

Vagrant y ansible. Una combinación explosiva



CEP Lora del Río
CONSEJERÍA DE EDUCACIÓN

José Ignacio Huertas Fernández
IES Polígono Sur

—
Alberto Molina Coballes
IES Gonzalo Nazareno



Curso "Seguridad y Alta Disponibilidad"
Brenes. Septiembre 2013

Cambio de paradigma

Maneja tu infraestructura como tu software

- Utiliza software de control de versiones para las configuraciones y especificaciones
- Todo debe ser legible
- Automatiza

DevOps . . . ¿Eso que lo qué?

Vagrant. Introducción

- Aplicación libre desarrollada en ruby para crear de forma sencilla un entorno de trabajo complejo
- El objetivo principal es aproximar los entornos de desarrollo y producción.
- Utiliza un entorno de máquinas virtuales (Virtualbox por defecto)
- Herramienta imprescindible para devops
- Potente uso didáctico
- Su lema:

Development environments made easy: Create and configure lightweight, reproducible, and portable development environments.

Vagrant. Instalación

- Instalar virtualbox
- Instalar vagrant (descargar de <http://vagrantup.com>)
- Notas:
 - Es posible utilizar otros proveedores, pero el mejor documentado es virtualbox.
 - Existen versiones de vagrant para GNU/Linux, Apple OS X y Ms windows
 - Vagrant tiene un fuerte desarrollo, por lo que hay bastantes diferencias entre versiones
 - En Debian Wheezy se incluye vagrant 1.0.3, mientras que la última versión disponible (Septiembre 2013) es 1.3.1
 - Si empiezas ahora es recomendable utilizar vagrant \geq 1.2.0 porque se ha modificado la API a la versión 2.

Vagrant. Vagrantbox

- Un *box* es una máquina virtual empaquetada.
- Vagrant proporciona algunos *boxes* oficiales, con idéntica clave ssh por lo que no son válidos para entornos en producción
- Listado incompleto y no oficial de *boxes*
<http://www.vagrantbox.es/>
- Siempre es posible hacerse un *box* a medida, aunque en la mayoría de los casos no es necesario
- Para poder utilizar un *box* es necesario agregarlo al equipo:

```
vagrant box add {title} {url}
```

- En el caso de sistemas UNIX, esto hace que se descargue y descomprima el “box” en `~/.vagrant.d/boxes/`

Vagrant. Vagrantfile

- Un fichero de nombre Vagrantfile describe una o varias instancias para crear un entorno vagrant en el directorio en el que se encuentre
- Se pueden tener varios entornos vagrant independientes, ubicando ficheros Vagrantfile en diferentes directorios.
- Ejemplo de Vagrantfile:

```
# -*- mode: ruby -*-  
# vi: set ft=ruby :
```

```
Vagrant.configure("2") do |config|  
  config.vm.box = "precise32"  
  config.vm.network :private_network, ip: "10.1.1.2"  
end
```

- Se puede crear un Vagrantfile mínimo con:
vagrant init

Vagrant. Utilización básica

Todas las instrucciones se aplican sobre el Vagrantfile del directorio en el que se ejecuten

- Levantar el entorno:

```
vagrant up
```

- Recargar el entorno (reinicia las máquinas y aplica los cambios):

```
vagrant reload
```

- Acceder a la instancia:

```
vagrant ssh [nombre_instancia]
```

- Suspender, apagar o destruir:

```
vagrant suspend
```

```
vagrant halt
```

```
vagrant destroy
```

Ansible. Introducción

- En ciencia ficción: Dispositivo que permite comunicarse a mayor velocidad que la luz
- Ansible es una aplicación libre de orquestación escrita en python
- La primera versión se desarrolló en febrero de 2012 (!)
- Ventajas frente a otro software de automatización (puppet o chef):
 - Muy fácil instalación
 - Comunidad de desarrolladores muy activa
 - Muy pocas dependencias
 - Muy pequeño
 - Ideal para entornos pequeños y homogéneos
 - No es necesario configurar nada en los clientes
- Inconvenientes:
 - Más complejo en entornos muy heterogéneos
 - Menos recetas desarrolladas previamente al ser muy nuevo



Ansible. Instalación

Para utilizar la última versión:

```
# apt-get install python-pip  
# pip install ansible
```

Voilà!

Ansible. Playbooks

Las tareas a realizar por ansible se describen en formato yaml en ficheros denominados “playbooks”, que pueden incluir una o varias jugadas, por ejemplo:

```
---  
- name: Apache installation  
  hosts: all  
  sudo: True  
  tasks:  
    - name: Set proper sources.list  
      copy: >  
        src=files/etc/apt/sources.list  
        dest=/etc/apt/sources.list  
        owner=root  
        mode=644  
  
    - name: Ensure apache is installed (after updating)  
      apt: pkg=apache2 update_cache=yes
```

Vagrant + Ansible

Imagina que la siguiente situación para una determinada configuración de máquinas virtuales se pudiera hacer automáticamente:

- Levantar las máquinas virtuales
- Configurar la red
- Realizar las actualizaciones de software
- Instalar los paquetes del software que se va a utilizar
- Copiar los ficheros de configuración de plantillas previamente creadas
- Levantar o parar los servicios que se determinen
- Ejecutar cualquier tarea en cada una de las máquinas virtuales
- Dejar el entorno completamente operativo

No es ningún sueño: Eso es vagrant + ansible :-)