

Ansible

or: How I learned to stop worrying and love
the configuration manager



POLITECNICO OPEN
unix LABS

Come hack with us.

Slides?

Le slides sono all'indirizzo:

slides.poul.org/2018/corsi-linux-avanzati/Ansible

Ansible: What is that?

Ansible

Ansible è un software che si occupa di automatizzare la configurazione di macchine remote (e non), del deploy di applicazioni e di fare software provisioning.



Caratteristiche principali:

- **Minimale**: non richiede che i nodi abbiano software particolare installato (solo python e OpenSSH)
- **Sicuro** : Tutte le operazioni vengono eseguite mediante OpenSSH
- **Semplice**: Le configurazioni sono basate su YAML
- **IaC**: La struttura dei nostri servizi e del/i nostro/i server può essere versionata e trattata come se fosse codice (Infrastructure as Code)

N.B.

Il fatto che possa essere trattato come codice vuol dire che può essere **versionato** (e.g. con **GIT**) e che possiamo scrivere dei **test**.

Come installarlo

Per **installarlo** si può:

- Usando il package manager della propria versione di GNU/Linux
- Installandolo con pip: `$ pip install ansible`
- Eseguendolo da **sorgente**

Ansible

Some super simple stuff

Define hosts

La lista degli hosts si chiama **inventory**.

I file sono in formato **INI**

La posizione di default dell'inventory è */etc/ansible/hosts*,
ma possiamo definire un'inventory alternativa passandola
con l'opzione *-i*.

Inventory Structure

Nell'inventory possiamo definire gli host:

- Singolarmente
- Come gruppi, definendo il nome prima degli host
- Come gruppi di gruppi, sempre definendoli prima della lista di gruppi membri.

Inventory: An Example

```
# Definisco un gruppo
[webserver_centos]
webcentos.example.com
webcentos2.example.com

[webserver_ubuntu]
# Definisco degli host con variabile numerica
web[0:4].example.com

# Definisco un gruppo di gruppi
[webserver:children]
webserver_centos
webserver_ubuntu
```

Playbook

Un **playbook** è un insieme di istruzioni che mette in relazione host, task, comandi e utenti, in un singolo file. I playbook sono in formato **YAML**.

I playbook sono i task che vogliamo vengano eseguiti, nel modo e con le opzioni specificate.

Playbook Structure

```
---  
- hosts: # Definisco il gruppo a cui applicare la descrizione  
  remote_user: root # Definisco come mi autentico  
  tasks: # Definisco le operazioni da eseguire  
    - name: Latest apache # Il nome mostrato all'esecuzione  
      yum:  
        name: httpd  
        state: latest  
# Posso definire altri task da eseguire  
- name: ensure apache is running (and enable it at boot)  
  service:  
    name: httpd  
    state: started  
    enabled: yes
```

Side Note:

Se l'host è una Ubuntu 16.04, probabilmente è installato python 3.

A noi serve **python 2**.

Dobbiamo aggiungere queste righe:

```
gather_facts: False
pre_tasks:
  - name: install python
    raw: apt -y update && apt install -y python-minimal
  - setup:
```

Variables

Si possono definire delle variabili all'interno del playbook.

```
- hosts: webservers
  vars:
    http_port: 80
```

Variables

Le variabili vengono utilizzate come i dizionari di python (o meglio, sono lo stesso tipo di struttura). Per utilizzarle si usa la sintassi di **Jinja2**.

```
{{ http_port }}
```

CLI Syntax

Ok, ho il mio playbook. Come faccio ora a eseguirlo?

```
ansible-playbook -i /path/alla/mia/inventory /path/al/playbook.yml
```

Demo



Demo

Ma Snake?!



Facciamo andare **Snake!**

Side Note:

Privilege Escalation

Ansible permette di cambiare l'utente con cui ci colleghiamo e farlo diventare un altro (e.g. user -> root). Questo procedimento si chiama **Privilege Escalation**.

Privilege Escalation:

Per farlo basta usare nel playbook:

```
become: yes  
become_user: root  
become_method: sudo
```

Ansible Galaxy

Ansible Galaxy

Devo eseguire un'operazione abbastanza comune (installare docker, installare un mailserver, ecc). Ho veramente voglia di scrivere tutte le volte la configurazione? Come posso fare?

Ansible Galaxy

Posso utilizzare **Ansible Galaxy**!

Ansible Galaxy: What is that?

Ansible Galaxy è un hub dove vengono condivisi diversi **roles** e da cui possiamo attingere nel momento in cui vogliamo riutilizzarli

Roles?

Roles!

I **roles** indicano un determinato insieme di variabili, task e template che vengono automaticamente caricati partendo da una determinata struttura file.

Roles

Nello specifico:

```
myfantastic_role/.  
├── tasks  
├── files  
├── templates  
└── vars
```

- *task* contiene i task che verranno eseguiti
- *files* contiene i file che verranno trasferiti
- *template* contiene i template che generano configurazione in base a delle variabili
- *vars* contiene la definizione di alcune variabili che possono servirci

Ansible Galaxy: CLI

Ansible Galaxy è utilizzabile mediante il comando
`ansible-galaxy`

Ansible Galaxy: CLI

```
ansible-galaxy [options]
```

- **search** ci permette di cercare un determinato termine
- **info** ci permette di avere più informazioni su un role specifico
- **install** ci permette di installare il role specificato per poterlo poi utilizzare

Demo



Demo: LAMP Server

LAMP è un acronimo e sta per:

- **L**inux
- **A**pache
- **M**ySQL
- **P**hp

Demo

```
ansible-galaxy install \
geerlingguy.mysql geerlingguy.apache geerlingguy.php
```

Ok, ho scaricato i roles necessari. Come faccio ora ad utilizzarli?

Demo

Con un bel playbook!

Ansible Tower



Ansible Tower

Ansible Tower è una dashboard che permette di controllare e gestire infrastrutture complesse.

Ansible Tower

Funzionalità interessanti:

- **Realtime jobs update**: Possiamo avere un output del task eseguito, per ogni macchina, in tempo reale
- **Logs**: Possiamo vedere chi ha eseguito cosa e quando
- **Scheduling**: Possiamo schedulare quando eseguire un playbook, quando fare l'update dell'inventary e altre task.
- **Open Source**: Da poco è diventato un progetto **open source**

That's All



Domande?

Ok, I Lied



Non vi ho detto proprio tutto quello che si può fare con Ansible.

Ok, I Lied

Di cosa non vi ho parlato:

- **Vault**: Non è una buona idea avere le (eventuali) password salvate in chiaro nel nostro playbook. Per questo esiste **Ansible Vault**.
- **Template**: Si possono definire dei template per i file da creare, sfruttando le variabili settate.
- **Testing**: Si possono scrivere dei test per i template definiti.

Tutte questo, assieme ad altre funzionalità, potete trovarle nei link alla fine!

Links

- [Ansible Docs Page](#)
- [Examples](#)
- [Ansible from beginner to pro](#)
- [Security Disclosures](#)

License



Quest'opera è distribuita con Licenza **Creative Commons**
Attribuzione - Condividi allo stesso modo 4.0 Internazionale.