

TP Final PROXMOX

Proxmox TP

I- Installation des VM

| | |
|---------------------------------------------------------|---|
| 1.1 Installation et configuration des VM..... | 3 |
| 1.2 Redirection des ports | 4 |
| 1.3 Configuration de Proxmox..... | 5 |
| 1.4 Création des différents conteneurs..... | 5 |
| 1.5 Adressage IP des différentes interfaces réseau..... | 8 |

II- Mise en place des serveurs web sous Apache

| | |
|--------------------------------------------------------|----|
| 2.1 Installation d'Apache sur les serveurs Alpine..... | 9 |
| 2.2 Installation d'apache sur le serveur Debian..... | 11 |

III- Mise en place du serveur de synchronisation

| | |
|-----------------------------------------------------|----|
| 3.1 Préliminaires avant la synchronisation | 13 |
| 3.2 Envoie du fichier index par protocole SCP..... | 16 |
| 3.3 Synchronisation du fichier index avec Cron..... | 19 |

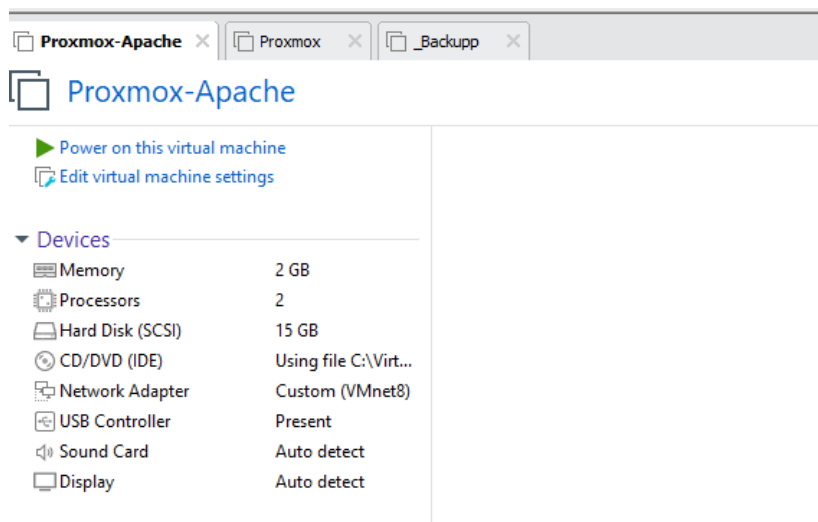
III- Loadbalancing et Backup Server

| | |
|-------------------------------------------|----|
| 3.1 Loadbalancing par serveur Nginx | 22 |
| 3.2 Mise en place du Backup Server..... | 24 |

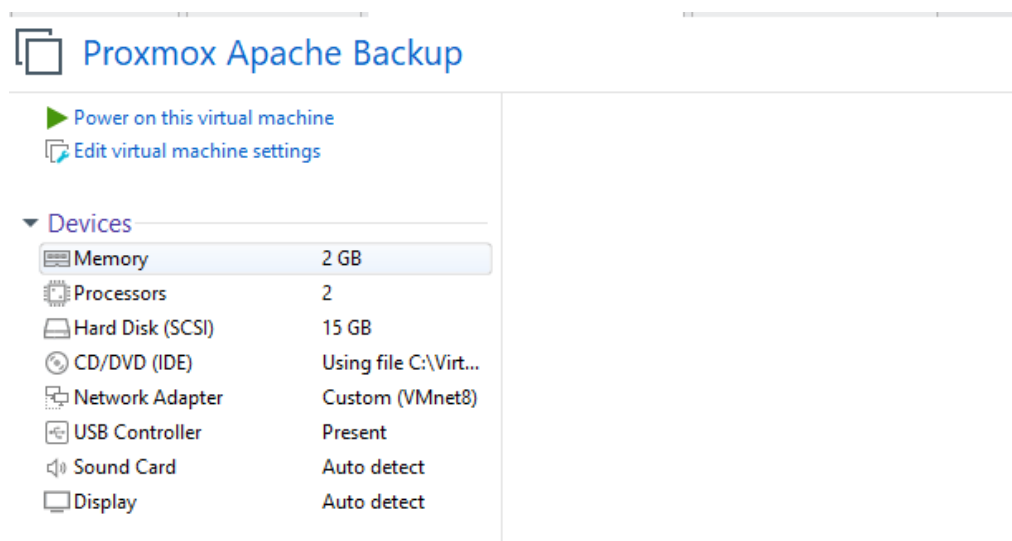
Installation des VM

Installation et configuration des VM

Voici la configuration de la VM qui supportera Proxmox :



Et enfin la configuration de notre Back-up qui se trouve dans le même réseau que notre Proxmox (VMnet8)



Redirection des ports

Voici l'adressage IP de nos VM :

Proxmox 192.168.127.138 :8006

```
Welcome to the Proxmox Virtual Environment. Please use your web browser to
configure this server - connect to:
```

```
https://192.168.127.138:8006/
```

Proxmox-Back-UP 192.168.127.140 :8007

```
Welcome to the Proxmox Backup Server. Please use your web browser to
configure this server - connect to:
```

```
https://192.168.127.140:8007/
```

On va maintenant rediriger nos ports :

NAT Settings

Network: vmnet8

Subnet IP: 192.168.127.0

Subnet mask: 255.255.255.0

Gateway IP: 192.168.127.2

Port Forwarding

| Host Port | Type | Virtual Machine IP Address | Description |
|-----------|------|----------------------------|-----------------|
| 8006 | TCP | 192.168.127.138:8006 | Proxmox |
| 8007 | TCP | 192.168.127.140:8007 | Proxmox back-up |

Add...

Remove

Properties

Advanced

☒ Allow active FTP

☒ Allow any Organizationally Unique Identifier

UDP timeout (in seconds): 30

Config port: 0

☐ Enable IPv6

IPv6 prefix: fd15:4ba5:5a2b:1008::/64

DNS Settings...

NetBIOS Settings...

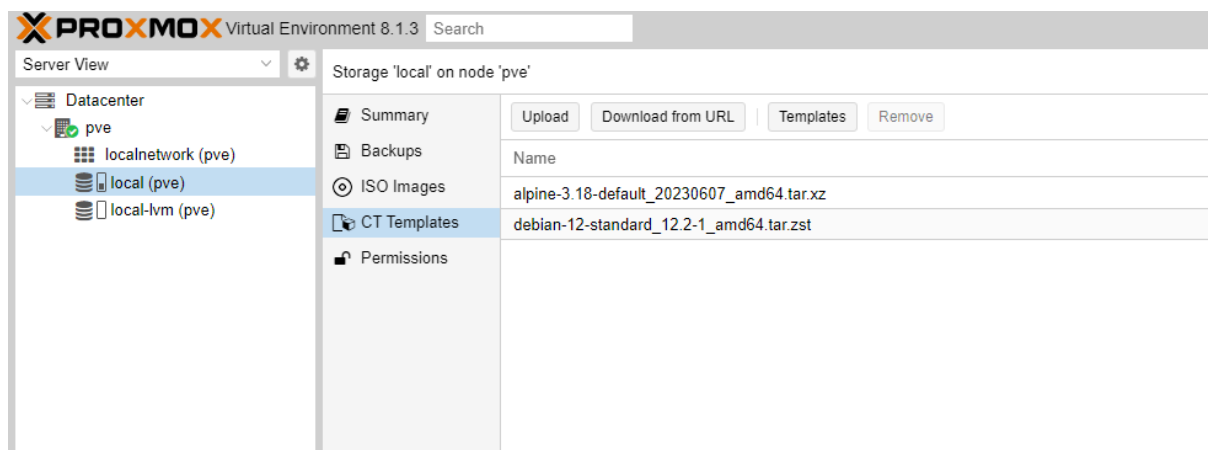
OK

Cancel

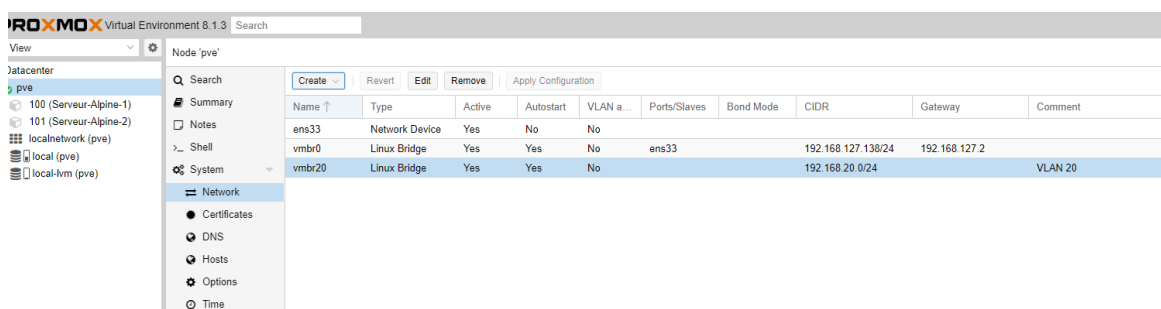
Help

Configuration de Proxmox

On ajoute les différentes templates que l'on va utiliser, en l'occurrence celle d'Alpine et de Debian.

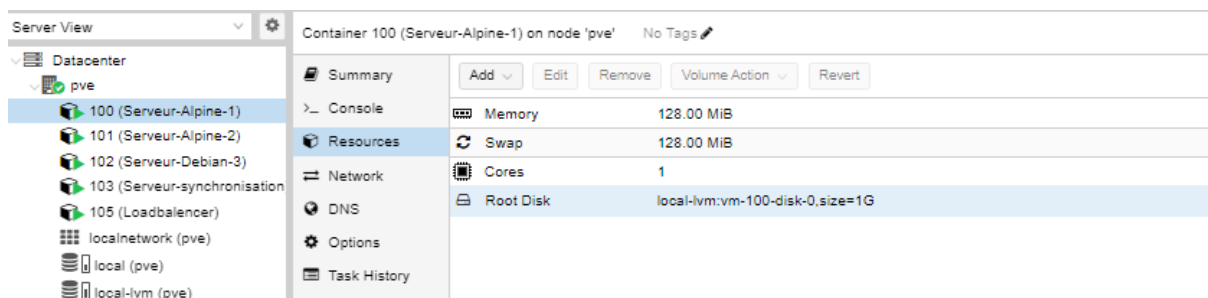










On rajoute une interface réseau qui sera notre Vlan 20.




Création des différents conteneurs

Pour **les serveurs Alpine**, on a une configuration très légère avec assez peu de mémoire ce qui reste quand même largement suffisant pour cette distribution qui est très peu gourmande au vu de sa simplicité.






| Serveur-Alpine-1 (Uptime: 04:27:05) | | Alpine |
|-------------------------------------------------------------------------------------------------|----------------------------------|--------|
|  Status | running | |
|  HA State | none | |
|  Node | pve | |
|  Unprivileged | Yes | |
|  CPU usage | 0.09% of 1 CPU(s) | |
|  Memory usage | 4.39% (5.61 MiB of 128.00 MiB) | |
|  SWAP usage | 0.31% (412.00 KiB of 128.00 MiB) | |
|  Bootdisk size | 2.35% (22.88 MiB of 973.42 MiB) | |

PROXMOX Virtual Environment 8.1.3









Server View 

- Datacenter
 - pve
 - 100 (Serveur-Alpine-1)
 - 101 (Serveur-Alpine-2)**
 - 102 (Serveur-Debian-3)
 - 103 (Serveur-synchronisation)
 - 105 (Loadbalancer)
 - localnetwork (pve)
 - local (pve)
 - local-lvm (pve)

Container 101 (Serveur-Alpine-2) on node 'pve' No Tags 

| | | | | | |
|--------------|---------------------------------------------------------------------------------------|------|--------|---------------------------------------------------------------------------------------------------|--------|
| Summary | Add  | Edit | Remove | Volume Action  | Revert |
| Console | | | | | |
| Resources | | | | | |
| Network | | | | | |
| DNS | | | | | |
| Options | | | | | |
| Task History | | | | | |
| Backup | | | | | |

| | |
|-----------|---------------------------------|
| Memory | 128.00 MiB |
| Swap | 128.00 MiB |
| Cores | 1 |
| Root Disk | local-lvm:vm-101-disk-0,size=1G |

| Serveur-Alpine-2 (Uptime: 04:26:30) | | Alpine |
|---------------------------------------------------------------------------------------------------|---------------------------------|--------|
|  Status | running | |
|  HA State | none | |
|  Node | pve | |
|  Unprivileged | Yes | |
|  CPU usage | 0.00% of 1 CPU(s) | |
|  Memory usage | 3.95% (5.05 MiB of 128.00 MiB) | |
|  SWAP usage | 0.96% (1.23 MiB of 128.00 MiB) | |
|  Bootdisk size | 2.35% (22.88 MiB of 973.42 MiB) | |

Pour les conteneurs tournant sous Debian on attribue une configuration un peu plus élevée car celle-ci demande plus de ressource.

Server View

Container 102 (Serveur-Debian-3) on node 'pve' No Tags

Summary Add Edit Remove Volume Action Revert

Console

Resources

Network

DNS

Options

Task History

| | |
|-----------|---------------------------------|
| Memory | 512.00 MiB |
| Swap | 512.00 MiB |
| Cores | 1 |
| Root Disk | local-lvm:vm-102-disk-0,size=4G |

Serveur-Debian-3 (Uptime: 04:30:55) Debian

Status running

HA State none

Node pve

Unprivileged Yes

CPU usage 0.00% of 1 CPU(s)

Memory usage 4.93% (25.22 MiB of 512.00 MiB)

SWAP usage 1.40% (7.18 MiB of 512.00 MiB)

Bootdisk size 16.85% (665.81 MiB of 3.86 GiB)

Server View

Container 103 (Serveur-synchronisation) on node 'pve' No Tags

Summary Add Edit Remove Volume Action Revert

Console

Resources

Network

DNS

Options

Task History

| | |
|-----------|---------------------------------|
| Memory | 512.00 MiB |
| Swap | 512.00 MiB |
| Cores | 1 |
| Root Disk | local-lvm:vm-103-disk-0,size=3G |

Serveur-synchronisation (Uptime: 00:30:43) Debian

Status running

HA State none

Node pve

Unprivileged Yes

CPU usage 0.00% of 1 CPU(s)

Memory usage 5.40% (27.64 MiB of 512.00 MiB)

SWAP usage 0.00% (0 B of 512.00 MiB)

Bootdisk size 20.61% (608.27 MiB of 2.88 GiB)

Container 105 (Loadbalancer) on node 'pve' No Tags

Summary
Console
Resources
Network
DNS
Options
Task History
Backup

Add Edit Remove Volume Action Revert

| | |
|-----------|---------------------------------|
| Memory | 512.00 MiB |
| Swap | 512.00 MiB |
| Cores | 1 |
| Root Disk | local-lvm:vm-105-disk-0,size=1G |

Loadbalancer @ Debian

| | |
|---------------|---------------------------|
| Status | stopped |
| HA State | none |
| Node | pve |
| Unprivileged | Yes |
| CPU usage | 0.00% of 1 CPU(s) |
| Memory usage | 0.00% (0 B of 512.00 MiB) |
| SWAP usage | 0.00% (0 B of 512.00 MiB) |
| Bootdisk size | 1.00 GiB |

Adressage IP des différentes interfaces réseau

Voici l'adressage IP de mes conteneurs :

Serveur-Alpine-1 : eth0 → 192.168.127.150/24

vlan-20 → 192.168.20.1/24

Container 100 (Serveur-Alpine-1) on node 'pve' No Tags

Summary
Console
Resources
Network
DNS

Add Remove Edit

| ID ↑ | Name | Bridge | Firewall | VLAN Tag | MAC address | IP address | Gateway | MTU | Disconnected |
|------|---------|--------|----------|----------|----------------|--------------------|---------------|-----|--------------|
| net0 | eth0 | vmbr0 | Yes | | BC:24:11:D7... | 192.168.127.150/24 | 192.168.127.2 | | No |
| net1 | vlan-20 | vmbr20 | Yes | | BC:24:11:D... | 192.168.20.1/24 | | | No |

Serveur-Alpine-2 : eth0 → 192.168.127.151/24

 vlan-20→ 192.168.20.2/24

| | | | | | | | | | | | |
|-----------------------------------------------------------|----------------------------|---------|--------|----------|----------|----------------|--------------------|---------------|-----|--------------|--|
| Container 101 (Serveur-Alpine-2) on node 'pve' No Tags | | | | | | | | | | | |
| Summary | <div>Add Remove Edit</div> | | | | | | | | | | |
| Console | ID ↑ | Name | Bridge | Firewall | VLAN Tag | MAC address | IP address | Gateway | MTU | Disconnected | |
| Resources | net0 | eth0 | vmbr0 | Yes | | BC:24:11:C5... | 192.168.127.151/24 | 192.168.127.2 | | No | |
| Network | net1 | Vlan-20 | vmbr20 | Yes | | BC:24:11:83... | 192.168.20.2/24 | | | No | |
| DNS | | | | | | | | | | | |
| Options | | | | | | | | | | | |

Serveur-Debian-3 : eth0 → 192.168.127.152/24

 vlan-20→ 192.168.20.3/24

| | | | | | | | | | | | |
|-----------------------------------------------------------|----------------------------|---------|--------|----------|----------|----------------|--------------------|---------------|-----|--------------|--|
| Container 102 (Serveur-Debian-3) on node 'pve' No Tags | | | | | | | | | | | |
| Summary | <div>Add Remove Edit</div> | | | | | | | | | | |
| Console | ID ↑ | Name | Bridge | Firewall | VLAN Tag | MAC address | IP address | Gateway | MTU | Disconnected | |
| Resources | net0 | eth0 | vmbr0 | Yes | | BC:24:11:0B... | 192.168.127.152/24 | 192.168.127.2 | | No | |
| Network | net1 | Vlan-20 | vmbr20 | Yes | | BC:24:11:6C... | 192.168.20.3/24 | | | No | |
| DNS | | | | | | | | | | | |

Serveur-Synchronisation : vlan-20→ 192.168.20.4/24

| | | | | | | | | | | | |
|------------------------------------------------------------------|----------------------------|---------|--------|----------|----------|----------------|-----------------|---------|-----|--------------|--|
| Container 103 (Serveur-synchronisation) on node 'pve' No Tags | | | | | | | | | | | |
| Summary | <div>Add Remove Edit</div> | | | | | | | | | | |
| Console | ID ↑ | Name | Bridge | Firewall | VLAN Tag | MAC address | IP address | Gateway | MTU | Disconnected | |
| Resources | net0 | Vlan-20 | vmbr20 | Yes | | BC:24:11:48... | 192.168.20.4/24 | | | No | |
| Network | | | | | | | | | | | |
| DNS | | | | | | | | | | | |
| Options | | | | | | | | | | | |
| Task History | | | | | | | | | | | |

Loadbalancer : eth0 → 192.168.127.153/24

| | | | | | | | | | | | |
|-------------------------------------------------------|----------------------------|------|--------|----------|----------|----------------|--------------------|---------------|-----|--------------|--|
| Container 104 (Loadbalancer) on node 'pve' No Tags | | | | | | | | | | | |
| Summary | <div>Add Remove Edit</div> | | | | | | | | | | |
| Console | ID ↑ | Name | Bridge | Firewall | VLAN Tag | MAC address | IP address | Gateway | MTU | Disconnected | |
| Resources | net0 | eth0 | vmbr0 | Yes | | BC:24:11:B6... | 192.168.127.153/24 | 192.168.127.2 | | No | |
| Network | | | | | | | | | | | |
| DNS | | | | | | | | | | | |

Mise en place des serveurs web sous Apache

Installation d'apache sur les serveurs Alpines

Dans un but de praticité, je montrerai uniquement les étapes sur un des deux conteneurs Alpine (vu que c'est strictement la même chose).

Avant d'installer Apache il est nécessaire de mettre à jour notre conteneur.

Par la commande suivante :

`apk update`

```
Serveur-Alpine-1:~# apk update
```

`apk upgrade`

```
Serveur-Alpine-1:~# apk upgrade
```

On peut maintenant installer apache, par la commande suivante :

`apk add apache2`

```
Serveur-Alpine-1:~# apk add apache2
```

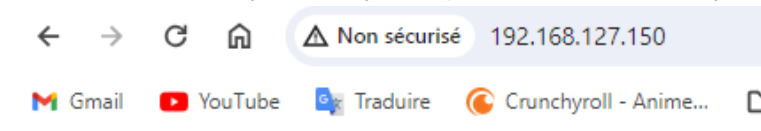
Après avoir installer apache, on lance le service apache2 et on vérifie que ce dernier est bien actif, par la commande suivante :

`rc-service apache2 start`

`service apache2 status`

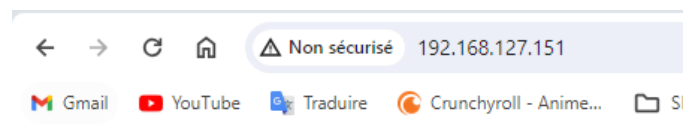
```
Serveur-Alpine-1:~# rc-service apache2 start
* Caching service dependencies ... [ ok ]
* Starting apache2 ... [ ok ]
Serveur-Alpine-1:~# service apache2 status
* status: started
Serveur-Alpine-1:~#
```

On constate que le service est bien actif. Par ailleurs, on peut vérifier l'accessibilité de notre serveur web à travers notre navigateur en tapant l'adresse ip de nos serveurs, c'est-à-dire `http://192.168.127.150` (serveur1) et `http://192.168.127.151` (serveur2)



It works!

Serveur-Alpine-1



It works!

Serveur-Alpine-2

On va maintenant modifier l'index.html de notre site pour pouvoir le différencier des autres serveurs. On se rend sur le dossier suivant : `/var/www/localhost/htdocs/index.html` puis on modifie le fichier index existant.

On modifie notre fichier par la commande suivante (installer au préalable le package « nano »):

`nano /var/www/localhost/htdocs/index.html`

```
nano /var/www/localhost/htdocs/index.html
```

```
GNU nano 7.2 /var/www/localhost/htdocs/index.html
<html><body><h1>Serveur-Alpine-1</h1></body></html>
```

On se rend sur notre navigateur afin de vérifier que les modifications ont bien été prise en compte. On constate que tout marche correctement.



Installation d'apache sur le serveur Debian

Avant d'installer Apache il est nécessaire de mettre à jour notre conteneur.

Par la commande suivante :

`apt update && apt -y upgrade`

```
root@Serveur-Debian-3:~# apt update && apt -y upgrade
```

On peut maintenant installer apache par la commande suivante :

apt install apache2

```
root@Serveur-Debian-3:~# apt install apache2
```

Après avoir installer apache, on vérifie que le service est bien actif, par la commande suivante :

systemctl status apache2

La commande systemctl permet de gérer les services et contrôler l'état du système du systemd.

```
root@Serveur-Debian-3:~# systemctl status apache2
* apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-01-16 19:50:37 UTC; 1min 37s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 743 (apache2)
    Tasks: 55 (limit: 2228)
   Memory: 5.5M
      CPU: 22ms
   CGroup: /system.slice/apache2.service
           |-743 /usr/sbin/apache2 -k start
           |-745 /usr/sbin/apache2 -k start
           `--746 /usr/sbin/apache2 -k start

Jan 16 19:50:37 Serveur-Debian-3 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jan 16 19:50:37 Serveur-Debian-3 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@Serveur-Debian-3:~#
```

On constate que le service est bien actif. Par ailleurs, on peut vérifier l'accessibilité de notre serveur web à travers notre navigateur en tapant l'adresse ip de notre serveur c'est-à-dire **http://192.168.127.152**



On va maintenant modifier l'index.html de notre site pour pouvoir le différencier des autres serveurs. On se rend sur le dossier suivant : **/var/www/html/** puis on supprime le fichier index existant pour créer le notre. Cela nous permettra de différencier nos serveurs entre eux.

On supprime le fichier index.html par la commande suivante :

rm /var/www/html/index.html

```
root@Serveur-Debian-3:/etc/apache2/sites-enabled# rm /var/www/html/index.html
```

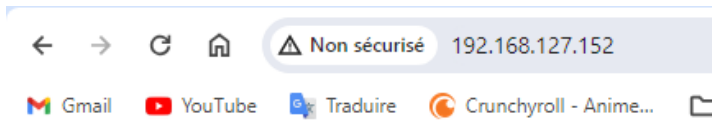
Puis on crée et on modifie par la commande suivante notre nouveau fichier index.html :

nano /var/www/html/index .html

```
root@Serveur-Debian-3:/var/www/html# nano /var/www/html/index.html
```

```
GNU nano 7.2 /var/www/html/index.html
<h1> Serveur-debian-3 </h1>
```

On se rend sur notre navigateur afin de vérifier que les modifications ont bien été prise en compte. On constate que tout marche correctement.



Serveur-debian-3

Mise en place du serveur de synchronisation

Préliminaires avant la synchronisation

La première étape va être d'installer SSH sur nos serveurs web et sur notre serveur de synchronisation. Le protocole SSH va nous permettre d'établir une communication sécurisée, grâce notamment à l'échange de clé de chiffrement, entre notre serveur de synchronisation et nos serveurs web.

Cela nous permettra ensuite de copier notre fichier index.html (que l'on créera au préalable) présent sur notre serveur de synchronisation vers nos serveurs web.

Commençons donc par l'installation d'SSH sur nos serveurs et la vérification qu'ils soient bien actif.

Par les commandes suivantes :

```
apt install openssh-server
systemctl status ssh
```

```
root@Serveur-synchronisation:~# apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
openssh-server is already the newest version (1:9.2p1-2+deb12u1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@Serveur-synchronisation:~#
```

```
root@Serveur-synchronisation:~# systemctl status ssh
* ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-01-17 11:13:57 UTC; 22min ago
 TriggeredBy: * ssh.socket
    Docs: man:sshd(8).
          man:sshd_config(5).
   Process: 123 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 128 (sshd)
    Tasks: 1 (limit: 2228)
   Memory: 4.8M
      CPU: 20ms
   CGroup: /system.slice/ssh.service
           └─128 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 17 11:13:57 Serveur-synchronisation systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Jan 17 11:13:57 Serveur-synchronisation sshd[128]: Server listening on :: port 22.
Jan 17 11:13:57 Serveur-synchronisation systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
root@Serveur-synchronisation:~#
```

Serveur-synchronisation

```
root@Serveur-Debian-3:~# apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:9.2p1-2+deb12u2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@Serveur-Debian-3:~#
```

```
root@Serveur-Debian-3:~# systemctl status ssh
* ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-01-17 09:04:24 UTC; 2h 26min ago
 TriggeredBy: * ssh.socket
    Docs: man:sshd(8).
          man:sshd_config(5).
   Process: 152 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 173 (sshd)
    Tasks: 1 (limit: 2228)
   Memory: 4.8M
      CPU: 16ms
   CGroup: /system.slice/ssh.service
           └─173 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 17 09:04:24 Serveur-Debian-3 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Jan 17 09:04:24 Serveur-Debian-3 sshd[173]: Server listening on :: port 22.
Jan 17 09:04:24 Serveur-Debian-3 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
root@Serveur-Debian-3:~#
```

Serveur-Debian-3

Sur Alpine on réalise les commandes suivantes :

```
apk search openssh
apk add openssh
rc-update add sshd
service sshd start
service sshd status
```

```
Serveur-Alpine-2:~# apk search openssh
acf-openssh-0.11.2-r2
aconf-mod-openssh-0.8.1-r2
lxqt-openssh-askpass-1.3.0-r0
lxqt-openssh-askpass-doc-1.3.0-r0
lxqt-openssh-askpass-lang-1.3.0-r0
openssh-9.3_p2-r1
openssh-askpass-9.0_p1-r0
openssh-client-common-9.3_p2-r1
openssh-client-default-9.3_p2-r1
openssh-client-krb5-9.3_p2-r1
openssh-client-yash-completion-2.54-r3
openssh-dbg-9.3_p2-r1
openssh-doc-9.3_p2-r1
openssh-keygen-9.3_p2-r1
openssh-keysign-9.3_p2-r1
openssh-server-9.3_p2-r1
openssh-server-common-9.3_p2-r1
openssh-server-krb5-9.3_p2-r1
openssh-server-pam-9.3_p2-r1
openssh-sftp-server-9.3_p2-r1
openssh-sk-helper-9.3_p2-r1
perl-net-openssh-0.83-r0
perl-net-openssh-doc-0.83-r0
Serveur-Alpine-2:~# apk add openssh
(1/8) Installing openssh-keygen (9.3_p2-r1)
(2/8) Installing libedit (20221030.3.1-r1)
(3/8) Installing openssh-client-common (9.3_p2-r1)
(4/8) Installing openssh-client-default (9.3_p2-r1)
(5/8) Installing openssh-sftp-server (9.3_p2-r1)
(6/8) Installing openssh-server-common (9.3_p2-r1)
(7/8) Installing openssh-server (9.3_p2-r1)
(8/8) Installing openssh (9.3_p2-r1)
Executing busybox-1.36.1-r5.trigger
OK: 21 MiB in 47 packages
Serveur-Alpine-2:~# rc-update add sshd
* service sshd added to runlevel default
Serveur-Alpine-2:~# service sshd start
* Caching service dependencies ... [ ok ]
ssh-keygen: generating new host keys: RSA ECDSA ED25519
* Starting sshd ... [ ok ]
Serveur-Alpine-2:~# service sshd status
* status: started
Serveur-Alpine-2:~#
```

(Serveur-Alpine-1 et Serveur-Alpine-2)

Maintenant que le SSH est installé, je peux passer à sa configuration.

Pour ce faire, nous allons nous rendre sur le fichier config de SSH, **des serveurs et du synchroniseur**.

`/etc/ssh/sshd_config`

Dans ce fichier nous allons activer les instructions pour le PermitRootLogin et la PubkeyAuthentication.

Les activer nous permettra de nous connecter, en root mais aussi à l'aide des clés publiques

```
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes
```

Je réalise exactement la même manipulation sur mes serveurs web

Je restart ensuite mon SSH pour que les modifications soient prises en compte, par la commande suivante :

`systemctl restart ssh` (pour Debian)

`service sshd restart` (pour Alpine)

Après cela, on peut générer nos clés privées/publiques SSH sur notre serveur de synchronisation (on verra dans la section suivante à quoi sert ces clés.) par la commande suivante :

`ssh-keygen`

```
root@Serveur-synchronisation:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:1EM9DjgPFQCuM/luT0pXrKi7bTnXNF3X3889wiFfz9k root@Serveur-synchronisation
The key's randomart image is:
+---[RSA 3072]-----+
|    . . . + + o    |
|    . + o . o    |
|    . + o o . . |
|    o . . . . . o |
|    = S o . . . o |
|    + . o o . . . o |
|    + . + o . + o o * |
|    == + . . + o E |
|    + = + + . . . |
+----[SHA256]-----+
root@Serveur-synchronisation:~#
```

Envoi du fichier index par protocole SCP

Maintenant, on va se rendre sur notre synchroniseur où je vais copier ma clé publique vers tous les serveurs web. Cela va me permettre de donner une « autorisation » à mon synchroniseur d'accéder à distance, à l'aide du SSH, à mes serveurs web, et cela, de manière instantanée (car on n'a pas demandé de vérification par password).

Pour ce faire, on va utiliser la commande suivante :

`ssh-copy-id root@192.168.20.1` (pour le serveur-Alpine-1)

`ssh-copy-id root@192.168.20.2` (pour le serveur-Alpine-2)

`ssh-copy-id root@192.168.20.3` (pour le serveur-Debian-3)

Et ensuite, on vérifiera si l'accès est bien établi entre les deux machines en se connectant par SSH, en utilisant la commande suivante :

`ssh root@192.168.20.1` (pour le serveur-Alpine-1)

`ssh root@192.168.20.2` (pour le serveur-Alpine-2)

`ssh root@192.168.20.3` (pour le serveur-Debian-3)

Ici on décide donc de se connecter en tant que root au serveur que l'on vise. Cela nous évite de créer un utilisateur sur nos serveurs d'où l'utilité d'avoir activé la ligne de code `PermitRootLogin`


```

root@Serveur-synchronisation:~# ssh-copy-id root@192.168.20.1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '192.168.20.1 (192.168.20.1)' can't be established.
ED25519 key fingerprint is SHA256:UNKPxIEHHbChWayMl2Swxm9/7jWa+lepEwtYhEJDH6w.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@192.168.20.1's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@192.168.20.1'"
and check to make sure that only the key(s) you wanted were added.

root@Serveur-synchronisation:~# █

```

La connexion est bien possible.

```

root@Serveur-synchronisation:~# ssh root@192.168.20.1
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <https://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

Serveur-Alpine-1:~# █

```

Serveur-Alpine-1

```

root@Serveur-synchronisation:~# ssh-copy-id root@192.168.20.2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '192.168.20.2 (192.168.20.2)' can't be established.
ED25519 key fingerprint is SHA256:aOgU9Y9hJVilFBuoOyGxcRaPLfvvX+bzQM127MUAMtg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@192.168.20.2's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@192.168.20.2'"
and check to make sure that only the key(s) you wanted were added.

```

La connexion est bien possible

```

root@Serveur-synchronisation:~# ssh root@192.168.20.2
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <https://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

```

Serveur-Alpine-2

```

root@Serveur-synchronisation:~# ssh-copy-id root@192.168.20.3
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@192.168.20.3's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@192.168.20.3'"
and check to make sure that only the key(s) you wanted were added.

root@Serveur-synchronisation:~# █

```

La connexion est bien possible

```

root@Serveur-synchronisation:~# ssh root@192.168.20.3
Linux Serveur-Debian-3 6.5.11-4-pve #1 SMP PREEMPT_DYNAMIC PMX 6.5.11-4 (2023-11-20T10:19Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jan 17 16:26:24 2024
root@Serveur-Debian-3:~# █

```

Serveur-Debian-3

Maintenant, que la connexion par SSH est bien effective entre mes serveurs web et mon synchroniseur, je peux m'occuper de la première étape vers la synchronisation du fichier index.html.

On va commencer par créer le fichier index.html que nous souhaitons synchroniser avec nos serveurs web.

```

root@Serveur-synchronisation:~# nano /var/local/index.html █

```

```

GNU nano 7.2 /var/local/index.html *
<h1> Ceci correspond au fichier index.html synchronise </h1> █

```

Maintenant passons à la copie de ce fichier vers mes serveurs web à l'aide de la commande suivante :

scp <chemin source><utilisateur destinataire>@<IP du destinataire> :<fichier de destination>

scp /var/local/index.html root@192.168.20.1:/var/www/localhost/htdocs/index.html

scp /var/local/index.html root@192.168.20.2:/var/www/localhost/htdocs/index.html

scp /var/local/index.html root@192.168.20.3:/var/www/html/index.html

```

root@Serveur-synchronisation:~# scp /var/local/index.html root@192.168.20.1:/var/www/localhost/htdocs/index.html
index.html                                100% 62    59.7KB/s   00:00
root@Serveur-synchronisation:~# █

root@Serveur-synchronisation:~# scp /var/local/index.html root@192.168.20.2:/var/www/localhost/htdocs/index.html
index.html                                100% 62    124.2KB/s   00:00
root@Serveur-synchronisation:~# scp /var/local/index.html root@192.168.20.3:/var/www/html/index.html
index.html                                100% 62    74.9KB/s    00:00
root@Serveur-synchronisation:~# █

```

On peut constater que l'index.html a correctement été copié et rendu effectif vers mes serveurs.



Ceci correspond au fichier index.html synchronise

Serveur-Alpine-1



Ceci correspond au fichier index.html synchronise

Serveur-Alpine-2



Ceci correspond au fichier index.html synchronise

Serveur-Debian-3

Synchronisation du fichier index avec Cron

On va dans cette partie synchroniser notre fichier index.html toutes les 5 min. On va réaliser cela grâce à Cron qui sert de planificateur de tâche. Donc on va commencer par installer Cron sur notre serveur de synchronisation uniquement, avec la commande suivante :

apt install cron

```
root@Serveur-synchronisation:~# apt install cron
Reading package lists... Done
Building dependency tree... Done
cron is already the newest version (3.0pl1-162).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@Serveur-synchronisation:~#
```

Ensuite on va activer le service Cron et vérifier que ce dernier est bien actif, par les commandes suivantes :

systemctl enable cron

systemctl status cron

```
root@Serveur-synchronisation:~# systemctl status cron
* cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-01-20 09:13:43 UTC; 2s ago
     Docs: man:cron(8).
  Main PID: 391 (cron)
    Tasks: 1 (limit: 2228)
   Memory: 324.0K
      CPU: 1ms
   CGroup: /system.slice/cron.service
           `-391 /usr/sbin/cron -f
```

On va maintenant accéder à notre fichier Cron qui est un fichier texte contenant une liste de commandes que l'on souhaite effectuer à un moment précis. C'est dans ce fichier que l'on va écrire notre commande SCP réalisé ultérieurement, servant à copier le fichier index.html d'un point A à un point B., mais en plus de cela, on va préciser le moment où cela se fera.

On utilise la commande suivante , pour accéder à notre fichier Cron :

Crontab -e

Explication de ces tâches :

Chaque étoile correspond à une unité de temps. La première pour les minutes (0-59), la deuxième pour les heures (0-23), la troisième pour les jours dans le mois (1-31), la quatrième pour le mois (1- 12), et la cinquième pour les jours dans la semaine (0-6, 0 pour dimanche).

Dans notre cas ce qui nous intéresse, c'est d'automatiser notre tâche toutes les 5 minutes. Pour activer cette notion de redondance, on rajoute le « / » ainsi notre tâche se fera toutes les 5 minutes et non pas dans 5 minutes.

Suis ensuite la tâche que l'on souhaite réaliser, c'est-à-dire copier par protocole SCP notre fichier index.html vers le destinataire avec le chemin où l'on souhaite que la copie apparaisse.

On va donc créer nos deux tâches avec les lignes suivantes

```
*/5 * * * * scp /var/local/index.html root@192.168.20.1:/var/www/localhost/htdocs/index.html
*/5 * * * * scp /var/local/index.html root@192.168.20.2:/var/www/localhost/htdocs/index.html
*/5 * * * * scp /var/local/index.html root@192.168.20.3:/var/www/html/index.html
```

```
GNU nano 7.2 /tmp/crontab.q8iNet/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# 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

*/5 * * * * scp /var/local/index.html root@192.168.20.1:/var/www/localhost/htdocs/index.html
*/5 * * * * scp /var/local/index.html root@192.168.20.2:/var/www/localhost/htdocs/index.html
*/5 * * * * scp /var/local/index.html root@192.168.20.3:/var/www/html/index.html
```

On peut restart cron pour appliquer les modifications.

```
root@Serveur-synchronisation:~# systemctl restart cron
root@Serveur-synchronisation:~# systemctl status cron
* cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-01-20 09:20:42 UTC; 3s ago
     Docs: man:cron(8)
    Main PID: 447 (cron)
      Tasks: 1 (limit: 2228)
    Memory: 316.0K
       CPU: 1ms
    CGroup: /system.slice/cron.service
           └─447 /usr/sbin/cron -f

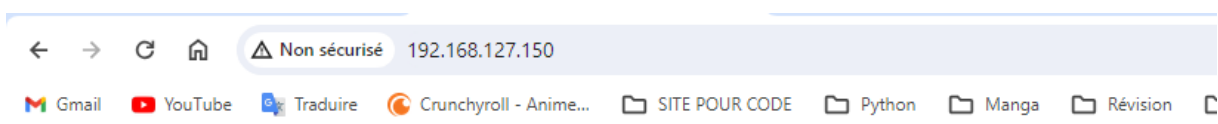
Jan 20 09:20:42 Serveur-synchronisation systemd[1]: Started cron.service - Regular background program processing daemon.
Jan 20 09:20:42 Serveur-synchronisation cron[447]: (CRON) INFO (pidfile fd = 3)
Jan 20 09:20:42 Serveur-synchronisation cron[447]: (CRON) INFO (Skipping @reboot jobs -- not system startup)
root@Serveur-synchronisation:~#
```

5 minutes plus tard...

```
Jan 20 09:20:42 Serveur-synchronisation systemd[1]: Started cron.service - Regular background program processing daemon.
Jan 20 09:20:42 Serveur-synchronisation cron[447]: (CRON) INFO (pidfile fd = 3)
Jan 20 09:20:42 Serveur-synchronisation cron[447]: (CRON) INFO (Skipping @reboot jobs -- not system startup)
root@Serveur-synchronisation:~# systemctl status cron
* cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-01-20 09:20:42 UTC; 5min ago
     Docs: man:cron(8)
    Main PID: 447 (cron)
      Tasks: 1 (limit: 2228)
    Memory: 316.0K
       CPU: 24ms
    CGroup: /system.slice/cron.service
           └─447 /usr/sbin/cron -f

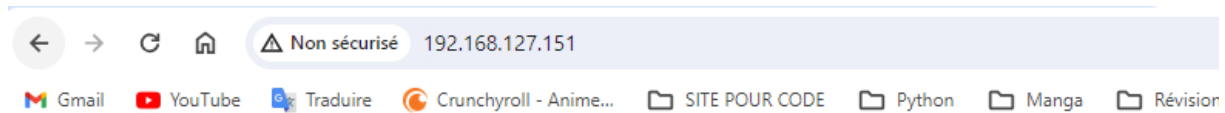
Jan 20 09:20:42 Serveur-synchronisation cron[447]: (CRON) INFO (Skipping @reboot jobs -- not system startup)
Jan 20 09:25:01 Serveur-synchronisation CRON[460]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Jan 20 09:25:01 Serveur-synchronisation CRON[459]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Jan 20 09:25:01 Serveur-synchronisation CRON[458]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Jan 20 09:25:01 Serveur-synchronisation CRON[461]: (root) CMD (/var/local/index.html root@192.168.20.1:/var/www/localhost/htdocs/index.html)
Jan 20 09:25:01 Serveur-synchronisation CRON[462]: (root) CMD (/var/local/index.html root@192.168.20.2:/var/www/localhost/htdocs/index.html)
Jan 20 09:25:01 Serveur-synchronisation CRON[463]: (root) CMD (/var/local/index.html root@192.168.20.3:/var/www/html/index.html)
Jan 20 09:25:01 Serveur-synchronisation CRON[460]: pam_unix(cron:session): session closed for user root
Jan 20 09:25:01 Serveur-synchronisation CRON[459]: pam_unix(cron:session): session closed for user root
Jan 20 09:25:01 Serveur-synchronisation CRON[458]: pam_unix(cron:session): session closed for user root
You have new mail in /var/mail/root
root@Serveur-synchronisation:~#
```

Le transfert devrait être effectif. On check par la même occasion ce qui se passe sur notre navigateur.



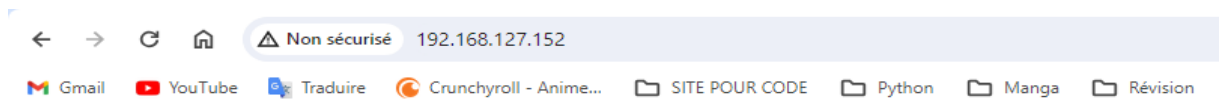
Ceci correspond au fichier index.html synchronise

Serveur-Alpine-1



Ceci correspond au fichier index.html synchronisé

Serveur-Alpine-2



Ceci correspond au fichier index.html synchronisé

Serveur-Alpine-2

Comme on le voit au-dessus notre fichier index.html a bien été synchronisé.

Loadbalancing et Backup Server

Installation et configuration du serveur Nginx

(Le loadbalancing permet la répartition des charges sur plusieurs serveurs.)

Sur notre 4ème conteneur on va installer un serveur Nginx qui nous servira de loadbalancing. On va se rendre dans le fichier de configuration de notre serveur Nginx qui se situe dans le fichier suivant :

`nano /etc/nginx/sites-available/default`

Dans ce fichier on va rajouter différentes lignes de code :

```
GNU nano 7.2 /etc/nginx/sites-available/default *
upstream backend {
    server 192.168.127.150:80;
    server 192.168.127.151:80;
    server 192.168.127.152:80;
}

server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;

    server_name _;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
        proxy_pass http://backend;
    }
    # pass PHP scripts to FastCGI server
    #
```

Le bloc upstream backend, permet de renseigner l'adresse de nos serveurs présents dans le backend. Ce sont ces serveurs qui seront soumis au loadbalancing.

La ligne de code présent dans le bloc location permet d'indiquer que le proxy se fera sur le bloc upstream « backend »

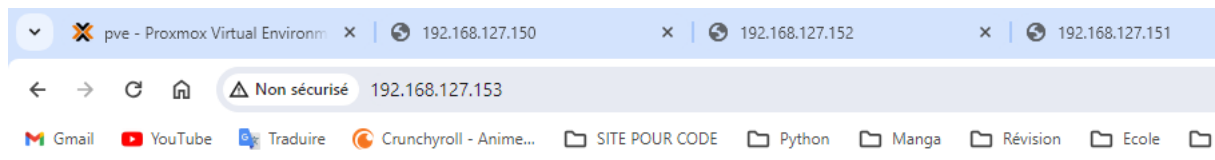
Pour vérifier que notre syntaxe est correcte, on peut utiliser la commande suivante :

nginx -t

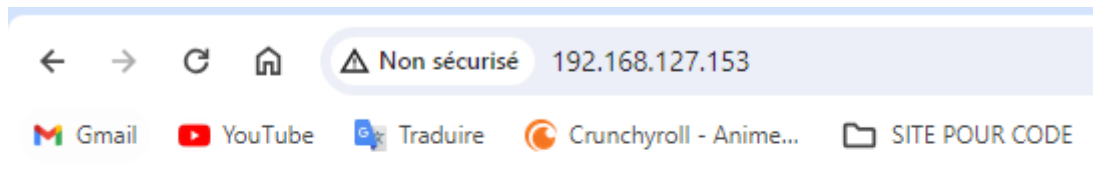
```
root@Loadbalancer:~# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Pour voir si notre loadbalancing fonctionne correctement j'ai modifié le html de mes serveurs Debian et Alpine-2. Et nous pouvons remarquer qu'à chaque actualisation le loadbalancing fonctionne correctement et alterne bien entre chaque serveur.

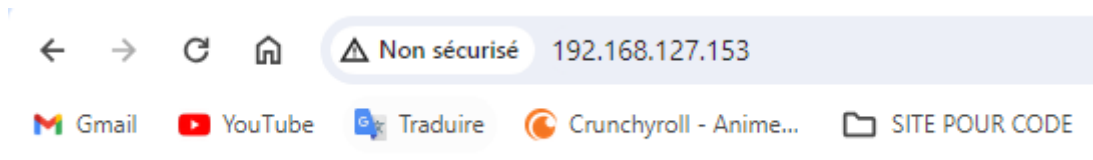
(Le serveur-Alpine-1 n'a pas changé, car je n'ai pas changé son html)



Ceci correspond au fichier index.html synchronisé



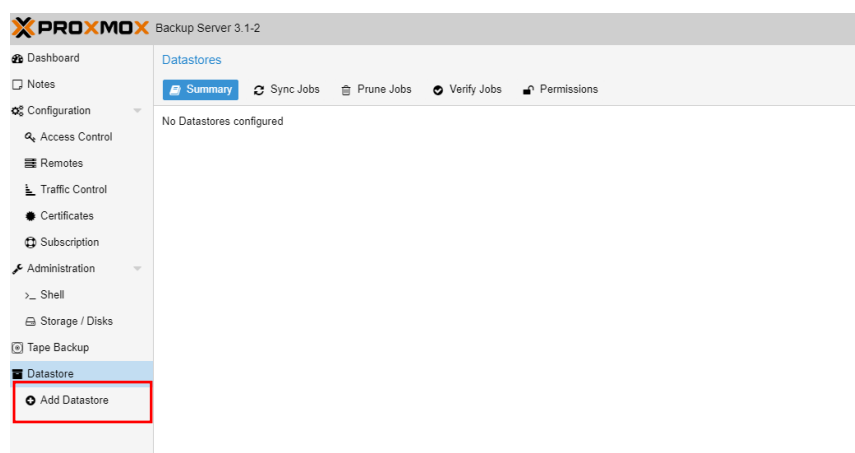
Serveur-Alpine2



Serveur1-DEBIAN test

Mise en place du Backup Server

On se rend sur notre serveur back-up proxmox et on va créer un datastore. Ce datastore sera tout simplement l'emplacement où seront stockées nos sauvegardes.



Add: Datastore

General
Prune Options

Name: Backup
GC Schedule: daily

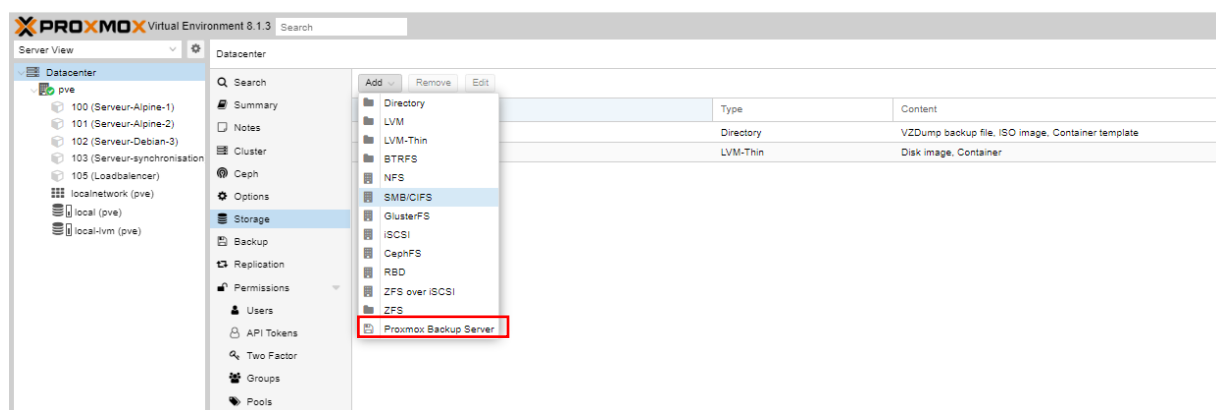
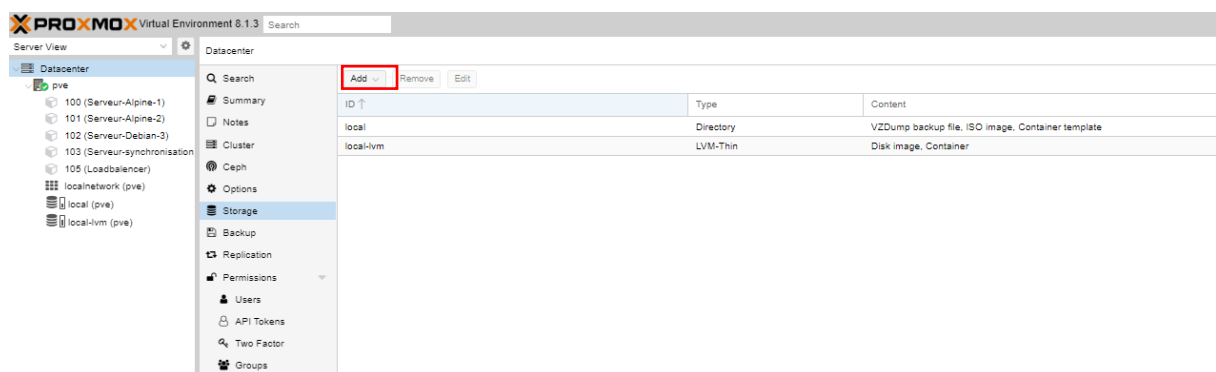
Backing Path: /mnt/datastore/backup-pbs
Prune Schedule: daily

Comment: Stockage des sauvegardes

Help
Add

On indique dans le backing path le chemin où sera stocké la sauvegarde.

Lorsque cela est fait, on revient sur notre proxmox serveur où l'on va ajouter notre backup à notre stockage. Le stockage sera lié à notre PBS.



Add: Proxmox Backup Server

General Backup Retention Encryption

ID: Nodes:

Server: Enable: ☒

Username: Content:

Password: Datastore:

Namespace:

Fingerprint:

[Help](#) [Add](#)

(Le fingerprint se trouve dans notre proxmox backup serveur)

PROXMOX Backup Server 3.1-2

Dashboard

Notes

Configuration

Access Control

Remotes

Traffic Control

Certificates

Subscription

Administration

Shell

Storage / Disks

pbs (Uptime: 00:30:07)

[Show Fingerprint](#)

CPU usage: 0.82% of 2 CPU(s) IO delay: 0.00%

RAM usage: 15.73% (302.42 MiB of 1.88 GiB) Load average: 0.0.0

HD space(root): 15.93% (2.13 GiB of 13.36 GiB) SWAP usage: 0.00% (0 B of 1.75 GiB)

CPU(s): 2 x 12th Gen Intel(R) Core(TM) i5-12400F (2 Sockets)

Kernel Version: Linux 6.5.11-6-pve (2023-11-29T08:32Z)

Boot Mode: Legacy BIOS

Repository Status: ✔ Production-ready Enterprise repository enabled ⚠ Enterprise repository needs valid subscription

Datastore Usage

| Name | Size |
|--------|----------|
| Backup | 12.66 GB |

Maintenant on peut mettre en place notre sauvegarde automatique des conteneurs tous les jours à 17 heures.

On se rend dans la section backup de notre datacenter et on crée une tâche de sauvegarde.

PROXMOX Virtual Environment 8.1.3

Datacenter

Server View

Search

Backup

création d'une tâche de

Add Remove Edit Job Detail Run now

| Enabled | Node | Schedule | Next Run | Storage | Comment | Retention |
|---------|------|----------|----------|---------|---------|-----------|
|---------|------|----------|----------|---------|---------|-----------|

Ici, on sélectionne donc l'ensemble de nos conteneurs que l'on souhaite sauvegarder dans notre backup. Et bien évidemment, on planifie la sauvegarde.

Create: Backup Job

General

Retention

Note Template

Node:

pve

X

▼

Storage:

backup

▼

Schedule:

17:00

▼

Selection mode:

Include selected VMs

▼

Notification mode:

Default (Auto)

▼

Send email:

Always

▼

Send email to:

▼

Compression:

ZSTD (fast and good)

▼

Mode:

Snapshot

▼

Enable:

☒

Job Comment:

| <input checked="" type="checkbox"/> | ID ↑ | Node | Status | Name | Type |
|-------------------------------------|------|------|---------|-----------------------|---------------|
| <input checked="" type="checkbox"/> | 100 | pve | stopped | Serveur-Alpine-1 | LXC Container |
| <input checked="" type="checkbox"/> | 101 | pve | stopped | Serveur-Alpine-2 | LXC Container |
| <input checked="" type="checkbox"/> | 102 | pve | stopped | Serveur-Debian-3 | LXC Container |
| <input checked="" type="checkbox"/> | 103 | pve | stopped | Serveur-synchronis... | LXC Container |
| <input checked="" type="checkbox"/> | 105 | pve | stopped | Loadbalancer | LXC Container |

Help

Advanced ☐

Create

Notre tâche de backup est bien créée.

| <div>AddRemoveEditJob DetailRun now</div> | | | | | | | |
|-------------------------------------------|------|----------|---------------------|---------|---------|------------------------------|---------------------|
| Enabled | Node | Schedule | Next Run | Storage | Comment | Retention | Selection |
| ✓ | pve | 17:00 | 2024-01-21 17:00:00 | backup | | Fallback from storage config | 100,101,102,103,105 |