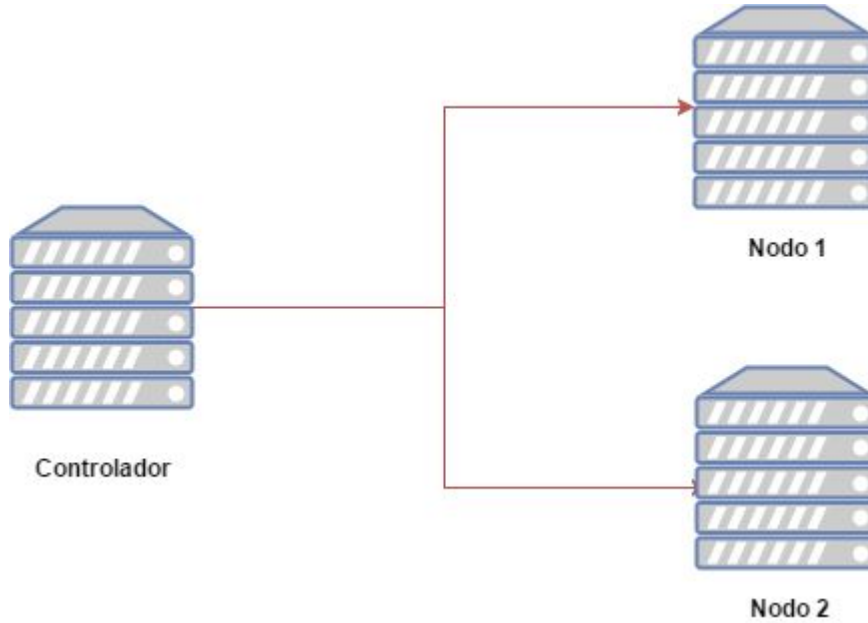


3.

EJECUTANDO LOS PRIMEROS COMANDOS

EL DÍA A DÍA DE UN SYSADMIN

ARQUITECTURA PARA PRUEBAS



INVENTARIOS

INVENTARIOS

Existen diferentes maneras de decirle a Ansible qué servidores va a gestionar.

La más fácil es añadir nuestras máquinas al **inventario** que **Ansible tiene en nuestro propio sistema**, el cual se localiza en **/etc/ansible/hosts**

```
vagrant@controller:~$ cat /etc/ansible/hosts
```

INVENTARIOS

Añadimos las direcciones de nuestras máquinas Vagrant IP al fichero `/etc/ansible/hosts` en la máquina *controller*:

```
192.168.33.11 # node-one  
192.168.33.12 # node-two
```

COMANDOS AD-HOC

COMANDOS **AD-HOC**

Una de las primeras cosas que necesitamos hacer es **comprobar que están sincronizados en fecha y hora.**

Además, **comprobaremos que tienen los recursos disponibles como para ejecutar una aplicación.**

COMANDOS AD-HOC



Muchas de las cosas que comprobamos manualmente deberían ser monitorizadas por un sistema automatizado como Zabbix, DataDog o New Relic.

Si estás ejecutando una web o servicio en producción, también deberías un servicio de monitorización externo como Pingdom.

COMANDOS AD-HOC

Vamos a comprobar el *hostname* de ambos nodos de forma paralela con Ansible:

```
vagrant@controller:~$ ansible all -a "hostname"  
192.168.33.11 | SUCCESS | rc=0 >>  
node-one  
  
192.168.33.12 | SUCCESS | rc=0 >>  
node-two
```

COMANDOS **AD-HOC**

El comando anterior es equivalente a:

```
$ ansible 192.168.33.11,192.168.33.12 -a "hostname"
```

COMANDOS **AD-HOC**

Por defecto, **Ansible ejecuta los comandos en paralelo para que acabe antes**. Si tenemos dos servidores prácticamente no vamos a notar la diferencia, pero **en cuanto añadamos varios servidores comprobaremos que va más rápido si paralelizamos**.

COMANDOS AD-HOC

Para comprobar la ejecución secuencial de comandos utilizaremos el parámetro **-f 1**.

```
vagrant@controller:~$ ansible all -a "hostname" -f 1
192.168.33.11 | SUCCESS | rc=0 >>
node-one

192.168.33.12 | SUCCESS | rc=0 >>
node-two
```

COMANDOS AD-HOC

Es muy poco común ejecutar los comandos de forma secuencial. Sin embargo, el parámetro **-f** se suele utilizar en caso de que haya muchos nodos a gestionar y el ancho de banda de nuestra red pueda verse afectado si accedemos a todos ellos al mismo tiempo.

En este caso, incrementaríamos el valor de **-f** para ejecutar los comandos en grupos.

COMANDOS **AD-HOC**

¿Tienen nuestros servidores suficiente espacio de almacenamiento libre?

```
vagrant@controller:~$ ansible all -a "df -h"
```

Tal y como vemos en la salida del comando, **tenemos bastante espacio disponible.**

COMANDOS **AD-HOC**

Memoria disponible

```
vagrant@controller:~$ ansible all -a "free -m"
```


COMANDOS **AD-HOC**

Sincronización de fecha y hora

Hay aplicaciones bastante tolerantes a que haya una pequeña diferencia de tiempos entre servidores, pero **nunca está de más comprobar que estos tiempos están lo más cerca posible.**

La manera más fácil de hacerlo es utilizar NTP (*Network Time Protocol*).

COMANDOS **AD-HOC**

Sincronización de fecha y hora

```
vagrant@controller:~$ ansible all -a "date"
192.168.33.11 | SUCCESS | rc=0 >>
Sat Jun  3 16:27:02 UTC 2017

192.168.33.12 | SUCCESS | rc=0 >>
Sat Jun  3 16:27:02 UTC 2017
```

COMANDOS **AD-HOC**

Facts

En la jerga Ansible, los *facts* son los detalles de un determinado servidor o grupo de servidores. Es posible obtener una lista exhaustiva de los *facts* haciendo uso del módulo **setup**:

```
vagrant@controller:~$ ansible all -m setup
```