

Informació:

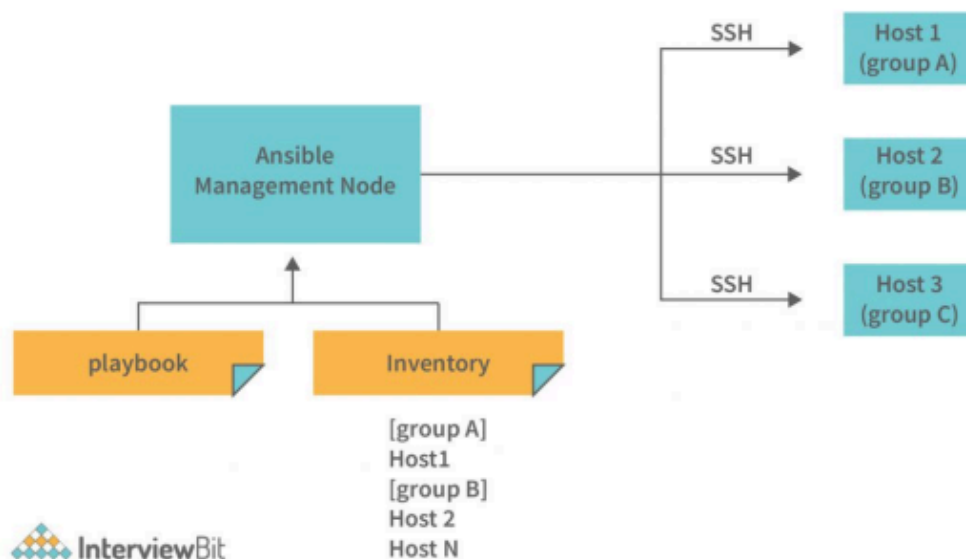
Agent-less: Baix manteniment: No requereix instal·lar cap software a la infraestructura a controlar.

Simplicitat: Els Playbook (fitxers d'automatització) s'escriuen en format YAML. Ansible fa servir les credencials SSH existents per accedir a les màquines remotes.

Escalabilitat i flexibilitat: Facil escalabilitat dels sistemes gràcies a l'utilització d'un gran nombre de mòduls específics pero diferents sistemes operatius, plataformes cloud i dispositius de xarxa.

Idempotencia i predictabilitat: Quan el sistema es troba en l'estat que descriu el Playbook, Ansible no canvia res, fins i tot si el Playbook s'executa diverses vegades.

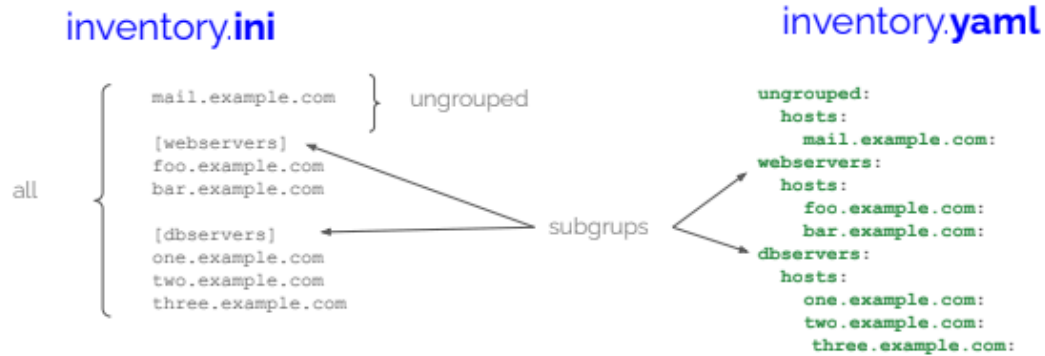
Arquitectura



Els **inventaris** organitzen nodes gestionats en fitxers centralitzats que proporcionen a Ansible informació del sistema..

Mitjançant un fitxer d'inventari, Ansible pot gestionar un gran nombre de nodes amb una sola ordre.

Inventory: 2 formats possibles



Grups per defecte de l'Inventory:

all: Inclou tots els hosts de l'inventory.
Ungrouped: Inclou tots els que no estiguin dintre d'un grup.

Variables d'entorn i fixers

Variables

`ANSIBLE_INVENTORY` – Override the default ansible inventory file

`ANSIBLE_LIBRARY` – Override the default ansible module library path

`ANSIBLE_CONFIG` – Override the default ansible config file

Files

`/etc/ansible/hosts` – Default inventory file

`/etc/ansible/ansible.cfg` – Config file, used if present

`~/ansible.cfg` – User config file, overrides the default config if present

Ad-hoc commands

```
$ ansible [pattern] -m [module] -a "[module options]"
```

```
ansible all -a "ls -l /etc/" -i inventory.ini
```

Llista màquines a inventory.ini

```
ansible all -m apt -a "name=iotop state=present" -b -i inventory.ini
```

Installa iotop

```
ansible all -m apt -a "name=glances state=present" -b -i inventory.ini
```

Installa glances

```
ansible all -m script -a "ruta/al/script.sh"
```

Executa script.sh

Ad-hoc commands

Reboot servers

```
$ ansible [pattern] -m [module] -a "[module options]"
```

```
$ ansible all -a "/sbin/reboot"
```

10 servers simultanís.
Default 5

Opera com a **usuari**

Opera com a **root** a tots els
hosts de l'inventari, i
demana el password

```
$ ansible all -a "/sbin/reboot" -f 10 -u username --become [--ask-become-pass]
```

Operacions amb fitxers

```
$ ansible all -m ansible.builtin.copy -a "src=/etc/hosts dest=/tmp/hosts"
```

copy

chmod

Chmod i chown

```
$ ansible webserver -m ansible.builtin.file -a "dest=/srv/foo/a.txt mode=600"
```

```
$ ansible webserver -m ansible.builtin.file -a "dest=/srv/foo/b.txt mode=600 owner=mdehaan group=mdehaan"
```

```
$ ansible webserver -m ansible.builtin.file -a "dest=/path/to/c mode=755 owner=mdehaan group=mdehaan state=directory"
```

```
$ ansible webserver -m ansible.builtin.file -a "dest=/path/to/c state=absent"
```

rmdir

mkdir

```
$ ansible all -m copy -a "content=foo dest=/root/bar.txt" -C
```

Check mode = Dry run mode = **simulació**

Inventory

```
$ ansible-inventory -i inventory.ini --list
```

```
$ ansible all -m ping -i inventory.ini -u isard
```

https://docs.ansible.com/ansible/latest/command_guide/command_line_tools.html

Ansible: Instal·lació

Al node de control d' Ansible SSH1:

Creeu un inventari al directori Ansible que heu creat al pas anterior:

```
Touch ~/git/ansible/inventory.ini
```

Afegim nou grup **[myhosts]** al fitxer **inventory.ini** i especifiqueu l'adreça **IP** o **el nom de domini complet (FQDN)** de cada sistema remot

```
inventory.ini
[hosts]
172.26.0.2
172.26.0.3
172.26.0.4
```

Creem **3 playbooks diferents** al node de control d'ANSIBLE SSH1:

<pre>playbook1.yaml --- - name: El meu primer playbook hosts: hosts become: true tasks: - name: Ping my hosts ansible.builtin.ping: - name: Print message ansible.builtin.debug: msg: Hello world</pre>	<pre>playbook2.yaml --- - name: update i upgrade hosts: all become: true tasks: - name: Update apt cache apt: update_cache: yes - name: Upgrade all packages apt: upgrade: dist autoremove: yes autoclean: yes</pre>	<pre>playbook3.yaml --- - name: iotop i glances hosts: hosts become: true tasks: - name: Install iotop apt: name: iotop state: present - name: Install glances apt: name: glances state: present</pre>
--	---	---

Als nodes gestionats:

Us heu d'assegurar que la vostra **clau SSH pública de SSH1** s'afegeixi al fitxer **authorized_keys** de cada node remot.

Verificar que es pot entrar als nodes remots via SSH sense password!!

Desde el node de control SSH1:

Verifiqueu el vostre inventari:

```
ansible-inventory -i inventory.ini --list
```

Feu ping al grup myhosts al vostre inventari (Passeu l'opció -u amb l'ordre ansible si el nom d'usuari és diferent al node de control d'Ansible i al sistema remot.)

```
ansible hosts -m ping -i inventory.ini -u isard
```

```
ansible-playbook -i inventory.ini playbook1.yaml -l hosts --ask-become-pass
```

Ansible retorna:

```
PLAY [El meu primer playbook] *****
TASK [ping my hosts] *****
ok: [172.26.0.2]
ok: [172.26.0.3]
ok: [172.26.0.4]

TASK [Print message] *****
ok: [172.26.0.2] => {
  "msg": "Hello world"
}
ok: [172.26.0.3] => {
  "msg": "Hello world"
}
ok: [172.26.0.4] => {
  "msg": "Hello world"
}

PLAY RECAP *****
172.26.0.2: ok=3  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

→ **playbook2.yaml**

```
---
- name: El meu primer playbook
  hosts: hosts
  become: true
  tasks:
    - name: Ping my hosts
      ansible.builtin.ping:

    - name: Print message
      ansible.builtin.debug:
        msg: Hello world
```

Exemples:

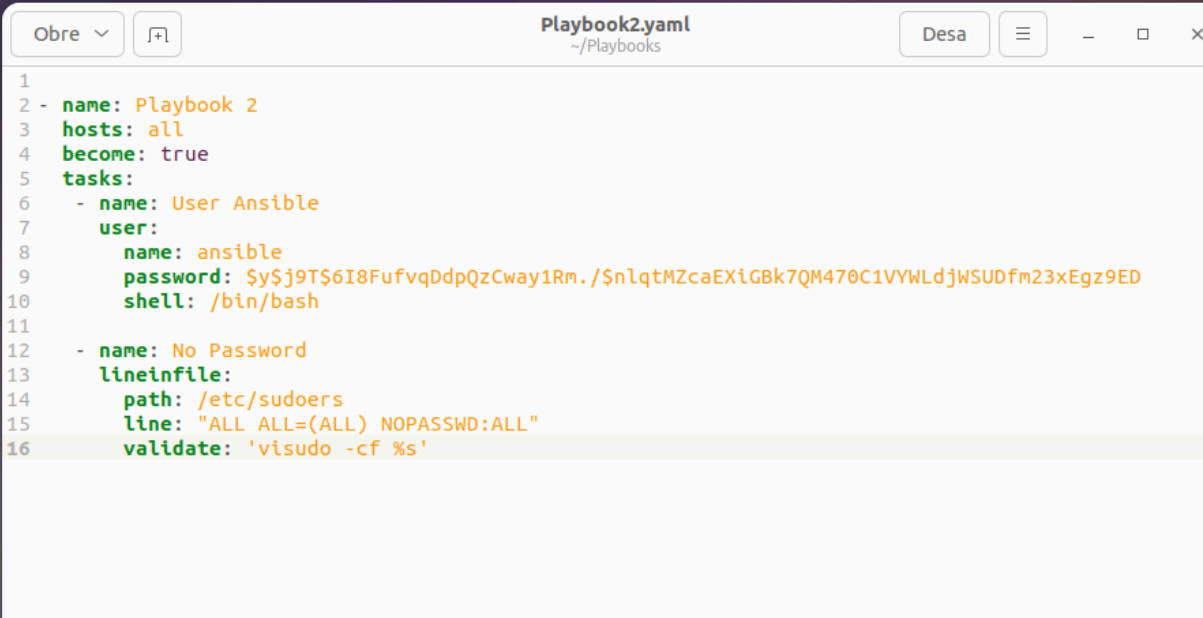
Playbook2.yaml

En aquest playbook una de les tasques es modificar el fitxer **/etc/sudoers** dels nodes gestionats per tal de no haver d'introduir cap password al realitzar operacions amb sudo.

Màquines: all

Tasques: Fer canvis a tots els nodes gestionats per tal de fer dues tasques:

- Creació d'un usuari anomenat **ansible**.
- Poder executar comandes amb sudo sense que demani password amb un usuari anomenat **ansible**.



```
1
2 - name: Playbook 2
3   hosts: all
4   become: true
5   tasks:
6   - name: User Ansible
7     user:
8       name: ansible
9       password: $y$j9T$6I8FufvqDdpQzCway1Rm./$n1qtMZcaEXiGBk7QM470C1VYWLdjWSUDfm23xEgz9ED
10      shell: /bin/bash
11
12  - name: No Password
13    lineinfile:
14      path: /etc/sudoers
15      line: "ALL ALL=(ALL) NOPASSWD:ALL"
16      validate: 'visudo -cf %s'
```

Playbook 3.yaml (aquest Playbook ja està sol·lucionat. proveu-lo)

Màquines: all

Tasques:

- Distribuir la clau id_rsa.pub de l'usuari isard a la resta de nodes gestionats.

En aquest playbook introduïrem un element nou. Farem servir un **fitxer de variables** d'on Ansible llegirà dades que es podran referenciar des dels playbooks.

Creeu un subdirectori anomenat **/vars/** al vostre directori de treball.

Dintre heu de crear un fitxer anomenat **defaults.yaml** , amb el següent contingut:

```

---
created_username: isard
copy_local_key: "{{ lookup('file', lookup('env','HOME') + '/.ssh/id_rsa.pub') }}"
paquets: [ 'curl', 'glances', 'git', 'ufw' ]

```

Introduïrem a l'inici de cada playbook, després de **become: true**, la part en negreta indicada a continuació per referenciar el fitxer on dipositem les variables:

```

---
- hosts: all
  become: true
  vars_files:
    - vars/default.yaml

```

Quedaria així:

```

Playbook3.yaml
---
- name: Distribuir clau SSH user isard
  hosts: all
  become: true
  vars_files:
    - vars/default.yaml
  tasks:
    - name: copiar clau SSH al node remot
      ansible.posix.authorized_key:
        user: "{{ created_username }}"
        state: present
        key: "{{ copy_local_key }}"

```

El mòdul `authorized_key`: Copiarà la clau pública (`id_rsa.pub`) de l'usuari ***isard*** del node de control, al fitxer `~/.ssh/authorized_keys` de l'usuari ***isard*** (si no especifiquem un diferent al executar el *playbook*) de la resta de nodes gestionats.

La funció ***lookup*** que apareix al **`default.yaml`** es fa servir per obtenir la ruta absoluta a la clau pública predeterminada de l'usuari ***isard*** al node de control Ansible.

Playbook4.yaml

Feu servir el fitxer **`/vars/default.yaml`** i la variable **`paquets`** que apareix, per resoldre aquest *playbook*.

Màquines: all

Tasques: apt update + instal·lar i paquets bàsics git, curl, glances, ufw.

```

Playbook4.yaml
---

```

Amb `latest` asseguren no només que instal·la el paquet si no existeix, també instal·larà l'última versió.

Si afegim `update_cache:` a una tasca, s'executa un *apt update* abans d'executar la tasca.



```
1 ---
2 - name: Update & Install packs
3   hosts: all
4   become: true
5   vars_files:
6     - vars/defaults.yaml
7   tasks:
8     - name: Update Cache
9       apt:
10         update_cache: yes
11     - name: Install git, curl, glances, ufw
12       apt:
13         name: "{{ paquets }}"
14         state: latest
```