

Tutorial 4 Création du mini CMDB.

Table des matières

- [Objectifs](#)
- [Ressources](#)
- [Pré-Requis](#)

Objectifs

Dans ce TP ,nous allons utiliser Ansible pour créer une mini CMDB. Une CMDB est une database cataloguant les configurations de nos infrastructure. TODO: un petit croquis ?

Ainsi nous allons :

- créer un nouveau projet en dupliquant le TP précédant.
- ajouter des tâches en nous aidant des modules pour :
- récupérer les différentes faits (gather_facts) concernant nos machines hôtes.
- formater des fichiers markdown en utilisant une template j2.
- Tester et Analyser les résultats.

Les actions de notre futur playbook :

- création d'un dossier de travail sur le home sur toutes nos machines (control et managed node)..
- affichage des gather_facts
- affichage du nom de host
- generation d'un rapport au format markdown via un template sur chaque machine.
- copie de tous les rapport sur les machines hôtes vers notre machine de contrpole (ubuntu-c).

Ressources

- Environnement
- Temps : 60 mn.

Pré-Requis

- avoir mis en place l'environnement dans cloud shell.
[TP01 - Mise en place du Lab](#)
- avoir configurer Ansible pour travailler avec notre environnement dans cloud shell.
[TP02 - Configurer Ansible](#)
- avoir lancer un premier playbook dans cloud shell.
[TP03 - Premier Playbook](#)

Énoncé

Etape 1 création du projet TP04 dans myWork.

1. copiez le projet TP03 en TP04. depuis la console d'ubuntu-c

```
cd /home/ansible/ansibleMelodie/myWork
cp -R ./TP03 TP04
```

ou copie dans solution [correction](#)

1. copier notre [premier playbook](#) en 20_miniCMDB.yml

Etape 2 création d'un dossier de travail

Pour faciliter notre travail nous allons faire de l'annule et remplace du dossier ~/tmp comme cela nous serons sur que le dossier ~/tmp soit vide à chaque génération du rapport.

1. ajout d'un tache de suppression du dossier et de son contenu pour chaque machine. Nous créons un jeu qui s'appelle [creation d'un dossier de travail](#) (ou comme vous voulez...)
- hosts = all puis nous créons une premiere tache que nous pouvons appeler [suppression du dossier de travail ~/tmp \(~ home\)](#)
 - module file state = absent path = ~/tmp

```
--
- name: tests de notre configuration.
  hosts: all
  tasks:
    - name: test de la connexion
      ping:
    - name: affichage d'un message avec ma variable.
      debug:
        msg: Le contenu de ma variable est {{ maVariable }}.

- name: creation d'un dossier ~/tmp.
  hosts: managed
  tasks:
    - name: ajout du dossier.
      file:
        state: directory
        path: ~/tmp

...
```

devient

```
----
- name: creation d'un dossier de travail
  hosts: all
  tasks:
    - name: suppression du dossier de travail ~/tmp (~ home)
      file:
        state: absent
```

```
path: ~/tmp
```

```
...
```

1. lançons notre playbook `ansible-playbook 20_miniCMDB.yml -i 00_inventory.yml` selon notre état d'origine nous obtenons

```
PLAY [creation d'un dossier de travail] *****

TASK [Gathering Facts] *****
ok: [ubuntu-c]
ok: [ubuntu1]
ok: [centos3]
ok: [centos1]
ok: [centos2]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [suppression du dossier de travail ~/tmp (~ home)] *****
ok: [ubuntu-c]
changed: [ubuntu1]
changed: [centos2]
changed: [centos3]
ok: [centos1]
changed: [ubuntu2]
changed: [ubuntu3]

PLAY RECAP *****
centos1      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu-c     : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ansible@ubuntu-c:~/ansibleMelodie/TP/04_miniCMDB/TP/correction/01$ ^C
ansible@ubuntu-c:~/ansibleMelodie/TP/04_miniCMDB/TP/correction/01$
```

si nous relançons nous devons avoir...

```
PLAY RECAP *****
centos1      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu-c     : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ansible@ubuntu-c:~/ansibleMelodie/TP/04_miniCMDB/TP/correction/01$
```

nous sommes bien idempotent !

2. ajoutons une nouvelle tâche pour créer notre dossier

```
- name: creation du dossier de travail ~/tmp (~ home)
  file:
    state: directory
    path: ~/tmp
```

- lançons

```

TASK [suppression du dossier de travail ~/tmp (~ home)] *****
ok: [ubuntu-c]
ok: [ubuntu1]
ok: [centos1]
ok: [centos2]
ok: [centos3]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [creation du dossier de travail ~/tmp (~ home)] *****
changed: [ubuntu-c]
changed: [ubuntu1]
changed: [centos1]
changed: [centos2]
changed: [ubuntu2]
changed: [centos3]
changed: [ubuntu3]

PLAY RECAP *****
centos1      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu-c     : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ansible@ubuntu-c:~/ansibleMelodie/TP/04_minicMDB/TP/correction/01$

```

- relançons

```

TASK [suppression du dossier de travail ~/tmp (~ home)] *****
changed: [ubuntu-c]
changed: [ubuntu1]
changed: [centos3]
changed: [centos2]
changed: [centos1]
changed: [ubuntu2]
changed: [ubuntu3]

TASK [creation du dossier de travail ~/tmp (~ home)] *****
changed: [ubuntu-c]
changed: [ubuntu1]
changed: [centos1]
changed: [centos2]
changed: [centos3]
changed: [ubuntu2]
changed: [ubuntu3]

PLAY RECAP *****
centos1      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu-c     : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ansible@ubuntu-c:~/ansibleMelodie/TP/04_minicMDB/TP/correction/01$

```

[correction](#)

affichage des gather_facts

Nous allons créer 2 nouvelles tâches :

- une pour afficher les gathers facts.
- une autre pour afficher le nom du host (variable ansible)

pour ce faire nous utiliserons le module **debug** que nous avons vu dans les TPs précédents.

1. ajout de l'affichage des gathers facts.

```
- name: affichage des gather_facts
  debug:
    var: ansible_facts
...
```

lançons

```

"selinux": {
  "status": "disabled"
},
"selinux_python_present": true,
"service_mgr": "systemd",
"ssh_host_key_ecdsa_public": "AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAIbmlzdHAyNTYAAABBDvswLWNhmQPIqTzUKsc0odDeExc
"ssh_host_key_ecdsa_public_keytype": "ecdsa-sha2-nistp256",
"ssh_host_key_ed25519_public": "AAAC3NzaC1lZDI1NTE5AAAAILNU9Jy42odb1wRXYNK5umCuUe0aNjkjbm80X+xbBnRc",
"ssh_host_key_ed25519_public_keytype": "ssh-ed25519",
"ssh_host_key_rsa_public": "AAAAB3NzaC1yc2EAAAADAQABAAQGDcCiLiLJ91MKtngIuT6pPRzEls2TmqUSVbvqhJ9dGE9LW0d8PCey
NaOXpdXcarAutEBTGVOJ4RAp7/gSy4KXbsen7gj+kvdet0JBwqnyzKRASUJ8z5g/0HurNC4HV1XF/L/S+wwH2lnEDEwVNY0W2DL/xeMppGXIU4HSUknkwC9
8iv6hJn463twE0igE=",
"ssh_host_key_rsa_public_keytype": "ssh-rsa",
"swapfree_mb": 1329,
"swaptotal_mb": 2047,
"system": "Linux",
"system_capabilities": [
  ""
],
"system_capabilities_enforced": "True",
"system_vendor": "HUAWEI",
"uptime_seconds": 13213,
"user_dir": "/home/ansible",
"user_gecos": "",
"user_gid": 1000,
"user_id": "ansible",
"user_shell": "/bin/bash",
"user_uid": 1000,
"userspace_architecture": "x86_64",
"userspace_bits": "64",
"virtualization_role": "guest",
"virtualization_tech_guest": [
  "docker",
  "container"
],
"virtualization_tech_host": [
  "kvm"
],
"virtualization_type": "docker"
}
}

PLAY RECAP *****
centos1      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu-c     : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3      : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ansible@ubuntu-c:~/ansibleMelodie/TP/04_minicMDB/TP/correction/02$
```

Comme vous pouvez le constater Ansible collecte beaucoup beaucoup d'informations sur les machines.

1. ajout de l'affichage dun nom du host (hostName).

```
- name: affichage du nom de host
  debug:
```

● ● ●

```
TASK [affichage du nom de host] *****
ok: [ubuntu-c] => {
    "ansible_hostname": "ubuntu-c"
}
ok: [centos1] => {
    "ansible_hostname": "centos1"
}
ok: [centos2] => {
    "ansible_hostname": "centos2"
}
ok: [centos3] => {
    "ansible_hostname": "centos3"
}
ok: [ubuntu1] => {
    "ansible_hostname": "ubuntu1"
}
ok: [ubuntu2] => {
    "ansible_hostname": "ubuntu2"
}
ok: [ubuntu3] => {
    "ansible_hostname": "ubuntu3"
}

PLAY RECAP *****
centos1      : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2      : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3      : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu-c     : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1      : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2      : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3      : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ansible@ubuntu-c:~/ansibleMelodie/TP/04_miniCMB/TP/correction/02$
```

Génération du rapport en markdown.

The screenshot shows a terminal window with the following command and output:

```
home > novy > devenvtoansible-lab > ansible home > ubuntu-c < ansible > tmp > *centos1.md > # Rapport du système centos1
1 # Rapport du système centos1 "centos": Unknown word.
2
3 | Interfaces IP | Architecture | OS | Nom du node |
4 | :-----: | :-----: | :-----: | :-----: |
5 | eth0 | x86_64 | CentOS 8 NA | centos1 | "centos": Unknown word.
6
```

On the right, a preview of the generated HTML report is shown, titled "Rapport du système centos1". It contains a table with the following data:

Interfaces IP	Architecture	OS	Nom du node
eth0	x86_64	CentOS 8 NA	centos1

- copier le dossier via la commande. `cp -R /home/ansible/ansibleMelodie/TP/04_miniCMDB/TP/correction/03/templates .`
- visualiser le fichier
 - il est écrit au format markdown mais cela pourrait être de l'html.
 - nos variables sont incluses avec la syntaxe `{{ma_variable}}`
`{{ ansible_default_ipv4.alias }}`

- VOLUBIS 6 / 9

le nouveau sera déposé dans notre répertoire de travail (~/.tmp) et se nommera `{{ inventory_hostname }}`.md Remarque l'usage des doubles accolades pour indiquer qu'il faut remplacer par le contenu de la variable.

```
- name: Génération du rapport en markdown
  template:
    src: templates/report.md.j2
    dest: "~/.tmp/{{ inventory_hostname }}.md"
```

lançons

```

TASK [Génération du rapport en markdown] *****
changed: [ubuntu-c]
changed: [ubuntu1]
changed: [centos3]
changed: [centos1]
changed: [centos2]
changed: [ubuntu2]
changed: [ubuntu3]

PLAY RECAP *****
centos1      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu-c     : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ansible@ubuntu-c:~/ansibleMelodie/TP/04_miniCMDB/TP/correction/03$

```

1. consulter le rapport sur **Centos1** via l'éditeur ou via la commande

```

centos1.md - ansibleMelodie - Visual Studio Code
... ① README.md M • centos1.md X report.md.j2 .../03/... U ! 20_miniCMDB.yml M report.i
home > novy > diveintoansible-lab > ansible_home > centos1 > ansible > tmp > centos1.md > # Rapport du système centos1
1 # Rapport du système centos1 "centos": Unknown word.
2
3 | Interfaces IP | Architecture | OS | Nom du node |
4 | :-----: | :-----: | :-----: | :-----: |
5 | eth0 | x86_64 | CentOS 8 NA | centos1 | "centos": Unknown word.
6

```

Collecte des rapports sur la machine de contrôle (**ubuntu-c**)

Nous allons devoir créer à nouveau jeu à notre playbook. En effet nous ne devons porter nos actions que sur les machines managées et non sur toutes, puisque cela inclût la machine de contrôle.

1. Création d'un nouveau jeu avec la portée du groupe **managed** de notre inventaire.

```
- name: download des rapports des machines hôtes
  hosts: managed
  tasks:
```

```

changed: [ubuntu3]

PLAY [download des rapports des machines hôtes] *****

TASK [Gathering Facts] *****
ok: [ubuntu2]
ok: [ubuntu1]
ok: [centos2]
ok: [centos3]
ok: [ubuntu3]
ok: [centos1]

PLAY RECAP *****
centos1      : ok=7    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2      : ok=7    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

la collecte des gathers facts ne sert à rien, supprimons cette tâche. ajoutons la clause `gather_facts: no`

```

- name: download des rapports des machines hôtes
  hosts: managed
  gather_facts: no
  tasks:
    ...

```

```

PLAY [download des rapports des machines hôtes] *****

PLAY RECAP *****
centos1      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu-c     : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ansible@ubuntu-c:~/ansibleMelodie/TP/04_miniCMDB/TP/correction/03$

```

1. ajout d'une tâche à ce nouveau jeu pour rapatrier les rapports dans le dossier `~/tmp` de la machine de contrôle.

```

- name: recup ~/tmp/*.html ==> localhost ~/tmp
  fetch:
    src: "~/tmp/{{ inventory_hostname }}.md"
    dest: ~/tmp
    flat: true
    validate_checksum: false

```


lançons

```
}

TASK [Génération du rapport en markdown] *****
changed: [ubuntu-c]
changed: [ubuntu1]
changed: [centos2]
changed: [centos1]
changed: [centos3]
changed: [ubuntu2]
changed: [ubuntu3]

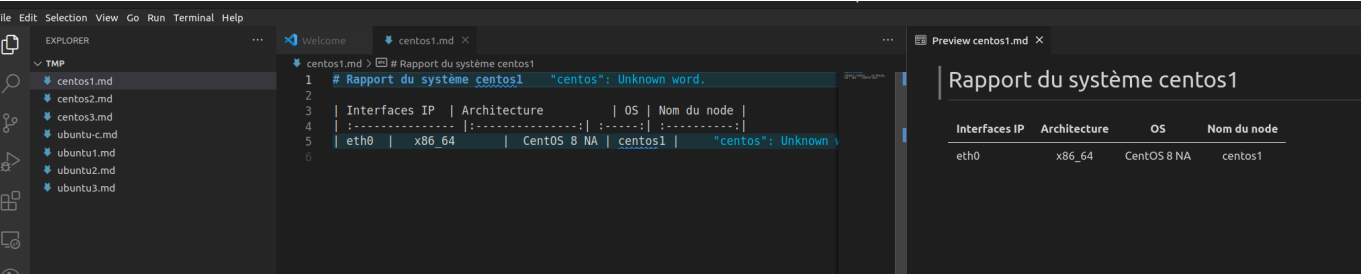
PLAY [download des rapports des machines hôtes] *****

TASK [recup é/tmp/*.html ==> localhost ~/tmp] *****
changed: [ubuntu2]
changed: [ubuntu1]
changed: [centos3]
changed: [centos1]
changed: [centos2]
changed: [ubuntu3]

PLAY RECAP *****
centos1      : ok=7  changed=4  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
centos2      : ok=7  changed=4  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
centos3      : ok=7  changed=4  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
ubuntu-c     : ok=6  changed=3  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
ubuntu1      : ok=7  changed=4  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
ubuntu2      : ok=7  changed=4  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
ubuntu3      : ok=7  changed=4  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

ansible@ubuntu-c:~/ansibleMelodie/TP/04_miniCMDB/TP/correction/03$
```

contrôlons sur ubuntu-c



Conclusion et feed-back

Correction

💡💡💡💡 Idées

-