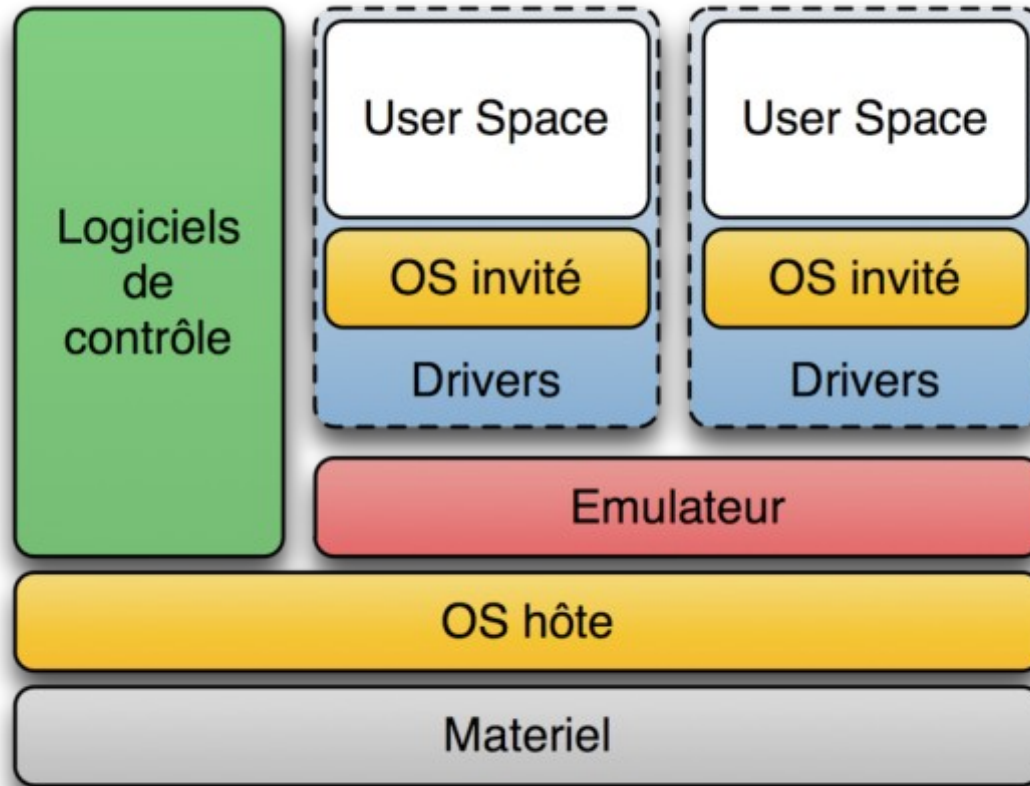


Vagrant, Puppet, Docker sont dans un bateau

20/05/2014

La virtualisation, c'est quoi ?



Exemple d'un Hyperviseur de type 2

Pourquoi ?

Isoler les services

Livraison, et déploiement facilités

Meilleure répartition/mutualisation de la charge

Comment ?

VirtualBox, VMWare Player

VMWare ESXi (vSphere)

Dans le “cloud”: Instances Amazon EC3, VPS, ...

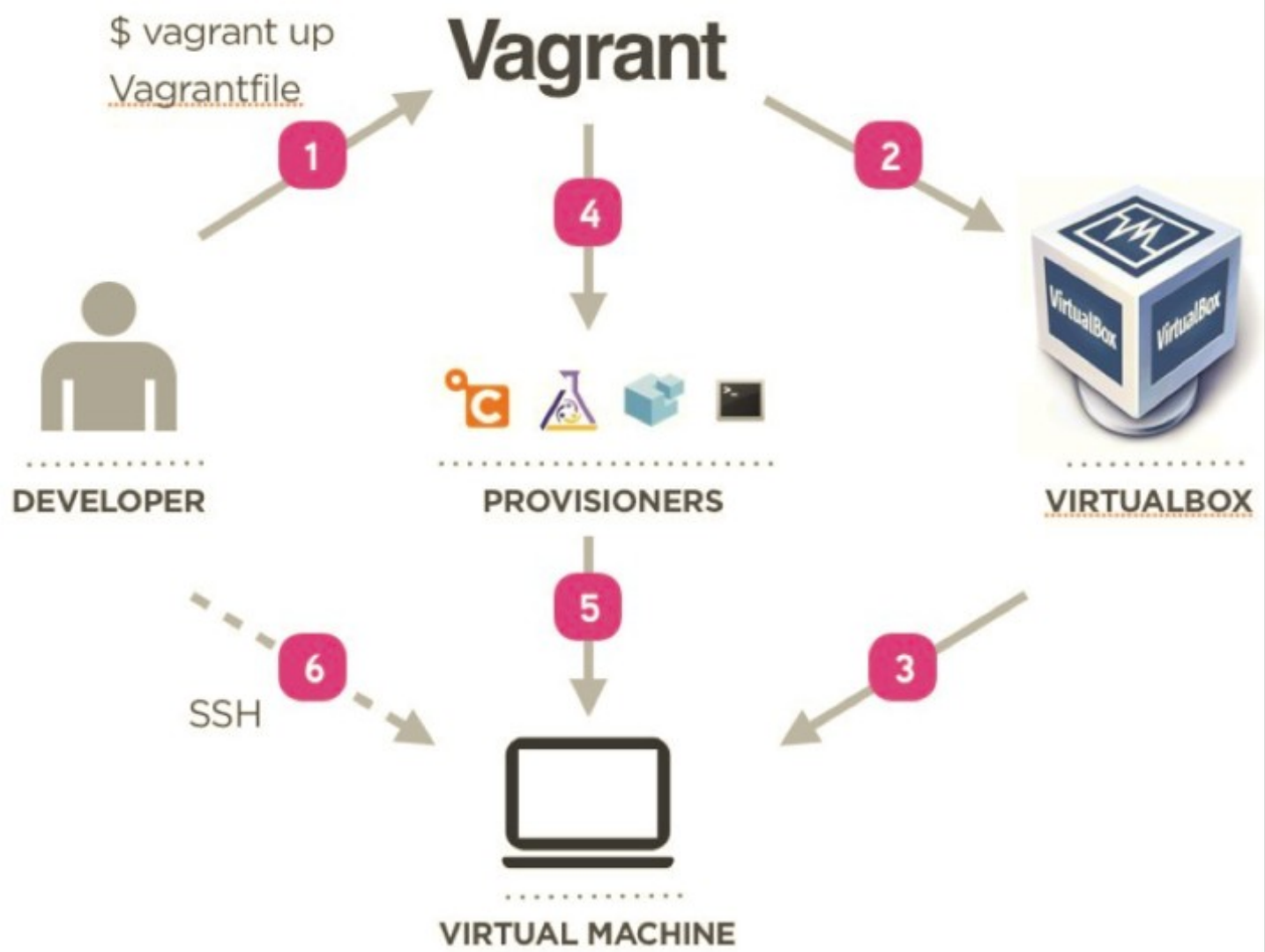
En local avec vagrant

Vagrant ?

“The idea is that developers would be working in these transient, portable boxes. They would move around and would have no permanent residence. Like a vagrant.”

Mitchell Hashimoto, creator of vagrant

<http://www.digitalforreallife.com/tag/chef/>



Provisionnement

Provisioning dans le contexte d'une VM de développement c'est:

- Le Déploiement
- La Configuration
- La gestion

des packages de la stack de développement souhaitée.

Effectuée manuellement c'est une activité

- Chronophage
- Peu fiable
- Difficilement reproductible.

Le provisioning automatisé "self-provisionng" c'est l'exécution sans intervention humaine des même taches en utilisant des scénarios et procédures prédéfinis.

Provisionnement

Le provisioning automatisé c'est tout simplement l'exécution sans intervention humaine des même tâches en utilisant des scénarios et procédures prédéfinis.

2 Solutions Possibles pour l'automatisation du provisioning

```
1 sudo debconf-set-selections <<< 'mysql-server-5.5 mysql-server/root_password password root'
2 sudo debconf-set-selections <<< 'mysql-server-5.5 mysql-server/root_password_again password root'
3 sudo add-apt-repository -y ppa:chris-lea/node.js
4 sudo add-apt-repository -y ppa:ondrej/php
5 sudo apt-get update
6 sudo apt-get upgrade
7 sudo apt-get -y install mysql-server-5.5 php5-mysql libapache2-dev apache2 php5-dev build-essential php-pear nodejs npm
8 sudo apt-get install -y python-software-properties python g++ make php-apc
9 sudo npm install -g bower
10
11 # Set timezone
12 echo "America/New_York" | sudo tee /etc/timezone
13 sudo dpkg-reconfigure --frontend noninteractive tzdata
14
15 # Setup database
16 if [ ! -f /var/log/databasesetup ];
17 then
18     echo "DROP DATABASE IF EXISTS mine-be-cool" | mysql -uroot -proot
19     echo "CREATE USER 'capi@localhost' IDENTIFIED BY 'capi'" | mysql -uroot -proot
20     echo "CREATE DATABASE mine-be-cool" | mysql -uroot -proot
21     echo "GRANT ALL ON capi.* TO 'capi@localhost'" | mysql -uroot -proot
22     echo "Flush privileges" | mysql -uroot -proot
23 fi
24
25 # Apply changes
26 if [ ! -f /var/log/webserversetup ];
27 then
28     echo "Servername localhost" | sudo tee /etc/apache2/httpd.conf & chmod 644
29     sudo a2enmod rewrite
30     sudo sed -i "s/AllowOverride None/AllowOverride All/" /etc/apache2/sites-available/default
31     sudo touch /var/log/webserversetup
32 fi
33
34 # Install Hericatcher
35 if [ ! -f /var/log/hericatchersetup ];
```

ou



Shell Demo

Démo



- Un outil de configuration de serveur, automatisable.
- Il utilise son propre langage
- Peut fonctionner en client/serveur (« puppetmaster »), ou seul.

Puppet Démo

Démo

Docker

“Docker is an open-source engine to easily create lightweight, portable, self-sufficient containers for any application.”



Docker

Docker étend la technologie Linux Containers (LXC) qui permet de faire tourner un Os Linux à l'intérieur d'un autre avec la particularité que les deux Os partageront le Kernel de la machine physique.

Contrairement à une VM traditionnelle, un container Docker n'embarque pas son propre système d'exploitation.

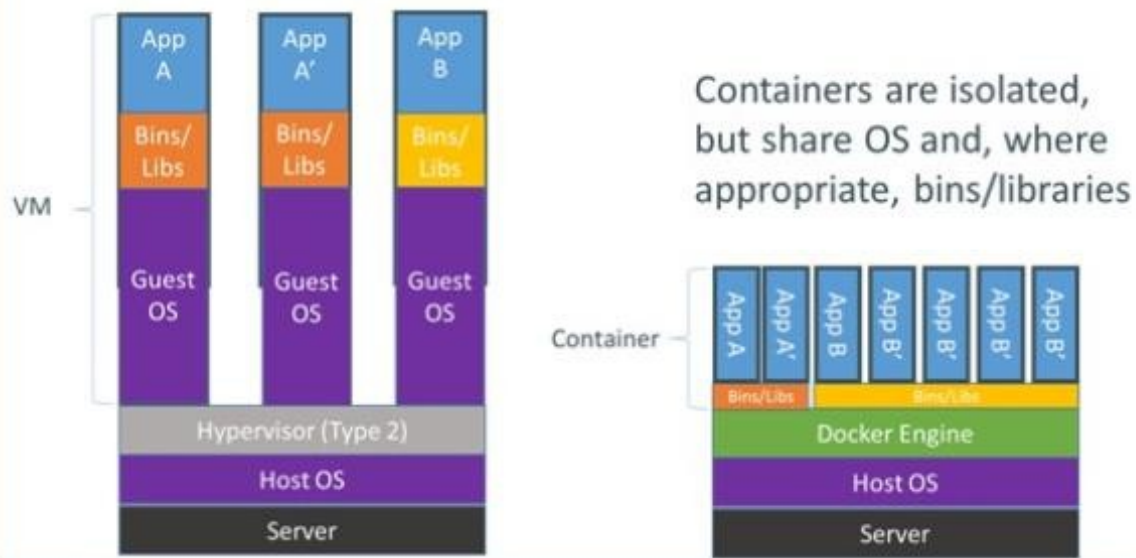
Un container Docker va donc essentiellement créer une enveloppe autour de l'application, des packages et des dépendances nécessaires pour que l'ensemble puisse tourner sur tout système Linux.

Versioning "git-like" qui permet la traçabilité des successives versions du container, inspection des diff entre versions, commit des nouvelles versions, pull/push de containers depuis/vers un dépôt distant, effectuer des roll back etc...

Docker définit une API qui permet de l'intégrer avec des outils d'orchestration tels que Puppet, Chef ou Ansible, d'intégration continue tels que Jenkins ou Travis

Docker

Containers vs. VMs



The End