

Gestion des Variables et Hiérarchie dans Ansible

Partie I : Exploration des Types de Variables

1. Préparer l'environnement

```
baraa@baraa-testVM:/etc/ansible/lab4$ ls
display_var.yml  my_inventory.ini
baraa@baraa-testVM:/etc/ansible/lab4$
```

2. Configurer l'inventaire

```
baraa@baraa-testVM:/etc/ansible/lab4$ cat my_inventory.ini
[workers]
worker1 ansible_host=192.168.244.144
baraa@baraa-testVM:/etc/ansible/lab4$
```

3. Afficher des informations réseau de la machine cible avec les variables facts

```
(gedit:3238): dconf-WARNING **: 11:11:34.023: Failed to commit changes to dconf: Failed to execute child process 'dbus-launch' (No
baraa@baraa-testVM:/etc/ansible/lab4$ ansible -i my_inventory.ini -m ping all
worker1 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
baraa@baraa-testVM:/etc/ansible/lab4$ ansible-playbook -i my_inventory.ini display_var.yml

PLAY [Afficher les informations réseau] *****

TASK [Gathering Facts] *****
ok: [worker1]

TASK [Afficher l'adresse IP] *****
ok: [worker1] => {
  "msg": "Adresse IP : 192.168.244.143"
}

TASK [Afficher l'adresse MAC] *****
ok: [worker1] => {
  "msg": "Adresse MAC : 00:0c:29:1a:c8:02"
}

TASK [Afficher les interfaces réseau] *****
ok: [worker1] => {
  "msg": "Interfaces réseau : ['flannel.1', 'cni0', 'lo', 'ens33', 'vethd8f4db77', 'docker0', 'veth83031a61']"
}

TASK [Afficher l'adresse de la passerelle] *****
ok: [worker1] => {
  "msg": "Passerelle : 192.168.244.2"
}

PLAY RECAP *****
worker1      : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

baraa@baraa-testVM:/etc/ansible/lab4$
```

4. Conditions selon les systèmes d'exploitation :

```
Open  [+] display_var.yml /etc/ansible/lab4 Save [ ] [ ] [X]
1 ---
2 - name: Afficher les informations réseau
3   hosts: workers
4   gather_facts: yes
5
6   tasks:
7     - name: Afficher l'adresse IP
8       debug:
9         msg: "Adresse IP : {{ ansible_default_ipv4.address }}"
10
11     - name: Afficher l'adresse MAC
12       debug:
13         msg: "Adresse MAC : {{ ansible_default_ipv4.macaddress }}"
14
15     - name: Afficher les interfaces réseau
16       debug:
17         msg: "Interfaces réseau : {{ ansible_interfaces }}"
18
19     - name: Afficher l'adresse de la passerelle
20       debug:
21         msg: "Passerelle : {{ ansible_default_ipv4.gateway }}"
22
23     - name: Afficher la distribution et la version du système
24       debug:
25         msg: "OS : {{ ansible_distribution }} {{ ansible_distribution_version }}"
26
27     - name: Arrêter le service UFW si l'OS est Ubuntu 22.04 ou 24.04
28       become: yes
29       command: systemctl stop ufw
30       when: ansible_distribution == "Ubuntu" and ansible_distribution_version in ["22.04",
"24.04"]
```

```
TASK [Afficher la distribution et la version du système] *****
ok: [worker1] => {
  "msg": "OS : Ubuntu 22.04"
}
TASK [Arrêter le service UFW si l'OS est Ubuntu 22.04 ou 24.04] *****
changed: [worker1]
PLAY RECAP *****
worker1      : ok=7  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
baraa@baraa-testVM:/etc/ansible/lab4$
```

5. Variables Magiques

```
31
32 - name: Afficher la liste des groupes et leurs hôtes
33   hosts: localhost
34   gather_facts: false
35
36   tasks:
37     - name: Afficher les groupes et leurs hôtes
38       debug:
39         msg: "{{ groups }}"
40
```

```
PLAY [Afficher la liste des groupes et leurs hôtes] *****
TASK [Afficher les groupes et leurs hôtes] *****
ok: [localhost] => {
  "msg": {
    "all": [
      "worker1"
    ],
    "ungrouped": [],
    "workers": [
      "worker1"
    ]
  }
}
PLAY RECAP *****
localhost   : ok=1  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
worker1     : ok=7  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
baraa@baraa-testVM:/etc/ansible/lab4$
```

6. Variable enregistrée

```
12 - name: Afficher la liste des groupes et leurs hôtes
13   hosts: localhost
14   gather_facts: false
15
16   tasks:
17     - name: Afficher les groupes et leurs hôtes
18       debug:
19         msg: "{{ groups }}"
20
21     - name: Exécuter la commande date et stocker la sortie
22       command: date
23       register: current_date
24
25     - name: Afficher la date et l'heure récupérées
26       debug:
27         msg: "Date et heure actuelles : {{ current_date.stdout }}"
28
29
30
31
```

```
TASK [Exécuter la commande date et stocker la sortie] *****
changed: [localhost]

TASK [Afficher la date et l'heure récupérées] *****
ok: [localhost] => {
  "msg": "Date et heure actuelles : 28 2025, 11:23:14 CET"
}

PLAY RECAP *****
localhost      : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
worker1       : ok=7  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
baraa@baraa-testVM: /etc/ansible/lab4$
```

Partie II : Hiérarchie des variables

1. Préparer l'environnement

```
baraa@baraa-testVM:/etc/ansible/lab4$ ls
display_var.yml  group_vars  host_vars  my_inventory.ini  precedence_order.yml.
baraa@baraa-testVM:/etc/ansible/lab4$
```

2. Définir la variable « ma_variable » comme variable de groupe

```
baraa@baraa-testVM:/etc/ansible/lab4$ sudo cat group_vars/all.yml
ma_variable: group_var
baraa@baraa-testVM:/etc/ansible/lab4$ sudo cat precedence_order.yml.
---
- name: Test de la hiérarchie des variables
  hosts: workers
  gather_facts: false

  tasks:
    - name: Afficher la valeur de ma_variable
      debug:
        msg: "Valeur de ma_variable : {{ ma_variable }}"
baraa@baraa-testVM:/etc/ansible/lab4$ ansible-playbook -i my_inventory.ini precedence_order.yml.

PLAY [Test de la hiérarchie des variables] *****

TASK [Afficher la valeur de ma_variable] *****
ok: [worker1] => {
  "msg": "Valeur de ma_variable : group_var"
}

PLAY RECAP *****
worker1      : ok=1  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
baraa@baraa-testVM:/etc/ansible/lab4$
```

3. Définir la variable « ma_variable » dans l'inventaire

```
baraa@baraa-testVM:/etc/ansible/lab4$ sudo nano my_inventory.ini
baraa@baraa-testVM:/etc/ansible/lab4$ sudo cat my_inventory.ini
[workers]
worker1 ansible_host=192.168.244.143 ma_variable=inv_var
```

```

baraa@baraa-testVM:/etc/ansible/lab4$ ansible-playbook -i my_inventory.ini precedence_order.yml.
PLAY [Test de la hiérarchie des variables] *****
TASK [Afficher la valeur de ma_variable] *****
ok: [worker1] => {
  "msg": "Valeur de ma_variable : inv_var"
}
PLAY RECAP *****
worker1 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
baraa@baraa-testVM:/etc/ansible/lab4$

```

Remarque : la variable définie dans l'inventaire a une priorité plus haute que la variable de groupe

4. Définir la variable « ma_variable » comme variable spécifique à un l'hôte

```

baraa@baraa-testVM:/etc/ansible/lab4$ sudo nano host_vars/worker1.yml
baraa@baraa-testVM:/etc/ansible/lab4$ sudo cat host_vars/worker1.yml
ma_variable: host_var
baraa@baraa-testVM:/etc/ansible/lab4$

```

```

baraa@baraa-testVM:/etc/ansible/lab4$ ansible-playbook -i my_inventory.ini precedence_order.yml.
PLAY [Test de la hiérarchie des variables] *****
TASK [Afficher la valeur de ma_variable] *****
ok: [worker1] => {
  "msg": "Valeur de ma_variable : host_var"
}
PLAY RECAP *****
worker1 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
baraa@baraa-testVM:/etc/ansible/lab4$

```

Remarque : La variable spécifique d'hôte a une priorité plus haute que l'inventaire et la variable de groupe

5. Définir la variable « ma_variable » dans le playbook

```

baraa@baraa-testVM:/etc/ansible/lab4$ ansible-playbook -i my_inventory.ini precedence_order.yml.
PLAY [Test de la hiérarchie des variables] *****
TASK [Afficher la valeur de ma_variable] *****
ok: [worker1] => {
  "msg": "Valeur de ma_variable : playbook_var"
}
PLAY RECAP *****
worker1 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
baraa@baraa-testVM:/etc/ansible/lab4$ sudo cat precedence_order.yml.
---
- name: Test de la hiérarchie des variables
  hosts: workers
  gather_facts: false
  vars:
    ma_variable: playbook_var

  tasks:
    - name: Afficher la valeur de ma_variable
      debug:
        msg: "Valeur de ma_variable : {{ ma_variable }}"
baraa@baraa-testVM:/etc/ansible/lab4$

```

La variable définie dans vars du playbook a une priorité plus haute que toutes les définitions précédentes.

6. Tester les variables en ligne de commande

```
msg: "Valeur de ma_variable : {{ ma_variable }}"
baraa@baraa-testVM:/etc/ansible/lab$ ansible-playbook -i my_inventory.ini precedence_order.yml. -e ma_variable="valeur_cmd"
PLAY [Test de la hiérarchie des variables] *****

TASK [Afficher la valeur de ma_variable] *****
ok: [worker1] => {
  "msg": "Valeur de ma_variable : valeur_cmd"
}

PLAY RECAP *****
worker1      : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

baraa@baraa-testVM:/etc/ansible/lab$
```

7. Analyser la hiérarchie des variables

D'après tous les tests effectués, nous avons constaté que l'ordre de priorité des variables dans Ansible, du plus prioritaire au moins prioritaire, est le suivant :

ligne de commande > playbook > host_vars > inventaire > group_vars