HANDLERS EN ANSIBLE

Publicado el 7 de agosto de 2020 por atareao *33.5K* Deja un comentario

Este es **uno** de los capítulos del tutorial Automatización con Ansible. Una introducción.. Encontrarás los enlaces a todos los de capítulos, al final de este artículo.

Durante los anteriores capítulos del tutorial sobre Ansible, has visto distintas opciones para *modificar* el comportamiento de tu *playbook*, en función de terminadas condiciones de contorno. Por ejemplo, decidías ejecutar una determinada tarea o bloque de tareas si el *host* tenía un terminado sistema operativo. Pero, también puedes decidir ejecutar determinadas tareas si se producen cambios en tu *host* cuando se aplican determinadas tareas. Para hacer esto tienes los *handlers* en Ansible.

HANDLERS EN ANSIBLE

¿Pero que son los handlers en Ansible? Los *handlers* no son mas que tareas. Eso si, unas tareas que solo se realizarán en determinados casos. En concreto solo se realizarán cuando debido a una tarea **se produce un cambio** y se **realiza una notificación**. Lo cierto es que se tienen que cumplir ambas condiciones, tanto el cambio como la notificación.

INSTALANDO UN PAQUETE

Un claro ejemplo de handlers en Ansible es la instalación de un paquete. Por ejemplo, cuando instalas un servicio, puedes decidir iniciar el servicio una vez instalado. Así, por ejemplo, quieres instalar un servidor **Mumble**, y configurarlo, pero solo quieres reiniciarlo en el caso de que no esté ya instalado y por supuesto no está configurado, pero además solo quieres hacerlo una vez.

Es decir, no es necesario reiniciar el servidor **Mumble** en cada ocasión, solo tienes que reiniciarlo una vez, al final, cuando has terminado el resto de tareas. Y solo en el caso, de que las tareas hayan modificado el estado de tu servidor Mumble. Esto lo puedes hacer de la siguiente forma,

```
tasks:
 - name: install mumble server
   apt:
     name: mumble-server
     state: present
   notify:
     - restart mumble
 - name: configure mumble server
   lineinfile:
     path: /etc/mumble-server.ini
     regexp: '^#autobanAttempts\s*='
     line: autobanAttempts = 10
```

```
    notify:
      - restart mumble
handlers:
  - name: restart mumble
    service:
      name: mumble-server
      state: restarted
```

Si te fijas, al final de cada una de las tareas, tanto la de instalación como la de configuración, tiene una nueva acción `notify` seguida por un nombre `restart mumble`. Esta es la que se encarga de notificar el cambio a la tarea *handler* que tiene el mismo nombre, `restart mumble`.

Ahora bien, aunque ambas tareas, tanto la de instalación como la de configuración cambiaran de estado, y fuera notificado, el reinicio del servidor Mumble, solo ser produciría una vez. Por supuesto, esto lo puedes aplicar a mas cambios, y solo se aplicará un reinicio del servicio.

HANDLERS QUE ESCUCHAN

En el ejemplo anterior, la tarea **notifica** a un *handler* concreto. Sin embargo, a partir de la versión 2.2 de Ansible, los *handler* pueden ser *oyentes genéricos*.

¿A que me refiero con oyentes genéricos? Me refiero a que los *handlers de Ansible* tienen la capacidad de *ser activados* por un *mensaje*, y para ello utilizan `listen`.

Así, es posible reescribir el ejemplo anterior utilizando este concepto, de forma sencilla y mucho más práctica, desde mi punto de vista.

```
 tasks:
  - name: install mumble server
    apt:
      name: mumble-server
      state: present
    notify: "reiniciar servicios"
  - name: configure mumble server
    lineinfile:
      path: /etc/mumble-server.ini
      regexp: '^#autobanAttempts\s*='
      line: autobanAttempts = 10
    notify: "reiniciar servicios"
handlers:
  - name: restart mumble
    service:
      name: mumble-server
      state: restarted
    listen: "reiniciar servicios"
```

Esta solución tiene ventajas respecto a la anterior en tanto en cuanto que el cambio en una tarea puede **activar** varios handlers. Por otro lado se desacopla el nombre del handler.

En este caso se denomina **tópico** al mensaje que se utiliza para invocar a los handlers en Ansible.

ALGUNAS CUESTIONES IMPORTANTES

Ahora que ya has visto lo que son los *handlers en Ansible* y las posibilidades que te ofrecen a la hora de trabajar con tus playbooks, es importante que tengas algunas cuestiones en consideración.

- El nombre de los handlers en Ansible debe ser único. En el caso de existir dos handler con el mismo nombre, se ejecutará el último de ellos.
- No debes utilizar variables en el nombre de un handler.
- Los handlers en Ansible se llaman al final del playbook por defecto. De esta manera, aunque se invoque por notificación a un handler en repetidas ocasiones, solo se ejecutará una vez.
- Es interesante el uso de `listen` cuando quieres invocar a varios handlers de forma simultánea con una sola notificación.
- El nombre del handler y del tópico se definen de forma global.

Imagen de portada de Rohan Makhecha en Unsplash

Listado de capítulos

Estos son todos los capítulos del tutorial Automatización con Ansible. Una introducción.,

Diálogos para KDE con kdialog
Zenity, diálogos para GNOME , Cinnamon, MATE...

DIÁLOGOS PARA KDE CON KDIALOG

En este capítulo del tutorial te muestro como crear diálogos para KDE que se integrarán perfectamente con tu escritorio KDE Plasma y con otros.

ZENITY, DIÁLOGOS PARA GNOME , CINNAMON, MATE...

Zenity es una herramienta perfecta para crear diálogos para tus scripts en Bash o otro lenguaje y que el usuario no recurra al terminal

Buscar

aKademy-es 2020

Del 20 al 22 de Noviembre

¡No te la pierdas!

YouTube

iVoox

Spotify

Apple podcast

Google podcast

Telegram

Twitter

Mastodon

GitHub

Facebook

Suscribir

Contactar

Quién soy