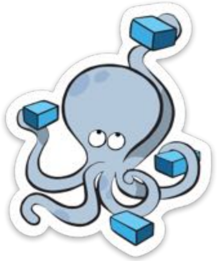
Creació d´imatges amb Docker

Institut de Ciències de l'Educació Josep Pallach (ICE)
Codi 0007190019
http://www2.udg.edu/tabid/6126/Default.aspx?ID=1968

INS Montilivi, Avinguda de Montilivi, 125, 17003 Girona

27, 28, 29 de Juny del 2017

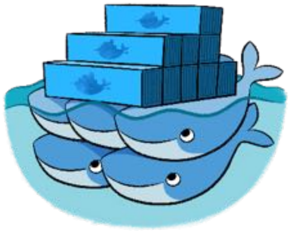
Altres conceptes Docker



Compose

Definir i fer deploy d’aplicacions multicontainer

Sessió 2



Swarm

Ús de multiples màquines com a una de sola, per controlar multiples entorns de containers.

Docker Machine

Crear i gestionar instàncies Docker localment i en cloud

Cicle de vida d'un container

La vida d'un container…

- Concepció
  - **BUILD** una imatge a partir d'un Dockerfile
- Neixement
  - **RUN** (create+start) un container
- Reprocció
  - **COMMIT** (persisteix) un container a una imatge
  - **RUN** un nou container des d'una imatge
- Sleep
  - **KILL** un continer en execució
- Wake
  - **START** un container aturat
- Death
  - **RM** (delete) un container aturat

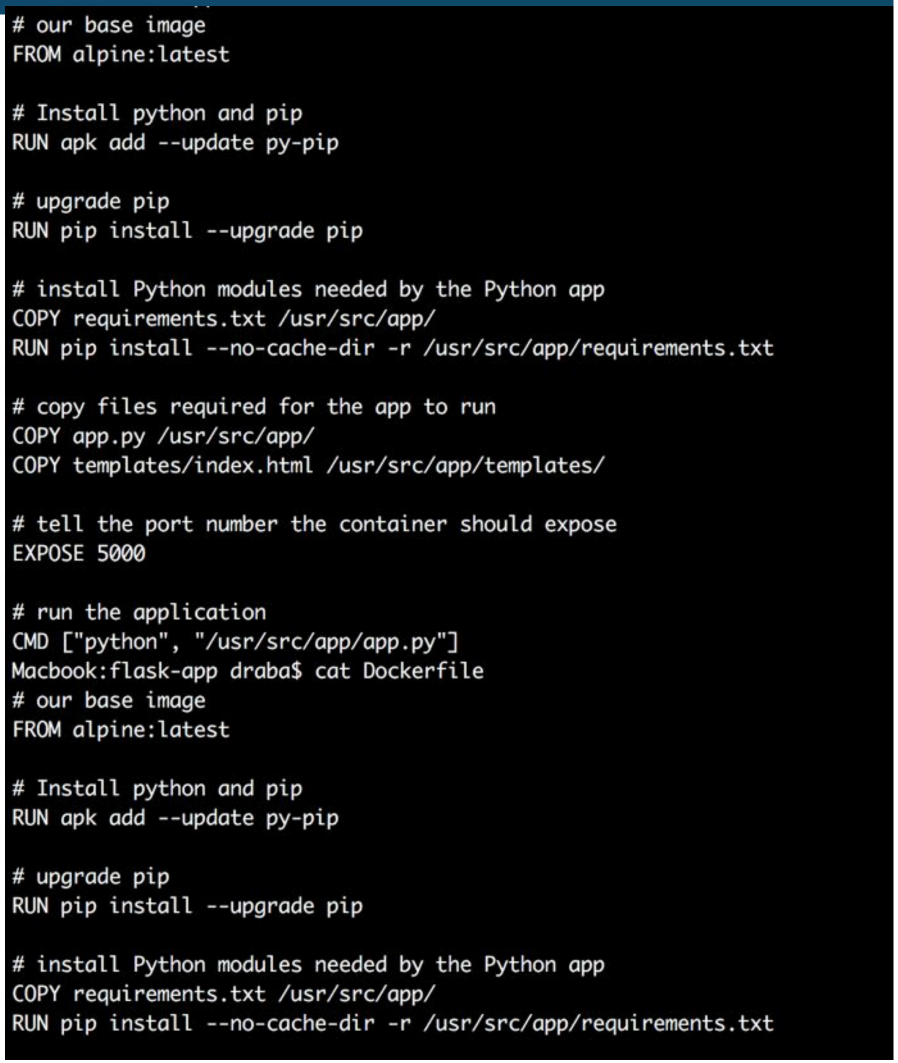
- Extinció
  - **RMI** una imatge de container (delete image)

Dockerfile

- Conceptualment és un Makefile
- Extend una imatge base
- Que es converteix en una nova imatge
- Imperatiu, no Declaratiu

Defineix la recepta per construir una imatge

- S’utilitza docker build per executar un dokerfile
- Es poden definir ordres per defecte per executar, definir ports exposats, etc.



```
# our base image
FROM alpine:latest

# Install python and pip
RUN apk add --update py-pip

# upgrade pip
RUN pip install --upgrade pip

# install Python modules needed by the Python app
COPY requirements.txt /usr/src/app/
RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt

# copy files required for the app to run
COPY app.py /usr/src/app/
COPY templates/index.html /usr/src/app/templates/

# tell the port number the container should expose
EXPOSE 5000

# run the application
CMD ["python", "/usr/src/app/app.py"]
Macbook:flask-app draba$ cat Dockerfile
# our base image
FROM alpine:latest

# Install python and pip
RUN apk add --update py-pip

# upgrade pip
RUN pip install --upgrade pip

# install Python modules needed by the Python app
COPY requirements.txt /usr/src/app/
RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt
```

Manage containers

```
$ docker create      # creates a container but does not start it.
$ docker run         # creates and starts a container.
$ docker stop        # stops it.
$ docker start       # will start it again.
$ docker restart     # restarts a container.
$ docker rm          # deletes a container.
$ docker kill        # sends a SIGKILL to a container.
$ docker attach      # will connect to a running container.
$ docker wait        # blocks until container stops.
$ docker exec        # runs a command in a running container.
```

http://www.jayway.com/2015/03/21/a-not-very-short-introduction-to-docker/

http://bytearrays.com/docker-most-useful-commands/

Inspect containers

```
$ docker ps           # shows running containers.
$ docker inspect      # info on a container (incl. IP address).
$ docker logs         # gets logs from container.
$ docker events       # gets events from container.
$ docker port         # shows public facing port of container.
$ docker top          # shows running processes in container.
$ docker diff         # shows changed files in container's FS.
$ docker stats        # shows metrics, memory, cpu, filsystem
```

http://www.jayway.com/2015/03/21/a-not-very-short-introduction-to-docker/

Containers

```
# Run a container interactively
$ docker run -it --rm ubuntu

# Run a container in the background
$ docker run -d hadoop

# Publish container port 80 on a random port on the Host
$ docker run -p 80 nginx

# Publish container port 80 on port 8080 on the Host
$ docker run -p 8080:80 nginx
```

http://www.jayway.com/2015/03/21/a-not-very-short-introduction-to-docker/

Containers

```
# Publish container port 80 on port 8080 on the localhost interface on the Host
$ docker run -p 127.0.0.1:8080:80 nginx
```

```
# Publish all EXPOSEd ports from the container on random ports on the Host
$ docker run -P nginx
```

http://www.jayway.com/2015/03/21/a-not-very-short-introduction-to-docker/

Containers

```
# Limit the amount of memory
$ docker run -m 256m yourapp

# Limit the number of shares of the CPU this process uses (out of 1024)
$ docker run --cpu-shares 512 mypp

# Change the user for the process to www instead of root (good for security)
$ docker run -u=www nginx
```

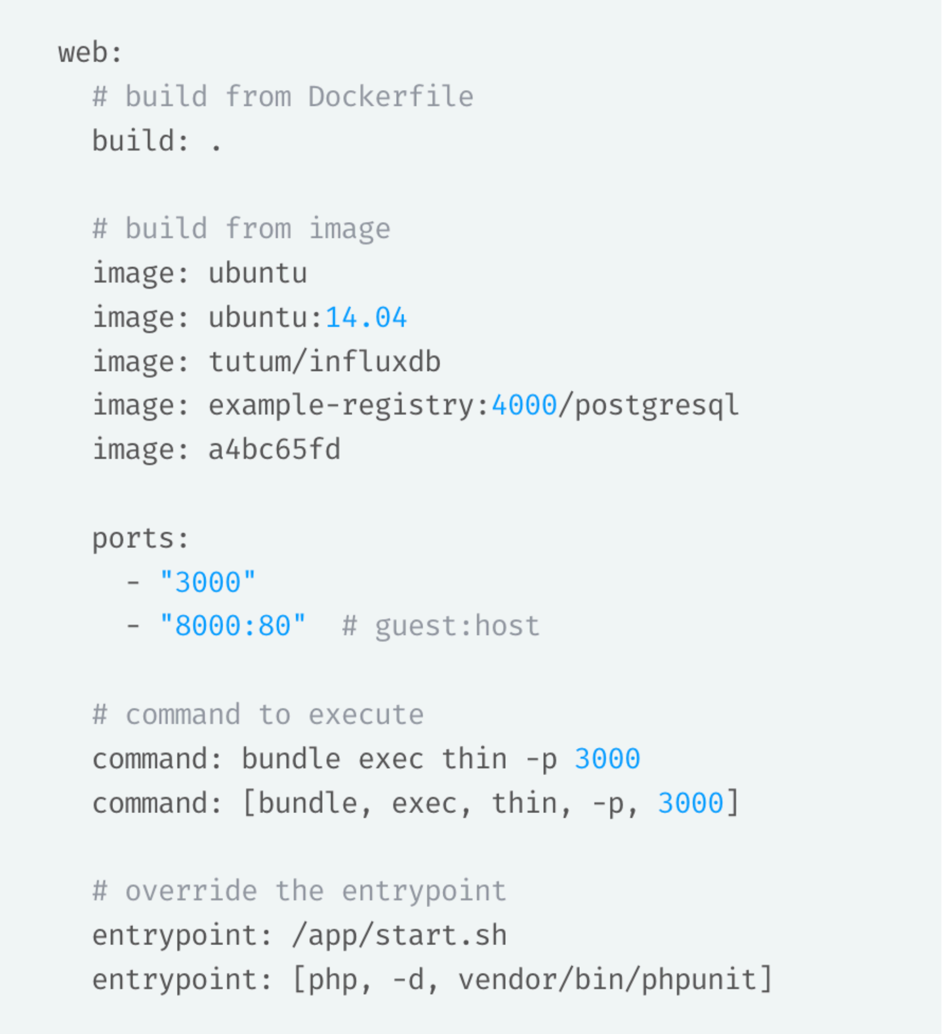
http://www.jayway.com/2015/03/21/a-not-very-short-introduction-to-docker/

Volumes

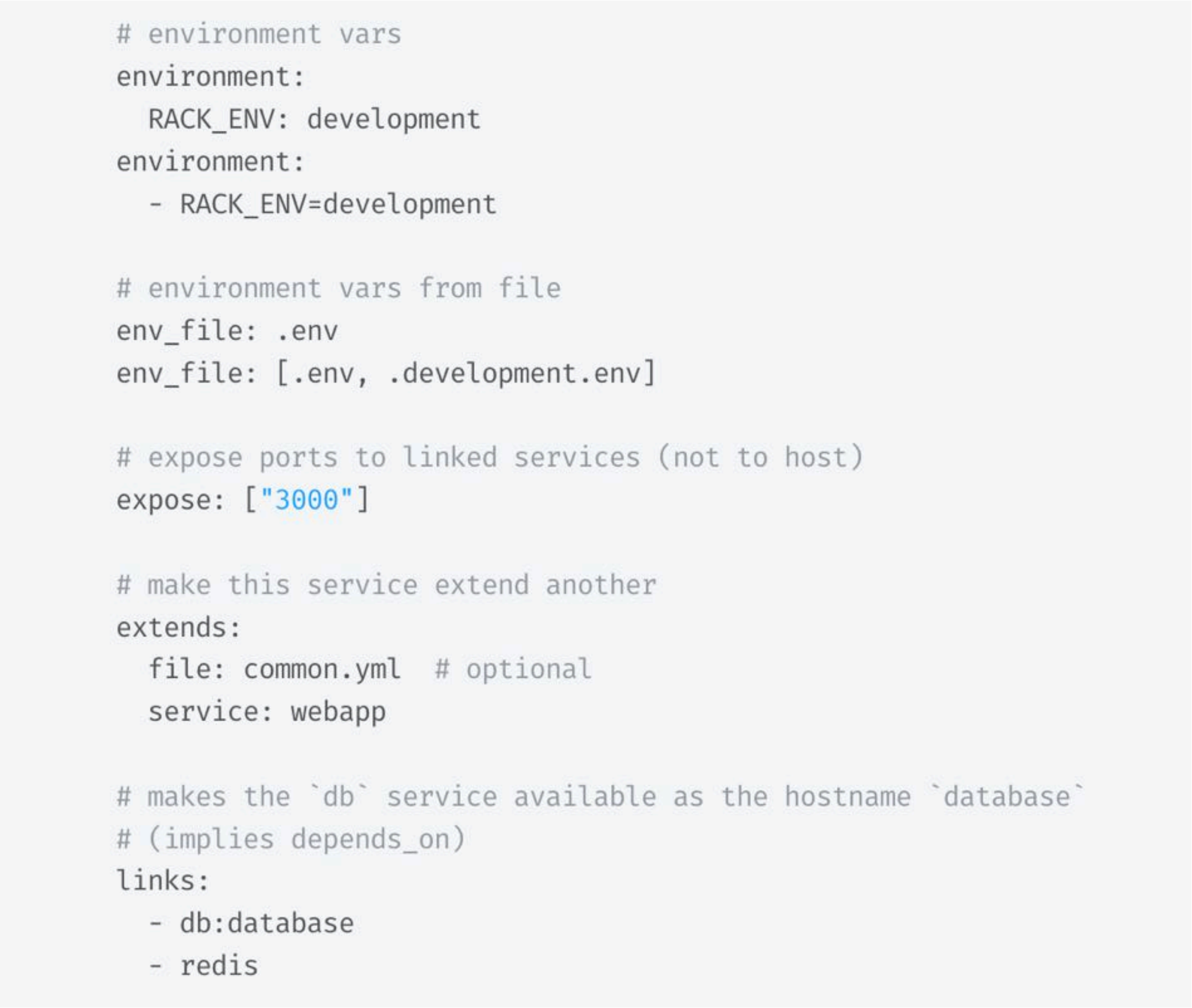
```
# Start a new nginx container with /var/log as a volume (created)
$ docker run  -v /var/log nginx

# Start a new nginx container with /var/log as a volume mapped to
/tmp on Host
$ docker run -v /tmp:/var/log nginx

# Start a db container
$ docker run -v /var/lib/postgresql/data --name mydb postgres

# Start a backup container with the volumes taken from the mydb
container
$ docker run --volumes-from mydb backup
```

http://www.jayway.com/2015/03/21/a-not-very-short-introduction-to-docker/

Docker-compose (v1)

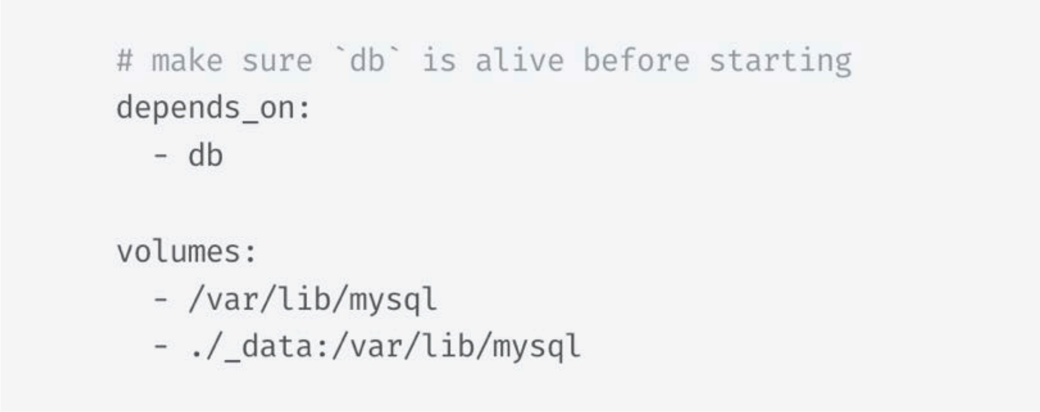


```
web:
  # build from Dockerfile
  build: .

  # build from image
  image: ubuntu
  image: ubuntu:14.04
  image: tutum/influxdb
  image: example-registry:4000/postgresql
  image: a4bc65fd

  ports:
    - "3000"
    - "8000:80"  # guest:host

  # command to execute
  command: bundle exec thin -p 3000
  command: [bundle, exec, thin, -p, 3000]

  # override the entrypoint
  entrypoint: /app/start.sh
  entrypoint: [php, -d, vendor/bin/phpunit]
```

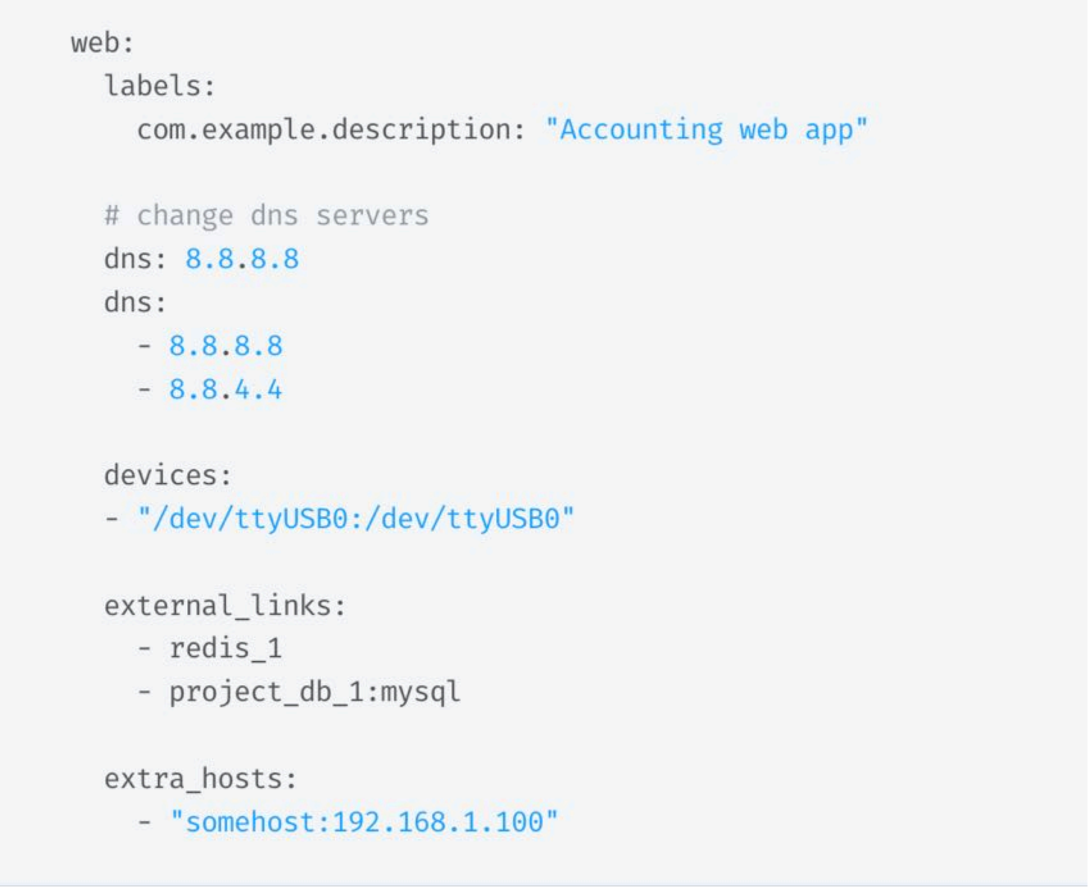


```
# environment vars
environment:
  RACK_ENV: development
environment:
  - RACK_ENV=development

# environment vars from file
env_file: .env
env_file: [.env, .development.env]

# expose ports to linked services (not to host)
expose: ["3000"]

# make this service extend another
extends:
  file: common.yml  # optional
  service: webapp

# makes the `db` service available as the hostname `database`
# (implies depends_on)
links:
  - db:database
  - redis
```

Docker-compose (v1)



```
# make sure `db` is alive before starting
depends_on:
  - db

volumes:
  - /var/lib/mysql
  - ./_data:/var/lib/mysql
```



```
web:
  labels:
    com.example.description: "Accounting web app"

  # change dns servers
  dns: 8.8.8.8
  dns:
    - 8.8.8.8
    - 8.8.4.4

  devices:
  - "/dev/ttyUSB0:/dev/ttyUSB0"

  external_links:
    - redis_1
    - project_db_1:mysql

  extra_hosts:
    - "somehost:192.168.1.100"
```