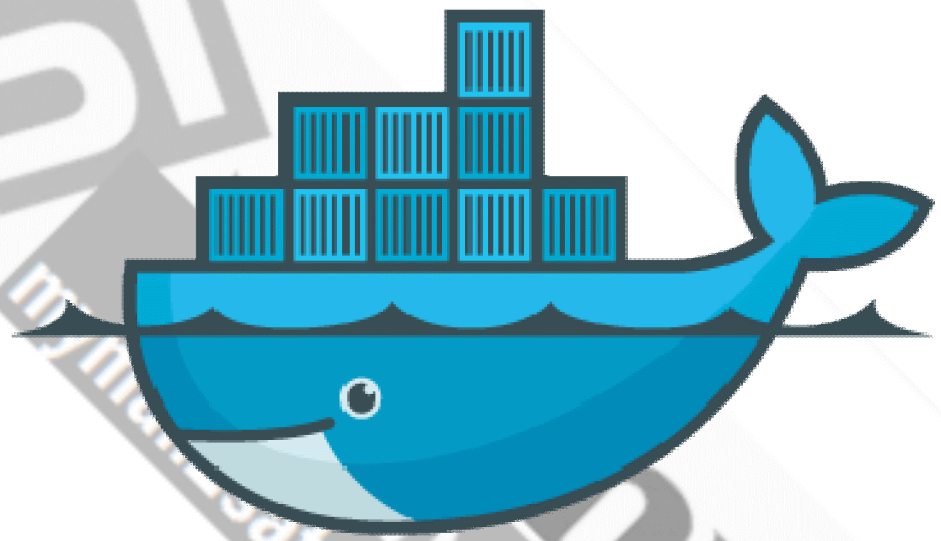
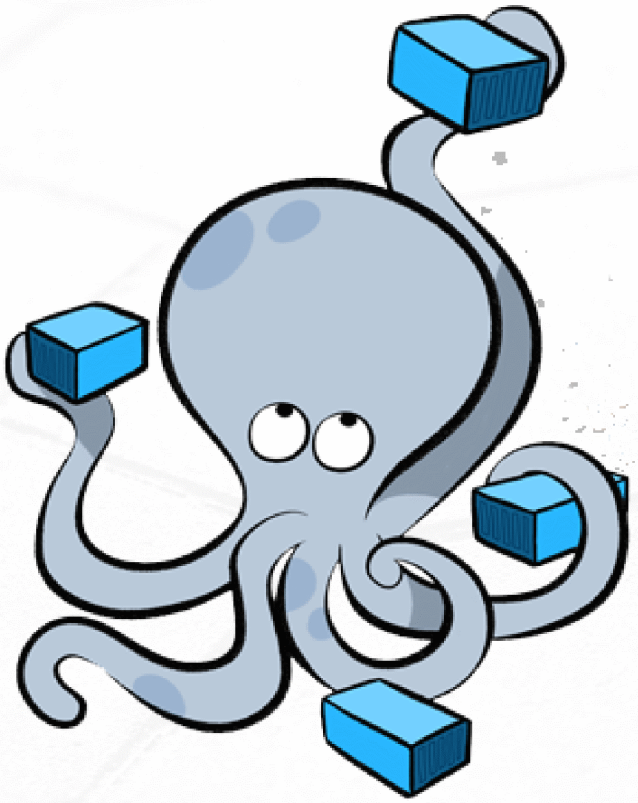


docker

Docker Compose

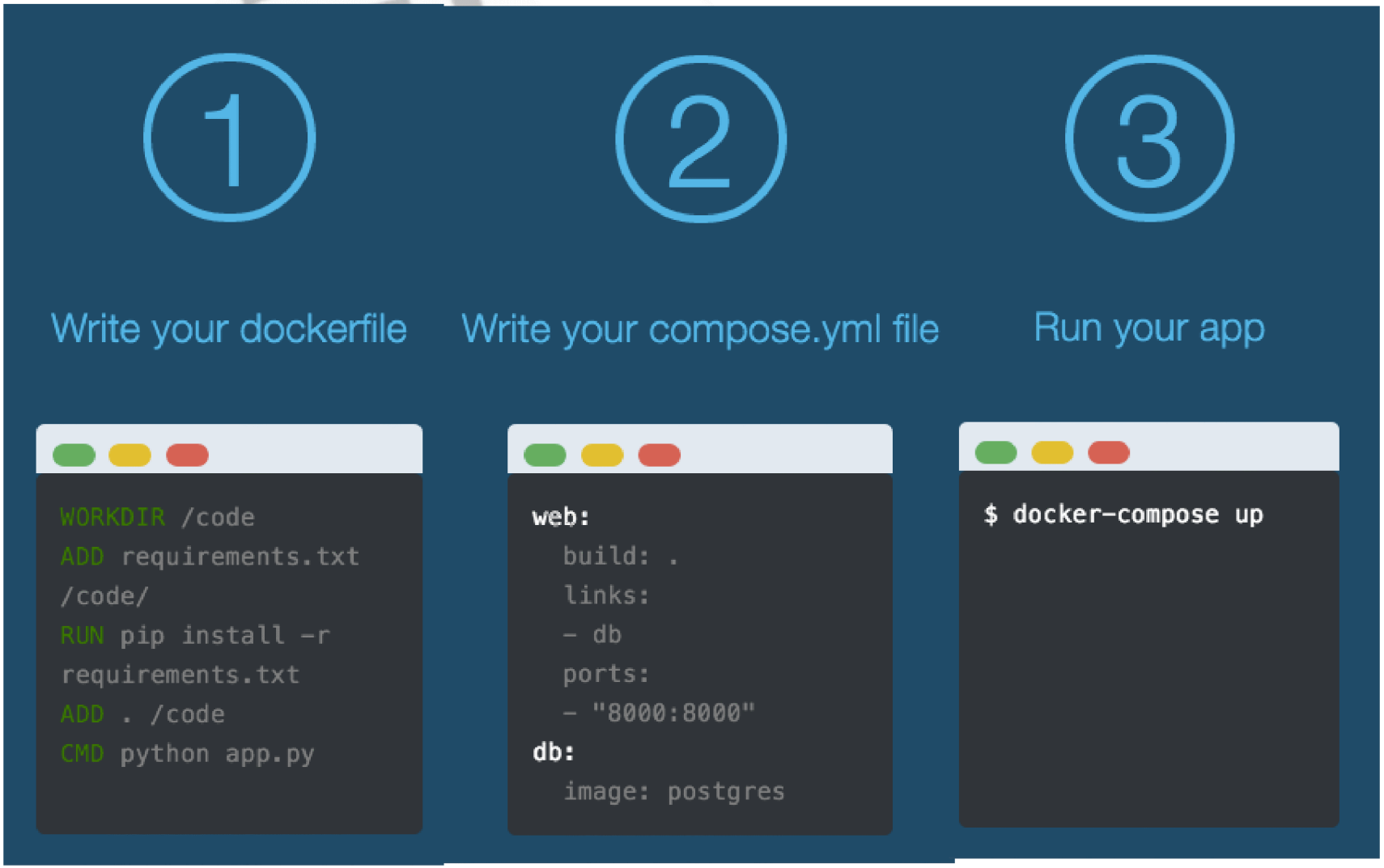


```
php:
  build: php
  ports:
    - "80:80"
    - "443:443"
  volumes:
    - ./php/www:/var/www/html
  links:
    - db
```

```
$ docker-compose up
```

a three-step process

- 1) Define your app's environment with a Dockerfile so it can be reproduced anywhere.
- 2) Define the services that make up your app in docker-compose.yml so they can be run together in an isolated environment.
- 3) Lastly, run docker-compose up and Compose will start and run your entire app.


1
2
3
Write your dockerfile
Write your compose.yml file
Run your app
WORKDIR /code
ADD requirements.txt /code/
RUN pip install -r requirements.txt
ADD . /code
CMD python app.py
web:
build: .
links:
- db
ports:
- "8000:8000"
db:
image: postgres
$ docker-compose up

Install Docker Compose

- you'll need to install Docker first.

- $ curl -L "https://github.com/docker/compose/releases/download/1.11.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

- $ **chmod** +**x** /usr/**local**/bin/docker-compose
- $ docker-compose --*version*

Define the project

1) Create an empty project directory.

```
#mkdir  wordpress
#cd wordpress
#vi  docker-compose.yml
```

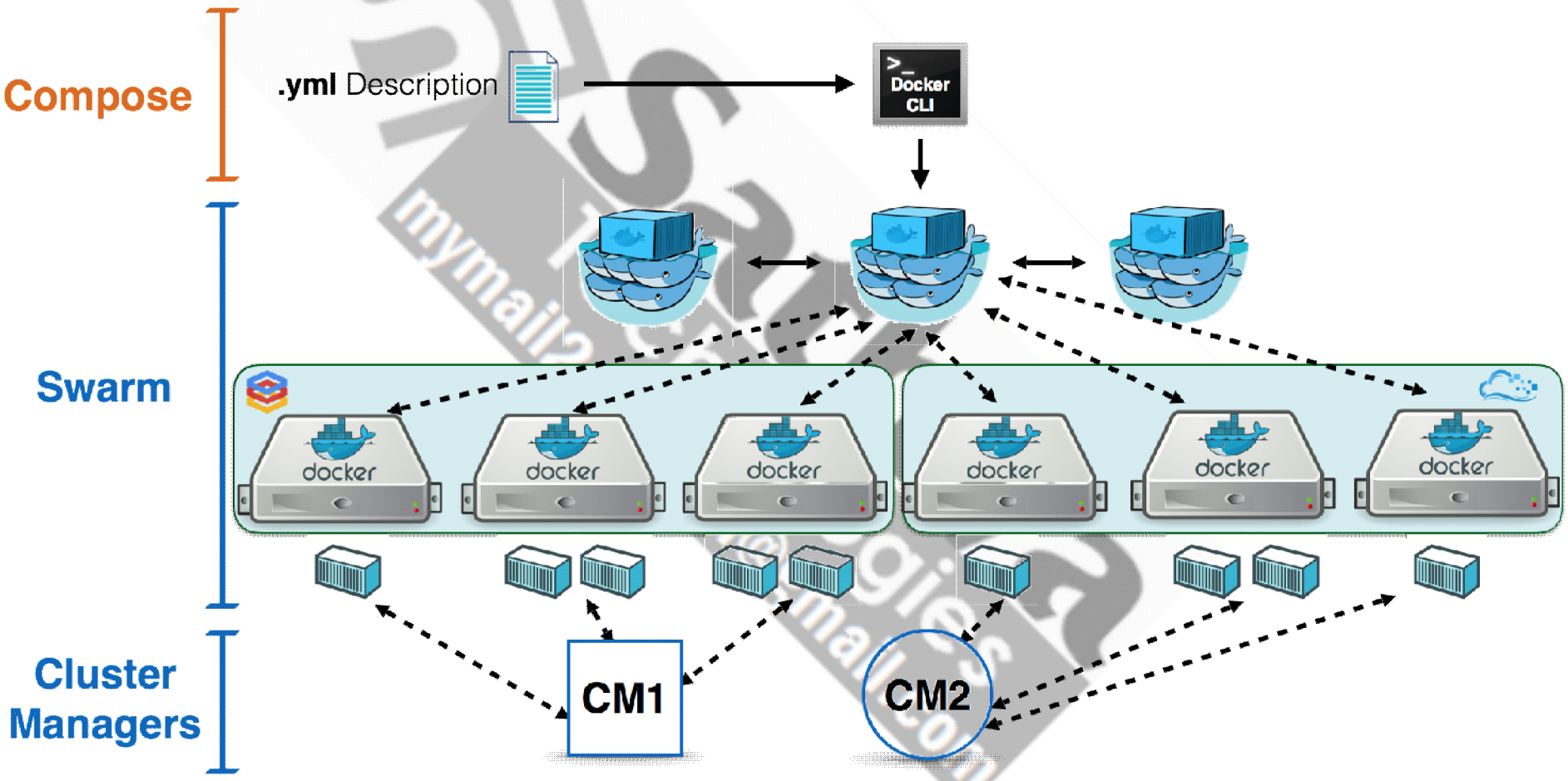
2) Build the project

```
$ docker-compose up –d
```

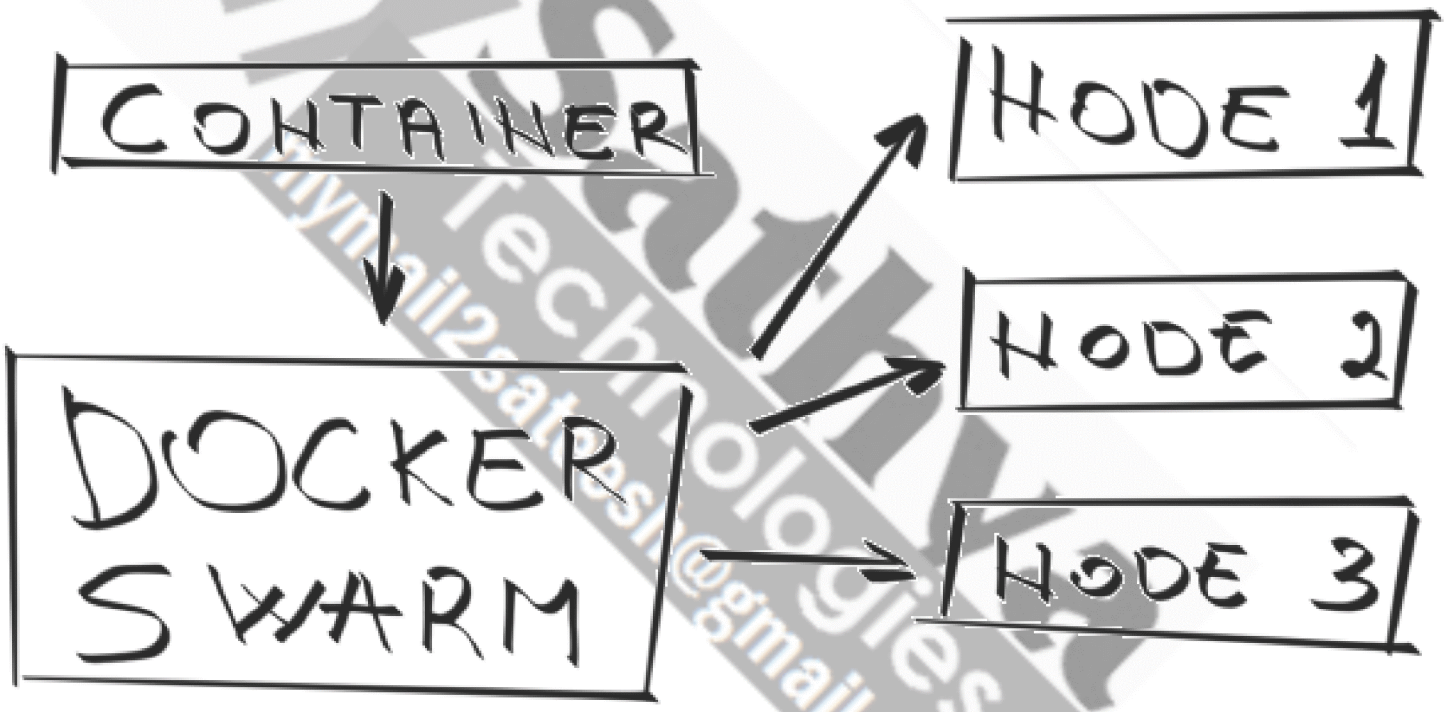
3) Open browser : https:\\ip:8000

```
version: '2'
services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/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
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_PASSWORD: wordpress
volumes:
   db_data:
```

Docker Swarm



Docker Swarm



curl -fsfl https://test.docker.com | sh

```
mgr# docker swarm init  --listen-addr 10.128.0.4:2377
mgr# docker swarm join-token manager
mgr# docker node list

nd1# docker swarm join \ --token SWMTKN-1-
   030ofl3to3lkylc0g04yitzjca9rbo3fmz6put4lsjm7ytkzb9-
   6djd19g0nitbl1q8bz3xlkm36 \  10.128.0.4:2377

nd2# docker swarm join \ --token SWMTKN-1-
   030ofl3to3lkylc0g04yitzjca9rbo3fmz6put4lsjm7ytkzb9-
   6djd19g0nitbl1q8bz3xlkm36 \  10.128.0.4:2377
```

```
# docker swarm leave
  # docker swarm leave --force
  # service docker stop
  # service docker start

  mgr# docker service ls
  mgr# docker run -itd -p 80:80 nginx
  mgr# docker service create --replicas 4 nginx
# docker service create -p 80:80 --name webserver nginx
# docker service scale webserver=5
# docker logs -f <cid>
  # docker exec -it <cid> hostname
```

```
mgr# docker service create --name mytom -p
   8080:8080 tomcat
mgr# netstat -lntp   ---> (you can find port: 8080)
nd1# netstat -lntp   ---> (you can find port: 8080)
nd2# netstat -lntp   ---> (you can find port: 8080)

#docker service ls
#docker service inspect <service id>
#docker service inspect <service id> --pretty
```

Scale Services:

```
#docker service update --replicas 10 mytom
```