Ansible 101

"Los fundamentos de la automatización"
#EducaciónPúblicaSiempre - UNAHUR - 2024

main() →

Santiago Javier Said

DevOps Engineer @ Eynes

ssaid santiagojaviersaid said.santiago@gmail.com

Python · Odoo · Debian · Vim · Gitlab · Github · Docker · Kubernetes · Ansible

¿Cómo administrarías cientos o miles de dispositivos a la vez?

¿Cómo podrías armar un servidor idéntico a otro?

¿Quienes lo usan?

NASA, IBM, Verizon, Atlassian y yo 😎 confiamos en Ansible para gestionar infraestructura y automatizar tareas, mejorando la eficiencia, reduciendo errores humanos, y gestionando entornos de gran escala con facilidad.

Que es ansible?

Es una herramienta de automatización de código abierto que nos permitirá gestionar dispositivos de manera eficiente, reduciendo errores y ahorrando tiempo en tareas repetitivas.

- 📝 **Declaratividad** - Usa un lenguaje declarativo para describir configuraciones (YAML)
- 🎨 **Idempotencia** - Las tareas solo se ejecutan si es necesario
- 🤹 **Escalabilidad** - Puede gestionar cientos o miles de dispositivos simultáneamente
- ✨ **Flexibilidad** - Podemos utilizarlo para automatizar una amplia variedad de tareas
- 📦 **Modularidad** - Módulos que nos permiten gestionar distintos aspectos de los dispositivos
- 🌎 **Comunidad** - Módulos, roles y colecciones disponibles en https://galaxy.ansible.com/
- 🛠️ **Inventarios estáticos o dinámicos** - Definición de hosts estándar o cloud
- 👮 **Agentless** - No requiere instalar agentes en los servidores gestionados
- 🛜 **Pull Mode** - Podemos gestionar dispositivos que no tenemos acceso ssh mediante *ansible-pull*
- 🔏 **Gestión de secretos** - Cifrado de archivos sensibles como passwords o claves ssh mediante *ansible-vault*

La magia de ansible

Gracias a estos principios nos aseguramos que las configuraciones siempre lleven los sistemas a un estado deseado de manera eficiente y sin repetir acciones innecesarias, reduciendo errores y garantizando consistencia en cada ejecución.

Idempotencia

Ansible asegura que un sistema alcance un estado deseado, solo realizando cambios si es necesario.

Lenguaje Declarativo

Ansible describe "qué" hacer (resultado final) en lugar de "cómo" hacerlo (pasos detallados).

Arquitectura de Ansible

Concepto	Definición
Control Node	El servidor donde se ejecuta Ansible.
Managed Nodes	Los dispositivos que se gestionan, los cuales no requieren agentes instalados.
Inventarios	Listas de hosts o dispositivos que Ansible gestionará.
Módulos	Componentes que ejecutan tareas específicas en los Managed Nodes.
Playbooks	Conjuntos de tareas organizadas en formato YAML.
Roles	Conjuntos de tareas, variables y archivos reutilizables.
Colecciones	Paquetes que agrupan roles, módulos, plugins y playbooks.

Inventario Estático

🛰️ Defición los hosts o dispositivos que serán gestionados.
⚠ `/etc/ansible/hosts`, `/etc/ansible/ansible.cfg`, `~/.ansible.cfg`

```
[grupo]
# alias ansible_host=<IP o hostname> ansible_user=<usuario SSH>

[all:vars]
ansible_ssh_private_key_file=/path/to/your/private/key

[webservers]
server1 ansible_host=192.168.1.10 ansible_user=ubuntu
server2 ansible_host=192.168.1.11 ansible_user=ubuntu

[dbservers]
db1 ansible_host=192.168.1.20 ansible_user=root
db2 ansible_host=192.168.1.21 ansible_user=root

[all_servers:children]
webservers
dbservers
```

Como saber los hosts disponibles?

```
ansible-inventory [-i inventory_file] --graph
```

Inventario Dinámico

☁ Idem IE con la diferencia que consulta una API o servicio para obtener una lista actualizada de los hosts en tiempo real.

```
# aws_ec2.yml
plugin: amazon.aws.aws_ec2
regions:
  - us-east-1
filters:
  instance-state-name: running
keyed_groups:
  - key: tags.Name
    prefix: tag_Name_
strict: False
```

```
#Run AdHoc Command
ansible all -i aws_ec2.yml -m ping
# Run Playbook
ansible-playbook -i aws_ec2.yml playbook.yml
```

*Consulta las instancias de AWS EC2 en la región us-east-1 que están en estado "running" y las agrupa según sus etiquetas de nombre (tags.Name).

Módulos: ¿Qué son y cómo usarlos?

- Los módulos son las unidades básicas de ejecución de tareas. Mayormente estan escritos en Python.
- Ansible incluye cientos de módulos para realizar diversas tareas como gestión de usuarios, servicios, etc.

```
ansible webservers -i inventory.cfg -m apt      -a "name=nginx state=present" --become --ask-become-pass
#       ^Grupo        ^ Inventario     ^ Módulo     ^ Argumentos              ^ Ejecutar con privilegios elevados
```

*Instalando nginx con un comando ad-hoc

> Los comandos adhoc son una forma rápida de ejecutar tareas sin necesidad de un playbook.

```yaml
# playbook.yaml
- name: Instalar Nginx
  hosts: webservers
  become: yes
  tasks:
    - name: Instalar paquete Nginx
      apt:
        name: nginx
        state: present
```

```
ansible-playbook -i inventory.cfg playbook.yml --ask-become-pass`
```

*Instalando nginx con un playbook

Playbooks

```yaml
- name: Playbook para instalar y configurar Nginx
  hosts: servidores_web
  become: yes # Escalar privilegios
  tasks:
    - name: Instalar Nginx
      apt:
        name: nginx
        state: present
        update_cache: yes
    - name: Copiar archivo de configuración de Nginx
      copy:
        src: /ruta/local/nginx.conf
        dest: /etc/nginx/nginx.conf
      notify:
        - Reiniciar Nginx
  handlers:
    - name: Reiniciar Nginx
      systemd:
        name: nginx
        state: restarted
```

```
ansible-playbook -i inventory.cfg playbook.yml --ask-become-pass
```

Roles: construyendo un rol propio

> Un rol en Ansible es una estructura organizada que agrupa *tareas, archivos, plantillas, variables y handlers* en directorios predefinidos.

```
$ ansible-galaxy role init myrol
roles/myrol
├── defaults
│   └── main.yml # Variables por defecto
├── files        # Archivos a ser deployados
├── handlers
│   └── main.yml # Handlers a ser usados por el rol
├── meta
│   └── main.yml # Info del rol y dependencias
├── tasks
│   └── main.yml # Tareas a ejecutar
├── templates    # Jinja2 templates a ser usados
└── vars
    └── main.yml # Variables que utiliza el rol
```

```yaml
---
- hosts: webservers
  roles:
    - role: roles/myrol
      variable1: true
```

Roles: reutilizando roles de la comunidad

> Ganaremos eficiencia al reutilizar configuraciones probadas y optimizadas, desarrolladas por expertos y la comunidad."

```
ansible-galaxy role install geerlingguy.docker [--force]
                                               # ^ Update to last version
```

```
# Install Docker (play level)
- hosts: all
  become: true
  roles:
    - role: geerlingguy.docker
      vars:
        docker_users:
          - vagrant
```

```
# Install Docker (tasks level)
- name: Install docker
  ansible.builtin.include_role:
    name: geerlingguy.docker
    apply:
      become: true
  vars:
    docker_users:
      - 'vagrant'
```

Colecciones: reutilizando paquetes de la comunidad

> Las colecciones permiten distribuir y compartir conjuntos completos de automatizaciones, facilitando la reutilización y manteniendo todo lo necesario para realizar una tarea o implementar una solución dentro de un único paquete.

```
ansible-galaxy collection install community.docker [--upgrade]
                                                   # ^ Update to last version
```

```yaml
# Run a docker container
---
- name: Correr un container de docker
  hosts: app1
  become: yes
  tasks:
    - name: Ejecutar el contenedor de Nginx
      community.docker.docker_container:
        name: nginx_server
        image: nginx:latest
        state: started
        ports:
          - "80:80"
        volumes:
          - /tmp/mount:/tmp/mount
```

La comunidad

La comunidad de Ansible Galaxy es fundamental para acelerar la automatización, ofreciendo miles de roles y colecciones reutilizables, que permiten implementar soluciones probadas y escalables de forma rápida y eficiente.

Queremos evitar "reinventar la rueda".

Welcome to Galaxy

Filter by keywords

Download

To be able to download content from galaxy it is required to have ansible-core>=2.13.9

Please, check it running the

Share

Help other Ansible users by sharing the awesome roles and collections you create.

Maybe you have automation for

Featured

Ansible Lightspeed

Generative AI, the Ansible way.
Try Ansible Lightspeed with
IBM watsonx Code Assistant

Gestión de Secretos (ansible-vault)

> Nos permite gestionar archivos sensibles de manera segura. Creamos un archivo que contenga variables que serán encriptadas.

1. Creamos archivo que contenga nuestra información sensible

```
ansible-vault create secrets.yml
# a) Se solicitará una password para la encriptación
# b) Se define la variable db_password: "mi_contraseña_secreta"
```

2. Usar el archivo en un playbook

```
---
- name: Playbook para configurar un servidor
  ...
  vars_files:
    - secrets.yml
  tasks:
    - name: Configurar base de datos con contraseña
      postgresql_user:
        name: "usuario_db"
        password: "{{ db_password }}"
```

3. Ejecutar el playbook con las variables definidas

```
ansible-playbook -i inventario playbook.yml --ask-vault-pass
```

Escalabilidad: utilizando forks

> El concepto de forks en Ansible está relacionado con su capacidad de ejecutar tareas en paralelo en múltiples servidores o dispositivos, lo que es esencial para gestionar grandes infraestructuras de manera eficiente.

1. El controlador (la máquina que ejecuta Ansible) crea forks, para manejar conexiones simultáneas.
2. Cada fork se encarga de ejecutar las tareas en un solo host.

```
# ansible.cfg
[defaults]
forks = 5  # Default
```

```
# Ejecutamos un playbook en 20 hosts a la vez
ansible-playbook playbook.yml -f 20
```

⚠ El número óptimo de forks depende de la capacidad de la máquina que ejecuta Ansible (CPU, memoria, ancho de banda)

Agentless Mode

> Ansible es agentless porque simplemente se conecta a los servidores a través de SSH (o WinRM para Windows) y ejecuta las tareas necesarias sin que sea necesario instalar software adicional en esos servidores.

1. **Copia de módulos**: Ansible envía scripts temporales al host.
2. **Ejecución en el host**: El host ejecuta los módulos localmente.
3. **Resultados**: Devuelve el estado de ejecución a la máquina controladora.
4. **Limpieza**: Elimina módulos temporales del host.

Pull Mode

> Los hosts gestionados son los que "extraen" (pull) las configuraciones y ejecutan los playbooks localmente.
> Esta modalidad es útil en entornos donde los servidores no tienen acceso SSH desde la máquina controladora.

```
ansible-pull -U https://github.com/usuario/repo-playbooks.git
```

🕑 Se puede agregar al crontab para que se ejecute periódicamente

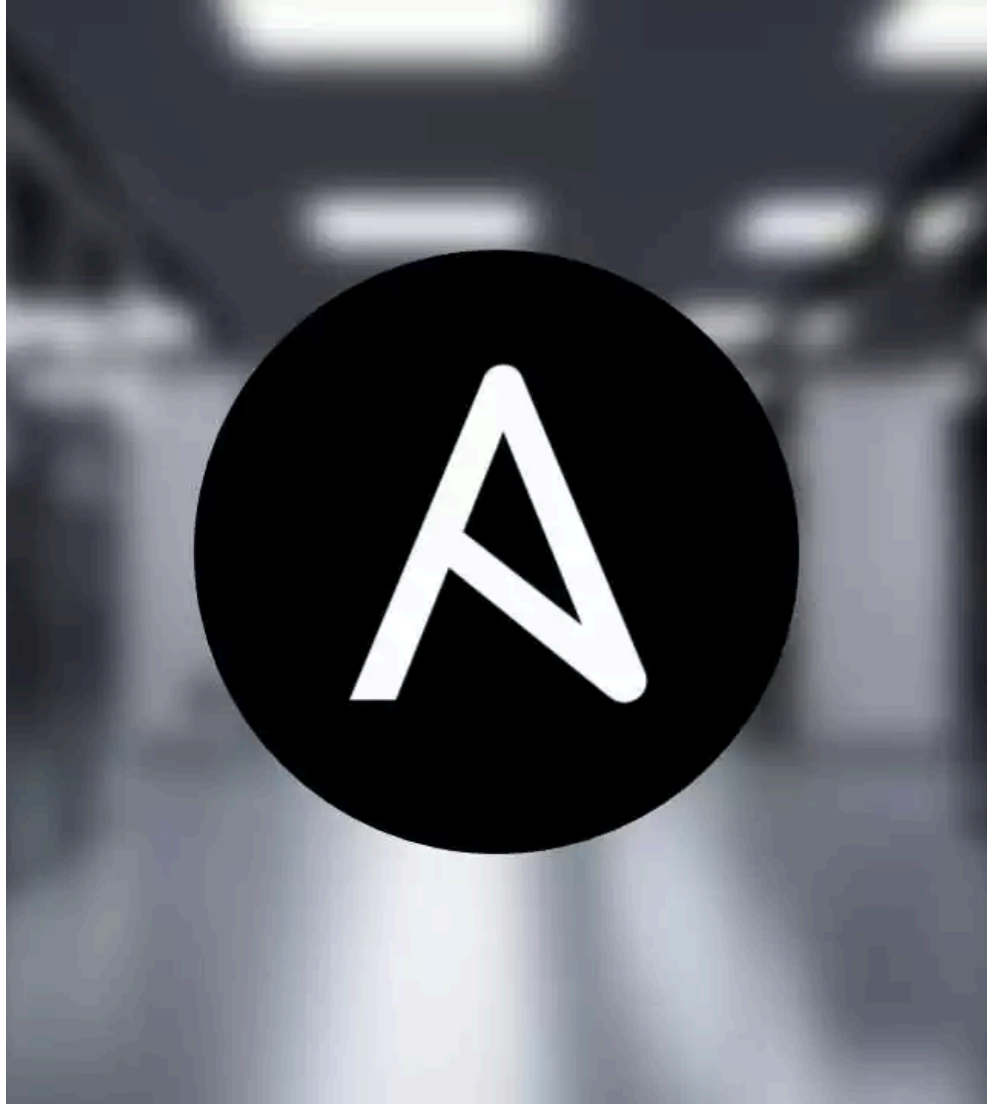
Instalación

Se recomienda usar pipx ✨

> *pipx* crea un ambiente aislado para cada aplicación y sus paquetes asociados.

```
pipx install --include-deps ansible
```

```
➜  ~ pipx list
venvs are in /home/ssaid/.local/pipx/venvs
  package ansible 9.1.0, 3.11.2
    - ansible
    - ansible-community
    - ansible-config
    - ansible-connection
    - ansible-console
    - ansible-doc
    - ansible-galaxy
    - ansible-inventory
    - ansible-playbook
    - ansible-pull
    - ansible-test
    - ansible-vault
```



¡Gracias!