Mon projet de serveur auto-hébergé

Contexte

L'idée me trotte depuis longtemps, et après de nombreuses lectures diverses un peu partout sur le net, j'ai finalement bien envie de me lancer.

Je possède actuellement deux noms de domaines (un .fr et un .org), ainsi qu'un espace sur un serveur mutualisé, le tout chez Gandi.net. J'en suis au passage très satisfait.

Seulement voilà, la curiosité aidant, l'envie d'en apprendre davantage, le plaisir d'avoir le contrôle intégral de ses données et de sa présence sur la toile, j'ai enfin l'intention de me lancer.

Objectif

L'objectif sera plutôt modeste dans un premier temps : installer mon serveur et faire les premières configurations basiques, apprendre à le maintenir à jour, et m'en servir pour du partage de fichiers sur le réseau local (qui contient des machines sous Windows, Linux, Android...). Puis, agrandir le champ des possibles, en y installant un client type OwnCloud, un logiciel FTP, un serveur web, et quelques outils sympa (RSS..). Le serveur mail n'est pas prévu pour le moment, car c'est d'une part très sensible (hors de question de défaillances!), mais également il semblerait que ce soit d'un niveau plus avancé, donc chaque chose en son temps.

Cette page est destiné à garder une trace reproductible de mon installation afin de pouvoir être capable de refaire une installation complète si j'opte finalement pour l'achat d'une machine optimisé pour cette fonction (pour l'instant je vais faire des tests en recyclant un vieux portable) et éventuellement servir de soutien pour d'autres qui voudraient se lancer!

De nombreuses sources de renseignement

Je ne suis pas seul dans ce projet : plein d'autres avant moi se sont lancés dans les joies de l'hébergement “at home”, et je n'ai pas l'intention de réinventer la roue! D'ailleurs je n'ai que quelques fragiles bases de réseau, programmation, ligne de commande... Voici quelques bonnes lectures qui m'ont inspirées au fil de l'eau :

Généralités sur l'auto-hébergement

- Rendre l'auto-hébergement facile et sans douleur, par Stéphane Bortzmeyer
- La page "Avantages et inconvénients" du wiki de auto-hebergement.fr
- Une (vieille) conférence sur l'auto-hébergement : Minitel 2.0, par Benjamin Bayart
- La page Wikipédia sur l'auto-hébergement

Des tutoriels techniques intéressants

- Tutoriel pour installer un serveur chez soi (debian) par Xavier Cartron
- L'hébergement maison, par Mitsu
- Auto-hébergement.fr
- Blog-libre.org, et en particulier la série d'article server@home par Cascador
- Turn an Old Computer into a Networked Backup, Streaming, or Torrenting Machine with Ubuntu, par Whitson Gordon
- Un tutoriel sur ddclient et No-IP, par Ben

... et des gens sympas!

Quelques remerciements en vrac!

- A Jambon, Nicolas et Perdouille du site horyax.fr pour tous leurs conseils techniques!
- Aux blogueurs Mitsu (suumitsu.eu); Timo (lehollandaisvolant.net) et Seb (sebsauvage.net) pour les années de suivi assidus de leurs paroles qui valent de l'or!
- Aux nombreux bénévoles qui trainent sur IRC, sur le serveur freenode en particulier, et particulièrement les peuplades de #ubuntu ; #ubuntu-fr ; #dokuwiki

Matériel 1 : Vieux PC HP...

Mon serveur actuel va se baser sur mon vieil ordinateur portable que je dépoussière tout spécialement.

Il s'agit très exactement d'un *HP Pavilion dv8365ea*, dont voici les spécifications, vite fait:

Processeur : Intel Core Duo T2400 à 1,83 GHz avec technologie Centrino Duo, mémoire vive : ~~2 Go~~ 1 Go, disque dur : 120 Go Combiné lecteur DVD-ROM et graveur CD/DVD biformat double couche, écran 17“ TFT BrightView

Et voici les spécifications détaillées, d'après un site commerçant.

```
Type de produit :       Portable - Notebook
Écran :             17 "
Poids en kg :           3,7 Kg
Dimensions (l x p x h) en mm :  397 x 282 x 46
Résolution écran :      TFT BrightView
Processeur :            Intel Core Duo T2400
Vitesse du processeur :     1,83 GHz / 667 MHz
Mémoire cache externe :     2 Mo de niveau 2
Mémoire RAM :           <del>2 Go</del> 1 Go, une des barrettes étant
défaillante
RAM maxi :          2 Go
Capacité du disque dur :    120 Go
Lecteur / Graveur :     Combiné lecteur DVD-ROM et graveur CD-RW, DVD -R/-
```

```
RW, DVD +R/RW, DVD +R double couche
Lecteur de cartes mémoire : 6 en 1 (Memory Stick, Memory Stick Pro, Secure
Digital, Smart Media, MultiMedia Card, xD Picture Card)
Carte graphique :       NVIDIA GeForce Go 7600
Mémoire vidéo dédiée :      256 Mo
Audio :             16 bits intégrée
Carte réseau Ethernet :     10/100 Mbps
Communication sans fil :    WiFi 802.11a/b/g, Bluetooth
Connecteurs :           4 USB 2.0, 1 modem RJ-11, 1 Ethernet RJ-45, 1
FireWire IEEE1394, 1 VGA, 1 S-Vidéo, 1 sortie casque avec sortie S/PDIF, 1
entrée micro, 1 PC Card de type I ou II, 1 ExpressCard/54 (compatible
ExpressCard/34)
Dispositif de pointage :    TouchPad (pavé tactile)
Type de batterie :      Lithium ion / NC
Résolution maxi avec mémoire installée : 1440 x 900
Sortie vidéo TV :       S-Vidéo
Tuner TV TNT :          Oui, avec télécommande
Connecteurs FireWire / IEEE 1394 : 1
Nombre de haut-parleurs intégrés : 2
Modem / Fax 56K :       Oui
```

Et... action ! (Vieux PC)

Je pars du principe que le lecteur a suffisamment de bagage technique pour comprendre ce que j'ai fait, donc je détaillerais les éléments parfois de façon succincte.

J'utiliserai ici Ubuntu Server, version de base (avec...unity).

Récupération d'Ubuntu Server

Procédé tout à fait classique :

1. Récupération de l'image iso de la dernière version d'Ubuntu Server LTS, adapté à mon pc. J'ai récupéré pour ma part la version 14.04.1 LTS (32 bits), correspondant au fichier ubuntu-14.10-server-i386.iso. A noter que j'ai choisi d'utiliser BitTorrent pour cela car cela m'éviter d'avoir a vérifier l'intégrité du fichier final reçu, et préserve la bande passante du serveur d'Ubuntu-fr.
2. Gravure de l'iso sur un bon vieux DVD. J'utilise pour ma part l'excellent CDBurnerXP.

Installation d'Ubuntu Server

blablabla.....

Relier le serveur au monde extérieur

On commence tranquillement en installant ddclient:

```
apt-get install ddclient
```

La configuration démarre, répondre n'importe quoi puisque nous allons configurer à la main.

Se rendre sur no-ip.com et créer un compte gratuit. Bien noter le login, mot de passe et le domaine dynamique choisi.



Ne pas installer le 'Dynamic Update Client' qu'ils proposent.

Nous allons éditer le fichier de configuration de ddclient. Dans un terminal :

```
sudo nano /etc/ddclient.conf
```

Remplacer le contenu par :

```
use=web, web=checkip.dyndns.com/, web-skip='IP Address'

protocol=dyndns2
server=dynupdate.no-ip.com
login=charpy1
password=un_mot_de_passe_costaud
charpyserv.ddns.net
```

En remplaçant bien sûr le login et le mot de passe, ainsi que le nom de domaine dynamique généré lors de la création du compte.

Ensuite on enregistre puis on redémarre ddclient :

```
sudo service ddclient restart
```

On attend quelques minutes : ddclient va initier les mises à jour automatiques des DNS avec No-Ip. On peut vérifier facilement :

```
ping charpyserv.ddns.net
```

Résultat :

```
PING charpyserv.ddns.net (109.30.12.65) 56(84) bytes of data.
64 bytes from 65.12.30.109.rev.sfr.net (109.30.12.65): icmp_seq=1 ttl=64
time=1.98 ms
```

En cas de problème, rendez vous dans /var/log/syslog pour vérifier d'éventuels messages d'erreurs.

Le nom de domaine dynamique est maintenant relié au routeur. Il faut maintenant activer une translation de port (NAT) pour activer le serveur Http :

```
Nom : servHTTP (par exemple)
Protocole : TCP
```

```
Type : Port
Ports externes : 80
Adresse IP de destination : 192.168.1.8 (L'adresse ip locale du serveur)
Ports de destination : 80
```

Hum... Ca devrait être bon! Essayez d'atteindre votre site par le nom de domaine dynamique, pour voir!

Matériel 2 : Raspberry 3

Nouveau projet, un serveur complet sur Raspberry pour gérer un petit coin tranquille d'internet pour moi, ma famille, quelques sites d'amis proches, mais aussi les sites web php/html/css/js de mes élèves et quelques outils en support.

Setup complet Raspberry 3

- Récupération de Raspberry Pi Imager (v 1.7.1 lors de la rédaction de ce tuto). C'est l'outil officiel pour créer la carte sd d'un raspi.
- Utilisation de Raspberry Pi Imager pour installer la carte sd avec Rasperry OS 32bits (anciennement Raspian) en dernière version. Ne pas hésiter à aller d'abord dans les paramètres, pour cocher l'activation automatique du ssh, définir le mot de passe web si Wifi, et mettre un mot de passe admin !
- Création d'un fichier vide nommé “ssh” à la racine de la carte sd.
- Carte sd dans le RPI, boot.
- Récupération de l'adresse IP du RPI sur le réseau local (par exemple dans la config de la box, ou sur le raspi avec “sudo ifconfig”), de la forme 192.168.*.*, et éventuellement mise en place d'une ip statique sur le réseau local pour lui.
- Redirection des ports 80 et 443 sur le routeur pour diriger vers le raspi.
- Normalement le rpi est maintenant accessible à distance, mais il n'a pas encore de serveur web opérationnel.
- Connexion en ssh avec le RPI à partir de maintenant, pour effectuer des opérations.
- Mise à jour :

```
sudo apt update
sudo apt upgrade
sudo apt update
```

- Installation de Apache, le serveur web :

```
sudo apt install apache2
```

- Config des bons droits sur le dossier des sites web :

```
sudo chown -R pi:www-data /var/www/html/
sudo chmod -R 770 /var/www/html/
```

- Vérifier que le serveur Apache est opérationnel :

```
wget -O verif_apache.html http://127.0.0.1 & cat ./verif_apache.html
```

- Si vous voyez marqué à un endroit dans le code « It works! », c'est qu'Apache fonctionne.
- Le répertoire /var/www/html est le répertoire des fichiers du serveurs web, c'est là qu'il faudra mettre les fichiers qui doivent aller être online ! On peut donc déjà mettre un site en html, css et javascript, mais il manque encore PHP et MySQL.
- Installation de PHP et création d'un fichier index de test :

```
sudo apt install php php-mbstring
echo "<?php phpinfo(); ?>" > /var/www/html/index.php
```

- Allez ensuite voir le site, soit avec un navigateur sur http://127.0.0.1, soit en y faisant un wget comme précédemment (cherchez alors "PHP Version"). Si tout fonctionne, c'est bon !
- Installation de MySQL, pour que les sites web hébergés puissent stocker des informations

```
sudo apt install mariadb-server php-mysql
```

- Connexion à la base de donnée (bdd):

```
sudo mysql --user=root
```

- Il faut maintenant supprimer puis recréer l'utilisateur root, car celui par défaut n'est utilisable que par le compte administrateur du système, et pas par les script PHP du serveur. Pour cela, une fois connecté à MySQL, lancez les commandes suivantes (remplacez password par le mot de passe de votre choix) :

```
DROP USER 'root'@'localhost';
CREATE USER 'root'@'localhost' IDENTIFIED BY 'votremotdepasse';
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;
```

- C'est fini, tout est en place ! Pour une prochaine connexion, on pourra faire :

```
mysql --user=root --password=votremotdepasse
ou
mysql --user=root --password  (pour plus de confidentialité)
```

- Pour plus de praticité, on va installer phpmyadmin, une interface plus simple pour administrer les bases de données que du SQL dans la console.

```
sudo apt install phpmyadmin
```

- Choisir "no" pour l'utilisation de dbconfig-common, car on a déja configuré la bdd. Choisir PHPMyAdmin pour un serveur Apache. Pour le mot de passe root, il s'agit de celui que vous aviez utilisé pour MySQL.
- Activation du service mysqli :

```
sudo phpenmod mysqli
sudo /etc/init.d/apache2 restart
```

- Vérification du bon fonctionnement de phpmyadmin : Aller dans le navigateur à http://127.0.0.1/phpmyadmin

- Possibilité d'accéder au raspi depuis le net à l'adresse IP externe, ou de mettre en place une URL chez noip.

- Installation de NextCloud
- Quelques prérequis:

```
sudo apt install apache2 mariadb-server libapache2-mod-php
sudo apt install php-gd php-json php-mysql php-curl php-mbstring php-intl
php-imagick php-xml php-zip
```

- Puis l'installation en elle-même :

```
cd /var/www/html
sudo wget https://download.nextcloud.com/server/releases/latest.zip
sudo unzip latest.zip
```

- Et les bons droits pour que le serveurs puissent accéder aux fichiers et les gérer :

```
sudo chmod 750 nextcloud -R
sudo chown www-data:www-data nextcloud -R
```

- NextCloud est pratiquement prêt à être utilisé, mais il lui faut la base de donnée MySQL. Par sécurité, on crée un couple “utilisateur + base de donnée associée” spécialement.

```
sudo mysql
CREATE USER nextcloud IDENTIFIED BY 'un_gros_mot_de_passe';
CREATE DATABASE nextcloud;
GRANT ALL PRIVILEGES ON nextcloud.* TO 'nextcloud'@localhost IDENTIFIED BY
'un_gros_mot_de_passe';
FLUSH PRIVILEGES;
quit
```

- Ouvrir /nextcloud dans le navigateur pour terminer l'installation : définir un compte admin, donner les infos de la base de donnée créee à l'instant, et, à partir du tableau de bord, activation des applications nécessaires (Calendrier, Contact, Group folders, Talk, Galerie d'image, édition collaborative, etc...)
- Installation d'un serveur FTP

```
sudo apt install proftpd
```

- Config du serveur à prévoir :

```
sudo nano /etc/proftpd/proftpd.conf
```



https://raspberry-pi.fr/installer-serveur-ftp-raspberry-pi/

- Shell in a box



https://www.tecmint.com/shell-in-a-box-a-web-based-ssh-terminal-to-access-remote-linux-servers/



https://github.com/shellinabox/shellinabox

Nouveaux ajout :

charpenel.hd.free.fr sur votre IP fixe 78.229.96.117

ssh -p 222 -l pi 78.229.96.117

SHELL IN A BOX :

sudo apt install shellinabox

sudo apt install locate

voi les services qui tournent : service –status-all

détail d'un service : service shellinabox statut

Voir QUI ecoute sur quel port (sécurité!) : netstat -plunt Les ip ouvertes en 0.0.0.0:XXXX sont accessibles depuis toutes les IP de notre propre machine Les ip en 127.0.0.1:XXXX sont accessibles que depuis l'ip de la machine dans le sous-reseau actuel

Les ip ouvertes en :::XXXX sont accessibles depuis l'extérieur (danger, mais parfois nécessaire)

Mettre une appli (shell in a box) derrière un reverse proxy :

éditez etc/apache2/sites-available/000-default.conf

Pour chaque service, ajouter un “ProxyPass” et un “ProxyPassReverse”.

Ajouter les modules Apache de proxy :

sudo a2enmod proxy sudo a2enmod proxy-http sudo a2enmod proxy_balancer sudo a2enmod lbmethod_byrequests

sudo systemctl restart apache2.service

vérif que le fichier de config d'apache est activé: sudo a2ensite 000-default.conf

Il y a 2 Fichiers de config pour shell in box :

* Dans /etc/sheelinabox/options-available/ (peu utile, css du shell) * Dans /etc/default/shellinabox/ : ajout du –disable-ssl dans les arguments, puis relance du service.

Activer les themes css dans shell in a box : compliqué… :x

Installation du firewall sudo apt install ufw sudo ufw allow 22 comment 'ssh et sftp' #autoriser le port 22 sudo ufw status numbered # voir l'état des régles (et leur numéro, pour les numéroter)

sudo ufw allow from 192.168.0.1/24 port 4200 comment 'shellinabox' # pour autoriser tous les accès depuis le réseau local

sudo ufw allow 80 comment 'http' #http

Début d'année :

Ajout d'un groupe nsi global sudo addgroup nsi adduser pi nsi #s'ajouter soi même

groups #affiche tous les groupes existants (attention, se relogger pour rendre les ajouts dans les groupes effectifs !)

sudo apt install members members nsi #voir les membres d'un groupe

sudo adduser jstern –ingroup nsi #ajout d'un eleve en precisant le groupe principal (nsi) –> puis son mot de passe

test : creer un fichier en tant que cet eleve, vérif les perm

Jupyther Hub : basé sur
https://towardsdatascience.com/setup-your-home-jupyterhub-on-a-raspberry-pi-7ad32e20eed

Mise à jour de python 3, et pip

```
sudo pip3 install --upgrade pip
```

Installation npm :

```
sudo apt-get install npm
sudo npm install -g configurable-http-proxy
```

Installation en elle-même :

```
sudo -H pip3 install notebook jupyterhub
```

Génération du fichier de config:

```
jupyterhub --generate-config
sudo mv jupyterhub_config.py /root
```

Editer en sudo le fichier pour décommenter cette ligne et enregistrer : c.JupyterHub.bind_url = 'http://:8000'

Editer le fichier de /etc/apache2/sites-vaailable/000-default.conf, et ajouter les 2 lignes de “ProxyPass” et “ProxyPassReverse”.

Editer root/jupyterhub_config.py pour décommenter la “base url” et la mettre à '/jupyter/'

```
sudo ufw allow from 192.168.0.1/24 port 8000 comment 'jupyter'  # pour
autoriser tous les accès depuis le réseau local
```

Enregistrer jupyter comme un service, pour éviter d'(avoir à le relancer à chaque redémarrage du rpi :

Créer le fichier /lib/systemd/system/jupyterhub.service et le remplir :

```
[Unit]
Description=JupyterHub Service
After=multi-user.target

[Service]
User=root
ExecStart=/usr/local/bin/jupyterhub --config=/root/jupyterhub_config.py
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

A suivre : * faire les groupes de travail et le -u 0002 dans /etc/ssh/sshd_config * Docker

From:
http://www.charpenel.org/wiki/ - **Tutos en vrac**

Permanent link:
http://www.charpenel.org/wiki/doku.php?id=mon_projet_de_serveur_personnel&rev=1647967902

Last update: **2022/03/22 17:51**