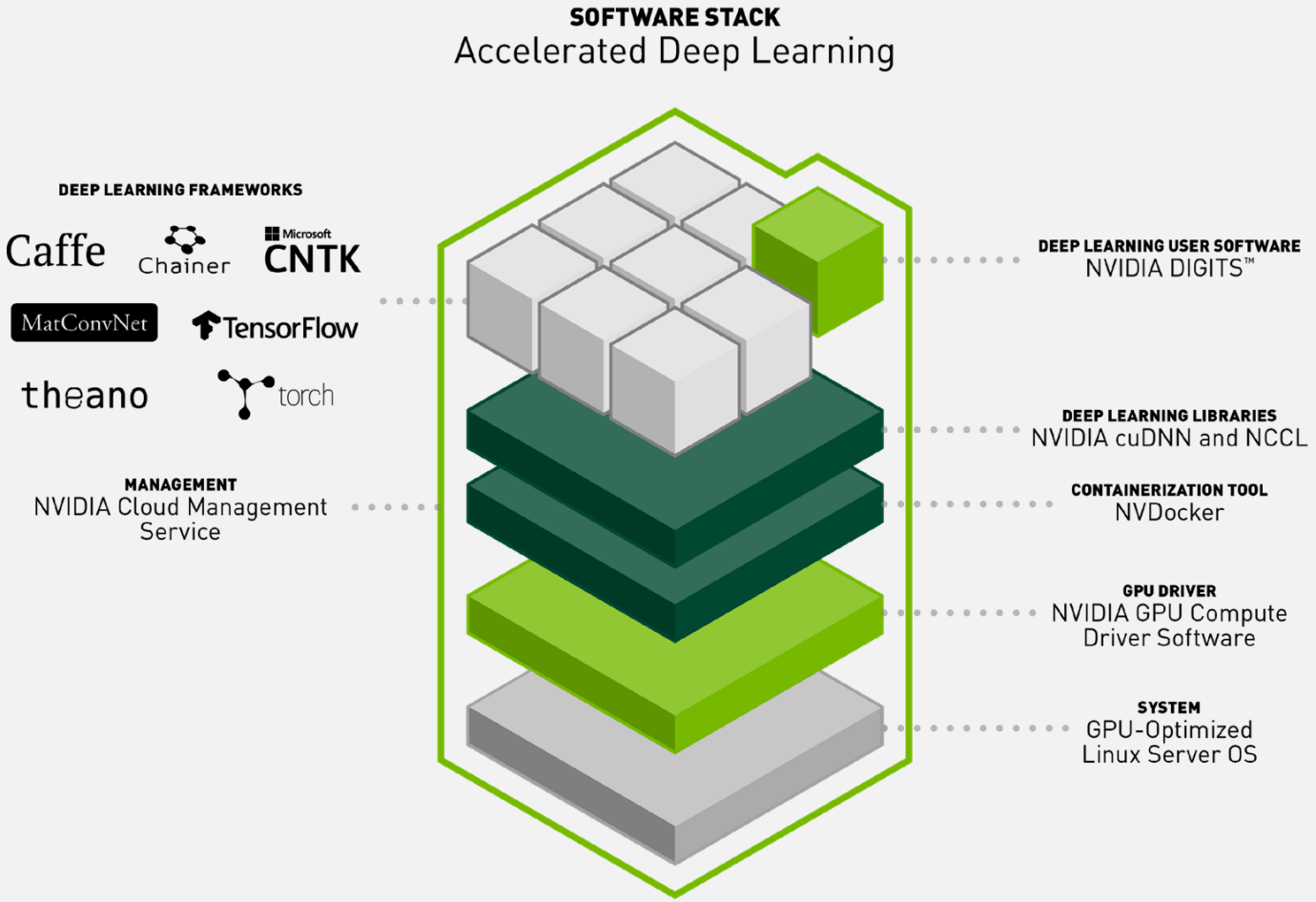
DOCKER'S ECHO SYSTEM

NVIDIA DGX-1

AI supercomputer-in-a-box

170 TFLOPS | 8x Tesla P100 16GB | NVLink Hybrid Cube Mesh

2x Xeon | 8 TB RAID 0 | Quad IB 100Gbps, Dual 10GbE | 3U — 3200W

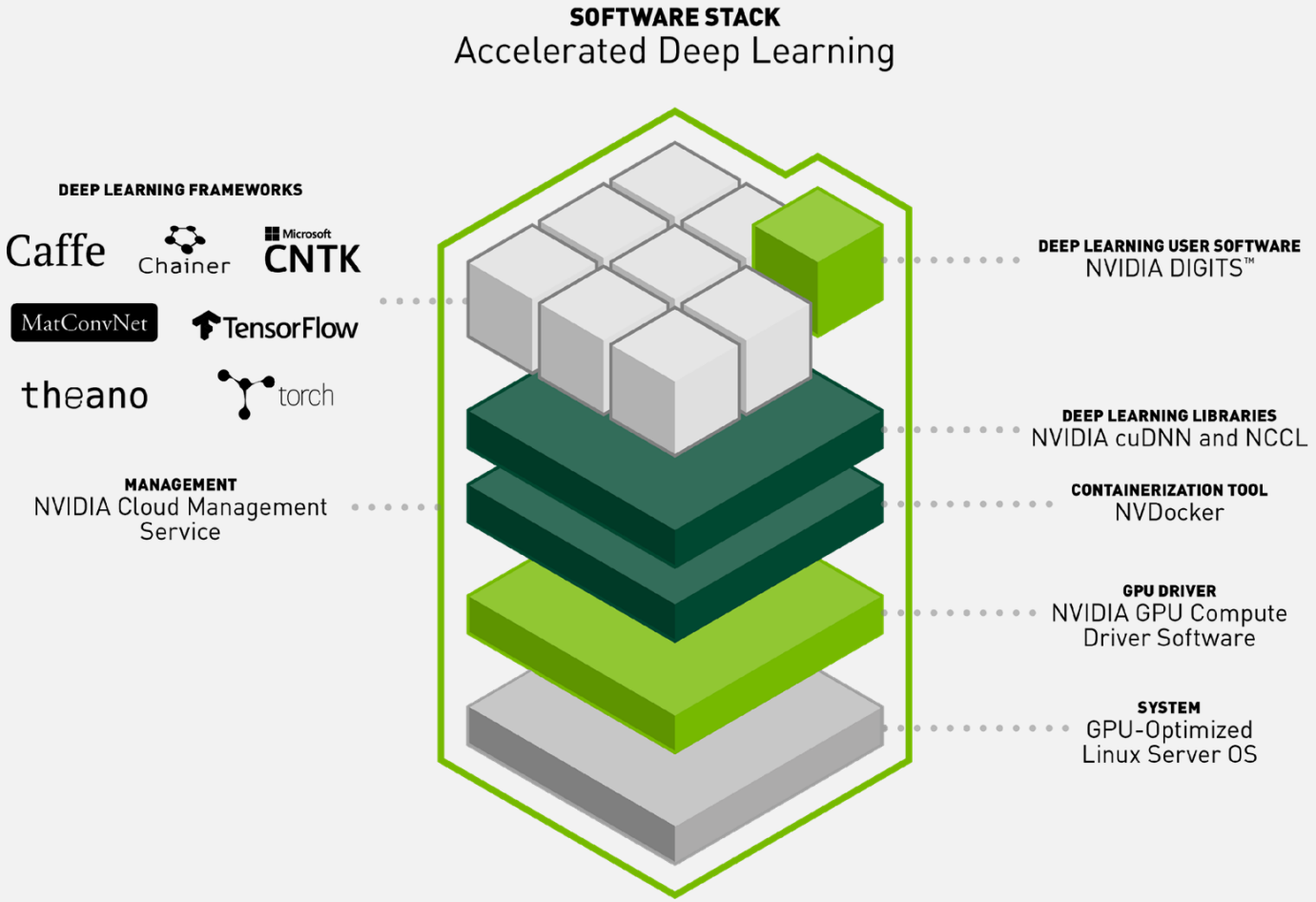


NVIDIA DGX-1V

AI supercomputer-in-a-box

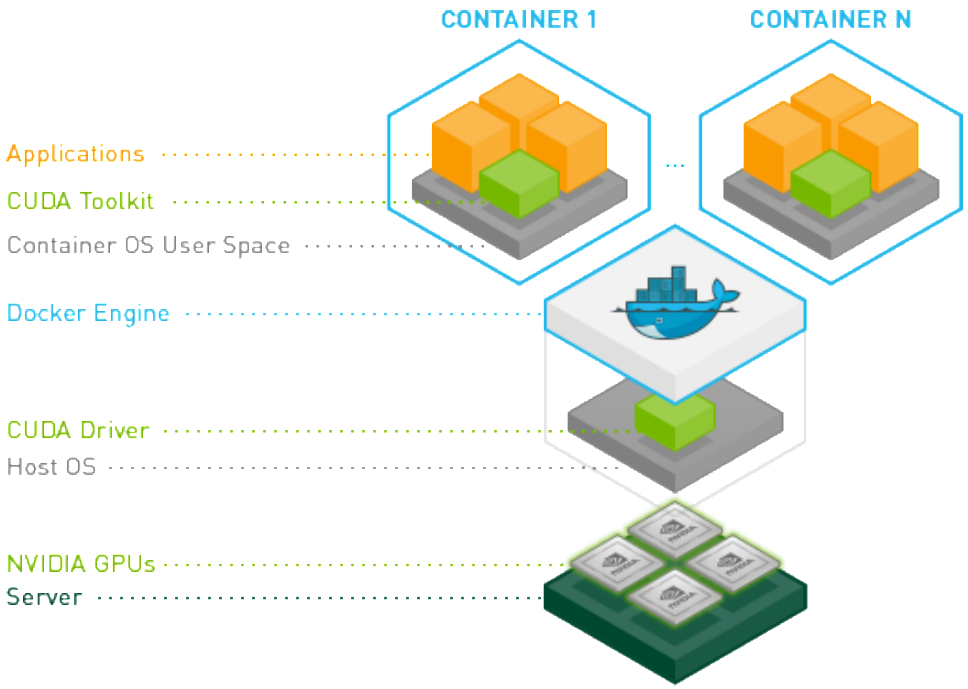
960 TFLOPS | 8x Tesla V100 16GB | NVLink Hybrid Cube Mesh

2x Xeon | 8 TB RAID 0 | Quad IB 100Gbps, Dual 10GbE | 3U — 3200W



NVIDIA DOCKER & DGX-1 SW STACK

Docker Mounting for NVIDIA GPU Hardware




```
[[~]$ docker images
REPOSITORY                    TAG                 IMAGE ID            CREATED             SIZE
atc.kt.com/atc/mxnet          17.06               a5db2b00844d        2 hours ago         2.33GB
atc.kt.com/atc/theano         17.06               93d2a37cb405        2 hours ago         2.92GB
atc.kt.com/atc/cntk           17.06               47a011b53d8a        2 hours ago         5.85GB
atc.kt.com/atc/tensorflow     17.06               03097c6df981        2 hours ago         3GB
atc.kt.com/atc/digits         17.06               d16952eb653a        3 hours ago         4.2GB
atc.kt.com/atc/pytorch        17.06               6fae8d7096cb        3 hours ago         3.83GB
atc.kt.com/atc/torch          17.06               2501196beefe        3 hours ago         2.92GB
atc.kt.com/atc/caffe2         17.06               324a9d7c3c20        3 hours ago         2.59GB
atc.kt.com/atc/caffe          17.06               a7e924a17051        3 hours ago         2.8GB
```

BASIC DOCKER USE

CONTAINER LIFE CYCLE

DOCKER MANAGEMENT RESOURCES

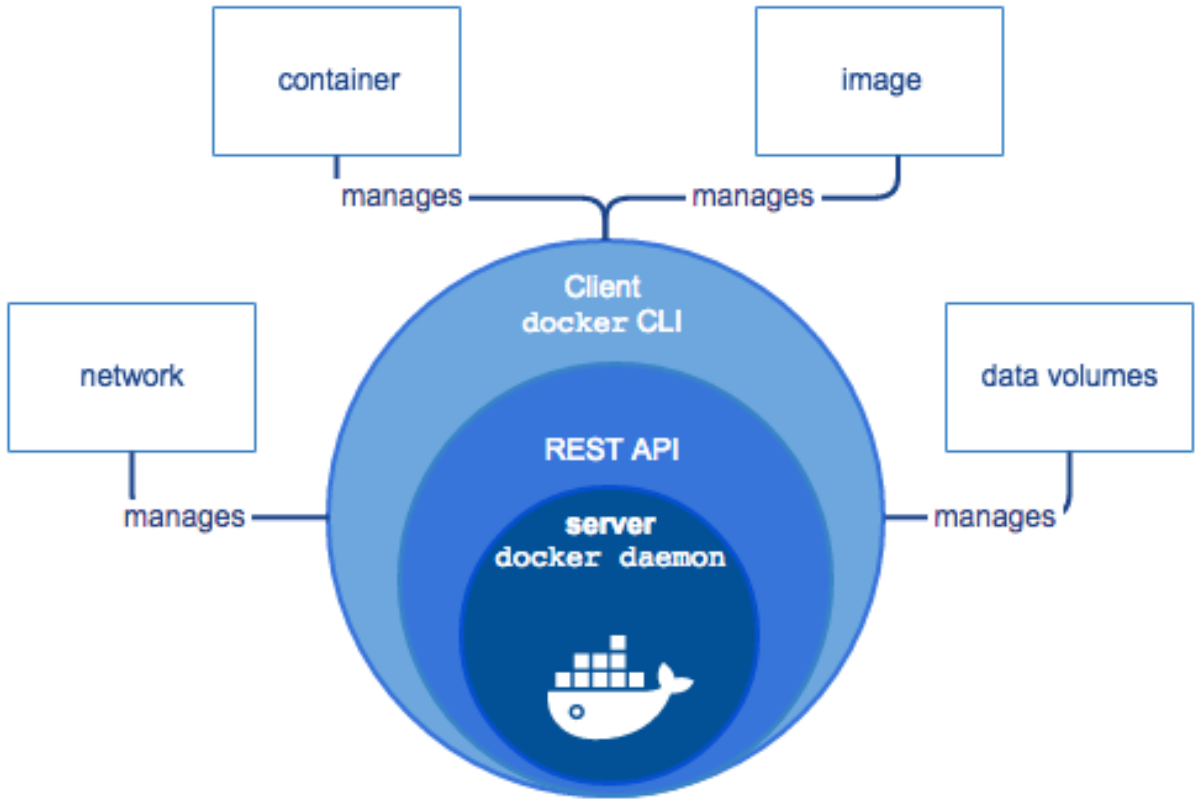


Image source: https://docs.docker.com/engine/docker-overview

DOCKER ARCHITECTURE

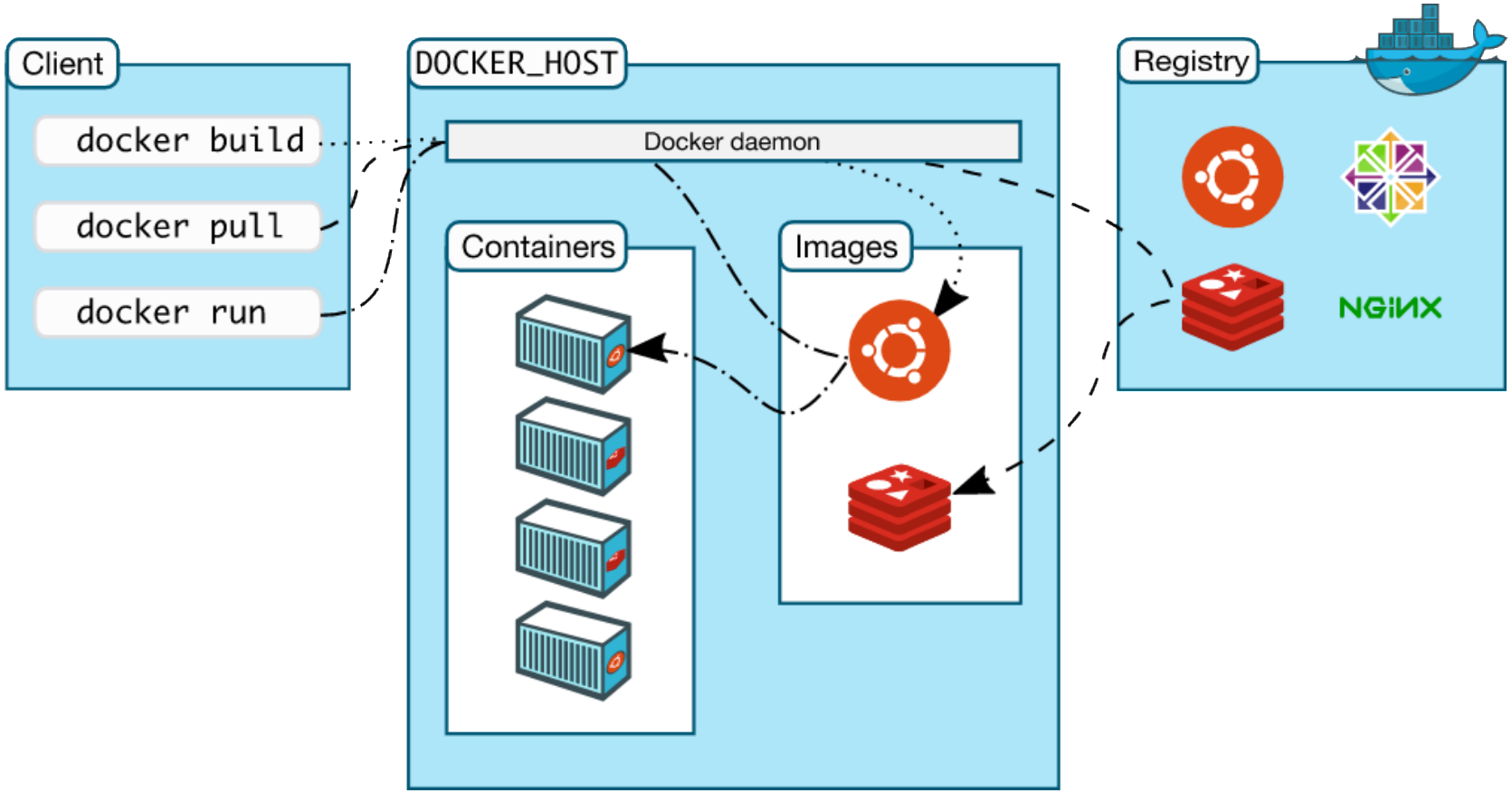
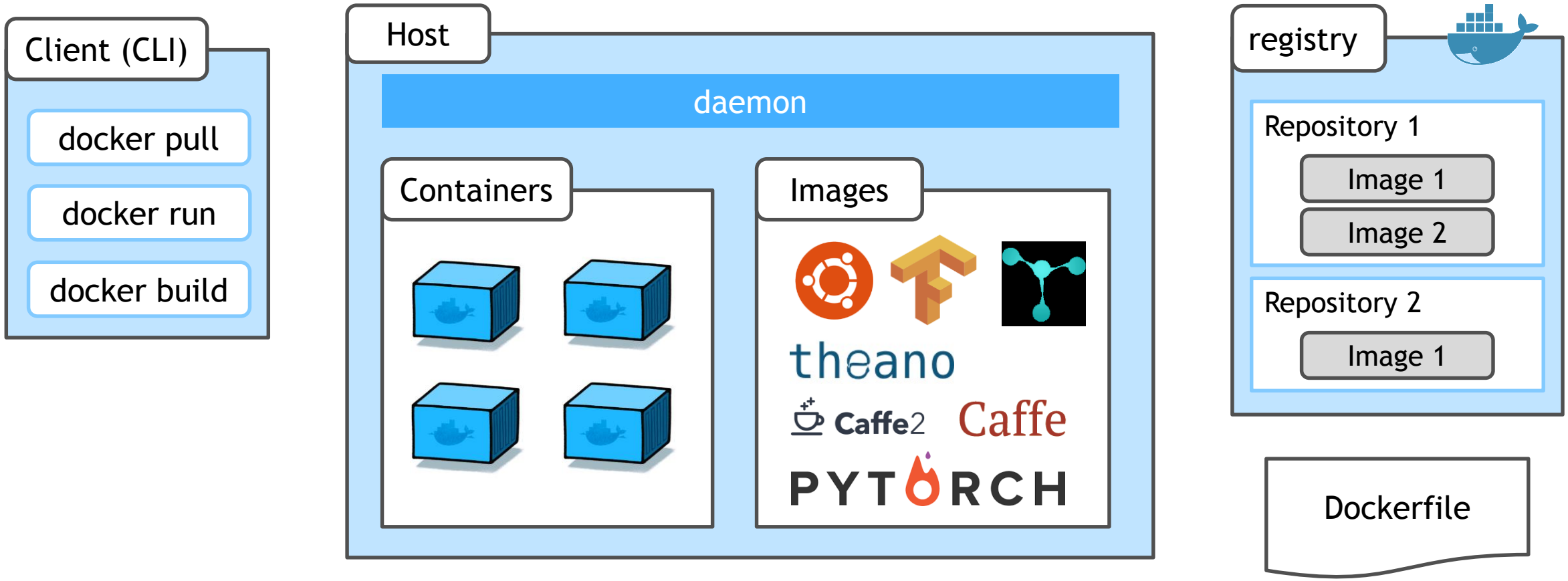


Image source: https://docs.docker.com/engine/docker-overview

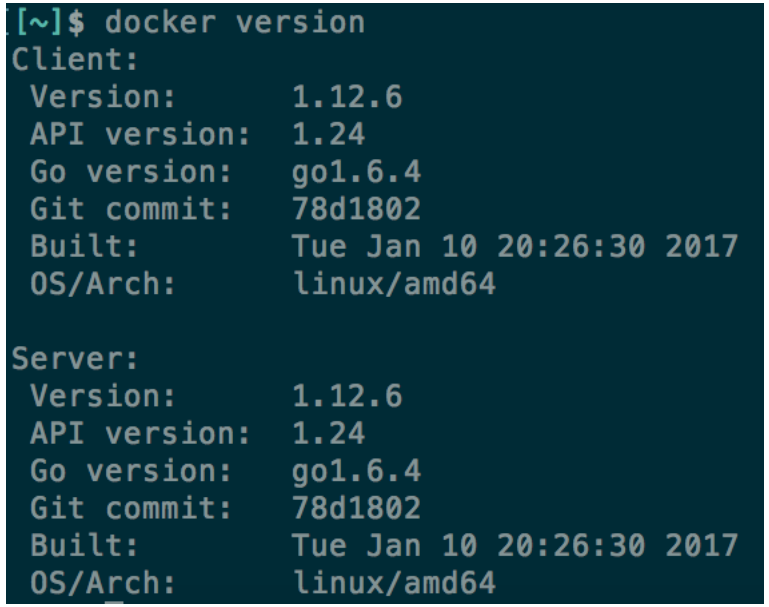
DOCKER ARCHITECTURE



DOCKER VERSION

docker server (dockerd) & docker client (docker) version info

```
$ docker version
```



DOCKER SERVER INFORMATION

```
$ docker info
```

```
[~]$ docker info
Containers: 3
 Running: 3
 Paused: 0
 Stopped: 0
Images: 38
Server Version: 1.12.6
Storage Driver: overlay2
 Backing Filesystem: extfs
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
 Network: bridge host null overlay
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Security Options: apparmor
Kernel Version: 4.8.0-58-generic
Operating System: Ubuntu 16.04.2 LTS
OSType: linux
Architecture: x86_64
CPUs: 8
Total Memory: 15.55 GiB
Name: jahan-ThinkPad
ID: RCNQ:4WD3:2NHV:76XX:PABI:YWQ7:WLXB:5ELC:EUW7:FUAW:II6I:TQM6
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): false
Registry: https://index.docker.io/v1/
WARNING: No swap limit support
Insecure Registries:
 atc.kt.com
 127.0.0.0/8
```

LIST OF INSTALLED DOCKER IMAGES

Name, Tags, ID, Age, Size

```
$ docker images
```

```
[[~]$ docker images
REPOSITORY                    TAG                                 IMAGE ID            CREATED             SIZE
nvcr.io/nvidia/digits         17.07                               dca1f2eca0dc        6 weeks ago         4.58GB
nvcr.io/nvidia/caffe          17.07                               dc9b93b2db88        6 weeks ago         3.27GB
nvcr.io/nvidia/caffe2         17.07                               5cb89724b942        6 weeks ago         3.13GB
nvcr.io/nvidia/theano         17.07                               fd1472e6f64e        6 weeks ago         3.64GB
nvcr.io/nvidia/torch          17.07                               f61062ea13f5        6 weeks ago         3.46GB
nvcr.io/nvidia/pytorch        17.07                               7c0a7658596e        6 weeks ago         4.44GB
nvcr.io/nvidia/mxnet          17.07                               eb33fe6fdc6d        6 weeks ago         2.78GB
nvcr.io/nvidia/cntk           17.07                               817221ed5240        7 weeks ago         6.39GB
nvcr.io/nvidia/tensorflow     17.07                               94b1afe1821c        7 weeks ago         4.4GB
nvcr.io/nvidia/cuda           8.0-cudnn6-devel-ubuntu16.04        14b54c5f2832        7 weeks ago         2.16GB
nvcr.io/nvidia/cuda           8.0-cudnn6.0-devel-ubuntu14.04      d8236a439e37        6 months ago        1.51GB
```

LIST OF CONTAINERS

List of running process image, age, status, name, and port information

```
$ docker ps
```

Options

```
-a: all (including Exited)
-f name=[]: name filter
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
bedb3a009f57	ubuntu	"/bin/bash"	6 minutes ago	Up 6 minutes		cpus
018d6795fd20	ubuntu	"/bin/bash"	26 minutes ago	Up 26 minutes		client
d26a9d62fd08	ubuntu	"/bin/bash"	27 minutes ago	Up 27 minutes		base
0bc96ce917c4	nvcr.io/nvidia/tensorflow:17.05	"/usr/local/bin/nvidi"	46 minutes ago	Up 46 minutes	6006/tcp	tf_job2
c6d0b3d0188e	nvcr.io/nvidia/tensorflow:17.05	"/usr/local/bin/nvidi"	46 minutes ago	Up 46 minutes	6006/tcp	tf_job1
37d308fdeb75	nvcr.io/nvidia/tensorflow:17.05	"/usr/local/bin/nvidi"	49 minutes ago	Up 49 minutes	6006/tcp	modest_liskov
8cf7b118b58b	nvcr.io/nvidia/caffe2:17.06	"/usr/local/bin/nvidi"	52 minutes ago	Up 52 minutes		nauseous_banach
e059c731ef8e	nvcr.io/nvidia/tensorflow:17.06	"/usr/local/bin/nvidi"	52 minutes ago	Up 52 minutes	6006/tcp	distracted_bardeen
714327f02687	nvcr.io/nvidia/pytorch:17.06	"/usr/local/bin/nvidi"	52 minutes ago	Up 52 minutes		condescending_snyder

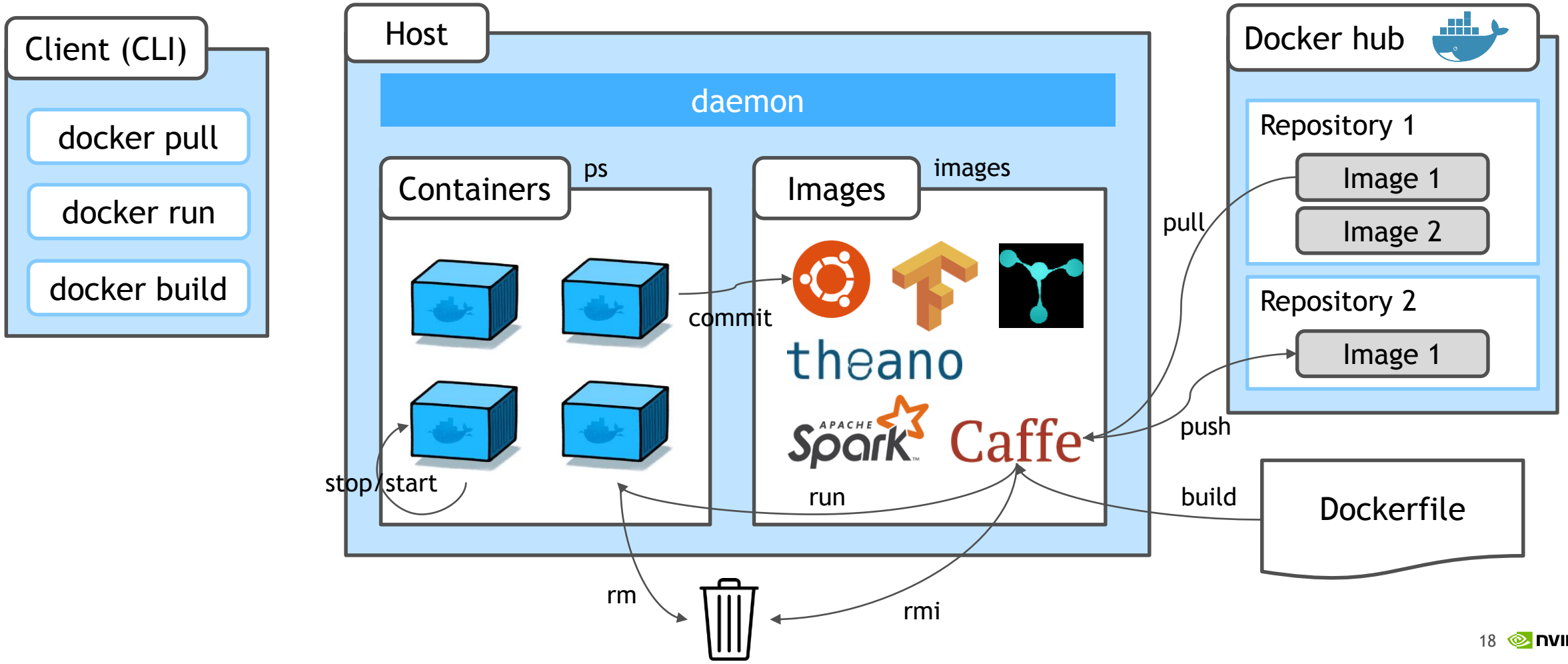
RESOURCE UTILIZATION MONITORING

```
$ docker stats
```

```
CONTAINER           CPU %               MEM USAGE / LIMIT       MEM %               NET I/O             BLOCK I/O           PIDS
ae03b60a6af2        0.00%               1.562 MiB / 15.55 GiB   0.01%               23.16 kB / 648 B    0 B / 0 B           1
6e8bd7cc79df        0.00%               1.363 MiB / 15.55 GiB   0.01%               27.13 kB / 648 B    0 B / 0 B           1
```

DOCKER LIFE CYCLE

Overview



CONTAINER CREATION

from docker images

```
$ docker run [OPTIONS] IMAGE[:TAG] [COMMAND] [ARG...]
```

Options

```
--name: container name
-d: run container in background mode
--rm: remove when it exits
-v: volume mount
-p: port forwarding (-P: open all)
-e: set environment in container
-t: terminal
-i: STDIN open (-ti: -t -i)
-u: set user UID
-w: working directory in container
-m: memory limit
--cpuset-cpus: limit cpus to run
--add-host: custom host:ip setting
--privileged: open kernel functions
```

- Docker running defines
  - detached or foreground running
  - container identification
  - network settings
  - runtime constraints on CPU and memory

NVIDIA-DOCKER

To use GPU, use once when launch container

Enables NVIDIA GPU use from containers

```
nvidia-docker run --rm nvidia/caffe nvidia-smi
```

use once when container create initially

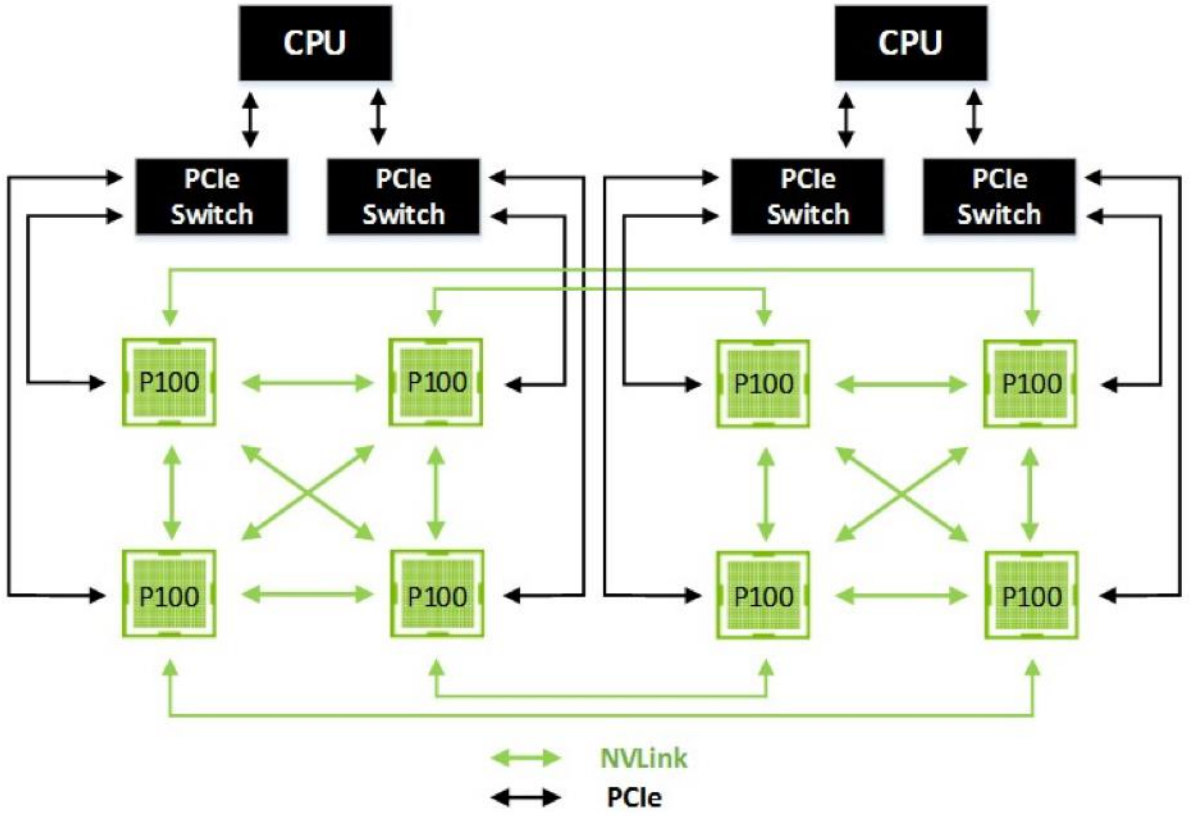
Enables GPU selection (with NV_GPU option)

```
NV_GPU=1,3 nvidia-docker run --rm nvidia/caffe nvidia-smi
```

```
[[~]$ NV_GPU='1,3' nvidia-docker run --rm compute.nvidia.com/nvidia/cuda nvidia-smi
Tue Nov 15 14:06:28 2016
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 361.77                 Driver Version: 361.77                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla P100-SXM2...  Off  | 0000:07:00.0     Off |                    0 |
| N/A   35C    P0    31W / 300W |      0MiB / 16280MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla P100-SXM2...  Off  | 0000:0B:00.0     Off |                    0 |
| N/A   34C    P0    32W / 300W |      0MiB / 16280MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
```

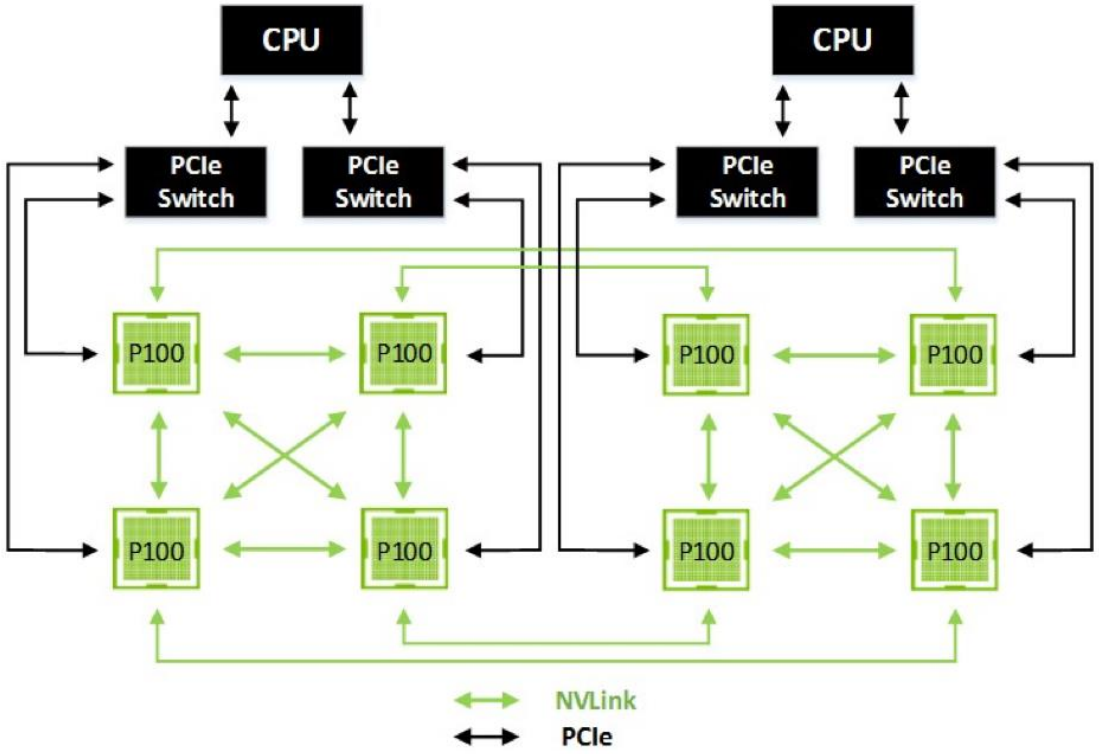
MULTI GPU SELECTION

Select with GPU Topology with NVLINK




```
nvidia-docker run --rm nvidia/cuda:8.0-cudnn6-devel-ubuntu16.04 nvidia-smi topo -m
```

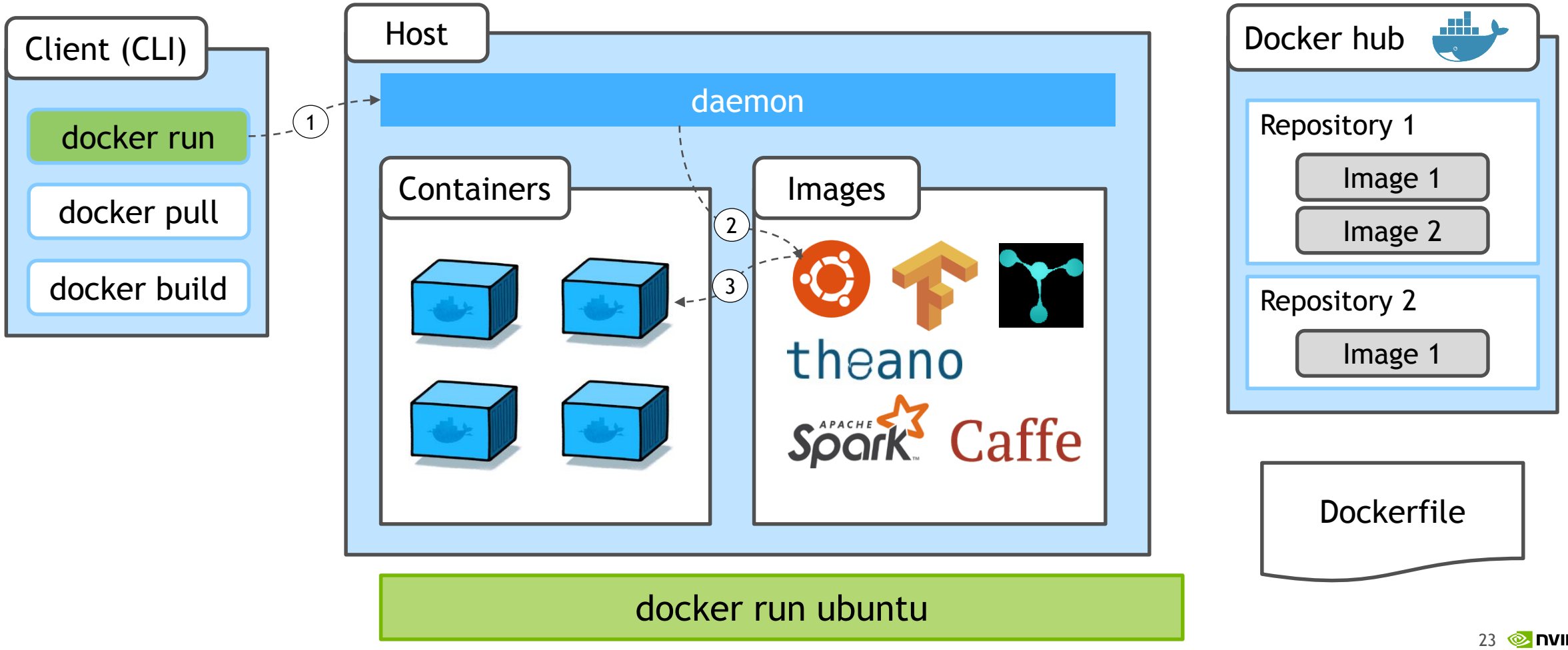
DGX DOCKER & NVIDIA/DOCKER



- Framework optimized
- NCCL Library
  - 1.3.1: pcie enabled library (public)
  - 1.6.1: NVLink enabled library (private)
  - 2.0.3: pcie/nvlink enabled library
    - inter-node & public

DOCKER CONTAINER LIFE CYCLE

Container run



DETACHED OR FORGROUND MODE

Detatched Mode (with `-d` option)

```
[~]$ docker run -d -ti ubuntu
934535be35a3b8591dc0f65bd93eca2938bb8b223368f537625f3d45f0f38808
[~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
934535be35a3        ubuntu              "/bin/bash"         4 seconds ago       Up 3 seconds                            evil_lumiere
```

Foreground Mode (without `-d` option)

```
[~]$ docker run ubuntu
[~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
9a01f7b5d5a4        ubuntu              "/bin/bash"         9 seconds ago       Exited (0) 8 seconds ago                        stupefied_shirley
```

Foreground mode with interactive terminal option

```
[~]$ docker run -ti ubuntu
root@997bb1544d10:/#
```

INTERACTIVE TERMINAL OPTION

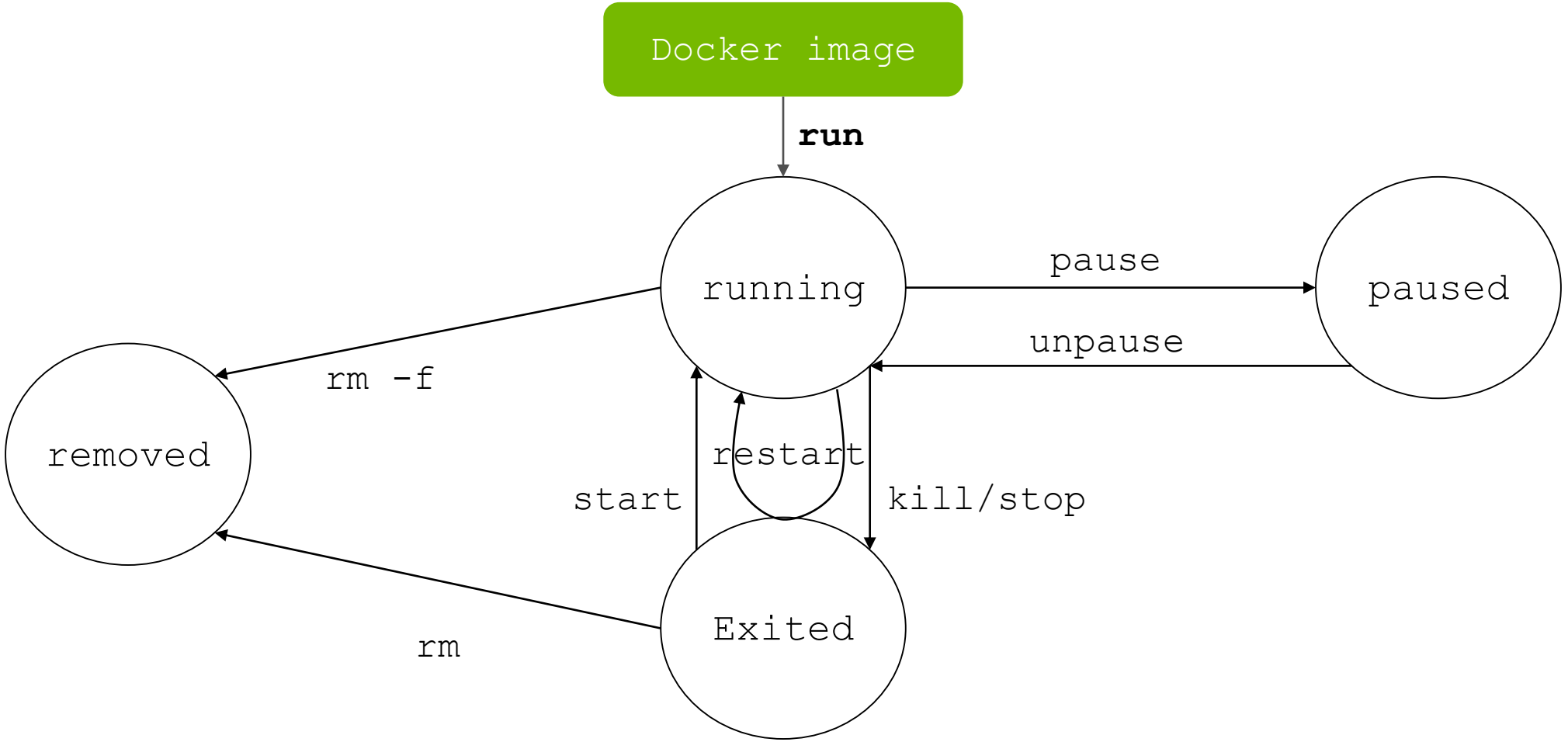
- -t option
  Enables shell can show
  container default shell's `stdout`
- -i option
  enables shell input key put
  to the launched container default shell
- -ti or -it option
  enables interactive terminal to the container default shell

CLEAN UP OPTION

Automated container remove when it exits

```
[~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
[~]$ docker run --rm --name ubuntu ubuntu which bash
/bin/bash
[~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
[~]$
```

DOCKER CONTAINER LIFE CYCLE



CONTAINER LIFE CYCLE

Example

Container status transition test with 4 containers…

```
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
ae03b60a6af2        ubuntu              "/bin/bash"         5 seconds ago       Up 4 seconds                            container_1
0d05b7af2cad        ubuntu              "/bin/bash"         8 seconds ago       Up 7 seconds                            container_2
6e8bd7cc79df        ubuntu              "/bin/bash"         11 seconds ago      Up 11 seconds                           container_3
69fcef2c66cb        ubuntu              "/bin/bash"         14 seconds ago      Up 13 seconds                           container_4
```

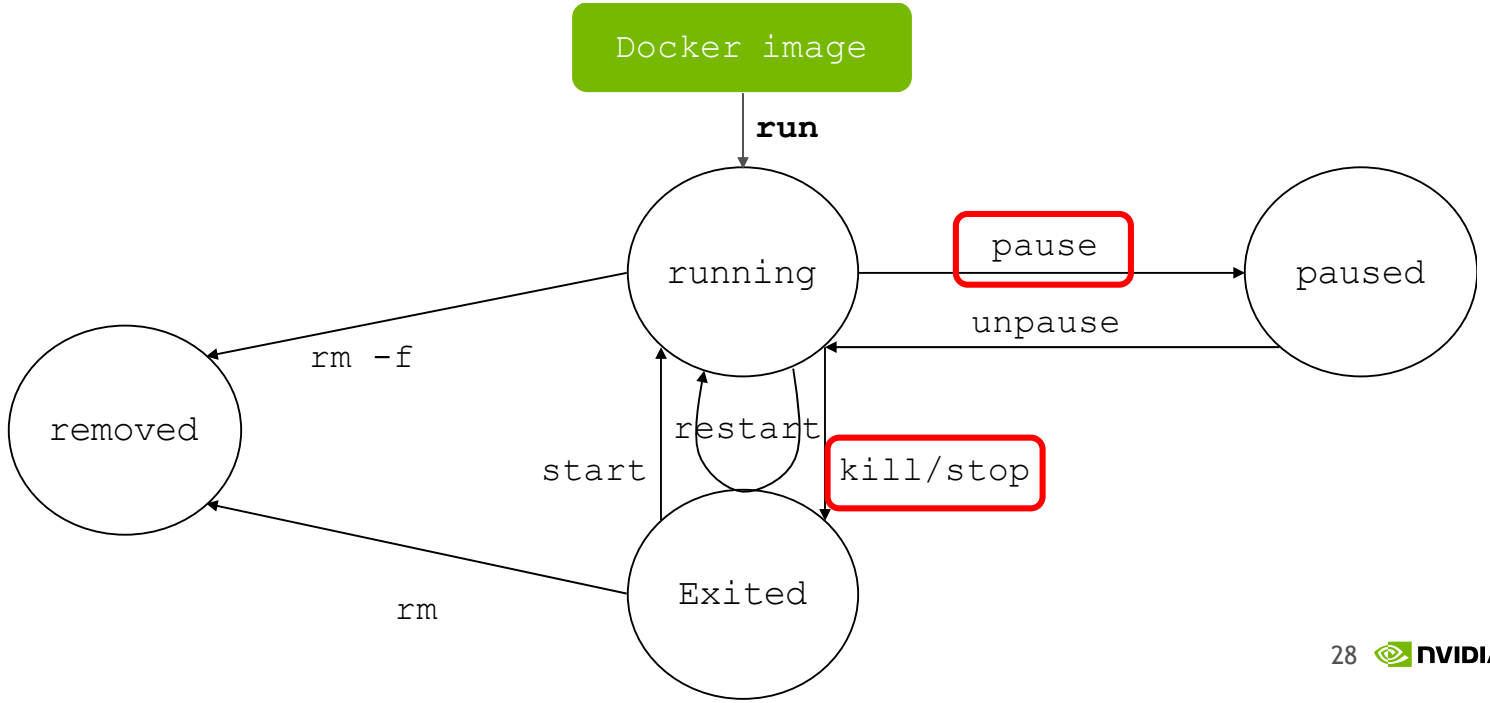
Test will be done below,

container_1 ← no operation

container_2 ← stop

container_3 ← pause

container_4 ← kill



CONTAINER LIFE CYCLE EXPERIMENT #1

```
[~]$ docker stop container_2 && docker pause container_3 && docker kill container_4
container_2
container_3
container_4
[~]$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                        PORTS               NAMES
ae03b60a6af2        ubuntu              "/bin/bash"         About a minute ago  Up About a minute                                 container_1
6e8bd7cc79df        ubuntu              "/bin/bash"         About a minute ago  Up About a minute (Paused)                        container_3
[~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                        PORTS               NAMES
ae03b60a6af2        ubuntu              "/bin/bash"         About a minute ago  Up About a minute                                 container_1
0d05b7af2cad        ubuntu              "/bin/bash"         About a minute ago  Exited (0) 8 seconds ago                          container_2
6e8bd7cc79df        ubuntu              "/bin/bash"         About a minute ago  Up About a minute (Paused)                        container_3
69fcef2c66cb        ubuntu              "/bin/bash"         About a minute ago  Exited (137) 7 seconds ago                        container_4
```

`docker stats` at this situation

```
CONTAINER           CPU %               MEM USAGE / LIMIT     MEM %               NET I/O             BLOCK I/O           PIDS
ae03b60a6af2        0.00%               1.562 MiB / 15.55 GiB   0.01%               23.16 kB / 648 B    0 B / 0 B           1
6e8bd7cc79df        0.00%               1.363 MiB / 15.55 GiB   0.01%               27.13 kB / 648 B    0 B / 0 B           1
```

CONTAINER LIFE CYCLE EXPERIMENT #2

container remove w/ or w/o force option

```
$ docker rm {container-name}
```

```
[~]$ docker rm container_1 container_2 container_3 container_4
container_2
container_4
Error response from daemon: You cannot remove a running container ae03b60d34e81e. Stop the container before attempting removal or use -f
Error response from daemon: You cannot remove a running container 6e8bd7cc79d0775. Stop the container before attempting removal or use -f
```

```
$ docker rm -f {container-name}
```

```
[~]$ docker rm -f container_1 container_3
container_1
Error response from daemon: Could not kill running container 6e8bd7cc79df33549acc40f3c7475a3f457dd01cad99fe7524e223adf34d0775, cannot remove - Container 6e8bd7cc79df33549acc40f3c7475a3f457dd01cad99fe7524e223adf34d0775 is paused. Unpause the container before stopping
```

```
[~]$ docker unpause container_3 && docker rm -f container_3
container_3
container_3
```

CONTAINER IDENTIFICATION & USER

Container naming

```
docker run --name cuda -v $(pwd):/workspace -v /mnt/vol:/data image-name
```

```
[[~]$ docker run -d --name cuda nvcr.io/nvidia/cuda:8.0-cudnn5.1-devel-ubuntu14.04
cb850d363a2bfe664eda1432923e3751541489871ccc044a73021fc513933e3e
[[~]$ docker ps -a
CONTAINER ID        IMAGE                                              COMMAND             CREATED             STATUS                      PORTS               NAMES
cb850d363a2b        nvcr.io/nvidia/cuda:8.0-cudnn5.1-devel-ubuntu14.04   "/bin/bash"         3 seconds ago       Exited (0) 2 seconds ago                        cuda
7d371be14c23        nvcr.io/nvidia/cuda:8.0-cudnn5.1-devel-ubuntu14.04   "/bin/bash"         38 minutes ago      Exited (0) 38 minutes ago                       peaceful_kare
```

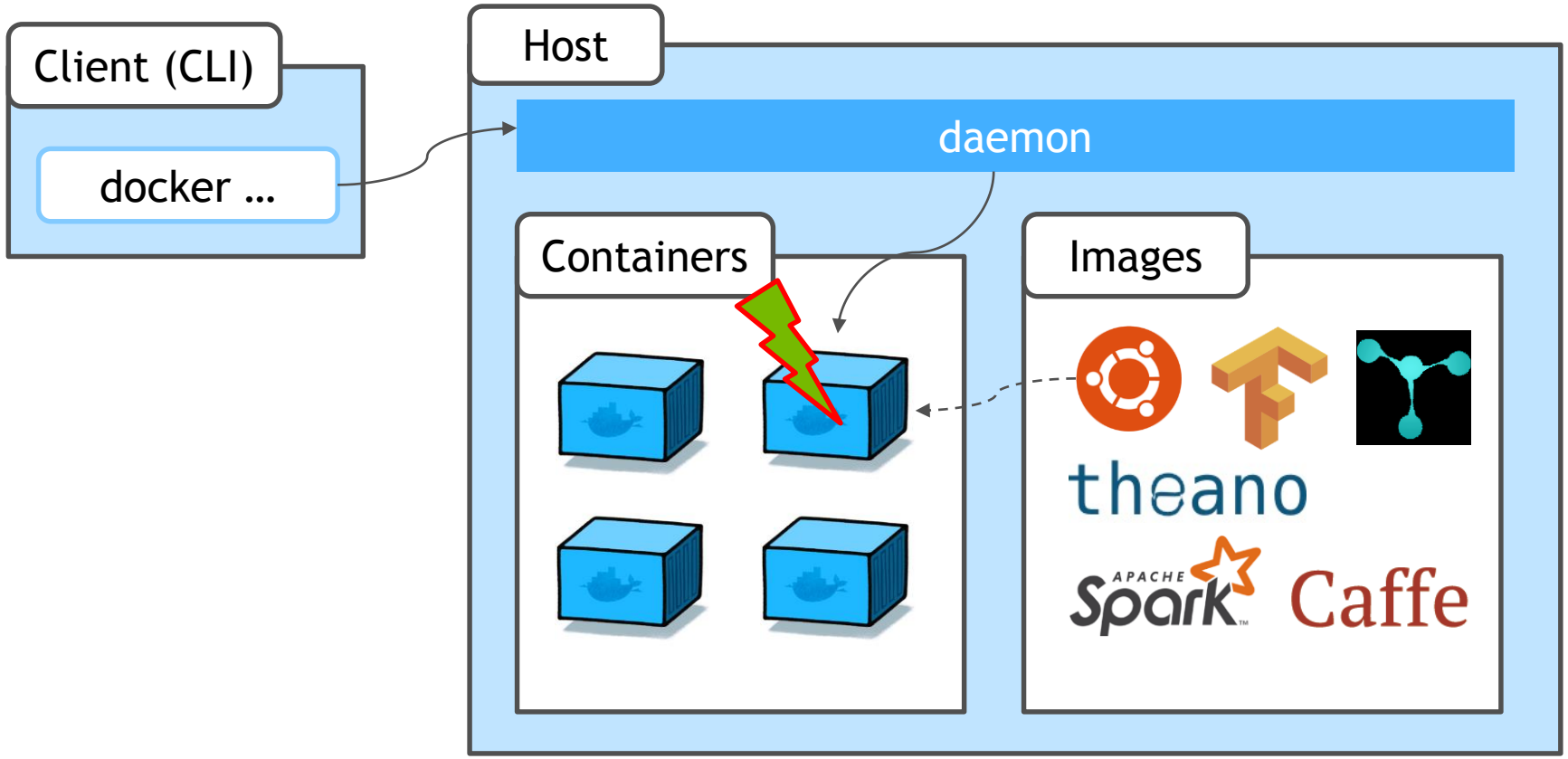
User forwarding

```
docker run -u $(id -u):$(id -g) image-name
```

HOST RESOURCE MOUNT FROM CONTAINERS

CONTAINER AS A ISOLATED ENVIRONMENT

Docker provides isolated environment



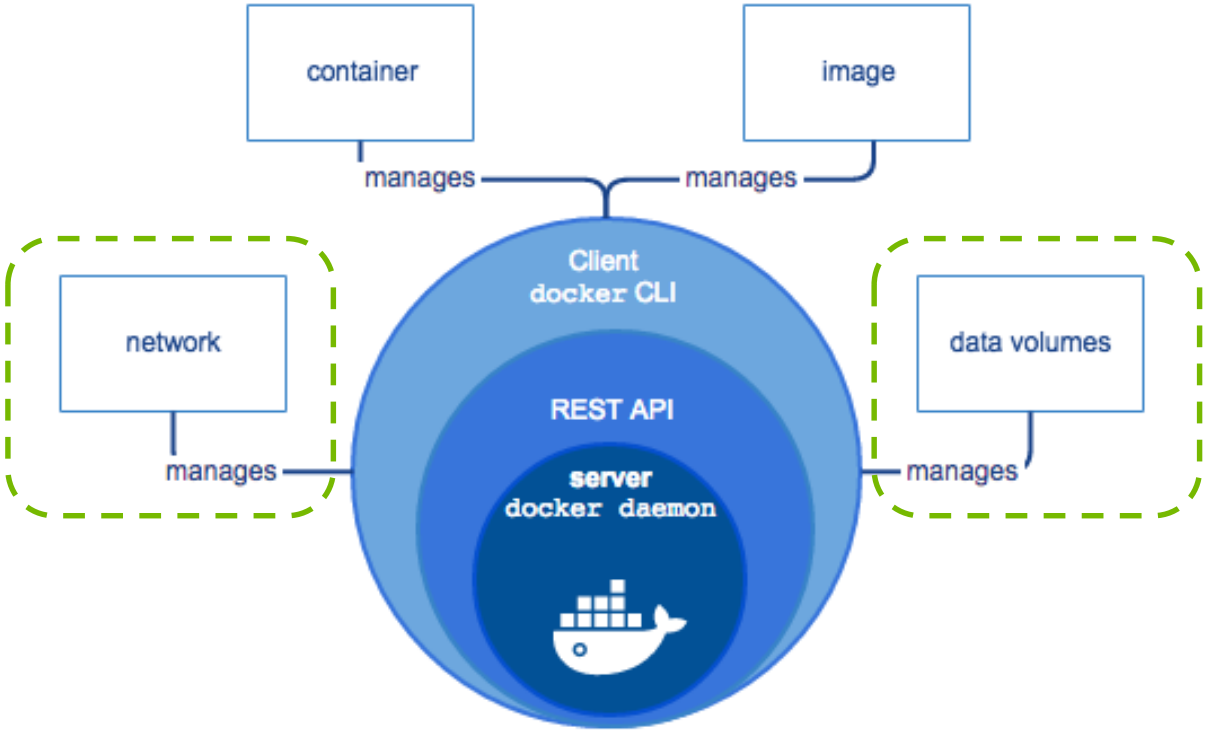
Host and container isolation

Volume & network should be shared

```
[[~]$ docker run --rm -ti nvcr.io/nvidia/cuda:8.0-cudnn5.1-devel-ubuntu14.04
[root@634de16f1779:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@634de16f1779:/#
```

DOCKER MANAGEMENT RESOURCES

Reminding.. docker can manage data volume



MOUNT FOR HOST RESOURCES

Volume mount:
`-v {host-volume}:{container-volume}[:ro]`

```
docker run -v $(pwd):/workspace -v /mnt/vol:/data image-name
```

Home creation: `user mounts to home space with HOME environment`

```
docker run  -u $(id -u):$(id -g) -e HOME=$HOME -v $HOME:$HOME image-name
```

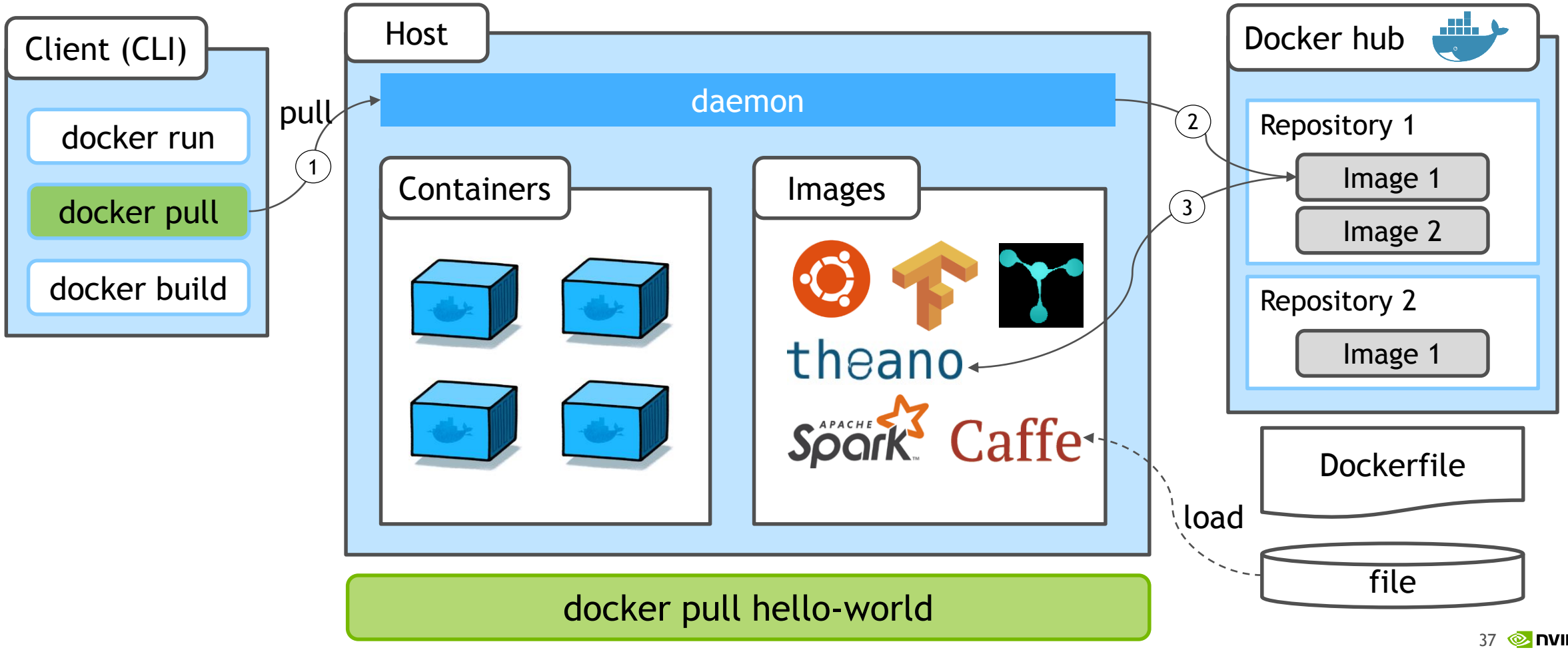
Port forwarding: `-p {host-port}:{container-port}`

```
docker run -p 8888:8888 image-name
```

DOCKER IMAGE MANAGEMENT

GETTING DOCKER IMAGE

pulling / loading

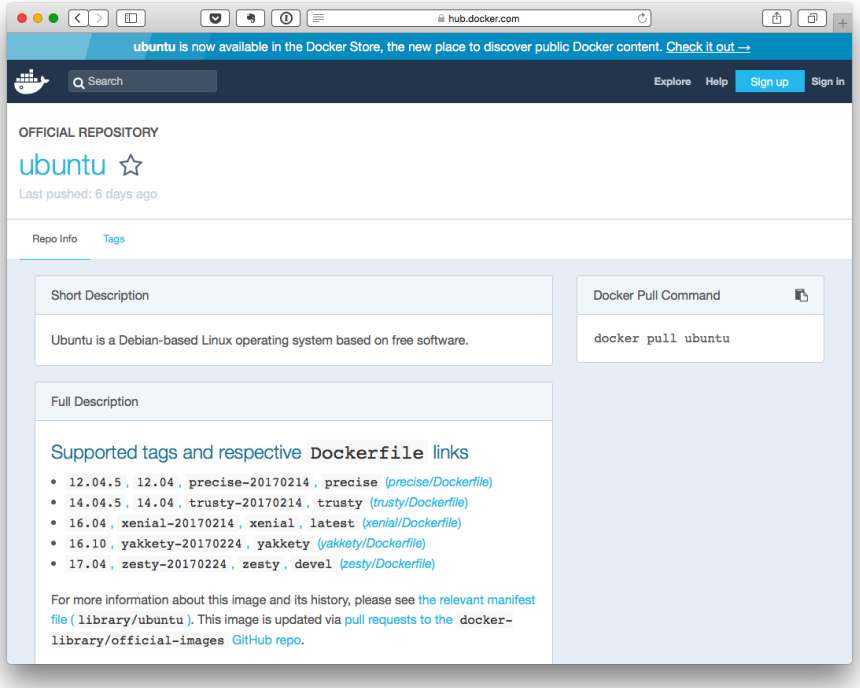


DOCKER PULL

Loading docker image to the host

docker pull ubuntu

```
[[~/workspace]$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
d54efb8db41d: Pull complete
f8b845f45a87: Pull complete
e8db7bf7c39f: Pull complete
9654c40e9079: Pull complete
6d9ef359eaaa: Pull complete
Digest: sha256:dd7808d8792c9841d0b460122f1acf0a2dd1f56404f8d1e56298048885e45535
Status: Downloaded newer image for ubuntu:latest
```




```
[[~/workspace]$ docker images
REPOSITORY          TAG                                 IMAGE ID            CREATED             SIZE
nvidia/cuda         8.0-cudnn5-devel-ubuntu14.04        9e4232af34a3        4 days ago          1.77 GB
ubuntu              16.04                               0ef2e08ed3fa        6 days ago          130 MB
ubuntu              latest                              0ef2e08ed3fa        6 days ago          130 MB
ubuntu              14.04                               7c09e61e9035        6 days ago          188 MB
hello-world         latest                              48b5124b2768        7 weeks ago         1.84 kB
```

EMBEDED IMAGE PULLING

Retrieves docker images registry when no image found on the host

```
[~/dgx/registry]$ docker run --rm -ti tensorflow/tensorflow:1.1.0-devel-gpu
Unable to find image 'tensorflow/tensorflow:1.1.0-devel-gpu' locally
1.1.0-devel-gpu: Pulling from tensorflow/tensorflow
c62795f78da9: Already exists
d4fceeeb758e: Already exists
5c9125a401ae: Already exists
0062f774e994: Already exists
6b33fd031fac: Already exists
7ab5dd833cf2: Already exists
df9cc763fcde: Already exists
9b0174a3640e: Already exists
1efd10acdd72: Already exists
f77b671e3092: Already exists
af9093817c44: Pull complete
b2fb381211f0: Downloading [============================>                      ] 59.83 MB/104.9 MB
9cffb3c924b3: Downloading [==================================>                ] 55.53 MB/78.84 MB
c806372eb166: Download complete
dc1f1d076fcb: Downloading [==>                                                ] 5.858 MB/132.6 MB
7c307214de53: Waiting
74da6021e273: Waiting
713545297e90: Waiting
3fff069066e3: Waiting
cb1cf9743861: Waiting
9930ba126722: Waiting
f6f3b5f97aa6: Waiting
84a20de63ca2: Waiting
```

GETTING ADDITIONAL DOCKER IMAGE

From docker hub

```
docker pull [image name][:tag]
```

From NVIDIA DGX registry

```
docker pull nvcr.io/nvidia/[framework]:[tag]
```

From local registry

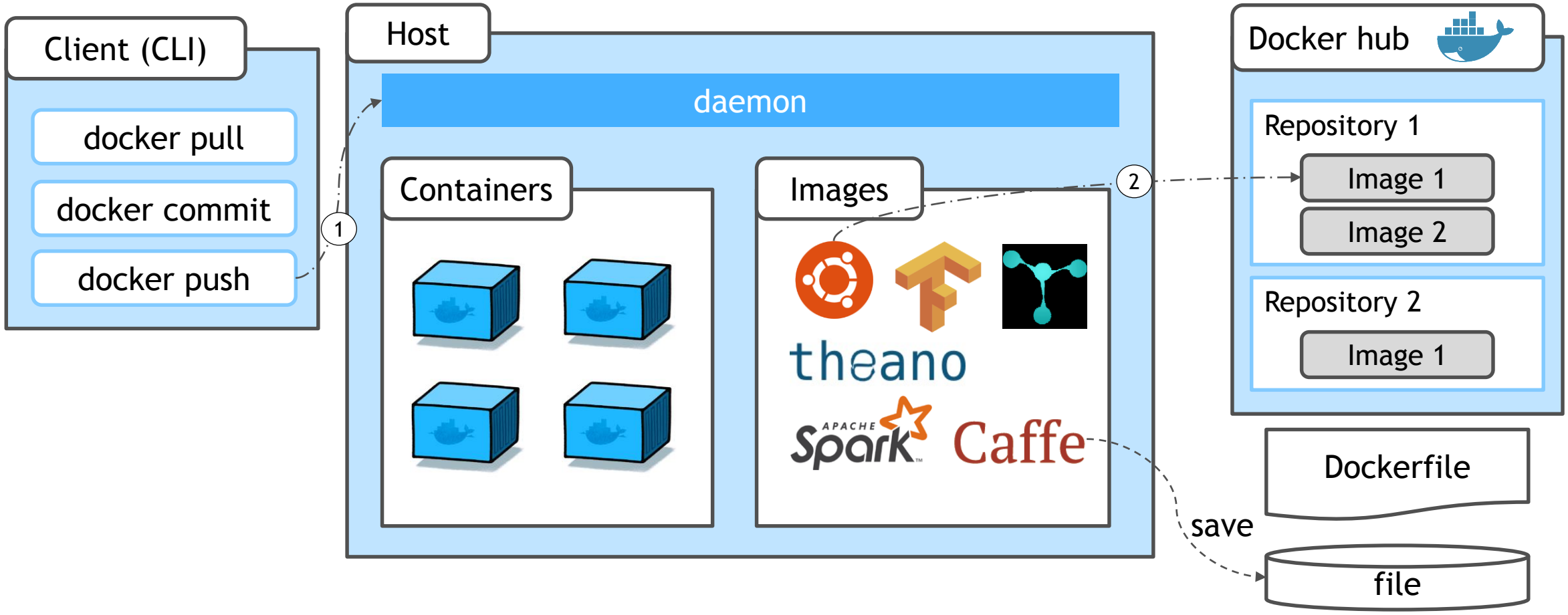
```
docker pull [local-registry addr]/[group name]/[image name][:tag]
```

From file

```
docker load --input {file-name}.tar.bz2
```

DOCKER IMAGE BACKUP

push / save



BACKUP DOCKER IMAGE

To docker hub

```
docker push [docker hub id]/[image name][:tag]
```

To NVIDIA DGX registry

```
docker push nvcr.io/[group name]/[image name][:tag]
```

To local registry

```
docker push [local-registry]/[group name]/[image name][:tag]
```

To file

```
docker save [image name] | bzip2 > {file-name}.tar.bz2
```

HOW TO PUSH OUTER IMAGE TO REGISTRY

Using local-registry

Pulling other's docker image

Getting basic dgx docker image from NVIDIA DGX Registry

```
docker pull nvcr.io/[group name]/[image-name][:tag]
```

Getting docker hub's public image

```
docker pull [docker hub id/][image-name][:tag]
```

Setting new docker image name

```
docker tag {/}[image-name][:tag] {local-registry}/[group-name]/[image-name][:tag]
```

Commit user modified docker image to DGX registry

```
docker push {local-registry}/[group-name]/[image-name][:tag]
```

HOW TO PUSH OUTER IMAGE TO REGISTRY

Using NVIDIA DGX Registry

Pulling other's docker image

Getting basic dgx docker image from NVIDIA DGX Registry

```
docker pull nvcr.io/[group name]/[image-name][:tag]
```

Getting docker hub's public image

```
docker pull [docker hub id/][image-name][:tag]
```

Setting new docker image name

```
docker tag {/}[image-name][:tag] nvcr.io/[group-name]/[image-name][:tag]
```

Commit user modified docker image to DGX registry

```
docker push nvcr.io/[group-name]/[image-name][:tag]
```

DOCKER IMAGE REMOVE

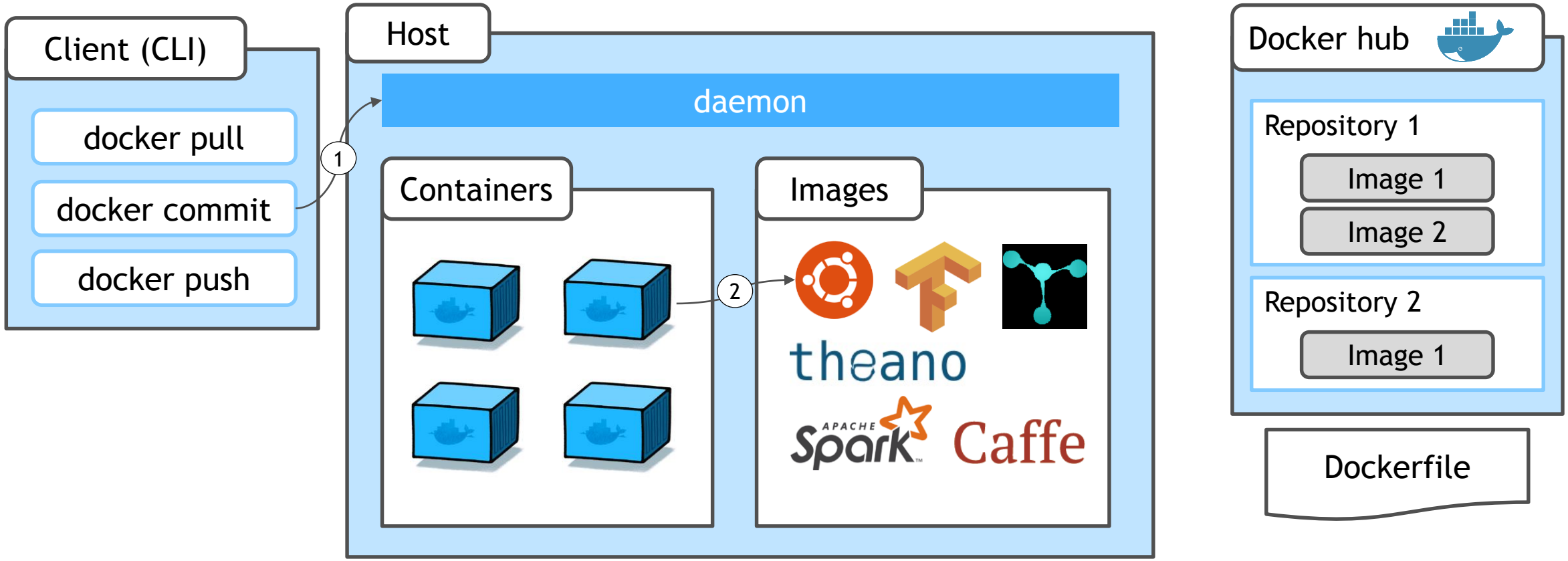
```
docker rmi [image-name][:tag]
```

```
[[~/docker]$ docker images
 REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
 ubuntu              latest              d355ed3537e9        3 weeks ago         119MB
[[~/docker]$ docker rmi ubuntu:latest
 Untagged: ubuntu:latest
 Untagged: ubuntu@sha256:a0ee7647e24c8494f1cf6b94f1a3cd127f423268293c25d924fbe18fd82db5a4
 Deleted: sha256:d355ed3537e94e76389fd78b77241eeba58a11b8faa501594bc82d723eb1c7f2
 Deleted: sha256:dd864b96a38e849779c42a04159bbb39c7ab47253bf222049b471d8f26b60d14
 Deleted: sha256:80e85c818fa0447c96a42501ca7457ad83e5834aa76f22c366342106889b7411
 Deleted: sha256:11a2a269cf6ec2cefcb4e24370b8b2d7a4875450bafd3a70bd42eb787481d798
 Deleted: sha256:1118f33a0ee7a874a04318248a886b2bdaf44cba286644ab7ded870aefe64b62
 Deleted: sha256:cb11ba6054003d39da5c681006ea346e04fb3444086331176bf57255f149c670
[[~/docker]$ docker images
 REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
 [~/docker]$
```

DOCKER IMAGE CREATION

CONTAINER BACKUP

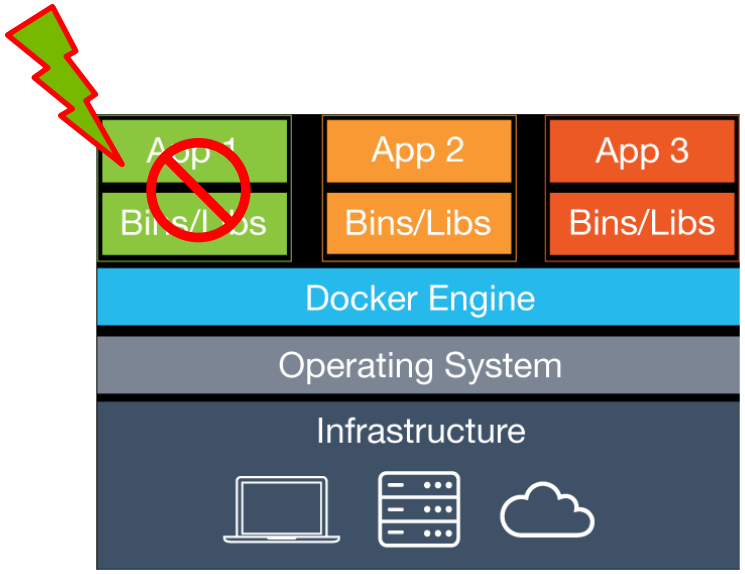
commit



docker commit [container-name] [image-name][:tag]

WHY THIS IS MATTER?

Container will lost all works when exit



Solution

1. Code & dataset

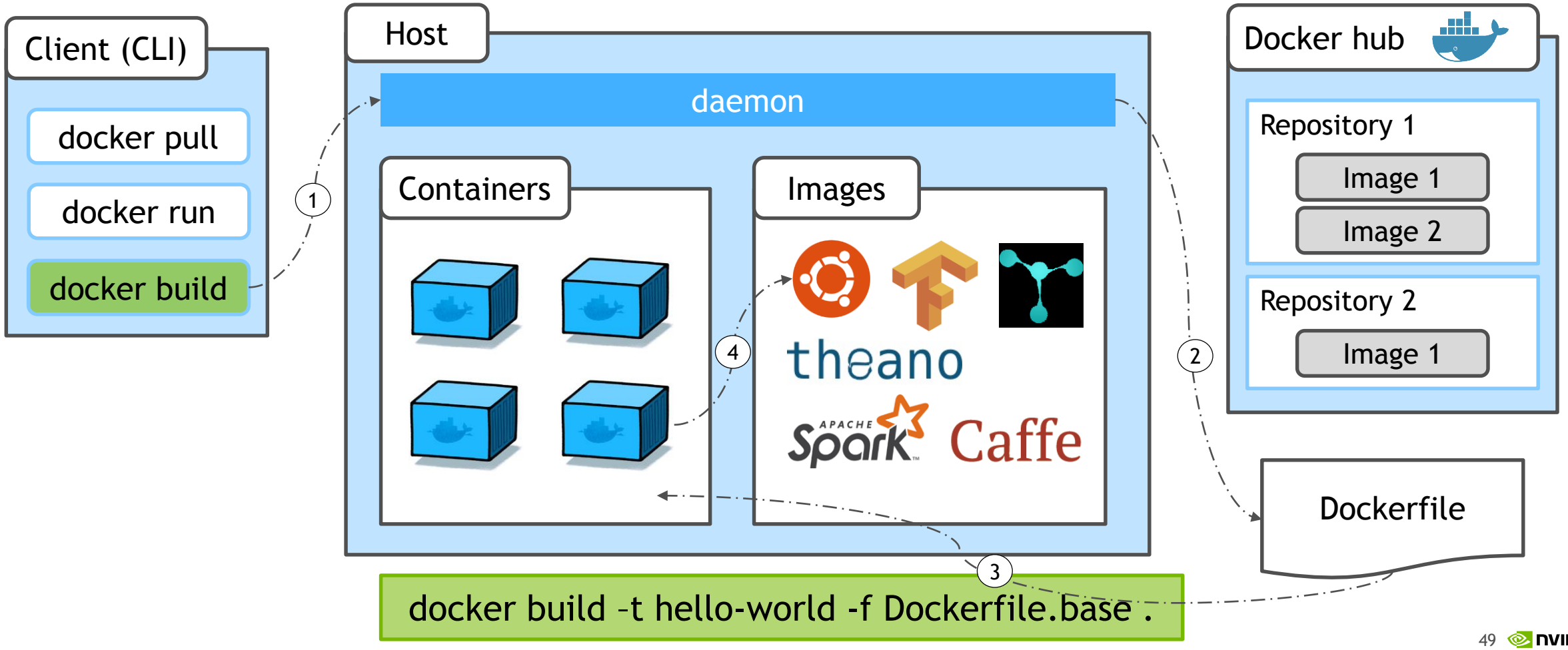
➔ USE volume mount to host volume

2. Environment work

➔ docker commit

BUILDING NEW IMAGE

Docker image build with script description



DOCKERFILE

docker image build script file

Any image can be customized from base image

Use provided CUDA docker image to build custom GPU accelerated image

```
FROM nvcr.io/nvidia/cuda:8.0-cudnn6.0-devel-ubuntu14.04
MAINTAINER Jack Han <jahan@nvidia.com>

RUN apt-get update && apt-get install -y --no-install-recommends \
    wget \
```

Detailed Guidance are,
Dockerfile reference
Best practice for writing Dockerfile

Examples
nvidia/cuda Tensorflow
nvidia/caffe pytorch

WRITING DOCKERFILE

Description of building development environment

Options

```
FROM base docker image
RUN  launch operation
COPY copy specified directory to image
CMD  set default launch command
USER specify container user name
     (default: root)
ARG  Dockerfile argument
EXPOSE expose port or volume to host
WORKDIR move current workdir
```

※ Excluded specified file or directory when `COPY`
Put .dockerignore file in current path

```
# comment
*/temp*       # exclude subdirectory
temp?         # exclude with filter
!temp         # exception of exclude
```

Dockerfile example (tensorflow-gpu)

```
FROM nvidia/cuda:8.0-cudnn6-devel-ubuntu16.04

MAINTAINER Jan Prach <jendap@google.com>

# In the Ubuntu 14.04 images, cudnn is placed in system paths. Move them to
# /usr/local/cuda
RUN cp -P /usr/include/cudnn.h /usr/local/cuda/include
RUN cp -P /usr/lib/x86_64-linux-gnu/libcudnn* /usr/local/cuda/lib64

# Copy and run the install scripts.
COPY install/*.sh /install/
ARG DEBIAN_FRONTEND=noninteractive
RUN /install/install_bootstrap_deb_packages.sh
RUN add-apt-repository -y ppa:openjdk-r/ppa && \
    add-apt-repository -y ppa:george-edison55/cmake-3.x
RUN /install/install_deb_packages.sh
RUN /install/install_pip_packages.sh
RUN /install/install_bazel.sh
RUN /install/install_golang.sh

# Set up the master bazelrc configuration file.
COPY install/.bazelrc /etc/bazel.bazelrc
ENV LD_LIBRARY_PATH /usr/local/cuda/extras/CUPTI/lib64:$LD_LIBRARY_PATH

# Configure the build for our CUDA configuration.
ENV TF_NEED_CUDA 1
ENV TF_CUDA_COMPUTE_CAPABILITIES 3.0
```

DOCKER IMAGE CREATION

```
$ docker build -t IMAGE[:TAG] -f Dockerfile {dockerfile path}
```

Pulling the base image

Install dependencies

Build source codes & install

Environment setting

Default command setting

default (no spcification): /bin/bash

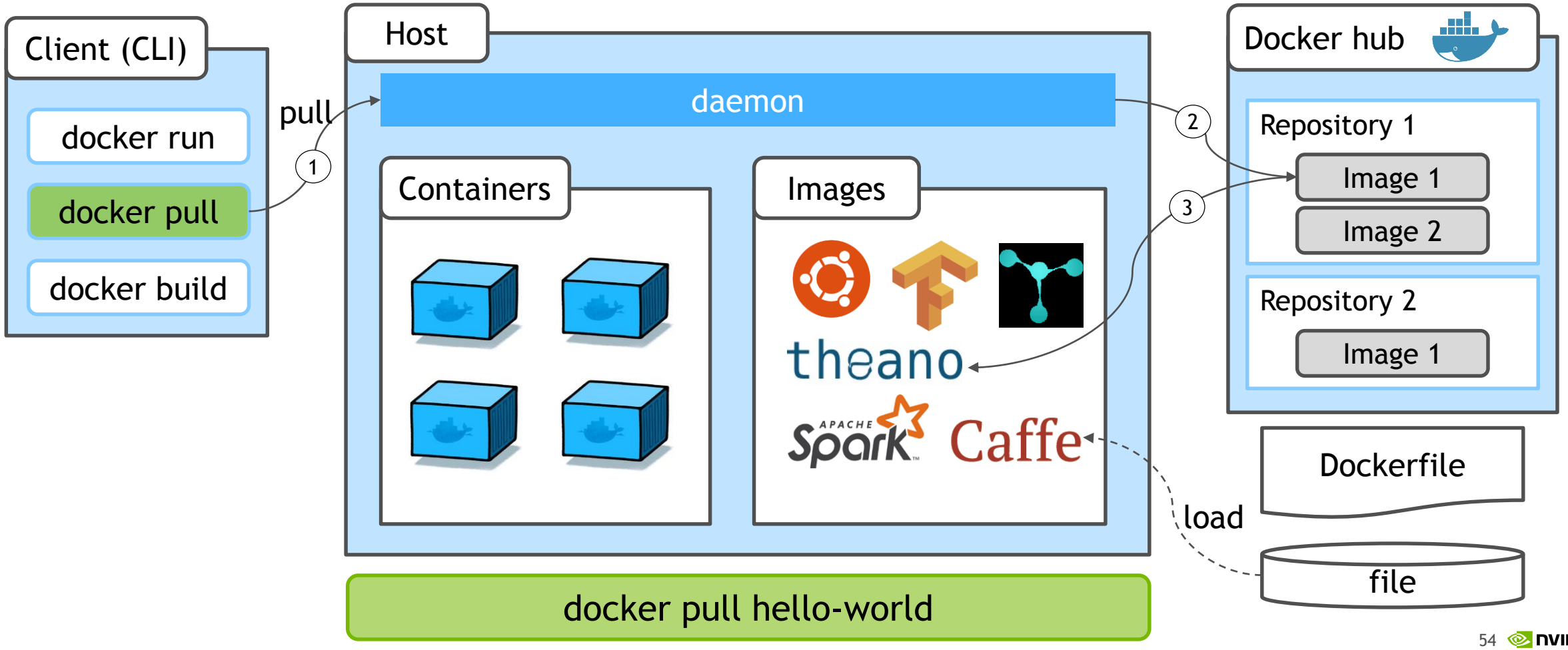
Dockerfile example (tensorflow-gpu)

```
FROM nvidia/cuda:8.0-cudnn6-devel-ubuntu16.04

MAINTAINER Jan Prach <jendap@google.com>

# In the Ubuntu 14.04 images, cudnn is placed in system paths. Move them to
# /usr/local/cuda
RUN cp -P /usr/include/cudnn.h /usr/local/cuda/include
RUN cp -P /usr/lib/x86_64-linux-gnu/libcudnn* /usr/local/cuda/lib64

# Copy and run the install scripts.
COPY install/*.sh /install/
ARG DEBIAN_FRONTEND=noninteractive
RUN /install/install_bootstrap_deb_packages.sh
RUN add-apt-repository -y ppa:openjdk-r/ppa && \
    add-apt-repository -y ppa:george-edison55/cmake-3.x
RUN /install/install_deb_packages.sh
RUN /install/install_pip_packages.sh
RUN /install/install_bazel.sh
RUN /install/install_golang.sh

# Set up the master bazelrc configuration file.
COPY install/.bazelrc /etc/bazel.bazelrc
ENV LD_LIBRARY_PATH /usr/local/cuda/extras/CUPTI/lib64:$LD_LIBRARY_PATH

# Configure the build for our CUDA configuration.
ENV TF_NEED_CUDA 1
ENV TF_CUDA_COMPUTE_CAPABILITIES 3.0
```

LOCAL REGISTRY

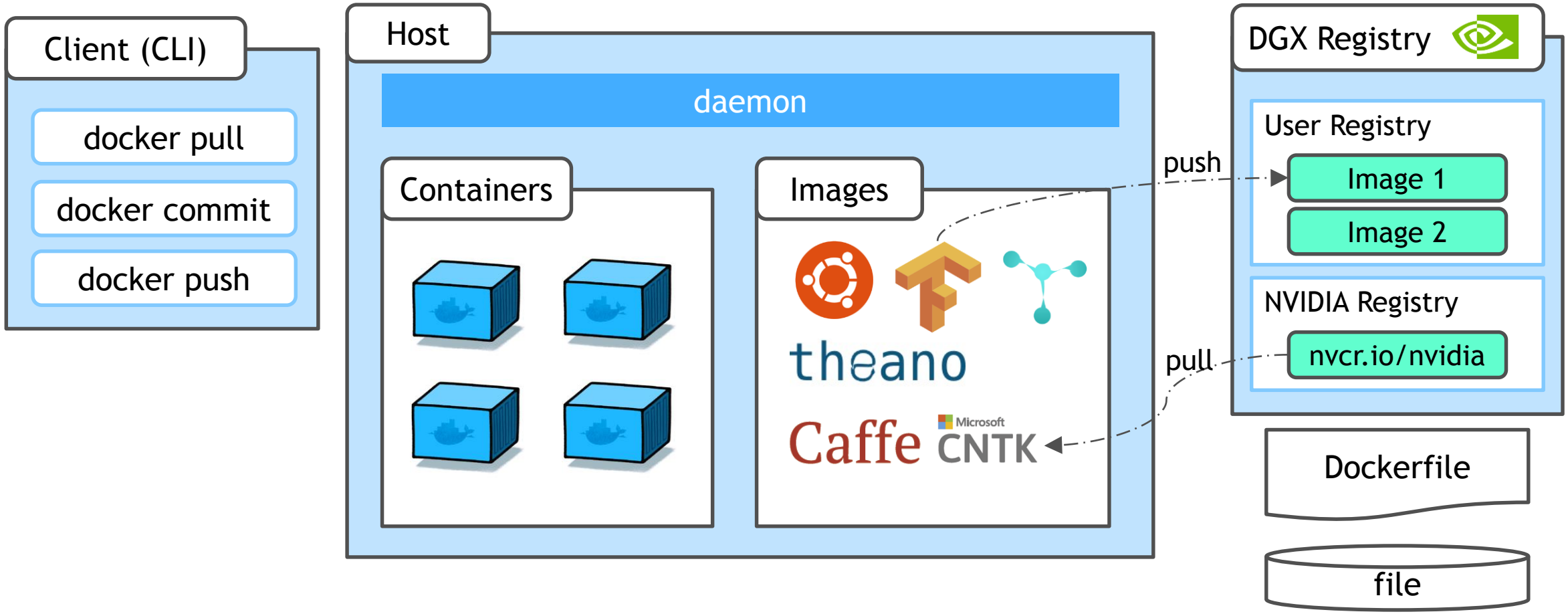
DOCKER HUB

pulling / loading



DGX REGISTRY

Private docker registry & Update service



DGX REGISTRY

Provides monthly updated deep learning frameworks via docker images

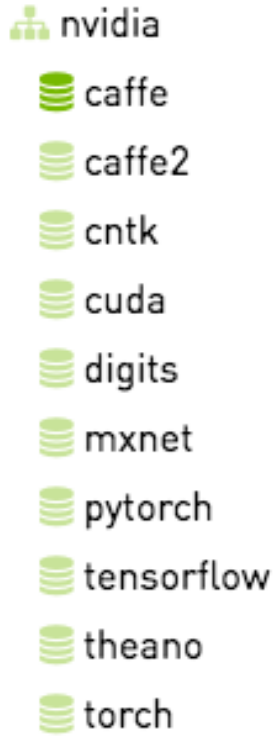
DGX users can use private registry service

Requires registry login for every host user

```
docker login nvcr.io
```

Put given password after registration from nvidia support

```
Username: $oauthtoken
Password: k7cqFTUvKKdiwGsPnWnyQFYGnlAlsCIRmlP67Qxa
```



DGX Registry Documentation
(http://docs.nvidia.com/dgx/dgx-registry-user-guide/index.html)

DGX REGISTRY USAGE GUIDE

Getting updated docker image from nvcr.io

```
docker pull nvcr.io/nvidia/[image-name]:[tag]
```

Setting new docker image name

```
docker tag nvcr.io/nvidia/[image-name]:[tag] nvcr.io/[group-name]/[image-name]:[tag]
```

Commit user modified docker image to DGX registry

```
docker push nvcr.io/[group-name]/[image-name]:[tag]
```

HOST CONFIGURATION

Configuration guide for dev-machines

Setting Docker daemon options,

- Change docker volume storage driver to overlay 2 (optional; recommended)
- Let docker daemon work with non-official registry (for local registry)
- Let recent docker deamon(docker-ce >= 17.03) works with DGX registry

Ubuntu 14.04

```
DOCKER_OPTS="--storage-driver=overlay2 \
    --insecure-registry='<addr-local-registry>' --disable-legacy-registy=false"
```

Ubuntu 16.04

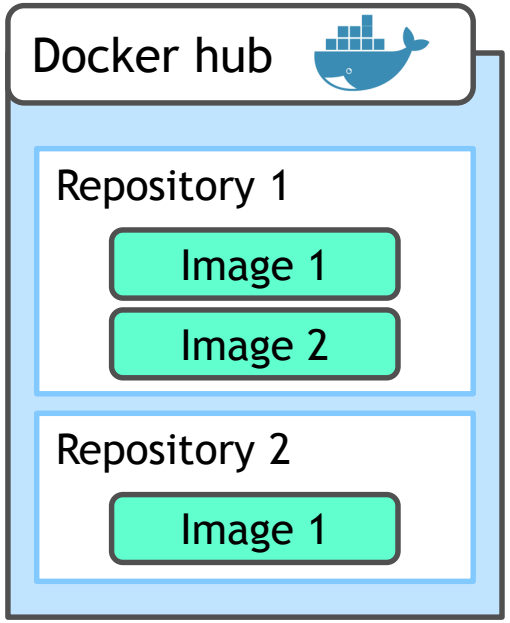
```
{
  "storage-driver": "overlay2",
  "insecure-registry": "<local-registry-addr>",
  "disable-legacy-registry": false
}
```

DGX-1 uses docker 1.12.6

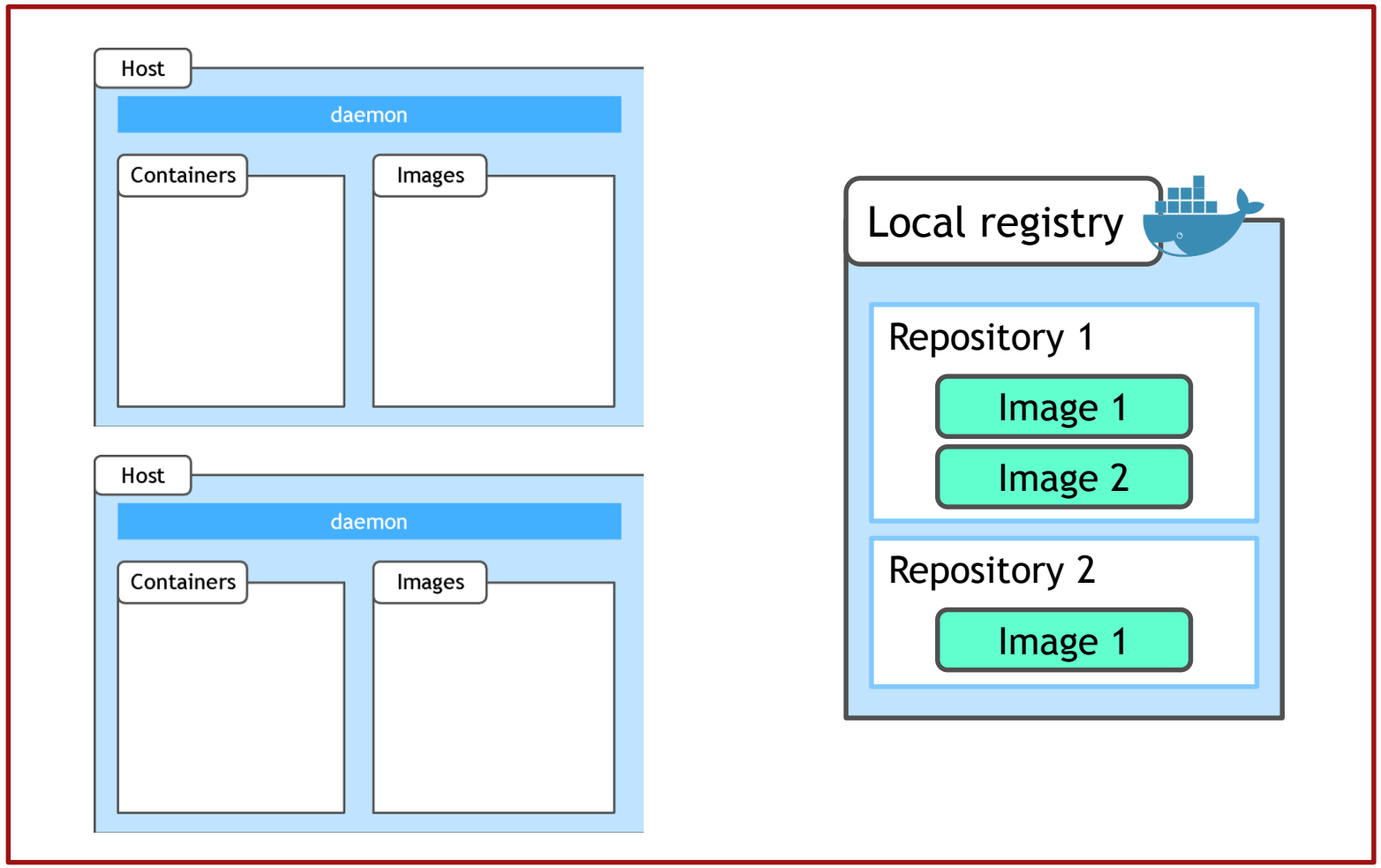
- storage volume is set, no need for DGX registry setting & local-registry is custom

LOCAL REGISTRY

SECURED NETWORK PROBLEM



LOCAL REPOSITORY!!



QUICK START LOCAL REPOSITORY

Getting repository image

```
docker pull registry
```

Start container

```
docker run -d -p 5000:5000 --restart=always --name localrepository registry
```

Tag target image

```
docker tag {source image} {local-registry}:{port}/{image-name}:{tag}
```

Push/pull target image

```
docker push {local-registry}:{port}/{image-name}:{tag}
```

List of local repository

Visit: http://localhost:5000/v2/_catalog

LOCAL REGISTRY VOLUME MOUNT

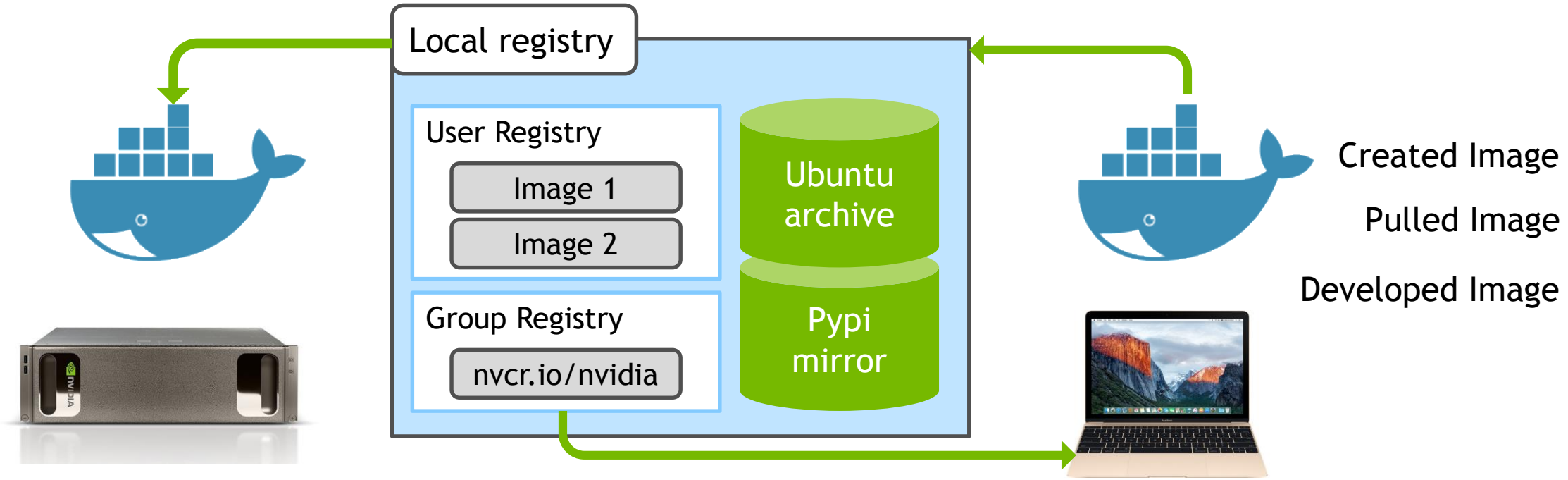
Of course, registry container can lost images when exit

Registry data is stored as a docker volume on the host system.

/var/lib/registry is default location

```
docker run -d -p 5000:5000 --restart=always \
    -v $(pwd)/registry:/var/lib/registry/ --name registry registry
```

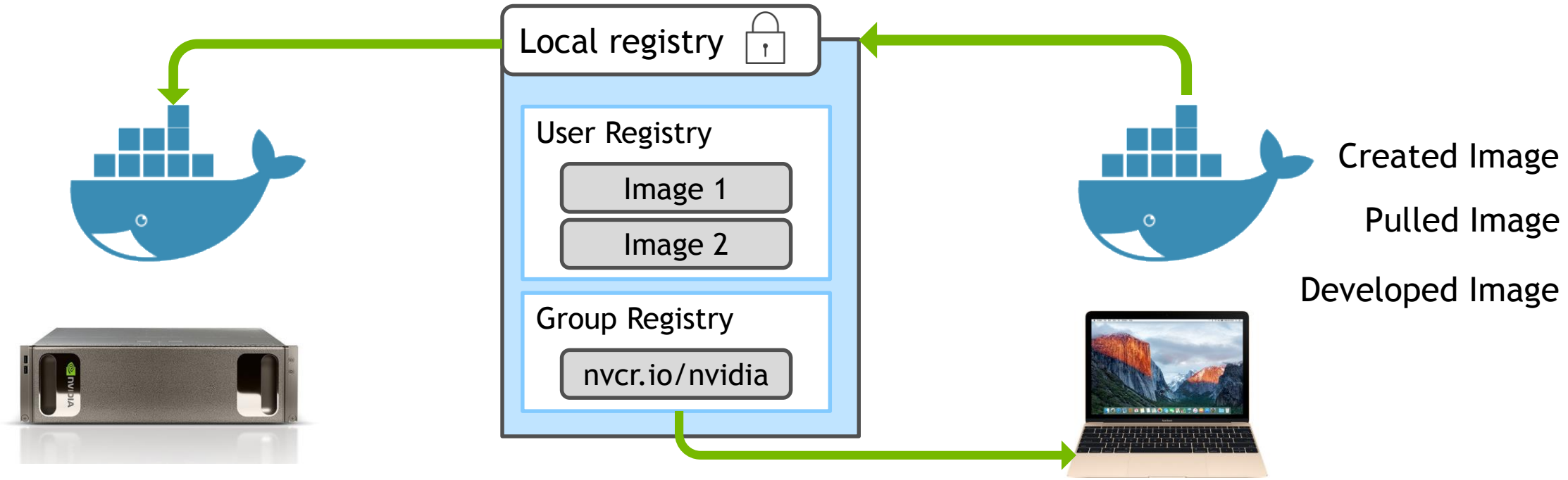
PRIVATE DOCKER REGISTRY EXAMPLE



DOCKER LOGIN FOR LOCAL REGISTRY

Host authorlization

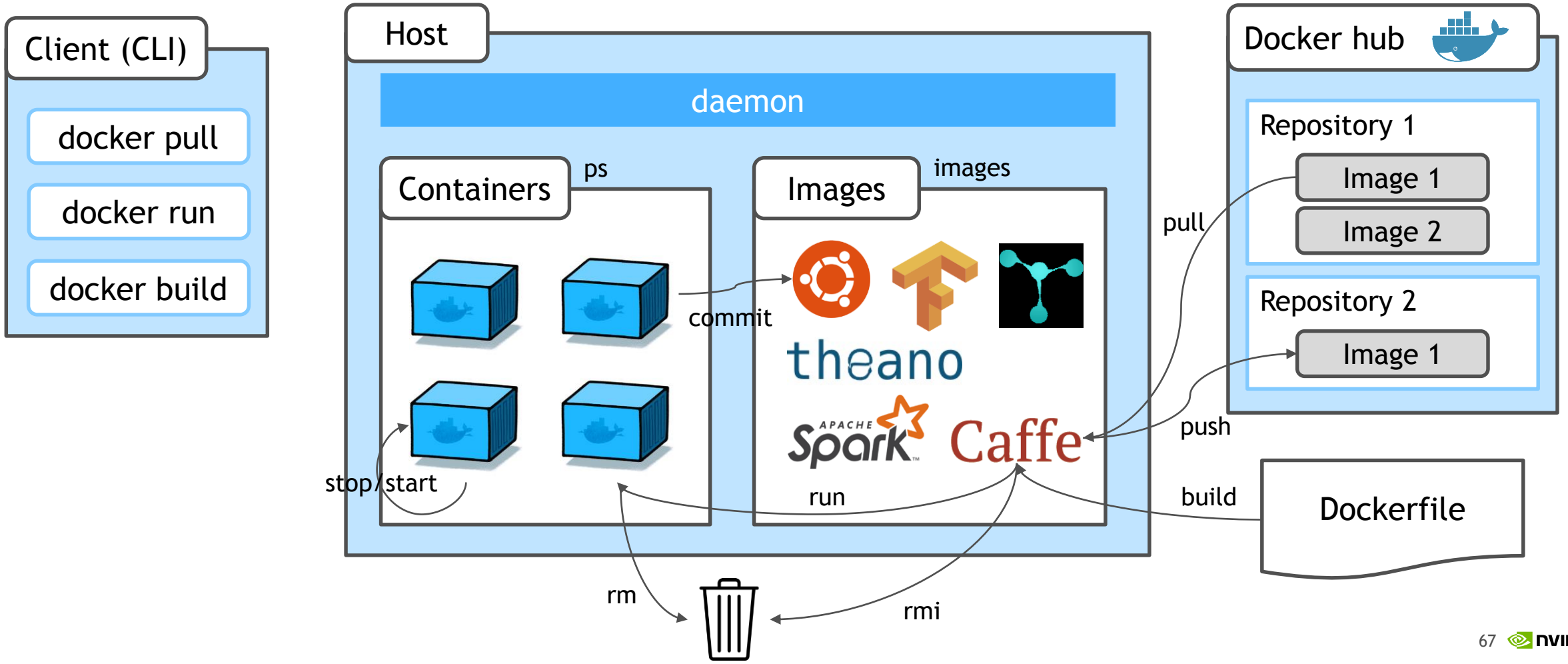
```
docker login <addr-local-registry>
```



CONCLUSION

DOCKER LIFE CYCLE

Overview



EXAMPLE OF DOCKER USE IN DGX-1

CUDA

```
nvidia-docker run --rm -ti –u $(id –u):$(id –g) --name cuda \
    nvcr.io/nvidia/cuda:8.0-cudnn6-devel-ubuntu16.04 nvidia-smi
```

Caffe

```
nvidia-docker run --rm -ti –u $(id –u):$(id –g) --name caffe
    -v $(pwd):/workspace
    nvcr.io/nvidia/caffe:17.06 caffe train --solver=solver.prototxt
```

Tensorflow

```
nvidia-docker run --rm -ti –u $(id –u):$(id –g) --name tensorflow \
    -p 8888:8888 -p 6006:6006 -v $(pwd):/workspace \
    nvcr.io/nvidia/tensorflow:17.06 python train.py
```

Digits

```
nvidia-docker run -d –u $(id –u):$(id –g) --name digits \
    --shm-size=1g --ulimit memlock=-1 --ulimit stack=67108864
    -p 5000:5000 -v /mnt/dataset:/data -v /mnt/digit-work:/workspace \
    nvcr.io/nvidia/digits:17.06
```

SOME USERFUL CLEANUP COMMANDS

Docker volume clean up

```
docker volume rm $(docker volume ls -qf dangling=true)
```

Docker image clean up

```
docker rmi -f $(docker images -q)
```

Docker images clean up which name is <none>

```
docker rmi -f $(docker images | grep "<none>" | awk "{print \$3}")
```

Docker container clean up which is Exited

```
docker rm $(docker ps -a -f status=exited)
```

THANK YOU