



Plan

Introduction

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

Hands-on

Conteneurs

Applications

Image build

Container deployment

Conclusion

# Docker

**Cécile Cavet**

`ccavet at apc.in2p3.fr`

Centre François Arago (FACe), Laboratoire APC, Université de Paris

<https://gitlab.in2p3.fr/cavet/tp-docker-obs/>

November 20 2019



# Plan

## Plan

### Introduction

### Image creation and management

Docker Hub

GitLab Registry

### Container deployment

### Docker hosts

### Hands-on

Conteneurs

Applications

Image build

Container deployment

### Conclusion

- Introduction
- Image creation and management
- Container deployment
- Docker hosts
- Hands-on
- Conclusion



# Introduction

Plan

**Introduction**

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

Hands-on

Conteneurs

Applications

Image build

Container deployment

Conclusion



- **Container**: isolation of processes/applications.
- **Microservice**: 1 application within its specific environnement.
- The same philosophy as **Virtual Machines (VM)**: isolation, share, reuse... but lighter and faster!



# VM vs container

Plan

## Introduction

### Image creation and management

Docker Hub

GitLab Registry

### Container deployment

### Docker hosts

### Hands-on

Conteneurs

Applications

Image build

Container deployment

### Conclusion

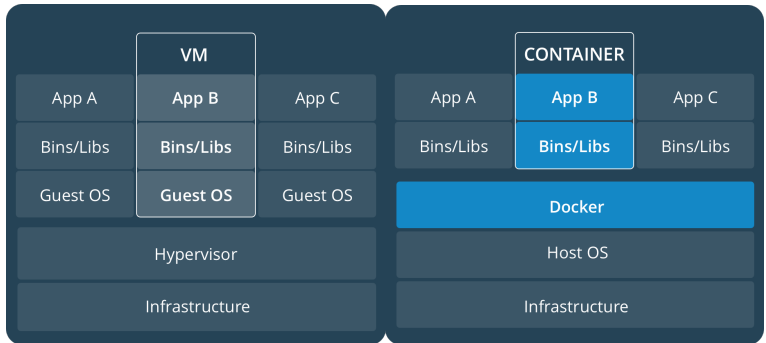


Figure: VM (left) vs Docker (right).



# Docker

Plan

## Introduction

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

## Hands-on

Containers

Applications

Image build

Container deployment

## Conclusion



- The **Docker** technology is written in Go and is based on:
  - LXC (Linux Containers).
  - Union File System (amalgamated FS of layers).
  - cgroups (resource limitation), namespaces (separated environment).



# Docker history

## Plan

### Introduction

#### Image creation and management

Docker Hub

GitLab Registry

#### Container deployment

#### Docker hosts

#### Hands-on

Conteneurs

Applications

Image build

Container deployment

#### Conclusion

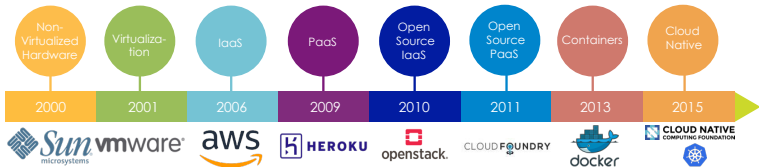


Figure: @CNCF.

- **Docker** used to be **dotCloud** (PaaS cloud).
  - **dotCloud** started in 2008 @Montrouge.
  - The **Docker** solution has been developed since 2013 (docker-0.1).
- **Docker** is now a big project: more than 3,300 contributors.



# Ecosystem

Plan

## Introduction

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

## Hands-on

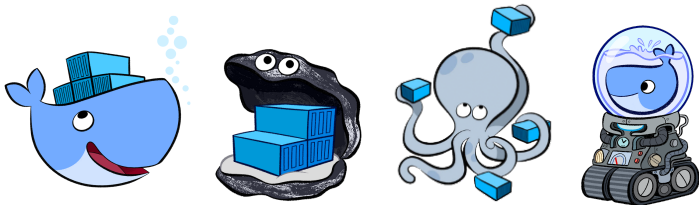
Conteneurs

Applications

Image build

Container deployment

## Conclusion



- ▶ **Engine**: daemon and CLI client.
- ▶ **Registry**: secure private registry.
- ▶ **Docker Hub/Store**: official Docker public registry.
- ▶ **Compose**: multi-container application.
- ▶ **Machine**: local and cloud VMs.



# Container definition

## Plan

### Introduction

#### Image creation and management

Docker Hub

GitLab Registry

#### Container deployment

#### Docker hosts

#### Hands-on

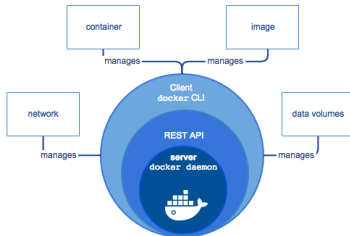
Conteneurs

Applications

Image build

Container deployment

#### Conclusion



- **Image:** a lightweight, stand-alone, executable package including the code, a runtime, libraries, environment variables, and config files.
- **Container:** an image runtime instance, what the image becomes in memory when actually executed.





# User mode

Plan

Introduction

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

Hands-on

Conteneurs

Applications

Image build

Container deployment

Conclusion

## Local machine:

- Linux: native **Engine**
- Non Linux: **Engine** in a light VM (HyperKit VM for macOS, Hyper-V VM for Windows).

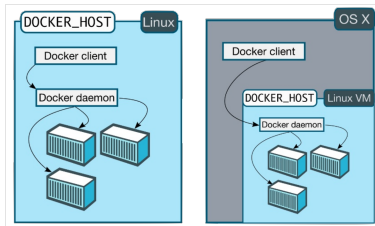


Figure: Linux vs macOS local machines.



# User mode

Plan

Introduction

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

Hands-on

Containers

Applications

Image build

Container deployment

Conclusion

## IaaS cloud:

- Linux VM + **Engine**.
- **Machine** + Linux VM.
- **OpenStack Magnum** plugging: container orchestrator + Linux VM.



# MAGNUM

*an OpenStack Community Project*



# Docker installation

Plan

**Introduction**

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

**Hands-on**

Conteneurs

Applications

Image build

Container deployment

Conclusion

## Version:

- Last version of Community Edition (CE): 19.03.5-ce
- Docker Engine (client/daemon) (v19.03.5-ce).
- Docker Compose (v1.24.1).
- Docker Machine (v0.16.2).

## Linux:

- Package manager (yum|apt).
- Tested on CentOS 7.6.1810.



# Docker installation

Plan

Introduction

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

Hands-on

Containers

Applications

Image build

Container deployment

Conclusion

## MacOS:

- Requirements: Mac hardware 2010+, OS El Capitan+ (10.11), 4GB of RAM.
- Docker Desktop: Engine, Compose, Machine and Kubernetes.

## Windows (not tested):

- Requirements: Hyper-V.
- Docker Desktop.



# Container life cycle

Plan

## Introduction

Image creation and management

Docker Hub

GitLab Registry

Container deployment

Docker hosts

## Hands-on

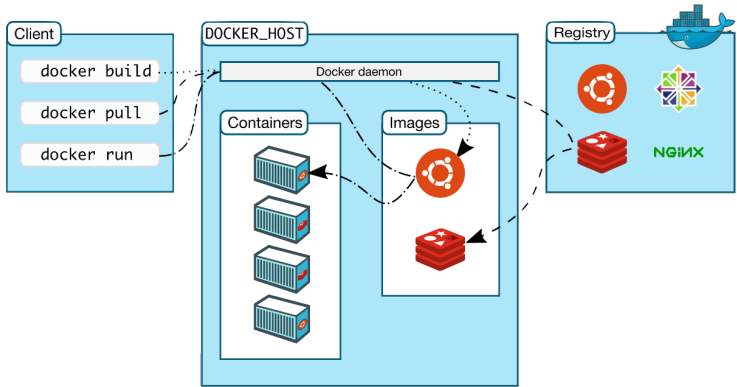
Conteneurs

Applications

Image build

Container deployment

## Conclusion





# Image creation process

Plan

Introduction

**Image creation and management**

Docker Hub

GitLab Registry

Container deployment

Docker hosts

Hands-on

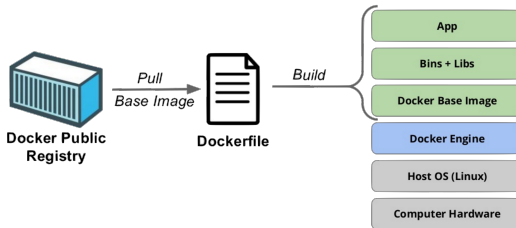
Containers

Applications

Image build

Container deployment

Conclusion



- ▶ **Docker Registry**: a marketplace for sharing images of various OS and applications.
- ▶ **Dockerfile**: a kind of shell script with specific instructions (RUN...).



# Image management

Plan

Introduction

**Image creation  
and management**

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

Hands-on

Conteneurs

Applications

Image build

Container deployment

Conclusion



- ▶ **Docker Hub**: official Docker public registry.
- ▶ **Docker Registry**: Docker image allowing to provide a Docker registry.
- ▶ **GitLab Registry**: Docker private registry provided by GitLab.



# Docker Hub

Plan

Introduction

Image creation  
and management

**Docker Hub**

GitLab Registry

Container  
deployment

Docker hosts

Hands-on

Conteneurs

Applications

Image build

Container deployment

Conclusion

The screenshot shows the Docker Hub interface for the 'python' official image. At the top, there's a blue navigation bar with the Docker Hub logo, a search bar containing 'python', and links for 'Explore', 'Sign In', 'Pricing', and a 'Get Started' button. Below the navigation bar, the 'python' image is featured with its logo and a star icon. It is labeled as 'Docker Official Images'. A description states: 'Python is an interpreted, interactive, object-oriented, open-source programming language.' Below this, it shows '10M+' downloads. A horizontal filter bar includes categories like 'Container', 'Linux', 'Windows', 'IBM Z', 'PowerPC 64 LE', 'ARM', '386', 'ARM 64', 'x86-64', and 'Programming Languages'. Under 'Container', the 'Official image' is selected. On the right side, a dropdown menu shows 'Linux - x86-64 ( latest )'. Below this, a text prompt says 'Copy and paste to pull this image', followed by a dark button with the command 'docker pull python' and a copy icon. At the bottom right, there is a link to 'View Available Tags'.

Figure: Python official images on Docker Hub.





# GitLab Registry

Plan

Introduction

Image creation  
and management

Docker Hub

**GitLab Registry**

Container  
deployment

Docker hosts

Hands-on

Conteneurs

Applications

Image build

Container deployment

Conclusion

## Registry linked to a project:

The screenshot shows the GitLab interface for a project named 'tp-docker-obs'. On the left is a sidebar with navigation links: Project, Repository, Issues (0), Merge Requests (0), CI / CD, Operations, Packages, and Container Registry (highlighted). The main content area is titled 'Container Registry' and includes a sub-header 'With the Docker Container Registry integrated into GitLab, every project can have its own space to store its Docker images. [More Information](#)'. Below this is a link to '^ cavet/tp-docker-obs' with a repository icon. A table lists the registry contents:

<input type="checkbox"/>	Tag	Tag ID	Size	Last Updated	<input type="checkbox"/>
<input type="checkbox"/>	master	c147ec6d7	420.09 MiB	2 hours ago	

Figure: GitLab Registry of the tutorial.



# Container deployment

Plan

Introduction

Image creation  
and management

Docker Hub

GitLab Registry

**Container  
deployment**

Docker hosts

Hands-on

Conteneurs

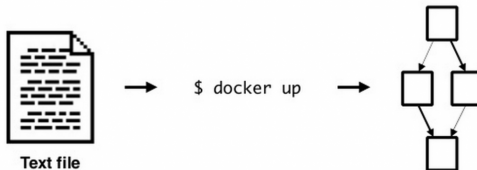
Applications

Image build

Container deployment

Conclusion

**Docker Compose:**  
**Get an app running in one command.**



- **Compose file:** a YAML file allowing to automatize the building of a multi-container application.



# Docker hosts

Plan

Introduction

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

**Docker hosts**

Hands-on

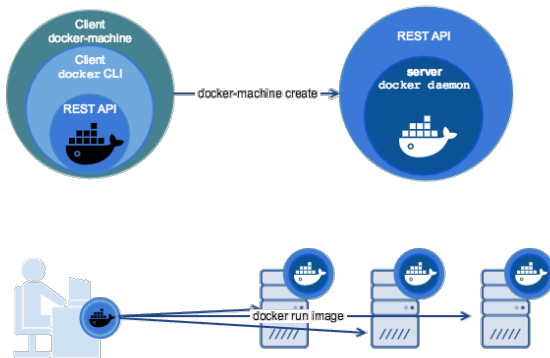
Conteneurs

Applications

Image build

Container deployment

Conclusion



► **Docker Machine:** Docker host provisioning on local and cloud VMs.



# Hands-on plan

## Plan

### Introduction

### Image creation and management

Docker Hub

GitLab Registry

### Container deployment

### Docker hosts

### Hands-on

Conteneurs

Applications

Image build

Container deployment

### Conclusion

#### Installation de Docker

Instructions globales pour le tutoriel

#### Mes premiers conteneurs

Exécution du premier conteneur

Quelques commandes pour gérer le système, nettoyer les conteneurs et les images

Exécution du second conteneur

#### Mes premières applications

Exécution de la première application

Exécution de la deuxième application

#### Création d'images

Création de la première image

Création de la deuxième image

#### Gestion des images

Docker Hub

Docker Registry

GitLab-CI

#### Déploiement de conteneurs avec Docker Compose

Ma première application composée

#### Création de Machines Virtuelles avec Docker Machine



# Conteneurs

Plan

Introduction

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

Hands-on

**Conteneurs**

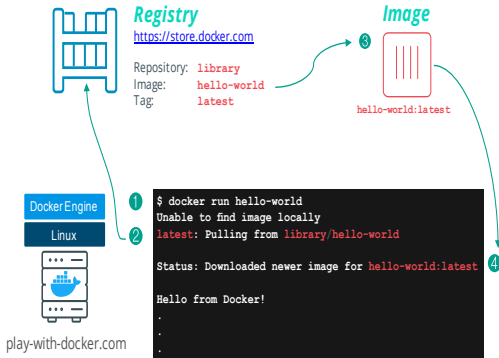
Applications

Image build

Container deployment

Conclusion

## Hello World: What Happened?





# Applications

Plan

Introduction

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

Hands-on

Conteneurs

**Applications**

Image build

Container deployment

Conclusion

## Web service:

- Image: Jupyter server for running Python Notebook.
- Environment: Conda Python 3.x.
- Librairies: Pandas, Matplotlib, Scipy...



jupyter

Logout

Files

Running

Clusters

Select items to perform actions on them.

Upload New ↻

0



Name ↓

Last Modified

work

2 months ago



# Image build

Plan

Introduction

Image creation  
and management

Docker Hub  
GitLab Registry

Container  
deployment

Docker hosts

Hands-on

Conteneurs  
Applications

**Image build**  
Container deployment

Conclusion

## Dockerfile :

- Base image: Python
- Librairies: Python modules.
- Scientific code: LOSC\_Event\_tutorial.

```
Dockerfile.app  x
FROM python:3.7
MAINTAINER Cecile Cavet "ccavet@apc.in2p3.fr"

ENV PYTHONUNBUFFERED 1

WORKDIR /app
COPY . /app/

RUN pip install --no-cache-dir -r requirements.txt

ENTRYPOINT ["python"]
CMD ["LOSC_Event_tutorial.py"]
```



# Container deployment

Plan

Introduction

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

Hands-on

Containers

Applications

Image build

Container deployment

Conclusion

## Docker Compose file:

### ■ Jupyter Notebook image

```
docker-compose.yml x
version: "3"
services:
  jupyter:
    image: jupyter/scipy-notebook
    container_name: jupyter
    volumes:
      - $LOCAL_PATH:/home/jovyan/work/local
    ports:
      - "8888:8888"

volumes:
  workspace:
```





# Conclusion

Plan

Introduction

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

Hands-on

Conteneurs

Applications

Image build

Container deployment

**Conclusion**

## Useful links:

- Play-with-Docker:  
<https://training.play-with-docker.com>
- IN2P3 tutorials: <https://gitlab.in2p3.fr/MaitresNageurs/EnBarque>
- Ecole informatique de l'IN2P3 : conteneur en production :  
<https://indico.in2p3.fr/event/17124/>



# The End

Plan

Introduction

Image creation  
and management

Docker Hub

GitLab Registry

Container  
deployment

Docker hosts

Hands-on

Containers

Applications

Image build

Container deployment

Conclusion

