



# INTRODUCTION TO CONTAINERS

Lê Ngọc Sơn - [linson@fit.hcmus.edu.vn](mailto:linson@fit.hcmus.edu.vn)

# Content

---

- ❑ History of Containers
- ❑ Containers vs Virtual Machine
- ❑ Docker platform Overview and Terminology
- ❑ Introduction to Images
- ❑ Getting Started with Containers

# History of Containers Technology



Bộ môn MMT&VT - Khoa CNTT - ĐH KHTN Tp. HCM

# Introduction to Containers Technology



HOW TO DEPLOY AND ISOLATE AN APPLICATION  
ANYWHERE WITHOUT TAKING CARE ABOUT THE  
ENVIRONMENT?

**Container-based Virtualization**

# Container-based virtualization



Bộ môn MMT&VT - Khoa CNTT - ĐH KHTN Tp. HCM

# Container-based virtualization



Bộ môn MMT&VT - Khoa CNTT - ĐH KHTN Tp. HCM

# Container – Architecture



Bộ môn MMT&VT - Khoa CNTT - ĐH KHTN Tp. HCM



# Non-Isolated Applications



Bộ môn MMT&VT - Khoa CNTT - ĐH KHTN Tp. HCM



# Isolated Applications

# Containers vs Virtual Machines



Bộ môn MMT&VT - Khoa CNTT - ĐH KHTN Tp. HCM

# Container vs Virtual Machines



Bộ môn MMT&VT - Khoa CNTT - ĐH KHTN Tp. HCM



# Benefits of VM

- Better resource pooling
  - One physical machine divided into multiple virtual machines
- Easier to scale
- VM's in the cloud
  - Rapid elasticity
  - Pay as you go model





# Limitations of VMs

- Each VM stills requires
  - CPU allocation
  - Storage
  - RAM
  - An entire guest operating system
- The more VM's you run, the more resources you need
- Guest OS means wasted resources
- Application portability not guaranteed





# What is Docker ?

*Docker is a platform for developing, shipping and running applications using container virtualization technology*

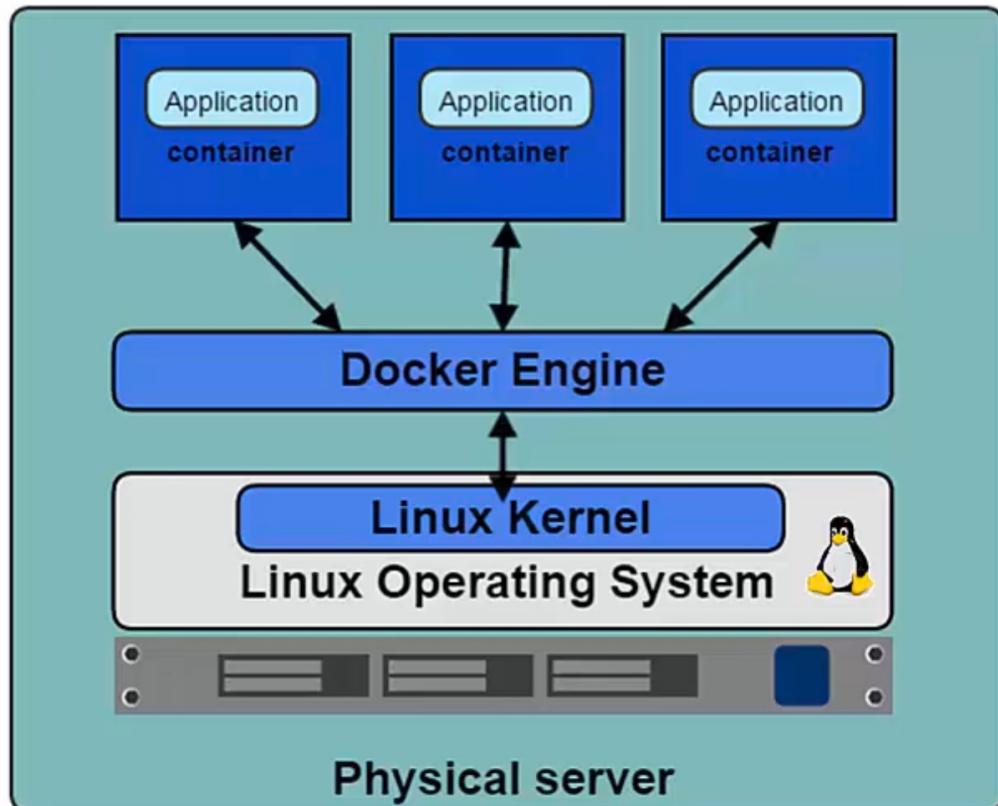
- The Docker Platform consists of multiple products/tools
  - Docker Engine
  - Docker Hub
  - Docker Machine
  - Docker Swarm
  - Docker Compose
  - Kitematic



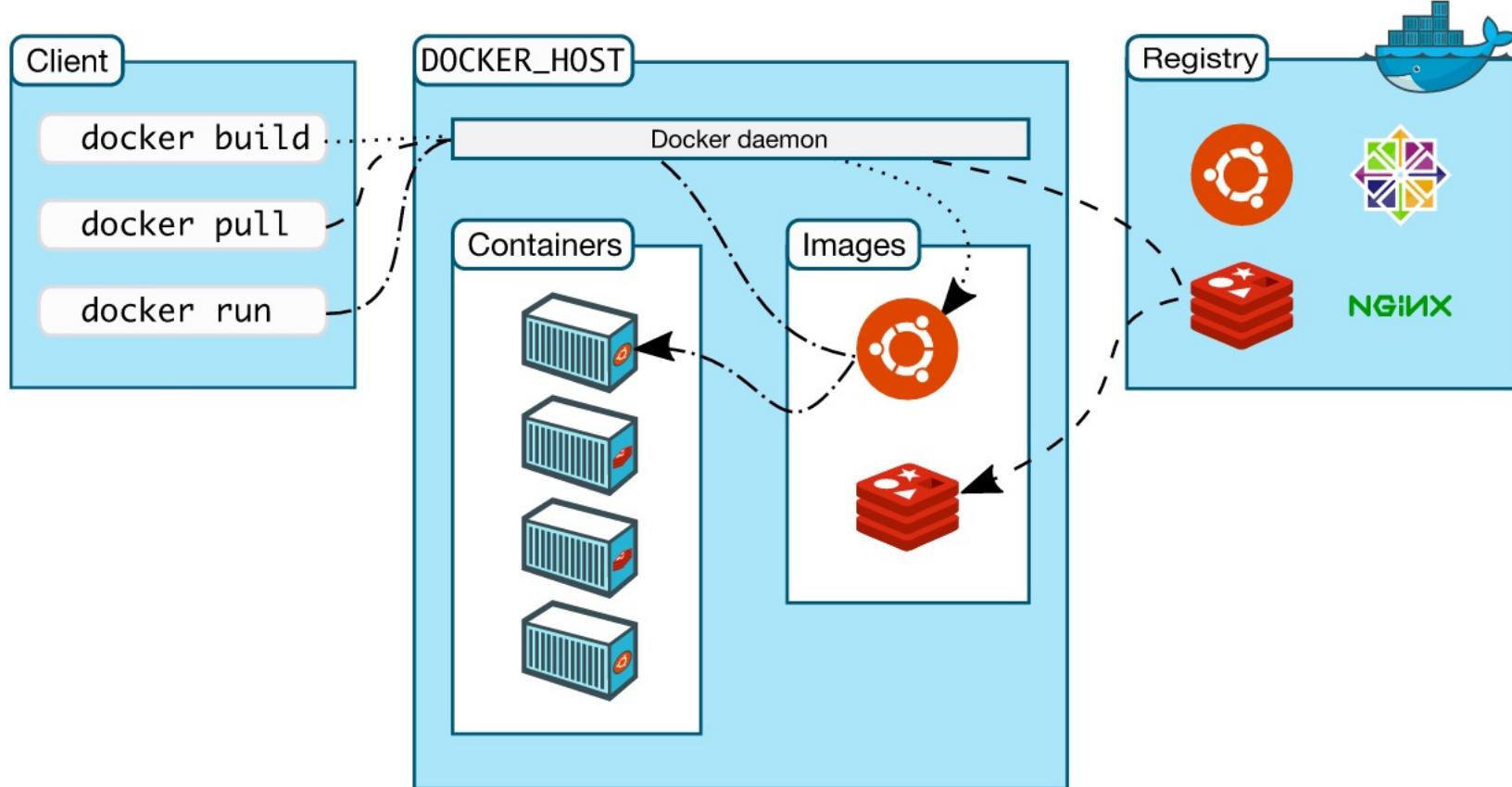
# Docker and the Linux kernel



- **Docker Engine** is the program that enables containers to be built, shipped and run.
- Docker Engine uses Linux Kernel namespaces and control groups
- Namespaces give us the isolated workspace



# Docker Architecture



# Docker Client and Daemon



- Client / Server architecture
- Client takes user inputs and send them to the daemon
- Daemon builds, runs and distributes containers
- Client and daemon can run on the same host or on different hosts
- CLI client and GUI (Kitematic)

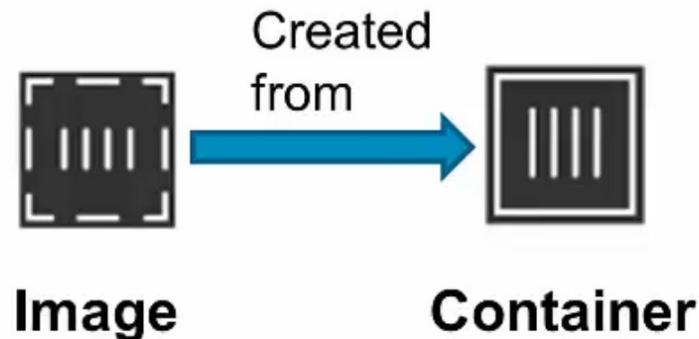
**Client**



# Docker Containers & Images



- **Images**
  - Read only template used to create containers
  - Built by you or other Docker users
  - Stored in the Docker Hub or your local Registry
- **Containers**
  - Isolated application platform
  - Contains everything needed to run your application
  - Based on one or more images



# Registry and Repository



Registry (example:Docker Hub)

Repo

ubuntu®

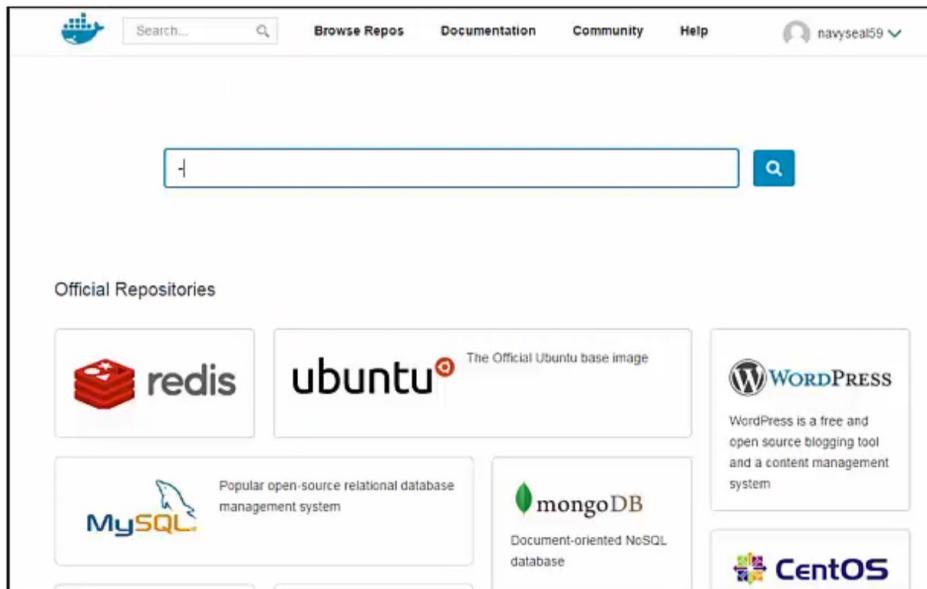
Repo

Repo



# Docker Hub

**Docker Hub** is the public registry that contains a large number of images available for your use





# Docker Orchestration

- Three tools for orchestrating distributed applications with Docker
- Docker Machine
  - Tool that provisions Docker hosts and installs the Docker Engine on them
- Docker Swarm
  - Tool that clusters many Engines and schedules containers
- Docker Compose
  - Tool to create and manage multi-container applications
- Covered in Docker Operations course



# Benefits of Docker

- Separation of concerns
  - Developers focus on building their apps
  - System admins focus on deployment
- Fast development cycle
- Application portability
  - Build in one environment, ship to another
- Scalability
  - Easily spin up new containers if needed
- Run more apps on one host machine

# Intro to Images

- Lots of Images available for use
- Images reside in various Repositories



The screenshot shows the Docker Hub interface. At the top, there is a search bar with the text "tomcat" and a magnifying glass icon. To the right of the search bar are links for "Browse Repos", "Documentation", "Community", and "Help". A user profile icon for "navyseal59" is also present. Below the search bar, there are three categories: "Repositories (498)", "Users (0)", and "Organizations (0)". On the right side, there is a search filter section with dropdowns for "Show: All", "Sort by: Relevance", and "Results: S". A search result card for "tomcat" is displayed, featuring the Docker logo, the repository name "tomcat" with a blue star icon, and a brief description: "Apache Tomcat is an open source implementation of the Java Servlet and JavaServer Pages technologies". The card also shows the last update time "7 days ago" and two engagement metrics: 93 stars and 11531 forks.

# Docker Hub



Screenshot of Docker Hub Content page for user 'trainingteam'.

The page shows:

- Recently Updated Repositories:**
  - helloworldauto (1 week ago)
  - myprivateapp (1 week ago)
  - helloworld (2 weeks ago)
  - testexample (2 weeks ago)
- Contributed Repositories:** No contributions... yet!
- Starred Repositories:** Browse repositories in the Registry
- Activity Feed:**
  - trainingteam pushed to the repository trainingteam/testexample (2 weeks ago)
  - trainingteam pushed to the repository trainingteam/testexample (2 weeks ago)
  - + trainingteam created the repository trainingteam/testexample (2 weeks ago)

# Docker Hub



Docker Hub Registry - Re x https://registry.hub.docker.com

Search... Browse Repos Documentation Community Help trainingteam

Search the Registry

Official Repositories

- redis**
- ubuntu** The Official Ubuntu base image
- MySQL** Popular open-source relational database management system
- mongoDB** Document-oriented NoSQL database
- CentOS** Official CentOS base image
- NGINX** High performance reverse proxy server
- node** Node.js is a platform for scalable server-side and networking applications

Top Contributors

- clue

Popular Repositories

- ubuntu

# Create a Docker Hub account



1. Go to <https://hub.docker.com/account/signup/> and signup for an account if you do not already have one.  
No credit card details are needed
2. Find your confirmation email and activate your account
3. Browse some of the repositories
4. Search for some images of your favourite dev tools, languages, servers etc...
  - a) (examples: Java, Perl, Maven, Tomcat, NGINX, Apache)

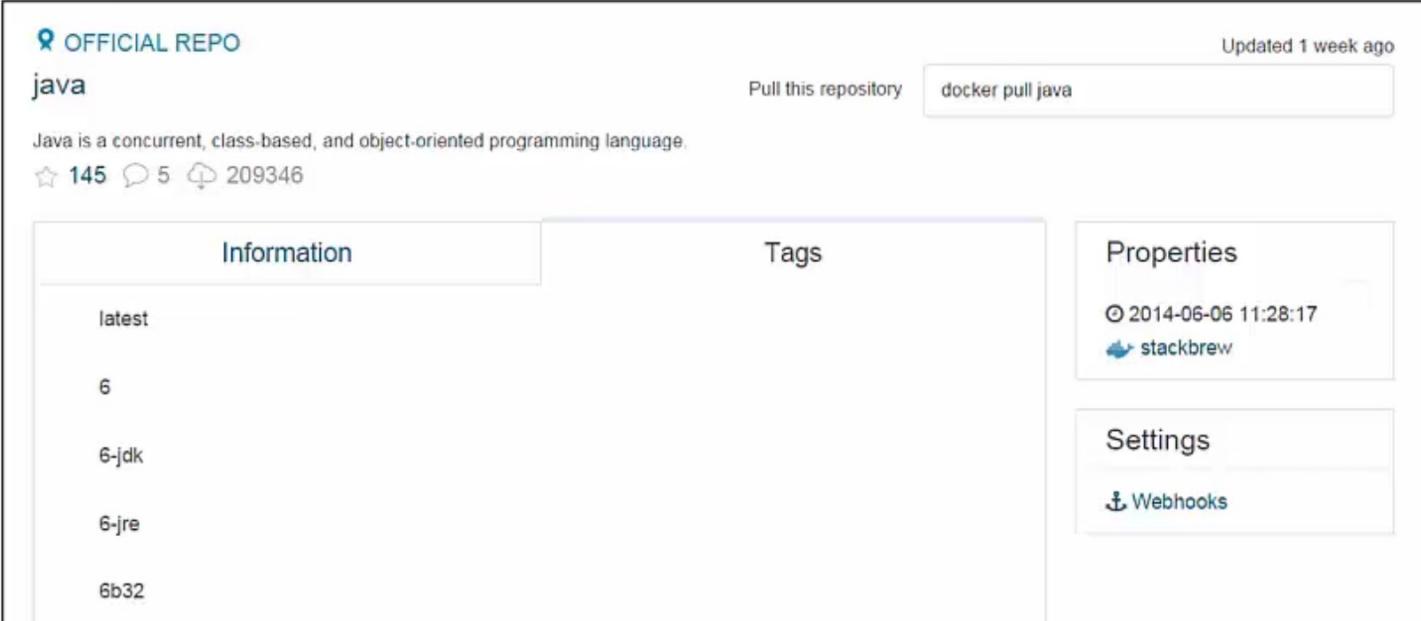
# Display local images

- Run `docker images`
- When creating a container NGINX will attempt to use a local image first
- If no local image is found, the Docker daemon will look in Docker Hub unless another registry is specified

```
johnnytu@dockertraining:~$ sudo docker images
[sudo] password for johnnytu:
REPOSITORY          TAG        IMAGE ID      CREATED       VIRTUAL SIZE
ubuntu              14.04      2103b00b3fdf   8 days ago    188.3 MB
ubuntu              14.04.2     2103b00b3fdf   8 days ago    188.3 MB
ubuntu              latest      2103b00b3fdf   8 days ago    188.3 MB
ubuntu              trusty      2103b00b3fdf   8 days ago    188.3 MB
ubuntu              trusty-20150228.11  2103b00b3fdf   8 days ago    188.3 MB
java                openjdk-7-jdk  fa5c5774f090   10 days ago   584.9 MB
nginx              latest      b17a02c942e1   10 days ago   93.43 MB
centos              7          88f9454e60dd   2 weeks ago   210 MB
```

# Images Tag

- Images are specified by **repository:tag**
- The same image may have multiple tags
- The default tag is `latest`
- Look up the repository on Docker Hub to see what tags are available



The screenshot shows the Docker Hub page for the 'java' repository. At the top, it says 'OFFICIAL REPO' and 'java'. It has a 'Pull this repository' button with the command 'docker pull java' and a note 'Updated 1 week ago'. Below this, there's a brief description: 'Java is a concurrent, class-based, and object-oriented programming language.' followed by statistics: 145 stars, 5 comments, and 209346 forks. On the left, there's a sidebar with links like 'Information', 'Tags', 'Properties', and 'Settings'. The 'Information' tab is active, showing the repository was created on 2014-06-06 at 11:28:17 by 'stackbrew'. The 'Tags' tab lists several tags: 'latest', '6', '6-jdk', '6-jre', and '6b32'. The 'Properties' tab shows the creation date and author. The 'Settings' tab has a 'Webhooks' link.



# Create a Container

- Use **docker run** command

- Syntax

```
sudo docker run [options] [image] [command] [args]
```

- Image is specified with repository:tag

## Examples

```
docker run ubuntu:14.04 echo "Hello World"
```

```
docker run ubuntu ps ax
```





# Run a simple container

1. On your terminal type

```
docker run ubuntu:14.04 echo "hello world"
```

2. Observe the output

3. Then type

```
docker run ubuntu:14.04 ps ax
```

4. Observe the output

5. Notice the much faster execution time compared to the first container that was run. This is due to the fact that Docker now has the Ubuntu 14.04 image locally and thus does not need to download the image



```
johnnytu@docker-demo:~$  
johnnytu@docker-demo:~$  
johnnytu@docker-demo:~$  
johnnytu@docker-demo:~$  
johnnytu@docker-demo:~$ docker images  
REPOSITORY          TAG      IMAGE ID      CREATED     VIRTUAL SIZE  
hello-world         latest    e45a5af57b00   3 months ago  910 B  
johnnytu@docker-demo:~$ docker run ubuntu:14.04 echo "hello world"  
Unable to find image 'ubuntu:14.04' locally  
f3c84ac3a053: Pull complete  
f3c84ac3a053: Download complete  
a1a958a24818: Pulling fs layer  
d0955f21bf24: Download complete  
d0955f21bf24: Pulling dependent layers  
511136ea3c5a: Download complete
```



# Container with Terminal

- Use `-i` and `-t` flags with `docker run`
- The `-i` flag tells docker to connect to STDIN on the container
- The `-t` flag specifies to get a pseudo-terminal
- **Note:** You need to run a terminal process as your command (e.g. `/bin/bash`)

## Example

```
docker run -i -t ubuntu:latest /bin/bash
```



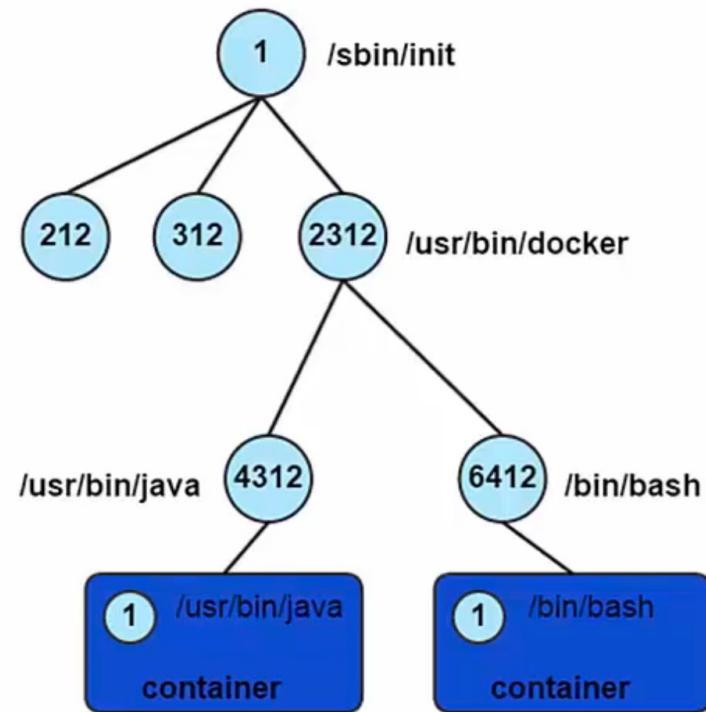
# Run a Container and get Terminal access



1. Create a container using the ubuntu 14.04 image and connect to STDIN and a terminal  
`sudo docker run -i -t ubuntu:14.04 /bin/bash`
2. In your container, create a new user using your first and last name as the username  
`adduser username`
3. Add the user to the sudo group  
`adduser username sudo`
4. Exit the container  
`exit`
5. Notice how the container shut down
6. Once again run:  
`sudo docker run -i -t ubuntu:14.04 /bin/bash`
7. Try and find your user
8. Notice that it does not exist

# Container process

- A container only runs as long as the process from your specified docker run command is running
- Your command's process is always PID 1 inside the container





# Container ID

- Containers can be specified using their ID or name
- Long ID and short ID
- Short ID and name can be obtained using `docker ps` command to list containers
- Long ID obtained by inspecting a container





# Container ID

```
johnny@docker-demo:~$  
johnnytu@docker-demo:~$  
johnnytu@docker-demo:~$  
johnnytu@docker-demo:~$ docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS          NAMES  
22e50a079686        ubuntu:14.04       "bash"              30 minutes ago   Up 30 minutes     loving_f
```

```
johnnytu@docker-demo:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
22e50a079686	ubuntu:14.04	"bash"	30 minutes ago	Up 30 minutes		loving_f
loving_feynman	ubuntu:14.04	"echo 'hello world'"	39 minutes ago	Exited (0) 39 minutes ago		
b2cee57aaaf92	ubuntu:14.04	"echo 'hello world'"	39 minutes ago	Exited (0) 39 minutes ago		
compassionate_thompson	ubuntu:14.04	"/bin/bash"	About an hour ago	Exited (0) About an hour ago		
901138dca328	ubuntu:14.04	"/bin/bash"	About an hour ago	Exited (0) About an hour ago		
jolly_mcclintock	ubuntu:14.04	"/bin/bash"	About an hour ago	Exited (0) About an hour ago		
7a0de6083648	ubuntu:14.04	"/bin/bash"	About an hour ago	Exited (0) About an hour ago		
ecstatic_thompson	ubuntu:14.04	"ps ax"	About an hour ago	Exited (0) About an hour ago		
84efc508ab90	ubuntu:14.04	"ps ax"	About an hour ago	Exited (0) About an hour ago		
gloomy_pasteur	ubuntu:14.04	"echo 'hello world'"	About an hour ago	Exited (0) About an hour ago		
731fce6fa185	ubuntu:14.04	"echo 'hello world'"	About an hour ago	Exited (0) About an hour ago		
adoring_archimedes	ubuntu:14.04	"echo 'hello world'"	About an hour ago	Exited (0) About an hour ago		
fabf3617079f	hello-world:latest	"/hello"	24 hours ago	Exited (0) 24 hours ago		
jolly_hoover	hello-world:latest	"/hello"	24 hours ago	Exited (0) 24 hours ago		
dcc3d3c46d2a	hello-world:latest	"/hello"	24 hours ago	Exited (0) 24 hours ago		
lonely_davinci						
johnnytu@docker-demo:~\$						





# Running in Detached Mode

- Also known as running in the background or as a daemon
- Use `-d` flag
- To observe output use `docker logs [container id]`

Create a centos container and run the ping command to ping the container itself 50 times

```
docker run -d centos:7 ping 127.0.0.1 -c 50
```





# Questions

Bộ môn MMT&VT - Khoa CNTT - ĐH KHTN Tp. HCM