

# Sesión 5

## Vagrant

- vagrant init
  - vagrant init hashicorp/precise64
- vagrant up
- vagrant ssh
- ubicación imágenes vagrant: **\$HOME/.vagrant.d/boxes**
- vagrant box list
- vagrant port <maquina : listado puertos máquinas

## ejemplos

### apache+php

#### Vagrantfile

```
Vagrant.configure(2) do |config|
#  config.vm.box = "debian/jessie64"
  config.vm.box = "hashicorp/precise64"
  config.vm.hostname = "apachephp"
  config.vm.provision "shell", path: "install.sh"

#  config.vm.network :private_network, ip: "192.168.100.10" # only host
#  config.vm.network :public_network, ip: "192.168.100.20"
#  config.vm.network :public_network, :bridge=>"eth0"
  config.vm.network :forwarded_port, guest: 80, host: 8080
  config.vm.synced_folder ".", "/var/www"

#  config.vm.provider "virtualbox" do |vb|
#    vb.name = "apachephp"
#    vb.memory = 1024
#    vb.cpus = 1
#    vb.linked_clone = true
#    vb.gui = true
#  end
end
```

#### install.sh

```
sudo apt-get update
sudo apt-get -y install apache2 libapache2-mod-php5
#rm /var/www/index.html
```

#### index.php

```
<?php
```

```
printf ("Hola Mundo!\n" );
```

## apache+mysql

habría que configurar el MYSQL para permitir conexiones y configurar user/pass, no era el objetivo de la práctica

### Vagrantfile

```
$mi_script=<<<SCRIPT
apt-get -y update
apt-get -y install apache2 mysql-client
SCRIPT

$otro_script=<<<SCRIPT
apt-get -y update
apt-get -y install default-mysql-server
SCRIPT

$tercero_script=<<<SCRIPT
apt-get -y install php7.0
SCRIPT

Vagrant.configure(2) do |config|
  config.vm.define "apache" do |config|
    config.vm.box = "debian/stretch64"
    config.vm.hostname = "apachefrontal"
    config.vm.network "private_network", ip: "10.0.7.11"
    config.vm.provision "shell", inline: $mi_script
    config.vm.synced_folder ".", "/vagrant", disabled: true
  end

  config.vm.define "mysql" do |config|
    config.vm.box = "debian/stretch64"
    config.vm.hostname = "mysqlbackend"
    config.vm.network "private_network", ip: "10.0.7.12"
    config.vm.provision "shell", inline: $otro_script
    config.vm.provision "shell", inline: $tercero
    config.vm.synced_folder ".", "/vagrant", disabled: true
  end
end
```

## swarm 3 nodos docker

hay que buscar una imagen que permita compartir de manera sincronizada una carpeta entre los 3 nodos para compartir la información de unirse al swarm (o NFS)

### Vagrantfile

```
$docker = <<<SCRIPT
apt-get -y update
```

```
apt-get -y install curl apt-transport-https
curl -s https://get.docker.com | bash
usermod -aG docker vagrant
SCRIPT

$swarminit = <<SCRIPT
docker swarm init --advertise-addr 10.0.7.11
docker swarm join-token manager | grep swarm | tail -1 >
/vagrant/jointoken.txt
SCRIPT

$swarmjoin = <<SCRIPT
bash /vagrant/jointoken.txt
SCRIPT

Vagrant.configure(2) do |config|
  config.vm.define "swarm1" do |config|
    config.vm.box = "debian/stretch64"
    config.vm.hostname = "swarm1"
    config.vm.network "private_network", ip: "10.0.7.11"
    config.vm.provision "shell", inline: $docker
    config.vm.provision "shell", inline: $swarminit
#   config.vm.synced_folder ".", "/vagrant", disabled: true
  end

  config.vm.define "swarm2" do |config|
    config.vm.box = "debian/stretch64"
    config.vm.hostname = "swarm2"
    config.vm.network "private_network", ip: "10.0.7.12"
    config.vm.provision "shell", inline: $docker
    config.vm.provision "shell", inline: $swarmjoin
#   config.vm.synced_folder ".", "/vagrant", disabled: true
  end

  config.vm.define "swarm3" do |config|
    config.vm.box = "debian/stretch64"
    config.vm.hostname = "swarm3"
    config.vm.network "private_network", ip: "10.0.7.13"
    config.vm.provision "shell", inline: $docker
    config.vm.provision "shell", inline: $swarmjoin
#   config.vm.synced_folder ".", "/vagrant", disabled: true
  end
end
```

## packer

Para construir imágenes exportables/intercambiables entre diferentes entornos cloud/virtualización/docker

- <https://www.packer.io/intro/getting-started/vagrant.html>
- `packer build -only=amazon-ebs example.json`
- <http://packer.io/docs/builders/index.html>

# ansible

hay que usar sus módulos para sacarle provecho

## instalación

- `sudo apt install python-pip`
- `sudo pip install ansible`
- [https://docs.ansible.com/ansible/latest/installation\\_guide/intro\\_installation.html](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html)

## inventory

- relación de máquinas, se pueden agrupar y agrupar los grupos
- un equipo puede forma parte de más de un grupo
- [https://docs.ansible.com/ansible/latest/user\\_guide/intro\\_inventory.html](https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html)

## 4 nodos (diferentes linux para ansible)

### Vagrantfile

```
$mi_script=<<SCRIPT
mkdir -m 0700 /root/.ssh
cp /vagrant/id_ed25519.pub /root/.ssh/authorized_keys
cat /vagrant/id_ed25519.pub >> /home/vagrant/.ssh/authorized_keys
chmod 600 /root/.ssh/authorized_keys
SCRIPT

Vagrant.configure(2) do |config|
  config.vm.define "centos7" do |config|
    config.vm.box = "centos/7"
    config.vm.hostname = "centos7"
    config.vm.network "private_network", ip: "10.0.7.11"
    config.vm.provision "shell", inline: $mi_script
#    config.vm.synced_folder ".", "/vagrant", disabled: true
  end

  config.vm.define "jessie64" do |config|
    config.vm.box = "debian/jessie64"
    config.vm.hostname = "debianjessie64"
    config.vm.network "private_network", ip: "10.0.7.12"
    config.vm.provision "shell", inline: $mi_script
#    config.vm.synced_folder ".", "/vagrant", disabled: true
  end

  config.vm.define "jessie66" do |config|
    config.vm.box = "debian/jessie64"
    config.vm.hostname = "debianjessie66"
    config.vm.network "private_network", ip: "10.0.7.14"
    config.vm.provision "shell", inline: $mi_script
#    config.vm.synced_folder ".", "/vagrant", disabled: true
  end
end
```

```

config.vm.define "trusty64" do |config|
  config.vm.box = "ubuntu/trusty64"
  config.vm.hostname = "ubuntustrusty64"
  config.vm.network "private_network", ip: "10.0.7.13"
  config.vm.provision "shell", inline: $mi_script
#   config.vm.synced_folder ".", "/vagrant", disabled: true
end

end

```

```

[debian]
debian1   ansible_host=127.0.0.1 ansible_ssh_port=2201 ansible_ssh_user=root
debian2   ansible_host=127.0.0.1 ansible_ssh_port=2203 ansible_ssh_user=root

[centos]
centos1   ansible_host=127.0.0.1 ansible_ssh_port=2200 ansible_ssh_user=root

[ubuntu]
ubuntu1   ansible_host=127.0.0.1 ansible_ssh_port=2202 ansible_ssh_user=root

[laboratorio:children]
debian
centos
ubuntu

```

- `ansible -m ping laboratorio`
- `ansible -m apt -a «package=pwgen state=latest update_cache=true» debian`
  - instala correctamente
- `ansible -m apt -a «package=pwgen state=latest update_cache=true» ubuntu`

código de colores:

- rojo: error
- naranja: cambios
- verde: no se ha tocado nada o correcto

## otros

- <http://www.gratisexam.com>
- <https://medium.com/@Joachim8675309/devops-concepts-pets-vs-cattle-2380b5aab313>
- ~
  - desde root: `ls -la ~devops` → te lista el home del usuario devops
  - `ls -la ~:` doble tabulador → te muestra los home de los usuarios definidos en **/etc/passwd**

From:  
<https://miguelangel.torresegea.es/wiki/> - miguel angel torres egea

Permanent link:  
<https://miguelangel.torresegea.es/wiki/info:cursos:pue:devops:sesion5?rev=1551467728>

Last update: **01/03/2019 11:15**

