



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea



Escuela Universitaria
de Ingeniería
Vitoria-Gasteiz

Ingeniaritzako
Unibertsitate Eskola
Vitoria-Gasteizko

Vagrant y Docker, 2017

Toma de contacto

Pablo González Nalda

Depto. de Lenguajes y Sistemas Informáticos

Febrero de 2017



Modificado el 27 de febrero de 2017

Contenidos de la presentación

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

- 1 Objetivos del taller
- 2 Automatización de la virtualización
- 3 Ventajas de la gestión con máquinas virtuales
- 4 chroot
- 5 Mecanismos de aislamiento
- 6 Docker
- 7 Ventajas de Docker frente a MV

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

- 1 **Objetivos del taller**
- 2 Automatización de la virtualización
- 3 Ventajas de la gestión con máquinas virtuales
- 4 chroot
- 5 Mecanismos de aislamiento
- 6 Docker
- 7 Ventajas de Docker frente a MV

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

En este taller vamos a ver las potencialidades de Vagrant y Docker, que nos permiten:

- Virtualización.
- Automatización.

El ejemplo está basado en

<https://www.sitepoint.com/vagrantfile-explained-setting-provisioning-shell>

con los ejemplos en <https://github.com/sitepoint-editors/vagrant-base-config>

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

- 1 Objetivos del taller
- 2 Automatización de la virtualización
- 3 Ventajas de la gestión con máquinas virtuales
- 4 chroot
- 5 Mecanismos de aislamiento
- 6 Docker
- 7 Ventajas de Docker frente a MV

Automatización de la virtualización

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

Gestión automatizada de máquinas virtuales mediante `scripts` con Vagrant, que nos permite:

- Crear y destruir máquinas virtuales a través de un fichero de configuración `Vagrantfile`.
- Se usan máquinas genéricas preparadas como base.
- Al arrancar se ***provisionan*** con otro `script`, que instala, configura y ejecuta los programas adecuados para dar el servicio objetivo de la máquina.

```
vagrant up  
vagrant ssh  
vagrant destroy
```

Se puede comprobar que son sistemas distintos con `uname` dentro y fuera de la máquina. La virtualización es a nivel hardware.

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

- 1 Objetivos del taller
- 2 Automatización de la virtualización
- 3 Ventajas de la gestión con máquinas virtuales
- 4 chroot
- 5 Mecanismos de aislamiento
- 6 Docker
- 7 Ventajas de Docker frente a MV

Ventajas de la gestión con máquinas virtuales

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

Gestionar un sistema con máquinas virtuales nos permite:

- diferenciar los sistemas y los recursos necesarios para proporcionar un servicio: cada sistema es una simulación de ordenador, un SO independiente.
- por lo que un problema en una MV sólo va a afectar a un servicio
- la creación es repetible y automatizable.

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

- 1 Objetivos del taller
- 2 Automatización de la virtualización
- 3 Ventajas de la gestión con máquinas virtuales
- 4 chroot
- 5 Mecanismos de aislamiento
- 6 Docker
- 7 Ventajas de Docker frente a MV

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

`chroot` es un mecanismo que ejecuta un programa cambiando la raíz de su sistema de ficheros a un directorio, por lo que ese programa puede ser el que arranque un sistema completo.

Sólo compartirá con el anterior el sub-árbol de directorios y el kernel.

Se pueden montar los subdirectorios especiales como `/proc`

Ejemplo de uso de chroot

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

Como root

```
mkdir jaula
cd jaula/
cp /bin/busybox .
./busybox ls
./busybox uname -a
ldd busybox
mkdir proc dev run var
mount --bind /proc proc # directorio /proc disponible en la
jaula
mount --bind /dev dev
chroot /home/pablo/jaula ./busybox ash
# En otra terminal
sudo ls -ld /proc/18582/root
lrwxrwxrwx 1 root root 0 abr 20 19:56
/proc/18582/root -> /home/pablo/jaula
```

Más ejemplos en <http://www.cyberciti.biz/faq/unix-linux-chroot-command-examples-usage-syntax/>

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

- 1 Objetivos del taller
- 2 Automatización de la virtualización
- 3 Ventajas de la gestión con máquinas virtuales
- 4 chroot
- 5 Mecanismos de aislamiento
- 6 Docker
- 7 Ventajas de Docker frente a MV

Mecanismos de aislamiento: *Namespaces*

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

<https://docs.docker.com/engine/understanding-docker/>

Docker usa diferentes mecanismos proporcionados por el kernel y contruidos sobre él. Usa los *namespaces* para crear entornos aislados en el sistema, los contenedores. Cada elemento del contenedor se ejecuta en su propio *espacio de nombres* y no tiene acceso al exterior.

Algunos de los espacios de nombres que usa Docker son:

- **pid** namespace: Process isolation (PID: Process ID).
- **net** namespace: Managing NETwork interfaces.
- **ipc** namespace: Managing access to IPC resources (IPC: InterProcess Communication).
- **mnt** namespace: Managing MouNT-points .
- **uts** namespace: Isolating kernel and version identifiers. (UTS: Unix Timesharing System).

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

Los `cgroups` o *grupos de control* controlan la cantidad de recursos que consume el contenedor. De esta forma los contenedores comparten los recursos de hardware y tienen un límite, por ejemplo de memoria disponible. Controlan:

- Limitación de recursos.
- Control de prioridades.
- Contabilidad.
- Control de grupos de procesos (congelar y rearrancar).

Mecanismos de aislamiento *Union file systems*

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

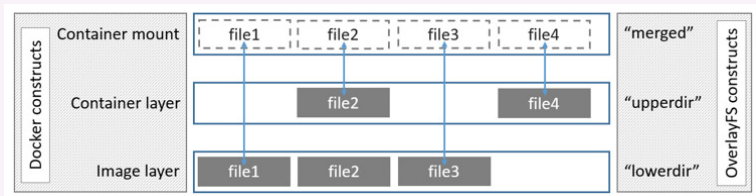
Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

Un sistema de ficheros *Union* es el que compone, una la información de varios sistemas de ficheros (las capas) para obtener uno. Esas capas son las fases de creación del sistema de ficheros del contenedor. Las capas pueden ser compartidas por varios contenedores (como el sistema operativo base).

```
ls -l /var/lib/docker/overlay/
```



<https://docs.docker.com/engine/userguide/storagedriver/overlayfs-driver/>

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

- 1 Objetivos del taller
- 2 Automatización de la virtualización
- 3 Ventajas de la gestión con máquinas virtuales
- 4 chroot
- 5 Mecanismos de aislamiento
- 6 Docker
- 7 Ventajas de Docker frente a MV

¿Qué es Docker?

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

Docker es un mecanismo para aislar, gestionar y modularizar la ejecución de un servicio, sus procesos y recursos necesarios, como almacenamiento, redes y otros dispositivos.

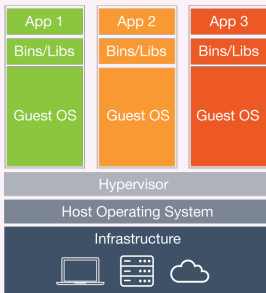


Figura: Virtualización a nivel de hardware

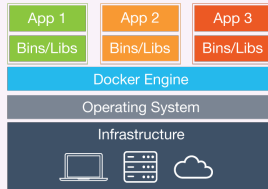


Figura: Contenedores

Estructura de un contenedor

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

Un contenedor comparte el kernel con el resto de procesos del sistema, puesto que el contenedor aísla servicios formados por varios procesos que colaboran. Se puede distinguir

imagen el conjunto de capas de sólo lectura que forman el sistema de ficheros

contenedor una imagen con una capa de lectura y escritura

Cuando hablamos de SO, pensamos en el conjunto de ficheros y mecanismos básicos para arrancar en el contenedor los servicios necesarios para un programa estándar de ese sistema. Sin embargo, el contenedor no arranca ningún proceso.

Estructura de un contenedor

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

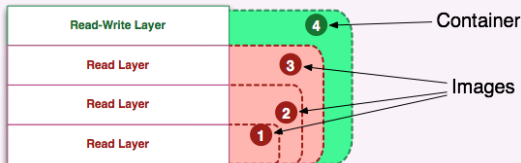
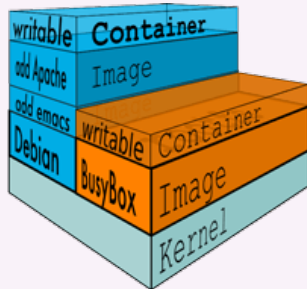
Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

Cada capa (*image-layer*) es un sistema de ficheros de sólo lectura, y es el resultado de una fase de instalación de software. Las capas se unen con un sistema de ficheros de UnionFS. Cuando arranca el contenedor se añade una capa vacía de escritura.



<http://merrigrove.>

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

Docker ejecuta un programa confinando ese proceso en

- un juego de PIDs separados,
- con sistemas de ficheros propio,
- volúmenes propios montados
- y acceso a una red definida por software.

docker pull ros

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

```
$ docker pull ros
Using default tag: latest
latest: Pulling from library/ros
203137e8afd5: Pull complete
2ff1bbbe9310: Pull complete
933ae2486129: Pull complete
a3ed95caeb02: Pull complete
efa7035358b8: Pull complete
492b493cc7ff: Pull complete
b95358183d2d: Pull complete
103c3645b87a: Pull complete
1cd6ee992a45: Pull complete
1fc2ba02a5ca: Pull complete
6eec5fc93cc9: Pull complete
f580de0c172a: Pull complete
Digest: sha256:078
      fbd221da8a3126eff2e2836572e38ef94631a1017568b86
Status: Downloaded newer image for ros:latest
```

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

```
$ docker run -it bitnami/minideb touch hola
Unable to find image 'bitnami/minideb:latest' locally
latest: Pulling from bitnami/minideb
15abd30c6809: Pull complete
Digest: sha256:07dbd4421b362a5d33665b868fb0dde89e06437905e04cc85f00e6bb862d

Status: Downloaded newer image for bitnami/minideb:latest

$ docker images # sistema de ficheros en capas + def. de recursos
$ docker ps     # instancias en funcionamiento, contenedores
$ docker ps -a  # instancias en funcionamiento y paradas
```

Otras instrucciones básicas

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

```
$ docker image inspect bitnami/minideb
$ docker run -it bitnami/minideb bash      # comando en
      instancia nueva
# si exit, se para
# si CTRL-PQ se desconecta de la terminal
$ docker inspect eloquent_chandrasekhar
$ docker start -ia eloquent_chandrasekhar # reiniciar
      instancia si parada
$ docker exec -it eloquent_chandrasekhar bash # otro
      proceso en el mismo CT si arrancado
$ docker rm eloquent_chandrasekhar        # borrar
      instancia
$ docker rmi ros # borrar imagen
```

docker inspect ubuntu

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

```
{
  "Id": "sha256:0109b6d281310b5f0f0ba41cd16cb04341f1eb4ced6851e869",
  "RepoTags": [
    "ubuntu:latest"
  ],
  "Parent": "sha256:800b0e0f49b20661fa41c19d1fb4cc637525a8a88305e",
  "Created": "2015-08-20T20:21:15.767240511Z",
  "Container": "74bb7db8d212f77ab6e54d2c60533f641de8c91e7ef343b88a146",
  "ContainerConfig": {
    "Hostname": "e611e15f9c9d",
    "Domainname": "",
    "User": "",
    "AttachStdin": false,
    "AttachStdout": false,
    "AttachStderr": false,
    "Tty": false,
    "Env": null,
    "Cmd": [
      "/bin/sh",
      "-c",
      "#(nop) CMD [\"/bin/bash\"]"
    ],
    "Image": "d74508fb6632491cea586dfc5274cd6fdfedee309ecdcbc2bf5cb82",
    "Volumes": null,
    "WorkingDir": "",
    "Entrypoint": null,
    "Labels": {}
  },
}
```


Red y puntos de montaje en Docker

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

Las redes en Docker se crean y destruyen por software. Los contenedores se pueden conectar a redes al crearlos y posteriormente.

```
$ docker network ls
```

NETWORK ID	NAME	DRIVER
7fca4eb8c647	bridge	bridge
9f904ee27bf5	none	null
cf03ee007fb4	host	host

```
# Punto de montaje de mi directorio en un directorio del contenedor
```

```
$ docker run -it -v /home/pablo/catkin_ws/:/catkin_ws ros
```

Red y puntos de montaje en Docker

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

```
$ docker volume ls
```

```
DRIVER
```

```
VOLUME NAME
```

```
local
```

```
5a8b7382c335eb2cdcc3.....6
```

```
a8ede36b7c1375ffd64d2eb20a70
```

```
local
```

```
b189be82eeee02085c93.....13122
```

```
b7d97d61805a1012c9aa2826
```

```
$ docker volume inspect 5a8b7382c335eb2cd...2876a8eeb20a70
```

```
[
```

```
{
```

```
"Driver": "local",
```

```
"Labels": null,
```

```
"Mountpoint": "/var/lib/docker/volumes/5
```

```
a8b7382c335eb2cdcc3ba2d...36
```

```
b7c1375ffd64d2eb20a70/_data",
```

```
"Name": "5
```

```
a8b7382c335eb2cdcc3ba2d0569966b2876a8ede36b7c1375f
```

```
",
```

```
"Options": {},
```

```
"Scope": "local"
```

```
}
```

```
]
```

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

- 1 Objetivos del taller
- 2 Automatización de la virtualización
- 3 Ventajas de la gestión con máquinas virtuales
- 4 chroot
- 5 Mecanismos de aislamiento
- 6 Docker
- 7 Ventajas de Docker frente a MV

Ventajas de Docker frente a máquinas virtuales

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

Gestionar un sistema con Docker tiene ciertas ventajas frente a usar máquinas virtuales:

- Usa menos recursos, no necesita todos los servicios de un SO estándar por lo que arranca mucho más rápidamente y con menos memoria.
- No emula hardware virtual sino que usa el mismo kernel que el sistema “anfitrión”, es mucho más rápido en ejecución.
- Los recursos están virtualizados pero se pueden compartir fácilmente entre contenedor y SO base.

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

Ejemplos de Dockerfiles:

<https://github.com/PabloGN/Docker-raspbian-ros-indigo>

Contenedores para Raspberry Pi y forma de uso:

<https://hub.docker.com/r/pablogn/>

Dockerfile # Add arduino to ros-kinetic

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

```
FROM ros
MAINTAINER Pablo Gonzalez Nalda pablo.gonzalez@ehu.eus
USER root
RUN apt-get update && \
    apt-get install --no-install-recommends -y arduino \
        arduino-core arduino-mk \
    ros-kinetic-rosserial-arduino ros-kinetic-rosserial && \
    mkdir -p /usr/share/arduino/ && \
    cd /usr/share/arduino/ && \
    rm -rf ros_lib /var/lib/apt/lists/* && \
    bash -c "source /opt/ros/kinetic/setup.sh && rosruntime \
        rosserial_arduino make_libraries.py . " && \
    useradd arduino && echo 'arduino:arduino' | chpasswd && \
        echo "arduino ALL=(ALL) NOPASSWD: ALL" >> /etc/\
        sudoers && mkdir -p /home/arduino && chown arduino:\
        arduino /home/arduino

USER arduino
WORKDIR /home/arduino/      # setup entrypoint
COPY ./rep.sh rep.sh
ENTRYPOINT ["./rep.sh"]
CMD ["/bin/bash"]
```

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

- 1 Objetivos del taller
- 2 Automatización de la virtualización
- 3 Ventajas de la gestión con máquinas virtuales
- 4 chroot
- 5 Mecanismos de aislamiento
- 6 Docker
- 7 Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

Ejemplo:

```
vagrant up
```

```
vagrant provision si no lo ha hecho al arrancar
```

```
vagrant ssh
```

```
http://localhost:8931/gitlab/
```

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

- 1 Objetivos del taller
- 2 Automatización de la virtualización
- 3 Ventajas de la gestión con máquinas virtuales
- 4 chroot
- 5 Mecanismos de aislamiento
- 6 Docker
- 7 Ventajas de Docker frente a MV

¿Más preguntas?

CONTENIDOS

Objetivos del taller

Automatización de la virtualización

Ventajas de la gestión con máquinas virtuales

chroot

Mecanismos de aislamiento

Docker

Ventajas de Docker frente a MV

Ejemplo de Vagrant + Docker

¿Más preguntas?

¿Más preguntas?



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea



Escuela Universitaria
de Ingeniería
Vitoria-Gasteiz

Ingeniaritzako
Unibertsitate Eskola
Vitoria-Gasteizko

Vagrant y Docker, 2017

Toma de contacto

Pablo González Nalda

Depto. de Lenguajes y Sistemas Informáticos

Febrero de 2017



Modificado el 27 de febrero de 2017