

mucho sentido que tengan todo un *playbook* para ellas, o tareas que realizar de forma esporádica.

COMO UTILIZAR LOS COMANDOS AD-HOC

Entre las diferentes herramientas que instalas con **Ansible**, como te indicaba en el apartado anterior, una de ellas es precisamente `/usr/bin/ansible`. El modo de uso es tan sencillo como,

```
ansible [patrón] -m [módulo] -a [opciones del módulo]
```

Respecto al concepto de módulo, esto es lo que trataba de comentarte en la introducción. Este concepto lo traté en el primer capítulo del tutorial, indicando que, se trata de una librería donde se encuentran las funcionalidades necesarias para desempeñar determinadas tareas.

Para probar esto de los comandos ad-hoc de Ansible, porque se que estás tan impaciente tu como yo por hacerlo, puedes hacerlo con el módulo `ping`. Este módulo, como bien te imaginas sirve para hacer `ping` sobre una máquina o *host* de tu entorno. Así, en mi caso, el uso es el siguiente, en el caso de querer hacer `ping` sobre todas las máquinas,

```
ansible all -i /datos/Sync/ansible/inv -m ping
```

En el ejemplo de inventario que he puesto anteriormente también podrías hacer `ping` a todos los hosts que cumplan con el patrón `host1*` de la siguiente forma,

```
ansible host1* -i inv -m ping
```

En el directorio `inv` es donde se encuentra mi inventario `inventory.yml`, pero como ya te indiqué en el capítulo anterior del tutorial, con indicar el directorio es suficiente. Ansible ya se encarga de sacar todos los hosts y prepararlo como necesite.

En este último ejemplo, en lugar de poner `/datos/Sync/ansible/inv` he indicado únicamente `inv` partiendo que me encuentro en el directorio `/datos/Sync/ansible`, que es mi *directorio de operaciones*.

Igual que hice `ping` a todas las máquinas, podía haber hecho a una en concreto, o a un grupo de ellas. Simplemente con sustituir por el nombre o alias de la máquina o del grupo, o utilizando un patrón, tal y como expliqué en un apartado anterior de este capítulo.

ALGUNOS CASOS DE USO INTERESANTES

Tal y como he indicado anteriormente, uno de los usos mas habituales de los comandos ad-hoc de Ansible es para ejecutar acciones que no realizas de forma habitual.

REINICIANDO UN SERVIDOR

Una de estas acciones, es sin lugar a dudas reiniciar un equipo. Esto lo puedes realizar de forma sencilla utilizando el módulo **reboot** tal y como te indico en la siguiente instrucción,

```
ansible host11 -i inv -m reboot --become
```

Al ejecutar esto, recibirás una respuesta parecida a la que te muestro a continuación,

```
host11 | CHANGED => {
  "changed": true,
  "elapsed": 15,
  "rebooted": true
}
```

Si lo ejecutas sin el argumento `--become` no funcionará, devolviendo algo como lo que te muestro a continuación,

```
host11 | FAILED! => {
  "changed": false,
  "elapsed": 0,
  "msg": "Reboot command failed. Error was Failed to set wall message, ignoring: Interactive",
  "rebooted": false,
  "start": "2020-06-18T05:27:57.287009"
}
```

Por supuesto también puedes ejecutar esta instrucción para todos los hosts que cumplan con el patrón `host1*`, por ejemplo, o también para el `grupo1`. Para esto tan solo tienes que reemplazar `host11` por el valor correspondiente.

Una vez ejecutado la instrucción anterior, a lo mejor quieres comprobar que la máquina se ha reiniciado. Para esto puedes ejecutar la siguiente instrucción,

```
ansible host11 -i inv -m command -a 'uptime'
```

Esto te devolverá un resultado similar al que te muestro a continuación,

```
host11 | CHANGED | rc=0 >>
07:36:07 up 7 min, 1 user, load average: 0.00, 0.06, 0.05
```

Indicarte que el comando anterior lo puedes simplificar ejecutando simplemente,

```
ansible host11 -i inv -a 'uptime'
```

Esto es así porque el módulo por defecto para Ansible es `command`. Este módulo te permite ejecutar los mismos comandos que ejecutarías en la línea de comandos, como por ejemplo `uptime`.

Así, por curiosidad puedes ejecutar estas dos instrucciones en la línea de comandos, y obtendrás seguro el resultado esperado,

```
ansible host11 -i inv -a 'whoami'
```

En este caso el resultado será `ansible` . Sin embargo, si lo ejecutas con la opción `become` , tal y como has hecho anteriormente, el resultado será, otro. Es decir, si ejecutas,

```
ansible host11 -i inv -a 'whoami' --become
```

En este caso, el resultado será `root`

CONCLUSIONES

Esto no es mas que el aperitivo de lo que está por llegar. Ya puedes ver las posibilidades de esta tecnología. No solo es la comodidad de realizar tareas sin tener que desplazarte a cada una de las máquinas que gestiones, también está la ventaja de que puedes hacerlo en varias máquinas a la vez, y como ya veremos en próximos capítulos, la de gestionar y controlar todas estas operaciones, utilizando para ello, el control de versiones.

Esto es un como el anuncio, **una vez lo destapas, ya no hay forma de parar**. Y vuelvo a insistir, no se trata de que tengas varias máquinas, sino de tener todos tus procesos automatizados bajo control, de forma que los puedas aplicar cuando quieras, o necesites.

En el próximo capítulo del tutorial, comentaré algunos comandos ad-hoc de Ansible mas, para que puedas seguir probando, o simplemente realizando las tareas mas habituales, de una forma alternativa, de una forma mas sencilla

Esto no es mas que el aperitivo de lo que está por llegar. Ya puedes ver las posibilidades de esta tecnología. No solo es la comodidad de realizar tareas sin tener que desplazarte a cada una de las máquinas que gestiones, también está la ventaja de que puedes hacerlo en varias máquinas a la vez, y como ya veremos en próximos capítulos, la de gestionar y controlar todas estas operaciones, utilizando para ello, el control de versiones.

Esto es un como el anuncio, **una vez lo destapas, ya no hay forma de parar**. Y vuelvo a insistir, no se trata de que tengas varias máquinas, sino de tener todos tus procesos automatizados bajo control, de forma que los puedas aplicar cuando quieras, o necesites.

En el próximo capítulo del tutorial, comentaré algunos comandos ad-hoc de Ansible mas, para que puedas seguir probando, o simplemente realizando las tareas mas habituales, de una forma alternativa, de una forma mas sencilla.

Más información,

- [Ansible](#)

Imagen de portada de [Franz Harvin Aceituna](#) en [Unsplash](#)

Listado de capítulos

Estos son todos los capítulos del tutorial [Automatización con Ansible. Una introducción.](#),

[1.- Instalar Ansible. Conceptos básicos y primeros pasos](#)