

ANSIBLE



TRABAJO TEÓRICO

Gestión y Administración de Red

Diego Dorado Galán

Tabla de contenidos

| | |
|---|----|
| INTRODUCCIÓN | 3 |
| QUÉ ES ANSIBLE..... | 3 |
| REQUISITOS | 4 |
| VERSIONES..... | 4 |
| VENTAJAS RESPECTO A PUPPET | 5 |
| LISTA DE PARÁMETROS DE ENTRADA | 5 |
| LISTA DE FORMA DE ACCEDER AL SERVICIO..... | 6 |
| LISTA DE CONTROLADORES DE MONITORIZACIÓN EN ANSIBLE | 7 |
| ¿GENERA EVENTOS Y/O NOTIFICACIONES? | 8 |
| MÉTODOS DE INSTALACIÓN | 8 |
| REDUNDANCIA EN ANSIBLE | 8 |
| DEMOSTRACIÓN | 9 |
| REFERENCIAS..... | 10 |
| BIBLIOGRAFÍA..... | 10 |
| TUTORIALES RECOMENDADOS..... | 11 |

INTRODUCCIÓN

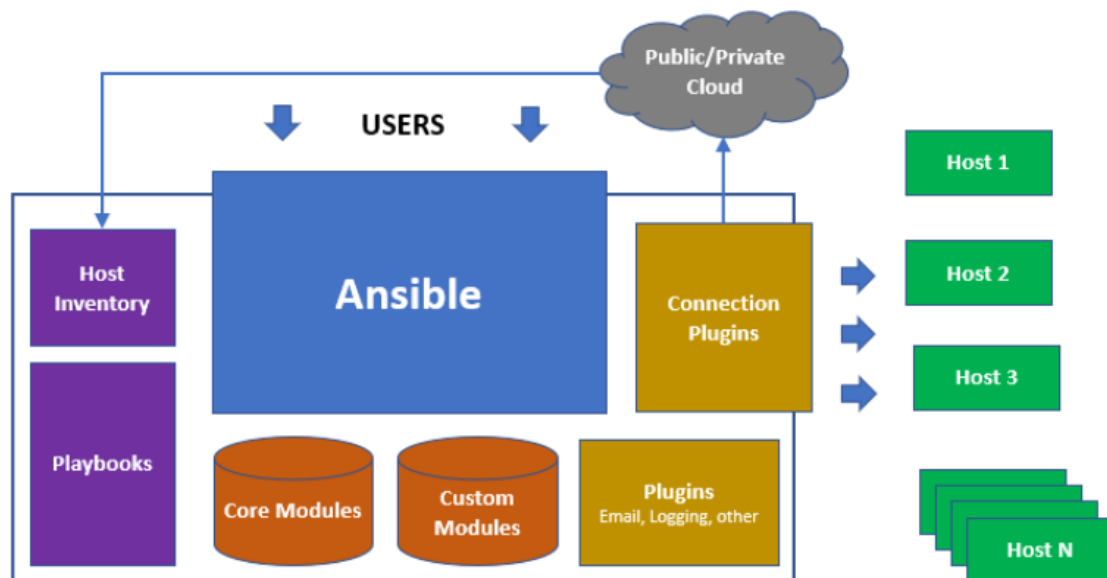
QUÉ ES ANSIBLE

Ansible es un software de automatización del aprovisionamiento de software, gestión de configuración y despliegue de aplicaciones. Básicamente es una herramienta de orquestación, posiblemente la más utilizada hoy en día. Permite gestionar servidores, configuraciones y aplicaciones de forma sencilla, robusta y paralela.

(Pérez, s.f.)

(Documentación Ansible, s.f.)

Un esquema del funcionamiento de la aplicación es el siguiente:



(Avi, s.f.)

El elemento principal de ansible es el playbook (formato YAML), conjunto de tareas organizadas con un fin. A partir de un fichero de inventario, dónde indicaremos los distintos nodos o grupos de nodos, determinaremos un estado deseado para estas, mediante las tareas definidas previamente en un playbook, que generalmente se apoyan en el uso de roles para lograr una lógica más compleja y ordenada.

Un elemento fundamental de Ansible es la idempotencia.

REQUISITOS

Ansible es compatible con múltiples sistemas operativos, como Linux (donde realizaré la demostración), Windows (aunque su instalación es más tediosa), macOS, Solaris, IBM AIX, etc.

También es compatible con distintos proveedores cloud como AWS, Azure, OpenStack o Google Cloud Platform.

Para mi demostración usaré tres máquinas Ubuntu 22.04.2 LTS, una como nodo orquestador o central (servicio) y dos como nodos (cliente y cliente-2).

Ansible se basta de distintas librerías de terceros, aunque únicamente mencionaré aquellas que he usado (directa o indirectamente):

- Jinja2, biblioteca de templates usada para crear configuraciones y archivos de configuración.
- PyYAML, biblioteca Python que usa Ansible para comprobar la sintaxis de ficheros YAML (--syntax).
- Paramiko, usada para realizar las conexiones SSH con los nodos remotos.

Ansible se ayuda también de SSH, Git y Python (lenguaje con el que está escrito Ansible y con el que se ejecuta), por lo que deberán estar instaladas previamente en los sistemas.

VERSIONES

Principalmente existen dos tipos de productos:

- Ansible Engine, el usado en la demo y ejecutado desde la Shell. Gratuito y open-source.
- Ansible Tower, también conocida como Red Hat Ansible Tower, es una interfaz web que ayuda en la gestión de automatización y orquestación de una forma más gráfica y que es un producto comercial. Su precio depende del número de nodos gestionados y soporte (suscripción anual).
- Ansible Galaxy, que realmente es otra extensión de Ansible que ofrece un repositorio público de funciones y módulos Ansible.

(Avi, s.f.)

VENTAJAS RESPECTO A PUPPET

Ventajas:

- Configuración no basada en agentes, lo que facilita la configuración y la escalabilidad, ya que no requiere ningún software adicional en los nodos.
- Flexibilidad, compatible con un gran número de sistemas operativos y dispositivos de red.
- Uso generalizado, lo que favorece a una comunidad mayor, siendo así más fácil la capacidad de aprendizaje y documentación.
- Escalabilidad

LISTA DE PARÁMETROS DE ENTRADA

Los únicos parámetros fundamentales que introducir en Ansible son la clave SSH utilizada en el nodo remoto, el inventario donde se recopilan los nodos o grupos de nodos y el playbook en cuestión que queremos ejecutar. A continuación listo un conjunto de parámetros usados frecuentemente:

| PARÁMETRO | VALOR POR DEFECTO | SIGNIFICADO | OBSERVACIONES |
|------------------------|-------------------|--|--|
| Clave SSH (id_rsa.pub) | none | Especificar la clave SSH para acceder a un servidor remoto. | Intentará usar la clave por defecto almacenada en <code>~/.ssh/id_rsa</code> . Se especifica mediante el parámetro <code>--key-file</code> |
| Inventario | - | Ruta al archivo de inventario para los hosts | <code>'-m'</code> o <code>'--module-name'</code> . |
| playbook | - | Archivo que queremos ejecutar en el host remoto | <code>ansible-playbook <nombre></code> |
| Clave root | False | Si se necesitan permisos de root Ansible nos pedirá introducir la clave del host remoto. | Normalmente utilizo <code>-K</code> para introducir la contraseña por terminal, pero se puede introducir como parámetro mediante <code>'-k'</code> o <code>'-e'</code> . |
| Variables adicionales | - | Variables adicionales | <code>'--extra-vars'</code> o <code>'-e'</code> |

| | | | |
|---------------------|-------|---|-------------------|
| Convertirse en root | False | Para realizar algunas operaciones (como actualizar o crear usuarios) debemos estar identificados como root. | '-b' o '--become' |
|---------------------|-------|---|-------------------|

Estos son los parámetros que he utilizado de forma más o menos recurrente, pero únicamente cuando realizo operaciones ad-hoc. El único parámetro que utilizo siempre es -KK, para realizar operaciones como root.

La clave SSH y el inventario es fundamental para ejecutar cualquier playbook, pero una buena práctica es indicarlo en el archivo de configuración para que se indiquen por defecto:

```
GNU nano 6.2          ansible.cfg
[defaults]
inventory = inventario
private_key_file = ~/.ssh/ansible
```

LISTA DE FORMA DE ACCEDER AL SERVICIO

La forma de establecer una conexión entre la máquina de control de Ansible y los hosts destino para así ofrecer el servicio es generalmente mediante SSH. Utiliza el protocolo SSH para establecer una conexión segura y asegurar la autenticidad, permitiendo además ejecutar comandos de forma remota.

Otras opciones son usar distintos connection plugin, como los de Docker y Kubernetes, útiles para la gestión de aplicaciones en contenedores y clúster respectivamente. Para sistemas Windows, el plugin WinRM utiliza HTTPS para establecer la conexión.

También es posible mediante Telnet (poco seguro) o acceso local (utilizado generalmente en labores de mantenimiento).

LISTA DE CONTROLADORES DE MONITORIZACIÓN EN ANSIBLE

Ansible tiene integrados algunos mecanismos o herramientas para la monitorización de la actividad:

- **Logs de los playbook ejecutados:** al lanzar un playbook, en consola observamos un listado de los registros de las tareas y resultados, pudiendo ser útiles para monitorizar la actividad y depurar.

```
PLAY [all] *****
TASK [Gathering Facts] *****
ok: [192.168.1.134]
TASK [install updates (Ubuntu)] *****
changed: [192.168.1.134]
PLAY [subred1] *****
TASK [Gathering Facts] *****
ok: [192.168.1.134]
TASK [install kubernetes y docker] *****
ok: [192.168.1.134]
PLAY [subred2] *****
skipping: no hosts matched
PLAY RECAP *****
192.168.1.134 : ok=4  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

- **Ansible Stats:** módulo capaz de recopilar estadísticas sobre el flujo de ejecución de un playbook, añadiendo tiempos de ejecución, número de tareas realizadas, etc. Se pueden añadir al archivo de configuración para que se muestren por defecto.
- **Ansible Tower /AWX:** plataforma mencionada anteriormente que nos permiten realizar un seguimiento y monitorización de una forma más gráfica e ilustrativa mediante el uso de dashboards.

Además, tiene posibilidad de utilizar otras herramientas para la monitorización, como Grafana, Prometheus, Nagios o Zabbix.

En un enlace de la bibliografía encontramos un post donde se trabaja con Prometheus, Grafana, Node Exporter y Ansible Tower para la monitorización:

¿GENERA EVENTOS Y/O NOTIFICACIONES?

Prácticamente todas las utilidades de generación de eventos y notificaciones se incorporan mediante módulos, es decir mediante task pluggins.

En lo que respecta a las notificaciones, si que existen distintos módulos o templates incorporados en Ansible Tower (versión comercial), siendo capaz de enviar distintos tipos de notificación como correos, con información y descripción relativa.

Ansible-rulebook es un software de automatización basado en eventos para Ansible. Tiene características como conectarse a flujos de eventos y manejarlos en tiempo real, ejecutar plantillas de Ansible Tower condicionalmente en función de la coincidencia de reglas con eventos. Adjunto en la bibliografía la documentación oficial con sus principales funcionalidades.

(Ansible-rulebook, s.f.)

MÉTODOS DE INSTALACIÓN

Generalmente se puede instalar mediante el gestor de paquetes del sistema operativo. En Linux se puede instalar mediante apt y en macOS mediante Homebrew (brew).

Al ser un programa escrito en Python también se puede instalar mediante pip.

La instalación en Windows (como ya mencioné) puede ser algo más tediosa. Se puede instalar con el subsistema de Ubuntu para Windows o mediante herramientas como Cygwin.

También se puede instalar en distribuciones algo más desfasadas mediante RPM o mediante Docker.

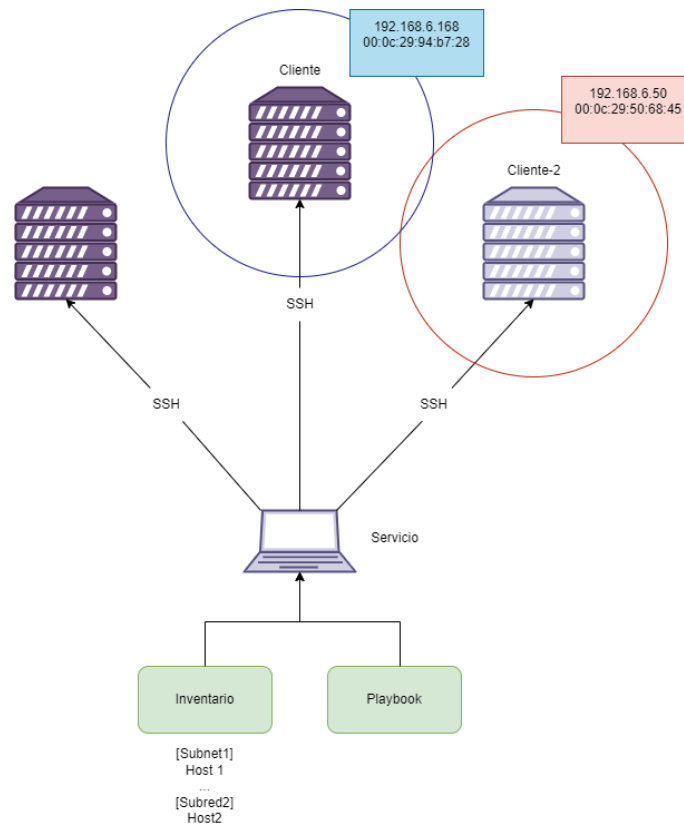
REDUNDANCIA EN ANSIBLE

Casi todas las funcionalidades de redundancia explícita solo están disponibles para Ansible Tower, ya que ejecuta distintas instancias en caso de que alguna falle. Esto está más detallado en un enlace en la bibliografía.

Si bien es cierto también se puede lograr cierta redundancia configurando múltiples nodos que ejecuten las mismas tareas. También se pueden usar funciones o módulos para implementar un LoadBalancer, que distribuya la carga para mejorar la redundancia.

DEMOSTRACIÓN

El diagrama del entorno virtual utilizado para mi demostración podría ser el siguiente:



Como ya he mencionado, mi demo se basará en una máquina Vmware que actuará como servicio y que constará de una imagen Ubuntu 22.04.02 LTS. Esta máquina será el nodo de control de la arquitectura diseñada, y que automatizará la instalación y puesta en marcha del software y configuración de los nodos cliente (Cliente y Cliente-2), que también usaran una imagen Ubuntu 22.04.02 LTS.

La comunicación en Ansible se realiza generalmente mediante el protocolo SSH en el puerto 22. Por ello, debemos generar un par de claves SSH y copiarlas en los nodos remotos.

El guion que seguiré en la presentación, que incluye también los casos de usos requeridos, está disponible en /doc/GUION_PRESENTACION.md, en el siguiente repositorio de GitHub.

https://github.com/23kdt/GAR_Ansible.git

REFERENCIAS

Ansible-rulebook. (s.f.). Obtenido de Documentation: <https://ansible-rulebook.readthedocs.io/en/stable/>

Avi. (s.f.). *GeekFlare*. Obtenido de Ansible para principiantes: conceptos básicos de Ansible y cómo funciona: <https://geekflare.com/es/ansible-basics/>

Avi. (s.f.). *GeekFlare*. Obtenido de What is Difference Between Ansible, Ansible Galaxy, and Ansible Tower?: <https://geekflare.com/es/ansible-basics/>

Documentación Ansible. (s.f.). Obtenido de https://docs.ansible.com/ansible/latest/getting_started/index.html

Leonardo Araujo. (6 de Mayo de 2020). *RedHat Ansible*. Obtenido de Red Hat Ansible Tower Monitoring: Using Prometheus + Node Exporter + Grafana: <https://www.ansible.com/blog/red-hat-ansible-tower-monitoring-using-prometheus-node-exporter-grafana>

Pérez, M. (s.f.). *OpenWebinars*. Obtenido de Qué es Ansible: <https://openwebinars.net/blog/que-es-ansible/>

BIBLIOGRAFÍA

[Ejemplo de módulo para implementar notificaciones](#)

[GitHub Ansible-rulebook](#)

[Monitorización mediante Ansible Tower + Node Exporter + Grafana + Prometheus](#)

[Instalación de Ansible en Windows](#)

[Instalación de Ansible AWX mediante Docker en CentOS7](#)

[Rdundancia en Ansible Tower](#)

[Cómo crear un balanceador de carga en Ansible](#)

TUTORIALES RECOMENDADOS

- [Primeros pasos con Ansible](#)
- [Roles de Ansible](#)
- [Getting started with Ansible \(muy útil\)](#)
- [Módulo usuario. Crear, eliminar y administrar agentes.](#)