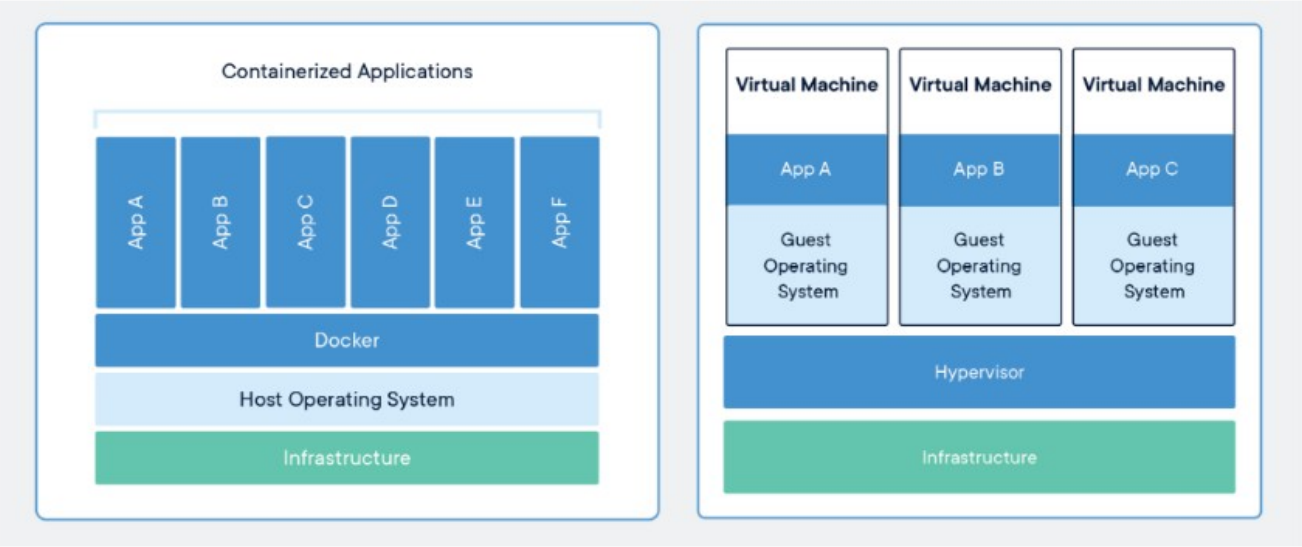
Docker

Docker es un proyecto de **código abierto** que automatiza el **despliegue de aplicaciones** dentro de **contenedores** de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

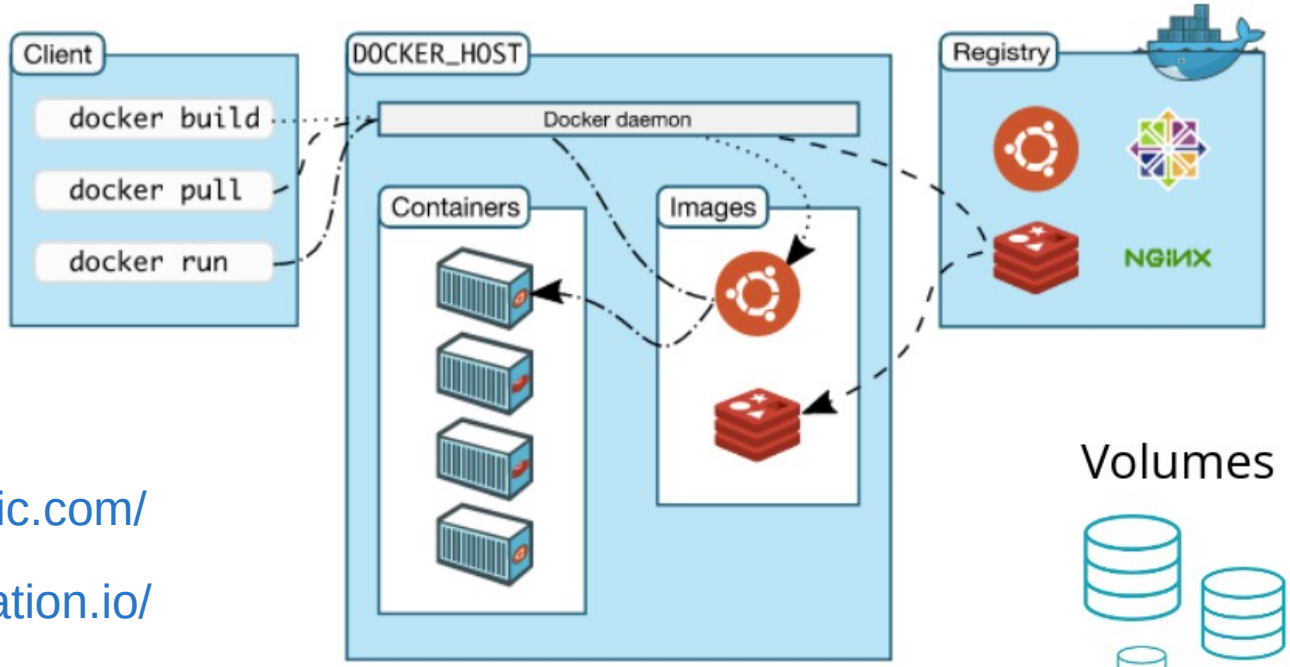
(Wikipedia)

Docker

Maquinas Virtuales



https://hub.docker.com/



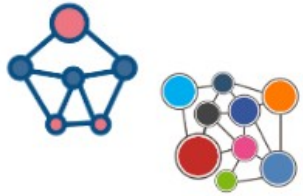
UI Clients

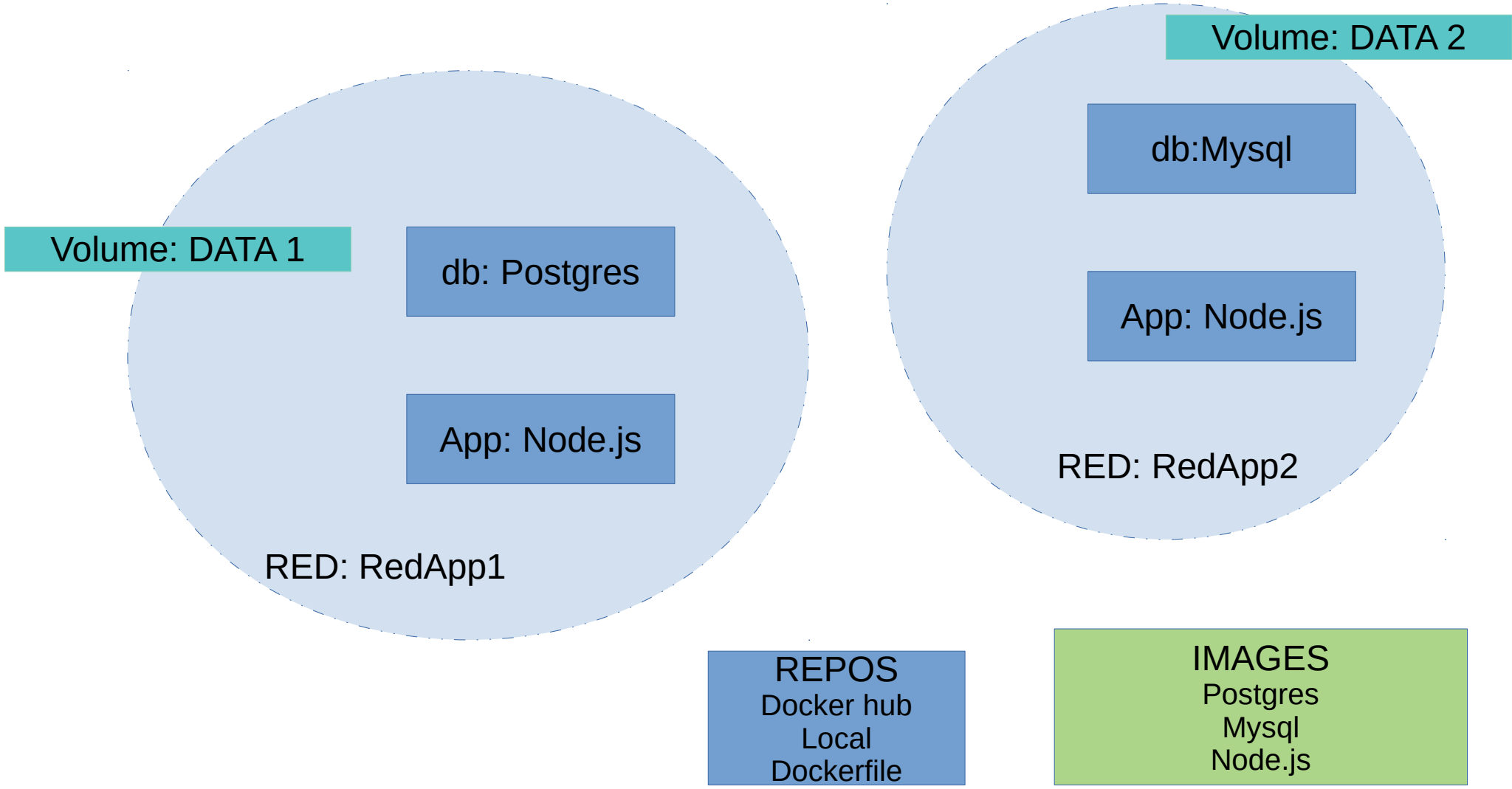
https://kitematic.com/

https://dockstation.io/

Volumes

Networks




Volume: DATA 1
db: Postgres
App: Node.js
RED: RedApp1
Volume: DATA 2
db:Mysql
App: Node.js
RED: RedApp2
REPOS
Docker hub
Local
Dockerfile
IMAGES
Postgres
Mysql
Node.js

https://hub.docker.com/_/postgres

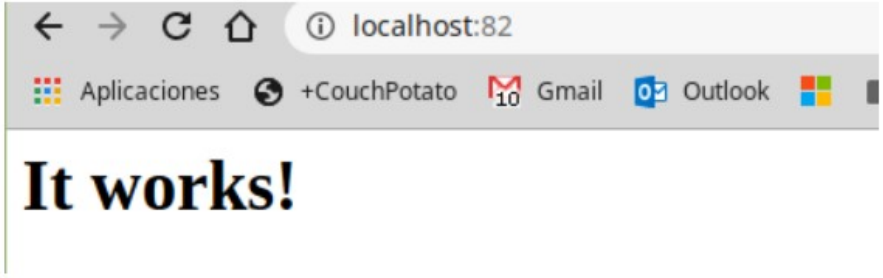
How to use this image

start a postgres instance

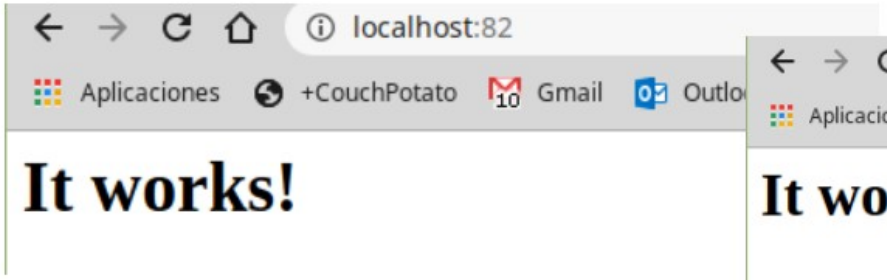
```
$ docker run --name some-postgres -e POSTGRES_PASSWORD=mysecretpassword -d postgres
```

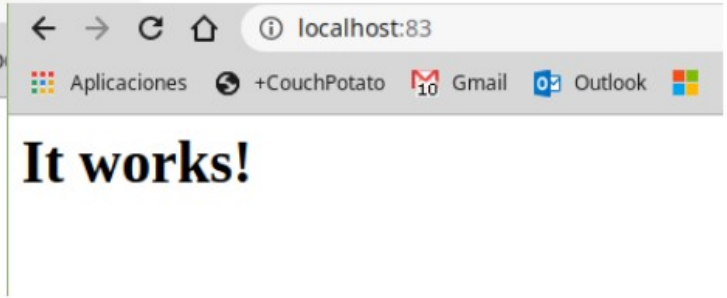
```
1 docker run -d -p 82:80 httpd
```

```
chaca@dgT460:~$ docker run -d -p 82:80 httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
bf5952930446: Pull complete
3d3fecf6569b: Pull complete
b5fc3125d912: Pull complete
679d69c01e90: Pull complete
76291586768e: Pull complete
Digest: sha256:3cbdff4bc16681541885ccf1524a532afa28d2a6578ab7c2d5154a7abc182379
Status: Downloaded newer image for httpd:latest
b0a6691c38d12a4bca4eb55d8308f2ec0f029b72957804fa2b57026e26e3710f
chaca@dgT460:~$
```



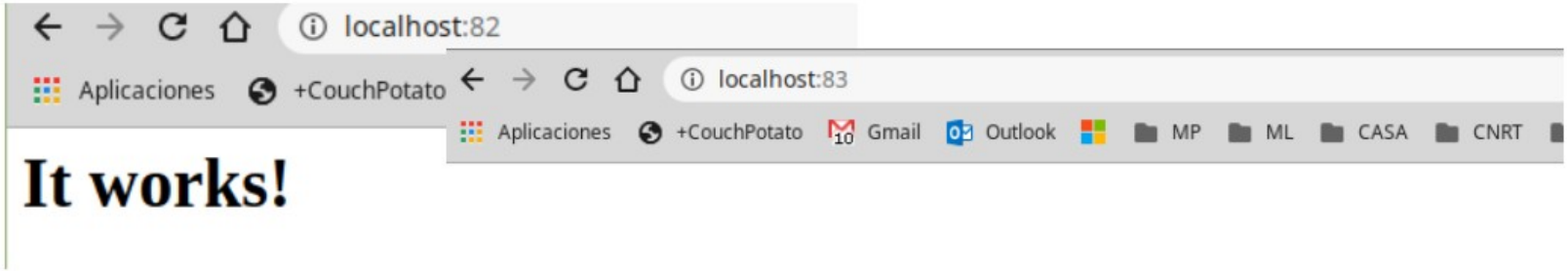
```
chaca@dgT460:~$ docker ps
CONTAINER ID        IMAGE               COMMAND              CREATED             STATUS
b0a6691c38d1        httpd               "httpd-foreground"   5 minutes ago       Up 5 minutes
chaca@dgT460:~$ docker run -d -p 83:80 --name=http1 httpd
68b7ef11a54ebd3c06ccfbb4c1fd1cac8f92f57aa909ce588ef41cbd3610777a
chaca@dgT460:~$ docker ps
CONTAINER ID        IMAGE               COMMAND              CREATED             STATUS
68b7ef11a54e        httpd               "httpd-foreground"   5 seconds ago       Up 4 seconds
b0a6691c38d1        httpd               "httpd-foreground"   6 minutes ago       Up 6 minutes
chaca@dgT460:~$
```





```
chaca@dgT460:~$ docker exec -it http1 bash
root@cdf40b40b0fd:/usr/local/apache2# ls
bin  build  cgi-bin  conf  error  htdocs  icons  include  logs  modules
root@cdf40b40b0fd:/usr/local/apache2#
```

```
chaca@dgT460:~$ docker stop http1
http1
chaca@dgT460:~$ docker ps
CONTAINER ID        IMAGE               COMMAND              CREATED
b0a6691c38d1        httpd               "httpd-foreground"   10 minutes ago
chaca@dgT460:~$
```



```
chaca@dgT460:~$ docker images httpd
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
httpd               latest              a6ea92c35c43        3 weeks ago         166MB
chaca@dgT460:~$ docker images alpine
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
alpine              3.9                 055936d39205        15 months ago       5.53MB
chaca@dgT460:~$ docker images postgres:9.6
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
postgres            9.6                 e61715500c31        2 weeks ago         200MB
chaca@dgT460:~$
```

my_script.py

```
1 print('Prueba desde python !!!')
```

Dockerfile

```
1 FROM python:3-alpine
2 ADD my_script.py /
3 CMD [ "python", "./my_script.py" ]
```

crea la imagen

```
docker build -t imagen_de_prueba_python .
```

```
chaca@dgT460:docker$ docker images *prue*
REPOSITORY          TAG                 IMAGE ID            CREATED              SIZE
imagen_prueba_p1    latest              6c76c15fa8df        About a minute ago   112MB
```

ejecuta el contenedor

```
1 chaca@dgT460:docker$ docker run imagen_prueba_p1
2 Prueba desde python !!!
```

https://github.com/TABI-UNLP/pentaho-biserver

docker-compose.yml

```yaml
version: '3.5'

services:
  postgres:
    container_name: tabi_postgres
    image: postgres
    environment:
      POSTGRES_USER: ${POSTGRES_USER:-postgres}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD:-212121}
      PGDATA: /data/postgres
    volumes:
       - postgres:/data/postgres
    ports:
      - "5432:5432"
    networks:
      - tabi
    restart: unless-stopped

  pgadmin:
    container_name: tabi_pgadmin
    image: dpage/pgadmin4
    environment:
      PGADMIN_DEFAULT_EMAIL: ${PGADMIN_DEFAULT_EMAIL:-admin}
      PGADMIN_DEFAULT_PASSWORD: ${PGADMIN_DEFAULT_PASSWORD:-212121}
    volumes:
       - pgadmin:/root/.pgadmin
    ports:
      - "${PGADMIN_PORT:-5050}:80"
    networks:
      - tabi
```

https://github.com/TABI-UNLP/pentaho-biserver

1

```
clone https://github.com/TABI-UNLP/pentaho-biserver.git
cd penthao-server
docker-compose up -d
```

2

```
docker-compose stop
```

```
docker-compose start
```

3

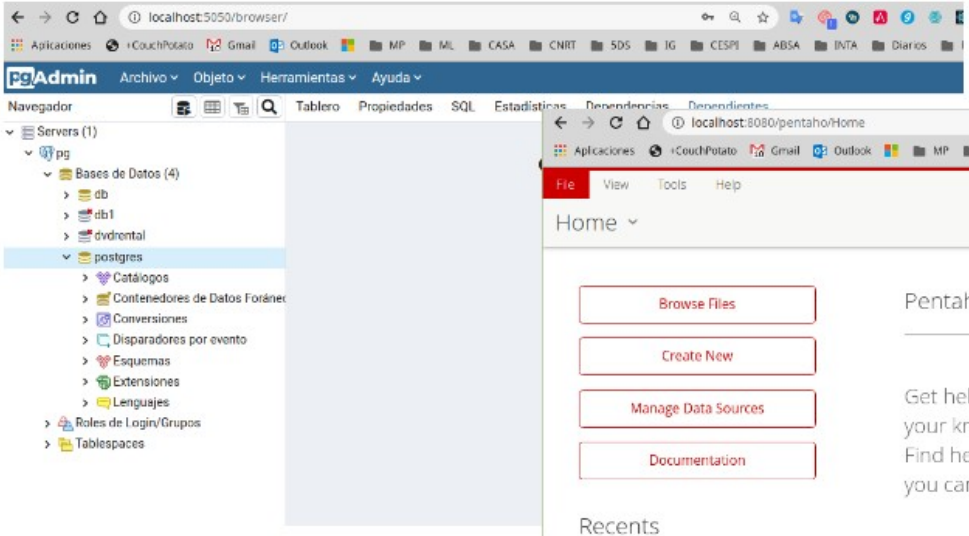
```
docker-compose down
```

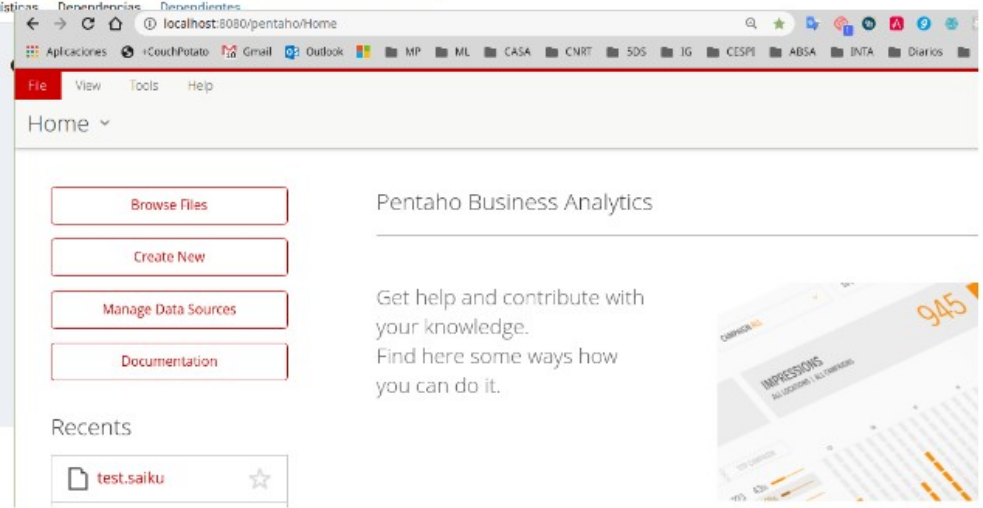
http://localhost:5050

admin / 212121

http://localhost:8080

admin / password





Imágenes útiles

- alpine: Linux reducido
- httpd: Servidor web Apache
- ubuntu: Ubuntu
- mongo: Base de datos MongoDB (documentos)
- mysql: Base de datos MySQL (relacional)
- postgres: Base de datos PostgreSQL (relaional)
- node: Node.js
- php, elasticsearch, haproxy, wordpress, rabbitmq, python, openjdk, tomcat, jenkins, redmine, flink, spark, …

Referencia

https://ualmtorres.github.io/SeminarioDockerPresentacion/