

# CONFIG OS

- Config template
- Hierarchy file system
- Firewall
- Créer un environnement virtuel python
- Créer SERVER local REPOSITORY
- Logrotate
- NMTUI
- Ajout Key SSh
- Ajout Proxy

# Config template

```
[root@localhost ~]# systemctl stop firewalld.service
[root@localhost ~]# systemctl disable firewalld.service
[root@localhost ~]# iptables -F
```

Cat /etc/ntp/step-tickers

Timedatectl status

Timedatectl set-ntp true

hwclock

Iptables -F

Rm -fr .bash\_history

sudo useradd -M -s /sbin/nologin ansible

sudo -u ansible ssh-keygen -t rsa

# Hierarchy file system

/bin : binaire

/lib : librairie de programme

/boot : fichier du système démarrage

/usr :

/usr/local :

/opt : fichier optionnel

/etc : editable text configuration

/srv : service héberger par d'autre système, http, ftp, cvs, DB

/var : donnée type variable, log mail ect

/home : donnée aux utilisateurs spécifique

/root : donné aux système

/tmp : donnée temporaire, pour pouvoir créer un programme dans le système

/mnt : monter tous les différents de système de fichier temporaire

/media : monter les média amovible '(clé usb, cd-rom, ID périphérique)'

/dev : Fichier virtuel, pas besoin de partitions

btrfs : système de fichier plus moderne + révolutionnaire + snapshot

ext : extend file system

reiserfs : premier système de fichier sur linux plus accessible aujd

jfs : système de fichier constante et super élever + gestionnaire de base de donnée

xfs : système de fichier constante et super élever particulièrement sur les entrées et sorties // favorisation pour les grosses hiérarchie à grosse données; meta donnée lente

fat: file allocation table

ntfs : new technologie file system

hfs : héarchie file système

udf : permet de gérer la création, suppression et modification des fichiers de disque optiques

dumpe2fs : récupérer des informations

tune2fs : modifier certain paramètre lié aux information récupérer de deumpe2fs

debugfs : relis tune2fs + dumpe2fs + autres outils = débbugger un système de fichier

xfs\_metadump : permet de copier les métadonnées du système de fichier vers un fichier

vim /etc/cups/cupsd.conf : configuration du pool cups : Imprimante

syslogd = log application

klogd = log pour le kernel k

Tous les messages d'erreur à fouiller : Authpriv, cron, kern, lpr, mail, new, syslog, user, uucp

/var/log

Logger = permet d'enregistrer des informations dans le /var/log/syslog

/etc/init.d/atd "stop" or "start" : pour mettre en actif nos tâches planifiées

# Firewall

## Commande

- **iptables** : Interface de bas niveau, nécessite une gestion manuelle et convient mieux aux systèmes plus anciens ou aux configurations spécifiques.
- **firewalld** : Plus haut niveau, offre une interface dynamique et facile à gérer via des zones.

la commande suivante permet de lister les ports ouverts et les services en écoute sur un système, en utilisant netstat :

```
netstat -tulnp
```

lister les règles/filtrage:

```
iptables -L -n -v
```

```
Firewall-cmd --list-all-zones
```

Pour ajouter une nouvelle règle de pare-feu à la zone par défaut, vous pouvez utiliser les commandes suivante :

```
firewall-cmd --permanent --add-rule type=tcp protocol=80 port=80
```

```
firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="10.242.0.0/27" port  
protocol="tcp" port="22" accept'
```

```
firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="10.242.0.0/27" port  
protocol="tcp" port="389" accept'
```

```
firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="10.242.0.0/27" port  
protocol="tcp" port="636" accept'
```

Pour appliquer les modifications à votre pare-feu, vous pouvez utiliser la commande suivante :

```
firewall-cmd --reload
```

# Créer un environnement virtuel python

Créer un environnement virtual sur sa machine pour ne pas toucher au versionning des paquets directement de votre OS.

Création d'un environnement virtuel Python

```
mkvirtualenv -p /usr/bin/python3 le_nom_de_ton_env
```

Action: Cette commande crée un nouvel environnement virtuel Python nommé "le\_nom\_de\_ton\_env". L'option -p /usr/bin/python3 spécifie que cet environnement utilisera Python 3 comme interpréteur.

```
TMP=/var/tmp pip3 install --upgrade pip
```

Action: Cette commande met à jour le gestionnaire de paquets pip dans l'environnement virtuel nouvellement créé.

L'option TMP=/var/tmp spécifie un répertoire temporaire pour effectuer l'installation.

Activation et utilisation de l'environnement virtuel

```
workon le_nom_de_ton_env
```

Action: Cette commande active l'environnement virtuel que vous avez créé précédemment. Une fois activé, les commandes pip s'exécuteront dans le contexte de cet environnement isolé, vous pouvez exemple installer une nouvelle version d'ansible sur votre environnement virtuel

# Créer SERVER local REPOSITORY

Yum install httpd -y

Mettre en place un disque avec bcp de GB selon les repo que vous allez synchroniser, Exemple ( le repos Appstream = prend 120G)

```
/dev/mapper/rhel-var 371G 170G 202G 46% /var
```

Ouverture des ports firewall-cmd 80/tcp & le service http :

```
services: cockpit dhcpv6-client http squid ssh
ports: 80/tcp
```

Aller dans [/etc/httpd/conf.d/httpd.conf](#) et changer le server name par l'adresse IP de votre server sur lequel vous pourrez acceder depuis internet via cette ip

```
ServerName 10.242.10.74:80
```

N'oublier pas de mettre le port sur le httpd.conf vous pouvez désormais sortir du fichier de conf et activer le service :

`systemctl start httpd.service`

```
Listen 80
```

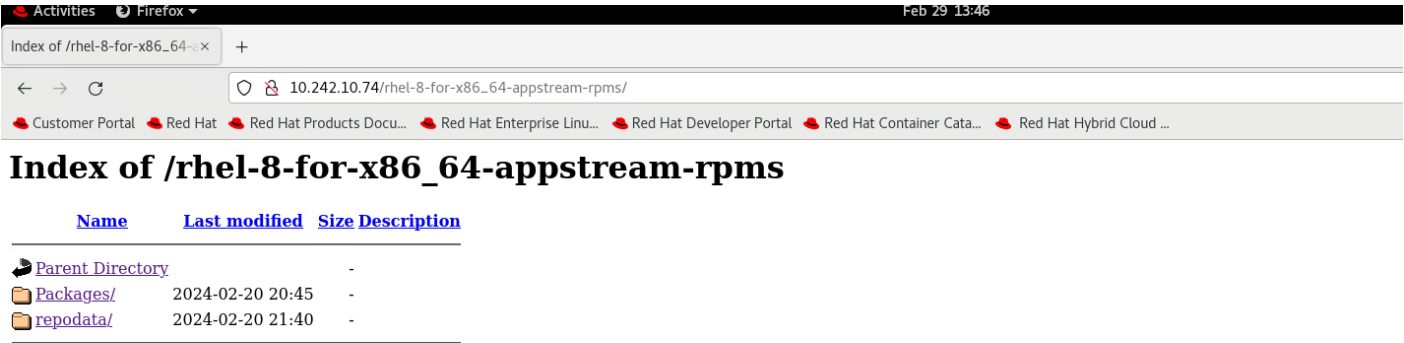
Commande pour sync des repos d'internet sur le site web interne dans le chemin  
/var/www/html/.....

```
177 cd /epel/
189 reposync -p /var/www/html/ --download-metadata --repo=epel
344 reposync -p /var/www/html/ --download-metadata --repo=pgdb-common
194 reposync -p /var/www/html/ --download-metadata --repo=zabbix
```

Après le reposync vous avez normalement quelque chose comme ca sur votre path :

```
/var/www/html/
[root@PRDRHNPRX01 ~]# ll /var/www/html/
total 84
drwxr-xr-x. 4 root root    38 Feb 20 18:04 ansible-2.9-for-rhel-8-x86_64-debug-rpms
drwxr-xr-x. 4 root root    58 Feb 20 17:07 epel
drwxr-xr-x. 3 root root 45056 Feb 26 11:00 pgdb-common
drwxr-xr-x. 4 root root    38 Feb 20 21:40 rhel-8-for-x86_64-appstream-rpms
drwxr-xr-x. 4 root root    38 Feb 26 09:52 rhel-8-for-x86_64-baseos-rpms
drwxr-xr-x. 4 root root    38 Feb 26 11:09 rhel-8-for-x86_64-highavailability-source-rpms
drwxr-xr-x. 3 root root 20480 Feb 20 17:13 zabbix
```

Vous pouvez désormais accéder sur une page web sur vos range ip local de votre server de repos :



Repos Important :

---

- name: ums

hosts: L

tasks:

- name: Create ums8.repo file

ansible.builtin.copy:

- dest: /etc/yum.repos.d/ums8.repo
- content: |
- [UMS8OB]
- name=UMS8 epel
- baseurl=http://ip/epel/
- enabled=yes
- gpgcheck=no





# Logrotate

# cat /etc/logrotate.d/prod

```
/production/www/clients/prod.log{  
    daily  
    missingok  
    compress  
    delaycompress  
    notifempty  
    copytruncate  
    create 644 nom nom  
}
```

```
/production/www/clients/dev.log{  
    daily  
    missingok  
    compress  
    delaycompress  
    notifempty  
    copytruncate  
    create 644 nom nom  
}
```

Pour lancer logrotate:

logrotate -f /etc/logrotate.d/(ta conf)

# NMTUI

#nmtui

Activate a connection

Set system hostname

Edit connection

Wired connection

Check device mac

Ipv4 configuration

Manual > show

Never use this network for ipm

Ipv4 configuration

Manual > show

Automatically connect

Deactivate et reactive pour appliquer les changements

# Ajout Key SSh

```
#!/bin/bash
if [ "$#" -ne 2 ]; then
    echo "Usage: $0 <username> <remote_host>"
    exit 1
fi

username="$1"
remote_host="$2"

### COPY
If ssh-copy-id "$username@$remote_host" 2>/dev/null; then
    cat ~/.ssh/id_rsa.pub | ssh $username@$remote_host 'mkdir -p ~/.ssh && cat >> ~/.ssh/authorized_keys'
    echo " COPY OK to host : $remote_host"
    echo "you can Test with  >>>  ssh $username@$remote_host"
else
    echo "Failed to copy public key on host : $remote_host "
    exit 1
fi
```

# Ajout Proxy

Il faut mettre le script sous /etc/profile.d/proxy.sh

```
PROXY_URL=http://adresseip:port/  
export http_proxy="$PROXY_URL"  
export https_proxy="$PROXY_URL"  
export ftp_proxy="$PROXY_URL"  
export no_proxy="127.0.0.1,localhost"
```

Puis il faut exécuter :

```
source /etc/profile.d/proxy.sh
```