

<https://bit.ly/3czwQTq>

## TP-1

# Docker et Jupyter notebook

## Introduction

Docker est une plate-forme de conteneurs de logiciels indépendante populaire qui vous permet de créer et d'expédier vos applications, ainsi que tous ses environnements, bibliothèques et dépendances dans des conteneurs. Les conteneurs sont portables, vous pouvez donc configurer exactement le même système sur une autre machine en exécutant quelques commandes Docker simples. Grâce à Docker, il est facile de créer, partager et exécuter votre application n'importe où, sans avoir à dépendre de la configuration actuelle du système d'exploitation.

Par exemple, si vous avez un ordinateur portable qui exécute Windows mais que vous devez configurer une application qui ne fonctionne que sous Linux, grâce à Docker, vous n'avez pas besoin d'installer un nouveau système d'exploitation ou de configurer une machine virtuelle. Vous pouvez configurer un conteneur Docker contenant toutes les bibliothèques dont vous avez besoin et le supprimer dès que vous avez terminé votre travail.

Durant ce TP, nous allons configurer un cluster Hadoop à 2 nœuds à l'aide de Docker et exécuter le programme classique Hadoop Word Count pour tester le système.

## 1. Configuration Docker

Si Docker n'est pas déjà installé, vous pouvez l'installer facilement en suivant les instructions de la page d'accueil officielle de Docker.

<https://github.com/docker/toolbox/releases> => version 19.3.0 ( le .exe pour les utilisateur windows)

(Pour aller plus loin)

[https://docs.docker.com/toolbox/toolbox\\_install\\_windows/](https://docs.docker.com/toolbox/toolbox_install_windows/)  
[https://docs.docker.com/toolbox/toolbox\\_install\\_mac/](https://docs.docker.com/toolbox/toolbox_install_mac/)

Pour vérifier la version de votre Docker Engine, Machine et Compose, utilisez les commandes suivantes:

```
$ docker --version
```

```
$ docker-compose --version
```

```
$ docker-machine --version
```

Lancer le terminal docker

```
To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: C:\Program Files\Docker Toolchain.exe env default

      ##
     ## ##
    ## ## ##
   { ~~~~~ }
  { ~~~~~ }
 { ~~~~~ }
{ ~~~~~ }
 { ~~~~~ }
  { ~~~~~ }
   { ~~~~~ }
    ## ## ##
     ## ##
      ##

=====
=====
=====
=====

docker is configured to use the default machine with IP 192.168.99.100
For help getting started, check out the docs at https://docs.docker.com

Start interactive shell
```

Si c'est la première fois que vous exécutez Docker, testez pour vous assurer que tout fonctionne correctement en lançant votre premier serveur Web Dockerisé:

```
$ docker run -d -p 80:80 --name myserver nginx
```

Comme c'est la première fois que vous exécutez cette commande et que l'image n'est pas encore disponible hors connexion, Docker la récupérera de la bibliothèque Docker Hub. Une fois que tout est terminé, visitez <http://ip> (<http://192.168.99.100>) pour afficher la page d'accueil de votre nouveau serveur.

Nous allons utiliser docker pour simuler environnement de développement Python.

<https://hub.docker.com/r/continuumio/anaconda3>

Télécharger l'image :

```
docker pull continuumio/anaconda3 (1G0)
docker pull continuumio/miniconda3 (80Mo)
```

Démarrer le container:

```
docker run -i -t -p 8888:8888 continuumio/anaconda3 /bin/bash -c "/opt/conda/bin/conda
install jupyter -y --quiet && mkdir /opt/notebooks && /opt/conda/bin/jupyter notebook
--notebook-dir=/opt/notebooks --ip='*' --port=8888 --no-browser --allow-root"
```

```
docker run -i -t -p 8888:8888 continuumio/miniconda3 /bin/bash -c "/opt/conda/bin/conda
install jupyter -y --quiet && mkdir /opt/notebooks && /opt/conda/bin/jupyter notebook
--notebook-dir=/opt/notebooks --ip='0.0.0.0' --port=8888 --no-browser --allow-root"
```

<http://192.168.99.100:8888/> ou <http://localhost:8888> ou <http://<DOCKER-MACHINE-IP>:8888>

For miniconda:

`docker ps`

`docker exec -it container_id /bin/bash`

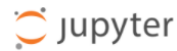
`conda install -c anaconda numpy`

`conda install -c anaconda pandas`

`conda install -c anaconda matplotlib`

```
Preparing transaction: ...working... done
Verifying transaction: ...working... done
Executing transaction: ...working... done
[I 14:31:51.093 NotebookApp] Writing notebook server cookie secret to /root/.local/share/jupyter/runtime/notebook_cookie_secret
[W 14:31:51.424 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[I 14:31:51.559 NotebookApp] JupyterLab extension loaded from /opt/conda/lib/python3.8/site-packages/jupyterlab
[I 14:31:51.560 NotebookApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 14:31:51.565 NotebookApp] Serving notebooks from local directory: /opt/notebooks
[I 14:31:51.566 NotebookApp] The Jupyter Notebook is running at:
[I 14:31:51.566 NotebookApp] http://fc8a1394906e:8888/?token=0f9f1f4ac6a8a8a889a5237bdab782d3145d9346e3598f0a
[I 14:31:51.567 NotebookApp] or http://127.0.0.1:8888/?token=0f9f1f4ac6a8a8a889a5237bdab782d3145d9346e3598f0a
[I 14:31:51.569 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 14:31:51.579 NotebookApp]

To access the notebook, open this file in a browser:
    file:///root/.local/share/jupyter/runtime/nbserver-1-open.html
Or copy and paste one of these URLs:
    http://fc8a1394906e:8888/?token=0f9f1f4ac6a8a8a889a5237bdab782d3145d9346e3598f0a
    or http://127.0.0.1:8888/?token=0f9f1f4ac6a8a8a889a5237bdab782d3145d9346e3598f0a
```



Password or token:

Log in

### Token authentication is enabled

If no password has been configured, you need to open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if you [enable a password](#).

The command:

```
jupyter notebook list
```

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

```
Currently running servers:  
http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks
```

or you can paste just the token value into the password field on this page.

See [the documentation on how to enable a password](#) in place of token authentication, if you would like to avoid dealing with random tokens.

Cookies are required for authenticated access to notebooks.

### Setup a Password

You can also setup a password by entering your token and a new password on the fields below:

The screenshot shows the Jupyter Notebook web interface. At the top, there's a header with the Jupyter logo and 'Quit' and 'Logout' buttons. Below the header, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, showing a message 'Select items to perform actions on them.' and buttons for 'Upload', 'New', and a refresh icon. Below this, there's a file browser showing a directory structure with a '0' icon and a '/' icon. At the bottom, there's a message 'The notebook list is empty.'

Alternative ( si docker ne fonctionne pas ou si vous n'avez pas les droits sur l'ordi)

Google colab

<https://colab.research.google.com/notebooks/intro.ipynb>