

# Installer un serveur Olimex A20, LIME 2



## Caractéristiques

- A20-OLinuXino-LIME2-s16M
- Cortex-A7 Dual core 1Ghz processor
- 1GB of RAM
- 32GB micro SD flash card
- Gigabit Ethernet
- Only 2W power consumption
- UPS build in Lipo battery up to 3 h stand alone work
- Metal Box for the computer and hard disk
- Option for SATA HDD and SSD — **ICI version 128Go SSD**
- Power supply 5V 2A adapter included
- Ethernet cable included |
- Allwinner A20 dual-core ARM Cortex-A7 @ 1GHz
- 1024Mb de mémoire vive
- 1 port Ethernet 1Gbit
- 1 port SATA (à utiliser avec le SATA-CALBESET)
- 2 ports USB 2.0 (puissance électrique limitée)
- 1 port MicroSD supportant les cartes jusqu'à 64Gb.

## Liens / références

- [Documentation de Yunohost](#)
- [Yunohost sur ARM](#)
- [Images yunohost](#)
- [Image yunohost mais version 3](#)
- [Images Olimex](#)
- [L'image minimal buster](#)
- [Transférer micro SD à SATA](#)

- [Transférer micro SD à SATA](#)
- [Manuel de l'Olimex A20, Lime2](#)

## Préconfiguration du FAI free

Ce serveur fera de l'autohébergement derrière [une box free mini 4K](#).

### Espace abonné

Voici les préalables de configuration de votre freebox (**espace abonné**)

- Demander une adresse IP fixe V4 full-stack
- Personnaliser mon reverse DNS.
- Blocage du protocole SMTP sortant : demander le déblocage.



Attention pour le rDNS, la littérature indique que ce réglage bien que disponible n'est en fait pas pris en compte par free.

Voir la section Problèmes pour plus de détails ->

### Configuration de la freebox

A réaliser une fois le serveur mis en route

**Dans la configuration de mafreebox, plusieurs étapes :**

- IP fixe sur le serveur
- Demander une DMZ sur le serveur
- Demander la redirection des ports

Rediriger ces ports

Web: 80 (HTTP), 443 (HTTPS)  
SSH: 22  
XMPP: 5222 (clients), 5269 (servers)  
Email: 25, 587 (SMTP), 993 (IMAP)

**Ce qui donne concrètement :**

mafreebox.freebox.fr/#Fbx.os.app.settings.ports.PortRedir

Redirections de ports Connexions entrantes

### Liste des redirections

Active	Redirection	IP source	Destination
Active	Protocole: tcp WAN : 80 LAN: 80 Commentaire:	Toutes	<a href="#">crust.ovh</a>
Active	Protocole: tcp WAN : 443 LAN: 443 Commentaire:	Toutes	<a href="#">crust.ovh</a>
Active	Protocole: tcp WAN : 22 LAN: 22 Commentaire:	Toutes	<a href="#">crust.ovh</a>
Active	Protocole: tcp WAN : 5222 LAN: 5222 Commentaire:	Toutes	<a href="#">crust.ovh</a>
Active	Protocole: tcp WAN : 25 LAN: 25 Commentaire:	Toutes	<a href="#">crust.ovh</a>
Active	Protocole: tcp WAN : 587 LAN: 587 Commentaire:	Toutes	<a href="#">crust.ovh</a>
Active	Protocole: tcp WAN : 993 LAN: 993 Commentaire:	Toutes	<a href="#">crust.ovh</a>
Active	Protocole: tcp WAN : 5269 LAN: 5269 Commentaire:	Toutes	<a href="#">crust.ovh</a>

Rafraichir Ajouter une redirection

DMZ

Activer la DMZ : ☒

IP DMZ :

## Graver l'image sur la carte micro SD

L'olimex A20 est une **image ARM**, la carte ne peut booter que sur le microSD. La carte SD sera utilisée pour le premier démarrage du serveur puis le système sera transféré vers le disque SATA. La carte pourra alors être retirée pour les prochains boots.

L'image yunohost étant en version 3, le choix s'est porté sur [l'image olimex debian buster \(stable\)](#) depuis laquelle sera installée par la suite yunohost.

**A réaliser depuis n'importe quel PC. Ici PC sous Linux.**



Il faut une carte microSD de classe 10

```
# apt-get install p7zip
```

```
$ 7za e A20-OLinuxino-buster-minimal-20201217-194545.img.7z
```

```
$ ls A20-OLinuxino-buster-minimal-20201217-194545.img*
A20-OLinuxino-buster-minimal-20201217-194545.img
A20-OLinuxino-buster-minimal-20201217-194545.img.7z
```

```
A20-0Linuxino-buster-minimal-20201217-194545.img.md5
```

Insérer la carte micro SD (class 10, ici une 32Ga)

```
[701710.426868] mmc0: cannot verify signal voltage switch
[701710.575537] mmc0: new ultra high speed SDR104 SDHC card at address 0001
[701710.576310] mmcblk0: mmc0:0001 SD 29.2 GiB
[701710.594672] mmcblk0: p1
```

### On flashe la carte

```
dd if=/home/ragnarok/Téléchargements/A20-0Linuxino-buster-
minimal-20201217-194545.img of=/dev/mmcblk0
```

## Premier démarrage du serveur

### Detecter le serveur sur le réseau local

Brancher le serveur (alim et RJ45). Pour un LIME2, le démonter au préalable et connecter la batterie. Puis on scanne le réseau local.

```
# arp-scan --local
Interface: wlp1s0, type: EN10MB, MAC: f4:8c:50:06:5a:90, IPv4: 192.168.0.36
Starting arp-scan 1.9.7 with 256 hosts
(https://github.com/royhills/arp-scan)
192.168.0.15    68:a3:78:2a:e3:8c    FREEBOX SAS
192.168.0.46    30:1f:9a:d0:33:ba    IEEE Registration Authority
192.168.0.17    a4:9b:4f:17:1e:60    HUAWEI TECHNOLOGIES CO.,LTD
192.168.0.29    00:71:cc:bd:cc:2b    Hon Hai Precision Ind. Co.,Ltd.
192.168.0.25    cc:25:ef:49:af:42    Apple, Inc.
192.168.0.11    6c:88:14:ad:7a:08    Intel Corporate
192.168.0.48    e4:fd:a1:db:eb:2d    HUAWEI TECHNOLOGIES CO.,LTD
192.168.0.30    a4:38:cc:cf:c1:e7    Nintendo Co.,Ltd
192.168.0.254   68:a3:78:78:86:f3    FREEBOX SAS
```

Le serveur est l'interface

```
192.168.0.46    30:1f:9a:d0:33:ba    IEEE Registration Authority
```

**(IEEE Registration Authority** indique la carte du serveur, dans le doute, faites un **# arp-scan -local avant et après démarrage du serveur** afin de voir l'IP ajoutée)

### Se connecter au serveur

On s'y connecte par ssh. Le couple login / password de sortie d'usine est :



login **olimex**

passwd **olimex**

```
$ ssh olimex@192.168.0.46
olimex@192.168.0.46's password:
Linux a20-olinuxino 5.8.18-olimex #122632 SMP Wed Dec 16 12:27:58 UTC 2020
armv7l
```

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/\*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.  
olimex@a20-olinuxino:~\$

## Transfert du système de la microSD vers le disque SATA

Le disque SSD ou HD fourni n'est pas formaté et le système est à ce stade sur la carte microSD.

Dans un premier temps :

- **formater le disque SATA**
- **puis transférer le système de la carte au SATA.**

### Formatage du disque SATA

Utiliser **cfdisk**:

```
olimex@a20-olinuxino:~ sudo cfdisk /dev/sda
olimex@a20-olinuxino:~$ sudo fdisk -l
```

```
Disk /dev/sda: 119.2 GiB, 128035676160 bytes, 250069680 sectors
Disk model: SPCC Solid State
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 58522E38-9E19-A84B-A1AA-6116052AD257
```

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	250069646	250067599	119.2G	Linux filesystem

### Puis formatage en ext4:

[download](#)

```
olimex@a20-olinuxino:~$ sudo mkfs.ext4 /dev/sda1
mke2fs 1.44.5 (15-Dec-2018)
Discarding device blocks: done
Creating filesystem with 31258449 4k blocks and 7815168 inodes
Filesystem UUID: 31e2f011-36dd-4013-930e-02fa62333233
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632,
    2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks):
done
Writing superblocks and filesystem accounting information: done
```

## Transfert microSD -> SATA

On transfère ensuite le système de la carte vers le disque. On utilise l'utilitaire **olinuxino-sd-to-sata**, disponible dans l'image de l'olimex.

[download](#)

```
olimex@a20-olinuxino:~$ sudo /sbin/olinuxino-sd-to-sata

Detecting board...

Detecting the root device...
Root partition:    /dev/mmcblk0p1
Root device:      /dev/mmcblk0

Detecting SATA device...
SATA partition:    /dev/sda1
SATA device:      /dev/sda

Checking disk size...
Required blocks:   1355024
Available blocks: 250069680

Unmount target device...
umount: /dev/sda1: not mounted.

Copying MBR record...

Resizing partition...
e2fsck 1.44.5 (15-Dec-2018)
```

```
ext2fs_open2: Bad magic number in super-block
e2fsck: Superblock invalid, trying backup blocks...
e2fsck: Bad magic number in super-block while trying to open /dev/sda1

The superblock could not be read or does not describe a valid
ext2/ext3/ext4
filesystem. If the device is valid and it really contains an
ext2/ext3/ext4
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate
superblock:
    e2fsck -b 8193 <device>
or
    e2fsck -b 32768 <device>

resize2fs 1.44.5 (15-Dec-2018)
resize2fs: Bad magic number in super-block while trying to open
/dev/sda1
Couldn't find valid filesystem superblock.

Formating disk...

Copying disk...
    632,730,175   99%    6.06MB/s    0:01:39 (xfr#21054, to-chk=0/26998)

Configuring...

Cleanup...
Writing u-boot-sunxi-with-spl
```

On peut booter même si la carte est retirée car le secteur de boot a été copié sur le SPI flash memory.

## Installation de yunohost

### Sans curl

A ce stade j'ai eu un soucis avec l'installation de yunohost car curl ne gère pas correctement les certificats de sécurité.

Donc installation à la main par récupération du script d'installation

[download](#)

```
$ wget https://install.yunohost.org/
$ mv index.html yunohost_install.sh
```

Lancement du script d'installation de yunohost. Elle prend un certain temps, faire du café ;)

[download](#)

```
$ sudo bash yunohost_install.sh
[INFO] Running upgrade_system
[INFO] Running install_script_dependencies
[INFO] Running create_custom_config
[INFO] Running confirm_installation
[INFO] Running manage_sshd_config
[INFO] Running fix_locales
Generating locales (this might take a while)...
  en_GB.UTF-8... done
  en_US.UTF-8...^[[C done
Generation complete.
/etc/environment: line 1: warning: setlocale: LC_ALL: cannot change
locale (en_US.UTF-8): No such file or directory
[INFO] Running setup_package_source
[INFO] Running apt_update
[INFO] Running register_debconf
[INFO] Running workarounds_because_sysadmin_sucks
[INFO] Workaround for avahi : creating avahi user with uid 954
[INFO] Running install_yunohost_packages
[INFO] Running restart_services
[INFO] Installation logs are available in /var/log/yunohost-
installation_20201226_122427.log
[ OK ] YunoHost installation completed !

=====
=====
You should now proceed with Yunohost post-installation. This is where
you will
be asked for :
  - the main domain of your server ;
  - the administration password.

You can perform this step :
  - from the command line, by running 'yunohost tools postinstall' as
root
  - or from your web browser, by accessing :
    - https://192.168.0.46/ (local IP, if self-hosting at home)

If this is your first time with YunoHost, it is strongly recommended to
take
time to read the administator documentation and in particular the
sections
'Finalizing your setup' and 'Getting to know YunoHost'. It is available
at
the following URL : https://yunohost.org/admindoc

=====
=====
```





Attention, il faut régler le soucis de **curl**

car sinon **vous ne pourrez pas utiliser de certificat Let's encrypt pour vos noms de domaines**

## Avec curl

Pour régler le soucis de curl. Curl n'est pas installé par défaut, l'installer et de suite régénérer les certificats

[download](#)

```
# sudo apt install curl
# sudo update-ca-certificates -f
```

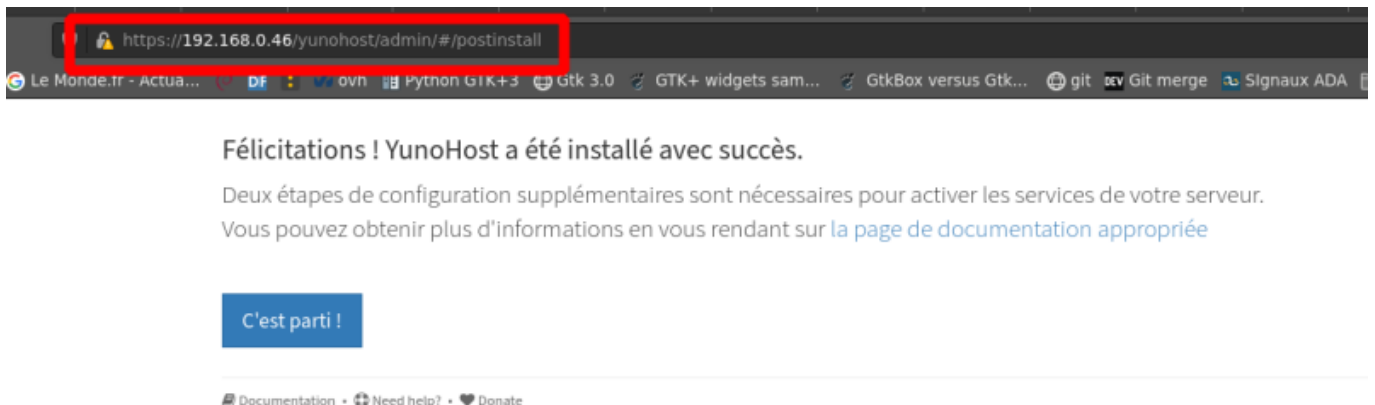
Curl est opérationnel, yunohost peut s'installer alors facilement par

[download](#)

```
# curl https://install.yunohost.org | bash
```

## Post installation de yunohost

Comme stipulé à la fin de l'installation de yunohost, on récupère le lien de post configuration de yunohost et on peut finir l'installation via un navigateur internet (elle peut être faite aussi en CLI, lire la documentation de yunohost dans ce cas)



L'ajout de l'utilisateur admin sera réalisée et il est impossible, pour des raisons de sécurité de se connecter

avec le compte root.

Dès lors se connecter

[download](#)

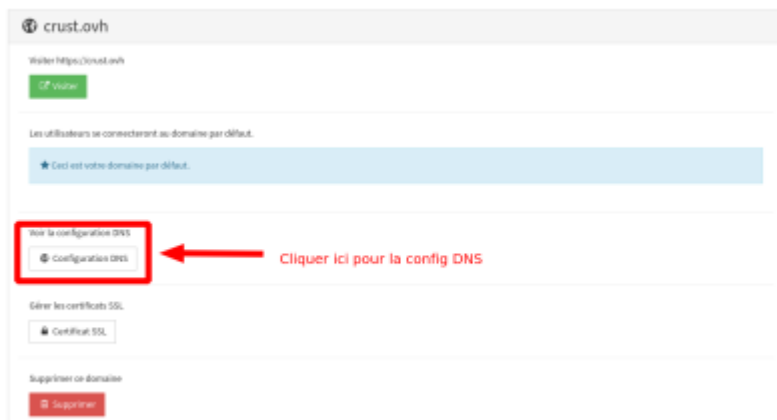
```
ssh admin@mon_non_de_domaine.ext
```

Pour aller plus loin sur la sécurisation de yunohost :

- [Sécurité Yunohost](#)
- [Sécuriser fail2ban et ssh](#)

## Relier un domaine à son yunohost

C'est dans le panel d'administration sur **DOMAINES, AJOUTER UN DOMAINES**

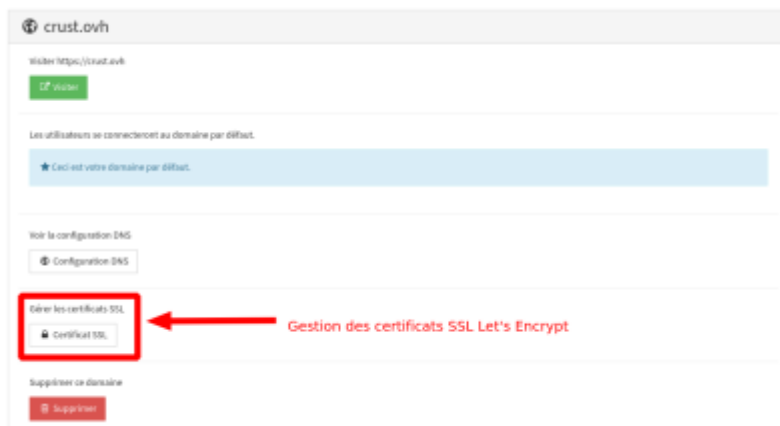


Yunohost vous fournira tous les données nécessaires pour remplir correctement la zone DNS du registrar où vous avez acheté le nom de domaine.



Se connecter au panel d'administration de son registrar et renseigner la zone DNS à partir des dernières informations. Il se peut qu'un **temps de prise en compte soit nécessaire** le temps de diffusion de la mise à jour de ces entrées.

**Demander un certificat SSL et Let's Encrypt**



## Problèmes rencontrés

### curl

Suite à l'installation de l'image d'Olimex ARM, j'ai dû demander la mise à jour des certificats afin que curl fonctionne correctement

#### download

```
# sudo update-ca-certificates -f
```

### Blacklist de mail

Le rapport de diagnostic de yunohost permet de voir les soucis de configuration. Il se peut que l'IP attribuée soit blacklistée, dans ce cas, suivre les indications afin de demander son déblocage.

### rDNS Free

[Mail tester](#) est un utilitaire en ligne gratuit afin de tester sa configuration mail (limité à 3 test gratuits par jour dans sa version gratuite).

Il semble que free ne tiennent pas compte de la configuration rDNS bien qu'il soit possible de la modifier dans l'espace abonné. Vous perdez donc 0,3 points. Ce qui reste acceptable. Pas trouvé de solution à ce jour.

Utiliser nslookup pour vérifier le rDNS

#### download

```
$ nslookup 82.65.208.68
68.208.65.82.in-addr.arpa    name = 82-65-208-68.subs.proxad.net.
```

Authoritative answers can be found from:

Le résultat attendu est : crust.ovh ...

Bien que correctement enregistrer et paramétrer pour mon nom de domaine, la prise en charge n'est pas effective. Les forums free et yunohost le confirment. Solutions :

- utiliser un VPN pour le rDNS
- suivre les conseils de mail-tester avec le hostname (cette dernière solution ne semble pas porter ses fruits...)

## NO\_FM\_NAME\_IP\_HOSTN

Alors là, je comprends très peu la situation. Car pour un autre nom de domaine réglé exactement pareil, pas de soucis.

0.1	DKIM_VALID_EF	Message has a valid DKIM or DK si
-2.5	NO_FM_NAME_IP_HOSTN	NO_FM_NAME_IP_HOSTN
-0.001	RCVD IN SORBS DUL	SORBS: sent directlv from dynamic

Mais ça fait perdre 2.5 points par rapport à spam assassin ce qui peut être gênant.

J'ai réglé ce soucis dans le paramétrage de mon compte mail (client et webmail) en mettant un nom composé plus qu'un nom simple dans le nom du compte.

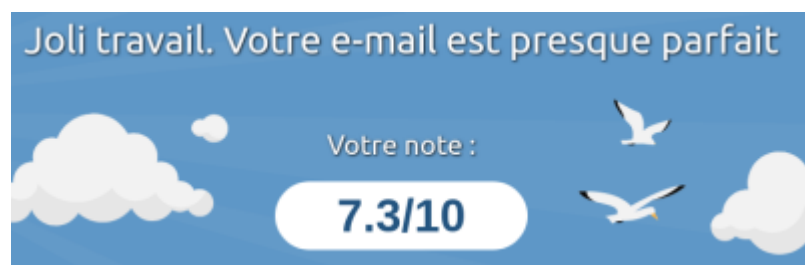
- crust -> -2,5 points
- Crust Your Mind -> ça passe

Honnêtement, je ne comprends pas pourquoi. Solution trouvée sur le forum yunohost :

<https://forum.yunohost.org/t/office-360-qui-refuse-mes-mails-yunohost/11521/10>

## Rapport de mail-tester

**Sans la configuration Nom + Prénom :**



**Avec la configuration Nom Prénom dans le client mail et le webmail :**



## Divers

### Batterie

#### Sur batterie

[download](#)

```
cat /sys/bus/i2c/devices/0-0034/axp20x-battery-power-
supply/power_supply/axp20x-battery/uevent
POWER_SUPPLY_NAME=axp20x-battery
POWER_SUPPLY_TYPE=Battery
POWER_SUPPLY_PRESENT=1
POWER_SUPPLY_ONLINE=1
POWER_SUPPLY_STATUS=Discharging
POWER_SUPPLY_VOLTAGE_NOW=4037000
POWER_SUPPLY_CURRENT_NOW=508000
POWER_SUPPLY_CONSTANT_CHARGE_CURRENT=1200000
POWER_SUPPLY_CONSTANT_CHARGE_CURRENT_MAX=1200000
POWER_SUPPLY_HEALTH=Good
POWER_SUPPLY_VOLTAGE_MAX_DESIGN=4200000
POWER_SUPPLY_VOLTAGE_MIN_DESIGN=2900000
POWER_SUPPLY_CAPACITY=97
```

#### Sur secteur

[download](#)

```
cat /sys/bus/i2c/devices/0-0034/axp20x-battery-power-
supply/power_supply/axp20x-battery/uevent
cat /sys/bus/i2c/devices/0-0034/axp20x-battery-power-
supply/power_supply/axp20x-battery/uevent
POWER_SUPPLY_NAME=axp20x-battery
POWER_SUPPLY_TYPE=Battery
POWER_SUPPLY_PRESENT=1
POWER_SUPPLY_ONLINE=1
POWER_SUPPLY_STATUS=Charging
POWER_SUPPLY_VOLTAGE_NOW=4215000
```

```
POWER_SUPPLY_CURRENT_NOW=800000
POWER_SUPPLY_CONSTANT_CHARGE_CURRENT=1200000
POWER_SUPPLY_CONSTANT_CHARGE_CURRENT_MAX=1200000
POWER_SUPPLY_HEALTH=Good
POWER_SUPPLY_VOLTAGE_MAX_DESIGN=4200000
POWER_SUPPLY_VOLTAGE_MIN_DESIGN=2900000
POWER_SUPPLY_CAPACITY=89
```

Pour plus de simplicité, réaliser l'alias

[download](#)

```
$ cat .bash_aliases
alias battery="cat /sys/bus/i2c/devices/0-0034/axp20x-battery-power-supply/power_supply/axp20x-battery/uevent"
```

[download](#)

```
$ source ~/.bashrc
```

Deconnexion SSH auto

Dans le .bashrc

[download](#)

```
# Export TMOUT
# Délai inactivité avant deconnexion
TMOUT=600
```

Outils yunohost essentiels (CLI)

sudo yunohost firewall list	Liste les ports ouverts
sudo yunohost app list	Liste les applications installées
sudo yunohost app info horde	Donne des informations sur l'application passée en IP
sudo yunohost domain list	Liste les domaines gérés par l'instance yunohost
sudo yunohost tools update	Mise à jour de la liste des paquets
sudo yunohost tools upgrade system	Mise à jour du système
sudo yunohost tools upgrade apps	Mise à jour des applications
sudo yunohost tools app upgrade NAME_APP	Mise à jour de l'application NAME_APP
sudo yunohost diagnosis run	Lancer un diagnostique système
sudo yunohost diagnosis run -force	Force la réitération d'un diagnostic système
sudo yunohost diagnosis show	Affiche le bilan du diagnostique système

<code>sudo yunohost diagnosis show -issues</code>	Affiche que les erreurs d'un diagnostique système
<code>sudo yunohost backup create</code>	Crée une sauvegarde complète du système
<code>sudo yunohost backup create -apps</code>	Crée une sauvegarde des applications
<code>sudo yunohost backup create -system</code>	Crée une sauvegarde des données système
<code>sudo yunohost backup create -system data_mail</code>	Crée une sauvegarde des mails
<code>sudo yunohost backup list</code>	Liste les sauvegardes effectuées

## Ajouter un fichier de swap

### Créer un fichier de swap de 2go

[download](#)

```
$ sudo dd if=/dev/zero of=/swapfile bs=1M count=2048 status=progress
```

On peut utiliser cette commande.

[download](#)



```
$ sudo fallocate -l 4G /swapfile
```

Mais personnellement, ça ne passe pas. Donc préférer la précédente

### Appliquer les droits suivants

[download](#)

```
$ sudo chmod 600 /swapfile
```

### Vérifier

[download](#)

```
$ ls -lh /swapfile
-rw----- 1 root root 2.0G Dec 31 15:28 /swapfile
```

### Transformer ce fichier en swap

[download](#)

```
$ sudo mkswap /swapfile
Setting up swapspace version 1, size = 2 GiB (2147479552 bytes)
no label, UUID=a5644688-4d53-4e30-a388-f2a023741ebf
```

## Appliquer le swap

[download](#)

```
$ sudo swapon /swapfile
```

## Vérifier la bonne prise en charge

[download](#)

```
$ sudo swapon -s
```

Filename	Type	Size	Used	Priority		
/swapfile		file		2097148	0	-2

## Pour rendre le montage permanent, ajouter à fstab

[download](#)

```
$ sudo nano /etc/fstab
```

Ajouter ces lignes

[download](#)

```
# Add swap
/swapfile none swap defaults 0 0
```

## Vérifier

[download](#)

```
$ free
```

	total	used	free	shared	buff/cache
available					
Mem:	1021080	229364	582704	13976	209012
754220					
Swap:	0	0	0		



## Etre informé des mises à jour

Installer simplement ces 2 paquets :

[download](#)

```
$ sudo apt install apt-listchanges apticron
```

## Rapport mail quotidien de fail2ban

Créer un répertoire de scripts pour admin

[download](#)

```
$ mkdir scripts
```

Créer le fichier fail2ban-status-ban.sh

[download](#)

```
$ nano fail2ban-status-ban.sh
#!/bin/sh
# Script de rapport fail2ban
# Prend en compte tous les jails
#
dest=mon_adress_mail@domaine.extension

msg=$(fail2ban-client status | sed -n 's/,//g;s/.*Jail list://p' |
xargs -n1 fail2ban-client status);
echo "$msg" | mail -s "Rapport Fail2ban De $(hostname) $(date)" $dest
```

Juste adapter la variable **dest**.

Positionner un cron chez le root

[download](#)

```
$ sudo crontab -e
```

[download](#)

```
# Rapport de fail2ban
15 6,19 * * * /home/admin/scripts/fail2ban-status-ban.sh
```

Vous recevrez **quotidiennement** à 6h15 et à 19h15 un rapport de fail2ban sous cette forme **par mail** :

[download](#)

```
Status for the jail: dovecot
|- Filter
|   |- Currently failed: 0
|   |- Total failed: 0
|   `-- File list:      /var/log/mail.log
`-- Actions
    |- Currently banned: 0
    |- Total banned: 0
    `-- Banned IP list:

Status for the jail: nginx-http-auth
|- Filter
|   |- Currently failed: 0
|   |- Total failed: 0
|   `-- File list:      /var/log/nginx/xmpp-upload.crust.ovh-error.log
/var/log/nginx/error.log /var/log/nginx/crust.ovh-error.log
`-- Actions
    |- Currently banned: 0
    |- Total banned: 0
    `-- Banned IP list:

Status for the jail: pam-generic
|- Filter
|   |- Currently failed: 1
|   |- Total failed: 2314
|   `-- File list:      /var/log/auth.log
`-- Actions
    |- Currently banned: 0
    |- Total banned: 0
    `-- Banned IP list:

Status for the jail: postfix
|- Filter
|   |- Currently failed: 0
|   |- Total failed: 2
|   `-- File list:      /var/log/mail.log
`-- Actions
    |- Currently banned: 0
    |- Total banned: 0
    `-- Banned IP list:

Status for the jail: recidive
|- Filter
|   |- Currently failed: 7
|   |- Total failed: 701
|   `-- File list:      /var/log/fail2ban.log
`-- Actions
    |- Currently banned: 38
    |- Total banned: 38
    `-- Banned IP list:  49.88.112.110 49.88.112.68 221.181.185.222
```

```
222.187.222.105 221.181.185.136 221.181.185.141 222.187.222.55
221.181.185.135 222.187.222.53 49.88.112.75 221.181.185.19
222.187.238.57 222.187.238.93 218.93.208.28 221.131.165.85
221.181.185.148 222.187.227.224 221.131.165.86 221.131.165.124
221.181.185.18 221.181.185.221 221.181.185.199 221.181.185.223
221.181.185.198 221.181.185.36 218.93.207.94 221.181.185.143
221.181.185.149 222.187.224.30 222.187.238.39 222.187.227.177
221.181.185.29 221.131.165.119 218.93.207.40 218.93.207.84
222.187.238.87 221.131.165.87 221.181.185.200
Status for the jail: sshd
|- Filter
| |- Currently failed: 1
| |- Total failed: 7588
| `-- File list:    /var/log/auth.log
`- Actions
   |- Currently banned: 0
   |- Total banned: 703
   `-- Banned IP list:
Status for the jail: yunohost
|- Filter
| |- Currently failed: 0
| |- Total failed: 0
| `-- File list:    /var/log/nginx/xmpp-upload.crust.ovh-error.log
/var/log/nginx/error.log /var/log/nginx/crust.ovh-error.log
/var/log/nginx/xmpp-upload.crust.ovh-access.log
/var/log/nginx/access.log /var/log/nginx/crust.ovh-access.log
`- Actions
   |- Currently banned: 0
   |- Total banned: 0
   `-- Banned IP list:
```

## Etre notifié des connexion ssh par SMS

- [https://cbiot.fr/dokuwiki/ssh-fail2ban#avoir\\_un\\_rapport\\_de\\_connexion\\_ssh](https://cbiot.fr/dokuwiki/ssh-fail2ban#avoir_un_rapport_de_connexion_ssh)

## Installer bpytop, moniteur de ressources

**bpytop** est un moniteur de ressources qui permet d'afficher les statistiques et de gérer l'usage du(es) processeur(s), des processus, de la mémoire et du réseau.

[download](#)

```
git clone https://github.com/aristocratos/bpytop.git
cd bpytop
sudo make install
```

## Pour le desinstaller

[download](#)

```
sudo make uninstall
```

## Installer **\*\*pflogsumm\*\*** , un analyseur de mails

[download](#)

```
sudo apt install pflogsumm
```

Tester les mails du jour :

[download](#)

```
sudo /usr/sbin/pflogsumm -d today /var/log/mail.log
```

Automatiser la tâche et être notifié par mail

[download](#)

```
sudo crontab -e
```

[download](#)

```
# Stats Mail ( pflogsumm )
00 19 * * * /usr/sbin/pflogsumm -d today /var/log/mail.log | mail -s
"Postfix Rapport du `date`" yourmail@domain.tld
```

avec yourmail@domain.tld -> adresse mail et vous recevrez un rapport détaillé tous les soirs à 19h.

## Redirigé si IP renseignée plutôt que le nom de domaine

Par défaut, si on entre son IPv4 plutôt que son nom de domaine, on tombe sur la page d'administration de son yunohost.

Pour personnaliser la redirection, éditer le fichier :

[download](#)

```
sudo nano /etc/nginx/conf.d/default.d/redirect_to_admin.conf
```

Apporter les modifications suivantes :

[download](#)

```
location / {  
#    return 302 https://$http_host/yunohost/admin;  
    return 302 https://crust.ovh ;  
}
```

En précisant votre nom de domaine ou l'URL précise de redirection.

Relancer nginx.

[download](#)

```
sudo systemctl reload nginx.servic
```

## Envoyer un mail à tous les utilisateurs du système

Voici un script (minimal) qui fonctionne

Créer un fichier **mail-all-users.sh**

[download](#)

```
#!/bin/bash  
  
# Test validité des arguments  
if [[ $# -ne 2 ]]; then  
    echo "Nombre argument incorrect"  
    echo "2 arguments nécessaires : le sujet et le nom du fichier du  
corps du mail"  
    exit  
else  
    echo "Nombre d'arguments correct"  
  
    # Test que le second argument soit un fichier  
    if [[ -f $2 ]]; then  
        echo "Second argument valide"  
    else  
        echo "Le fichier indiqué n'est pas valide"  
        exit  
    fi  
fi  
  
corps=$(cat $2)  
users=$(ldapsearch -x -LLL uid=* | grep uid: | cut -d: -f2)
```

```
for line in $users
do
    echo $corps | mail -s "$1" $line
done
```

Il suppose que le corps du mail soit saisi dans un fichier au même niveau d'arborescence que le script

Lancement du script

[download](#)

```
mail-all-users.sh parametre1 parametre2
```

Avec **paramètre 1** le sujet du mail et **paramètre 2** le nom du fichier contenant le corps du mail  
Exemple pour un fichier corps se nommant test\_corps.txt au même niveau d'arborescence que le script, ça donnerait

[download](#)

```
mail-all-users.sh "sujet du jour" test_corps.txt
```

From:

<https://cbiot.fr/dokuwiki/> - **Cyrille BIOT**

Permanent link:

<https://cbiot.fr/dokuwiki/homeserver:olinolinux?rev=1669218873>

Last update: **2022/11/23 16:54**

