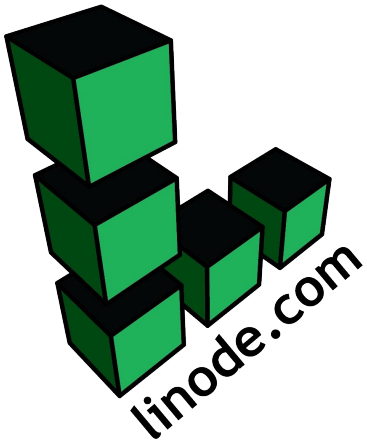
2.2
CONFIGURACIÓN DEL ENTORNO

CONFIGURACIÓN DEL SERVIDOR

Una vez hemos configurado el entorno en el que se ejecutará Ansible, pasaremos a **configurar el nodo**, es decir, el **servidor que vamos a gestionar desde Ansible**.

CONFIGURACIÓN DEL SERVIDOR

https://goo.gl/Eaxafs



CONFIGURACIÓN DEL SERVIDOR

Dado que **ya conocemos Vagrant** y **es gratuito**, lo utilizaremos para hacer nuestras pruebas con Ansible.

CONFIGURACIÓN DEL SERVIDOR

vagrant-node/Vagrantfile

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

# Vagrantfile API/syntax version. Don't touch unless you know what you're doing!
VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.box = "bento/ubuntu-16.04"
  config.vm.network "private_network", ip: "192.168.33.11"
  config.vm.provider "virtualbox" do |v|
    v.memory = 2048
    v.cpus = 1
  end
end
```

CONFIGURACIÓN DEL SERVIDOR

Iniciamos la máquina virtual

```
$ vagrant up
```

CONFIGURACIÓN DEL SERVIDOR

Ahora tenemos dos máquinas virtuales:

- **vagrant-ansible** (192.168.33.10)
- **vagrant-node** (192.168.33.11)

Tenemos que comprobar si hay conexión entre ellas.
Desde **vagrant-ansible:**

```
$ ping 192.168.33.11
```

CONFIGURACIÓN DEL SERVIDOR

¿Podemos hacer SSH desde **vagrant-ansible** a **vagrant-node**?

```
$ ssh vagrant@192.168.33.11
```

CONFIGURACIÓN DEL SERVIDOR

Tenemos el **mismo *hostname* en ambas máquinas virtuales**, lo cual puede llevar a confusión. Los nuevos nombres de host serán:

- **vagrant-ansible**: controller
- **vagrant-node**: node

CONFIGURACIÓN DEL SERVIDOR

Añadimos al archivo Vagrantfile de cada máquina virtual:

vagrant-ansible: **config.vm.hostname = "controller"**

vagrant-node: **config.vm.hostname = "node"**

CONFIGURACIÓN DEL SERVIDOR

Reiniciamos las máquinas con *provision*:

```
$ vagrant reload --provision
```

CONFIGURACIÓN DEL SERVIDOR

Ahora que ya hemos cambiado el *hostname* de las máquinas virtuales, ¿podemos hacer SSH como root a **vagrant-node** desde **vagrant-ansible**?

```
$ ssh root@192.168.33.11
```

CONFIGURACIÓN DEL SERVIDOR

No, porque **el login como root está deshabilitado.**

La solución es **entrar con el usuario vagrant, pasarse a root, cambiar la contraseña** y **habilitar el acceso por SSH.**

CONFIGURACIÓN DEL SERVIDOR

```
vagrant@node:~$ sudo su
root@node:/home/vagrant# passwd
Enter new UNIX password:  vagrant
Retype new UNIX password: vagrant
passwd: password updated successfully
root@node:/home/vagrant# vi /etc/ssh/sshd_config
```

Cambiar **PermitRootLogin** a **yes**

```
root@node:/home/vagrant# service ssh restart
```

CONFIGURACIÓN DEL SERVIDOR

¿Podremos ahora entrar como root?
(Contraseña: vagrant)

```
vagrant@controller$ ssh root@192.168.33.11
```

CONFIGURACIÓN DEL SERVIDOR

Es preferible que Ansible pueda entrar como usuario root sin necesidad de introducir la contraseña el servidor. Para ello, copiamos las claves SSH de *controller* a *node*:

```
vagrant@controller:~$ ssh-copy-id root@192.168.33.11
```

CONFIGURACIÓN DEL SERVIDOR

```
vagrant@controller:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
...
```

CONFIGURACIÓN DEL SERVIDOR

```
vagrant@controller:~$ ssh-copy-id root@192.168.33.11

…

Number of key(s) added: 1

Now try logging into the machine, with:
"ssh 'root@192.168.33.11'"
and check to make sure that only the key(s) you
wanted were added.
```

CONFIGURACIÓN DEL SERVIDOR

También copiaremos las claves al usuario vagrant porque no siempre hará falta ejecutar comandos como root.

```
vagrant@controller:~$ ssh-copy-id vagrant@192.168.33.11
```