

Manuel Utilisateur du projet de Data science

Kelun Chai
Oussama Bouzaouit
Djaber Solimani
Nogaye Gning



Table des matières

1 - Configuration de l'environnement	3
1.1 Git	3
1.2 Docker Engine & Docker-compose	3
1.2.1 Installer Docker Engine sur Ubuntu	3
1.2.2 Installer Docker Compose sur Ubuntu	4
1.3 Libsasl	4
1.4 Python	4
ou	4
1.5 Android Studio	4
2- Démarrer la Hive et le Server	5
3 - Démarrer l'application Android	7
4 - Q&A	11

1 - Configuration de l'environnement

Nous recommandons d'exécuter ce projet sur système Linux.

1.1 Git

Tout d'abord, vous devrez installer git sur votre système afin de télécharger le code à partir de notre dépôt. Si git n'est pas inclus dans votre système, veuillez vous référer au guide d'installation suivant.

Déterminez sur quelle distribution Linux votre système est basé. Voir [List of Linux distributions](#) pour une liste. La plupart des systèmes Linux - y compris Ubuntu - sont basés sur Debian.

Systèmes linux basés sur Debian

[Ouvrez une fenêtre de terminal](#). Copiez et collez ce qui suit dans la fenêtre du terminal et appuyez sur `Retour`. Vous serez peut-être invité à entrer votre mot de passe.

```
sudo apt update
sudo apt upgrade
sudo apt install git
```

Vous pouvez utiliser Git maintenant.

Systèmes linux basés sur Red-Hat

Ouvrez une fenêtre de terminal. Copiez et collez ce qui suit dans la fenêtre du terminal et appuyez sur `Retour`. Vous serez peut-être invité à entrer votre mot de passe.

```
sudo yum upgrade
sudo yum install git
```

Vous pouvez utiliser Git maintenant.

1.2 Docker Engine & Docker-compose

1.2.1 Installer Docker Engine sur Ubuntu

Pour installer Docker Engine sur Ubuntu, voir le guide d'installation officiel:

<https://docs.docker.com/engine/install/ubuntu/>

Après une installation réussie, il y a quelques réglages de suivi à effectuer :

<https://docs.docker.com/engine/install/linux-postinstall/>

1.2.2 Installer Docker Compose sur Ubuntu

Pour installer Docker Compose sur Ubuntu, consultez le guide d'installation officiel : <https://docs.docker.com/compose/install/>

1.3 Libsasl

Ouvrez une fenêtre de terminal. Copiez et collez ce qui suit dans la fenêtre du terminal et appuyez sur `Retour`. Vous serez peut-être invité à entrer votre mot de passe.

```
sudo apt-get install -y libsasl2-modules libsasl2-dev
```

1.4 Python

Ouvrez une fenêtre de terminal. Copiez et collez ce qui suit dans la fenêtre du terminal et appuyez sur `Retour`. Vous serez peut-être invité à entrer votre mot de passe.

```
sudo apt-get update  
sudo apt-get install -y python3-dev
```

Une fois le processus terminé, nous pouvons vérifier la version de Python 3 qui est installée dans le système en tapant:

```
python3 -V
```

Vous recevrez une sortie dans la fenêtre du terminal qui vous indiquera le numéro de version. Le numéro de version peut varier, mais il sera similaire à celui-ci:

```
Python 3.7.6
```

Pour gérer les paquets logiciels pour Python, installons pip :

```
sudo apt-get install -y python3-pip
```

Installez les paquets Python suivants (pip ou pip3)

```
pip install --upgrade pip six thrift_sasl "pyhive[hive]" flask  
fbprophet statsmodels pandas
```

ou

```
pip3 install --upgrade pip six thrift_sasl "pyhive[hive]" flask  
fbprophet statsmodels pandas
```

1.5 Android Studio

Pour installer Android Studio sur Ubuntu, voir le guide d'installation officiel: <https://developer.android.com/studio/install>

2- Démarrer la Hive et le Server

Sélectionnez un dossier et ouvrez le terminal, copiez et collez ce qui suit dans la fenêtre du terminal et appuyez sur Retour.

```
git pull https://github.com/katanarga/dashboard\_coronavirus.git
cd dashboard_coronavirus/server/
docker-compose up -d
```

Vous verrez l'écran suivant, indiquant que la Hive a démarré avec succès.

```
Starting server_namenode_1      ... done
Starting server_presto-coordinator_1 ... done
Starting server_hive-metastore_1 ... done
Starting server_hive-metastore-postgresql_1 ... done
Starting server_hive-server_1  ... done
Starting server_datanode_1     ... done
```

Attendre une dizaine de secondes pour que le port de serveur Hive se lance. Ensuite, exécutez la commande suivante:

```
python3 web_server.py
```

```
Begin prepare.
Prepare finished.
* Serving Flask app "web_server" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

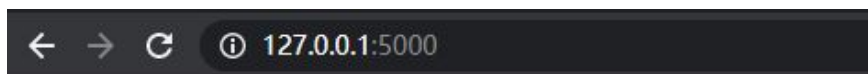
Si vous rencontrez l'erreur suivante, le port de la Hive ne fonctionne pas encore, veuillez attendre une dizaine de secondes et re-exécuter la commande ci-dessus.

```
Begin prepare.
Prepare finished.
Traceback (most recent call last):
  File "web_server.py", line 79, in <module>
    cursor = hive.connect(host='localhost').cursor()
  File "/home/klchai/.local/lib/python3.7/site-packages/pyhive/hive.py", line 94, in connect
    return Connection(*args, **kwargs)
  File "/home/klchai/.local/lib/python3.7/site-packages/pyhive/hive.py", line 192, in __init__
    self._transport.open()
  File "/home/klchai/.local/lib/python3.7/site-packages/thrift_sasl/__init__.py", line 93, in open
    status, payload = self._recv_sasl_message()
  File "/home/klchai/.local/lib/python3.7/site-packages/thrift_sasl/__init__.py", line 112, in _recv_sasl_message
    header = self._trans_read_all(5)
  File "/home/klchai/.local/lib/python3.7/site-packages/thrift_sasl/__init__.py", line 198, in _trans_read_all
    return read_all(sz)
  File "/home/klchai/.local/lib/python3.7/site-packages/thrift/transport/TTransport.py", line 62, in readAll
    chunk = self.read(sz - have)
  File "/home/klchai/.local/lib/python3.7/site-packages/thrift/transport/TSocket.py", line 143, in read
    message='TSocket read 0 bytes')
thrift.transport.TTransport.TTransportException: TSocket read 0 bytes
```

Si vous ne pouvez toujours pas vous connecter, exécutez le code suivant pour fermer l'image du docker et répétez les étapes précédentes.

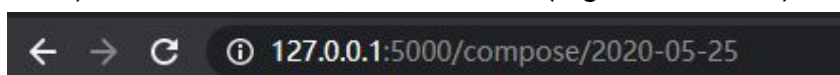
```
docker-compose down
```

Ouvrez votre navigateur système et tapez l'adresse <http://127.0.0.1:5000>, vous devriez voir cette page.



Home page of the web server of the dashboard_coronavirus.

Le serveur fournit les interfaces suivantes que vous pouvez tester. Pour l'interface compose/<jour>, vous devez donner une date (avant le 25 mai 2020) sous la forme année-mois-date (e.g. 2020-05-25).

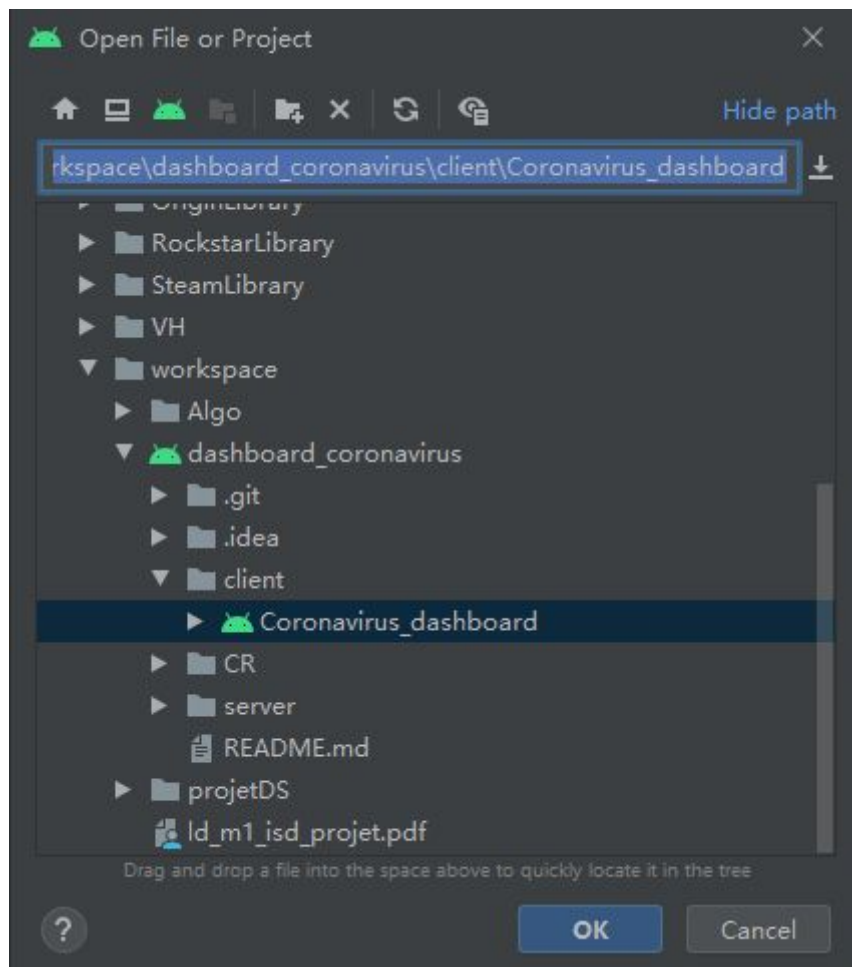


[{"Confirmed":183067,"Deaths":28460,"Recovered":65317}]

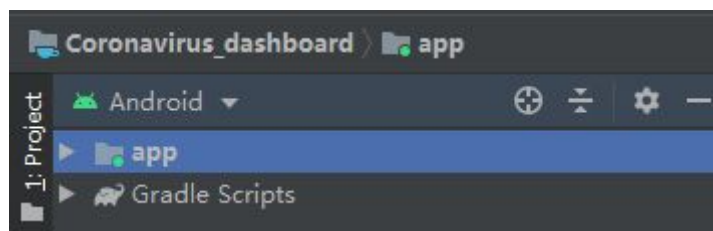
127.0.0.1:5000/deathbyday	Nombre de décès par jour en France (valeur agrégée)
127.0.0.1:5000/confbyday	Nombre de cas confirmés par jour en France (valeur agrégée)
127.0.0.1:5000/recovbyday	Nombre de personnes guéries par jour en France (valeur agrégée)
127.0.0.1:5000/compose/<day>	Situation épidémiologique en France à un jour <day> donné (nombre de cas confirmés, guéries, décès)
127.0.0.1:5000/pred_conf	Prévision du nombre de cas confirmés en France dans les 30 prochains jours - Modèle Prophet
127.0.0.1:5000/pred_conf_arima	Prévision du nombre de cas confirmés en France dans les 30 prochains jours - Modèle Arima
127.0.0.1:5000/pred_death_P	Prévision du nombre de morts en France dans les 30 prochains jours - Modèle Prophet
127.0.0.1:5000/pred_death_A	Prévision du nombre de morts en France dans les 30 prochains jours - Modèle Arima

3 - Démarrer l'application Android

Ouvrez Android Studio et sélectionnez le fichier du projet (dans le dossier que vous avez sélectionné pour git clone).



Cliquez sur la fenêtre de gauche et sélectionnez l'application:



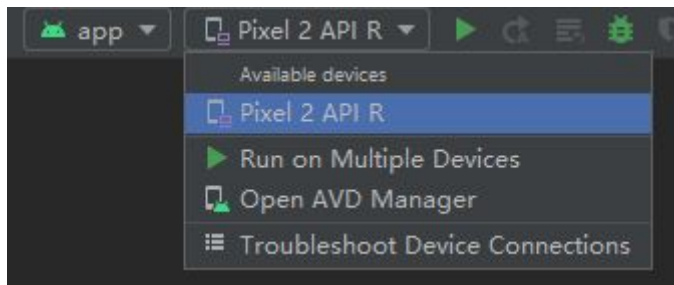
Cliquez sur la première icône dans la barre d'outils en haut de la fenêtre (en forme de marteau)



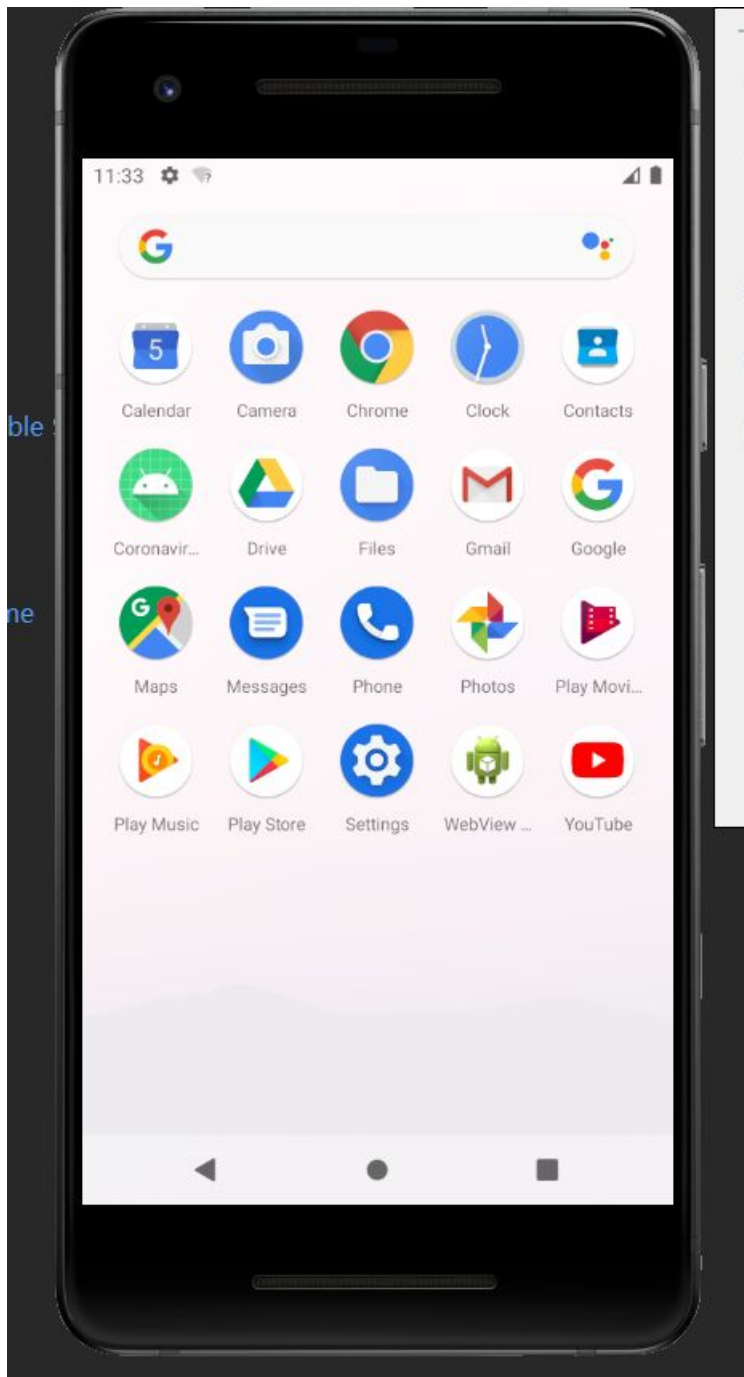
Ensuite, créez votre machine virtuelle Android selon ce document officiel:

<https://developer.android.com/studio/run/managing-avds>

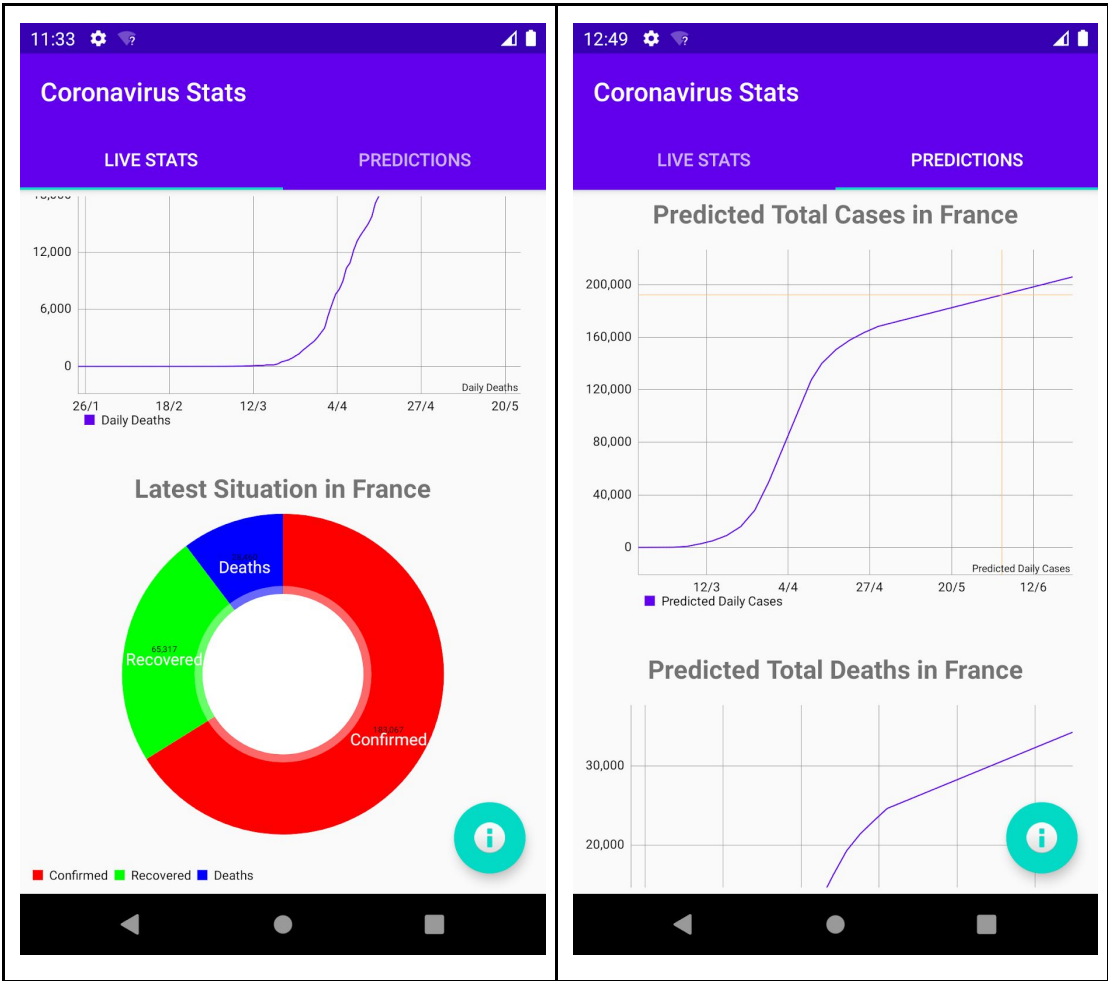
Sélectionnez votre machine virtuelle Android et cliquez sur le triangle en haut à droite (bouton de démarrage).

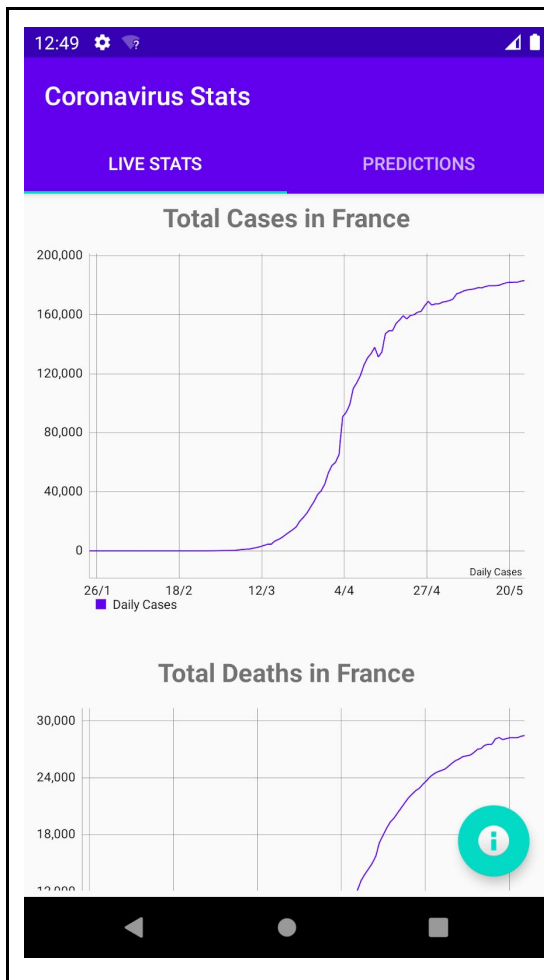


Vous verrez l'écran suivant.



Cliquez sur le programme coronavirus.





Vous pouvez actualiser les données en cliquant sur le bouton dans le coin inférieur droit.

Si votre fenêtre n'affiche pas le graphique, ne vous inquiétez pas. Dans la fenêtre du serveur, vous pouvez voir que le serveur renvoie les données pertinentes.

```

klchai@MSI: /mnt/d/workspace, x + v
Initial log joint probability = -5.62705
Iter   log prob   ||dx||   ||grad||   alpha   alpha0   # evals   Notes
 99     348.065     0.0196404   678.176     1         1        123
Iter   log prob   ||dx||   ||grad||   alpha   alpha0   # evals   Notes
199     397.7      0.000706356   436.402   0.1726    0.5846    243
Iter   log prob   ||dx||   ||grad||   alpha   alpha0   # evals   Notes
299     409.709     0.0118165   640.984     1         1        369
Iter   log prob   ||dx||   ||grad||   alpha   alpha0   # evals   Notes
399     414.351     0.00112003   112.154   0.4799    1        485
Iter   log prob   ||dx||   ||grad||   alpha   alpha0   # evals   Notes
499     416.903     0.0128451   372.694     1         1        607
Iter   log prob   ||dx||   ||grad||   alpha   alpha0   # evals   Notes
599     419.726     0.00116615   227.75     1         1        721
Iter   log prob   ||dx||   ||grad||   alpha   alpha0   # evals   Notes
699     421.918     0.00231131   205.308     1         1        844
Iter   log prob   ||dx||   ||grad||   alpha   alpha0   # evals   Notes
799     425.619     0.0045631   403.325     1         1        965
Iter   log prob   ||dx||   ||grad||   alpha   alpha0   # evals   Notes
827     426.238     3.18046e-05   157.736   3.128e-07   0.001    1036   LS failed, Hessian reset
899     427.246     0.00216999   124.722     1.746     0.1746    1134
Iter   log prob   ||dx||   ||grad||   alpha   alpha0   # evals   Notes
908     427.286     2.42648e-05   130.589   2.106e-07   0.001    1196   LS failed, Hessian reset
999     427.538     2.02904e-06   53.4748     1.876     0.6078    1323
Iter   log prob   ||dx||   ||grad||   alpha   alpha0   # evals   Notes
1014    427.538     9.74179e-09   54.5575     0.07282    0.2913    1350
Optimization terminated normally:
Convergence detected: absolute parameter change was below tolerance
127.0.0.1 - - [06/Jun/2020 01:34:45] "GET /pred_death_P HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [06/Jun/2020 01:34:45] "GET /pred_death_P HTTP/1.1" 200 -

```

Cliquez sur le bouton fermer pour quitter la machine virtuelle Android.
Dans la fenêtre du serveur, appuyez sur Ctrl+C pour fermer le serveur.
Utilisez la commande suivante pour arrêter l'image du docker.

```
docker-compose down
```

Vous verrez l'écran suivant.

```
Stopping server_presto-coordinator_1    ... done
Stopping server_hive-server_1           ... done
Stopping server_hive-metastore-postgresql_1 ... done
Stopping server_hive-metastore_1        ... done
Stopping server_namenode_1              ... done
Stopping server_datanode_1               ... done
Removing server_presto-coordinator_1    ... done
Removing server_hive-server_1           ... done
Removing server_hive-metastore-postgresql_1 ... done
Removing server_hive-metastore_1        ... done
Removing server_namenode_1              ... done
Removing server_datanode_1               ... done
Removing network server_default
```

4 - Q&A

Si vous rencontrez des problèmes, veuillez utiliser la fonction github **issue** pour nous poser des questions.