Version - **2024.01**

Dernière mise-à-jour : 2024/06/10 11:46

LDF905 - Automatiser l'Administration Système de CentOS 8

Contenu du Module

LAB #1 - Gestion des Utilisateurs et des Mots de Passe

Connectez-vous à la machine virtuelle **CentOS_8** :

```
trainee@debian11:~$ ssh -l trainee 10.0.2.45
The authenticity of host '10.0.2.45 (10.0.2.45)' can't be established.
ECDSA key fingerprint is SHA256:Q7T/CP0SLiMbMAIgVzTuEHegYS/spPE5zzQchCHD5Vw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.45' (ECDSA) to the list of known hosts.
trainee@10.0.2.45's password: trainee
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Thu Feb 16 16:46:22 2023 from 10.0.2.1

[trainee@centos8 ~]$ ls -la | grep .ssh

[trainee@centos8 ~]$ mkdir .ssh

[trainee@centos8 ~]$ chmod 700 .ssh

[trainee@centos8 ~]$ ls -la | grep .ssh
drwx------.  2 trainee trainee      4096 Sep 20 09:50 .ssh

[trainee@centos8 ~]$ exit
logout
Connection to 10.0.2.45 closed.
```

1.1 - Configurer SSH et sudo

Copiez le fichier **.ssh/authorized_keys** vers **/home/trainee/.ssh/** :

```
trainee@debian11:~$ scp .ssh/authorized_keys trainee@10.0.2.45:/home/trainee/.ssh/authorized_keys
trainee@10.0.2.45's password: trainee
authorized_keys                                   100%  846     1.3MB/s   00:00
```

Testez la connexion SSH :

```
trainee@debian11:~$ ssh -l trainee 10.0.2.45
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Wed Sep 20 09:49:19 2023 from 10.0.2.46
[trainee@centos8 ~]$
```

Créez le fichier **/etc/sudoers.d/ansible_users** :

```
[trainee@centos8 ~]$ su -
Password: fenestros

[root@centos8 ~]# vi /etc/sudoers.d/ansible_users

[root@centos8 ~]# cat /etc/sudoers.d/ansible_users
trainee ALL=(ALL)       NOPASSWD:ALL

[root@centos8 ~]# chmod 440 /etc/sudoers.d/ansible_users

[root@centos8 ~]# ls -l /etc/sudoers.d/ansible_users
-r--r-----. 1 root root 37 Sep 20 09:56 /etc/sudoers.d/ansible_users
```

Vérifiez que vous pouvez devenir l'utilisateur root sans saisir de mot de passe :

```
[root@centos8 ~]# exit
logout

[trainee@centos8 ~]$ sudo su -

[root@centos8 ~]# exit
logout

[trainee@centos8 ~]$ exit
logout
Connection to 10.0.2.45 closed.

trainee@debian11:~$
```

1.2 - Configurer Ansible

Pour gérer les utilisateurs, nous devons faire appel au module **users**. Pour obtenir de l'information sur ce module, utilisez la commande **ansible-doc** :

```
trainee@debian11:~$ ansible-doc user
> ANSIBLE.BUILTIN.USER    (/usr/lib/python3/dist-packages/ansible/modules/user.py)

        Manage user accounts and user attributes. For Windows targets,
        use the [ansible.windows.win_user] module instead.

OPTIONS (= is mandatory):

- append
        If `yes', add the user to the groups specified in `groups'.
        If `no', user will only be added to the groups specified in
        `groups', removing them from all other groups.
        [Default: False]
        type: bool
```

```
- authorization
        Sets the authorization of the user.
        Does nothing when used with other platforms.
        Can set multiple authorizations using comma separation.
        To delete all authorizations, use `authorization=''.
        Currently supported on Illumos/Solaris.
        [Default: (null)]
        type: str
        version_added: 2.8
:
```

Créez maintenant les répertoires **host_vars** et **group_vars** dans le rôle **users** :

```
trainee@debian11:~$ mkdir -p ansible/users/host_vars

trainee@debian11:~$ mkdir ansible/users/group_vars
```

Placez-vous dans le rôle :

```
trainee@debian11:~$ cd ansible/users/

trainee@debian11:~/ansible/users$
```

Constatez le fichier de configuration d'Ansible actuellement utilisé :

```
trainee@debian11:~/ansible/users$ ansible --version
ansible 2.10.8
  config file = None
  configured module search path = ['/home/trainee/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.9.2 (default, Feb 28 2021, 17:03:44) [GCC 10.2.1 20210110]
```

Puisque la valeur est **none**, créez le fichier **ansible.cfg** :

```
trainee@debian11:~/ansible/users$ vi ansible.cfg

trainee@debian11:~/ansible/users$ cat ansible.cfg
[defaults]
inventory = inventory
remote_user = trainee
[privilege_escalation]
become = true
become_method = sudo
become_user = root
```

Vérifiez la prise en compte de ce fichier :

```
trainee@debian11:~/ansible/users$ ansible --version
ansible 2.10.8
  config file = /home/trainee/ansible/users/ansible.cfg
  configured module search path = ['/home/trainee/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.9.2 (default, Feb 28 2021, 17:03:44) [GCC 10.2.1 20210110]
```

```
trainee@debian11:~/ansible/users$ ansible-config view
[defaults]
inventory = inventory
remote_user = trainee
[privilege_escalation]
become = true
become_method = sudo
become_user = root
```

Pour voir l'ensemble des variables possibles, utilisez la commande **ansible-config list** :

```
trainee@debian11:~/ansible/users$ ansible-config list
ACTION_WARNINGS:
  default: true
  description:
  - By default Ansible will issue a warning when received from a task action (module
    or action plugin)
  - These warnings can be silenced by adjusting this setting to False.
  env:
  - name: ANSIBLE_ACTION_WARNINGS
  ini:
  - key: action_warnings
    section: defaults
  name: Toggle action warnings
  type: boolean
  version_added: '2.5'
AGNOSTIC_BECOME_PROMPT:
  default: true
  description: Display an agnostic become prompt instead of displaying a prompt conta>
    the command line supplied become method
  env:
  - name: ANSIBLE_AGNOSTIC_BECOME_PROMPT
  ini:
  - key: agnostic_become_prompt
    section: privilege_escalation
:
```

Pour consulter l'ensemble des valeurs actuelles des variables, utilisez la commande **ansible-config dump** :

```
trainee@debian11:~/ansible/users$ ansible-config dump
ACTION_WARNINGS(default) = True
AGNOSTIC_BECOME_PROMPT(default) = True
ALLOW_WORLD_READABLE_TMPFILES(default) = False
ANSIBLE_CONNECTION_PATH(default) = None
ANSIBLE_COW_PATH(default) = None
```

```
ANSIBLE_COW_SELECTION(default) = default
ANSIBLE_COW_WHITELIST(default) = ['bud-frogs', 'bunny', 'cheese', 'daemon', 'default'>
ANSIBLE_FORCE_COLOR(default) = False
ANSIBLE_NOCOLOR(default) = False
ANSIBLE_NOCOWS(default) = False
ANSIBLE_PIPELINING(default) = False
ANSIBLE_SSH_ARGS(default) = -C -o ControlMaster=auto -o ControlPersist=60s
ANSIBLE_SSH_CONTROL_PATH(default) = None
ANSIBLE_SSH_CONTROL_PATH_DIR(default) = ~/.ansible/cp
ANSIBLE_SSH_EXECUTABLE(default) = ssh
ANSIBLE_SSH_RETRIES(default) = 0
ANY_ERRORS_FATAL(default) = False
BECOME_ALLOW_SAME_USER(default) = False
BECOME_PLUGIN_PATH(default) = ['/home/trainee/.ansible/plugins/become', '/usr/share/a>
CACHE_PLUGIN(default) = memory
CACHE_PLUGIN_CONNECTION(default) = None
CACHE_PLUGIN_PREFIX(default) = ansible_facts
CACHE_PLUGIN_TIMEOUT(default) = 86400
:
```

Pour ne consulter que les valeurs modifiées, utilisez l'option **–only-changed** :

```
trainee@debian11:~/ansible/users$ ansible-config dump  --only-changed
DEFAULT_BECOME(/home/trainee/ansible/users/ansible.cfg) = True
DEFAULT_BECOME_METHOD(/home/trainee/ansible/users/ansible.cfg) = sudo
DEFAULT_BECOME_USER(/home/trainee/ansible/users/ansible.cfg) = root
DEFAULT_HOST_LIST(/home/trainee/ansible/users/ansible.cfg) = ['/home/trainee/ansible/>
DEFAULT_REMOTE_USER(/home/trainee/ansible/users/ansible.cfg) = trainee
(END)
```

Pour informer Ansible du type de connexion à utiliser pour contacter le contrôleur, injectez les texte **ansible_connection: local** dans le fichier **host_vars/10.0.2.46** :

```
trainee@debian11:~/ansible/users$ echo "ansible_connection: local" > host_vars/10.0.2.46
```

Créez le fichier **inventory** :

```
trainee@debian11:~/ansible/users$ vi inventory
trainee@debian11:~/ansible/users$ cat inventory
[centos]
10.0.2.45
```

1.3 - Créer un Utilisateur Unique

Créez ensuite le fichier **user1.yml** :

```
trainee@debian11:~/ansible/users$ vi user1.yml

trainee@debian11:~/ansible/users$ cat user1.yml
---
- name: User Creation
  hosts: all
  gather_facts: false
  become: true
  tasks:
    - name: Create User
      user:
        name: 'jodi'
```

Vérifiez la syntaxe du fichier créé :

```
trainee@debian11:~/ansible/users$ ansible-playbook user1.yml --syntax

playbook: user1.yml
```

En exécutant le playbook, on peut constater qu'une modification a eu lieu dans la VM 10.0.2.45 :

```
trainee@debian11:~/ansible/users$ ansible-playbook user1.yml

PLAY [User Creation]
*****************************************************************************************************
*****************************************************************************

TASK [Create User]
*****************************************************************************************************
*****************************************************************************
changed: [10.0.2.45]

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

Vérifiez que cette modification est celle attendue :

```
trainee@debian11:~/ansible/users$ ansible all -m command -a 'getent passwd jodi'
10.0.2.45 | CHANGED | rc=0 >>
jodi:x:1001:1001::/home/jodi:/bin/bash
```

1.4 - Supprimer un Utilisateur Unique

Modifiez le fichier **user1.yml** :

```
trainee@debian11:~/ansible/users$ vi user1.yml

trainee@debian11:~/ansible/users$ cat user1.yml
---
- name: Delete User
```

```
  hosts: all
  gather_facts: false
  become: true
  tasks:
    - name: Delete User Account
      user:
        name: 'jodi'
        state: 'absent'
        remove: true
```

Important : Notez l'utilisation de **remove** qui permet de supprimer le répertoire personnel de l'utilisateur, son spool de mail et ses cronjobs éventuels.

Exécutez le playbook :

```
trainee@debian11:~/ansible/users$ ansible-playbook user1.yml

PLAY [Delete User] *************************************************************

TASK [Delete User Account] *****************************************************
changed: [10.0.2.45]

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

1.5 - Créer de Multiples Utilisateurs

Modifiez le fichier **user1.yml** :

```
trainee@debian11:~/ansible/users$ vi user1.yml

trainee@debian11:~/ansible/users$ cat user1.yml
---
- name: Multiple Users
  hosts: all
  become: true
  tasks:
    - name: create multiple accounts
      user:
        name: "{{ item }}"
      loop:
        - user1
        - user2
        - user3
```

Important : Notez l'utilisation de **“**

item

“ et **loop** qui permettent l'exécution d'un boucle pour la création des trois utilisateurs. Dans le cas où la version d'Ansible < 2.5, il convient de remplacer **loop** avec **with_items**.

Exécutez le playbook :

```
trainee@debian11:~/ansible/users$ ansible-playbook user1.yml

PLAY [Multiple Users] **********************************************************

TASK [Gathering Facts] *********************************************************
```

```
ok: [10.0.2.45]

TASK [create multiple accounts] ************************************************
changed: [10.0.2.45] => (item=user1)
changed: [10.0.2.45] => (item=user2)
changed: [10.0.2.45] => (item=user3)

PLAY RECAP *********************************************************************
10.0.2.45                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Constatez la création des trois utilisateurs :

```
trainee@debian11:~/ansible/users$ ansible all -m command -a 'getent passwd user1'
10.0.2.45 | CHANGED | rc=0 >>
user1:x:1001:1001::/home/user1:/bin/bash

trainee@debian11:~/ansible/users$ ansible all -m command -a 'getent passwd user2'
10.0.2.45 | CHANGED | rc=0 >>
user2:x:1002:1002::/home/user2:/bin/bash

trainee@debian11:~/ansible/users$ ansible all -m command -a 'getent passwd user3'
10.0.2.45 | CHANGED | rc=0 >>
user3:x:1003:1003::/home/user3:/bin/bash
```

Constatez la création des répertoires personnels :

```
trainee@debian11:~/ansible/users$ ansible all -m command -a 'ls -l /home'
10.0.2.45 | CHANGED | rc=0 >>
total 32
drwx------.  2 root    root    16384 Jul 19  2021 lost+found
drwxr-xr-x. 19 trainee trainee  4096 Sep 20 09:59 trainee
drwx------.  3 user1   user1    4096 Sep 20 11:37 user1
drwx------.  3 user2   user2    4096 Sep 20 11:37 user2
```

```
drwx------.  3 user3   user3    4096 Sep 20 11:37 user3
```

1.6 - Supprimer de Multiples Utilisateurs

Modifiez le fichier **user1.yml** :

```
trainee@debian11:~/ansible/users$ vi user1.yml

trainee@debian11:~/ansible/users$ cat user1.yml
---
- name: Multiple Users
  hosts: all
  become: true
  tasks:
    - name: delete multiple accounts
      user:
        name: "{{ item }}"
        state: 'absent'
        remove: true
      loop:
        - user1
        - user2
        - user3
```

Exécutez le playbook :

```
trainee@debian11:~/ansible/users$ ansible-playbook user1.yml

PLAY [Multiple Users]
**********************************************************************************************************
**********************************************************************

TASK [Gathering Facts]
```

```
**********************************************************************************************************************
**********************************************************************
ok: [10.0.2.45]

TASK [delete multiple accounts]
**********************************************************************************************************************
**********************************************************************
changed: [10.0.2.45] => (item=user1)
changed: [10.0.2.45] => (item=user2)
changed: [10.0.2.45] => (item=user3)

PLAY RECAP
**********************************************************************************************************************
**********************************************************************************
10.0.2.45                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Vérifiez la suppression d'un des utilisateurs ainsi que son répertoire personnel :

```
trainee@debian11:~/ansible/users$ ansible all -m command -a 'getent passwd user1'
10.0.2.45 | FAILED | rc=2 >>
non-zero return code

trainee@debian11:~/ansible/users$ ansible all -m command -a 'ls -l /home'
10.0.2.45 | CHANGED | rc=0 >>
total 20
drwx------.  2 root    root    16384 Jul 19  2021 lost+found
drwxr-xr-x. 19 trainee trainee  4096 Sep 20 09:59 trainee
```

1.7 - Créer un Utilisateur en utilisant une Variable

Modifiez le fichier **user1.yml** :

```
trainee@debian11:~/ansible/users$ vi user1.yml

trainee@debian11:~/ansible/users$ cat user1.yml
---
- name: User Creation from Variable
  hosts: all
  gather_facts: false
  become: true
  tasks:
    - name: Create User Account
      user:
        name: "{{ user_name }}"
```

Important : Notez l'utilisation de la variable **user_name**.

Exécutez le playbook en passant la valeur de la variable **user_name** sur la ligne de commande :

```
trainee@debian11:~/ansible/users$ ansible-playbook -e "user_name=user4" user1.yml

PLAY [User Creation from Variable] *********************************************

TASK [Create User Account] *****************************************************
changed: [10.0.2.45]

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

Vérifiez la création de l'utilisateur :

```
trainee@debian11:~/ansible/users$ ansible all -m command -a 'getent passwd user4'
```

```
10.0.2.45 | CHANGED | rc=0 >>
user4:x:1001:1001::/home/user4:/bin/bash
```

1.8 - Gestion des Mots de Passe

Jusqu'à maintenant, nous nous ne sommes pas occupé de la création des mots de passe. Modifiez donc le fichier **user1.yml** :

```
trainee@debian11:~/ansible/users$ vi user1.yml

trainee@debian11:~/ansible/users$ cat user1.yml
---
- name: Manage Users
  hosts: all
  gather_facts: false
  become: true
  tasks:
    - name: Create User Account
      user:
        name: "{{ user_name }}"
        password: "{{ 'trainee' | password_hash('sha512', 'A512') }}"
        update_password: on_create
      when: user_create == 'yes'

    - name: Delete User Account
      user:
        name: "{{ user_name }}"
        state: 'absent'
        remove: true
      when: user_create == 'no'
```

Important : Notez la création de deux tâches : **Create User Account** et

Delete User Account. L'exécution de la tâche est conditionnée par la valeur de la variable **user_create** qui sera passée sur la ligne de commande lors de l'exécution du playbook. Notez que le mot de passe **trainee** est haché grâce à l'utilisation de **password_hash('algorithme', 'salt')** où salt est du texte aléatoire. Dernièrement **update_password: on_create** implique que le mot de passe sera créé pour l'utilisateur spécifié par la valeur de la variable **user_name** uniquement lors de sa création.

Créez donc l'utilisateur **user1** :

```
trainee@debian11:~/ansible/users$ ansible-playbook -e "user_create=yes user_name=user1" user1.yml

PLAY [Manage Users]
*********************************************************************************************************
*********************************************************************

TASK [Create User Account]
*********************************************************************************************************
*********************************************************************
changed: [10.0.2.45]

TASK [Delete User Account]
*********************************************************************************************************
*********************************************************************
skipping: [10.0.2.45]

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

Modifiez la valeur de la variable **user_create** pour supprimer l'utilisateur **user1.yml** :

```
trainee@debian11:~/ansible/users$ ansible-playbook -e "user_create=no user_name=user1" user1.yml

PLAY [Manage Users]
**********************************************************************************************************
**************************************************************************

TASK [Create User Account]
**********************************************************************************************************
**************************************************************************
skipping: [10.0.2.45]

TASK [Delete User Account]
**********************************************************************************************************
**************************************************************************
changed: [10.0.2.45]

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

1.9 - Créer un Utilisateur Dédié pour Ansible

Créez le fichier **devops.yml** :

```
trainee@debian11:~/ansible/users$ vi devops.yml

trainee@debian11:~/ansible/users$ cat devops.yml
---
- name: Deploy devops account
```

```
hosts: all
become: true
tasks:
  - name: create account
    user:
      name: devops

  - name: sudo access
    copy:
      dest: /etc/sudoers.d/devops
      content: 'devops ALL=(ALL) NOPASSWD: ALL'
      validate: /usr/sbin/visudo -cf %s

  - name: ssh key
    authorized_key:
      user: devops
      state: present
      manage_dir: true
      key: "{{ lookup( 'file', '/home/trainee/.ssh/id_rsa.pub') }}"
```

Important : Notez la création de trois tâches : **create account**, **sudo access** et **ssh key**. Dans la tâche **create account**, aucun mot de passe n'est spécifié car l'utilisateur s'authentifiera par clefs. Dans la tâche **sudo access**, la chaîne **devops ALL=(ALL) NOPASSWD: ALL** est injecté dans le fichier **/etc/sudoers.d/devops** dont le contenu sera ensuite validé par l'utilisation de la commande **/usr/sbin/visudo -cf %s** ou **%s** et **/etc/sudoers.d/devops**. Dans la tâche **ssh key**, le module **authorized_key** est utilisé. Ce module permet la création, si besoin, du répertoire **.ssh** de l'utilisateur **devops** qui doit être **present** grâce à **manage_dir: true**. La valeur de **key** de devops est créer par la copie du contenu du fichier **/home/trainee/.ssh/id_rsa.pub**.

Vérifiez la syntaxe du fichier :

```
trainee@debian11:~/ansible/users$ ansible-playbook devops.yml --syntax

playbook: devops.yml
```

Exécutez le playbook :

```
trainee@debian11:~/ansible/users$ ansible-playbook devops.yml

PLAY [Deploy devops account]
*********************************************************************************************************************
*****************************************************************

TASK [Gathering Facts]
*********************************************************************************************************************
*****************************************************************
ok: [10.0.2.45]

TASK [create account]
*********************************************************************************************************************
*****************************************************************
changed: [10.0.2.45]

TASK [sudo access]
*********************************************************************************************************************
*****************************************************************
changed: [10.0.2.45]

TASK [ssh key]
*********************************************************************************************************************
*****************************************************************
changed: [10.0.2.45]
```

```
PLAY RECAP
*********************************************************************************************************
*********************************************************************************
10.0.2.45                  : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Testez la connexion SSH pour le compte **devops** :

```
trainee@debian11:~/ansible/users$ ssh -l devops 10.0.2.45
Activate the web console with: systemctl enable --now cockpit.socket

[devops@centos8 ~]$ whoami
devops

[devops@centos8 ~]$ exit
logout
Connection to 10.0.2.45 closed.
```

LAB #2 - Gestion des Services

2.1 - Interdire des Connexions par Root

Créez le fichier **ssh.yaml** :

```
trainee@debian11:~/ansible/users$ vi ssh.yml

trainee@debian11:~/ansible/users$ cat ssh.yml
---
- name: sshd
  hosts: all
  gather_facts: false
```

```
handlers:
  - name: restart_sshd
    systemd:
      name: sshd.service
      state: restarted
tasks:
  - name: Enable SSHD
    systemd:
      name: sshd
      enabled: true
      state: started

  - name: No Root
    lineinfile:
      path: /etc/ssh/sshd_config
      regexp: '^PermitRootLogin'
      insertafter: '#LoginGraceTime'
      line: 'PermitRootLogin no'
    notify: restart_sshd
```

Important : Notez la création d'un **handler** nommé **restart_sshd**. Deux tâches sont créées : **Enable SSHD** et **No Root**. **Enable SSHD** sert à activer et à démarrer le service sshd en utilisant le module **systemd** tandis que **No Root** édite le fichier **/etc/ssh/sshd_config** afin d'empêcher les connexions directes par l'utilisateur **root**. Notez l'utilisation du module **lineinfile** qui recherche une expression régulière **regexp** dans le fichier **/etc/ssh/sshd_config** et qui modifie la ligne en fonction de la valeur de **line**. La valeur d'**insertafter** spécifie la où doit être insérer la ligne si elle n'existe pas dans le fichier. Dernièrement en cas de modification du fichier le handler est appelé par **notify**.

Consultez la valeur de **PermitRootLogin** avant l'exécution du playbook :

```
trainee@debian11:~/ansible/users$ ansible all -m command -a 'grep PermitRootLogin /etc/ssh/sshd_config'
10.0.2.45 | CHANGED | rc=0 >>
PermitRootLogin yes
# the setting of "PermitRootLogin without-password".
```

Exécutez le playbook :

```
trainee@debian11:~/ansible/users$ ansible-playbook ssh.yml

PLAY [sshd]
**********************************************************************************************************
**********************************************************************

TASK [Enable SSHD]
**********************************************************************************************************
**********************************************************************
ok: [10.0.2.45]

TASK [No Root]
**********************************************************************************************************
**********************************************************************
changed: [10.0.2.45]

RUNNING HANDLER [restart_sshd]
**********************************************************************************************************
**********************************************************************
changed: [10.0.2.45]

PLAY RECAP
**********************************************************************************************************
**********************************************************************
10.0.2.45                  : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
```

```
ignored=0
```

Consultez la valeur de **PermitRootLogin** après l'exécution du playbook :

```
trainee@debian11:~/ansible/users$ ansible all -m command -a 'grep PermitRootLogin /etc/ssh/sshd_config'
10.0.2.45 | CHANGED | rc=0 >>
PermitRootLogin no
# the setting of "PermitRootLogin without-password".
```

Important : Notez la modification de la valeur de **PermitRootLogin**.

Dernièrement, exécutez de nouveau le playbook :

```
trainee@debian11:~/ansible/users$ ansible-playbook ssh.yml

PLAY [sshd] ********************************************************************************

TASK [Enable SSHD] *************************************************************************
ok: [10.0.2.45]

TASK [No Root] *****************************************************************************
ok: [10.0.2.45]

PLAY RECAP *********************************************************************************
10.0.2.45                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Important : Notez que le handler n'a pas été appelé car le fichier **/etc/ssh/sshd_config** n'a pas été modifié.

LAB #3 - Gestion du Stockage

3.1 - Préparation

Commencez par créer le répertoire **/home/ansible/storage** :

```
trainee@debian11:~/ansible/users$ cd ..

trainee@debian11:~/ansible$ mkdir storage
```

Copiez ensuite les fichiers **/home/ansible/users/inventory** et **/home/ansible/users/ansible.cfg** dans le répertoire **/home/ansible/storage** :

```
trainee@debian11:~/ansible$ cp users/{inventory,ansible.cfg} storage/

trainee@debian11:~/ansible$ cd storage

trainee@debian11:~/ansible/storage$ ls
ansible.cfg  inventory
```

Consultez ensuite l'organisation des disques et des partitions de la VM **CentOS_8** :

```
trainee@debian11:~/ansible/storage$ ansible all -m command -a 'lsblk'
10.0.2.45 | CHANGED | rc=0 >>
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                   8:0    0   32G  0 disk
├─sda1                8:1    0    1G  0 part /boot
└─sda2                8:2    0   31G  0 part
  ├─cl_centos8-root 253:0    0 27.8G  0 lvm  /
  └─cl_centos8-swap 253:1    0  3.2G  0 lvm  [SWAP]
sdb                   8:16   0    4G  0 disk
sdc                   8:32   0   64G  0 disk
└─sdc1                8:33   0   64G  0 part /home
```

```
sdd                      8:48   0   32G  0 disk
sr0                     11:0    1 1024M  0 rom
```

Important : Notez la présence du disque **sdb** qui est inutilisé.

3.2 - Création des Partitions

Créez ensuite le fichier **storage.yml** :

```
trainee@debian11:~/ansible/storage$ vi storage.yml

trainee@debian11:~/ansible/storage$ cat storage.yml
---
- name: storage
  hosts: all
  become: true
  gather_facts: false
  tasks:
    - name: part1
      parted:
        device: /dev/sdb
        part_start: 0%
        part_end: 50%
        number: 1
        state: present

    - name: part2
      parted:
        device: /dev/sdb
        part_start: 50%
```

```
        part_end: 100%
        number: 2
        state: present
        flags: [ lvm ]
```

Important : Notez l'utilisation du module **parted**. Le **device** est **/dev/sdb**. La partition commence au début du disque. La taille peut être spécifiée en GB, MB, secteurs ou en pourcentage. Dans le cas de la partition 2, le drapeau de la partition est fixé à **lvm**.

3.3 - Création des VG et LV

Éditez le fichier **storage.yml** :

```
trainee@debian11:~/ansible/storage$ vi storage.yml

trainee@debian11:~/ansible/storage$ cat storage.yml
---
- name: play1
  hosts: all
  become: true
  gather_facts: false
  tasks:
    - name: part1
      parted:
        device: /dev/sdb
        part_start: 0%
        part_end: 50%
        number: 1
        state: present
```

```
    - name: part2
      parted:
        device: /dev/sdb
        part_start: 50%
        part_end: 100%
        number: 2
        state: present
        flags: [ lvm ]

    - name: vg
      lvg:
        vg: vg1
        pvs: /dev/sdb2

    - name: lv
      lvol:
        lv: lv1
        vg: vg1
        size: 100%FREE
        shrink: false
```

Important : Notez l'utilisation du module **lvg** pour créer le groupe de volumes **vg1** qui utilise le volume physique **/dev/sdb2**. L'exécution de la commande **pvcreate** est accomplie par l'entrée **pvs**. Notez ensuite l'utilisation du module **lvol** pour créer le volume logique **lv1**. La taille est fixée à 100% de la taille disponible. L'entrée **shrink: false** indique que la taille du volume ne peut pas être réduite.

3.4 - Création des Filesystems

Éditez de nouveau le fichier **storage.yml** :

```
trainee@debian11:~/ansible/storage$ vi storage.yml

trainee@debian11:~/ansible/storage$ cat storage.yml
---
- name: play1
  hosts: all
  become: true
  gather_facts: false
  tasks:
    - name: part1
      parted:
        device: /dev/sdb
        part_start: 0%
        part_end: 50%
        number: 1
        state: present

    - name: part2
      parted:
        device: /dev/sdb
        part_start: 50%
        part_end: 100%
        number: 2
        state: present
        flags: [ lvm ]

    - name: vg
      lvg:
        vg: vg1
```

```
        pvs: /dev/sdb2

    - name: lv
      lvol:
        lv: lv1
        vg: vg1
        size: 100%FREE
        shrink: false

    - name: sdb1
      filesystem:
        fstype: xfs
        dev: /dev/sdb1

    - name: lv
      filesystem:
        fstype: xfs
        dev: /dev/vg1/lv1
```

Important : Notez l'utilisation du module **filesystem** pour créer des systèmes de fichiers de type **xfs** sur **/dev/sdb1** et **/dev/vg1/lv1**.

3.5 - Création des Points de Montage

Éditez ensuite le fichier **storage.yml** :

```
trainee@debian11:~/ansible/storage$ vi storage.yml
trainee@debian11:~/ansible/storage$ cat storage.yml
---
- name: play1
```

```
hosts: all
become: true
gather_facts: false
tasks:
  - name: part1
    parted:
      device: /dev/sdb
      part_start: 0%
      part_end: 50%
      number: 1
      state: present

  - name: part2
    parted:
      device: /dev/sdb
      part_start: 50%
      part_end: 100%
      number: 2
      state: present
      flags: [ lvm ]

  - name: vg
    lvg:
      vg: vg1
      pvs: /dev/sdb2

  - name: lv
    lvol:
      lv: lv1
      vg: vg1
      size: 100%FREE
      shrink: false

  - name: sdb1
```

```
      filesystem:
        fstype: xfs
        dev: /dev/sdb1

    - name: lv
      filesystem:
        fstype: xfs
        dev: /dev/vg1/lv1

    - name: dir1
      file:
        path: "{{ item }}"
        state: directory
      loop:
        - /data
        - /data/sales
        - /data/marketing
```

Important : Notez l'utilisation du module **file** pour créer les répertoires **/data**, **/data/sales** et **/data/marketing** grâce au mot clef **loop**.

3.6 - Monter les Partitions

Éditez une dernière fois le fichier **storage.yml** :

```
trainee@debian11:~/ansible/storage$ vi storage.yml

trainee@debian11:~/ansible/storage$ cat storage.yml
---
- name: play1
```

```
hosts: all
become: true
gather_facts: false
tasks:
  - name: part1
    parted:
      device: /dev/sdb
      part_start: 0%
      part_end: 50%
      number: 1
      state: present

  - name: part2
    parted:
      device: /dev/sdb
      part_start: 50%
      part_end: 100%
      number: 2
      state: present
      flags: [ lvm ]

  - name: vg
    lvg:
      vg: vg1
      pvs: /dev/sdb2

  - name: lv
    lvol:
      lv: lv1
      vg: vg1
      size: 100%FREE
      shrink: false

  - name: sdb1
```

```
    filesystem:
      fstype: xfs
      dev: /dev/sdb1

  - name: lv
    filesystem:
      fstype: xfs
      dev: /dev/vg1/lv1

  - name: dir1
    file:
      path: "{{ item }}"
      state: directory
    loop:
      - /data
      - /data/sales
      - /data/marketing

  - name: mount sales
    mount:
      path: /data/sales
      src: /dev/sdb1
      fstype: xfs
      state: mounted

  - name: mount marketing
    mount:
      path: /data/marketing
      src: /dev/vg1/lv1
      fstype: xfs
      state: mounted
```

Important : Notez l'utilisation du module **mount** pour monter les

filesystems **/dev/sdb1** et **/dev/vg1/lv1** sur les répertoires **/data/sales** et **/data/marketing** respectivement.

3.7 - Exécution du Playbook

Exécutez maintenant le playbook **storage.yml** :

```
trainee@debian11:~/ansible/storage$ ansible-playbook storage.yml

PLAY [play1]
*********************************************************************************************************************
*************************************************************************

TASK [part1]
*********************************************************************************************************************
*************************************************************************
changed: [10.0.2.45]

TASK [part2]
*********************************************************************************************************************
*************************************************************************
changed: [10.0.2.45]

TASK [vg]
*********************************************************************************************************************
*************************************************************************
changed: [10.0.2.45]

TASK [lv]
*********************************************************************************************************************
*************************************************************************
```

```
changed: [10.0.2.45]

TASK [sdb1]
*********************************************************************************************************************
*********************************************************************************
changed: [10.0.2.45]

TASK [lv]
*********************************************************************************************************************
*********************************************************************************
changed: [10.0.2.45]

TASK [dir1]
*********************************************************************************************************************
*********************************************************************************
changed: [10.0.2.45] => (item=/data)
changed: [10.0.2.45] => (item=/data/sales)
changed: [10.0.2.45] => (item=/data/marketing)

TASK [mount sales]
*********************************************************************************************************************
*********************************************************************************
changed: [10.0.2.45]

TASK [mount marketing]
*********************************************************************************************************************
*********************************************************************************
changed: [10.0.2.45]

PLAY RECAP
*********************************************************************************************************************
*********************************************************************************
10.0.2.45                  : ok=9    changed=9    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

3.8 - Vérification des Modifications

Vérifiez la modification dans la VM **CentOS_8** :

```
trainee@debian11:~/ansible/storage$ ansible all -m command -a 'lsblk'
10.0.2.45 | CHANGED | rc=0 >>
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                   8:0    0   32G  0 disk
├─sda1                8:1    0    1G  0 part /boot
└─sda2                8:2    0   31G  0 part
  ├─cl_centos8-root 253:0    0 27.8G  0 lvm  /
  └─cl_centos8-swap 253:1    0  3.2G  0 lvm  [SWAP]
sdb                   8:16   0    4G  0 disk
├─sdb1                8:17   0    2G  0 part /data/sales
└─sdb2                8:18   0    2G  0 part
  └─vg1-lv1         253:2    0    2G  0 lvm  /data/marketing
sdc                   8:32   0   64G  0 disk
└─sdc1                8:33   0   64G  0 part /home
sdd                   8:48   0   32G  0 disk
sr0                  11:0    1 1024M  0 rom

trainee@debian11:~/ansible/storage$ ansible all -m command -a 'cat /etc/fstab'
10.0.2.45 | CHANGED | rc=0 >>

#
# /etc/fstab
# Created by anaconda on Wed Jun 16 06:21:32 2021
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
```

```
#
/dev/mapper/cl_centos8-root /                       xfs     defaults        0 0
UUID=1c04981e-5317-4b73-9695-3ce25246835d /boot                   ext4    defaults        1 2
/dev/mapper/cl_centos8-swap swap                    swap    defaults        0 0
UUID=f76d6b66-985b-4a91-af9c-4987e8c1443c /home     ext4          defaults            1 2
/dev/sdb1 /data/sales xfs defaults 0 0
/dev/vg1/lv1 /data/marketing xfs defaults 0 0
```

Important : Notez la présence des filesystems **/dev/sdb1** et **/dev/vg1/lv1** ainsi que les points de montage **/data/sales** et **/data/marketing**. Notez aussi les modifications apportées au fichier **/etc/fstab**.

LAB #4 - Gestion du Tâches

4.1 - Création d'un Cron Job

Créez le fichier **schedule.yml** :

```
trainee@debian11:~/ansible/storage$ vi schedule.yml

trainee@debian11:~/ansible/storage$ cat schedule.yml
---
- name: scheduling
  hosts: all
  become: true
  gather_facts: false
  tasks:
    - name: cron_job
      cron:
```

```
        name: my cron job
        hour: '11'
        minute: '25'
        job: 'cat /etc/passwd > /tmp/file1'
        user: root
        cron_file: mycron
```

Important : Notez l'utilisation du module **cron** qui crée un cron job dont la description est **my cron job** et le fichier est **/etc/cron.d/mycron**. Notez que les valeurs de **hour** et **minute** doivent être passées en tant que texte.

Exécutez le playbook :

```
trainee@debian11:~/ansible/storage$ ansible-playbook schedule.yml

PLAY [scheduling]
**********************************************************************************************************
**************************************************************************

TASK [cron_job]
**********************************************************************************************************
**************************************************************************
changed: [10.0.2.45]

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

Vérifiez le contenu du fichier **/etc/cron.d/mycron** :

```
trainee@debian11:~/ansible/storage$ ansible all -m command -a 'cat /etc/cron.d/mycron'
10.0.2.45 | CHANGED | rc=0 >>
#Ansible: my cron job
25 11 * * * root cat /etc/passwd > /tmp/file1
```

Important : Notez que la valeur de **name:** figure sur la ligne commençant par **#Ansible:**.

4.1 - Création d'un AT Job

Créez le fichier **abc.txt** dans le répertoire **/home/trainee** de la VM **CentOS_8** :

```
trainee@debian11:~/ansible/storage$ ansible all -m command -a 'touch /home/trainee/abc.txt'
[WARNING]: Consider using the file module with state=touch rather than running 'touch'.  If you need to use
command because file is insufficient you can add 'warn: false' to this command task or set
'command_warnings=False' in ansible.cfg to get rid of this message.
10.0.2.45 | CHANGED | rc=0 >>
```

Éditez le fichier **schedule.yml** :

```
trainee@debian11:~/ansible/storage$ vi schedule.yml

trainee@debian11:~/ansible/storage$ cat schedule.yml
---
- name: scheduling
  hosts: all
  become: true
  gather_facts: false
  tasks:
```

```
    - name: cron_job
      cron:
        name: my cron job
        hour: '11'
        minute: '25'
        job: 'cat /etc/passwd > /tmp/file1'
        user: root
        cron_file: mycron

    - name: at_job
      at:
        command: cp /home/trainee/abc.txt /tmp/
        count: 1
        units: minutes
        unique: true
```

Important : Notez l'utilisation du module **at** qui crée un at job qui s'exécutera une minute après l'exécution du playbook. Puisque le module **at** ne comprend pas l'élément **name:**, la valeur **unique:true** spécifie que si le job existe déjà dans la queue, celui-ci ne sera pas créé. Notez que la valeur de **count:** est un entier.

Exécutez le playbook :

```
trainee@debian11:~/ansible/storage$ ansible-playbook schedule.yml

PLAY [scheduling]
*********************************************************************************************************
*****************************************************************

TASK [cron_job]
```

```
**********************************************************************************************************
**********************************************************************
ok: [10.0.2.45]

TASK [at_job]
**********************************************************************************************************
**********************************************************************
changed: [10.0.2.45]

PLAY RECAP
**********************************************************************************************************
**********************************************************************
10.0.2.45                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Vérifiez immédiatement la présence du fichier at job :

```
trainee@debian11:~/ansible/storage$ ansible all -m command -a 'ls /var/spool/at'
10.0.2.45 | CHANGED | rc=0 >>
a0000201af2968
spool
```

Copyright © 2024 Hugh Norris.