

Lab | Book

Manuel d'Installation d'un environnement de développement



Fondation
Mérieux

Table des matières

Introduction	3
Procédure d'Installation	3
Etape 1 : Installer les dépendances : Git ; Make ; podman ; mysql	3
Etape 2 : Cloner le dépôt distant de LabBook en local	4
Etape 3 : Modifier la configuration du fichier labbook.conf	4
Etape 4 : Modifier la configuration du Mysql	5
Etape 5 : Configurer la base de données.....	5
Etape 6 : Lancer les commande make	7
Génération d'une mise à jour LabBook	8
Génération d'un iso LabBook.....	12



Introduction

Ce manuel est dédié aux informaticiens, développeurs, experts voulant faire des modifications avec le code source de LabBook qui est accessible gratuitement par le lien : https://github.com/fondationmerieux/labbook_python

Il décrit le processus d'installation du LabBook en mode développement pour pouvoir ajouter et /ou modifier le code source en se basant sur le tuto readme disponible sur le GitHub de la fondation Mérieux.

Procédure d'Installation

Pour pouvoir installer une machine de développement LabBook il faut disposer d'un ordinateur avec Ubuntu 22.04, 4go de RAM minimum (idéalement 8go) et 75Go de stockage disque. Les différentes étapes d'installation sont décrites ci-dessous.

Etape 1 : Installer les dépendances : Git ; Make ; podman ; mysql

- `sudo apt-get upgrade`
- `sudo apt-get update`
- `sudo apt-get install make`
- `sudo apt-get install git`
- `sudo apt-get install -y podman`
- `sudo apt-get install mysql-server`
- `sudo systemctl start mysql.service`

Pour vérifier si les dépendances sont bien installées :

- `make --version`



- *git --version*
- *mysql --version*
- *podman --version*

Etape 2 : Cloner le dépôt distant de LabBook en local

Pour cloner le dépôt distant de LabBook en local, veuillez ouvrir un terminal et saisir la commande suivante :

git clone https://github.com/fondationmerieux/labbook_python.git

Etape 3 : Modifier la configuration du fichier labbook.conf

Il y'a une copie du fichier labbook.conf au niveau du répertoire doc (dans le dossier labbook_python obtenu après clonage). Il faut copier ce fichier aux emplacements suivants :

- `$HOME/.config/labbook.conf`
- `$HOME/labbook.conf`

Puis le modifier en ajoutant les informations suivantes :

`LABBOOK_DB_USER=labbook`

`LABBOOK_DB_PWD=labbook`

`LABBOOK_DB_NAME=SIGL`

`LABBOOK_DB_HOST=192.168.255.209`

`LABBOOK_DEBUG=1`

`LABBOOK_TEST_OK=`

`LABBOOK_TEST_KO=`

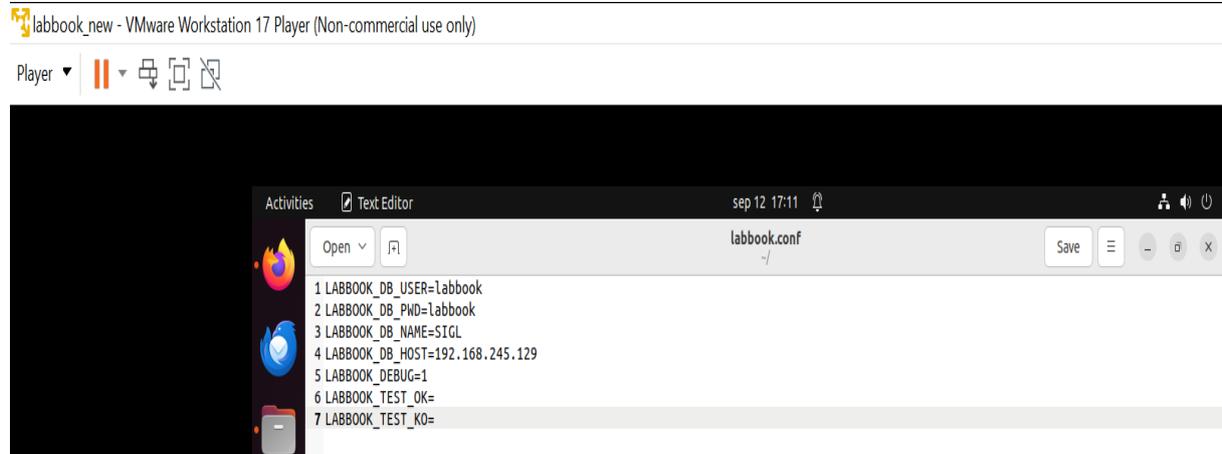
Remarque :

Remplacer `192.168.255.209` par l'adresse ip de la machine.



Pour retrouver son @ ip :

Exécuter sur terminal la commande *ifconfig* . Si *ifconfig* est introuvable, il faut l'installer avec la commande *sudo apt-get install net-tools* .



Etape 4 : Modifier la configuration du Mysql

Si vous avez besoin de modifier les configurations par défaut de Mysql, voici les étapes à suivre :

- Exécuter sur le terminal la commande : *cd /etc/mysql/mysql.conf.d* Puis *sudo nano mysqld.cnf*

Etape 5 : Configurer la base de données

Pour pouvoir configurer la base de données on se connecte en tant que l'utilisateur root :

- *sudo mysql -u root -h localhost*
- *mysql> CREATE USER labbook@'192.168.255.209' IDENTIFIED BY 'labbook';*
- *mysql> GRANT ALL PRIVILEGES ON *.* TO labbook@'192.168.255.209';*
- *mysql> CREATE USER labbook@'localhost' IDENTIFIED BY 'labbook';*
- *mysql> GRANT ALL PRIVILEGES ON *.* TO labbook@'localhost';*
- *mysql> FLUSH PRIVILEGES ;*



- `mysql> quit`

Redémarrer Mysql avec la commande :

`sudo systemctl restart mysql`

```
fatou@fatou-virtual-machine: /etc/mysql/mysql.conf.d
8b622f2175fb3f09238d6edf6d637edc28304728d3cabdb4938a7adae08678d9
fatou@fatou-virtual-machine:~/labbook_python$ sudo mysql -u root -h localhost
[sudo] password for fatou:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.39-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

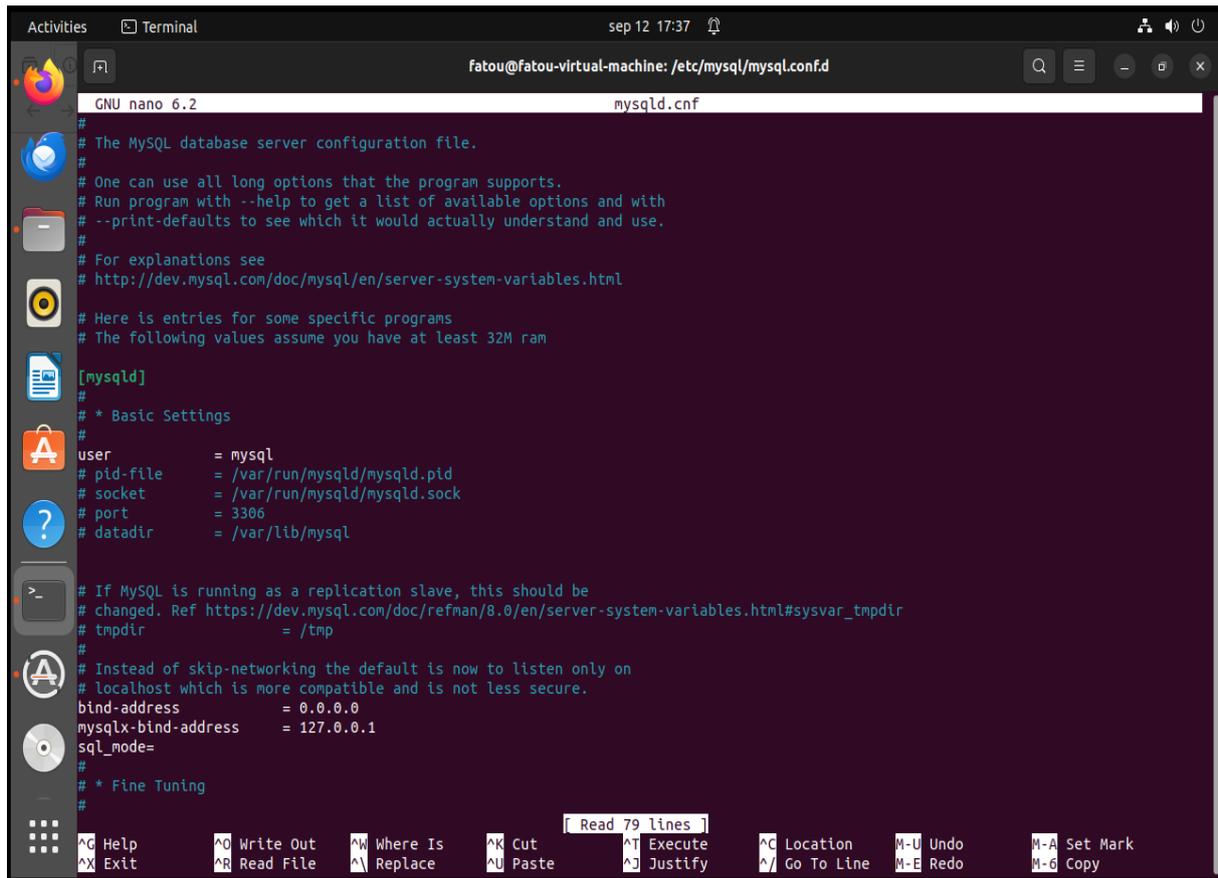
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select user,host from mysql.user;
+-----+-----+
| user          | host          |
+-----+-----+
| labbook       | 192.168.245.129 |
| debian-sys-maint | localhost    |
| labbook       | localhost    |
| mysql.infoschema | localhost    |
| mysql.session | localhost    |
| mysql.sys     | localhost    |
| root          | localhost    |
+-----+-----+
7 rows in set (0,01 sec)

mysql> quit
Bye
fatou@fatou-virtual-machine:~/labbook_python$ cd /etc/mysql/mysqlld.conf
```





```

Activities Terminal sep 12 17:37 fatou@fatou-virtual-machine: /etc/mysql/mysql.conf.d
GNU nano 6.2 mysqld.cnf
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram
[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket             = /var/run/mysqld/mysqld.sock
# port               = 3306
# datadir            = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir             = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address        = 0.0.0.0
mysqlx-bind-address = 127.0.0.1
sql_mode=
#
# * Fine Tuning
#
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^M Replace    ^U Paste      ^J Justify
^C Location  ^U Undo      ^M Set Mark
^_ Go To Line ^B Redo      ^G Copy

```

Etape 6 : Lancer les commande make

Une fois toutes les configurations faites, les commandes make sont exécutées pour construire l'image, tester la bdd ,ajouter des données par défaut à la base de données, lancer LabBook sur navigateur et aussi l'arrêter .

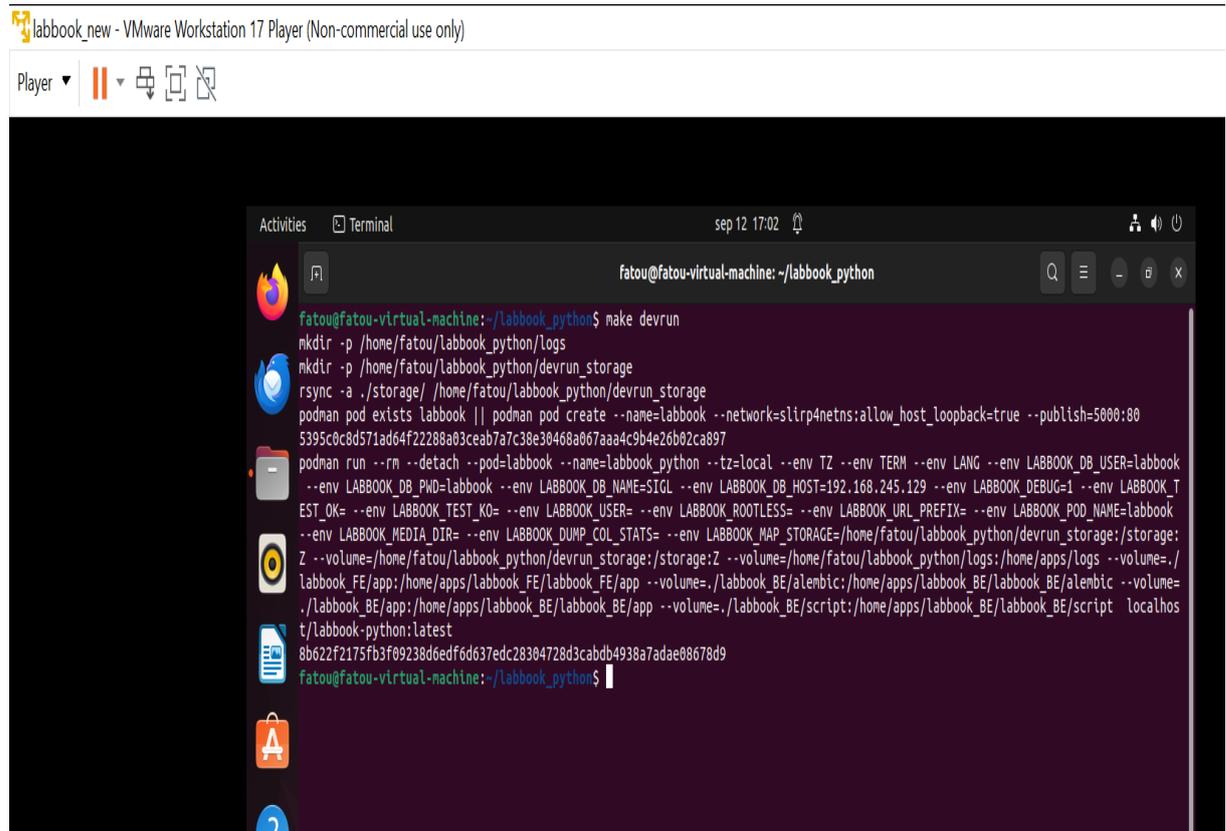
Pour ce faire il faut ouvrir un terminal et se positionner sur le dossier labbook_python en utilisant la commande cd ou bien sur l'explorateur de fichiers, ouvrir le dossier labbook_python, faire un clic droit et clic sur « ouvrir dans un terminal ».

A partir de là, lancer les commandes suivantes :

- *make devbuild* → il permet de construire l'image LabBook.C'est la partie la plus longue.
- *make dbtest* → il permet de tester la connexion entre l'application LabBook et la bdd SIGL



- `make dbinit` → cette commande permet d'initialiser la bdd SIGL en remplissant les tables avec des données tests.
- `make devrun` → cette commande va lancer LabBook sur le navigateur à l'adresse `localhost :5000/`.
- `make devstop` → permet quant à lui de stopper l'instance LabBook qui s'exécute.
- `make devclean` → **Attention !** cette commande va supprimer l'image créée auparavant avec le `devbuild`.



```

fatou@fatou-virtual-machine: ~/labbook_python
fatou@fatou-virtual-machine:~/labbook_python$ make devrun
mkdir -p /home/fatou/labbook_python/logs
mkdir -p /home/fatou/labbook_python/devrun_storage
rsync -a ./storage/ /home/fatou/labbook_python/devrun_storage
podman pod exists labbook || podman pod create --name=labbook --network=slirp4netns:allow_host_loopback=true --publish=5000:80
5395c0c8d571ad64f22288a03ceab7a7c38e30468a067aaa4c9b4e26b02ca897
podman run --rm --detach --pod=labbook --name=labbook_python --tz=Local --env TZ --env TERM --env LANG --env LABBOOK_DB_USER=labbook
--env LABBOOK_DB_PWD=labbook --env LABBOOK_DB_NAME=SIGL --env LABBOOK_DB_HOST=192.168.245.129 --env LABBOOK_DEBUG=1 --env LABBOOK_T
EST_OK= --env LABBOOK_TEST_KO= --env LABBOOK_USER= --env LABBOOK_ROOTLESS= --env LABBOOK_URL_PREFIX= --env LABBOOK_POD_NAME=labbook
--env LABBOOK_MEDIA_DIR= --env LABBOOK_DUMP_COL_STATS= --env LABBOOK_MAP_STORAGE=/home/fatou/labbook_python/devrun_storage:/storage:
Z --volume=/home/fatou/labbook_python/devrun_storage:/storage:Z --volume=/home/fatou/labbook_python/logs:/home/apps/logs --volume=/
labbook_FE/app:/home/apps/labbook_FE/Labbook_FE/app --volume=/labbook_BE/alemnic:/home/apps/labbook_BE/Labbook_BE/alemnic --volume=
./labbook_BE/app:/home/apps/labbook_BE/labbook_BE/app --volume=/labbook_BE/script:/home/apps/labbook_BE/labbook_BE/script localhos
t/labbook-python:latest
8b622f2175fb3f09238d6edf6d637edc28304728d3cabdb4938a7adae08678d9
fatou@fatou-virtual-machine:~/labbook_python$

```

Génération d'une mise à jour LabBook

Pour générer un fichier de mise à jour LabBook, il faut modifier et enregistrer le code de la partie concernée et bien vérifier qu'il est fonctionnel. Les modifications apportées au code sont visibles directement sur le navigateur il n'est donc pas nécessaire de recréer l'image ou de redémarrer l'instance LabBook.



Lancer les 2 commandes suivantes :

- make build

```
fatou@fatou-virtual-machine: ~/labbook_python
STEP 26/37: COPY apache/apache.sh /home/apps/apache/
--> Using cache acd53f7ce7ee57d276a60cf104cd0238b896cda7f0bf20b0dd1dce251e43a651
--> acd53f7ce7e
STEP 27/37: COPY labbook_FE/Pipfile /home/apps/labbook_FE/labbook_FE
--> Using cache a3153219610680c4b219ab9753e9e0f76840213beb2fa0d5583cce5806200d70
--> a3153219610
STEP 28/37: WORKDIR /home/apps/labbook_FE/labbook_FE
--> Using cache 1f140d3715bce1b6eb48899e2ebb71e227b2698176e739ae27ae365ad10c618d
--> 1f140d3715b
STEP 29/37: RUN python3 -m venv venv
--> Using cache 7786e643fe243ebe87c5b7947f418a9db13225ed37dd0ea81b74bdb4305fd086
--> 7786e643fe2
STEP 30/37: RUN source venv/bin/activate && pip install --upgrade pip && pip install pipenv && pipenv install
--> Using cache 713eea57c8c750963a3cdf992065f40ad78f86e8ac26bfe376c8fbb94bb82cb
--> 713eea57c8c
STEP 31/37: COPY labbook_BE/Pipfile /home/apps/labbook_BE/labbook_BE
--> Using cache 7952b2d3fd6e2e06db7dd001d460479d4033166ef0ed3aaace7c2e117e83824
--> 7952b2d3fd6
STEP 32/37: WORKDIR /home/apps/labbook_BE/labbook_BE
--> Using cache 1ea5a79bb4fcf12b96f9c7856cf032ff6cabf51696723bb9466b924e1eee1e01
--> 1ea5a79bb4f
STEP 33/37: RUN python3 -m venv venv
--> Using cache 2c965136dac91587438c89e44785c72b01f0b5d446a5ccc570af3d43cb4a0df2
--> 2c965136dac
STEP 34/37: RUN source venv/bin/activate && pip install --upgrade pip && pip install pipenv && pipenv install
--> Using cache 807ad04265294193b082aefcebc7708f82f5ecacc63de136f74d1eb2f4320e
--> 807ad042652
STEP 35/37: COPY labbook_FE /home/apps/labbook_FE/labbook_FE
--> Using cache 91389dfccc628c1d294232f7303e4b8dcc1f67cfddf3b742a991c3acd94cf984
--> 91389dfccc6
STEP 36/37: COPY labbook_BE /home/apps/labbook_BE/labbook_BE
--> Using cache aa0d59dc1e7b94b8680b354408a74ad15556dda3075d88011f2455a3e49ebe
--> aa0d59dc1e7
STEP 37/37: CMD ["supervisord", "-c", "/home/supervisor/etc/supervisor.conf", "--pidfile", "/home/supervisor/tmp/supervisor.
id", "--user", "root"]
--> Using cache ca429519cec5ee1717d8190cbb65a2d47abfd209972cc6d2ca3e1b026b03a3e9
COMMIT localhost/labbook-python:3.4.11
--> ca429519cec
Successfully tagged localhost/labbook-python:3.4.11
Successfully tagged localhost/labbook-python:latest
ca429519cec5ee1717d8190cbb65a2d47abfd209972cc6d2ca3e1b026b03a3e9
git checkout master
M labbook_BE/default_settings.py
M labbook_FE/app/templates/login.html
Basculement sur la branche 'master'
Votre branche est à jour avec 'origin/master'.
fatou@fatou-virtual-machine:~/labbook_python$
```

- make save



```
fatou@fatou-virtual-machine:~/labbook_python$ make save
rm -f /tmp/labbook-python-3.4.11.tar /tmp/labbook-python-3.4.11.tar.xz
podman save --output=/tmp/labbook-python-3.4.11.tar localhost/labbook-python:3.4.11
Copying blob b97f5cdd369d done
Copying blob b06b0612ddd6 done
Copying blob 340da03852e4 done
Copying blob 676017fb7fcc done
Copying blob d0651719be2a done
Copying blob 2c3b5fc2c73d done
Copying blob e34f5f48a01f done
Copying blob 2a16af92147a done
Copying blob 4c29ed538b59 done
Copying blob fdec58bae23e done
Copying blob 938eb785a19a done
Copying blob 950b486f20d8 done
Copying blob 27aab195b997 done
Copying blob 99a78bd0ae3c done
Copying blob 7a979e6c7af6 done
Copying blob ef4b1d9f570f done
Copying blob 8eb8794d4153 done
Copying blob 31ff1048d7be done
Copying blob 78245e876e3d done
Copying blob 01eda8a4fc0c done
Copying blob 86e2bbd166b9 done
Copying blob feb808dac97a done
Copying blob 52cf0760eb2f done
Copying blob 170231deae53 done
Copying blob 710f0866ecca done
Copying blob 7ed50d17e94c done
Copying blob 548da3bff365 done
Copying blob 8894458bae2e done
Copying blob 4a9212c7e04c done
Copying blob c6ceb499bb49 done
Copying blob dd914ed3073f done
Copying blob 8bcb71b86436 done
Copying blob 021926e6d74d done
Copying blob 4016689d3983 done
Copying config ca429519ce done
Writing manifest to image destination
Storing signatures
xz --keep /tmp/labbook-python-3.4.11.tar
(cd /tmp && \
  md5sum labbook-python-3.4.11.tar > labbook-python-3.4.11.tar.md5sum && \
  md5sum labbook-python-3.4.11.tar.xz > labbook-python-3.4.11.tar.xz.md5sum)
fatou@fatou-virtual-machine:~/labbook_python$
```

Remarque :

Si après avoir exécuté la commande `make build` vous avez une erreur concernant le tag il faut essayer les commandes :

- `git fetch`



```
fatou@fatou-virtual-machine:~/labbook_python$ git fetch
Depuis https://github.com/fondationmerieux/labbook_python
* [nouvelle étiquette] v3.3.0 -> v3.3.0
* [nouvelle étiquette] v3.3.1 -> v3.3.1
* [nouvelle étiquette] v3.3.10 -> v3.3.10
* [nouvelle étiquette] v3.3.11 -> v3.3.11
* [nouvelle étiquette] v3.3.12 -> v3.3.12
* [nouvelle étiquette] v3.3.13 -> v3.3.13
* [nouvelle étiquette] v3.3.14 -> v3.3.14
* [nouvelle étiquette] v3.3.15 -> v3.3.15
* [nouvelle étiquette] v3.3.2 -> v3.3.2
* [nouvelle étiquette] v3.3.3 -> v3.3.3
* [nouvelle étiquette] v3.3.4 -> v3.3.4
* [nouvelle étiquette] v3.3.5 -> v3.3.5
* [nouvelle étiquette] v3.3.6 -> v3.3.6
* [nouvelle étiquette] v3.3.7 -> v3.3.7
* [nouvelle étiquette] v3.3.8 -> v3.3.8
* [nouvelle étiquette] v3.3.9 -> v3.3.9
* [nouvelle étiquette] v3.4.0 -> v3.4.0
* [nouvelle étiquette] v3.4.1 -> v3.4.1
* [nouvelle étiquette] v3.4.10 -> v3.4.10
* [nouvelle étiquette] v3.4.11 -> v3.4.11
* [nouvelle étiquette] v3.4.2 -> v3.4.2
* [nouvelle étiquette] v3.4.3 -> v3.4.3
* [nouvelle étiquette] v3.4.4 -> v3.4.4
* [nouvelle étiquette] v3.4.5 -> v3.4.5
* [nouvelle étiquette] v3.4.6 -> v3.4.6
* [nouvelle étiquette] v3.4.7 -> v3.4.7
* [nouvelle étiquette] v3.4.8 -> v3.4.8
* [nouvelle étiquette] v3.4.9 -> v3.4.9
fatou@fatou-virtual-machine:~/labbook_python$
```

- *git log*

```
fatou@fatou-virtual-machine:~/labbook_python
* [nouvelle étiquette] v3.4.6 -> v3.4.6
* [nouvelle étiquette] v3.4.7 -> v3.4.7
* [nouvelle étiquette] v3.4.8 -> v3.4.8
* [nouvelle étiquette] v3.4.9 -> v3.4.9
fatou@fatou-virtual-machine:~/labbook_python$ git log
commit 5e2871a0fa3dfe32152493ffee87a03c4293c9f4 (HEAD -> master, tag: v3.4.11, origin/master, origin/HEAD)
Author: Alexandre CHARLES <ac@aegle.fr>
Date: Sun Sep 8 15:44:41 2024 +0200

    Release 3.4.11

commit d7976bf92a0a651e7859c4dfda938afad2f07bb1 (tag: v3.4.10)
Author: Alexandre CHARLES <ac@aegle.fr>
Date: Mon Jul 29 08:39:46 2024 +0200

    Release 3.4.10

commit 3ac5d9bb8e1903f76c80253a2ba329cbbba45bfe (tag: v3.4.9)
Author: Alexandre CHARLES <ac@aegle.fr>
Date: Wed Jul 10 11:39:58 2024 +0200

    Release 3.4.9

commit 0878f86eee000a364db1fcbe279a62d33df06e84 (tag: v3.4.8)
Author: Alexandre CHARLES <ac@aegle.fr>
Date: Wed Jun 26 14:40:37 2024 +0200

    Release 3.4.8

commit 768098b7143fdf5f7a96c0d23ec3d2adf26ee26f (tag: v3.4.7)
Author: Alexandre CHARLES <ac@aegle.fr>
Date: Mon Jun 24 17:48:34 2024 +0200

    Release 3.4.7

commit c627aa98bde948e9ae57b07b070254bdf7fa62b8
Author: Alexandre CHARLES <ac@aegle.fr>
Date: Mon Jun 24 17:07:41 2024 +0200

    Release 3.4.7

commit 5d672fa5d7ee2ab107cd7a58b87865826f891445 (tag: v3.4.6)
Author: Alexandre CHARLES <ac@aegle.fr>
Date: Mon Jun 17 12:13:48 2024 +0200

    Added checkbox ana_ast
```

Puis refaire *make build*.



Génération d'un iso LabBook

Pour générer un iso LabBook il faut une certaine configuration et quelques fichiers nécessaires.

Prérequis :

- Être sur une machine Ubuntu 22.04
- Avoir un iso Ubuntu 22.04 LTS
- Installer le logiciel générateur d'iso CUBIC.
- Sur la machine hôte : désactiver momentanément mysql avec la commande `sudo systemctl stop mysql`
- Puis `sudo systemctl status mysql` pour vérifier
- Disposer des fichiers ci-dessous ([Télécharger en suivant ce lien](#)) :
 - Dossier : *Manuel*
 - Fichiers tar :
 1. *formation-FMX-LabBook.tar.xz*
 2. *labbook-connect-1.0.0.tar.xz*
 3. *labbook-pod.tar.xz*
 4. *labbook-python-3.4.10.tar.xz*
 - Autres fichiers :
 5. *exec_update_labbook.sh*
 6. *Export_labbook3*
 7. *grub.cfg*
 8. *lab-book.org.desktop*
 9. *Labbook*
 10. *Labbook.desktop*
 11. *labbook.service*
 12. *labbook-elearning.desktop*
 13. *labbook_postinstall.sh*



14. *MysqlLabBook.tgz*
15. *README.md*
16. *rustdesk-1.2.3-2-x86_64.deb*
17. *sources.list*
18. *sudoers_labbook*
19. *ubuntu.seed*
20. *update_labbook.sh*
21. *user*

➤ Images :

1. *labbook.png*
2. *labbook-screen-background.png*
3. *ubuntu-wallpapers.xml*

Utilité des fichiers clés :

- *labbook.desktop*, *labbook-elearning.desktop*, *lab-book.org.desktop* : des raccourcis de lancements
- *labbook.png*, *labbook-screen-background.png*, *ubuntu-wallpapers.xml* : des images labbook
- *labbook.service* : permet de démarrer les services LB
- *sudoers_labbook*
- *export_labbook3* : export de la BDD MySQL. N'est présent que pour info, la base étant en fait déjà créée pour des raisons de performance au démarrage
- Vérifier que ces fichiers *labbook_postinstall.sh*, *create.sh* et *labbook* ont les permissions d'exécution sinon accorder les droits avec la commande : *sudo chmod +x /home/create.sh*

On peut retrouver ces fichiers dans le répertoire */home* d'un Labbook déjà installé : *formation-FMX-LabBook.tar.xz*, *MysqlLabBook.tgz*, répertoire Manuel, conteneurs compressés *labbook-connect-1.0.0.tar.xz*, *labbook-python-3.4.10.tar.xz* *labbook-pod.tar.xz* etc...



Etapes :

- 1) Lancer cubic et sélectionner le dossier contenant déjà tous les fichiers précités et dans lequel sera généré l'iso
 - 2) cliquer sur Next et choisir l'iso ubuntu 22.04
 - 3) personnaliser (sur les champs à droite) le nom, l'id et la version du futur iso à créer puis cliquer sur Next
 - 4) on est redirigé vers un terminal chroot cubic ,
 - 5) ouvrir un terminal sur la machine hôte et se positionner sur le dossier de création de l'iso et lancer la commande `cp /create.sh /custom-root/home`
- Tjrs sur la machine physique lancer les commandes :
- ```
cp ubuntu.seed /custom-disk/preseed
```
- ```
cp grub.cfg /custom-disk/boot/grub.
```
- 6) revenir sur le terminal de cubic et lancer la commande `/home/create.sh | tee /home/create.log` pour exécuter toutes les commandes permettant de générer un iso labbook et enregistrer la sortie dans un fichier create.log.
 - 7) Cliquer sur Next, Next. Pour le choix du zip , sélectionner xz puis next , à la fin cliquer sur close.

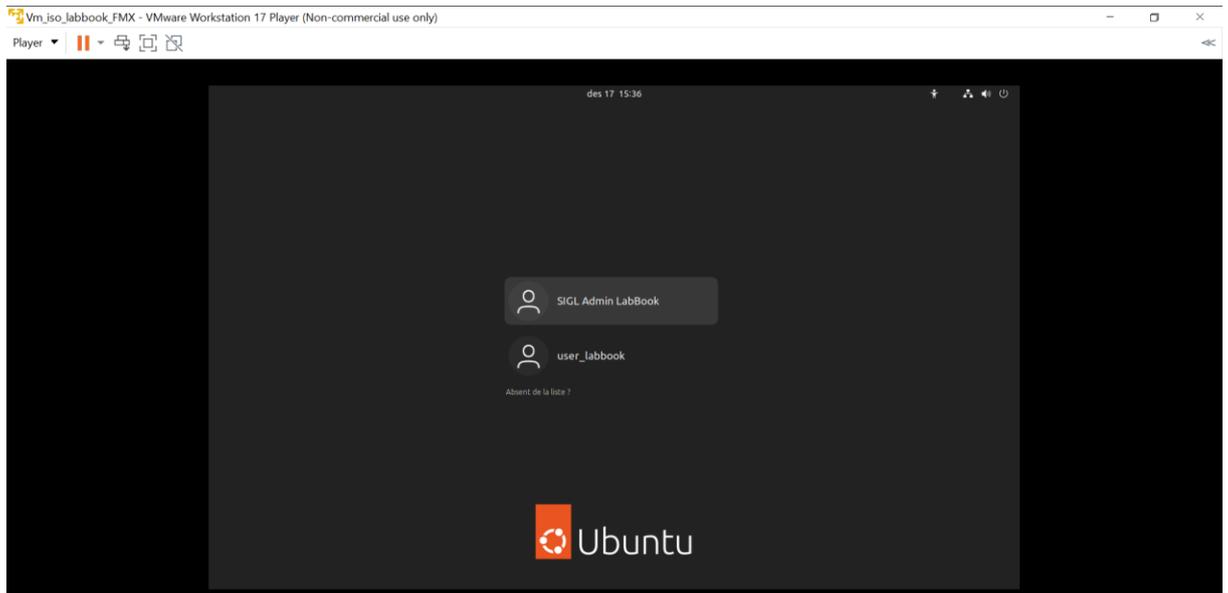
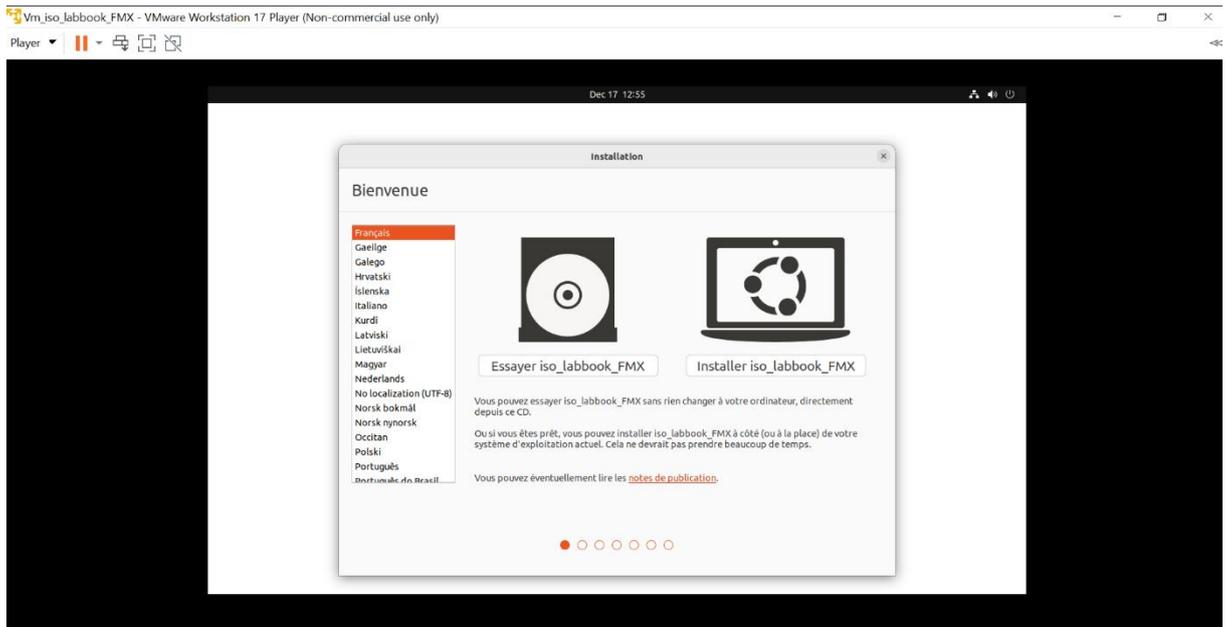
Test :

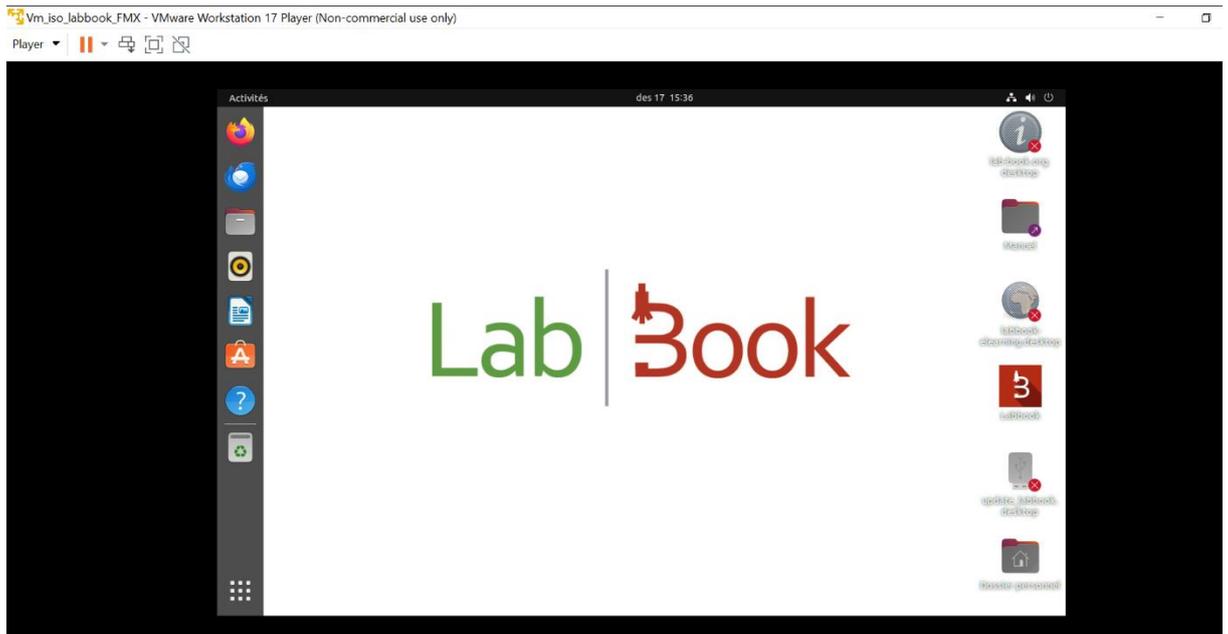
- 8) Pour tester l'iso généré :

Le plus simple c'est de le tester directement sur VMWare en créant une machine virtuelle

Sinon aussi il est possible de booter une clé avec Rufus et faire l'installation sur une machine physique.







Dans les 2 cas une fois l'installation terminée et le redémarrage effectué on peut lancer directement labbook en faisant :

Clique-droit sur l'icône labbook -> autoriser l'exécution puis double-clique.

Remarque :

Pour les fichiers .tar.xz il faut bien veiller à avoir les bons sans double compression c'est à dire que si le fichier est décompressé on doit tomber directement sur les fichiers.

