

Objectif du Bloc 6

Ce bloc consiste à installer et configurer deux serveurs Debian :

- **MyWebApp** : doit contenir un utilisateur **devops** et une crontab qui sauvegarde les fichiers de configuration du serveur Web tous les jours à 2h du matin.
- **MyDatabase** : doit contenir un utilisateur **dbadmin**.

Les deux serveurs doivent pouvoir communiquer entre eux et accéder à Internet.

Etape 1 : Création des machines virtuelles

Deux machines virtuelles ont été créées sous VMware avec les configurations suivantes :

MyWebApp

- ✗ 1 CPU, 2 Go RAM
- ✗ 40 Go de stockage
- ✗ Interface réseau : NAT
- ✗ Nom d'hôte : MyWebApp

MyDatabase

- ✗ 2 CPU, 4 Go RAM
- ✗ 40 Go de stockage
- ✗ Interface réseau : NAT
- ✗ Nom d'hôte : MyDatabase

Test de connexion entre les deux machines

Ping vers MyDatabase

```
seveur@MyWebApp: ~  
seveur@MyWebApp:~$ ping 192.168.254.153  
PING 192.168.254.153 (192.168.254.153) 56(84) bytes of data.  
64 bytes from 192.168.254.153: icmp_seq=1 ttl=64 time=0.502 ms  
64 bytes from 192.168.254.153: icmp_seq=2 ttl=64 time=1.39 ms  
64 bytes from 192.168.254.153: icmp_seq=3 ttl=64 time=1.56 ms  
64 bytes from 192.168.254.153: icmp_seq=4 ttl=64 time=2.37 ms  
64 bytes from 192.168.254.153: icmp_seq=5 ttl=64 time=1.45 ms  
^Z  
[4]+  Stoppé                  ping 192.168.254.153
```

Ping vers MyWebApp

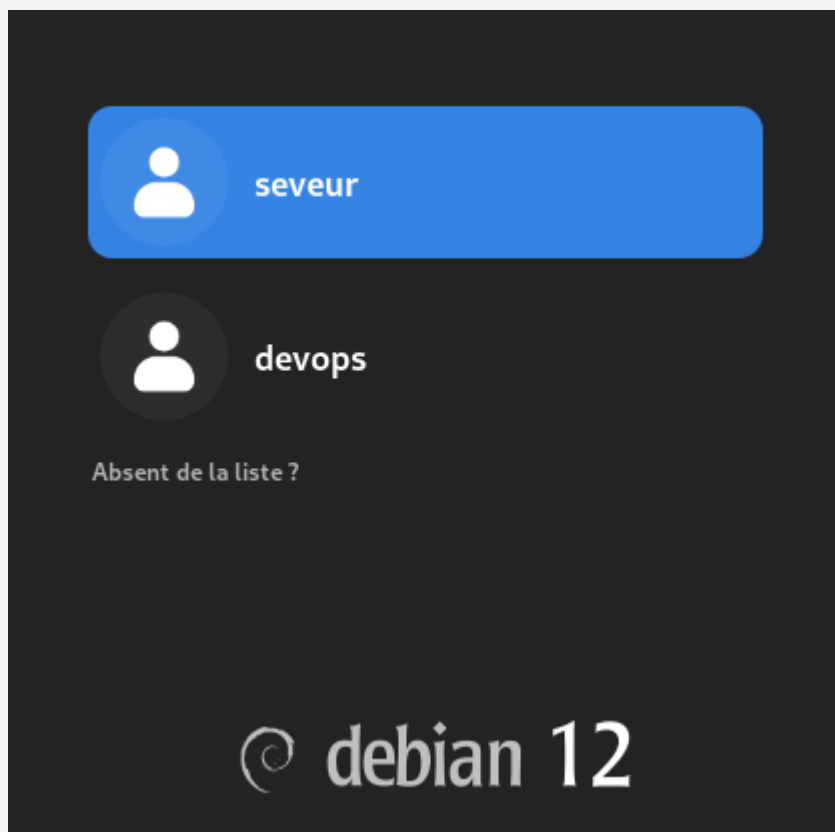
```
serveur@MyDatabase: ~  
serveur@MyDatabase:~$ ping 192.168.254.152  
PING 192.168.254.152 (192.168.254.152) 56(84) bytes of data.  
64 bytes from 192.168.254.152: icmp_seq=1 ttl=64 time=0.396 ms  
64 bytes from 192.168.254.152: icmp_seq=2 ttl=64 time=1.28 ms  
64 bytes from 192.168.254.152: icmp_seq=3 ttl=64 time=1.53 ms  
64 bytes from 192.168.254.152: icmp_seq=4 ttl=64 time=1.52 ms  
64 bytes from 192.168.254.152: icmp_seq=5 ttl=64 time=1.47 ms  
^Z  
[1]+  Stoppé                  ping 192.168.254.152
```

Etape 2 : Création des utilisateurs

Sur MyWebApp

✗ Connexion entant que root

```
seveur@MyWebApp: ~$ su
Mot de passe :
root@MyWebApp:/home/seveur# sudo adduser devops
Ajout de l'utilisateur « devops » ...
Ajout du nouveau groupe « devops » (1001) ...
Ajout du nouvel utilisateur « devops » (1001) avec le groupe « devops » (1001) .
..
Création du répertoire personnel « /home/devops » ...
Copie des fichiers depuis « /etc/skel » ...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
Modifier les informations associées à un utilisateur pour devops
Entrer la nouvelle valeur, ou appuyer sur ENTER pour la valeur par défaut
  NOM []:
  Numéro de chambre []:
  Téléphone professionnel []:
  Téléphone personnel []:
  Autre []:
Cette information est-elle correcte ? [0/n]o
Ajout du nouvel utilisateur « devops » aux groupes supplémentaires « users » ...
Ajout de l'utilisateur « devops » au groupe « users » ...
root@MyWebApp:/home/seveur#
```



Sur MyDatabase l'utilisateur dbadmin a été créer

```
serveur@MyDatabase: ~  
Mot de passe :  
root@MyDatabase:/home/serveur# sudo adduser dbadmin  
Ajout de l'utilisateur « dbadmin » ...  
Ajout du nouveau groupe « dbadmin » (1001) ...  
Ajout du nouvel utilisateur « dbadmin » (1001) avec le groupe « dbadmin » (1001)  
...  
Création du répertoire personnel « /home/dbadmin » ...  
Copie des fichiers depuis « /etc/skel » ...  
Nouveau mot de passe :  
Retapez le nouveau mot de passe :  
passwd : mot de passe mis à jour avec succès  
Modifier les informations associées à un utilisateur pour dbadmin  
Entrer la nouvelle valeur, ou appuyer sur ENTER pour la valeur par défaut  
  NOM []:  
  Numéro de chambre []:  
  Téléphone professionnel []:  
  Téléphone personnel []:  
  Autre []:  
Cette information est-elle correcte ? [0/n]  
Ajout du nouvel utilisateur « dbadmin » aux groupes supplémentaires « users » ..  
.  
Ajout de l'utilisateur « dbadmin » au groupe « users » ...  
root@MyDatabase:/home/serveur#
```



Etape 3 : Mise en place de la sauvegarde automatique (crontab)

Objectif

Mettre en place un script qui s'exécute tous les jours à 2h du matin pour sauvegarder les fichiers de configuration du serveur Web dans /var/backup/.

- ✗ Création du dossier de destination
- ✗ **Création du script** : Chemin : `/usr/local/bin/sauvegarde.sh`

```
root@MyWebApp:/home/seveur# mkdir -p var/backup
root@MyWebApp:/home/seveur# nano /usr/local/bin/sauvegarde.sh
root@MyWebApp:/home/seveur#
```



- ✗ Rendre le script exécutable : `chmod +x /usr/local/bin/sauvegarde.sh`
- ✗ Test manuel du script :
`/usr/local/bin/sauvegarde.sh`
Résultat : fichier
`backup_YYYY-MM-DD_HH-MM.tar.gz` créé dans `/var/backup`

- ✗ Ajout à la crontab root :

```
root@MyWebApp:/home/seveur# crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.tiny

Choose 1-2 [1]: 1
crontab: installing new crontab
root@MyWebApp:/home/seveur#
```



```
seveur@MyWebApp: ~
GNU nano 7.2 /tmp/crontab.5tmAEP/crontab *
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 5 * * * /usr/local/bin/sauvegarde.sh
```

```
root@MyWebApp:/home/seveur# ls /var/backup
backup_2025-07-01_16-51.tar.gz
```

Fin du rapport

