

Plateforme de création des images ISO

(Inspiration : <https://gist.github.com/matsimon/4187828>)

Installation de FreeBSD 8.x

A la fin de l'installation :

- No ports
- No packages
- Activer ssh

Au reboot, mise à jour :

```
freebsd-update fetch install  
portsnap fetch extract
```

Préparation de l'environnement de build :

```
echo "WITHOUT_X11=yo" >> /etc/make.conf  
echo "BATCH=yo" >> /etc/make.conf  
echo "NO_WARNING_PKG_INSTALL_EOL=yes" >> /etc/make.conf
```

Installation de portmaster :

```
cd /usr/ports/ports-mgmt/portmaster  
make depends install  
rehash
```

Outils nécessaires au builder (ça va être long, pour git) :

```
portmaster textproc/expat2  
portmaster devel/git  
portmaster devel/subversion  
portmaster sysutils/fastest_cvsup  
rehash
```

Installation du "pfsense-tools" avec modification UnivNautes :

```
cd /home/pfsense  
git clone ssh://git@repos.entroutvert.org/univnautes-tools.git tools  
chmod a+r tools/builder_scripts/*.sh
```

Copie des sources de l'installateur FreeBSD saveur pfSense:

```
cd /home/pfsense  
git clone https://github.com/pfsense/freesbie2 freesbie2
```

Quelques répertoires à créer, aussi...

```
mkdir -p /home/pfsense/pfSenseGITREPO /usr/pfSensesrc
```

build.sh

L'outils de build général et son aide intégrée :

```

cd /home/pfsense/tools/builder_scripts
./build.sh

Usage ./build.sh [options] [ iso | nanobsd | ova | nanobsd-vga | memstick | memstickserial | fullupdate | all ]
  all = iso nanobsd nanobsd-vga memstick memstickserial fullupdate
  [ options ]:
    --flash-size|-f size(s) - a list of flash sizes to build with nanobsd i.e. '512m 1g'. Default: 512m
    --flash-vendor|-m vendor - flash vendor type. check the code for supported vendors. Default: sandisk
    --no-buildworld|-c - Will set NO_BUILDWORLD NO_BUILDKERNEL to not build kernel and world
    --no-cleanobjdir|--no-cleanrepos|-d - Will not clean FreeBSD object built dir to allow restarting a build with NO_CLEAN
    --resume-image-build|-r - Includes -c -d and also will just move directly to image creation using pre-staged data
    --apply-patches - Fetch FreeBSD sources and apply patches
    --build-pfPorts - Rebuild all ports required to generate an image
    --build-pfPort port - Rebuild a single port required to generate an image
    --builder-required-ports - Build ports required from builder tools to operate
    --update-sources - Refetch all sources
    --print-flags - Show current builder configuration
    --clean-builder - clean all builder used data/resources
    --configure - generate a config file to be used by builder tools
    --build-kernels - build all configured kernels
    --build-kernel argument - build specified kernel. Example --build-kernel pfSense_SMP.10
    --install-extra-kernels argument - Put extra kernel(s) under /kernel image directory. Example --install-extra-kernels pfSense_wrap.10.i386
    --enable-memorydisks - This will put /usr/local/pfsense-fs and /usr/local/pfsense-clone as MFS filesystems
    --disable-memorydisks - Will just teardown these filesystems created by --enable-memorydisks

```

Compilation des ports

Pour tout reconstruire :

```
./build.sh --build-pfPorts
```

Pour un port spécifique :

```
./build.sh --build-pfPort lasso
```

Les ports nécessaires à UnivNautes sont visibles en bas de
`/home/pfsense/tools/builder_scripts/conf/pfPorts/buildports.RELENG_2_1` :

bash	/usr/ports/shells/bash	/usr/local/bin/bash
python	/usr/ports/lang/python	/usr/local/bin/python
python2	/usr/ports/lang/python2	/usr/local/bin/python2
python27	/usr/ports/lang/python27	/usr/local/bin/python2.7
py-sqlite3	/usr/ports/databases/py-sqlite3	/usr/local/lib/python2.7/lib-dynload/_sqlite3.so
lasso	/usr/ports/security/lasso	/usr/local/lib/liblasso.so.13
py-django15	/usr/ports/www/py-django15	/usr/local/bin/django-admin.py
py-flup	/usr/ports/www/py-flup	/usr/local/lib/python2.7/site-packages/flup/__init__.py
py-authentic2	/usr/ports/www/py-authentic2	/usr/local/lib/python2.7/site-packages/authentic2/__init__.py

Construction image ISO

- Initialisation :

```
cd /home/pfsense/tools/builder_scripts  
git pull
```

- nettoyage

```
./build.sh --clean-builder
```

- mise à jour des sources freebsd & univnautés

```
./build.sh --update-sources
```

- application des patches sur FreeBSD

```
./build.sh --apply-patches
```

- construction des ports

```
rm /tmp/pfSense_do_not_build_pfPorts  
./build.sh --build-pfPorts
```

- construction iso

```
./build.sh iso
```

- re-construction sans recompilation & co

```
./build.sh --resume-image-build iso
```