Ansible

Ansible

Ansible este un tool de automatizare ce este utilizat pentru a instalarea diferitelor pachete și configurarea serverelor. Scopul acestui tool este de a elimina configurarea manuală a serverlor și a elimina posibilele erori ce ar veni cu aceasta.

Sintaxa Ansible este scrisă în yaml

De ce îl folosim? Simplu de folosit și configurat, idempotent , are plugin-uri pentru diferite servicii de cloud (aws, azure, digitalocean etc)

Un alt tool asemanator cu ansible este **chef**

Exemplu de fisier yaml

```
# Employee records
-   martin:
      name: Martin D'vloper
      job: Developer
      skills:
        - python
        - perl
        - pascal
```

Cum arată un fișier yaml în ansible?

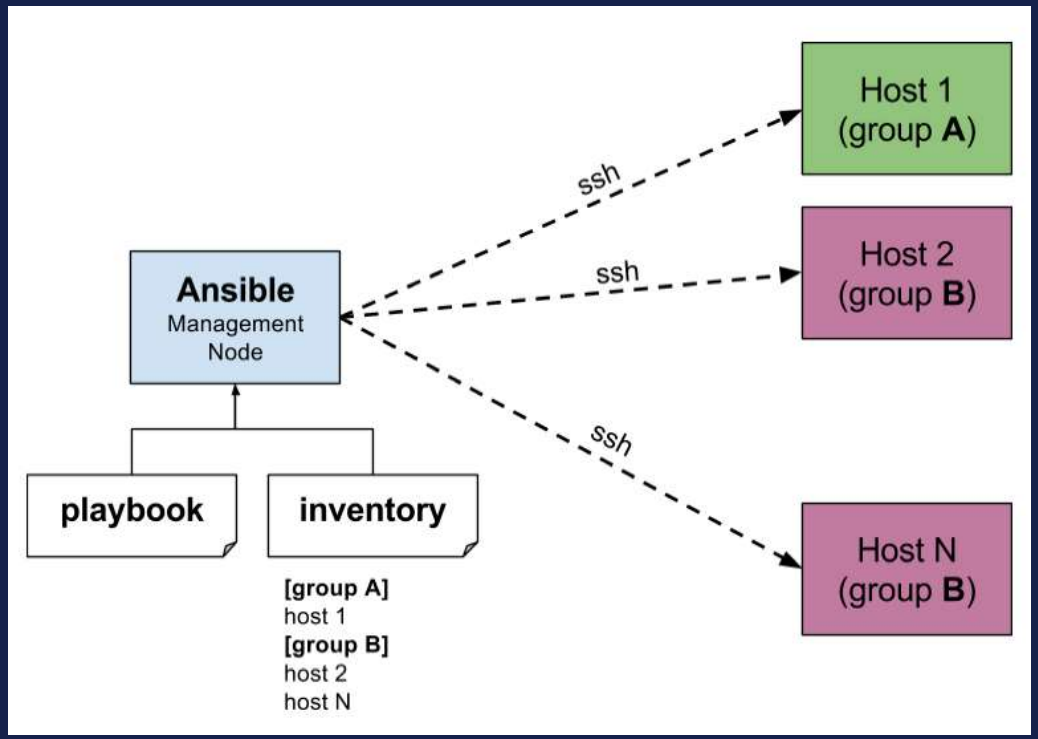
Sintaxa Ansible este scrisă în yaml. De asemenea, mai multe detalii puteți găsi aici

Pentru conversie yaml -> json și invers

```
---
- hosts: all
  tasks:
  - name: Package installation
    dnf:
      name:
        - sysstat
        - httpd
        - mariadb-server
      state: latest
```

IT School
FROM ZERO TO HERO!

Cum functioneaza?



Sursa imagine

Cum funcționează?

Prima dată este necesar să avem un **server** (poate fi numit și nod) **master** , unde avem ansible instalat și un **nod client** unde se va conecta serverul de master prin ssh și va rula configurația citită din fișierul **.yaml** (de obicei un playbook file .yaml)

Mai întâi trebuie să ne asigurăm de faptul că nodul de master are acces ssh la nodul client. Putem face această conexiune prin username și parolă, dar ideal este să fie dat accesul printr-o pereche de chei (publică-privată). Putem face asta foarte ușor de pe nodul de master cu comanda
ssh-copy-id username_client@ip_client

Sau ca exercițiu se poate configura manual

Ce conține?

Serverul de ansible: Acesta este serverul care are ansible instalat și de aici vor rula playbook-urile și task-urile

Modulul: Este o comanda sau un set de comenzi pe care le trimitem de pe server pe host-uri pentru a fi executate (ex: modulul să instaleze un anumit pachet)

Inventory: Un fișier ce conține informații despre serverele client

Task: Cea mai mică unitate de acțiune ce se poate automatiza în cadrul unui playbook

Playbook: O listă de task-uri ce se execută pe un host

Role: Un mod de a organiza task-urile

Instalarea ansbile

Pași de instalare

Este nevoie să aveți python și pip instalat . Prin această metodă se instaleaza ansible prin python

python3 -m pip install --user ansible
python3 -m pip install --user ansible-core==2.12.3
ansible --version

Altă metoda de instalare

Inventory file

Ansible va fi rulat pe diferite host-uri de workeri din infrastructura ta în același timp folosind un grup de host-uri numit inventory

Acest fișier poate avea diferite formate cele mai comune fiind INI și YAML. De obicei este implicit în **/etc/ansible/hosts** , dar putem specifica atunci când rulăm ansible să utilizeze alt fișier de inventory (ceea ce este și indicat)

Inventory file

Dacă ați instalat ansible direct prin python, este posibil să nu aveți acest folder /etc/asnbile/ . La nevoie se poate crea manual, dar nu este indispensabil rulării ansbile

Exemplu de inventory file:
mail.example.com

[webservers]
foo.example.com
bar.example.com

Add hoc command

inv.yml ->

[node1]
192.168.1.115

[node2]
192.168.1.116

ansible node1 -i inv.yml -m ping
ansible all -i inv.yml -m ping

ansible-doc -l | grep ping -> pentru a lista modulele disponibile și pentru a căuta unul anume

Add hoc command

inv.yaml ->

[others]
myclient ansible_host=192.168.1.116

[node]
192.168.1.116

ansible myclient -i inv.yml -m ping

În inv.yaml poți seta și cu ce user să se conecteze nodul de master la nodul worker

myclient ansible_host=192.168.1.116 ansible_user=george

Add hoc command with sudo

ansible myclient -i inv.yml -m ansible.builtin.apt -a "name=apache2 state=latest" --become -K

ansible myclient -i inv.yml -m ansible.builtin.service -a "name=apache2 state=restarted" --become -K
(cu flag-ul “-K” se va cere introducerea parolei pentru userul respectiv de pe nodul client)

ansible clients -m ansible.builtin.apt -a "name=nginx state=latest" -b

(-b de la become, adică privilegii ridicate pentru user echivalent cu a executa o comandă cu sudo)

Add hoc command with sudo

Pentru a putea rula comenzi de sudo pe serverul client fără să specificăm mereu parola , este necesar ca userul remote (al clientului) prin care rulăm ansible să aibă drepturi de sudo fără a mai fi necesară parola. Pentru asta trebuie să îi dăm acele drepturi

Adăugăm în /etc/sudores -> client_user ALL=(ALL) NOPASSWD:ALL

Ansible playbook

Ansible playbook reprezintă un set de task-uri automate, ce se execută cu intervenția minimă umana (sau chiar deloc). Playbook-ul se poate executa pe un singur host sau pe serie de host-uri.

```yaml
---
- hosts: webservers
  sudo: yes

  vars:
    app_name: PleaseDeployMe
    repo_url: https://github.com/username/repo_name.git
    repo_remote: origin
    repo_version: master
    webapps_dir: /deployed
    virtualenv_root: /deployed/PleaseDeployMe/mac
  tasks:

    - name: git pull project
      git: repo={{repo_url}} dest={{webapps_dir}}/{{app_name}} version=master

      notify:
        - restart app

    - name: install things
      pip: name=virtualenv

    - name: create virtualenv
      command: virtualenv /deployed/PleaseDeployMe/venv

    - name: activate virtualenv
      command: /bin/bash /deployed/PleaseDeployMe/venv/bin/activate

    - pip: requirements=/deployed/{{app_name}}/requirements.txt virtualenv=/deployed/{{app_name}}/mac

    - name: run supervisord
      command: "supervisord -c /deployed/PleaseDeployMe/supervisord.conf"

    - name: begin flask app
      supervisorctl: name=flask_app state=started


  handlers:
    - name: restart app
      supervisorctl: name={{app_name}} state=restarted


```

Sursa [imagine]

Ansible role

Permite refolosirea mai multe componente prin gruparea și încapsularea lor într-o structură de task-uri, handlers, fișiere de configurație etc

Un folder cu role ansible va conține o structură standard cu mai multe directoare, cum ar fi: defaults, vars, tasks, files, templates, meta, handlers

Ansible configuration file

Unele configurații de ansible pot fi modificate prin fișierul **ansible.cfg**
De exemplu ce cheie ssh să utilizeze, sau unde să caute fișierul de inventory

george@george-VirtualBox:~/Ansible$ cat ansible.cfg

[defaults]

private_key_file=/home/george/Ansible/id_rsa
inventory = /home/george/Ansible/inv.yaml

Variabile

Poți defini variabile în playbook-ul tău, în inventory sau role-uri sau chiar în linia de comandă pentru a face configurațiile ansible cât mai dinamice

ansible-playbook -i inv.yaml --extra-vars mysql_port=9999 mariadb.yaml

Acestea vor avea prioritate față de variabilele din mariadb.yaml
Ansible are o ierarhie bine definită a variabilelor dacă acestea se suprapun.

Register

Poți crea variabile din output-ul unui task de ansible folosing “register” și te poți folosi aceste valori mai tarziu.

```
george@george-VirtualBox:/tmp/test$ cat play.yml
- hosts: myclient
  become: True
  tasks:
    - name: install pwgen
      apt:
         name: pwgen
         state: present
         update_cache: True
     #you can test it with: pwgen -s -1 15
    - name: Generate password
      shell: pwgen -N 1 -s 30
      register: mypass

    - name: Print the generated password
      debug:
        msg: "The password is {{ mypass }}"
```

IT School®
FROM ZERO TO HERO!

Template

Ansible folosește șablonul Jinja2 pentru a permite expresii dinamice și acces la variabile.

```
- hosts: all
  vars:
    variable_to_be_replaced: 'Hello world'
    inline_variable: 'hello again'
  tasks:
    - name: Ansible Template Example
      template:
        src: hello_world.j2
        dest: /Users/mdtutorials2/Documents/Ansible/hello_world.txt

hello_world.j2
--------------
{{ variable_to_be_replaced }}
This line won't be changed
Variable given as inline - {{ inline_variable }} - :)

output - hello_world.txt
------
Hello world
This line won't be changed
Variable given as inline - hello again - :)

mdtutorials2$ ls -lrt hello_world.txt
-rw-r--r--  1 root  wheel  81 Oct 16 07:23 hello_world.txt
```

Sursa imagine

IT School FROM ZERO TO HERO!

Handler

Uneori vrei să rulezi un task de ansible doar în cazul în care un eveniment sau o schimbare s-a produs. De exemplu poți restarta un serviciu pentru as reinițializa configurația doar dacă configurația s-a schimbat, altfel ai restarta serviciul degeaba .

Pentru aceste cazuri poți folosi handler-ul din ansible. Acestea sunt task-uri ce rulează doar când sunt notificate să o facă.

Ansible when

Poți adauga intrucțiunea condițională **when** pentru a face ansible să ruleze doar când o anumită condiție este indeplinită. De exemplu poți instala un pachet (și defini în ce mod vrei să îl instalezi **apt install** sau **yum install**) doar dacă sistemul pe care vrei să instalezi este de tip ubuntu și vrei să instalezi altfel dacă sistemul este de tip centos

Loop

Ansible oferă câteva structuri repetive loop, with_<lookup> și until pentru a executa niste task-rui de un numar de n ori . Acest lucru poate reduce semnificativ dimensiunea fisierelor .yaml. Similar cu structurile repetitive din programare

```
- name: Add several users
  ansible.builtin.user: name: "{{ item }}“
  state: present
  groups: "wheel“
  loop:
    - testuser1
    - testuser2
```

IT School
FROM ZERO TO HERO!

Ansible vault

Vault te ajuta să protejezi variabile sau fisiere ce conțin infirmații sensibile cum ar fi chei, parole, variable secrete de environment. Le poti ascunde cu ansible vault și doar cine are cheia de decriptare le poate vedea. Deoarece codul de ansible va sta în git, nu putem permite ca aceste valori să fie în clear text pe repo-ul de git unde toată lumea le poate vedea.

Definim un fișier mysecret.yml care va arăta așa:

```
---
password: “secretpass"
```

Ansible vault

ansible-vault encrypt mysecret.yml -> pentru a obține valoarea criptată

ansible-vault decrypt mysecret.yml -> pentru a decripta

ansible-vault view mysecret.yml -> pentru a vedea valoarea

ansible-vault edit mysecret.yml -> pentru a edita

Ansible galaxy

Ansible galaxy este un repository public ce conține diferite role-uri de ansible pe care le poți folosi pentru diferite configurații. Este un repo public mereu în evoluție și vă poate ajuta să gasiți deja configurația dorită pentru diferite instalări, fără a fi nevoiți să le creați de la zero

Aici pot fi folosite condițiile **when** din ansible, pentru că cineva poate face un role sau playbook la modul general pentru a anumită configurație (ex instalează și configurează grafana) și noi o vom putea folosi indiferent de sistemul de operare utilizat deoarece se va instala într-un fel pe centos și în altfel pe ubuntu

Pentru a genera o structură a unui role ansible puteți folosi:
ansible-galaxy init test-role-1

Mai multe detalii aici

IT School®
FROM ZERO TO HERO!

Ansible check

Ansible check este doar o simulare pentru a ne asigura de faptul ca fișierul nostru de yaml funcționează. Este o variantă foarte bună dacă vrem să validam un playbook înainte de a-l rula

ansible-playbook foo.yml –check

Ignore erros

Cu ansible ignore putem forța ca o eroare să fie ignorată și playbook-ul să meargă mai departe. Sunt cazuri în care un output non zero poate indica un succes sau în care un fail să opreasca playbook-ul și pentru celalalte

```
- name: Ignora eroarea generata
  ansible.builtin.shell: orice mesaj
  ignore_errors: yes
```

Sfârșit