

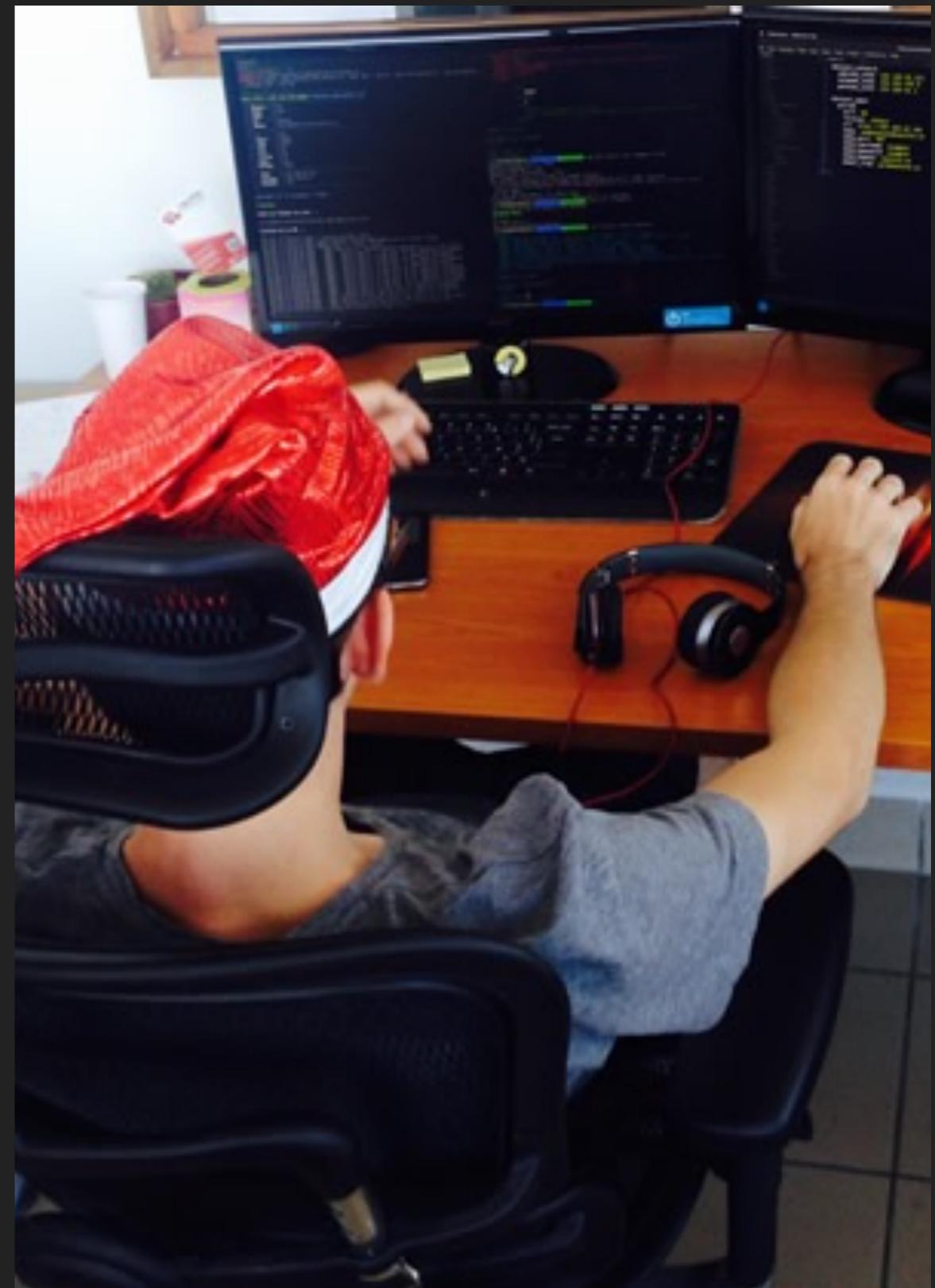
ORCHESTRER SES CONTAINERS AVEC

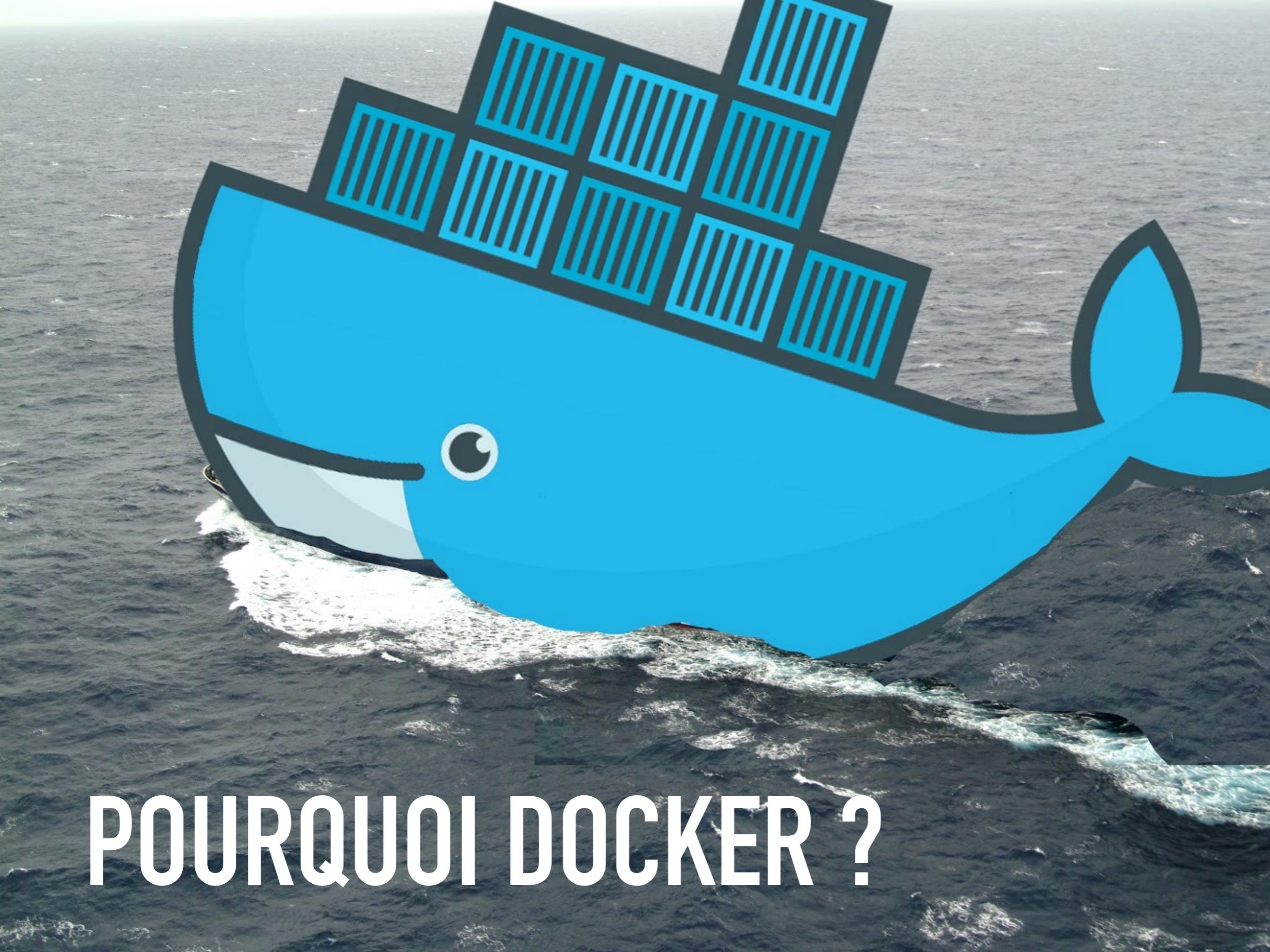
---

# DOCKER-COMPOSE

# VALENTIN OUVRARD

- ▶ SysAdmin @nautilenc
- ▶ Assembleur de containers 🐳
- ▶ <https://OpsNotice.xyz>
- ▶ Twitter @Valentin\_NC
- ▶ Blog at [blog.ouvrard.it](http://blog.ouvrard.it)
- ▶ more at [valentin.ouvrard.it](http://valentin.ouvrard.it)





**POURQUOI DOCKER ?**



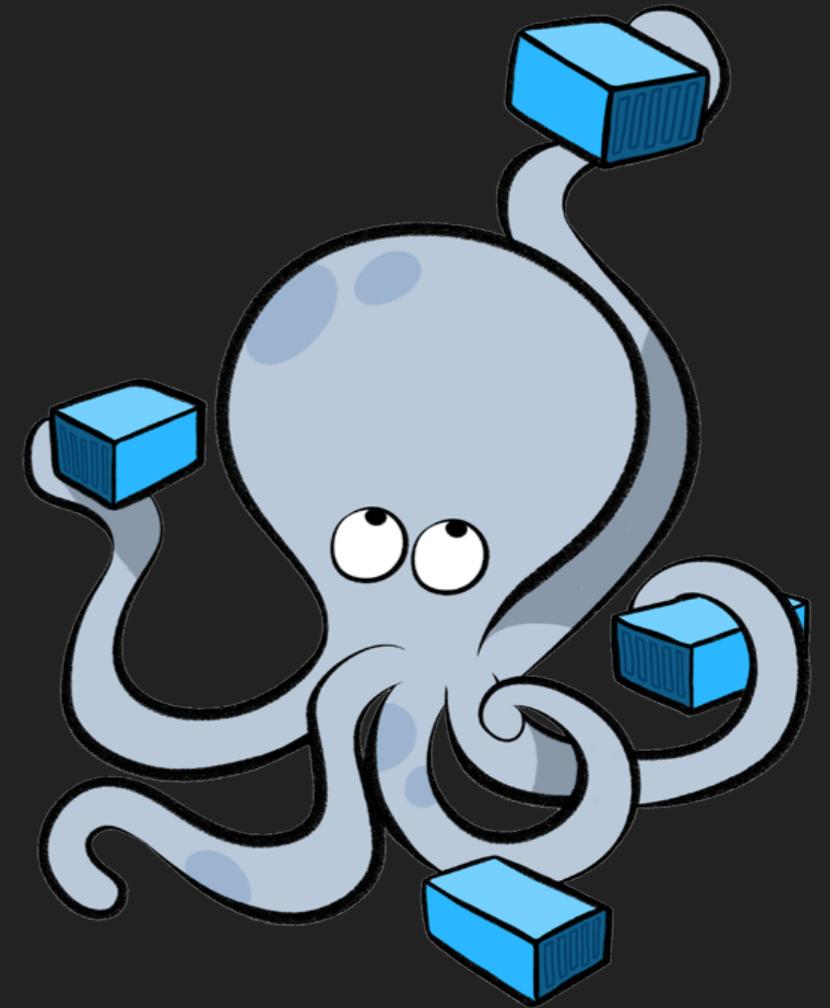
PROBLÈME :

---

LES ORCHESTER

## DOCKER-COMPOSE

- ▶ Text to containers
- ▶ Codé en Python
- ▶ YAML File
- ▶ Ports, Volumes, Links, Network ...
- ▶ CLI Docker-like (docker-compose ps)
- ▶ Réutilisable, partageable, scalable.



# EXAMPLE : WORDPRESS

```
services:  
  db:  
    image: mysql:5.7  
    volumes:  
      - "./.data/db:/var/lib/mysql"  
    restart: always  
  environment:  
    MYSQL_ROOT_PASSWORD: wordpress  
    MYSQL_DATABASE: wordpress  
    MYSQL_USER: wordpress  
    MYSQL_PASSWORD: wordpress  
  wordpress:  
    depends_on:  
      - db  
    image: wordpress:latest  
    links:  
      - db  
    ports:  
      - "8000:80"  
    restart: always  
  environment:  
    WORDPRESS_DB_HOST: db:3306  
    WORDPRESS_DB_PASSWORD: wordpress
```



# DEMO #1

# WORDPRESS LOCAL



# DOCKER-COMPOSE WORKFLOW

1

Write your dockerfile

```
WORKDIR /code
ADD requirements.txt
/codes/
RUN pip install -r
requirements.txt
ADD . /code
CMD python app.py
```

2

Write your compose.yml file

```
web:
  build: .
  links:
    - db
  ports:
    - "8000:8000"
db:
  image: postgres
```

3

Run your app

```
$ docker-compose up
```

`$ docker-compose build web`

`$ docker-compose up --no-deps -d web`

# POURQUOI RÉINVENTER LA ROUE ?

 Pull requests Issues Gist

## Search

in:path docker-compose.yml extension:yml

-  [Repositories](#) 65
-  [Code](#) 39,554
-  [Issues](#) 4,757
-  [Users](#)

### Languages

YAML	39,405
Python	1
Shell	1

We've found 39,554 code results

 [mozilla/mozillians](#) – **docker-compose.yml**  
Last indexed on 16 Mar.  
 [docker-compose.yml](#)

 [desertpy/presentations](#) – **docker-compose.yml**  
Last indexed on 19 Mar.  
 [rq-godber/docker-compose.yml](#)

 [crs4/pydoop](#) – **docker-compose.yml**  
Last indexed on 19 Mar.



COMPOSE

---

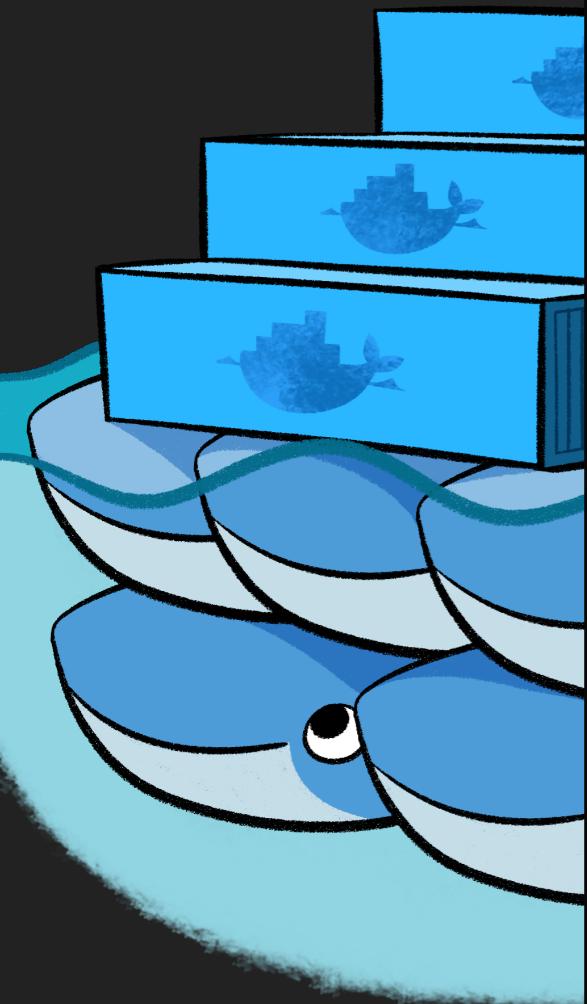
NETWORK



COMPOSE

---

VOLUMES



tion Bundle

Container applications

[com/dab](#)

```
1 {
2   "Services": {
3     "db": {
4       "Env": [
5         "MYSQL_ROOT_PASSWORD=wordpress",
6         "MYSQL_PASSWORD=wordpress",
7         "MYSQL_USER=wordpress",
8         "MYSQL_DATABASE=wordpress"
9       ],
10      "Image": "mysql@sha256:5bfc6c2a3adbcb9aad",
11      "Networks": [
12        "default"
13      ]
14    },
15    "wordpress": {
16      "Env": [
17        "WORDPRESS_DB_HOST=db:3306",
18        "WORDPRESS_DB_PASSWORD=wordpress"
19      ],
20      "Image": "wordpress@sha256:44902067a29a05",
21      "Networks": [
22        "default"
23      ],
24      "Ports": [
25        {
26          "Port": 80,
27          "Protocol": "tcp"
28        }
29      ]
30    }
31  }
32}
```

NORMAL

PASTE

master > ./demo01wordpress.dab

# EXAMPLE : OWN CLOUD

```
cloud_web:  
  image: nginx  
  restart: always  
  ports:  
    - 80:80  
    - 443:443  
  log_driver: syslog  
  link:  
    - cloud_engine  
volumes:  
  - ./etc/nginx/nginx.conf:/etc/nginx/nginx.conf:ro  
  - ./var/log/nginx:/var/log/nginx  
  - ./etc/letsencrypt:/etc/letsencrypt  
cloud_engine:  
  image: owncloud:latest  
  expose:  
    - "80"  
volumes:  
  - /srv/Owncloud/data:/var/www/html/data  
  - /srv/Owncloud/config:/var/www/html/config  
links:  
  - cloud_db  
  - cloud_cache
```



```
1  
2  cloud_db:  
3    image: mysql:latest  
4    expose:  
5      - "3306"  
6    environment:  
7      MYSQL_DATABASE: owncloud  
8      MYSQL_USER: owncloud  
9      MYSQL_PASSWORD: password  
10     MYSQL_ROOT_PASSWORD: r00tpassword  
11    volumes:  
12      - /srv/Owncloud/mysql:/var/lib/mysql  
13  cloud_cache:  
14    image: redis:latest  
15    expose:  
16      - "6379"  
17  
18  
19  
20  
21  
22  
23  
24
```



I LOVE THE CLOUD!

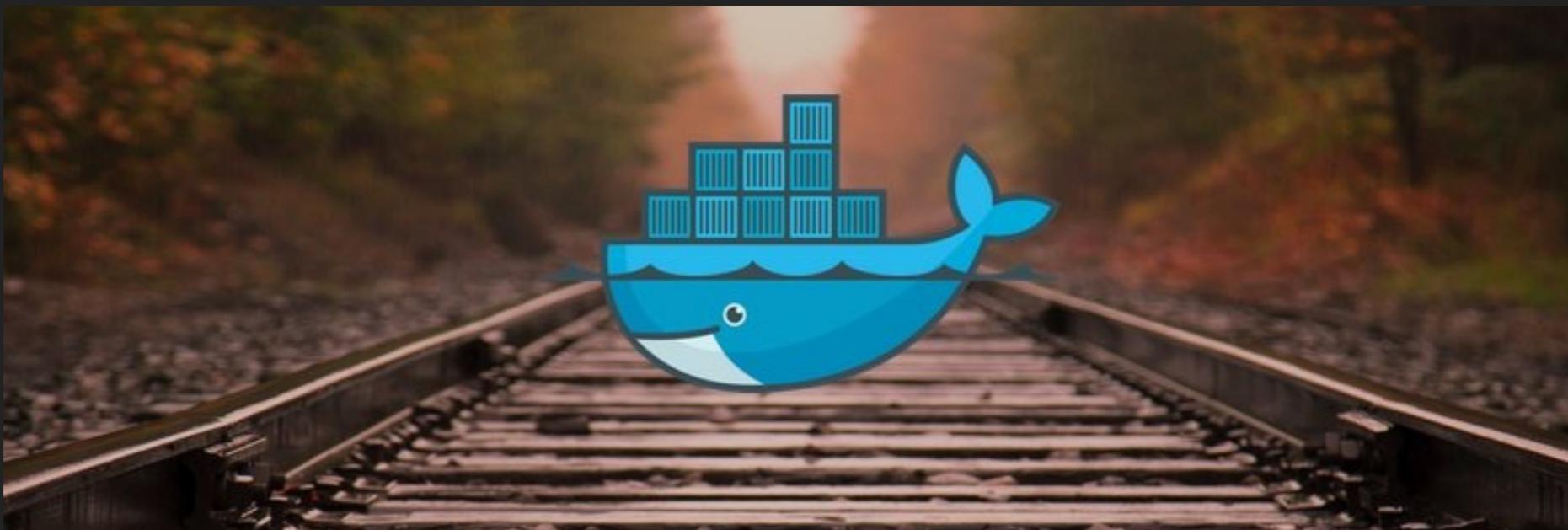
# DEMO #2

# OWNCLOUD SUR LE CLOUD

---

# LIMITES

- ▶ Manage UN serveur Docker
- ▶ Requière Python
- ▶ Ligne de commande uniquement
- ▶ Rien d'autre, c'est génial !





## FULLY AUTOMATED DOCKER DEPLOYMENT

---

# FADD

<https://github.com/valentin2105/FADD>

Based on Docker-Compose

# FULLY AUTOMATED DOCKER DEPLOYMENT

<https://github.com/valentin2105/FADD>

## Features :

- Nginx TLS Reverse-proxy
- Auto-Let's Encrypt support
- SNI Multi-Domain support
- Multi-image deployment (Wordpress, LEMP, Ghost, Drupal ...)
- Automated script for launch a deployment

## Example :

```
add_stack.sh --image=wordpress --domain=site01.example.com --expose=8101
```

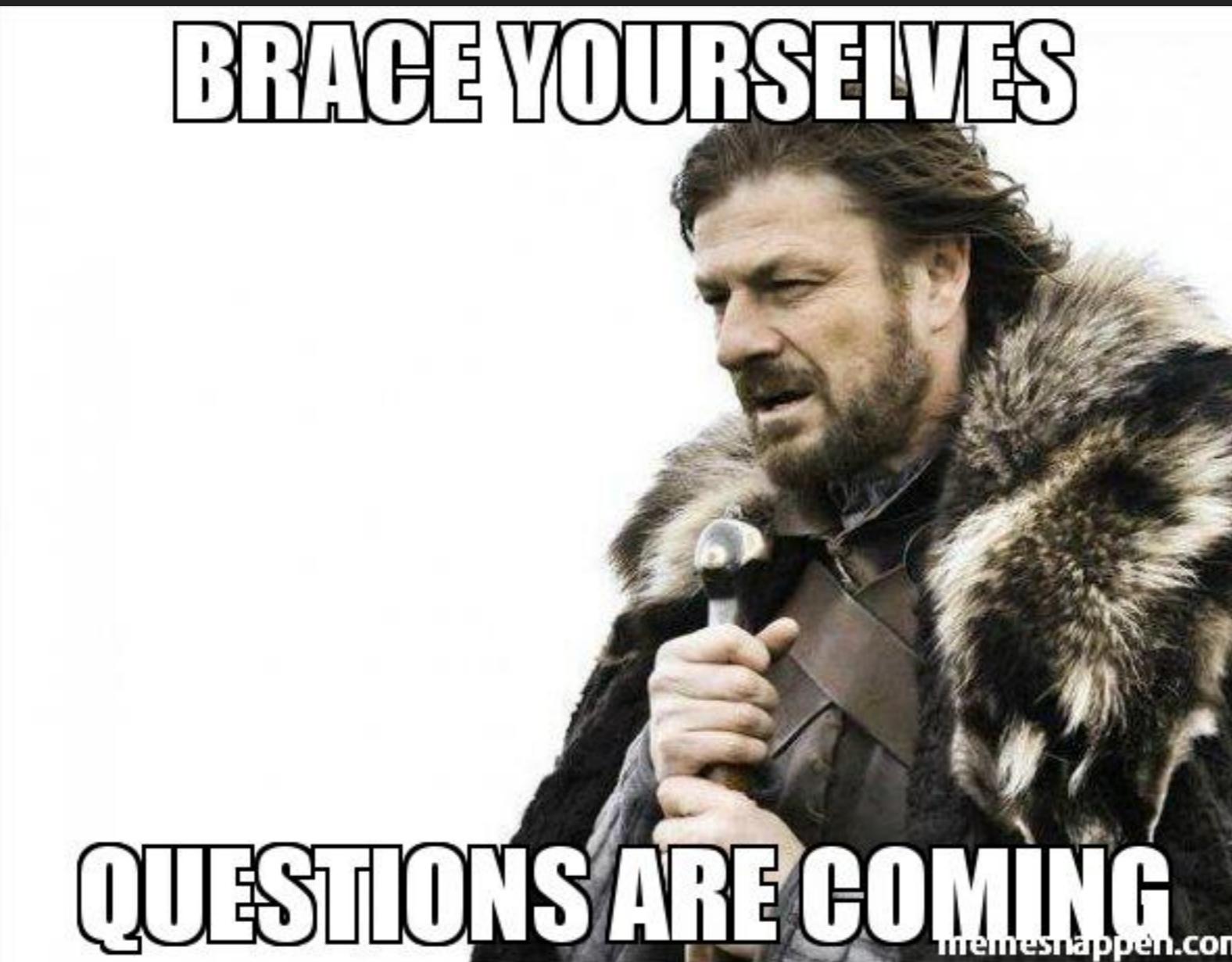
# FULLY AUTOMATED DOCKER DEPLOYMENT

<https://github.com/valentin2105/FADD>

# DEMO

# QUESTIONS ?

**BRACE YOURSELVES**



**QUESTIONS ARE COMING**

memesnappen.com