Applications

- Wiki
  - BookStack LAMP
  - BookStack on Docker
- Base SQL
  - PostgreSQL
  - MariaDB
  - MySQL
- FTP
  - ProFTPd
- Base NoSQL
- Gitlab
  - CI/CD
  - CLI git
- OpenSSH
  - SFTP over SSH
- DNS
  - Bind9 | Named
  - Power DNS
- Mailserver
  - Postfix
- Docker

    - Docker
- WAF
    - mod security
- Restic Backup
    - Restic : Déploiement
    - Restic : Ligne de commandes
    - Restic : Annexes
    - Restic : Scénarios de sauvegarde
- Kubernetes
    - TP : Déploiement K8s 3 noeuds avec NFS
- Troubleshooting
    - Médicat

Wiki

BookStack LAMP

Installation: Installing BookStack (Debian 11)

1. Install dependencies.

```
apt install -y git unzip apache2 php7.4 curl php7.4-fpm php7.4-curl php7.4-mbstring
php7.4-ldap \
php7.4-tidy php7.4-xml php7.4-zip php7.4-gd php7.4-mysql libapache2-mod-php7.4 \
default-mysql-server
```

2. Setup database.

```
DB_PASS="$(head /dev/urandom | tr -dc A-Za-z0-9 | head -c 13)"
mysql -u root --execute="CREATE DATABASE bookstack;"
mysql -u root --execute="CREATE USER 'bookstack'@'localhost' IDENTIFIED WITH
mysql_native_password AS PASSWORD('$DB_PASS');"
mysql -u root --execute="GRANT ALL ON bookstack.* TO 'bookstack'@'localhost';FLUSH
PRIVILEGES;"
```

3. Install PHP package manager.

```
# Install composer
EXPECTED_CHECKSUM="$(php -r 'copy("https://composer.github.io/installer.sig",
"php://stdout");')"
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
ACTUAL_CHECKSUM="$(php -r "echo hash_file('sha384', 'composer-setup.php');")"

if [ "$EXPECTED_CHECKSUM" != "$ACTUAL_CHECKSUM" ]
then
    >&2 echo 'ERROR: Invalid composer installer checksum'
    rm composer-setup.php
    exit 1
```

```
fi

php composer-setup.php --quiet
rm composer-setup.php

# Move composer to global installation
mv composer.phar /usr/local/bin/composer
```

4. Download BookStack PHP webapp files.

```
cd /var/www
git clone https://github.com/BookStackApp/BookStack.git --branch release --single-branch bookstack
cd bookstack
```

5. Configure BookStack.

```
# Install BookStack composer dependencies
export COMPOSER_ALLOW_SUPERUSER=1
php /usr/local/bin/composer install --no-dev --no-plugins

# Copy and update BookStack environment variables
cp .env.example .env
sed -i.bak "s@APP_URL=.*\$@APP_URL=http://$DOMAIN@" .env
sed -i.bak 's/DB_DATABASE=.*$/DB_DATABASE=bookstack/' .env
sed -i.bak 's/DB_USERNAME=.*$/DB_USERNAME=bookstack/' .env
sed -i.bak "s/DB_PASSWORD=.*\$/DB_PASSWORD=$DB_PASS/" .env

# Generate the application key
php artisan key:generate --no-interaction --force
# Migrate the databases
php artisan migrate --no-interaction --force

# Set file and folder permissions
chown www-data:www-data -R bootstrap/cache public/uploads storage && chmod -R 755 bootstrap/cache public/uploads storage
```

6. Configure Apache to serve new BookStack app.

```
# Set up apache
a2enmod rewrite
a2enmod php7.4
```

```
# Create apache config file to serve webapp
vim /etc/apache2/sites-available/bookstack.conf
    <VirtualHost *:80>
            ServerName 10.10.10.100           # Set to static IP otherwise define
hostname in DNS entry for it to work.
            ServerAdmin webmaster@localhost
            DocumentRoot /var/www/bookstack/public/
        <Directory /var/www/bookstack/public/>
            Options Indexes FollowSymLinks
            AllowOverride None
            Require all granted
            <IfModule mod_rewrite.c>
                <IfModule mod_negotiation.c>
                    Options -MultiViews -Indexes
                </IfModule>
                RewriteEngine On
                # Handle Authorization Header
                RewriteCond %{HTTP:Authorization} .
                RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
                # Redirect Trailing Slashes If Not A Folder...
                RewriteCond %{REQUEST_FILENAME} !-d
                RewriteCond %{REQUEST_URI} (.+)/$
                RewriteRule ^ %1 [L,R=301]
                # Handle Front Controller...
                RewriteCond %{REQUEST_FILENAME} !-d
                RewriteCond %{REQUEST_FILENAME} !-f
                RewriteRule ^ index.php [L]
            </IfModule>
        </Directory>
            ErrorLog ${APACHE_LOG_DIR}/error.log
            CustomLog ${APACHE_LOG_DIR}/access.log combined
    </VirtualHost>
```

7. Verify config file and enable it

```
apachectl configtest
a2dissite 000-default.conf             # Disable default website
a2ensite bookstack.conf
systemctl restart apache2
```

8. Login into Bookstack: Browser > $BOOKSTACK_IP > Login('admin@admin.com', 'password')

BookStack on Docker

Base SQL

PostgreSQL

Replication

Check from master

select * from pg_stat_replication;

Check from slave

select * from pg_stat_wal_receiver;

CentOS 7

PostgreSQL, also known as Postgres, is a free and open-source relational database management system emphasizing extensibility and SQL compliance.

Installation

source : https://tecadmin.net/install-postgresql-11-on-centos/

sudo yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm

yum install postgresql${version}-server

/usr/pgsql-${version}/bin/postgresql-${version}-setup initdb

systemctl enable postgresql-${version}.service systemctl start postgresql-${version}.service

postgres=# \password postgres

Upgrade

https://www.postgresql.org/docs/11/upgrading.html

Minor version

1. Stopper le service postgreSQL

```
$ systemctl stop postgresql-11.service
```

2. Faire un dump global

```
$ pg_dumpall > db.out
```

3. Upgrader la version de postgreSQL

```
yum install postgresql11-${MINOR_VERSION}
yum install postgresql11-server-11.11-1PGDG.rhel7

Show available minor version
yum list postgresql11 --showduplicates
```

4. Initier la base de données

```
/usr/pgsql-${version}/bin/postgresql-${version}-setup initdb
```

5. Relancer le service

```
$ systemctl start postgresql-11.service
```

Effective MEP

```
service postgresql-11 stop
cp -fra /mnt/data1/pgsql /mnt/data1/pgsql-old-20220510
yum remove postgresql11
yum install postgresql11-11.11-1PGDG.rhel7 postgresql11-server-11.11-1PGDG.rhel7 postgresql11-
contrib-11.11-1PGDG.rhel7
/usr/pgsql-11/bin/postgresql-11-setup initdb

service postgresql-11 start
```

Major version (WIP)

MariaDB

Install

https://computingforgeeks.com/how-to-install-mariadb-on-centos-rhel-7/

Binlog

https://smarttechways.com/2021/05/27/enable-disable-the-binary-log-in-mysql/

Check bin log is enabled (ON) or disabled (OFF)

```
mysql> show variables like 'log_bin';
```

Show the list of binary file created.

```
mysql> SHOW BINARY LOGS;
```

Enable the Binary Log in MySQL

```
--For enable, edit the my.ini file and enter following parameter
log-bin="RAC1-bin"
```

Disable the Binary log in MySQL

```
-- For disable the binary log, Edit the my.ini file
#log-bin="RAC1-bin"  --Comment the log-bin parameter
skip-log-bin   -- enter this in my.ini file and save and restart the mysql
```

MySQL

MySQL 8.0

- https://www.mysqltutorial.org/install-mysql-centos/
- https://www.mariuszantonik.com/2022/03/the-gpg-keys-listed-for-the-mysql-8-0-community-server-repository-are-already-installed-but-they-are-not-correct/

FTP

ProFTPd

mode SFTP

```
LoadModule mod_sftp.c
LoadModule mod_sftp_pam.c

<IfModule mod_sftp.c>
# On écoute sur le port 2222
Port                2222
# On active le mode SFTP
SFTPEngine          on

# Emplacement des clés de cryptage
SFTPHostKey         /etc/ssh/ssh_host_rsa_key2
SFTPHostKey         /etc/ssh/ssh_host_dsa_key2

# Pour le moment, on met authentification par mot de passé
# On changera après par authentification par clé
SFTPAuthMethods     password

# L’utilisateur est bloqué au repertoire qu’on lui a attribué
lors de sa création
DefaultRoot         ~

   # On ajoute les fichiers log pour le sftp
SFTPLog         /etc/sftp/log/sftp.log
TransferLog         /etc/sftp/log/sftp-transfer.log
</IfModule>
```

Base NoSQL

Gitlab

Gitlab

CI/CD

BASH

Require :

- https://github.com/mvdan/sh
- https://www.shellcheck.net/

.gitlab-ci.yml :

```
stages:
  - shell-code-check
  - shell-code-formater

shellcheck:
  image: koalaman/shellcheck-alpine:stable
  stage: shell-code-check
  before_script:
    - shellcheck --version
    - apk update
    - apk add git
  script:
    - git ls-files --exclude='*.sh' --ignored -c -z | xargs -0r shellcheck

shfmt:
  dependencies:
    - shellcheck
  image: mvdan/shfmt:latest-alpine
  stage: shell-code-formater
  before_script:
    - shfmt -version
    - apk update
    - apk add git
  script:
```

```
    - git ls-files --exclude='*.sh' --ignored -c -z | xargs -0r shfmt -i 2 -ci -d
```

Gitlab

CLI git

git fetch upstream/master ; git rebase -i upstream/master

OpenSSH

SFTP over SSH

DNS

Bind9 | Named

Cache

https://linuxconfig.org/how-to-view-and-clear-bind-dns-server-s-cache-on-linux

Power DNS

Mailserver

Mailserver

Postfix

https://electrictoolbox.com/postfix-email-size-limit/ postconf | grep message_size_limit

Docker

Docker

Docker

Delete all containers,networks,images don't use

`docker system prune`

Envoie/export de fichier dans un conteneur

• Local system to container : `docker cp myfile.txt mycontainer:/path/to/destination`

• Container to local system : `docker cp mycontainer:/path/to/source myfile.txt`

Check logs

• On service : `docker service logs $ContainerName$`

• On container : `docker logs -f $ContainerName$`

Gestion données du conteneur

Sauvegarder le conteneur

```
docker commit mon-conteneur backup/mon-conteneur
docker run -it backup/mon-conteneur
```

Exporter une image

```
docker save -o mon-image.tar backup/mon-conteneur
```

Importer une image

```
docker import mon-image.tar backup/mon-conteneur
```

Se connecter à un conteneur

```
docker exec
docker attach
```

Détruire un conteneur

```
docker kill (SIGKILL)
docker stop (SIGTERM puis SIGKILL)
docker rm (détruit complètement)
```

WAF

mod security

La commande suivante permet de générer les instructions SecRuleUpdateTargetById à partir des logs d'un serveur.

```
cat /var/log/messages | grep ModSec | perl -pe 's/.*at\s([^\s\.]+)\s*.*\[id "([0-9]+)"\].*/SecRuleUpdateTargetById $2 !$1/' | sort | uniq
```

```
cat /var/log/messages | grep ModSec | perl -pe 's/.*at\s([^\s\.]+)\s*.*\[id "([0-9]+)"\].*\[uri "([^\s\.]+)"\].*/<Location $3>SecRuleRemoveById $2<\/Location>/' | sort | uniq
```

Restic Backup

Restic : Déploiement

Docker Restic REST with SSL

Pré-requis :

- Posséder un binaire restic : `apt install restic`
- Posséder la suite openssl : `apt install openssl`
- Posséder docker et docker compose : `apt install docker docker-compose`
- Savoir lire :)

1) Préparation du répertoire projet

Variable de la documentation :

```
NAME_PROJECT=projet
PATH_PROJECT=/mnt/"${NAME_PROJECT}"
FOLDER_DATA_NAME=data
PATH_DATA="${PATH_PROJECT}"/"${FOLDER_DATA_NAME}"
PORT_EXPOSED=8009
USER_PROJECT=projet_user
SRV_RESTIC=127.0.0.1 #TEST EN LOCAL
```

Créer votre répertoire de projet :

```
mkdir /mnt/projet && cd $_
```

Créer le répertoire de sauvegarde dans le projet =

```
mkdir /mnt/projet/data
```

2) Générer un utilisateut restic

Créer un utilisateur pouvant intéragir avec le répertoire de sauvegarde :

```
htpasswd -B -c ${PATH_DATA}/.htpasswd projet_user
```

Un prompt apparaîtra afin de valider votre mot de passe.

3) Générer une paire de clés SSL

Créer un bundle de clés SSL ou importer les vôtre sous le format :

- public_key
- private_key

Méthode 1 : Let's Encrypt

Il est préférable d'utiliser une paire de clés valide via let's encrypt par exemple :

```
certbot certonly --standalone -d ${DOMAINE}
```

Il faudra suite intégrer les clés dans votre espace data :

```
cp -fra /etc/letsencrypt/archive/${DOMAINE}/fullchain1.pem ${PATH_DATA}/public_key
cp -fra /etc/letsencrypt/archive/${DOMAINE}/privkey1.pem ${PATH_DATA}/private_key
```

Méthode 2 : Snakeoil

Voici une manière manuelle de les générer :

```
openssl req -newkey rsa:4096  -x509  -sha512  -days 365 -nodes -out ${PATH_DATA}/public_key -
keyout ${PATH_DATA}/private_key
```

Cette méthode vous obligera à indiquer à restic de ne pas prendre en compte la validité du certificat via l'option :

```
restic --insecure-tls
```

4) Préparer votre docker-compose.yml

Créer votre fichier docker-compose.yml en modifiant les valeurs entre "{{ }}" : `vim ${PATH_PROJECT}/docker-compose.yml`

```
  services:
  rest_server_{{ NAME_PROJECT }}:
    command:
      - "/entrypoint.sh"
    container_name: "rest_server_{{ NAME_PROJECT }}"
    environment:
      - "OPTIONS=--tls --prometheus"
      - "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
      - "DATA_DIRECTORY=/data"
      - "PASSWORD_FILE=/data/.htpasswd"
    hostname: "rest_server_{{ NAME_PROJECT }}"
    image: "restic/rest-server"
    ipc: "private"
    labels:
      org.opencontainers.image.source: "https://github.com/restic/rest-server"
      org.opencontainers.image.title: "rest-server"
    logging:
      driver: "json-file"
      options: {}
    ports:
      - "{{ PORT_EXPOSED }}:8000/tcp"
    volumes:
      - "{{ PATH_DATA }}:/data"
version: "3.6"
```

Lancer ensuite votre instance restic

```
cd "${PATH_PROJECT}"
docker-compose up -d
```

5) Initialiser le workspace de sauvegarde

Init : initialiser un répertoire de sauvegarde

```
restic -r rest:https://${USER_PROJECT}:{{ MOT DE PASSE ETAPE 3
}}@${SRV_RESTIC}:${PORT_EXPOSED} init
```

Votre dépôt comporte désormais un répertoire de sauvegarde.

6) Bashrc && alias

Sur votre utilisateur courant (différent de root), éditer votre fichier bashrc `vim /home/user/.bashrc`

```
export RESTIC_PASSWORD="${USER_MDP}"
export
RESTIC_REPOSITORY="rest:https://${USER_PROJECT}:${USER_MDP}@${SRV_RESTIC}:${PORT_EXPOSED}/"
```

Se reconnecter sur sa session ou charger le fichier bashrc :

```
source /home/user/.bashrc
```

Cela permet ensuite d'utiliser le binaire vers votre dépôt de sauvegarde sans à divulguer toutes les informations :

```
zayad@vmi2007645:~$ restic stats
repository 6d0ed1e0 opened (repository version 2) successfully, password is correct
scanning...
Stats in restore-size mode:
Snapshots processed:   2
   Total File Count:   281
         Total Size:   24.191 KiB
```

Restic : Ligne de commandes

Bashrc & Alias

Sur votre utilisateur courant (différent de root), éditer votre fichier bashrc `vim /home/user/.bashrc`

```
alias restic='restic --insecure-tls'
export RESTIC_PASSWORD="${USER_MDP}"
export
RESTIC_REPOSITORY="rest:https://${USER_PROJECT}:${USER_MDP}@${SRV_RESTIC}:${PORT_EXPOSED}/"
```

Cela permet ensuite d'utiliser le binaire vers votre dépôt de sauvegarde sans à divulguer toutes les informations :

```
zayad@vmi2007645:~$ restic stats
repository 6d0ed1e0 opened (repository version 2) successfully, password is correct
scanning...
Stats in restore-size mode:
Snapshots processed:   2
   Total File Count:   281
         Total Size:   24.191 KiB
```

Backing Up

```
restic backup ~/directory-to-backup --pack-size=60
```

Cleanup

With every backup, Restic is creating a new snapshot with contents of a directory at the moment. To remove old and unused snapshots we need to execute the forget command:

```
restic forget --keep-last 2 --prune
```

Check

If you want to verify the consistency of your backup, run the check command:

```
restic check
```

Restore

To restore the latest snapshot of your backup:

```
restic restore latest --target ~/restore
```

Restic : Annexes

Logiciels clients

Testé et approuvé :

- Restic-browser https://github.com/emuell/restic-browser

Restic

- Restatic : https://github.com/Mebus/restatic
- autorestic : https://github.com/cupcakearmy/autorestic
- crestic : https://github.com/nils-werner/crestic
- Runrestic : https://github.com/sinnwerkstatt/runrestic
- restic-windows-backup : https://github.com/kmwoley/restic-windows-backup

Restic : Scénarios de sauvegarde

Sauvegarde MariaDB

```
#!/bin/bash
DATE=$(date +%Y%m%d-%Hh%M)
DUMP="/mnt/data/dump"
## Récupérer les bdd présent dans le SGBDR
listBdd=$(echo "SHOW DATABASES" |mysql |grep -v "Database" |grep -v "schema")

##Backup bases de données
for bdd in ${listBdd};
do
    echo "Dump des bases de données avant sauvegarde"
    echo "Création du répertoire temporaire pour les dump"
    mkdir -p ${DUMP}
    echo "dump de la base $bdd "
    echo ""
    mysqldump $bdd | gzip -9 > ${DUMP}/${bdd}.sql.tar.gz && bddName=${bdd}.sql.tar.gz
    echo "Lancement de la sauvegarde pour la BDD ${bdd}"
    restic backup --tag="bdd-$bdd-${DATE}" ${DUMP}/${bddName} --quiet
    if [ $? == 0 ]
    then
        echo "La sauvegarde de la bdd ${bdd} est effective"
    else
        echo "Echec de la sauvegarde"
        let "ERROR+=1"
    fi
done
```

Sauvegarde Conf & Logs

```
#!/bin/bash
DATE=$(date +%Y%m%d-%Hh%M)
CONF="/etc/apache2 /etc/mysql /etc/ssh /root /etc/letsencrypt /etc/fail2ban /etc/php
/etc/redis /etc/memcached.conf"
SITE="/var/www/html/"
LOG="/var/log"

## Récupérer les bdd présent dans le SGBDR
listBdd=$(echo "SHOW DATABASES" |mysql |grep -v "Database" |grep -v "schema")

##Backup conf & logs
for var in ${CONF} ${SITE} ${LOG}
do
    echo "Lancement de la sauvegarde pour les données du répertoire ${var}"
    DATE=$(date +%Y%m%d-%Hh%M)
    restic backup --tag="$var-${DATE}" $var --quiet
    if [ $? == 0 ]
    then
        echo "La sauvegarde du répertoire ${var} est effective"
    else
        echo "Echec de la sauvegarde"
        let "ERROR+=1"
    fi
done
```

Kubernetes

TP : Déploiement K8s 3 noeuds avec NFS

Source :

- https://akylson.com/kubernetes-cluster-on-debian-12-34d7d0ef5e92
- https://computingforgeeks.com/configure-nfs-as-kubernetes-persistent-volume-storage/

Lien pour avoir des ressources K8s à gogo :

- https://killercoda.com/playgrounds/scenario/kubernetes
- https://labs.play-with-k8s.com/

Périmètre de test

- Workers K8s inet 192.168.1.184/24 brd 192.168.1.255 scope global dynamic enp0s3
- Workers K8s inet 192.168.1.84/24 brd 192.168.1.255 scope global dynamic enp0s3
- Workers K8s inet 192.168.1.175/24 brd 192.168.1.255 scope global dynamic enp0s3
- 1 Control Plane K8s : inet 192.168.1.27/24 brd 192.168.1.255 scope global dynamic enp0s3
- 1 serveurs NFS : inet 192.168.1.82/24 brd 192.168.1.255 scope global dynamic enp0s3

Mise en place d'un cluster k8s 3 noeuds

- https://akylson.com/kubernetes-cluster-on-debian-12-34d7d0ef5e92

```
root@ctrl:~# apt -y install containerd iptables apt-transport-https gnupg2 curl sudo
root@ctrl:~# cat > /etc/sysctl.d/99-k8s-cri.conf <<EOF
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-ip6tables=1
net.ipv4.ip_forward=1
EOF
root@ctrl:~# sysctl --system
root@ctrl:~# modprobe overlay; modprobe br_netfilter
root@ctrl:~# echo -e overlay\\nbr_netfilter > /etc/modules-load.d/k8s.conf
# needs [iptables-legacy] for iptables backend
# if nftables is enabled, change to [iptables-legacy]
```

```
root@ctrl:~# update-alternatives --config iptables
There are 2 choices for the alternative iptables (providing /usr/sbin/iptables).

  Selection    Path                       Priority   Status
------------------------------------------------------------
* 0            /usr/sbin/iptables-nft      20        auto mode
  1            /usr/sbin/iptables-legacy   10        manual mode
  2            /usr/sbin/iptables-nft      20        manual mode

Press <enter> to keep the current choice[*], or type selection number: 1
update-alternatives: using /usr/sbin/iptables-legacy to provide /usr/sbin/iptables (iptables)
in manual mode

# disable swap
root@ctrl:~# swapoff -a
root@ctrl:~# vi /etc/fstab
# comment out
#/dev/mapper/debian--vg-swap_1 none            swap    sw              0       0

# switch to Cgroup v1 (v2 is the default)
root@ctrl:~# vi /etc/default/grub
# line 10 : add
GRUB_CMDLINE_LINUX="systemd.unified_cgroup_hierarchy=0"
root@ctrl:~# update-grub
root@ctrl:~# reboot
```

COngifurer le serveur NFS

- https://computingforgeeks.com/configure-nfs-as-kubernetes-persistent-volume-storage/

Configurer le pool de stockage

```
 apt install lvm2

 fdisk /dev/sdb
 pvcreate /dev/sdb1
 vgcreate data /dev/sdb1
 lvcreate -n k8s_data -L 50G data
 mkfs.ext4 /dev/data/k8s_data
 mkdir -p /data/k8s
/etc/fstab :
```

```
/dev/data/k8s_data   /data/k8s ext4 defaults 0 0

mount /data/k8s/
root@k8snfs:~# lsblk
sdb                 8:16   0   50G  0 disk
└─sdb1              8:17   0   50G  0 part
  └─data-k8s_data 254:0    0   45G  0 lvm  /data/k8s
```

Configure NFS

```
apt update
apt install nfs-kernel-server

/etc/idmapd.conf :
[General]
Domain = zayad.k8s

/etc/exports :
/data/k8s/ 192.168.1.0/24(rw,no_root_squash)



systemctl restart nfs-server
systemctl daemon-reload
```

Installer helm : https://helm.sh/docs/intro/install/

Troubleshooting

Médicat

URL project : https://medicatusb.com/

Source de la documentation : https://lecrabeinfo.net/tutoriels/reinitialiser-mot-de-passe-windows-perdu-ou-oublie-avec-cle-usb-medicat/

> Pré-requis: Avoir une clé USB d'au moins 32 Go (ou du moins une partition)

> Pré-requis: Désactiver son anti-virus

Installation

Installer le script sur l'environnement souhaité (il gère automatiquement la clé bootable ainsi que l'intégration de l'ISO)

Prévoir plus de 22Go d'archive sur la clé USB qui tournera sous ventoy : https://www.ventoy.net/en/index.html

- https://url.medicatusb.com/installerwindows
- https://url.medicatusb.com/installerlinux

Premier Pas

Modifier la langue et le clavier dans médicat

1. Modifier la disposition du clavier pour le français en utilisant le raccourci clavier CTRL + TAB
2. Une icône drapeau français devrait aprendre la place dans la zone de notification

Password Removal Tools

Il existe 3 méthodes pour supprimer un mot de passe

PasswordChanger (PCUnlocker)

1. Ouvrez PCUnlocker Entreprise :
   - sur le Bureau, double-cliquez sur **Reset User Passwords**.
   - dans les PortableApps, sélectionnez **Reset User Account Passwords**.
2. Sélectionnez le compte utilisateur avec le mot de passe à réinitialiser puis cliquez sur **Reset Password**.
3. Confirmez la réinitialisation du mot de passe du compte utilisateur en cliquant sur **Yes**.
4. Voilà, le mot de passe de l'utilisateur a bien été effacé !
5. Redémarrez votre ordinateur puis connectez-vous avec le compte utilisateur. N'oubliez pas de définir un nouveau mot de passe !

Reset User Account Passwords (Active@ Password Changer Ultimate)

1. Ouvrez Active@ Password Changer Ultimate en sélectionnant **PasswordChanger** dans les PortableApps. Sur le premier écran, cliquez sur **Next**.
2. Sélectionnez **Search all volumes for Microsoft Security Accounts Manager Database (SAM)** puis cliquez sur **Next**. Si vous souhaitez sélectionner manuellement la partition où se trouve Windows, sélectionnez une des deux autres options.
3. Sélectionnez le fichier SAM de votre installation de Windows puis cliquez sur **Next**.
4. Sélectionnez le compte utilisateur avec le mot de passe à réinitialiser puis cliquez sur **Next**.
5. Vérifiez que la case **Clear this User's Password** est bien cochée puis cliquez sur **Next**.
6. Confirmez la réinitialisation du mot de passe en cliquant sur **Yes**.
7. Voilà, le mot de passe de l'utilisateur a bien été effacé !
8. Redémarrez votre ordinateur puis connectez-vous avec le compte utilisateur. N'oubliez pas de définir un nouveau mot de passe !
9.

Windows Password Recovery Pro

1. Ouvrez Windows Password Recovery Pro en sélectionnant **Windows Password Recovery** dans les PortableApps.
2. Sélectionnez l'installation de Windows, le compte utilisateur avec le mot de passe à réinitialiser puis cliquez sur **Reset Password**.
3. Cliquez sur **Yes**. Voilà, le mot de passe de l'utilisateur a bien été effacé !
4. Redémarrez votre ordinateur puis connectez-vous avec le compte utilisateur. N'oubliez pas de définir un nouveau mot de passe !