

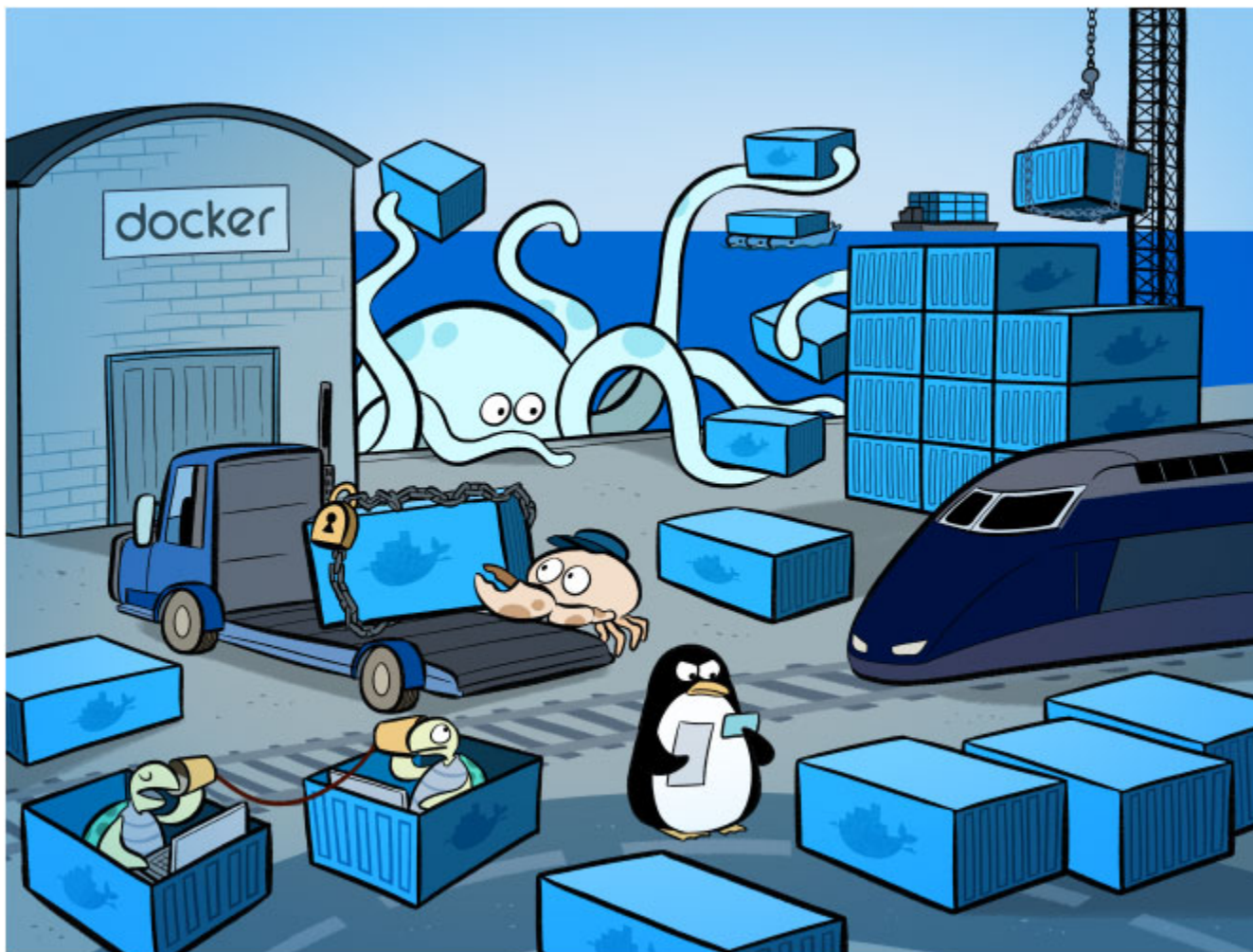


Intérêt de docker « environnement de développement agile et de compilations »

Aymeric Blondel
CEISAM Lab



Docker : A quoi ça peut bien me servir ?



Cas 1 : « **Disposer d'un environnement de développement agile** : associer un environnement de développement python avec virtualenv et une architecture microservice avec docker »

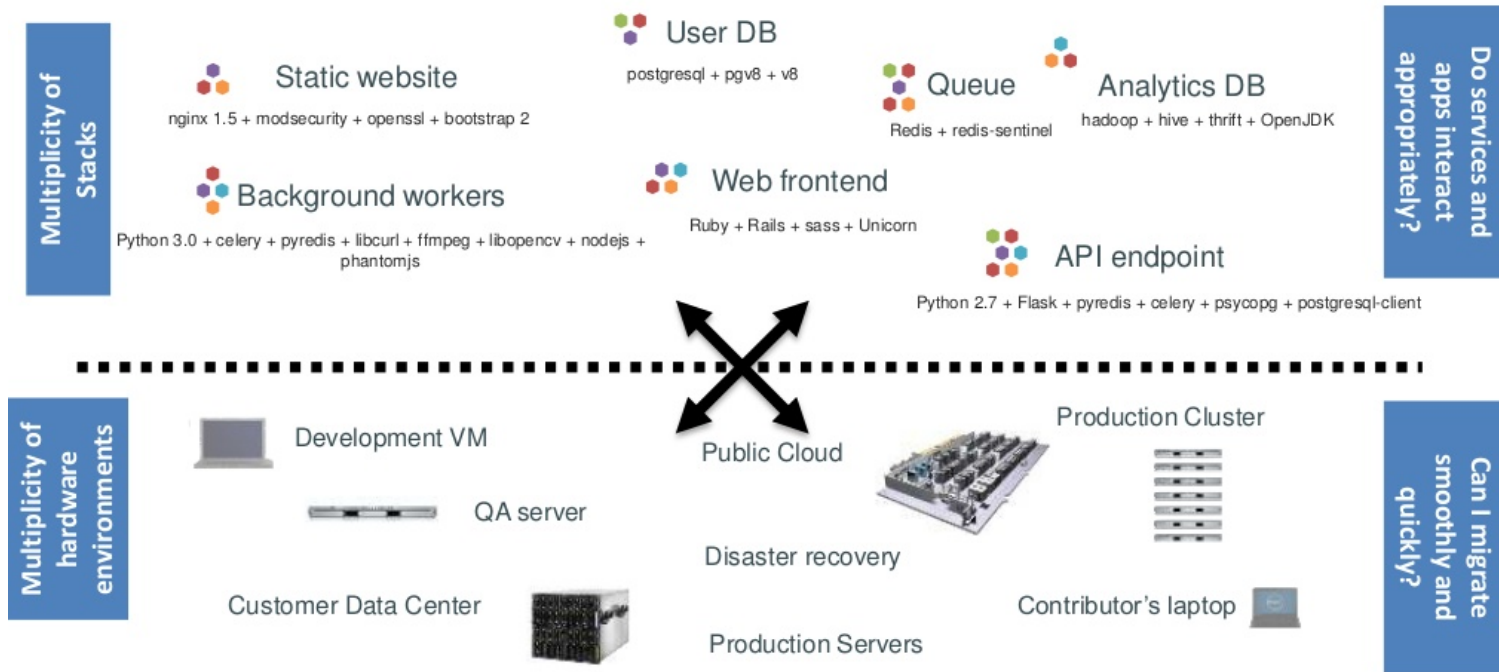


Cas 2 : « Compilations et tests pour différents systèmes d'exploitations sur un seul serveur physique avec docker »



Cas1 : Microservice et docker.

The Challenge



Slide Reference: DockerCon

Cas1 : Projet Modelling (docker & Microservice)

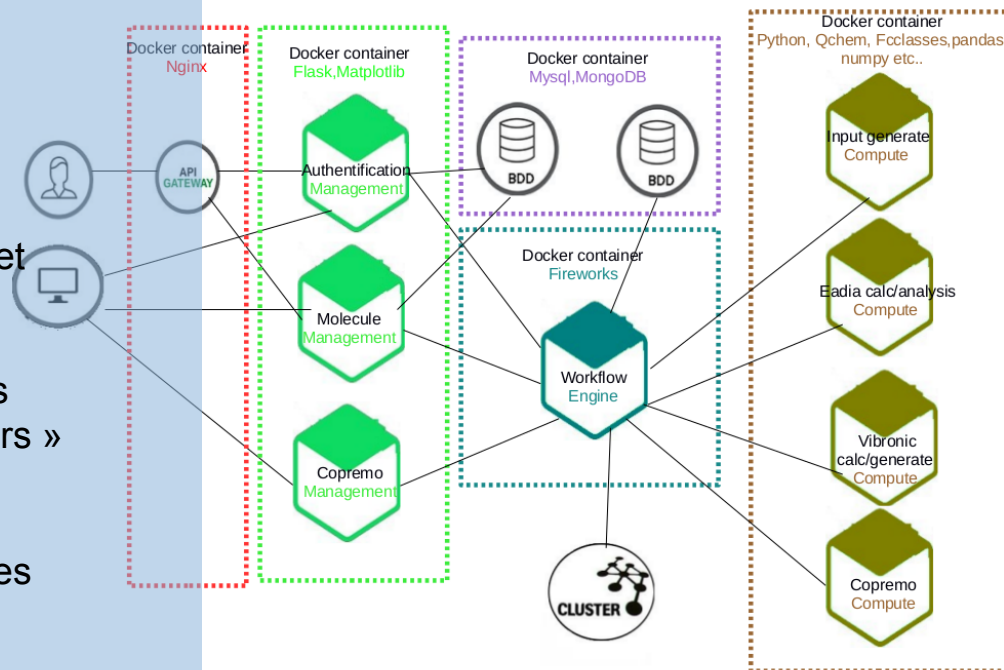
Docker : Solution d'orchestration de Microservices

Intérêt pour développement de Modelling :

- Sécurité : Multi- Conteneurs, permet d'isoler l'utilisateur du cluster (Communication limitée aux requêtes Rest entre les dockers).
- Processus dev/test/prod facilité (docker-registry et « tags d'images, push,pull,commit »)
- Docker sait se faire oublier pour développer sans trop de contraintes bind de port « host - conteneurs » et partage de volumes (docker-compose)
- Facilité de mixer des conteneurs en prod avec des Conteneurs en tests.

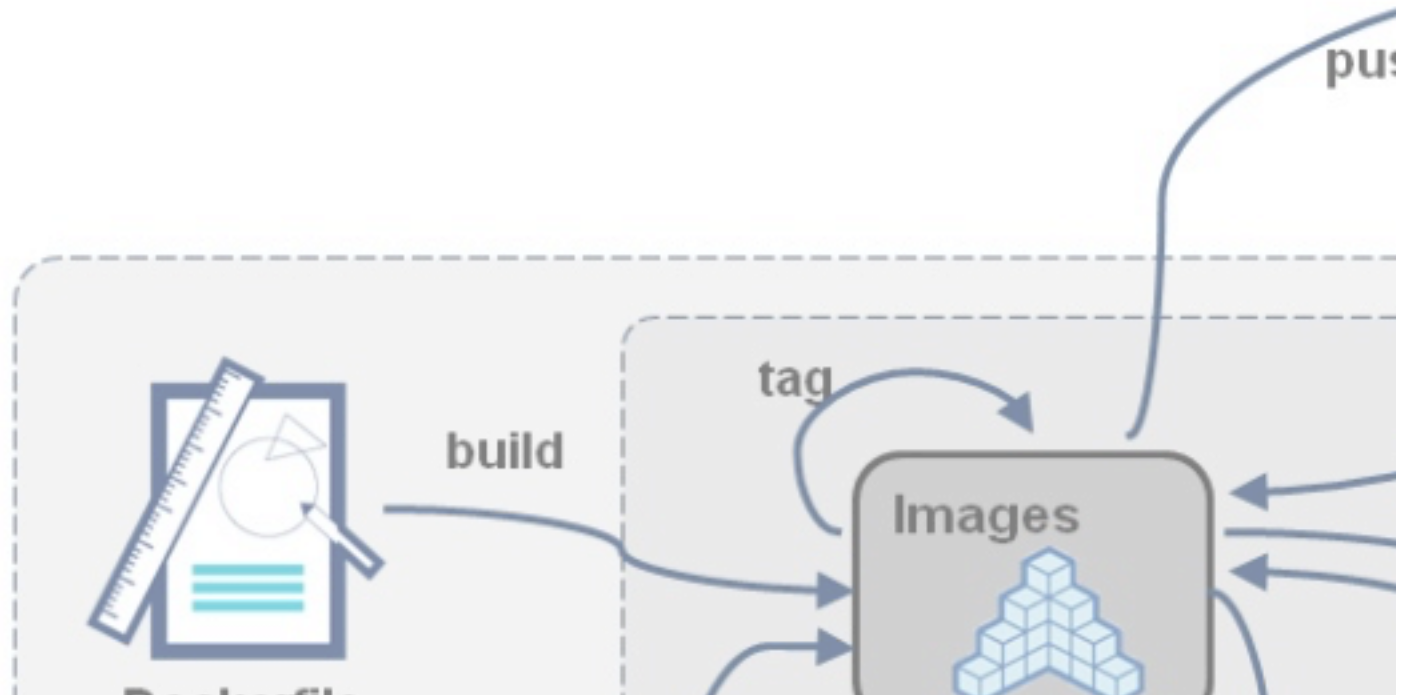
Le synoptique ci-dessous permet d'appréhender la plate forme Modelling dans sa globalité.

Synoptique Modelling



Cas1 : Docker moteur/client

Docker Moteur/C



Cas1 : Projet Modelling (virtualenv)

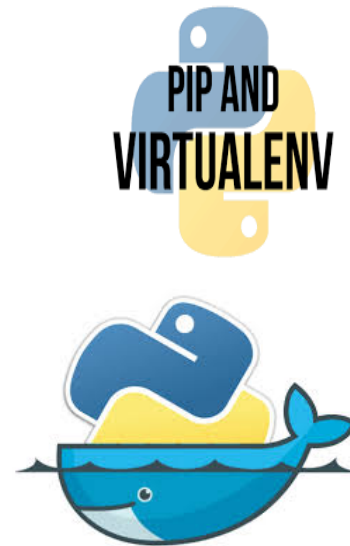
Utiliser docker signifie t'il la fin de l'utilisation de virtualenv ?

Docker est bon pour containeriser et isoler l'OS de l'Host
Virtualenv isole l'environnement python de l'OS.

Actuellement interconnexion forte entre le Python et l'OS.
Partage de site-package entre l'OS et le dev, c'est pas top.

virtualenvwrapper facilite son utilisation.
>workon modelling

Conda est une autre solution de gestion d'environnements
Python



Cas1 : Projet Modelling (Installation 2 conteneurs)

Conteneur installé en prod : eadia

```
mricos@mricos-HP-EliteBook-820-G3 /home2/workspace/python-scripts/trunk/modelling $ docker-compose up -d
```

Tester :
<http://0.0.0.0:5001/>

Conteneur en court de dev :

Prérequis : svn client installé, Docker V11 installé, montage fait, conteneur db (mongo) lancé.

```
Récupérer dernière version de modelling  
Cd /home2/workspace/python-scripts/trunk/modelling_2  
>svn update
```

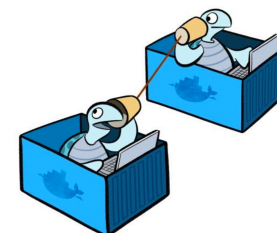
Lancer le docker

```
> docker run --name molecule-mgt -v /home2/workspace/python-scripts/trunk/modelling_1:/svn/modelling/generate_input -v /home2/workspace/python-scripts/trunk/lib:/svn/lib -p 5002:5002 --link db:db -it 172.X.X.X:5000/generate_input:latest /bin/bash
```

Source /env/bin/activate

```
(env)modelling@e00e4478f6b1:/svn/modelling/generate_input  
$ python controller.py &  
control P control Q
```

Tester :
<http://0.0.0.0:5002/>



Cas1 : Docker-compose eadia

```
version: '2'
services:
  web:
    #restart: always
    image : X.X.X.X:5000/eadia
    expose:
      - "5001"
    ports:
      - "5001:5001"
    volumes:
      - /home2/workspace/python-scripts/trunk/modelling/:/svn/modelling
      - /home2/workspace/python-scripts/trunk/lib:/svn/lib
      - /usr/src/app/static
    working_dir: /svn/modelling/
    environment:
      - http_proxy=172.16.1.12:3128
      - https_proxy=172.16.1.12:3128
    env_file: .env
    command : /env/bin/python /svn/modelling/controller.py
```

Cas2 : Serveur de compilation Gaussian

Avant : 1 serveur physique et une licence standalone de PGI

Objectif :

Compilation de Gaussian beaucoup plus fréquentes (développeur au labo) sur diverses machines de constructeurs, générations différentes (Sles11sp3 sur Troy) (Centos6.7 sur le Cluster local). 1 seule licence PGI workstation.

Machine	Waha (serveur compil)	Pauli	Troy Trolls et Loire	Troy Swiips	Pekka Pyykko	Pekka Cixi	CCIP L next
Core type	E5-2620 v4	E5472	X5650	E5-2670-V1	E5-2670-V1	E5-2697-V3	E5-26 30V4
OS	opensuse12	SLES 11	SLES11	SLES11	Centos6.7	Centos6.7	cento s7
libc	2.19	2.11.3	2.11.3	2.11.3	2.12	2.12/2.14	?
compil	X	-tp core-2-64	-tp nehalem-64	-tp nehalem-64	-tp sandybridge- 64	-tp sandybridge- 64	?

Cas2 : Serveur de compilation Gaussian

Problème : PGI accessible uniquement sur un noeud du cluster Troy.

- Impossible de compiler pour profiter nouveau matériel/nouvel OS
- Problème de mise à jour de PGI sans mettre à jour le nœud.

Solution : Achat un serveur de compilations disposant de PGI dernière version et kernel à jour.

Problème : Version de glibc exigée par les versions PGI installées sur Waha et les divers nœuds différents.

Solution : Mettre à jour glibc sur les nœuds ou compiler sur Waha dans un conteneur docker correspondant au nœud.

Problème : Version standalone de PGI bloquée sur une machine @mac,hostid

Solution : Compiler dans un conteneur docker @mac,hostid identique à Host Docker



FLEXlm™

Cas2 : Serveur de compilation Gaussian

Lancer le conteneur :

```
docker run --rm --mac-address="F4:8E:38:C3:6D:70" -h waha --name pgi -v /opt/pgi:/tmp/pgi -v /compil:/compil -p 27000:27000 -it d9a2959dcfb5 /bin/bash
```

Serveur de licence flottante dans conteneur :

```
[root@waha pgi2016]# set path=($path /opt/pgi/linux86-64/2016/bin)
[root@waha pgi2016]# /opt/pgi/linux86-64/2016/bin/lmgrd -c /opt/pgi/license.dat
[root@waha pgi2016]# /opt/pgi/linux86-64/2016/bin/makelocalrc /opt/pgi/linux86-64/2016/bin
```

Droit accès utilisateur nis docker :

```
sudo vi /etc/pam.d/common-auth

# add line
auth optional pam_group.so
vim /etc/security/group.conf

"*;*;*;A10000-2400; docker"
```

Docker mytips

```
Utiliser dernière version docker-engine au lieu docker package de la distribution (/usr/bin)
Docker Private Repository : V2 car moi j'ai mis la V1 et galère pour passer image en V2
Mettre un minimum de contenu perso dans l'image. Utiliser les volumes plutôt
Doker-data a changer au lieu par défaut /lib/docker-data
Utilisateur de docker doit appartenir au groupe docker
Firewall FW_ROUTE = « YES »
Régler le proxy dans docker
[Service]
Environment="HTTP_PROXY=http://172.16.1.12:3128/" "NO_PROXY=localhost,127.0.0.1,X.X.X.X,registry_sphax.univ-
nantes.prive"
Utiliser au maximum Dockerfile et DockerCompose si possible
```

Exemple lancement daemon docker :

```
/lib/systemd/system/docker.service
```

```
ExecStart=/usr/bin/docker daemon -H fd:// --graph="/home3/docker" --storage-driver=overlay --insecure-
registry :X.X.X.X :5000
```

Merci de votre attention

