Serveur / Server

Des astuces un peu en français, un peu en anglais... suivant mon humeur !

- Certificat TLS/SSL avec Cockpit
- Configurer DNSSEC sur un serveur CentOS8 faisant authorité avec Bind □□□□
- Configurer IPv6 sur un VPS OVH sous Red Hat Family
- Configurer un serveur DNS avec BIND sous CentOS 8 □□□□
- Create swape file □□□□
- Insérer sa vérification Mastodon dans BookStack □□□□
- Installation de Certbot □□□□
- Recevoir un e-mail lors d'une connection SSH
- Script de sauvegarde avec Borg

Certificat TLS/SSL avec Cockpit

Contexte

Par défaut, *cockpit* utilise des certificats auto-signés (si je ne dis pas de bêtises). Il est possible de lui faire profiter du certificat du domaine.

Commande

```
sudo cat <main_certificate_path> <private_certificate_path> | sudo tee /etc/cockpit/ws-certs.d/ssl.cert
```

Sources

- Server setup Part-3 Cockpit Setup with SSL Certificates
- How to Fix Permission Denied Error While Using Cat Command - Fedingo

Configurer DNSSEC sur un serveur CentOS8 faisant authorité avec Bind □□□

Prérequis

Vérifier la version de *Bind* et (re)prendre connaissance des emplacements de fichier par défaut :

```
named -V
```

```
BIND 9.11.26-RedHat-9.11.26-6.el8 (Extended Support Version) <id:3ff8620>
running on Linux x86_64 4.18.0-305.17.1.el8_4.x86_64 #1 SMP Wed Sep 8 14:00:07 UTC 2021
built by make with [...]

default paths:
  named configuration:  /etc/named.conf
  rndc configuration:   /etc/rndc.conf
  DNSSEC root key:      /etc/bind.keys
  nsupdate session key: /var/run/named/session.key
  named PID file:       /var/run/named/named.pid
  named lock file:      /var/run/named/named.lock
  geoip-directory:      /usr/share/GeoIP
```

On voit ainsi que le fichier de configuration principal est `/etc/named.conf`.
Les fichiers de zones sont dans le dossier `/var/named` (l'info est dans le dossier de configuration).

Fichier de configuration

Dans la déclaration des options du fichier de configuration, ajouter les options suivantes :

```
options {
  [...]
  recursion no;
  dnssec-enable yes;
  dnssec-validation yes;
  dnssec-lookaside auto;
};
```

Génération des clés

Je me suis mis en root pour la suite des opérations...

```
sudo su
```

Terminologie

Source : DNSSEC - Signer la zone DNS de l'Active Directory #B_Terminologie

- **RRSIG** - ***R**esource **R**ecord **Sig**nature*
  - Enregistrement signé et associé à un enregistrement d'origine (ce que l'on verra dans la console DNS à la fin de la configuration)
- **NSEC3** - ***N**ext **Sec**ure **3***
  - Mécanisme pour prouver qu'un enregistrement n'existe pas (on ne retourne pas rien, mais on retourne une réponse négative signée)
- **DNSKEY** - ***DNS Key***
  - Stocker la clé publique (SHA256) pour permettre la vérification de la signature
- **DS** - ***D**elegation **S**igner*
  - Créer une délégation sécurisée (chaîne d'authentification sur les zones enfants)

Par ailleurs, il ne faut pas négliger le principe des clés KSK et ZSK :

- **Clé KSK** - ***K**ey **S**igning **K**ey*
  - Clé privée qui sert à signer les clés privées ZSK
- **Clé ZSK** - ***Z**one **S**igning **K**ey*
  - Clé privée pour signer les données d'une zone

Se rendre dans le dossier de zone

```
cd /var/named
```

Créer la clés *Zone Signing Key* (ZSK)

D'après la doc.

```
dnssec-keygen -a ECDSAP256SHA256 -n ZONE <domain.name>
```

Ce qui produit la sortie suivante :

```
Generating key pair.
K<domain.name>.+xxx+xxxxx
```

⚠ À noter la lettre `K`.

La clés publique doit être insérée dans le fichier de zone :

```
vim /var/named/zone.<domain.name>
```

```
[...]


$INCLUDE /var/named/K<domaine.name>.+xxx+xxxxx.key
```

Créer la clés *Key Signing Key* (KSK)

Problème

En continuant à suivre la documentation de *bind9*, à savoir :

```
dnssec-signzone -o <domain.name> zone.<domain.name>
```

Je me suis retrouvé avec une erreur :

```
dnssec-signzone: fatal: No self-signed KSK DNSKEY found.  Supply an active
key with the KSK flag set, or use '-P'.
```

Solution

D'après les instructions trouvées sur le site www.digitalocean.com.

```
dnssec-keygen -f KSK -a ECDSAP256SHA256 -n ZONE <domain.name>
```

```
Generating key pair.
K<domain.name>.+xxx+xxxxx
```

La clés publique doit également être insérée dans le fichier de zone :

```
vim /var/named/zone.<domain.name>
```

```
[...]


[...]
$INCLUDE K<domaine.name>.+xxx+xxxxx.key
```

Signer la zone

D'après la doc.

Pour le coup, il devient possible de signer la zone :

```
dnssec-signzone -o <domain.name> zone.<domain.name>
```

```
Verifying the zone using the following algorithms: ECDSAP256SHA256.
Zone fully signed:
Algorithm: ECDSAP256SHA256: KSKs: 1 active, 0 stand-by, 0 revoked
                            ZSKs: 1 active, 0 stand-by, 0 revoked
zone.<domain.name>.signed
```

Ce qui crée un nouveau fichier qu'il faut renseigner dans le fichier de configuration local des zones :

```
vim /etc/named.conf.local
```

```
zone "<domain.name>" {
  type master;
  file "zone.<domain.name>.signed";
};
```

En fait, il suffit de rajouter le `.signed` au nom de zone précédemment renseigné.

Récap des fichiers créés

Le dossier de zone contient donc 6 nouveaux fichiers :

1. `K<domaine.name>.+xxx+xxxxx.key` pour la clés ZSK
2. `K<domaine.name>.+xxx+xxxxx.private` pour la clés ZSK
3. `K<domaine.name>.+xxx+xxxxx.key` pour la clés KSK
4. `K<domaine.name>.+xxx+xxxxx.private` pour la clés KSK
5. `zone.<domain.name>.signed` pour la signature de zone
6. `dsset-<domain.name>.` contenant des *DS Record*

Récupérer le `DS Record`

Deux façons de récupérer le `DS Record` pour procéder à l'enregistrement auprès de son fournisseur de zone de nom de domaine supérieur (*désolé, je ne sais pas mieux m'expliquer à l'instant*) :

dnssec-dsfromkey

```
dnssec-dsfromkey -2 K<domain.name>.+xxx+xxxxx.key
```

Où la clés correspond à la dernière clés créée, c'est à dire KSK.

cat

```
cat dsset-<domain.name>.
```

Pour ma part, j'ai choisi la première méthode afin de ne pas avoir de tabulation dans ma sortie.

Enfin

On relance `bind` :

```
systemctl restart named
```

Quitter l'utilisateur `root`...

On peut vérifier avec les sites https://dnssec-analyzer.verisignlabs.com, https://dnschecker.org, https://www.nslookup.io/...

Ou en ligne de commande :

```
dig @9.9.9.9 <domain.name>. DNSKEY +dnssec +cd +multiline
```

```
; <<>> DiG 9.10.6 <<>> @9.9.9.9 <domaine.name>. DNSKEY +dnssec +cd +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: <number>
;; flags: qr rd ra ad cd; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;<domaine.name>.□IN DNSKEY

;; ANSWER SECTION:
<domaine.name>.□43200 IN DNSKEY□257 3 13 (
□□□□1WclJbuqBHhaiqL0W9EO7ZaLd9aaBe1BGXNbE4cjVqutjH1T
□□□□XZP9kFzcQopEZjwIbvv2LT18tB6GdvBfyyXxV3cA==
□□□□) ; KSK; alg = ECDSAP256SHA256 ; key id = <KSK_Id>
<domaine.name>.□43200 IN DNSKEY□256 3 13 (
□□□□RcFlnT/aqL+UNcrLlSe5DFRuX5Srhie5UYkNbnm+5J4rZE
□□□□3qRTKSQqqpKrxb4XwtdsSZ1wl5zixer3XR2ngXqPSw==
□□□□) ; ZSK; alg = ECDSAP256SHA256 ; key id = <ZSK_Id>
<domaine.name>.□43200 IN RRSIG DNSKEY 13 4 86400 (
□□□□<expiration_date> <creation_date> 20998 <domaine.name>.
□□□□XPTzd2+tTxVfhVrz2+haiqLKdfXx5QyWPwFfi+utUKTK5oPGf
□□□□1WwdRdwDy3Pu8P7Wc55ZBEaqyayM+BJsUGPp2XBnovAXEpqvQ== )
<domaine.name>.□43200 IN RRSIG DNSKEY 13 4 86400 (
□□□□<expiration_date> <creation_date> 61619 <domaine.name>.
□□□□3fac0v2znWPzv777Wc55ZBi6yTZTf7Wc55ZB4N6ckds8VjZqIlq
□□□□3kpRF3texF4bZi2zh4Bw2o1CuWPwFfi+5kJfh0lzfg== )

;; Query time: 359 msec
;; SERVER: 9.9.9.9#53(9.9.9.9)
;; WHEN: Thu Jan 20 22:08:13 CET 2022
```

```
;; MSG SIZE  rcvd: 435
```

Le *fournisseur de zone de nom de domaine supérieur* peut avoir un peu de délai avant de procéder à l'enregistrement.

Renouvellement des dates d'expiration

Manuellement

On renouvelle les clés :

```
dnssec-signzone -o <domain.name> -k /var/named/K<domain.name>.+013+<KSK_Id>.key
/var/named/zone.<domain.name> /var/named/K<domain.name>.+013+<ZSK_Id>.key
```

```
Verifying the zone using the following algorithms: ECDSAP256SHA256.
Zone fully signed:
Algorithm: ECDSAP256SHA256: KSKs: 1 active, 0 stand-by, 0 revoked
                            ZSKs: 1 active, 0 stand-by, 0 revoked
zone.<domain.name>.signed
```

On relance Bind9 :

```
systemctl reload named.service
```

Script

```
#!/bin/bash


declare -r ZONE_KEY_DIRECTORY="/var/named/"


declare -a ZONE_DOMAINS=("<domain1.name>" "<domain2.name>")
declare -a ZONE_NAMES=("zone.<domain1.name>" "zone.<domain2.name>")
declare -a ZONE_KEYS_KSK=("K<domain1.name>.+xxx+xxxxx.key" "K<domain2.name>.+xxx+xxxxx.key")
declare -a ZONE_KEYS_ZSK=("K<domain1.name>.+xxx+xxxxx.key" "K<domain2.name>.+xxx+xxxxx.key")
```

```
cd $ZONE_KEY_DIRECTORY || exit 1

# Loop on domain name zones
for (( i=0; i < "${#ZONE_DOMAINS[@]}"; i++ ));
do
    # Search for serial number
    SERIAL=$(grep -Po '\d{10}(?= {3}; sn)' "${ZONE_NAMES[$i]}")
    if [[ ! "$SERIAL" =~ ^[[:digit:]]+$ ]]; then exit 1; fi
    SERIAL_NEW=$((SERIAL + 1))
    # Increment serial number
    sed -i "s/""$SERIAL""/""$SERIAL_NEW""/" "${ZONE_NAMES[$i]}"

    /usr/sbin/dnssec-signzone -o "${ZONE_DOMAINS[$i]}" -k "${ZONE_KEYS_KSK[$i]}" "${ZONE_NAMES[$i]}"
"${ZONE_KEYS_ZSK[$i]}"
    EXIT_CODE=$?

    if [ $EXIT_CODE -eq 0 ];
    then
        echo "Renew is done correctly for ${ZONE_DOMAINS[$i]}! (SN: $SERIAL_NEW)"
    else
        echo "Something went wrong with ${ZONE_DOMAINS[$i]}... (SN: $SERIAL_NEW)"
        exit 1
    fi
done

systemctl reload named.service
EXIT_CODE=$?
if [ $EXIT_CODE -eq 0 ];
then
    echo "Everything seems correct."
    exit 0
else
    echo "There is an error with Bind!!!"
    exit 1
fi
```

On lance un *cronjob* :

```
0 4 25 * * /bin/sh /root/.scripts/renew-dnskey.sh
```

Différentes sources ou ressources

- DNSSEC Guide — BIND 9 documentation - #Semi automatic signing
- DNSSEC Guide — BIND 9 documentation - #Setting key timing information
- DNSSEC Guide — BIND 9 documentation - #Manual signing
- How To Setup DNSSEC on an Authoritative BIND DNS Server DigitalOcean - #Modifying zone records
- DNSSEC - Signer la zone DNS de l'Active Directory

Configurer IPv6 sur un VPS OVH sous Red Hat Family

Obtenir les informations nécessaires

Il y a deux façons de faire...

Panneau de contrôle

Se connecter au panneau de contrôle de son compte : OVHcloud Control Panel. Puis récuperer l'***IPv6*** et le ***Gateway***.

Source : https://docs.ovh.com/us/en/vps/configuring-ipv6/

API OVHCloud

Se rendre sur la console https://API.OVH.com.

1. Dans la section GET /vps/{serviceName}/ips récupérer l'IPv6 du VPS.
2. Dans la section GET /vps/{serviceName}/ips/{ipAddress} récupérer le *Gateway*.

Appliquer la configuration IPv6

La configuration de réseau se trouve dans le fichier `/etc/sysconfig/network-scripts/ifcfg-eth0`. Il est conseillé de faire une sauvegarde...

Copie de sauvegarde

```
sudo mkdir /etc/sysconfig/network-scripts/backup
cp /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/backup/ifcfg-eth0.bak
```

Éditer le fichier

```
sudo vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

Ajouter les informations suivantes :

```
IPV6INIT=yes
IPV6ADDR=<YOUR_IPV6>/<IPV6_PREFIX>
IPV6_DEFAULTGW=<IPV6_GATEWAY>
```

Normalement l'`IPV6_PREFIX` est `128`.

Relancer le service de réseau

```
sudo systemctl restart NetworkManager
```

Vérification

Sur le serveur

```
ip -6 addr show eth0
```

Depuis un autre ordinateur

```
ping6 <YOUR_IPV6>
```

Depuis un navigateur

Si le serveur écoute sur le port 80 ou 443, entrer l'adresse IPv6 entre crochets (`[YOUR_IPV6]`) dans la barre d'adresse.

Désactiver le Cloud-init

Le *Cloud-init* permet lancer un script au démarrage des VPS, notamment pour gérer le *hostname*, le *resolve.conf* ou le partitionnement automatique en cas de redimentionnement.

```
echo "network: {config: disabled}" | sudo tee /etc/cloud/cloud.cfg.d/98-disable-network-config.cfg
```

Pour retrouver une gestion automatique, il suffit de supprimer le fichier, changer l'extension ou le déplacer.

Source

- Configuring IPv6 on a VPS OVH Guides.
- How to Restart Network Service on CentOS 8 or RHEL 8 – TecAdmin.

Configurer un serveur DNS avec BIND sous CentOS 8

Tuto depuis blog.microlinux.fr.

Installation

```
$ sudo yum install bind bind-utils
```

Serveur cache DNS

1. On fait une sauvegarde de fichier d'origine.
2. On édite un nouveau fichier named.conf

```
$ sudo mv /etc/named.conf /etc/named.conf.orig
$ sudo vim /etc/named.conf
```

```
// etc/named.conf
options {
  directory "/var/named";
};

// journalisation propre à BIND
logging {
  channel single_log {
    file "/var/log/named/named.log" versions 3 size 2m;
    severity info;
    print-time yes;
    print-severity yes;
    print-category yes;
```

```
  };
  category default {
    single_log;
  };
};

zone "." IN {
  type hint;
  file "named.ca";
};

include "/etc/named.rfc1912.zones";
// zone DNS afin de déclarer en serveur maître primaire
include "/etc/named.conf.local";
```

1. On attribut user:group du fichier.
2. On règle les permissions du fichier.
3. On active et démarre BIND.
4. On vérifie si le service tourne correctement.

```
$ sudo chown root:named /etc/named.conf
$ sudo chmod 0640 /etc/named.conf
$ sudo systemctl enable named --now
$ systemctl status named
● named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2021-01-11 14:31:42 CET; 1 day 1h ago
```

Configurer la journalisation

Comme vu précédemment, la journalisation propre à BIND a été mise en place. Cependant, il ne peut pas créer ce fichier à la volée.
On le crée à sa place en attribuant les permissions correctes.

```
$ sudo mkdir /var/log/named
$ sudo touch /var/log/named/named.log
$ sudo chown -R named:named /var/log/named/
$ sudo chmod 0770 /var/log/named
```

1. Avec SELinux en mode renforcé, on réétiquette le répertoire.
2. On recharge la configuration de BIND.

```
$ sudo restorecon -R -v /var/log/named
$ sudo systemctl reload named
```

Serveur maître primaire

```
$ sudo vim /etc/named.conf.local
```

```
zone "<domaine.name>" {
  type master;
  file "zone.<domaine.name>";
};
```

On attribue les mêmes permissions que `named.conf`.

```
$ sudo chown root:named /etc/named.conf.local
$ sudo chmod 0640 /etc/named.conf.local
```

On édite le fichier `zone.<domaine.name>`.

```
$ sudo vim /var/named/zone.<domaine.name>
```

```
; /var/named/zone.<domaine.name>
$TTL 86400
$ORIGIN <domaine.name>.
@ IN SOA ns.<domaine.name>. <nom>.<domaine.name>. (
   2021011201   ; sn: serial number must be incremented each time
        10800   ; refresh (3 heures)
          600   ; retry (10 minutes)
      1814400   ; expiry (3 semaines)
        10800 ) ; minimum (3 heures)
□□□□□IN      NS      ns.<domaine.name>.
ownercheck□□□IN      TXT     "<hash>" ; to check owner with provider
<domaine.name>.□□IN      AAAA    2001:41d0:305:2100::100e
<domaine.name>.□□IN      A       51.38.177.69
ns□□□□□IN      AAAA    2001:41d0:305:2100::100e
ns□□□□□IN      A       51.38.177.69
shaarli□□□□IN      AAAA    2001:41d0:305:2100::100e
shaarli□□□□IN      A       51.38.177.69
www□□□□□IN      CNAME   <domaine.name>.
```

```
www.shaarli□□□IN      CNAME   shaarli
```

> La partie `<nom>.<domaine.name>.` correspond à une adresse e-mail joignable. Étant donné que le `@` à une signification particulière dans le fichier de configuration, il est substitué par un point (`.`). Si l'adresse e-mail contient un permier point de séparation (exemple `jean.dupond@example.com`) il doit être échappé (`jean\.dupond.example.com`).
>
> *Source : SOA Record Explained How to Perform an SOA Record Check (+Example) - IONOS*

1. On règle les droits de propriétés.
2. On règle les permissions qui vont bien.
3. On vérifie la définition correcte de la zone.
4. On recharge la configuration de BIND.
5. On refait une expansion des certificats au besoin.

```
$ sudo chown root:named /var/named/zone.<domaine.name>
$ sudo chmod 0640 /var/named/zone.<domaine.name>
$ sudo named-checkzone <domaine.name> /var/named/zone.<domaine.name>
zone <domaine.name>/IN: loaded serial 2021011201
OK
$ sudo systemctl reload named
$ sudo certbot --nginx # Si certbot est bien installé
```

Create swape file □□□

Tuto from itsfoss.com and linuxhandbook.com.

Check swap space in Linux

```
$ free -h
               total        used        free      shared  buff/cache   available
Mem:           1,7Gi       844Mi       321Mi        18Mi       603Mi       759Mi
Swap:             0B          0B          0B
```

There is no swap space.

Make a new swap file

1. Use the fallocate command to create a file of size 1 GB.
2. It is recommended to allow only root to read and write to the swap file.

```
$ sudo fallocate -l 1G /swapfile
$ sudo chmod 600 /swapfile
```

> The name of the swap file could be anything.

Mark the new file as swap space

```
$ sudo mkswap /swapfile
Configure l'espace d'échange (swap) en version 1, taille = 1024 MiB (1073737728 octets)
pas d'étiquette, UUID=3bbb4a53-e2a4-495f-8e42-d7cb43e6ad40
```

Enable the swap file

1. Enable the swap file.
2. Check the swap space.

```
$ sudo swapon /swapfile
$ swapon --show
NAME      TYPE  SIZE USED PRIO
/swapfile file 1024M   0B   -2
```

Make the changes permanent

1. It’s always a good idea to make a backup before make any changes to the /etc/fstab file.
2. Add the line to the end of /etc/fstab file.

```
$ sudo cp /etc/fstab /etc/fstab.back
```

Change Swapiness kernel parameter

Temporary change

Check *swapiness*:

```
cat /proc/sys/vm/swappiness
```

Change setting:

```
sudo sysctl vm.swappiness=10
```

Persistent change

Edit the config file:

```
sudo vim /etc/sysctl.conf
```

Set swapiness:

```
# /etc/sysctl.conf
vm.swappiness=10
```

Reload *sysctl*:

```
sudo sysctl -p
```

[Bonus] Resize swap file

Make sure the swapfile is on the system and not a swap partition.

```
$ swapon --show
NAME      TYPE SIZE USED PRIO
/swapfile file   1G   0B   -2
```

Now before you resize the swap file, you should turn the swap off.
You should also make sure that you have enough free RAM available to take the data from swap file.
Otherwise, create a temporary swap file.

A solution to clear swapfile is to reboot the system...

Disable the swapfile.

```
$ sudo swapoff /swapfile
```

Now, use the fallocate to change the size of the swap file.

```
$ sudo fallocate -l 2G /swapfile
```

Mark the file as a swap file.

```
$ sudo mkswap /swapfile
mkswap: /swapfile : avertissement : effacement de l'ancienne signature swap.
Configure l'espace d'échange (swap) en version 1, taille = 2 GiB (2147479552 octets)
pas d'étiquette, UUID=bf273c02-f44a-48ef-9b04-b212d9003a4a
```

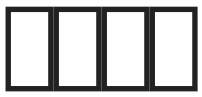
Enable the swap file.

```
$ sudo swapon /swapfile
```

Check!

```
$ free -h
              total        used        free      shared  buff/cache   available
Mem:          1,7Gi       1,0Gi       138Mi        65Mi       619Mi       541Mi
Swap:         2,0Gi       0,0Ki       2,0Gi
```

Insérer sa vérification Mastodon dans BookStack



Contexte

D'après la documentation de *Mastodon* :

> **Vérification**
>
> Vous pouvez vous vérifier en tant que propriétaire des liens dans les métadonnées de votre profil. Pour cela, le site web lié doit contenir un lien vers votre profil Mastodon. Le lien de retour doit avoir un attribut rel="me" . Le texte du lien n’a pas d’importance.

Or l'ajout de liens de page dans les préférences de *Bookstack* ajoute l'attribut `rel="noopener"` et c'est tout...

Manipulation

Préférence de *Mastodon*

- Se rendre dans les préférences de son profil.
- Dans les options de *Métadonnées du profil*, ajouter le nom de l'élément puis le lien internet vers son *BookStack*.
- Enregistrer les préférences.

Préférences de *BookStack*

- Se rendre dans les préférences de l'appli.
- Ajouter le libellé du lien ainsi que l'URL correspondante à son profil *Mastodon* (exple : « https://mstdn.fr/@mickge »).
- Enregistrer les préférences.

Sur le serveur

Utiliser la personnalisation d'après la doc.

Créer un dossier de thème personnalisé avec le dossier qui vont bien aller et y copier le fichier `footer.blade.php` :

```
sudo -u www-data sh -c "mkdir -p <racine>/themes/my_theme/common && cp
<racine>/resources/views/common/footer.blade.php <racine>/themes/my_theme/common/"
```

Modifier le fichier `<racine>/themes/my_theme/common/footer.blade.php` à la ligne 4 par les lignes suivantes :

```
[…]
        <a href="{{ $link['url'] }}" target="_blank"
        @if(strpos(strtolower($link['label']), 'mastodon') !== false)
            rel="noopener me"
        @else
            rel="noopener"
        @endif
        >{{ strpos($link['label'], 'trans::') === 0 ? trans(str_replace('trans::', '', $link['label'])) : $link['label']
}}</a>
[…]
```

Ensuite, changer la variable de thème en ajoutant dans le fichier `<racine>/.env` :

```
# Custom Theme
APP_THEME=my_theme
```

That's it!

Installation de Certbot □□□

⚠ Vérifier que le mode opératoire est toujours valable...

Installation de `snap`

1. On installe `snapd`.
2. On rend opérationnel maintenant le moteur de `snap`.
3. On crée un lien symbolique.

```
$ sudo dnf upgrade && sudo dnf install snapd
$ sudo systemctl enable --now snapd.socket
$ sudo ln -s /var/lib/snapd/snap /snap
```

Installation de `certbot`

```
$ sudo snap install core; sudo snap refresh core
$ sudo dnf remove certbot # On s'assure que l'ancien certbot est bien désintallé
$ sudo snap install --classic certbot
$ sudo ln -s /snap/bin/certbot /usr/bin/certbot
$ sudo certbot --nginx
```

Site officiel ici avec les étapes sous RHEL8 & nginx.

Recevoir un e-mail lors d'une connection SSH

Création du script

Évidemment, il faut que le serveur soit en mesure d'envoyer des e-mails...
(Par exemple `sudo apt install mailutils` sur *Debian Family*.)

```
sudo vim /usr/local/bin/send-mail-on-ssh-login.sh
```

Remplacer `<EMAIL_ADDRESS>` par l'adresse de destination voulue.

```
#!/bin/sh
if [ "$PAM_TYPE" != "open_session" ]
then
  exit 0
else
  {
    echo "User: $PAM_USER"
    echo "Remote Host: $PAM_RHOST"
    echo "Service: $PAM_SERVICE"
    echo "TTY: $PAM_TTY"
    echo "Date: `date`"
    echo "Server: `uname -a`"
  } | mail -s "$PAM_SERVICE login on `hostname -s` for account $PAM_USER" <EMAIL_ADDRESS>
fi
exit 0
```

Rendre le script exécutable.

```
sudo chmod +x /usr/local/bin/send-mail-on-ssh-login.sh
```

Configurer le *PAM* (*Pluggable Authentication Modules*)

```
sudo vim /etc/pam.d/sshd
```

```
session    optional     pam_exec.so /usr/local/bin/send-mail-on-ssh-login.sh
```

Sources

- Send email on SSH login using PAM Yeah!
- linux - Send email when anyone logs on - Server Fault

Script de sauvegarde avec Borg

Pour faire mes sauvegardes j'utilise *BorgBackup*. Les données sont envoyées sur un serveur de sauvegarde.

Les sauvegardes ont lieu quatre fois par jour en lançant le script dans un *cronjob*. S'en suit la réception d'un e-mail à chaque sauvegarde.

Au passage, une fois par jour, le script vérifie les mises à jour disponibles.

```
#!/bin/sh

export BORG_PASSPHRASE='<BORG_PASSPHRASE>'

# some helpers and error handling:
printMessage() { printf "\n*** %s => %s ***\n\n" "$( date +"%A %d %B %T" )" "$*" >&2; }
trap 'echo $( date ) Backup interrupted >&2; exit 2' INT TERM

printMessage "Starting mysql backup"

mysqldump --defaults-file=<variables_file_path> --all-databases > <output_file_path>

printMessage "Starting backup"

/usr/bin/borg create                        \
    --stats                             \
    --filter=AME                        \
    --compression auto,zstd,10          \
    --exclude-caches                    \
    --exclude='/var/lib/{snapd,selinux}'    \
    <backup_server_url>:repo::{now} \
    /var/{www,lib}                      \
    /root                               \
    /etc                                \
    /home
```

```
backup_exit=$?

printMessage "Pruning repository"

# Use the `prune` subcommand to maintain 7 daily, 2 weekly and 2 monthly
# archives of THIS machine. The '{hostname}-' prefix is very important to
# limit prune's operation to this machine's archives and not apply to
# other machines' archives also:

borg prune                                \
    --list                                \
    --show-rc                             \
    --keep-hourly       3                 \
    --keep-daily        7                 \
    --keep-weekly       2                 \
    --keep-monthly      2                 \
    --save-space                          \
    <backup_server_url>:repo

prune_exit=$?

# use highest exit code as global exit code
global_exit=$(( backup_exit > prune_exit ? backup_exit : prune_exit ))

if [ ${global_exit} -eq 0 ]; then
    printMessage "Backup and Prune finished successfully"
elif [ ${global_exit} -eq 1 ]; then
    printMessage "Backup and/or Prune finished with warnings"
else
    printMessage "Backup and/or Prune finished with errors"
fi

if [ $(date "+%H") -eq 6 ] ; then
    printMessage "Check updates"
    dnf check-update
fi

exit ${global_exit}
```