

Lección 3 - P3 ISE

Ansible

Proyecto inicializado por Red Hat. Producto realizado para la configuración, es decir, las labores de los admin de sistemas. Creación de las bases de datos, etc.

Historicamente estas configuraciones se llevaban a cabo, a base de una lista e ir tachando la ya realizado, despues meidante VMs.

Vagrant: herramienta que gestiona muchas máquinas virtuales.

Hoy en día se usan herramientas que definen la configuración en archivos en texto plano para poder cambiar de tecnologías, plataformas, etc, y mucho git, la verdad es que git es super importante, es la polla, si eso, toma toma oleee

Herramientas con un documento de entrada estructurado en texto plano donde especificamos la configuración sobre nuestro sistema. Ansible hace esto.

Ansible coge este documento, se conecta al dispositivo y hace estas vainas,es decir, pasa lo que está en texto plano a las diferentes máquinas. Se puede realizar en paralelo, vamos, to polluo.

! Nosotros lo veremos en automatización, pero su mayor uso es Playbooks, ficheros de texto para definir la config de un equipo. Competidores: Puppet y Chef.

Comensamos a ver cositas ricas que puede hacer esta rechida app

Vamos a lanzar Ansible desde nuestra máquina anfitriona. En mi caso Ubuntu

! Inventario: lista de ordenadores en los que vamos a trabajar, se encuentra en `/etc/ansible/hosts`

Instalamos con `apt install ansible`



En `/etc/ansible/ansible.conf` la configuración de esto.



En Ansible hay un fichero que se instala en `/home`, `.ansible.cfg`, por si no eres para poder configurar la herramienta.

En host añadimos una ip así: `nombre_que_quieras_poner`
`ansible_host=ip_a_la_que_nos_conectaremos` si el nombre esta activo en una DNS o algo de eso pues con el nombre listo, no hay que darle la ip.

Ejecutamos en la máquina que actuará como server: `ansible -m (m de modulo que es un conjunto de comandos) ping nombre del pc al que haremos ping(nombre dado en hosts)`, da error, vamos a arreglarlo.

Vamos a solucionar el problema, el usuario que quiere lanzar es el propio de nuestra máquina, entonces no lo reconoce la máquina a la que nos queremos conectar. ¿Como arreglamos esto? `ansible -m ping -u usuario(nuestro user del pc al que nos conectaremos) nombre_pc` Vuelve a dar error, vaya por dios!!

Es porque nuestro user pide contraseña, ya que realmente ansible lo que hace es conectarse por `ssh`.

Vamos a pasarle clave pública de nuestro ordenador personal, para eso como sabemos, ejecutaremos `ssh-keygen`, posteriormente `ssh-copy-id usuario@ip` y en usuario e ip marcaremos nuestros usuarios de las ips a las que queremos tener acceso. Ya ya nos pedirá contraseñas. Ole ole.

Ansible necesita acceso `ssh` sin contraseña y `python`.

Probamos de nuevo a ejecutar `ansible -m ping -u usuario nombre_pc` ahora si, todo ha ido guachi piruli.



Para pasar el parámetro que queramos hacemos `-a 'data="hola radiola"'` después de `ping` por defecto nos devuelve pong al ejecutar `ping` si usamos `-a 'loquesea'` devolverá lo entrecomillado.



Para no tener que sabernos de memoria los usuarios en el inventario (archivo `/etc/ansible/hosts`), con `ansible_user=usuario` ya no hace falta pasarle el `-u usuario`. Tal que así:

```
ubuntuise ansible_user=josemanu ansible_host=192.168.56.105
centosise ansible_user=josemanu ansible_host=192.168.56.110
```

Si queremos ejecutar algo en todas las máquinas podemos terminar nuestra línea de comando con `all` en lugar de poner el nombre de la máquina.

! Esta bacanería de herramienta hace las cosas en paralelo, es la hostia (máximo número de hebras 5 pero se puede ampliar meu hermano)

En inventario (`/etc/ansible/hosts`) podemos asignar nombres simbólicos (lamp por ejemplo) y debajo añade las máquinas que tienen lamp (siguiendo el ejemplo tomado) ahora podemos terminar nuestra línea de comandos con `lamp` y solo ejecutará las máquinas que haya debajo de esa línea

Otro módulo: `-m shell` para ejecutar comandos remotos `-m shell comando` si queremos que muestre el path va a ser el que tengamos definido en `/etc/profile`

! Inciso: es normal configurar el equipo en dos fases, primero los usuarios y sus paths y posteriormente el resto de la configuración

Sin el módulo también funciona igual, `-m shell` es el módulo por defecto (vamos que podemos pasar de poner `-m shell`)

Módulo de servicio: `-m service` equivalente a `systemctl`

Ansible tiene programación declarativa, describe el estado al que queremos llevar la máquina. Es decir, le pasas el estado a donde quieres llegar, es por ello que si le decimos que pare apache, si ya está parao no hace na, y si no pues lo para

▼ Vamos a parar apache como ejemplito

`ansible nombrepc -m service -a "name=httpd state=stopped"` da error porque no tenemos permisos de super usuario. Podemos configurar pa que esto no pase, lo veremos más adelante.

! Inciso: podemos hacer nuestros propios módulos. Ansible lo que hace es conectarse por `ssh`, después ejecuta el servicio y después con los parametros que le hemos pasao.

Lo que haceos tiene que ser idempotente, es decir, no guarda estados

Para ser root, existe un comando `--become` que habrá que introducirlo en nuestra línea de comandos, pues creamos un user sin contraseña que pueda ser root sin contraseña (mu peligroso) Para hacer esto accedemos a `/etc/sudoers`, y ahí podemos modificar. cada sistema es un mundo y cada configuración es un mundo. En la mía, permito a todos los users ser root, para que puedan serlo eliminaré lo que hay atrás : en la siguiente foto →

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
```

y ya funciona lo de antes.

Si queremos ejecutar lo de antes en dos máquinas a la vez, en este caso Ubuntu y CentOS, las cuales usan diferente nombre para el servicio `http` deberemos de crear un servicio. Lo haremos así:

En cada línea dentro del archivo `hosts` crearemos un servicio con el nombre que queramos, siguiente el ejemplo en la línea de Ubuntu será:

`httpfuncional_service='apache2'`, y en CentOS: `httpfuncional_service='httpd'`

Cambiamos en `name` lo que teníamos por `{{httpfuncional}}`

Practicar en casa: apagar todas las maquinas, que guay

```
ansible all -m shell -a "/usr/sbin/reboot" --become
```