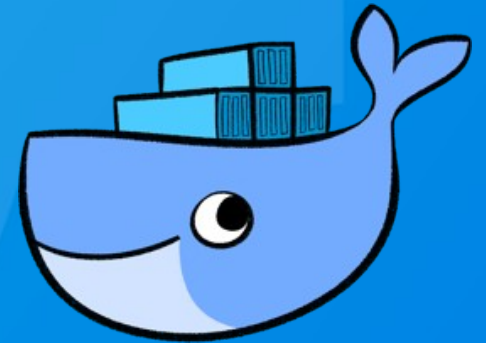


**Welcome to
Docker !**



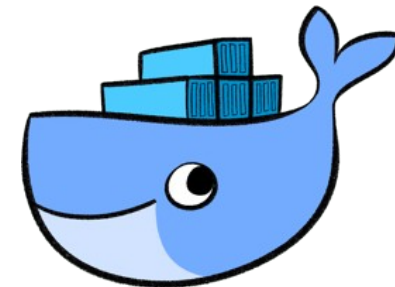
Alexandra BOMANE, PhD Student

CRCM – Inserm U 1068

Dr Pedro BALLESTER's team

Contact : alexandra.bomane@inserm.fr


Docker : History




Created by the French developer **Solomon Hykes**

Written in **Go**, Apache licence 2.0 

Initial release : **March 13, 2013**

2015 : 25,600+ Github stars, **6,800+** forks & ~ **1,100** contributors 

Officially bundled with **Ubuntu 14.04** and **RHEL 7**  

First **DockerCon** and **Docker 1.0** released June 9, 2014

Partnership with **VMWare**, **Microsoft**  

Amazon EC2 Container Service 

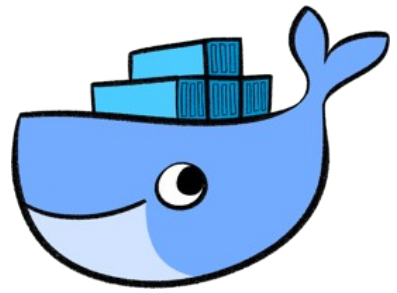
Docker support for Google's Compute Engine  Google Cloud Platform

March 2017 : Docker 17.03.0-ce

Docker



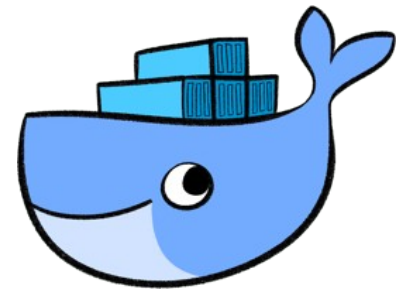
What is Docker ?



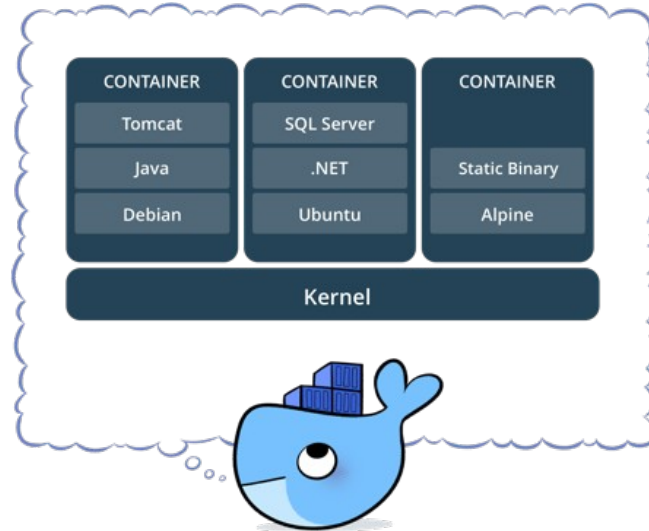
Docker is an open-source project that automates the **deployment of applications** inside software **containers**

Wikipedia

What is a container ?



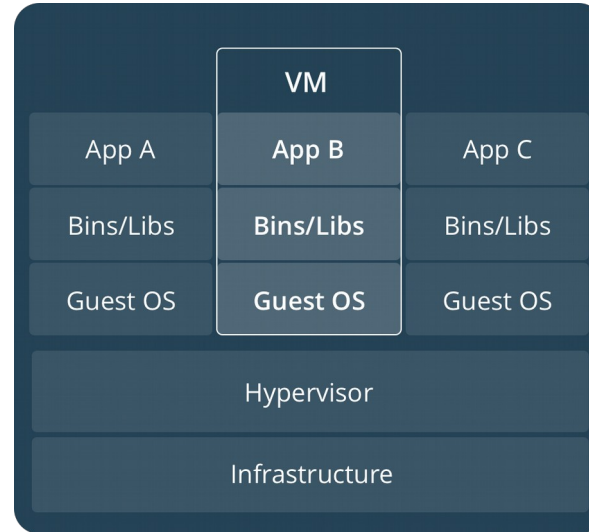
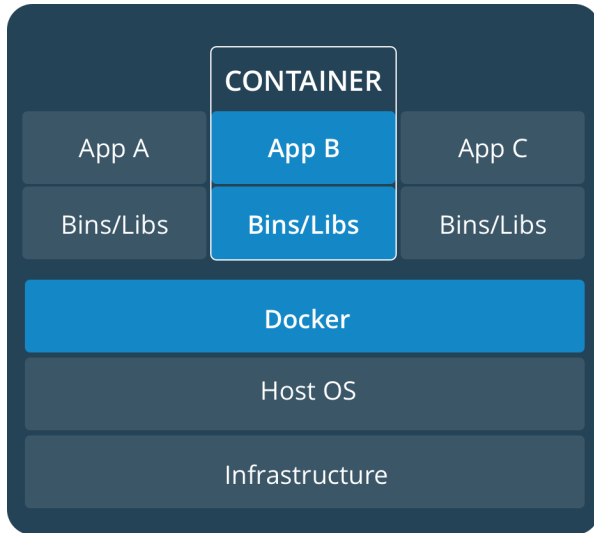
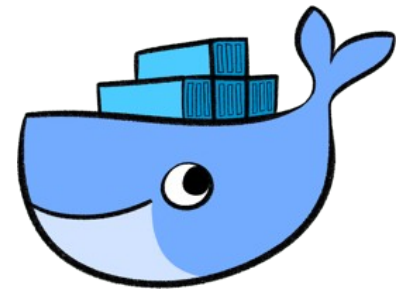
Operating system–level virtualization (container) is a **server virtualization** method where the kernel of an operating system allows for multiple isolated user space instances, instead of just one



Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space.

Wikipedia

Comparing Containers and Virtual Machines (VMs)



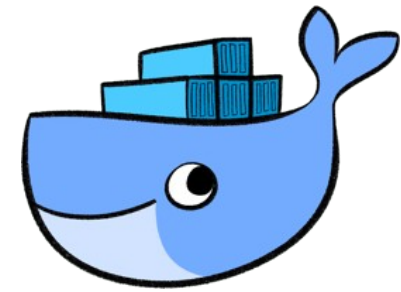
(Docker) Container

advantages :

- Lighter
- Portable
- Faster
- More efficient

- A Docker container **contains only it needs to run** : Minimal Base OS, Libraries and frameworks, Application code
- **Docker image** = particular state of a container (backup)
- Docker image is **immutable**

Docker Hub : Central repository for images



Search the Registry

Official Repositories

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

Top Contributors

- clue ~Aachen, Germany 158
- cpuguy83 Florida 153
- radial Los Angeles 126
- pinterb Wisconsin, USA 116
- guilhem Paris 78
- joaodubas São Paulo, Brazil 75

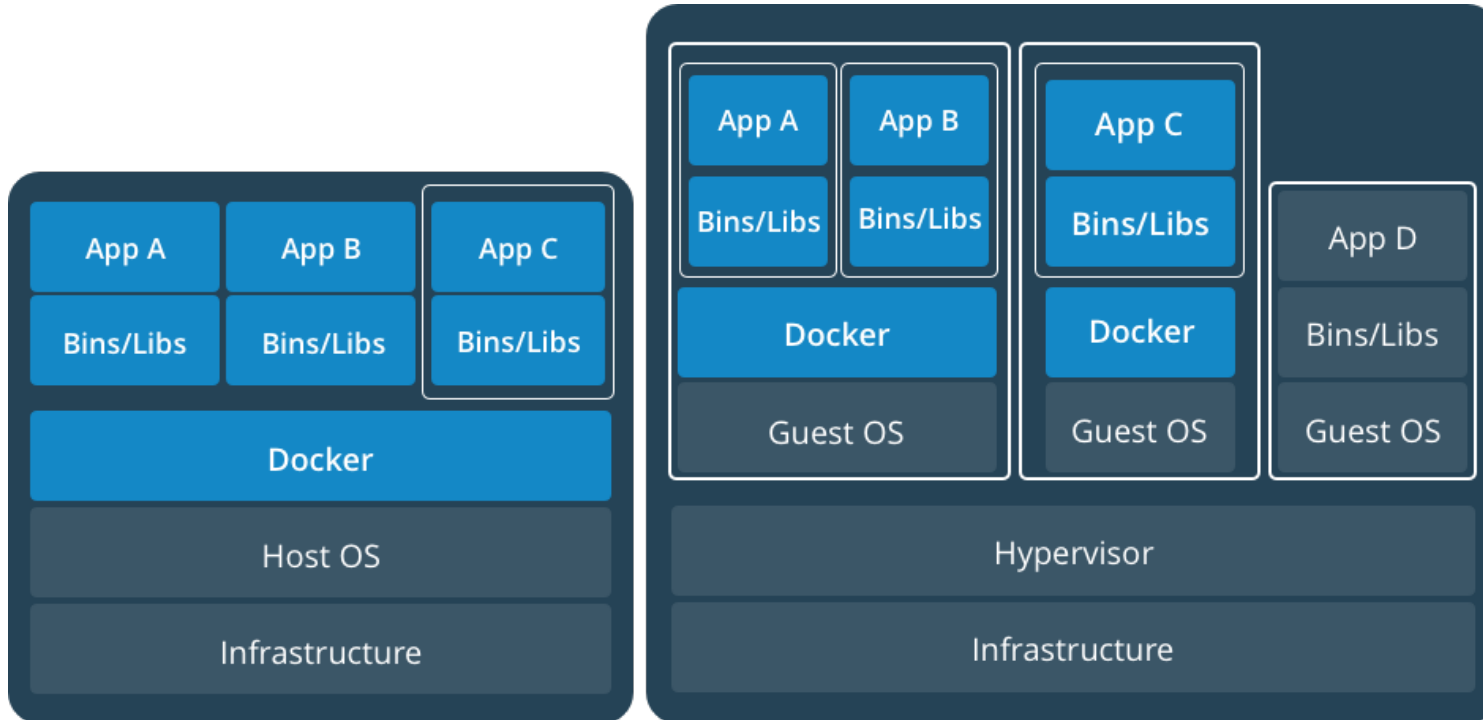
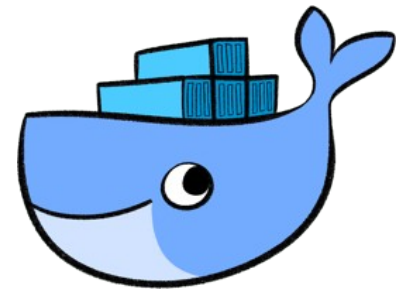
Popular Repositories

- ubuntu Official Ubuntu base image 330
- stackbrew 181
- centos The official build of CentOS. 138
- stackbrew 138
- phusion/baseimage A special image that is configured for correct use within Docker containers. It is Ubuntu, plus some modifications fo... 138
- phusion

- More than 75,000 applications in Docker Hub registry
- Storing public images on Docker Hub is **free**.
- Only requires a **Docker Hub account**.
- Organisation accounts can also be created
- Business model of Docker inc. is to sell private image storage
- A **private repository** can be created (not tested)
- Pull an image/repository:

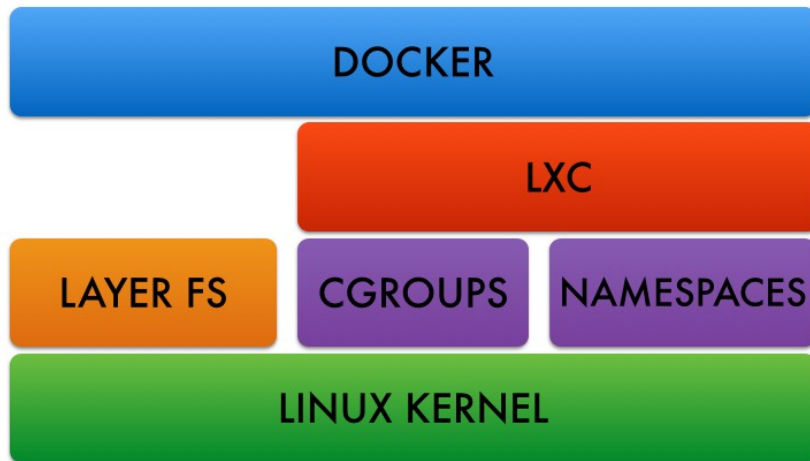
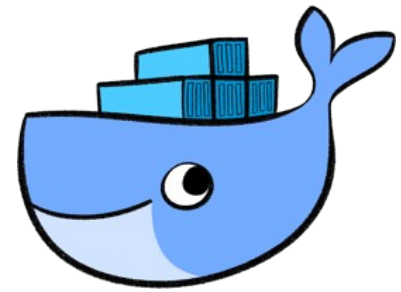
\$ docker pull image_name

Containers and Virtual Machines Together



→ Great deal of **flexibility** in deploying and managing apps

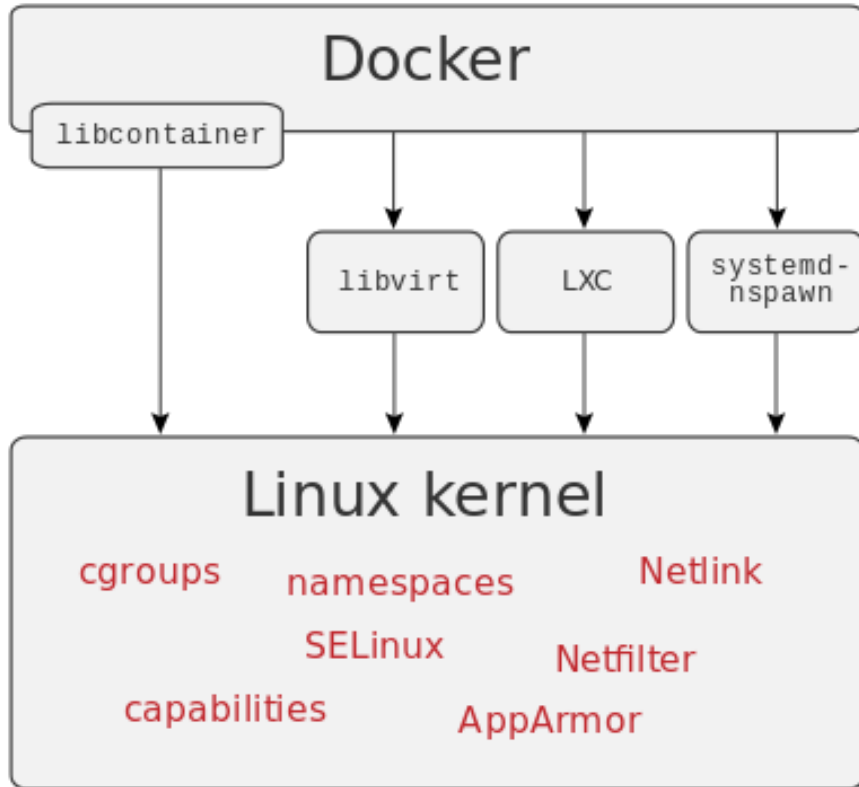
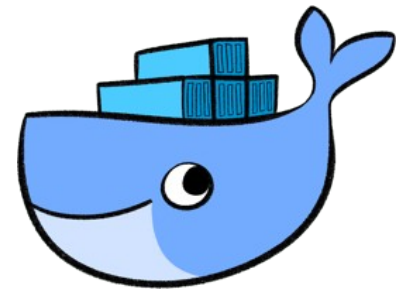
Docker is a layer upon containers



Source : stevenborrelli

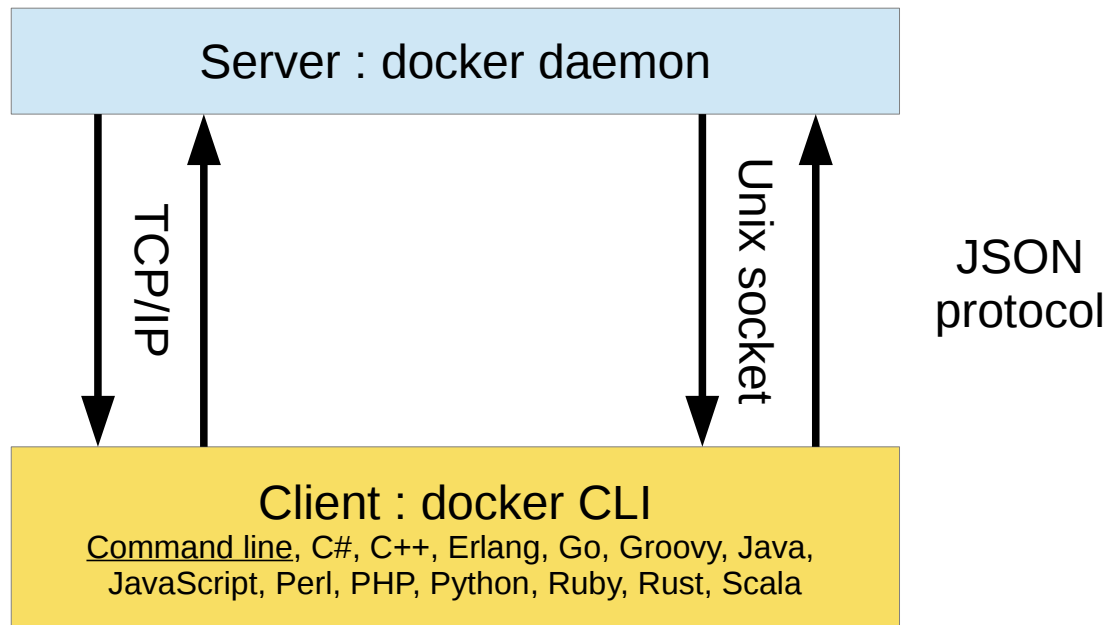
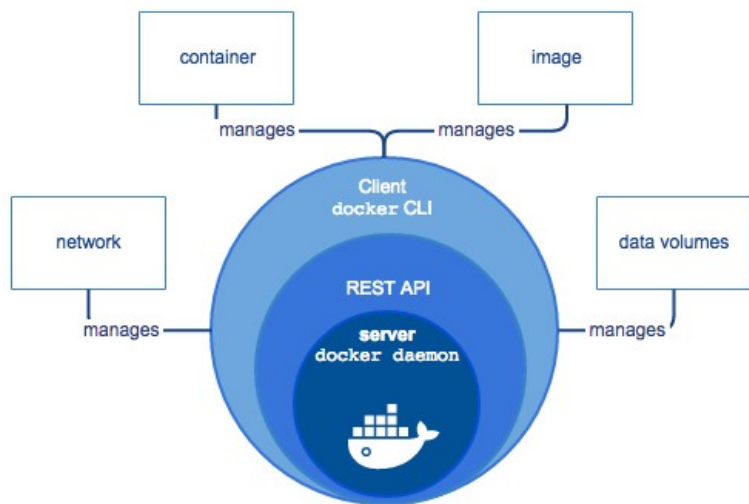
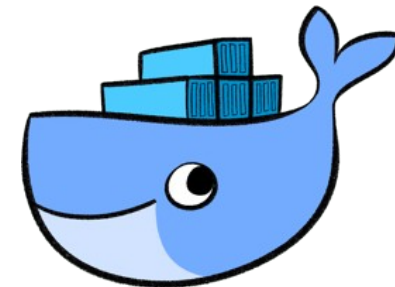
- **Layered filesystem** are used to save space (AUFS/LVM)
- **Copy-on-write filesystem** is used to track changes
- A container has its own private **IP address**

Docker is now THE container abstraction layer



- Docker is *de facto* now the standard layer for containers
- Since 0.9, libcontainer library is the default Docker execution environment instead of LXC
- Docker will be integrated to Windows Server 10

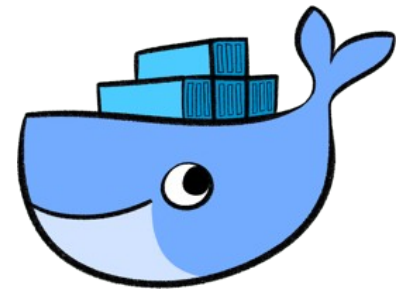
Docker Engine is a client-server application



By **default Unix sockets** are used

TCP/IP exclusively used when the server is on a **different host**

Docker : Installation for Ubuntu



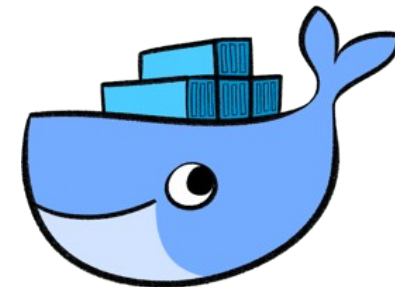
- Ubuntu versions : Yakkety 16.10, Xenial 16.04 (LTS), Trusty 14.04 (LTS) :

\$ sudo apt-get update

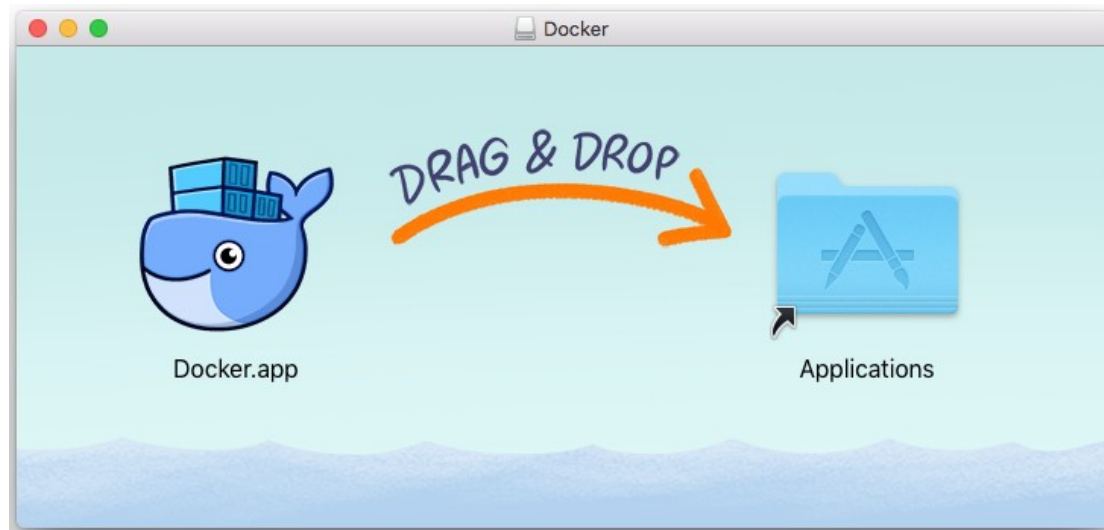
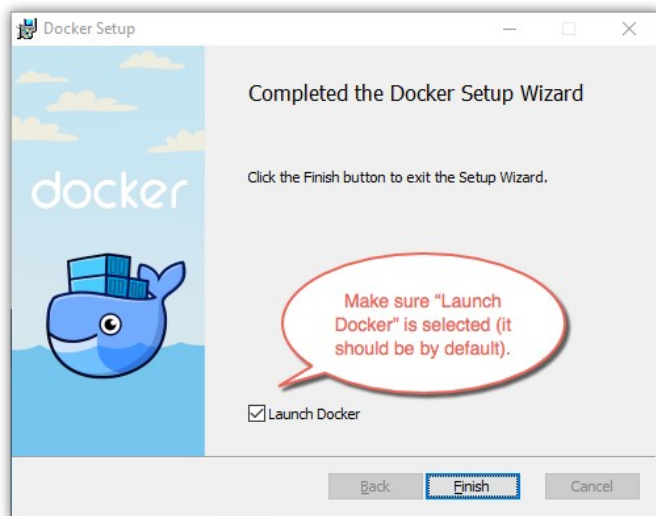
\$ sudo apt-get install docker-ce

- **Root/sudo status required** to use Docker → users can be added to the “**docker**” group in `/etc/groups`
- **64 bits Linux kernel required**

Docker : Installation for Windows & OS X



The Docker for Windows/Mac install package includes everything you need to run Docker on a Windows/Mac system



Docker accessible from Terminal

Portainer : Simple UI for Docker

The screenshot shows the Portainer web interface. On the left is a dark blue sidebar with navigation options: Home, App Templates, Containers, Images, Networks, Volumes, Events, Docker, and Portainer Settings (Password, Users, Endpoints). The main content area is titled 'Home Dashboard' and shows the following information:

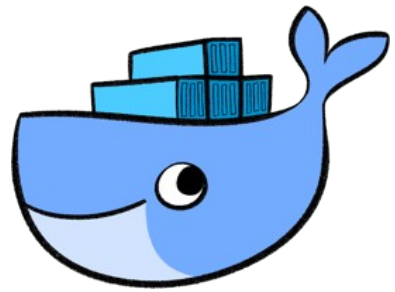
- Node info:** Name: thunderstruck, Docker version: 17.03.0-ce, CPU: 8, Memory: 8.3 GB.
- Containers:** 10 total (8 running, 2 stopped).
- Images:** 21 total (2.6 GB).
- Volumes:** 21 total (aufs driver).
- Networks:** 4 total.

The top right corner shows the user 'admin' with a 'log out' link. The bottom left corner indicates 'Portainer v1.12.1'.

An open-source lightweight management UI which allows you to **easily manage your Docker host**

Available on Linux, Windows & OS X

Docker : Launching a container



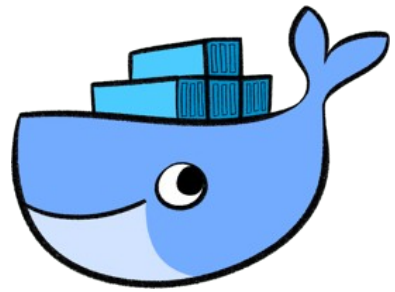
Launching bash in interactive mode on an Ubuntu 14.04 image

```
$ docker run -t -i ubuntu:14.04 bash
```

Mounting host /tmp in /root of the container

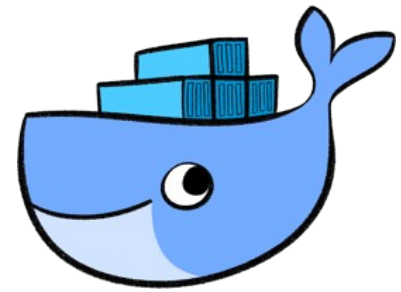
```
$ docker run -t -i -v /tmp:/root ubuntu:14.04 bash
```

Docker : Creating a image from an existing container



```
$ docker run -t -i ubuntu:14.04 bash  
root@533198e47882:/# echo "Hello world !" > /hello.txt  
root@533198e47882:/# exit  
Exit  
  
$ docker diff 533198e47882  
A /root/.bash_history  
C /root  
A /hello.txt  
  
$ docker commit 533198e47882 genomicpariscentre/bidon:0.01  
C75397b350ae1473eaf8f395b786b0fe3740fb6a7cfbb3c2b2d9b781
```

Docker : Pushing images to repository



```
$ docker push genomicpariscentre/bidon
```

```
The push refers to a repository [genomicpariscentre/bidon]
```

```
(len: 1)
```

```
Sending image list
```

```
Pushing repository genomicpariscentre/bidon (1 tags)
```

```
511136ea3c5a: Image already pushed, skipping
```

```
35f6dd4dd141: Image already pushed, skipping
```

```
7baf0ef6f14a: Image already pushed, skipping
```

```
e497c7c1bfbb: Image already pushed, skipping
```

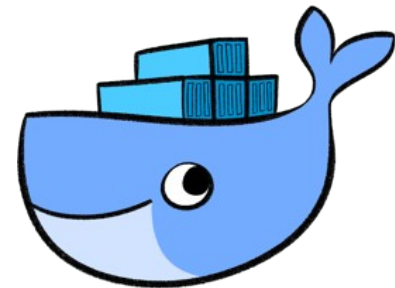
```
5cf8fd909c6c: Image already pushed, skipping
```

```
c75397b350ae: Image successfully pushed
```

```
Pushing tag for rev [c75397b350ae] on {https://registry-  
1.docker.io/v1/repositories/genomicpariscentre/bidon/tags/0.01}
```

Use the `--rm` option of Docker to avoid saving the changes of the file system of the container (use lot of disk space)

Docker : Creating a image using a Dockerfile



1) Create a “Dockerfile” file in a new directory :

```
# Set the base image to Ubuntu
```

```
FROM ubuntu:14.04
```

```
# File Author / Maintainer
```

```
MAINTAINER Sophie Lemoine, slemoine@biologie.ens.fr
```

```
# Update the repository sources list and install essential libraries
```

```
RUN apt-get update && apt-get install --yes build-essential
```

```
# Install Python
```

```
RUN apt-get install --yes python-pip libpython2.7-dev
```

```
# Install Cutadapt
```

```
RUN pip install 'cutadapt==1.8.3'
```

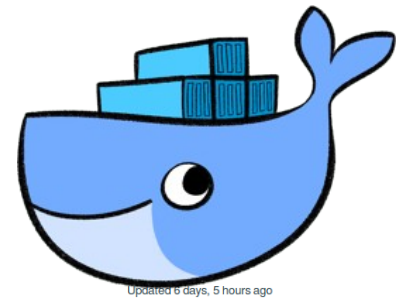
```
# Clean
```

```
RUN apt-get clean ; apt-get remove --yes --purge build-essential
```

2) Build the image :

```
$ docker build -t \
genomicpariscentre/cutadapt:1.8.3 .
```

Docker : Automated Builds on Docker Hub using a GitHub/BitBucket repository



PUBLIC [GenomicParisCentre / dockerfiles](#) Unwatch 1

Description
Short description of this repository

Website
Website for this repository (optional) Save or cancel

1 commit 1 branch 0 releases 1 contributor

branch: master dockerfiles / +

first commit

Author	Commit	Time
jourden	latest commit f678e53be	7 days ago
bcl2fastq	first commit	7 days ago
casava	first commit	7 days ago
deseq	first commit	7 days ago
eoulsan	first commit	7 days ago
fastqc	first commit	7 days ago
htseq	first commit	7 days ago
macs	first commit	7 days ago
oib	first commit	7 days ago
README.md	first commit	7 days ago

README.md

Dockerfiles

This repository contains Dockerfile used to make Docker images used at Genomic Paris Centre.

AUTOMATED BUILD REPOSITORY
[genomicpariscentre / htseq](#) Pull this repository `docker pull genomicpariscentre/htseq` Updated 6 days, 5 hours ago

HTSeq is a Python package that provides infrastructure to process data from high-throughput sequencing assays.
0 stars 0 comments 0 forks

Information **Build Details** **Tags** Start a Build

Build Details Edit Build Details

Type	Name	Dockerfile Location	Tag Name
Branch	master	/htseq/htseq-0.6.1p1	0.6.1p1
Branch	master	/htseq/htseq-0.5.4p1	0.5.4p1
Branch	master	/htseq/htseq-0.5.3p9	0.5.3p9

Builds History

build id	Status	Created Date	Last Updated
bevpw999seiw6wq3f6tuya	Finished	2014-06-24 18:04:14	2014-06-24 18:11:27
bhpkognqntjwz8oygj6nwom	Finished	2014-06-24 18:04:14	2014-06-24 18:08:56
bejzg7q8z3oabtswxpwuyyb	Finished	2014-06-24 18:04:14	2014-06-24 18:07:24
brv2yxa99xtbu6nnyedldg	Finished	2014-06-24 17:49:57	2014-06-24 17:53:13
b4c8kc3hzuucujhr7xhmuk	Finished	2014-06-24 17:49:57	2014-06-24 17:53:36
baobzdurxgsqjblhu5arzzw	Finished	2014-06-24 17:49:57	2014-06-24 17:53:29
bhoectckpiuhnqqtcjfyppg	Finished	2014-06-24 17:49:57	2014-06-24 17:53:00

Build Details
Links
[Source Project Page](#)
[Source Repository](#)

Files
[Build Bundle](#)
[Dockerfile](#)

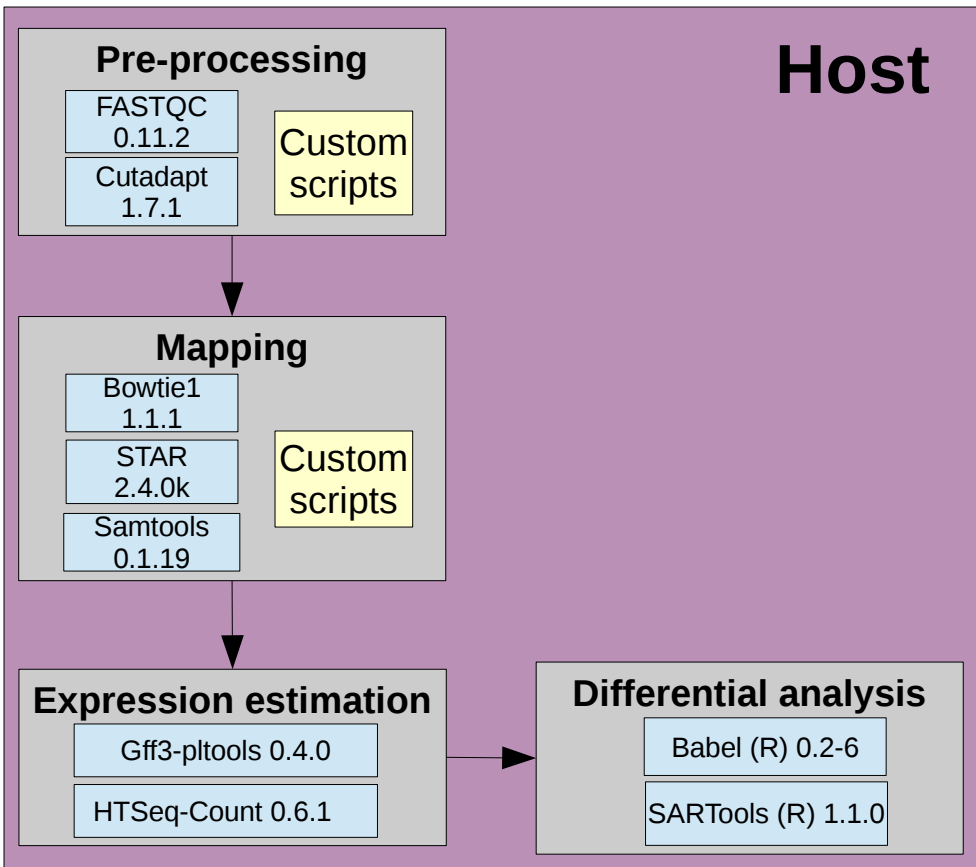
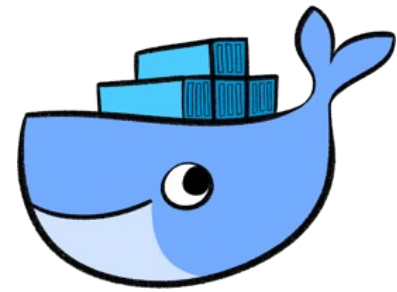
Settings
[Description](#)
[Build Details](#)
[Webhooks](#)
[Collaborators](#)
[Build Triggers](#)
[Repository Links](#)
[Make Private](#)
[Delete repository](#)

Manage
[Source Project Page](#)
[Source Repository](#)

Launch a new build when new pushes occur

“Trusted image” as Dockerfile is available in Docker Hub

Docker : Execution in a pipeline for Ribosome Profiling data analysis



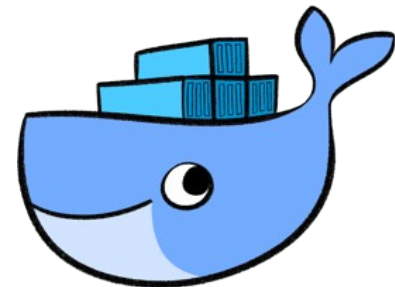
1st case : “classical” launch

- All steps are launched by a command line via a Bash script
- Bioinformatic tools run in Docker containers based on respective Docker images

Pattern of Docker commands in the Bash script :

```
$ docker run --rm -u $(id -u):$(id -g) \  
-v $TMPDIR:/tmp \  
-v $WORKDIR:/home -w /home \  
genomicpariscentre/tool:version CMD
```

Docker : Execution in a pipeline for Ribosome Profiling data analysis



Container

Pre-processing

FASTQC
0.11.2

Cutadapt
1.7.1

Custom
scripts

Mapping

Bowtie1
1.1.1

STAR
2.4.0k

Samtools
0.1.19

Custom
scripts

Expression estimation

Gff3-pltools 0.4.0

HTSeq-Count 0.6.1

Differential analysis

Babel (R) 0.2-6

SARTools (R) 1.1.0

2nd case : “Docker in Docker“ launch

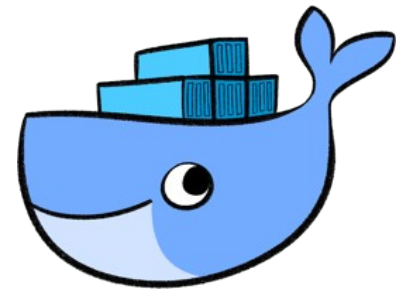
- Bash script launched within a Docker container
- Tip 1 : mounting the **Docker socket** of the host to the container

Docker command to run Docker in Docker :

```
$ docker run --rm --privileged --name ribopro \  
-v /var/run/docker.sock:/var/run/docker.sock \  
-v $TMPDIR:/tmp \  
-v $WORKDIR:/home -w /home ..... \  
genomicpariscentre/tool:version command
```

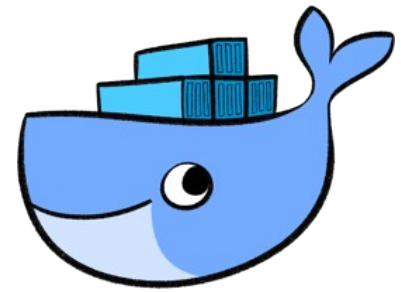
- Tip 2 : add **--volume-from ribopro** in Docker commands

Docker : Use cases in Bioinformatics



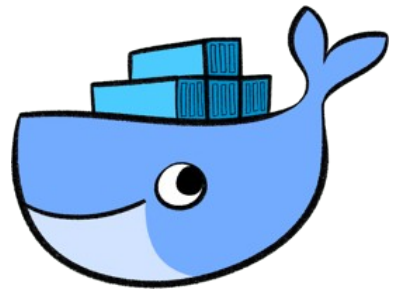
- Using **several versions** of a software on the same computer that usually cannot cohabit
- **Fixing the version** of a software (e.g. bioperl, bioconductor) that changes every often
- Using a **different Linux distribution** than the host distribution (e.g. Centos on Ubuntu)
- Using a software that requires **dependencies** that are **no more available** on the host
- Easy to **deploy/upgrade software** and **faster production deployment**
- Relaunching quickly & easily the analysis described in the article by the reviewers and the scientific community → **Reproducibility**
 - Package all the scripts and the programs used in an article

Docker : Conclusion



- Under **heavy development** (major contributions by RedHat, Google...).
- Will be soon also the standard to deploy cloud application (Amazon EC2 Container Service).
- **Hardware virtualization is not dead**, it still useful for hardware management.
- **Very good solution** to easily package and deploy bioinformatic tools that had complex dependencies (R packages, bioperl...).

Docker : Useful links



Reference documentation :

<http://docs.docker.com/reference/>

Online tutorials :

<http://blog.docker.com/2015/03/docker-tutorial-1-installing-docker/>

<http://blog.docker.com/2015/03/docker-tutorial-2-docker-run-command-basics/>

<http://blog.docker.com/2015/04/docker-tutorial-3-fun-with-volumes/>

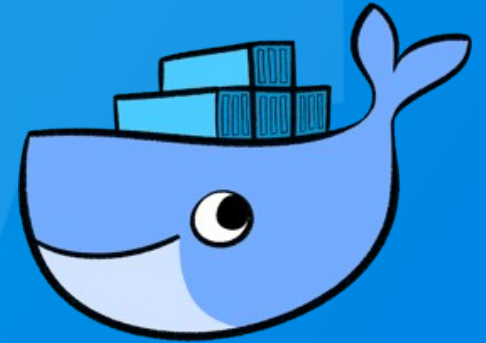
<http://blog.docker.com/2015/04/docker-tutorial-4-more-run-with-a-couple-of-extra-goodies/>

<http://blog.docker.com/2015/04/docker-tutorial-5-basic-networking/>

Dockerfile best practices :

http://docs.docker.com/articles/dockerfile_best-practices/

Thank you !



Contact : alexandra.bomane@inserm.fr