# SI du LOV

## Sécurité

- ne jamais détenir un mot de passe tout seul
- tous les mots de passe doivent être centralisés: keepass
- ne jamais modifier un mot de passe sans le mettre à jour dans la base keepass du bureau
- ne jamais utiliser un mot de passe faible ou générique
- ne jamais communiquer des mots de passe en dehors du bureau sauf:
  - accord explicite du bureau
  - membre ou contributeur formé au service qu'il va administrer modifier
- ne pas déployer de nouveaux services ou systèmes sans les documenter ici

## SI du local

- Schéma du réseau, switch Netgear et VLAN 836 : Réseau du local
- Ouverture du garage et de la porte arrière (boîtier Arduino, badges) : Système d'ouverture du garage
- Serrure de la porte du local (Raspberry Pi gatepi, lecteur de badges, bouton garage) : Serrure de la porte

La Freebox a été retirée le 12/10/2026 : l'accès à internet passe directement par le serveur “atom”, qui sert de routeur sur le VLAN 836 de la fibre Free (détails : Réseau du local). Si l'atom est éteint, il n'y a plus d'internet au local.

Le DHCP et le DNS du local sont assurés par l'atom (dnsmasq).

**TODO**: Documenter le point d'accès wifi qui remplace celui de la freebox (modèle, emplacement, mot de passe dans le keepass) et faire un point d'accès wifi pour les visiteurs avec un mot de passe qui change régulièrement.

### Installation du serveur DHCP/DNS/PXE

Comme dis plus haut, nous utilisons notre propre serveur DHCP et DNS car les PC du LOV démarrent en PXE. C'est à dire que les PC démarrent à partir d'un système d'exploitation qui est envoyé depuis le serveur.

Le serveur local “atom” assure les fonctions suivantes:

- DNS et DHCP via dnsmasq
- boot PXE via dnsmasq et NFS
- serveur NFS pour les répertoires /home des différents PC du LOV et le boot diskless (i.e. le démarrage réseau).

L'accès à atom se fait par mot de passe ou clé SSH. Le mot de passe se trouve dans le keepass du bureau, la clé SSH est à demander au bureau.

Voici la procédure d'installation du serveur “atom”.

### Etape 1 : Installation du Serveur

Installation d'un débian.

Lors de l'installation, définir le mot de passe root tel que définit dans le keepass du bureau. Créer l'utilisateur lov avec le mot de passe tel que définit dans le keepass du bureau.

Lors du partitionnement, actuellement le serveur n'a qu'un seul disque dur sur lequel il a deux partition configuré ainsi :

1. 20Go formaté en ext4 avec comme point de montage /
2. Le reste est formaté en ext4 avec comme point de montage /srv

**TODO**: Monter /srv sur un autre disque dur qui sera en raid 1 pour assurer la sauvegarde des dossiers /home des membres de l'asso.

Une fois debian installé, il faut installer les paquets sudo et vim

Pour cela, dans un terminal, il faut se mettre en root en tapant la commande suivante:

```
su
```

Il vous sera demandé le mot de passe root tel que définit au moment de l'installation (ou dans le keepass du bureau). **Remarque**: Ne soyez pas surpris si vous ne voyez pas de caractère ou même d'étoile '*' quand vous taper le mot de passe, c'est une procédure de sécurité normal pour Linux pour cacher la longueur du mot de passe.

Une fois root, installé les paquets sudo et vim

```
apt install sudo vim
```

Vous devriez alors voir quelques qui ressemble à ça :

```
TODO: Mettre un exemple de sortie d'apt
```

Appuyez sur “Entrée” pour continuer l'installation des paquets.

Une fois l'installation terminée, il faut donner les droits sudo à l'utilisateur lov. Pour cela, taper la commande suivante, toujours en root:

```
adduser lov sudo
```

Une fois fait, il faut configurer la connexion SSH. Pour cela, il faut modifier le fichier /etc/ssh/sshd_config et s'assurer que les lignes suivantes soient présentes et non commenté:

```
PermitRootLogin prohibit-password
ChallengeResponseAuthentication yes
```

Maintenant, il faut configurer l'utilisateur root et lov pour que l'on puisse se connecter en SSH avec une clé de connexion SSH. Pour cela faut ajouter la clé publique de la clé SSH du bureau dans les fichiers:

- /root/.ssh/authorized_keys
- /home/lov/.ssh/authorized_keys

Le contenu à rajouter est le suivant:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDMK3cqsLxUtgs/k2Wx692ly+pkFEe1RjM0Id1yhmz1eSq5
Jk3f/XC7mLfyTjIJgyV5uBYhW2qcrVHlb9kYzol7KafdmctLcLaxG+gcIwJGU3SN5uEaFnJgdPfr
tgE8tFV0oHoXsykYLjmEpfLP1+Gw87fBGnOzgnCCetJqZoHp/BJ2FkV4MMdpf76jLgJUhSA7PsmA
Hg106fFzvlDZ6TMiwirbUUt+qd0Q3AhYaqk5kFr4XtpeWWSG6rRX/6KHsmxQ0VBcuBJZRnNDt3cc
Z8wKJP1ChT9exmjbDUE6kmw+2pg/pzXrNecw5uxmHaI3mULy+ObyC0nGJFUnEbh7Jk91
```

**Attention** : la configuration ci-dessous date d'avant le retrait de la Freebox (passerelle 192.168.42.254). Depuis le 12/10/2026, l'atom n'a plus de passerelle sur `enp2s0` et sort sur internet par `enp2s0.836` : voir la configuration actuelle dans Réseau du local.

Il faut ensuite modifier le fichier /etc/network/interfaces. Pour cela, vous pouvez copier/coller tous le contenu de la commande suivante:

```
cat > /etc/network/interfaces << EOF
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp2s0
iface enp2s0 inet static
        address 192.168.42.1/24
        gateway        192.168.42.254
# This is an autoconfigured IPv6 interface
iface enp2s0 inet6 auto
EOF
```

#### Etape 2 : Installation du DHCP/DNS/PXE

Nous allons maintenant installer dnsmasq qui fera office de serveur DHCP, DNS et PXE. Pour cela, taper la commande suivante:

```
apt install dnsmasq
```

Il faut ensuite le configurer, pour cela, vous pouvez copier/coller la commande suivante:

```
cat > /etc/dnsmasq.conf << EOF
# dhcp et dns de base
# avec routage relais DNS vers la freebox
no-resolv
expand-hosts
server=192.168.42.254
local=/mn.labovilleurbanne.fr/
domain=mn.labovilleurbanne.fr
dhcp-option=option:router,192.168.42.254
dhcp-range=192.168.42.100,192.168.42.150,12h

# boot PXE
dhcp-boot=pxelinux.0
enable-tftp
tftp-root=/srv/tftp
EOF
```

La configuration va: * Définir dnsmasq comme résolveur DNS du réseau * Définir le nom de domain du réseau * Définir la plage d'adressage IPv4 (192.168.42.0/24) * Définir la place DHCP (192.168.42.100 → 192.168.42.150) * Configurer le boot en PXE en lui disant d'utiliser le fichier pxelinux.0 qui se trouve dans /src/tftp

#### Etape 3 : Configuration de base du PXE

Il nous faut maintenant configurer le PXE, i.e. préparer l'OS qui sera envoyer au PC du LOV. Dans un premier temps, il faut aller dans le dossier /src/tftp

```
cd /src/tftp
```

Dans ce dossier, on va télécharger les fichiers de boot en PXE fournis par débian en tapant les commandes suivantes:

```
wget
http://ftp.nl.debian.org/debian/dists/buster/main/installer-amd64/current/im
ages/netboot/netboot.tar.gz
tar -zxvf netboot.tar.gz
```

On télécharge aussi le memtest pour être capable de faire des tests mémoires en cas de besoin:

```
wget http://www.memtest.org/download/5.01/memtest86+-5.01.bin.gz -O
memtest.gz
gunzip memtest.gz
```

Ensuite, on va personnaliser le fichier pxelinux.cfg pour ajouter différentes options de démarrage en plus de l'installation débian (lancement de débian, de memtest, etc.). Pour cela, copier/coller la commande suivante:

```
cat > pxelinux.cfg << EOF
# D-I config version 2.0
```

```
# search path for the c32 support libraries (libcom32, libutil etc.)
path debian-installer/amd64/boot-screens/
#include debian-installer/amd64/boot-screens/menu.cfg
default debian-installer/amd64/boot-screens/vesamenu.c32

menu title Welcome to LOV PXE

label run
        menu label run Debian Buster
        kernel diskless/buster/vmlinuz
        append ro initrd=diskless/buster/initrd.img root=/dev/nfs ip=dhcp
nfsroot=192.168.42.1:/srv/diskless/buster

label install
        menu label install Debian Buster
        kernel debian-installer/amd64/linux
        append vga=788 initrd=debian-installer/amd64/initrd.gz --- quiet

label memtest
        menu label run Memtest
        kernel memtest

prompt 0
timeout 0
EOF
```

Voilà, nous venons de configurer basiquement le démarrage des PC du LOV depuis le serveur “atom”. Maintenance, nous allons configurer le partage de fichier NFS pour les fichiers /home des PC du LOV soit accessible.

Pour cela, il faut installer nfs-kernel-server:

```
apt get install nfs-kernel-server
```

On crée ensuite le dossier /src/home qui servira de répertoir utilisateur au PC du LOV:

```
mkdir /srv/home
```

Enfin, on configure l'export des dossier /srv/diskless et /srv/home en partage NFS.

```
cat >/etc/exports <<EOF
/srv/diskless 192.168.42.0/24(ro,no_subtree_check,no_root_squash)
/srv/home 192.168.42.0/24(rw,async,no_subtree_check,no_root_squash)
EOF
```

Maintenant, il nous reste à configurer l'OS que nous envoyons au PC du LOV, ici une débian. Pour cela, on va faire une installation de base de l'OS dans le dossier /etc/diskless/buster:

```
debootstrap buster /src/diskless/buster
```

On va ensuite configurer un nom à l'OS de base pour que l'on puisse savoir que l'on est dedans lorsque qu'en cas de besoin on fait un chroot (qui ne sera pas aborder ici).

```
echo "diskless-buster" > /srv/diskless/buster/etc/debian_chroot
```

Ensuite, on va créer un script qui va nous permettre de lancer des commandes dans cette OS de base:

```
mkdir -p /root/bin/
cat >/root/bin/buster.sh << EOF
#!/bin/sh
chroot /srv/diskless/buster $@
EOF
```

On rends le script exécutable:

```
chmod +x /root/bin/buster.sh
```

Enfin, on peut maintenant lancer des commandes dans l'OS de base. Notament, ce qu'il faut faire aussitôt est d'installer le noyau et quelques outils:

```
buster.sh apt install linux-image-amd64 console-data locales nfs-common
```

Et maintenant, il faut générer le initrd.

```
buster.sh update-initramfs -k all -u
ln -s /srv/diskless /srv/tftp/diskless
```

Voilà, l'OS qui est envoyé sur les PC est maintenant configurer de base.

#### Résolution de problème

Lors que l'on est sur le réseau du LOV, les admins du serveur peuvent vouloir se connecter au serveur “atom” en passant par son nom dans le réseau pour son IP. Cependant, il se peut que lors que l'on fait

```
ping atom
```

ou

```
ssh atom
```

, il se peut que l'on tombe sur un PC dont l'IP est 127.0.1.1 (qui n'est autre que le PC sur lequel on est).

Ce problème viens du fait que DNSmasq utilise le fichier /etc/hosts du PC sur lequel il est installer pour faire la résolution DNS de celui-ci (cf: https://stackoverflow.com/questions/9326438/dnsmasq-serve-different-ip-addresses-based-on-interface-used).

Pour résoudre le problème, il va falloir modifier le fichier /etc/hosts et /etc/dnsmasq.conf. Tout d'abord dans le fichier /etc/hosts, il faut ajouter la ligne suivante:

```
192.168.1.42 atom
```

Ensuite, il faut dire à DNSmasq qu'il doit faire la résolution DNS en fonction de l'interface par laquel il est connectée. Ainsi, si on fait

```
ping atom
```

depuis “atom”, alors on va pinger 127.0.1.1, si on le fait depuis un autre PC du lov alors cela va pinger l'IP d'“atom” à savoir 192.168.42.1.

Pour ce faire, il faut ajouter les ligne suivante au fichier /etc/dnsmasq.conf

```
localise-queries
```

Et si on veut être vraiment sûr de la résolution DNS, on peut ajouter manuellement l'entrée du serveur atom dans /etc/dnsmasq.conf en ajoutant la ligne suivante:

```
host-record=atom,192.168.42.1
```

Enfin, il faut redémarrer le service DNSmasq pour prendre en compte les modifications. Dans un terminal sur “atom” en root, il faut taper les commandes suivantes:

```
systemctl daemon-reload
systemctl restart dnsmasq
```

**TODO**: Mettre en place et documenter la sauvegarde du SI (internet et externe).

**TODO**: Documenter la mise en place des gestion des noms de PC lors d'un boot en PXE.

## Installation de logiciels

- Se connecter sur atom (`ssh lov@192.168.42.1`, mdp dans les keepass bureau)
- passage root
- Créer une archive de backup en cas d'erreur
  - `tar -czf bookwork.bak_$(date –iso-8601) –directory=/srv/diskless/bookwork .`
- `/root/bin/buster.sh` pour passer dans le chroot du systme diskless
- un coup d' `apt install nom-paquet` si dispo debian
- sinon, installation dans /opt/ et on fait un lien symbolique dans /usr/local/bin/

**TODO** documenter l'ajout dans les menus de lxde (ou moyen d'ajouter facilement des raccourcis sur le bureau de tous les utilisateurs lovXXX)

# SI externe

## Boites mails

Hébergées chez Gandi comme le DNS

- Boite **bureau**
- Boite **contact**

## labovilleurbanne.fr

serveur VPS sous Debian hébergé chez Scaleway (mots de passe keepass du bureau).

nom de domaine hébergé sur gandi (TODO où sont les mots de passe ?).

tous les services sont gérés via yunohost:

- site web statique (pull depuis github)
- wordpress (accès /wp-admin à donner en dehors du SSO)
- dokuwiki
- nextcloud
- keeweb
- kanboard
- open sondage (TODO: encore utile ça ?)

administration de yunohost: mots de passe dans le keepass du bureau, créer les comptes pour les membres ou contributeurs

## framateam.org

service framasoft: https://framateam.org/labolov/

## mailing liste

http://listes.illyse.org/ ~~problème: pour administrer la liste il faut avoir un compte sur le SSO de illyse, réservé au membres de illyse, le seul contournement que j'ai trouvé est de me désabonner de la liste et au moment de cliquer sur le lien je me retrouve authentifié le temps d'une session et peux gérer les membres TODO: illyse nous ont signalé (été 2018) qu'ils n'ont pas vocation à être un provider de listes et que ça serait bien qu'on bouge~~

ARRÊTÉ DEPUIS 2022

# github

https://github.com/labovilleurbanne/site

TODO: qui a les accès à ce compte en écriture ?

```
=> Jeremy
```

From:
https://labovilleurbanne.fr/dokuwiki/ - **DokuWiki du LOV**

Permanent link:
**https://labovilleurbanne.fr/dokuwiki/asso:si:start?rev=1791842701**

Last update: **2026/10/12 22:05**