Ansible

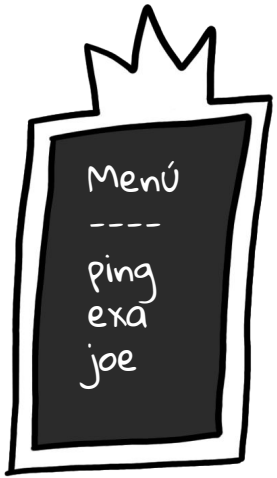
Dia 3: Playbooks i roles

Xavier Sala Pujolar

Universitat de Girona

Maig 2021

Roles



L’objectiu dels Roles és partir els playbooks en peces per fer-los més fàcils de reusar i compartir

La idea bàsica és partir un playbook en roles que només facin una sola tasca

Els roles són una forma d'organitzar un playbook per fer-lo reusable

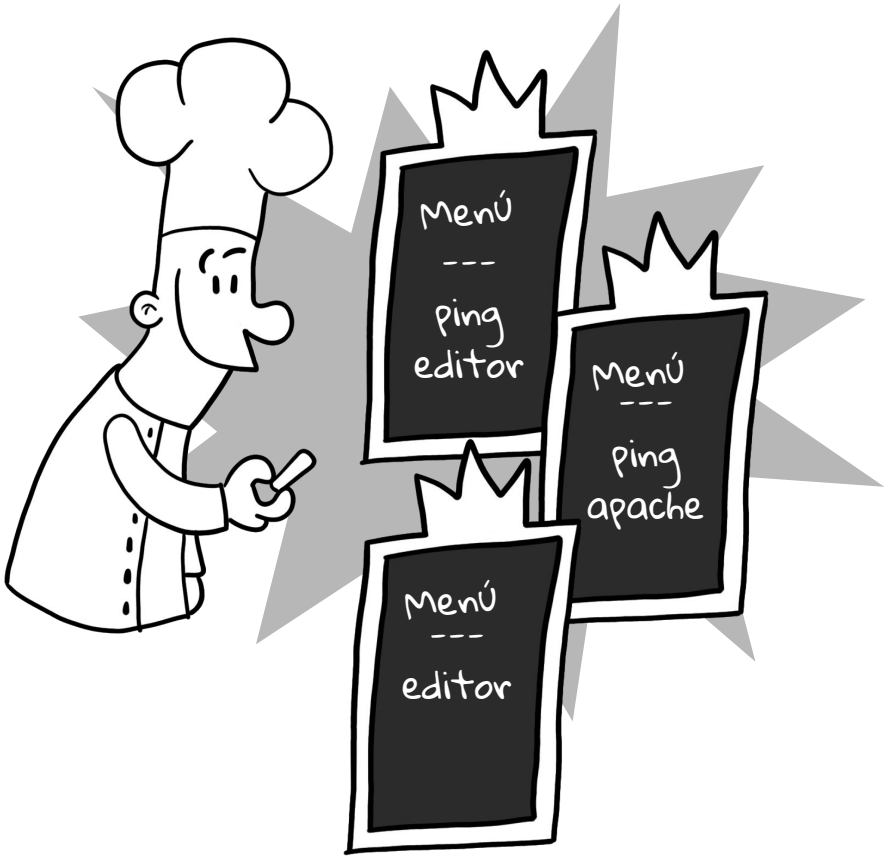
joe.yml

```
---
- hosts: xinxan.local
  user: pi

  vars:
    - editor: joe

  tasks:

    - name: Comprovar que està viu
      ping:

    - name: Instal·lar editor
      apt:
         name: {{ editor }}
         state: present
      become: yes
```

fer ping

Instal·lar editor

Cada funcionalitat es pot usar per crear diversos playbooks



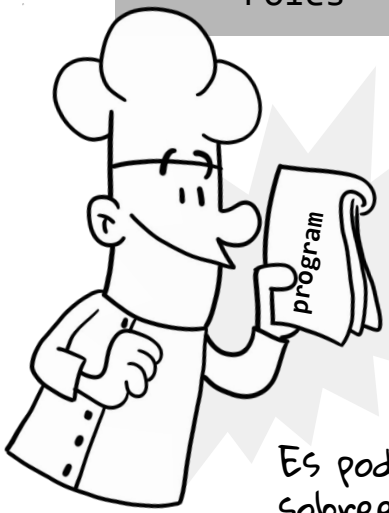
Crear playbooks amb roles

fantastic.yml

```
---
- hosts: xinxan.local
  user: pi

  roles:
   - ping
   - editor
```

Només cal fer referència al role

Es poden passar variables als roles



Es poden sobreescriure les variables per **defecte**

```
---
- hosts: xinxan.local
  user: pi

  roles:
   - edit
```

```
---
- hosts: xinxan.local
  user: pi

  roles:
   - role: edit
     vars:
       program: “nano”
```

Roles de la comunitat

La comunitat comparteix roles a **Ansible Galaxy**

https://galaxy.ansible.com/

Roles de la comunitat

```
ansible-galaxy search ntpd
```

```
ansible-galaxy info geerlingguy.ntp
```

```
ansible-galaxy install geerlingguy.ntp
```

```
ansible-galaxy list
```

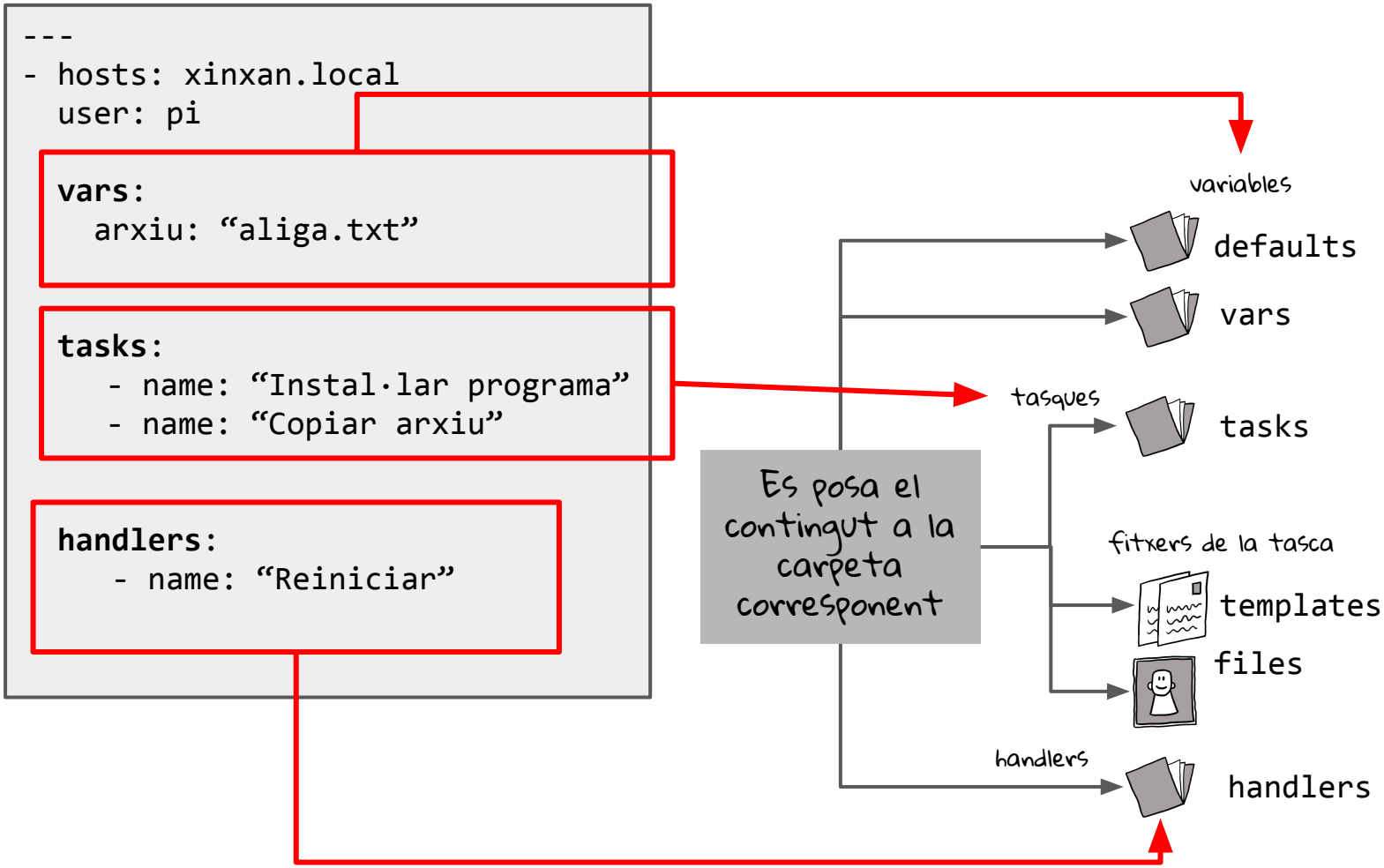
Creació d'un Role

Nom del role

```
mkdir roles
cd roles
ansible-galaxy init aliga
```

Crea l'estructura de directoris

```
.
└── aliga
    ├── defaults
    │   └── main.yml
    ├── files
    ├── handlers
    │   └── main.yml
    ├── meta
    │   └── main.yml
    ├── README.md
    ├── tasks
    │   └── main.yml
    ├── templates
    ├── tests
    │   ├── inventory
    │   └── test.yml
    └── vars
        └── main.yml
```



- hosts: xinxan.local
user: pi
vars:
arxiu: “aliga.txt”
tasks:
- name: “Instal·lar programa”
- name: “Copiar arxiu”
handlers:
- name: “Reiniciar”
Es posa el contingut a la carpeta corresponent
variables
defaults
vars
tasques
tasks
fitxers de la tasca
templates
files
handlers
handlers

Vault

A vegades cal entrar informació confidencial en els playbooks.

Ansible Vault permet xifrar aquesta informació per evitar que caigui en males mans

secret.yml

```
secret:
   programa: joe
```

```
ansible-vault encrypt secret.yml
```

secret.yml

```
$ANSIBLE_VAULT;1.1;AES256
30393635353361393139613733326638363239613933343530393433623963353137366363303832
32323266613532646661366637636431363137653361393 60a6166373636396263623764306332 63
33663935363430633665343835363931323535383937303866343434313834363062353662386431
62613438336338643 90a623730323266613563336265313732616236633730663531373731663130
33613662376635333235613962656133643530353365326566353463386234633132386530653862
64326335343139313961343366343461363737383030643034 63
```

programa.yml

```
---
- hosts: xinxan.local
  user: pi

  tasks:

   - name: Instal·lar programa secret
     apt:
        name= {{ secret.programa }}
        state=present
     become: yes
```

Es podrà usar com una variable normal

S'ha de proporcionar la contrasenya de desxifrat

```
ansible-playbook --ask-vault-pass program.yml
```

Plugins

Els plugins permeten incrementar la funcionalitat bàsica d'Ansible.

Afegeixen noves opcions com suport per cachés, nous modes de connexió, etc..

Hi ha plugins de diferents categories

Normalment es carreguen automàticament

Els de **connection** afegeixen tipus de connexions

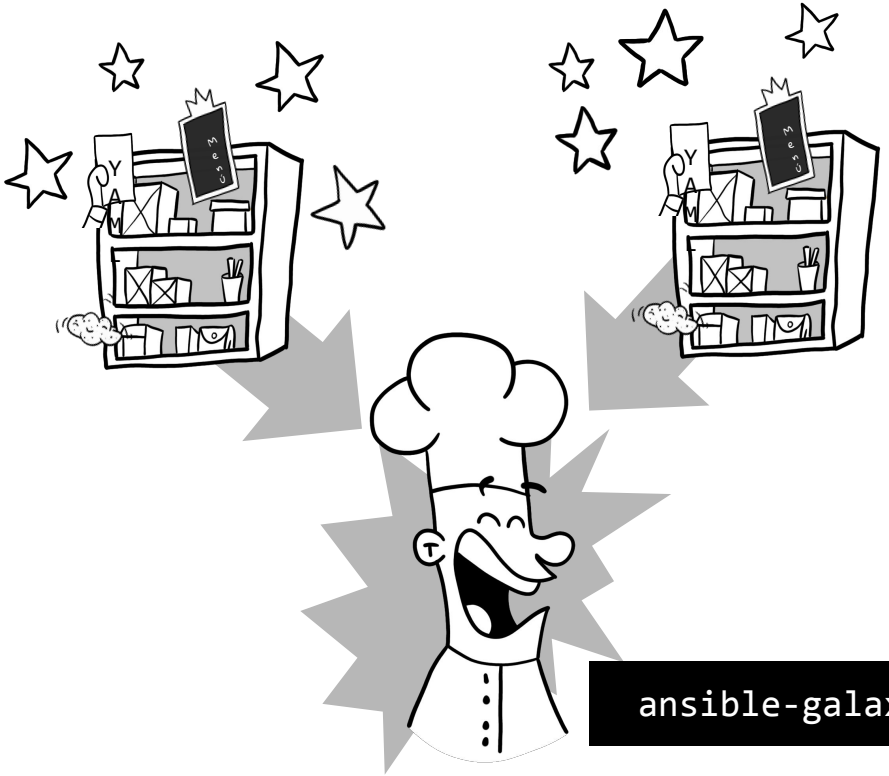
```
ansible-doc -t connection -l
```

categories de plugins

Collections

Les col·leccions són un format per distribuir playbooks, roles, mòduls i plugins.

Els mòduls bàsics també s'estan agrupant en col·leccions



Ansible Galaxy també permet importar col·leccions

```
ansible-galaxy collection install community.docker
```

Es poden instal·lar de repositoris git

Dependències dels playbooks

Es poden definir les dependències en un fitxer

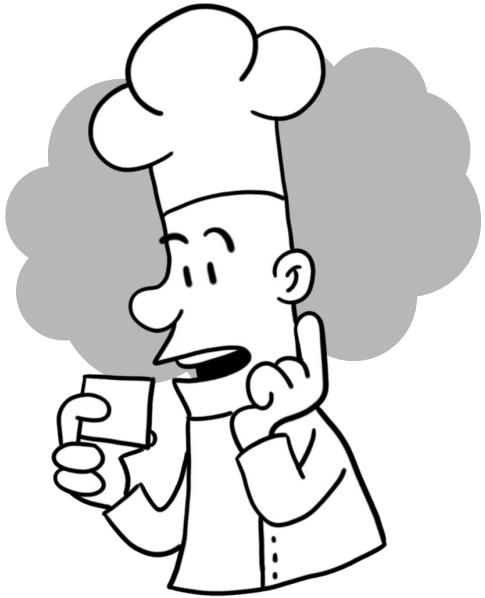
requirements.yml

```
---
collections:
  - name: community.docker
  - name: community. podman

roles:
  - aliga
```

Es poden instal·lar totes d'un cop

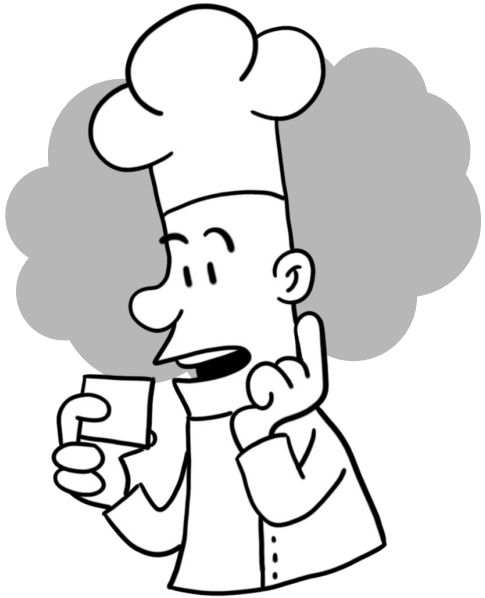
```
ansible-galaxy install -r requirements.yml
```

Usar col·leccions



Es fa servir la col·lecció especificant el mòdul, etc.. **amb el namespace**

```
---
- hosts: xinxan.local
  user: pi

  tasks:
   - name: fer servir la col·lecció
     community.docker.docker_container:
        name: x
        image: utrescu/testdb
```



El mateix passa amb els roles de la col·lecció

```
---
- hosts: xinxan.local
  user: pi
  collections:
   - udg.prova

  role:
   - aliga
```

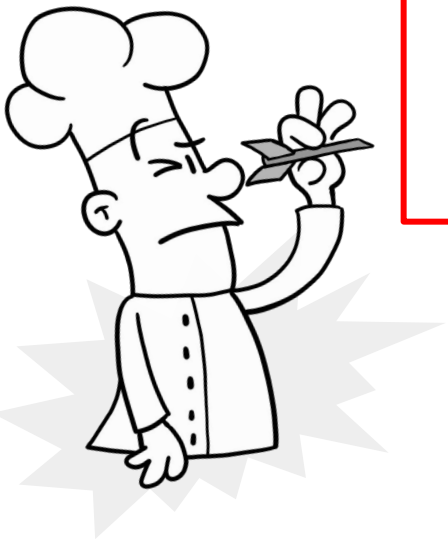
Es pot definir el namespace

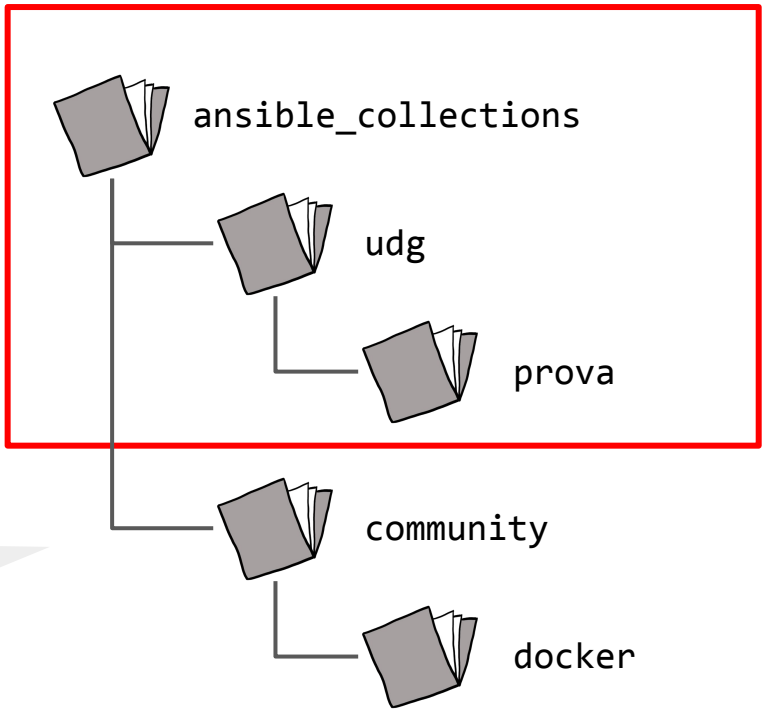
Crear una col·lecció

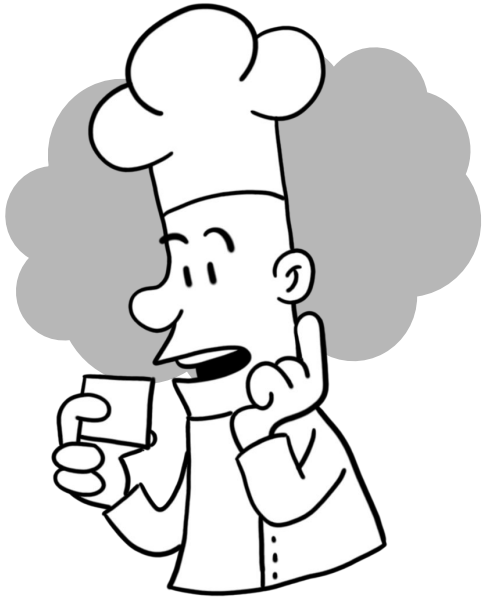
```
ansible-galaxy collection init udg.prova
```

Crea l'estructura de directoris

Les col·leccions es col·loquen en carpetes predefinides







Es copia un role a la carpeta "**roles**" i ja es pot usar

```
---
- hosts: xinxan.local
  user: pi
  collections:
   - udg.prova

  role:
   - aliga
```

Provablement en les futures versions tot acabarà en col·leccions

Ansible Tower

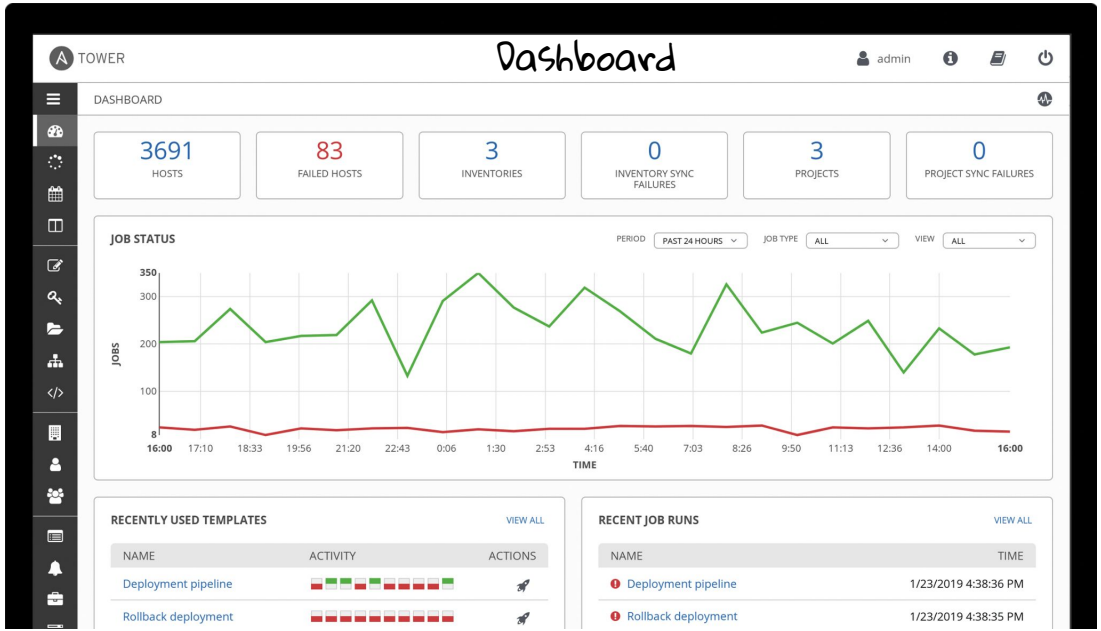
Ansible Tower és la solució empresarial de Red Hat.

En infraestructures grans cal alguna forma de tenir usuaris diferents nivells d'accés a Ansible.

Defineix grups i usuaris amb nivells d'accés

Control en temps real de les execucions

Auditoria de canvis



Editor de fluxos de treball

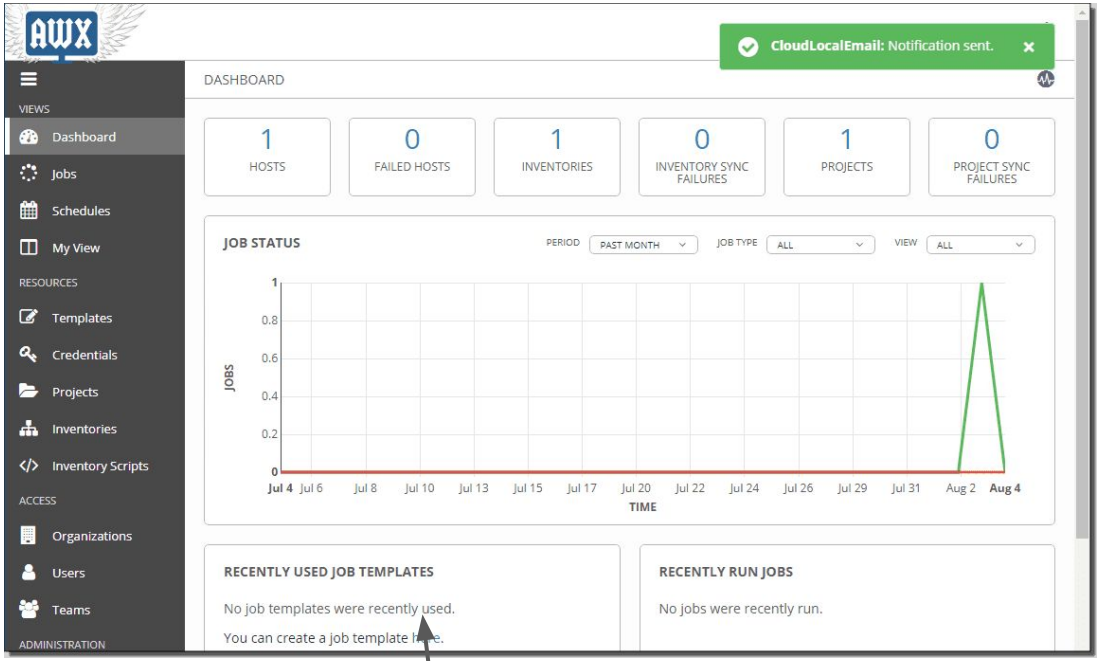
Gestió de l'inventari

API Rest

Programació de cada quan aplicar playbooks

Projecte de codi obert

Desenvolupament ràpid



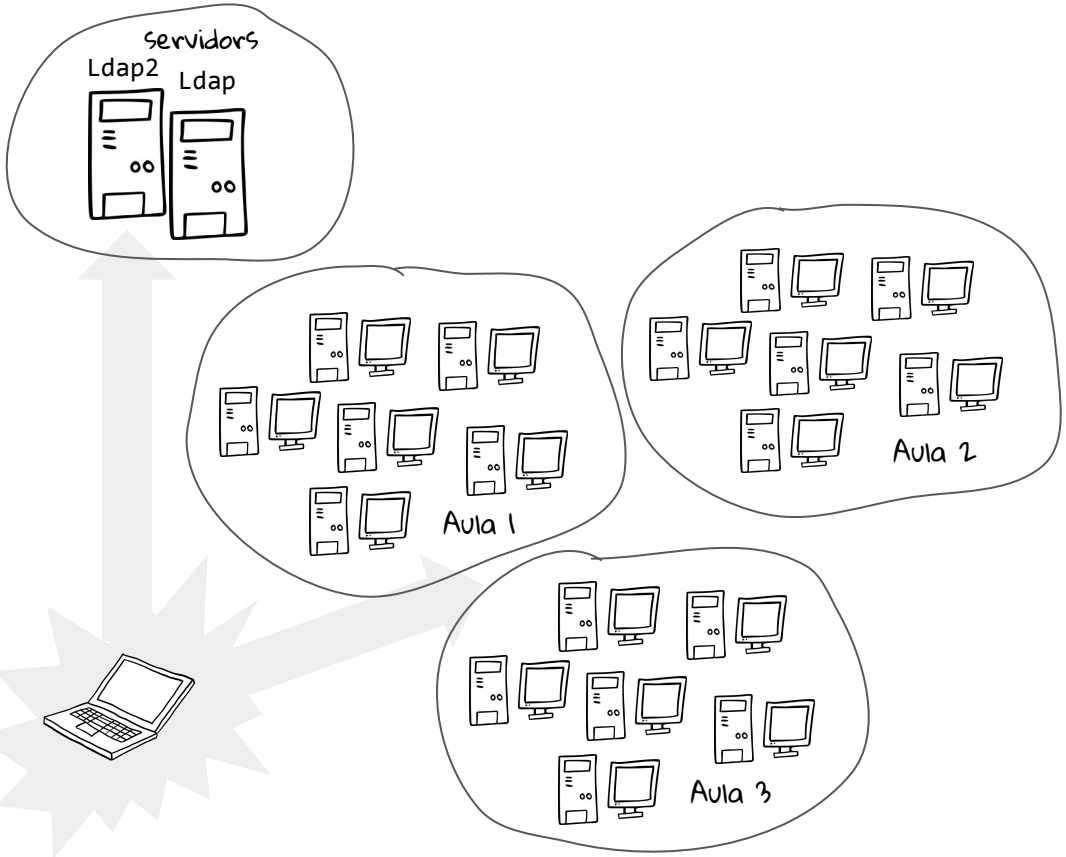
Tower s'hi basa

Exemples

Alguns exemples d’execució de playbooks fent servir Ansible

Exemple

- El sistema està controlat per dos servidors Ldap
  Els equips de les aules
- Les aules tenen una llista de software que prové de diferents fonts
- Han de fer login contra LDAP però tenir un usuari genèric per fallades de xarxa



2021