

## Contents

<b>INSTALLATION DE LIBHYDRO SOUS LINUX</b>	<b>1</b>
Installation packagée de Python (CONSEILLE) . . . . .	1
Compilation de Python (NON CONSEILLE) . . . . .	1
Installer les dépendances . . . . .	2
Compiler la version la plus récente de la série 2 de Python . . . .	2
Installer pip et les setuptools . . . . .	2
Installation et utilisation de virtualenv (Réf 2) . . . . .	3
Installer virtualenv avec pip . . . . .	3
Utiliser un environnement virtuel . . . . .	3
Installer les dépendances de libhydro dans l'environnement virtuel . .	3
Numpy . . . . .	3
Installer libhydro . . . . .	5

## INSTALLATION DE LIBHYDRO SOUS LINUX

- Auteur: philippe.gouin@developpement-durable.gouv.fr
- Version: 0.1d
- Mise à jour: 2017-03-17

### Installation packagée de Python (CONSEILLE)

Installer python et pip, un installateur de librairies.

Si yum ou apt ne proposent pas la bonne version de Python, utiliser un dépôt ou un paquet tiers ou bien passer par une compilation.

Exemple avec yum:

```
#!/sh
yum install python python-pip.noarch
```

Exemple avec apt:

```
#!/sh
apt-get install python python-pip
```

### Compilation de Python (NON CONSEILLE)

La procédure décrite ci-dessous est pour CentOS 6.2 (Réf 1).

NB: la version python 2.4.3 de l'os, nécessaire pour pas mal de programmes, est /usr/bin/python.

## Installer les dépendances

```
#!/sh
# the first one is big but requested to enable all python's features
yum groupinstall "Development tools"
yum install gcc zlib-devel bzip2-devel openssl-devel ncurses-devel \
    readline-devel sqlite sqlite-devel
yum install readline-devel.x86_64 readline-devel.i386
yum install zlib-devel.x86_64 zlib-devel.i386
# sqlite is requested by ipython
yum install sqlite.x86_64 sqlite-devel.x86_64
# optional
yum install openssl-devel.i386 openssl-devel.x86_64
```

## Compiler la version la plus récente de la série 2 de Python

```
#!/sh
wget http://www.python.org/ftp/python/2.x.y/Python-2.x.y.tar.bz2
./configure
make
# use 'make altinstall' if you care overwriting a previous python installation
# the default install is in /usr/local/bin/python
su -c 'make install'
```

Si problème relatif à sqlite3, il faut patcher les sources:

```
#!/sh
wget -O patch_sqlite https://raw.github.com/gist/2727063/
cat patch_sqlite | patch -p1
```

## Installer pip et les setuptools

```
#!/sh
wget --no-check-certificate https://pypi.python.org/packages/source/s/setuptools/setuptools-
wget --no-check-certificate https://pypi.python.org/packages/source/p/pip/pip-x.y.z.tar.gz
python setup.py install # pour chacun des modules
```

Configurer les variables `http_proxy`, `https_proxy` et `ftp_proxy` pour permettre à pip de télécharger les paquets par tous les moyens possibles.

On peut aussi préciser le proxy directement dans la commande install:

```
#!/sh
pip --proxy http://(proxy) install (paquet)
```

## Installation et utilisation de virtualenv (Réf 2)

### Installer virtualenv avec pip

```
#!/sh
pip install virtualenv
```

### Utiliser un environnement virtuel

La commande virtualenv permet de créer un nouvel environnement:

```
#!/sh
virtualenv --system-site-packages (dest_dir)
```

On peut:

- utiliser ou pas les paquets de l'environnement de base avec `--system-site-packages` ou `--no-site-packages` (dans le premier cas les paquets sont surchargés par ceux du virtualenv)
- forcer l'interpréteur à utiliser avec `-p` (python path)
- formater le prompt avec `--prompt=(prompt)`

L'environnement virtuel s'active:

- avec la commande `'source (virtualenv)/bin/activate'` (deactivate pour quitter)
- en modifiant le python PATH
- en utilisant virtualenvwrapper

Exemple de création d'un environnement virtuel avec le wrapper:

```
#!/sh
mkvirtualenv --system-site-packages --prompt '[libhydro]' ~/.virtualenvs/libhydro
```

## Installer les dépendances de libhydro dans l'environnement virtuel

### Numpy

Numpy est un pré-requis pour Pandas.

```
#!/sh
(virtualenv)/bin/pip install numpy
```

Difficultés éventuelles possibles, Numpy est un module en C qui peut nécessiter le package python-devel.

En cas de warnings inappropriés émis par `numpy.loadtxt()`:

- edit the `npzio.py` file (can be in `(python)/lib/python2.7/site-packages/numpy-1.6.2-py2.7-linux-x86_64.egg/numpy/lib/` or `/lib/python2.7/site-packages/numpy/lib/`)
- comment the line 773 `”warnings.warn(’loadtxt: Empty input file: “%s” ’ % fname)”`
- comment the line 1311 `”warnings.warn(’genfromtxt: Empty input file: “%s” ’ % fname)”`

### **Pandas (do not forget the ‘s’ !!)**

Pandas est utilisé pour stocker les séries de données temporelles (observations et prévisions).

```
#!/sh
pip install pandas
```

Les dépendances `python-dateutil`, `pytz` et `six` sont automatiquement installées.

### **Lxml**

Lxml est utilisé par le codec xml.

```
#!/sh
sudo yum install libxslt-devel.x86_64 libxml2-devel.x86_64
pip install lxml
```

### **Suds (version suds-jurko)**

Suds est utilisés pour les connections à la BdHydro.

```
#!/sh
pip install suds-jurko
```

### **Check or install with pip freeze**

```
#!/sh
pip freeze
numpy==1.7.1
pandas==0.11.0
lxml==3.2.3
python-dateutil==2.1
pytz==2013b
six==1.3.0
suds-jurko==0.6

pip install -r (pip_freeze_file)
```

## Installer libhydro

Ouvrir un terminal et faire:

```
#!/sh
```

```
pip install 'https://bitbucket.org/pch_fr/libhydro/downloads/libhydro-0.5.3-py2-none-any.whl'
```