SYSTÈME DE SURVEILLANCE

Procédure d’installation

version PUSH

Table des matières

1 Installation du serveur central

Les différentes documentations en ligne sont données à titre indicatif. Il est très important de les lire avant de procéder aux différentes installations des logiciels. Chacune doit être interprétée et adaptée aux manipulations.
Par exemple sous Debian la commande sudo n'existe pas, il faut utiliser la commande « su ». Sous raspberry cette commande existe, et pour se mettre en root, on devra utiliser la commande « sudo su ».

1.1 Installer le système Debian Stretch 9 ou Buster 10

Insérer le dvd, suivre les indications.
Lors du partitionnement il faut respecter les données suivantes :

- 6 Go pour la partition racine,
- une partition swap de la taille de la mémoire RAM (la taille de la RAM peut se lire en ouvrant la machine : c'est la somme de chaque barrette de RAM pluguée sur la carte mère ; on peut aussi quelquefois accéder à la RAM via le BIOS),
- 16 Go pour la partition /home,
- le reste du disque pour la partition /var.

Choisir l'interface utilisateur xfce4.

Pour le ccf, choisir comme mot de passe pour root le mot « root » et créer un utilisateur « surveillance », mot de passe « surveillance ».

Mettre le système à jour (*apt-get update && apt-get upgrade*).

1.2 Installer le serveur lamp

Se connecter au shell en root.

Installation de la base de données :
Documentation :
https://www.geek17.com/fr/content/debian-9-stretch-installer-et-configurer-mariadb-65
vérification : on doit pouvoir se connecter en console au serveur de base de données :
mysql -u root -p

Installation du serveur http « Apache » et des logiciels php et phpmyadmin :
Documentation :
https://docs.ovh.com/fr/dedicated/installer-lamp-debian-ubuntu/

Vérification de l'installation d'apache : on doit voir la page d'accueil d'apache via un navigateur (http://{adr ip serveur})

Vérification de l'installation de php :
créer une page phpinfo.php :
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
On doit pouvoir visualiser la page dans un navigateur (htt://{adresse ip serveur}/phpinfo.php). La page est violette et grise et présente l'environnement du serveur, notamment mysql.

Pour phpmyadmin, il faut autoriser la connexion en root :
Documentation :
https://www.citizenz.info/mariadb-mysql-connexion-root-avec-phpmyadmin-sous-ubuntu-16-04

https://www.debian-fr.org/t/installation-de-mariadb-phpmyadmin-compte-admin/74012/5

vérification : via un navigateur on peut se connecter en root sur http://{adr ip serveur}/phpmyadmin

1.3 Installer la base de donnée du système

Ouvrir phpmyadmin via un navigateur et créer (onglet « comptes d'utilisateurs ») un nouvel utilisateur nommé « surveillance » avec un mot de passe (pour le ccf choisir le mot de passe « surveillance », par la suite, ce mot de passe pourra être changé dans phpmyadmin ET en modifiant le fichier « *base.php* »).
Cocher « créer une base de données portant son nom et donner à cet utilisateur tous les privilèges sur cette base ».
Il est important de recharger les privilèges. Cela se fait dans l'onglet SQL avec la commande suivante :
« flush PRIVILEGES »
Vérification : on doit pouvoir se connecter depuis une console avec la commande :
mysql -h localhost -u surveillance -p

Pour installer la base, dans phpmyadmin, faire un copier-coller du fichier surveillance-structure_data.sql dans l'onglet SQL après s'être connecté à la base surveillance.

Pour la sécurité : enlever les droits de connexion en root sur phpmyadmin.

vérification :
la connexion en root interdite sur http://{adr ip serveur}/phpmyadmin
la connexion avec l'utilisateur surveillance doit être autorisée avec un navigateur avec la requête suivante :
http://{adr ip serveur}/phpmyadmin

Autoriser les connexions distantes sur la base de données :
Documentation :
https://www.adsysteme.com/lacces-a-distance-aux-bases-de-donnees-mysql-mariadb/
La procédure est assez bien expliquée.

Vérification : on doit pouvoir se connecter depuis le raspberry depuis une console avec la commande :
mysql -h {adresse ip du serveur central} -u surveillance -p

1.4 Installer les scripts php

Créer le dossier /var/www/html/surveillance et y transférer scripts php
Créer le dossier /var/www/html/surveillance/videos
Modifier le propriétaire de /home/www/html/surveillance en www-data
Vérifier droits en écriture de /home/www/html/surveillance/videos
Vérification : *ls -l /home/www/html/surveillance/*

```
-rwxr-x--- 1 www-data www-data  796 févr.  5 10:12 base.php
-rwxr-x--- 1 www-data www-data  171 févr. 16 08:00 connexion.html
-rwxr-x--- 1 www-data www-data  173 févr. 16 09:29 deconnexion.html
-rwxr-x--- 1 www-data www-data 8251 mars   5 10:54 index.php
-rwxr-x--- 1 www-data www-data   12 févr.  9 00:36 session.php
-rwxr-x--- 1 www-data www-data 1842 févr.  9 15:36 style.css
drwxr-x--- 2 www-data www-data 4096 mars   5 07:42 videos
```

À ce stade l'interface graphique doit fonctionner :

Vérification : navigateur http://{adr ip serveur}/surveillance
Pour la session administrateur :

- essayer de se loguer (le mot de pass de l'administrateur est « root »).
- si un problème survient, enlever le mot passe dans la table Administration sous phpmyadmin
- depuis l'interface graphique se loguer avec un mot de passe vide puis changer le mot de passe.

2 Envoie des mails d'alerte et configuration ssh

Les différentes documentations en ligne sont données à titre indicatif. Il est très important de les lire avant de procéder aux différentes installations des logiciels. Chacune doit être interprétée et adaptée aux manipulations.
Par exemple sous Debian la commande sudo n'existe pas, il faut utiliser la commande « su ». Sous raspberry cette commande existe, et pour se mettre en root, on devra utiliser la commande « sudo su ».

2.1 Créer un compte dédié à l'usage du système

Aller sur google et créer un compte gmail.
vérification : se connecter avec le nouveau compte sur gmail.
Attention : il faut autoriser les connexion moins sécurisée.

Documentation :
http://tutoriels.domotique-store.fr/content/109/272/fr/configurer-l_envoi-de-captures-d_ecran-par-mail-avec-une-camera-ip-wanscam.html

2.2 Configurer postfix

Documentation :
https://www.it-connect.fr/configurer-postfix-pour-envoyer-des-mails-avec-gmail/
https://www.ionos.fr/assistance/email/autres-programmes-de-courrier-electronique/configurer-postfix-linux/
https://xael.org/2015/raspberry-pi-postfix.html
https://www.milbako.com/raspberry-pi-serveur-mail-smtp-imap-dovecot-raspberry-pi/

Résumé :

- installer postfix mailutils libsasl2-2 ca-certificates libsasl2-modules
- copier une configuration minimale et la modifier dans /etc/postfix/main.cf
  modifier l'information relayhost par le smtp de gmail ([smtp.gmail.com]:587)
- définir le compte SMTP à utiliser : *nano /etc/postfix/sasl_passwd* ajouter :
  [smtp.gmail.com]:587 adresseMailGmail:motDePAsse
- protéger /etc/postfix/sasl_passwd (modification des droits à 400)
- mettre à jour postmap (*postmap cheminDuFichierSasl_passwd*)
- recharger la configuration de postfix (relancer le service)

Exemple de configuration minimale (/etc/postfix/main.cf) :

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version


# Debian specific:  Specifying a file name will cause the first
# line of that file to be used as the name.  The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname
```

```
smtpd_banner = $myhostname ESMTP $mail_name (Raspbian)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# See http://www.postfix.org/COMPATIBILITY_README.html -- default to 2 on
# fresh installs.
compatibility_level = 2

relayhost = $mydomain
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
smtp_use_tls = yes
```

vérification :
echo "Test mail depuis postfix" | mail -s "Test Postfix yeba" adresseMailAdministrateur -a "From: nouveaucompte@gmail.com"
où nouveaucompte@gmail.com représente le compte mail créé précédemment.

2.3 Installation et configuration des clients-serveur ssh

Mettre à jour le serveur et le raspberry : *apt-get update, apt-get upgrade*
Installer ssh :
sur le raspberry : *apt-get install openssh-client* (si l'étudiant responsable du raspberry ne l'a pas encore fait).
sur le serveur : *apt-get install openssh-server*
Configurer la connexion ssh sans mot de passe depuis le raspberry en tant que utilisateur pi sur le serveur en tant qu'utilisateur www-data :
Documentation :
https://tutox.fr/2017/04/08/se-connecter-ssh-de-passe/
http://www.planet-libre.org/index.php?post_id=4599

Sur le serveur :
Tout d'abord, en root, modifier le bash de l'utilisateur www-data :
nano /etc/passwd et modifier la ligne :

```
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
```

en

```
www-data:x:33:33:www-data:/var/www:/bin/bash
```

vérification : la commande su www-data doit amener au bash en tant qu'utilisateur www-data

Donner un mot de passe à l'utilisateur www-data (à faire en root) :

Modifier ensuite la config OpenSSH : dans /etc/ssh/sshd_config vérifier que la ligne suivante n'est pas commentées (modifier éventuellement).

AuthorizedKeysFile %h/.ssh/authorized_keys
AllowRootLogin no
(ou PermitRootLogin yes)

puis relancer le serveur ssh :

/etc/init.d/ssh reload

Créer un dossier *.ssh* dans le dossier de l'utilisateur www-data et un fichier *id_rsa.pub*

cd /var/www
mkdir .ssh

Modifier l'utilisateur et le groupe de ce dossier et de ce fichier en www-data
Vérification :

ls -al

```
drwxr-xr-x  2 www-data www-data 4096 oct.  20 13:13 .ssh
```

Sur le raspberry :

générer une clé publique en tant qu'utilisateur pi
su pi
ssh-keygen -t rsa
(ne rein remplir et valider plusieurs fois)

copier la clé publique sur le serveur et la rajouter dans les clés autorisées de l'utilisateur www-data :

scp -r -p ~/.ssh/id_rsa.pub www-data@{adr ip serveur}:/var/www/.ssh
ou
ssh-copy-id www-data@{adr ip serveur}

Enfin, sur le serveur, redémarrer le serveur ssh :

/etc/init.d/ssh reload

Vérification : la connexion depuis le raspberry en tant qu'utilisateur www-data doit se faire sans mot de passe.

sécuriser les fichiers :
documentation :
https://medium.com/@stadja/tout-sur-comment-se-connecter-%C3%A0-un-serveur-ssh-sans-mot-de-passe-b2cc5c0a86e1)
résumé :

sur le raspberry, en tant que pi
chmod 400 ~/.ssh/id_rsa
sur le serveur, en tant que www-data
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys

vérification : en tant qu'utilisateur pi depuis le raspberry, on doit pouvoir se connecter en tant qu'utilisateur www-data sur le serveur. La connexion ne doit pas nécessiter de mot de passe.

ssh www-data@{adr ip serveur}

3 Raspberry

3.1.1 Installer Raspian

Brancher écran, clavier et réseau et alimenter le raspberry. Normalement la carte micro est prête à

l'emploi pour installer le système raspian.

Une fois installé, mettre le système à jour :
sudo apt-get update
sudo apt-get upgrade

Paramétrer la configuration du raspberry :
sudo raspi-config
changer les options de localisation : choisir fr_FR.UTF-8 UTF-8 (dans « change Locale »)

Installer le serveur ssh :
sudo apt-get install openssh-server

Installer le client mysql :
sudo apt-get install default-mysql-client
et la librairie de développement :
sudo apt-get install libmariadbclient-dev

3.1.2 Installer la caméra

Documentation :
https://raspberry-lab.fr/Composants/Utilisation-Camera-sur-Raspberry-Pi-Francais/

Résumé :

- arrêter le raspberry (à faire hors tension) et brancher la caméra (attention à être extrêmement minutieux et appliqué pour cette opération car la nappe de la caméra est très fragile).
- configurer le raspberry en ligne de commande : sudo raspi-config \ interfacing options \ yes

vérification : *vcgencmd get_camera*
vérification : prendre une photo : *raspistill -o ~/image.jpg -q 100*
vérification : prendre une vidéo : *raspivid -o video.h264 -t 10000*

convertir le format h264 en mp4
installer MP4Box : *sudo apt-get install gpac*
vérification : *MP4Box -add video.h264 video.mp4* (le fichier video.mp4 doit exister).

3.1.3 Installation du capteur de présence infrarouge

Installation matérielle : (à faire hors tension)

connecter la masse du capteur à la masse du gpio du raspberry
connecter l'entrée du capteur à une pin libre du gpio (attention à bien choisir sa broche) :

Broche	GPIO
3	0 (rev.1) ou 2 (rev.2)
5	1 (rev.1) ou 3 (rev.2)
7	4
11	17
12	18
13	21 (rev.1) ou 27 (rev.2)
15	22
16	23
18	24
22	25

Puis alimenter le capteur (masse et alimentation 12 V).

Vérification : (suivre https://www.blaess.fr/christophe/2012/11/26/les-gpio-du-raspberry-pi/)
exemple avec les broche 23 et 24 :
#cd /sys/class/gpio/
/sys/class/gpio # echo 23 > export
/sys/class/gpio # echo 24 > export
/sys/class/gpio # ls
export gpio23 gpio24 gpiochip0 unexport
/sys/class/gpio # cd gpio23/
/sys/devices/virtual/gpio/gpio23 # cat direction
in
/sys/devices/virtual/gpio/gpio23 # cat value
0
Faire fonctionner le capteur...
/sys/devices/virtual/gpio/gpio23 # cat value
1

3.1.4 Installer les sources et compiler le programme

En tant qu'utilisateur pi, créer un dossier /home/pi/surveillance_push
cd ~
mkdir surveillance_push
Y transférer les fichiers sources via scp (ou par clé usb)
scp {fichiers} pi@{adr ip raspberry}:surveillance_push/

Compiler les exécutables :
rappel : il y a deux exécutables à compiler le programme « cronSurveillance » et le programme « surveillance ».

pour compiler le programme cronSurveillance :
Dans le dossier ~/*surveillance_push,* copier le fichier « Makefile_cron » en « Makefile » :
cp Makefile_cron Makefile
make
Le fichier cronSurveillance doit apparaître.

pour compiler le programme surveillance :
Dans le dossier ~/*surveillance_push,* copier le fichier « Makefile_surveillance » en « Makefile » :
cp Makefile_surveillance Makefile
make
Créer un sous dossier nommé « vidéo » où seront créer les vidéos :
mkdir video

modifier le fichier de configuration :
nano fichierConfig.conf
renseigner convenablement le fichier (base de données, identifiant de la caméra, numéro du gpio, adresse du serveur http, chemin des vidéos sur le serveur et chemin des vidéos sur le raspberry).

À ce stade l'acquisition des vidéo et le transfert doit fonctionner.
Vérification de l'acquisition : ./surveillance.sh
Des vidéos doivent apparaître dans le sous dossier « video » lorsque le capteur est sollicité.
Vérification du transfert : ./cron.sh
Les vidéos créées doivent être transférées sur le serveur. Les noms doivent apparaître dans la base de données et les vidéos transférées doivent être supprimées du raspberry.

3.1.5 Automatiser le lancement sous cron :

Documentation :
https://technique.arscenic.org/commandes-linux-de-base/article/cron-gestion-des-taches-planifiees
https://debian-facile.org/viewtopic.php?id=24125

Résumé (sous utilisateur pi) :

rendre les scripts exécutables :
chmod +x cron.sh
chmod +x surveillance.sh
paramétrer le cron :
crontab -e
puis ajouter les lignes suivantes :
* * * * * /home/pi/surveillance_push/cron.sh > /dev/null 2>&1
@reboot /home/pi/surveillance_push/surveillance.sh > /dev/null 2>&1

vérification : redémarrer et vérifier que le programme est lancé

sudo reboot

attendre que le raspberry se relance...

ps aux | *grep surveillance*

L'exécutable *surveillance* doit être lancé.
Par ailleurs on doit voir que les fichiers sont transférés automatiquement sur le serveur.

Si le système fonctionne en le lançant manuellement (*./surveillance fichierConfig.conf*) mais qu'il n'est pas lancer automatiquement par le cron au démarrage de la machine, c'est qu'il y a un problème au niveau des scripts de lancement *surveillance.sh* et *cron.sh*.
Soit ils contiennent des informations fausses, soit ils ne sont tout simplement pas exécutables.
Il peut aussi sagir d'une erreur de configuration dans le fichier *crontab (*celui qui s'affiche avec la commande *crontab -e)*