Debian

- Sécurisation d'un VPS sous Debian 10
- SFTP
- Générer un certificat Wildcard
- Créer un alias sous Linux
- Bind en tant que DNS secondaire d'un DNS sous Windows
- Etendre disque

Sécurisation d'un VPS sous Debian 10

Mise à jour

Il est important de mettre à jour régulièrement Debian.

```
apt update && apt upgrade -y
```

Accès SSH

On gère ensuite la connexion SSH.

On génère une clé SSH.

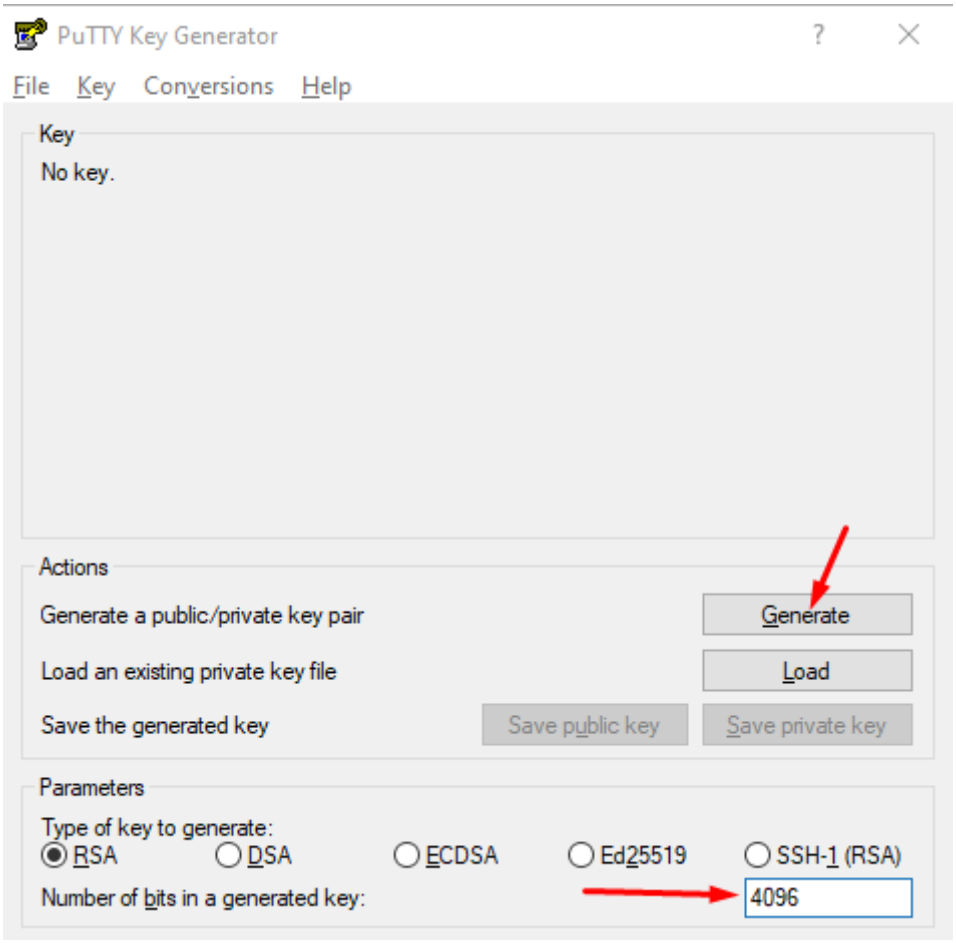
```
ssh-keygen
```

On rentre un passphrase.

On copie ensuite le contenu de la clé privée dans notre outil de connexion SSH (PuTTY par exemple).

```
cat ~/.ssh/id_rsa.pub
```

On peut également le faire à l'aide de PuTTYgen.



On copie le contenu de la clé dans le fichier /home/$USER/.ssh/authorized_keys.

On rentre une passphrase, on clique sur "Save Private Key".

On modifie ensuite le fichier de configuration sshd.

```
nano /etc/ssh/sshd_config
```

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
```

```
PasswordAuthentication no
```

On modifie également le port SSH.

Fail2ban

On installe fail2ban pour éviter les attaques via SSH.

```
apt install fail2ban
```

Pare-feu

On installe iptables.

```
apt install iptables
```

```
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p tcp --dport 2022  -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 80  -m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp --dport 443  -m state --state NEW -j ACCEPT
iptables -A INPUT -p ICMP --icmp-type 8 -j ACCEPT
iptables -A INPUT -j DROP
ip6tables -A INPUT -i lo -j ACCEPT
ip6tables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
ip6tables -A INPUT -p tcp --dport 2022  -m state --state NEW -j ACCEPT
ip6tables -A INPUT -p tcp --dport 80  -m state --state NEW -j ACCEPT
ip6tables -A INPUT -p tcp --dport 443  -m state --state NEW -j ACCEPT
ip6tables -A INPUT -p ICMPV6 --icmpv6-type 8 -j ACCEPT
ip6tables -A INPUT -j DROP
```

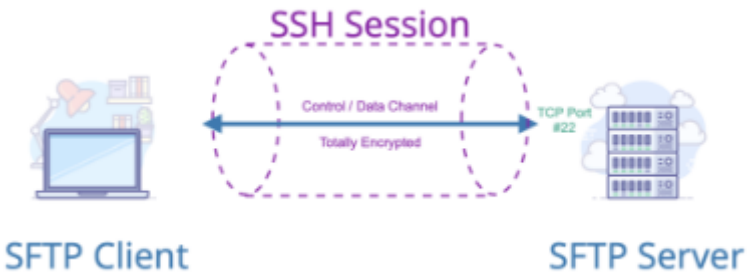
```
apt install iptables-persistent
iptables-save > /etc/iptables/rules.v4
ip6tables-save > /etc/iptables/rules.v6
```

Pour ajouter une règle à la ligne 7 :

```
iptables -I INPUT 7 -p tcp -m tcp --dport 3000 -j ACCEPT
```

SFTP

SFTP, pour Secure File Transfer Protocol (ou protocole de transfert de fichiers SSH). Un tunnel est alors créé et chiffré.



(Source : https://exavault.medium.com/what-is-sftp-the-smart-business-file-transfer-solution-3763174503c0)

Pour la mise en place, on va créer un utilisateur, créer le répertoire sftp et modifier les autorisations (interdire l'accès en shell par exemple).

```
adduser sftpuser
#mot de passe : sisrsisr
```

On crée ensuite le répertoire et les droits/permissions.

```
mkdir -p /var/sftp/uploads
chown root:root /var/sftp
chmod 755 /var/sftp
chown sftpuser:sftpuser /var/sftp/uploads
```

On restreint ensuite l'accès à ce répertoire.

```
nano /etc/ssh/sshd_config
```

On rajoute en fin de fichier :

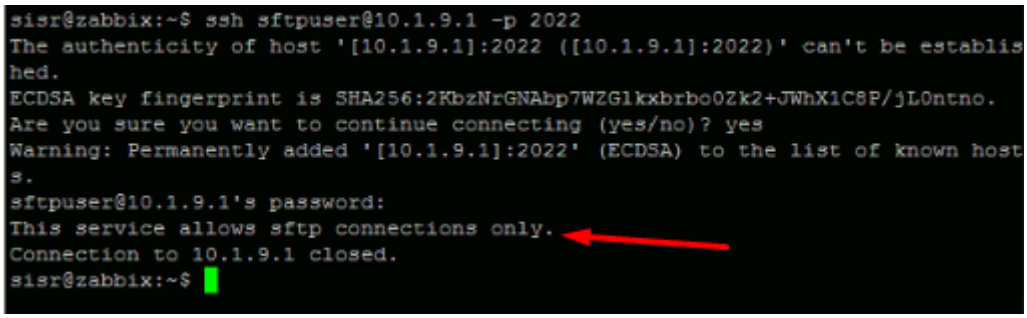
```
Match User sftpuser
ForceCommand internal-sftp
```

```
PasswordAuthentication yes
ChrootDirectory /var/sftp
PermitTunnel no
AllowAgentForwarding no
AllowTcpForwarding no
X11Forwarding no
```

On redémarre le service sshd.

```
systemctl restart sshd
```

On vérifie :



Générer un certificat Wildcard

Installation des dépendances

```
apt install python3-pip
apt install certbot
apt install python3-certbot-dns-ovh
```

Création des accès à l'API d'OVH

On se rend sur le site de l'API d'OVH pour créer nos clés. Voici les droits à attribuer :

- `GET /domain/zone/`
- `GET: /domain/zone/{domain}/`
- `GET /domain/zone/{domain}/status`
- `GET /domain/zone/{domain}/record`
- `GET /domain/zone/{domain}/record/*`
- `POST /domain/zone/{domain}/record`
- `POST /domain/zone/{domain}/refresh`
- `DELETE /domain/zone/{domain}/record/*`

On remplace {domain} par notre nom de domaine, sans les accolades. Par exemple : exemple.fr.

Accound ID correspond à votre Nichandle, présent dans l'interface de gestion de compte OVH, sous la forme xx11111-ovh.

On obtient ensuite une liste d'identifiants : Application Key, Application Secret, Consumer Key.

Sur notre serveur, on crée un fichier ".ovhapi". A titre d'exemple, je le place dans /root/.

```
nano /root/.ovhapi
```

On rentre les identifiants sous cette forme :

```
dns_ovh_endpoint = ovh-eu
dns_ovh_application_key = xxx
dns_ovh_application_secret = xxx
dns_ovh_consumer_key = xxx
```

On attribue ensuite les droits sur ce fichier.

```
chmod 600 /root/.ovhapi
```

Génération du certificat Wildcard

La génération du certificat s'effectue en une seule ligne de commande :

```
certbot certonly --dns-ovh --dns-ovh-credentials ~/.ovhapi -d exemple.fr -d *.exemple.fr
```

Pensez bien à remplacer “exemple.fr” par votre nom de domaine.

Dans ce cas, le domaine exemple.fr ainsi que tous ses sous-domaines seront certifiés.

Créer un alias sous Linux

Il faut modifier le fichier de configuration de notre Shell.

- Bash – **~/.bashrc**
- ZSH – **~/.zshrc**
- Fish – **~/.config/fish/config.fish**

On rajoute une ligne :

```
alias docker-restart='docker-compose down && docker-compose up -d'
```

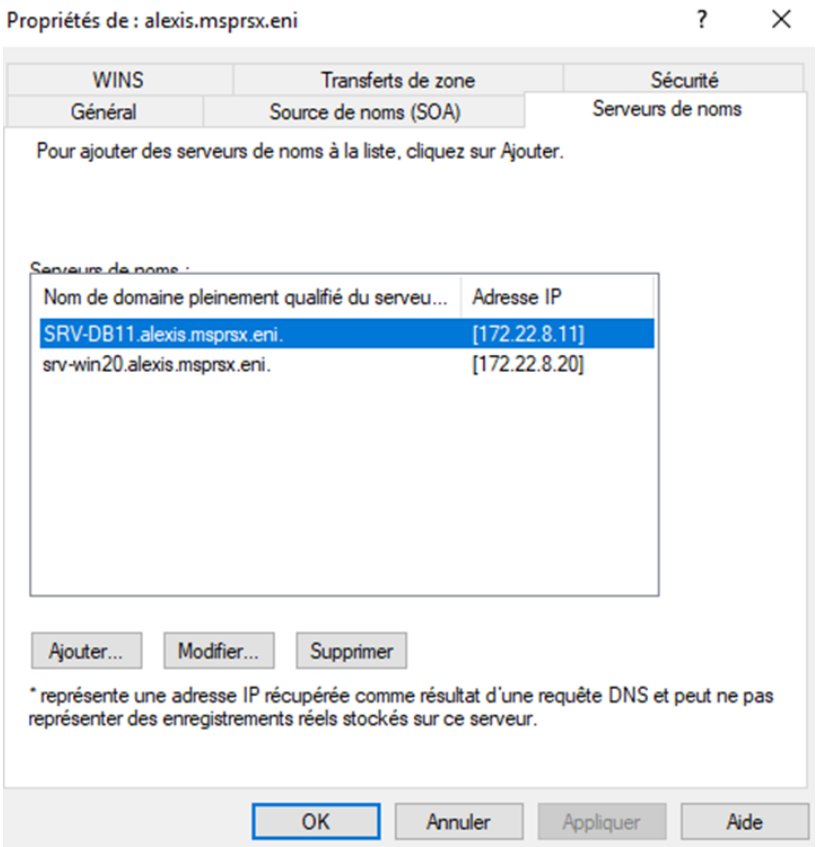
Quand on rentrera la commande "docker-restart", la commande 'docker-compose down && docker-compose up -d' sera lancée.

Ce sera effectif à la prochaine session. Pour l'appliquer tout de suite :

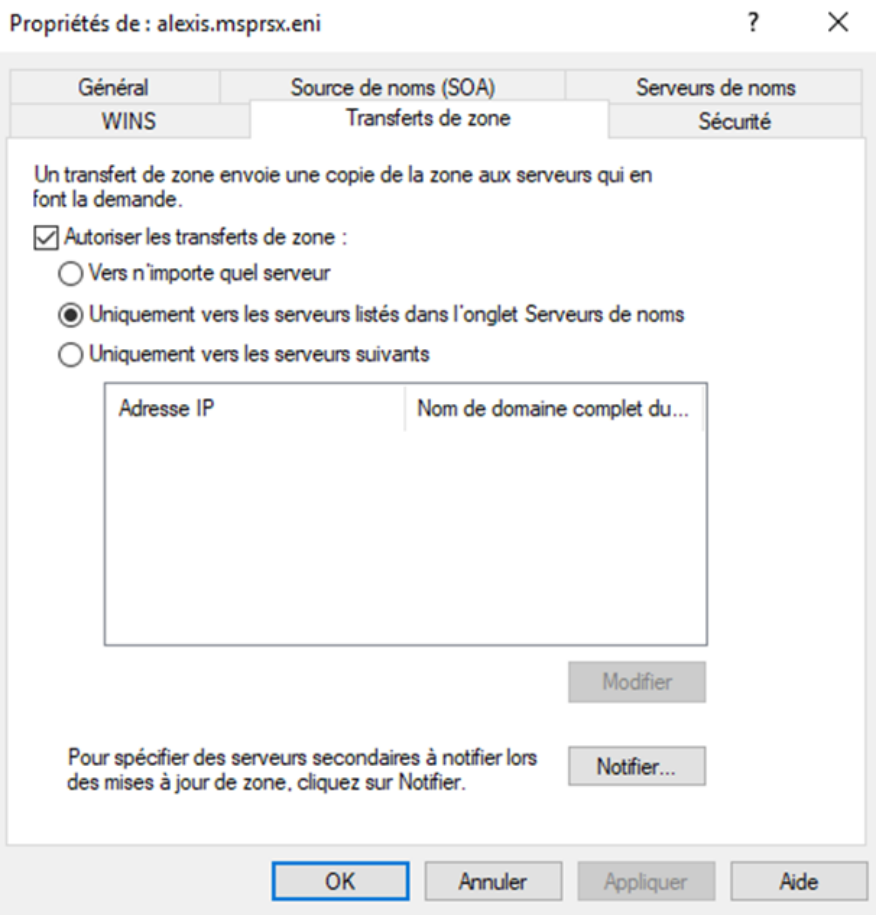
```
source ~/.bashrc
```

Bind en tant que DNS secondaire d'un DNS sous Windows

Dans la console DNS, on ajoute le serveur SRV-DB11 dans la liste des serveurs de noms.



On autorise ensuite le transfert de zone vers les serveurs listés dans l’onglet Serveur de noms.



On répète l’opération pour toutes les zones.

Sur SRV-DB11, on installe bind9.

apt update && apt install bind9 -y

On modifie le fichier « /etc/bind/named.conf.local » pour y déclarer les zones secondaires.

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "alexis.msprsx.eni" IN {
        type slave;
        file "/var/cache/bind/forward.alexis.msprsx.eni.db";
        masters { 172.22.8.20; };
};

zone "_msdcs.alexis.msprsx.eni" IN {
        type slave;
        file "/var/cache/bind/forward._msdcs.alexis.msprsx.eni.db";
        masters { 172.22.8.20; };
};

zone "8.22.172.in-addr.arpa" IN {
        type slave;
        file "/var/cache/bind/reverse.8.22.172.db";
        masters { 172.22.8.20; };
};

zone "8.16.172.in-addr.arpa" IN {
        type slave;
        file "/var/cache/bind/reverse.8.16.172.db";
        masters { 172.22.8.20; };
};
```

On modifie aussi les options globales :

```
options {
        directory "/var/cache/bind";
        allow-query {
                any;
        };
        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.
        masterfile-format text;
        forwarders {
                10.35.0.3;
                172.22.8.20;
        };

        //========================================================================
        // If BIND logs error messages about the root key being expired,
        // you will need to update your keys.  See https://www.isc.org/bind-keys
        //========================================================================
        dnssec-validation auto;

        listen-on-v6 { any; };
};
```

On redémarre ensuite le service et on observe le statut :

```
root@SRV-DB11:/etc/default# systemctl status bind9
● named.service - BIND Domain Name Server
     Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2022-02-07 14:45:23 CET; 4s ago
       Docs: man:named(8)
   Main PID: 8799 (named)
      Tasks: 5 (limit: 2300)
     Memory: 14.2M
        CPU: 48ms
     CGroup: /system.slice/named.service
             └─8799 /usr/sbin/named -f -u bind

févr. 07 14:45:23 SRV-DB11 named[8799]: zone _msdcs.alexis.msprsx.eni/IN: sending notifies (serial >
févr. 07 14:45:23 SRV-DB11 named[8799]: zone 8.22.172.in-addr.arpa/IN: transferred serial 6
févr. 07 14:45:23 SRV-DB11 named[8799]: transfer of '8.22.172.in-addr.arpa/IN' from 172.22.8.20#53:>
févr. 07 14:45:23 SRV-DB11 named[8799]: transfer of '8.22.172.in-addr.arpa/IN' from 172.22.8.20#53:>
févr. 07 14:45:23 SRV-DB11 named[8799]: zone 8.22.172.in-addr.arpa/IN: sending notifies (serial 6)
févr. 07 14:45:23 SRV-DB11 named[8799]: transfer of '8.16.172.in-addr.arpa/IN' from 172.22.8.20#53:>
févr. 07 14:45:23 SRV-DB11 named[8799]: zone 8.16.172.in-addr.arpa/IN: transferred serial 6
févr. 07 14:45:23 SRV-DB11 named[8799]: transfer of '8.16.172.in-addr.arpa/IN' from 172.22.8.20#53:>
févr. 07 14:45:23 SRV-DB11 named[8799]: transfer of '8.16.172.in-addr.arpa/IN' from 172.22.8.20#53:>
févr. 07 14:45:23 SRV-DB11 named[8799]: zone 8.16.172.in-addr.arpa/IN: sending notifies (serial 6)
lines 1-21/21 (END)
```

Etendre disque

fdisk -l

fdisk /dev/sda

n

p

defaut

defaut

w

gdisk /dev/sda

pvcreate /dev/sda4 (sda4 étant la nouvelle partition)

vgextend media-vg /dev/sda4

lvresize -l +100%FREE --resizefs media-vg/root