AUTOMATIZACIÓN CON ANSIBLE. UNA INTRODUCCIÓN.

Publicado el 22 de mayo de 2020 por atareao *37.3K* Deja un comentario

Si has llegado hace poco a esto de Linux. Si hace poco que has empezado a trabajar con el terminal, o si hace poco que estás trabajando con docker, es posible que esto de **Ansible** te suene a **chino**, o incluso que ni siquiera te suene a nada. Sin embargo, **todo es empezar**.

Un día te compras una Raspberry y al mes siguiente te encuentras con una granja de ellas, como si se hubieran reproducido. Es un tremendo vicio. Y lo mismo te puede suceder con VPS. Inicialmente levantas uno para proveer un servicio, y a los dos meses, tienes dos o tres para hacer, vete a saber que cosa. Como ya te puedes imaginar, instalar y configurar una Raspberry, tiene su gracia. Lo mismo te puedo decir de un VPS, pero ¿Y cuando tienes que hacer **cinco**? Y cuando tienes que actualizar los cinco. La solución se encuentra en la automatización. La solución se encuentra en **Ansible**.

¿Pero que es esto de Ansible?¿Como me puede ayudar a mi?¿Si solo tengo una Raspberry también me puede resultar interesante conocer Ansible?

Con independencia de aquello de que el saber no ocupa lugar, lo cierto es que Ansible, te puede valer, tanto para una Raspberry, como cuando tienes una docena o un centenar. Y por supuesto lo mismo te puedo decir, para VPS, por supuesto. No solo se trata de gestionar varios dispositivos de forma sencilla, sino también de poder realizar un proceso repetidas veces con garantías de éxito, porque ¿cuantas veces has instalado una Raspberry?.

AUTOMATIZACIÓN CON ANSIBLE

¿QUE ES ANSIBLE?

Ansible es una herramienta de automatización que se utiliza para configurar, desplegar aplicaciones, orquestación o aprovisionamiento. Es posible que algunas de estas palabras te *suenen raro*, pero poco a poco las irás descubriendo.

Una vez cumplidas con las presentaciones formales, decirte que **Ansible**, es una herramienta que te va a permitir hacer tareas repetitivas en todas las máquinas de tu red de forma sencilla y segura.

UN EJEMPLO PRÁCTICO

Supongamos, que a ti te ha pasado como a mi, y empezaste comprando una *Raspberry*. Sin embargo, con el paso del tiempo, una sola se te quedaba corta, porque querías destinarla para unas tareas, pero querías tener otra para hacer pruebas. Compraste una segunda, e incluso, es posible que compraras una tercera, o como yo, que al final, entre unas y otras tengo una decena. De todos los modelos y colores.

Si ahora, quiero instalar un determinado servicio en cada una de las 10 Raspberry, ¿tengo que ir una

por una instalándolo?¿en serio?. A estas alturas del siglo XXI, ¿es la única solución?. No, para eso está precisamente **Ansible**.

Así, para actualizar todos tus servidores solo tendrías que utilizar la siguiente tarea en un *playbook*,

```
 name: Update apt-get repo and cache
apt: update_cache=yes force_apt_get=yes cache_valid_time=3600
```

¿Y SI SOLO TENGO UN SERVIDOR?

Pues incluso en el caso de que tengas un solo servidor o una sola Raspberry, utilizar Ansible es toda una ventaja. Porqué, te preguntarás... pues sencillo, porque instalar todos los paquetes, y dejar tu Raspberry o tu servidor, a punto, es tan fácil, como ejecutar un *playbook*, que no es mas que una lista de tareas.

De esta manera, cada vez que quieras poner a punto una determinada Raspberry, simplemente tienes que ejecutar ese *playbook*. Pero, no solo esto.

Existen determinadas tareas que siempre son iguales. Por ejemplo, levantar un servidor web con WordPress. Los pasos siempre son los mismos, y no solo esto, sino que al final se convierte en una tarea tediosa y repetitiva, y por supuesto, no exenta de posibles errores. Sin embargo, una vez completado tu **playbook**, tantas veces como quieras y siempre con garantias de éxito. De esta manera, puedes tener tu propia librería de *playbooks* con todas las operaciones mas habituales que sueles hacer.

Por ejemplo, en el caso que acabo de comentar, un *playbook* para montar un servidor web con **WordPress**, primero montarías el servidor *Nginx*, y luego seguirías con los distintos pasos. Pero fíjate que montar un servidor, es tan sencillo como ejectuar este *playbook*,

```
 ---
- hosts: all
  tasks:
    - name: ensure nginx is at the latest version
      apt:
        name: nginx
        state: latest
      become: yes
    - name: start nginx
      service:
        name: nginx
        state: started
      become: yes
```

Una vez instalado **Nginx**, realizarías la siguiente tarea, que la podrías tener en el mismo *playbook* o en otro distinto. Y así sucesivamente. Sencillo, seguro y práctico.

LAS VENTAJAS DE ANSIBLE

Ahora que ya tienes una idea de que es esto de **Ansible**, y empiezas a vislumbrar el potencial de esta herramienta, quiero hablarte sobre sus ventajas,

- Es **Open Source**
- Se trata de una herramienta que es **tremendamente sencilla de configurar y utilizar**. Esto lo

podrás comprobar en el siguiente capítulo de este tutorial, y realmente te va a sorprender.

- No necesitas un perfil de desarrollador para crear tus propios **playbooks**. En capítulos posteriores, te comentaré que esto de los *playbooks*, aunque con los ejemplos anteriores, ya te has podido hacer una idea.

- Con los ejemplos anteriores, ya has podido **percibir** la **potencia de esta herramienta**. Y te puedo asegurar, aunque me repita, que esto que has visto no es mas que la punta del iceberg, de las posibilidades de esta herramienta, con lo que imáginate cual es su **potencial**.

- Es **flexible**. Ansible te permite abstraerte de los entornos en los que trabajas y sobre los que trabajas.

- **No necesita agentes**. No necesitas instalar software o configurar puertos en los nodos que vas a gestionar o que quieres automatizar. Solo tienes que instalar **Ansible** en un equipo desde el que lo gestionarás todo.

- Es **eficiente**, por la misma razón indicada en el punto anterior, no necesitas instalar nada.

CARACTERÍSTICAS

A lo lardo este artículo he ido comentando algunas de las características de esta herramienta, sin embargo, permíteme que las condense en unos pocos puntos,

- **Gestión de la configuración**. Ansible está diseñado para ser sencillo, confialbe y consistente con la gestión de configuraciones. Las configuraciones de Ansible son sencillas descripciones, como tu mismo has podido comprobar en los diferentes ejemplos, que he mencionado a lo largo de este capítulo. Estas *descripciones* son perfectamente legibles y *parseables*,

- **Despliegue de aplicaciones**. Igualmente, en los distintos ejemplos, has podido comprobar lo sencillo que es *desplegar* (instalar y configurar) una aplicacón utilizando Ansible. No necesitas escribir instrucciones *ad-hoc* para cada uno de los equipos y dispositivos de tu sistema. Simplemente, enumera las tareas que hay que realizar, escribiendo tu *playbook*, y Ansible se encarga de conseguir que cada uno de tus dispositivos se *encuentre en el estado* que tu le has indicado.

- **Orquestación**. Esto de la orquestación seguro que lo escuchas en mas ocasiones a partir de ahora. El concepto es sencillo y te lo explico con un ejemplo. Una página web, implementada en WordPress, necesita de un servidor, con PHP, una base de datos. Es posible que si tu página web tiene mucha carga, necesites un balanceador. Pues Ansible se encarga no solo de desplegar todas estas aplicaciones y servicios, sino de organizarlos para que se relacionen correctamente entre si, es decir, de dirigir la función de **orquestar**.

- **Seguridad**. Una de las grandes ventajas de Ansible es la posibilidad de ejecutar el mismo *playbook* en todas las máquinas de tu ecosistema. Esto te va a permitir, por ejemplo, establecer **políticas de seguridad** idénticas para todos tus equipos, o al menos, según una clasificación que tu mismo puedes establecer. Esto además tiene la ventaja de que, una actualización de estas **reglas** sea sencilla de implementar.

Por último, señala la gran ventaja que te ofrecen los *playbooks*. Al ser sencillas instrucciones en *texto plano*. Esto te va a permitir llevar un control de versiones sobre estos *playbooks*, con todas las ventajas que esto conlleva.

Así, aunque simplemente quieras configurar una Raspberry, si lo haces con **Ansible** y sus *playbooks*, todo será mas sencillo. Por supuesto, que la primera vez no, la primera vez que lo intentes, no será

un camino de rosas. Sin embargo, una vez conseguido, **repetir** la instalación de un servicio o el despliegue de una aplicación, será sencillo y completamente confiable.

CONCLUSIÓN

Espero, que esta introducción haya servido para que veas las posibilidades que ofrece **Ansible** a la hora de desplegar un servicio o configurar una aplicación. En el siguiente capítulo comentaré los conceptos básicos de Ansible, que como verás son *poquitos*, y te ayudaré a instalar *Ansible* en tu propio equipo. De esta manera, podrás controlar tu *enjambre* de Raspberry o tus VPS distribuidos a lo largo y ancho de internet.

Más información,

- Documentación de Ansible
- SimpliLearn

Imagen de portada Franck V. 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

2.- YAML el reemplazo de JSON

3.- El inventario de Ansible

4.- Comandos ad-hoc

5.- Comandos básicos de Ansible

6.- Playbooks de Ansible

7.- Bucles y condicionales en Ansible

8.- Bloques en Ansible

9.- Handlers en Ansible

Vim, un editor atemporal
Diálogos para scripts

VIM, UN EDITOR ATEMPORAL

¿Porque aprender a utilizar Vim?¿Porque no conformarte con tus editores de texto o de código? ¿Porque ir un paso mas allá en el mundo de los editores?

DIÁLOGOS PARA SCRIPTS

Si quieres utilizar tu o quien tu quieras tus scripts de forma más cómoda y sin necesidad de terminal, puedes recurrir a los diálogos para scripts