**LAF FABRIC installation procedure**

1. Install Ubuntu 14.10 64 bit (either server or desktop version)
   1. run a software-update
      1. *sudo apt-get updat*e
      2. *sudo apt-get upgrade*
2. Package installation (in this order): either with the "*sudo apt-get install*" command or with a GUI tool like synaptic.
   1. virtualbox-guest-dkms
   2. dpkg-dev
   3. ssh
   4. openssh-server
   5. sqlite3
   6. python-all-dev
   7. python3-all-dev
   8. python3-markdown
   9. python3-pip
   10. python3-pandas
   11. ipython3
   12. ipython3-notebook
   13. libmysqlclient-dev
   14. libwxgtk3.0-0 (only when you want to install also emdros tools with GUI [e.g. Emdros query tool])
   15. libwxgtk3.0-dev (only when you want to install also emdros tools with GUI [e.g. Emdros query tool])
   16. libwxgtk-webview3.0-0 (only when you want to install also emdros tools with GUI [e.g. Emdros query tool])
   17. libwxgtk-media (only when you want to install also emdros tools with GUI [e.g. Emdros query tool])
   18. nano
   19. vim-nox
   20. unzip
   21. git
3. Download EMDROS tar source files (<http://emdros.org/download.html>):
   1. unpack tar file (*tar xzvf \*.tar.gz*)
   2. cd into unpacked emdros folder
   3. run: *./configure --prefix=/usr/local --with-sqlite3=local --with-mysql=yes --with-swig-language-java=no --with-swig-language-python=yes --with-sqlite=no --with-postgresql=no --with-wx=no --with-swig-language-csharp=no --with-swig-language-perl=no --with-swig-language-ruby=no --with-bpt=no --disable-debug*
   4. run: *make*
   5. run: *sudo make install*
   6. create conf file: *sudo nano /etc/ld.so.conf.d/usrlocal.conf*
      1. write: */usr/local/lib*
      2. save the textfile
   7. run: *sudo ldconfig*
   8. test:
      1. go to the folder where you have your sqlite database generated from mql
      2. run: *echo 'select all objects where [book]' | mql -b s3 -d "name of database"*. In my case: *echo 'select all objects where [book]' | mql -b s3 -d bhs4\_20150227.sqlite3*
      3. if you do not get an error message you can proceed with the installation
4. Get laf-fabric-data:
   1. download from DANS laf-fabric-data.zip (<https://easy.dans.knaw.nl/ui/datasets/id/easy-dataset:58245/tab/2>)
   2. unpack laf-fabric-data.zip (*unzip laf-fabric-data.zip*)
   3. move laf-fabric-data to home folder. In my case: *mv /home/shebanq/Downloads/laf-fabric-data /home/shebanq/*
5. Create output folder
   1. *mkdir /home/shebanq/laf-fabric-output*
6. Installation of laf-fabric:
   1. run: *git clone* [*https://github.com/ETCBC/laf-fabric*](https://github.com/ETCBC/laf-fabric)
   2. move laf-fabric to home folder. In my case: *mv laf-fabric /home/shebanq/laf-fabric*
   3. cd into \*/laf-fabric/dist
   4. unpack the tar file (*tar xvf \*.tar*)
   5. cd into the unpacked tar file
   6. run: *sudo python3 setup.py install*
   7. cd into /home/shebanq/laf-fabric/
   8. run: *python3 lf-gallery.py fulls*
   9. In case it is needed change config of laf-fabric-data to make data fit the right location. In my case "data\_dir = /home/shebanq/laf-fabric-data
7. Create a notebook folder:
   1. in my case: *mkdir /home/shebanq/laf-notebooks*
   2. move into your notebook folder and run: *ipython3 notebook* (see for remote access next installation point)
   3. you can now create notebooks and run notebooks of others (after you have downloaded them in to the folder)
8. Remote access to ipython3 notebook web interface: This is optional but makes sense especially when you run a headless server
   1. install the nbserver by running the command: *ipython3 profile create nbserver*
   2. make the following changes to your nbserver configuration file by running:

*nano ~/.ipython/profile\_nbserver/ipython\_notebook\_config.py*

add the following information:

c = get\_config()

*# Kernel config*

c.IPKernelApp.pylab = 'inline' *# if you want plotting support always*

*# Notebook config*

c.NotebookApp.certfile = u'/absolute/path/to/your/certificate/mycert.pem'

c.NotebookApp.ip = '\*'

c.NotebookApp.open\_browser = False

c.NotebookApp.password = u'sha1:bcd259ccf...[your hashed password here]'

*# It is a good idea to put it on a known, fixed port*

c.NotebookApp.port = 9999

You can then start the notebook and access it later by pointing your browser to your.guest.ip:9999 with *ipython3 notebook --profile=nbserver*.

In case you want to access the web interface of your notebooks within the guest machine (thus without remotely accessing the notebooks) just run the command: *ipython3 notebook* in your notebook folder.

1. Networking:

You can change the network address of your eth interfaces with: *sudo nano /etc/network/interfaces*

In my case I have two network adapters (one runs NAT and the other one is Host access only) and I use the following configuration:

1# This file describes the network interfaces available on your system

2# and how to activate them. For more information, see interfaces(5).

3# The loopback network interface

4 auto lo

5 iface lo inet loopback

6 # The primary network interface

7 auto eth0 eth1

8 iface eth0 inet dhcp

9 iface eth1 inet dhcp

10# iface eth1 inet static

11# address 192.168.56.2

12# netmask 255.255.255.0

13# network 192.168.56.0

14# broadcast 192.168.56.255

15# gateway 192.168.56.1

16# dns-nameservers 8.8.8.8 8.8.4.4

I hashtagged away a static eth1 configuration in case I quickly want to change settings at a later time. After having adjusted your networking configuration you can either reboot the vm or run *sudo service networking restart.*

**Notice:** If you get an error like "failed to start unit user@1000.service" on the shell after login, then you hit this [bug](https://bugs.launchpad.net/ubuntu/+source/systemd-shim/+bug/1359439) in the current Ubuntu version. This message is uncritical and your server will work normally, so don’t worry, you can proceed with your server setup. For more information about this bug see <https://bugs.launchpad.net/ubuntu/+source/systemd-shim/+bug/1359439> .