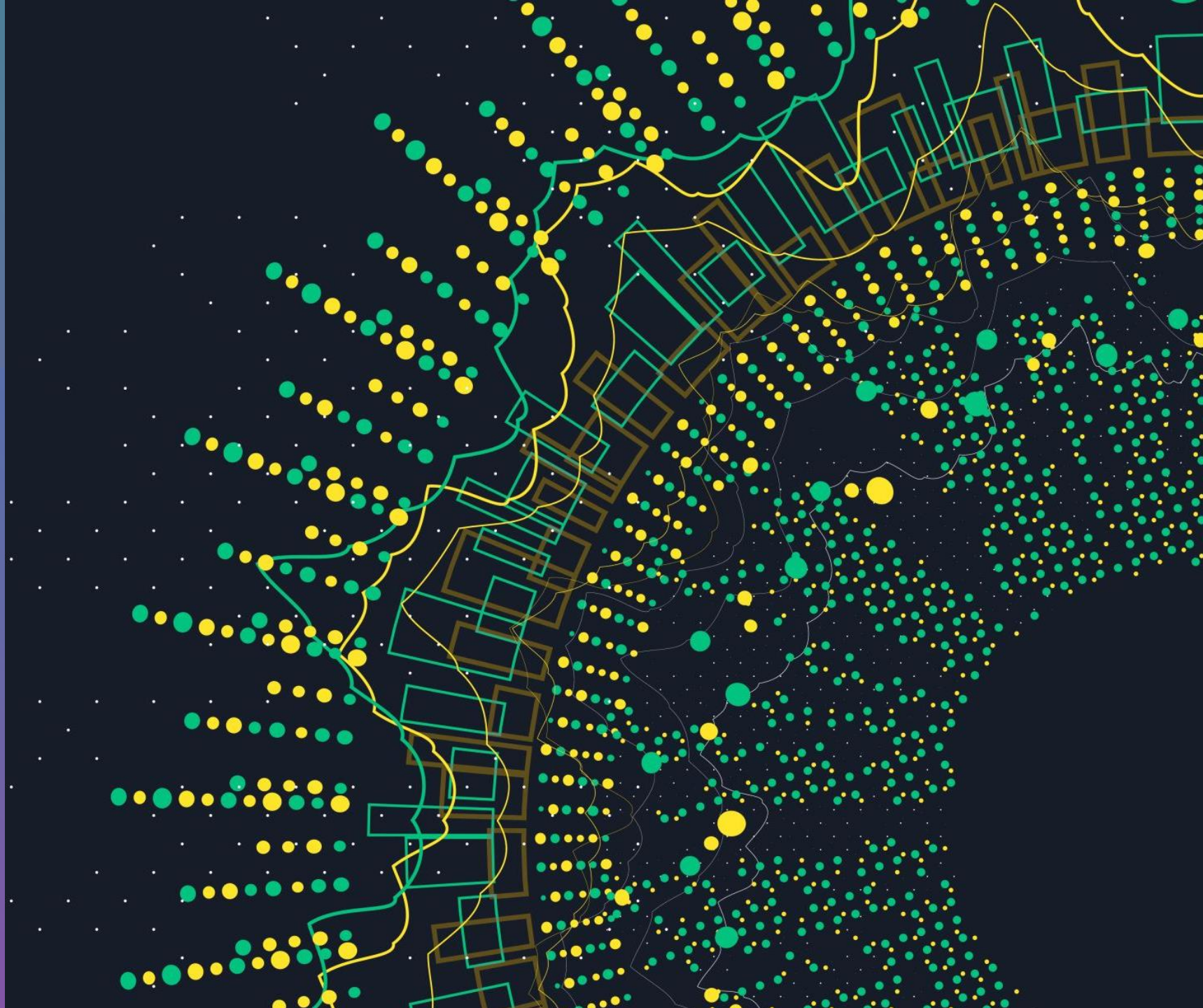


INSTALLATION DE DOCKER ENGINE SOUS DEBIAN 11.6

Thomas D'amore



CONFIGURATION DE DOCKER ENGINE DEPUIS LE REPOSITORY OFFICIEL

- Commandes :
- `# sudo apt-get remove docker docker-engine docker.io containerd runc`
- `# sudo apt-get update`
- `# sudo apt-get install ca-certificates curl gnupg`
- `# sudo install -m 0755 -d /etc/apt/keyrings`
- `# curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg`
- `# sudo chmod a+r /etc/apt/keyrings/docker.gpg`
- `# echo \`
`"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]`
`https://download.docker.com/linux/debian \`
`"$(. /etc/os-release && echo "$VERSION_CODENAME)" stable" | \`
- `# sudo tee /etc/apt/sources.list.d/docker.list > /dev/null`

```
[root@pc ~]# docker run hello-world
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

INSTALLATION DE DOCKER ENGINE ET TEST DE FONCTIONNEMENT AVEC UN CONTENEUR

- `# sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin`
- `# sudo docker run hello-world`

EXEMPLE DE CONTENEUR AVEC UN LANCEMENT PAR LA MÉTHODE « RUN »

```
# docker run --name pandora_community --rm \  
-p 8085:80 \  
-p 41121:41121 \  
-p 162:162 \  
-e DBHOST=mysqlhost.local \  
-e DBNAME=pandora \  
-e DBUSER=pandora \  
-e DBPASS=pandora \  
-e DBPORT=3306 \  
-e SLEEP=5 \  
-e RETRIES=3 \  
-e INSTANCE_NAME=pandora_community \  
-ti rameijeiras/pandorafms-community:740
```

EXEMPLE DE CONTENEUR AVEC UN LANCEMENT PAR LA MÉTHODE « COMPOSE »

- Création et édition du fichier docker-compose.yml avec les spécificités requises
- Lancement du conteneur avec la commande : `# docker-compose up -d`

```
1 $ docker-compose up -d
2 Creating network "my_wordpress_default" with the default driver
3 Pulling db (mysql:5.7)...
4 5.7: Pulling from library/mysql
5 efd26ecc9548: Pull complete
6 a3ed95caeb02: Pull complete
7 ...
8 Digest: sha256:34a0aca88e85f2efa5edff1cea77cf5d3147ad93545dbec99cfe705b03c520de
9 Status: Downloaded newer image for mysql:5.7
10 Pulling wordpress (wordpress:latest)...
11 latest: Pulling from library/wordpress
12 efd26ecc9548: Already exists
13 a3ed95caeb02: Pull complete
14 589a9d9a7c64: Pull complete
15 ...
16 Digest: sha256:ed28506ae44d5def89075fd5c01456610cd6c64006addfe5210b8c675881aff6
17 Status: Downloaded newer image for wordpress:latest
18 Creating my_wordpress_db_1
19 Creating my_wordpress_wordpress_1
```

```
1 version: '3'
2 services:
3   db:
4     image: mysql:5.7
5     volumes:
6       - db_data:/var/lib/mysql
7     restart: always
8     environment:
9       MYSQL_ROOT_PASSWORD: somewordpress
10      MYSQL_DATABASE: wordpress
11      MYSQL_USER: wordpress
12      MYSQL_PASSWORD: wordpress
13
14   wordpress:
15     depends_on:
16       - db
17     image: wordpress:latest
18     ports:
19       - "8000:80"
20     restart: always
21     environment:
22       WORDPRESS_DB_HOST: db:3306
23       WORDPRESS_DB_USER: wordpress
24       WORDPRESS_DB_PASSWORD: wordpress
25       WORDPRESS_DB_NAME: wordpress
26
27 volumes:
28   db_data: {}
```