

Grado en Ingeniería Informática

Sistemas Industriales

Docker

Pablo Casado

pcasado@dtic.ua.es

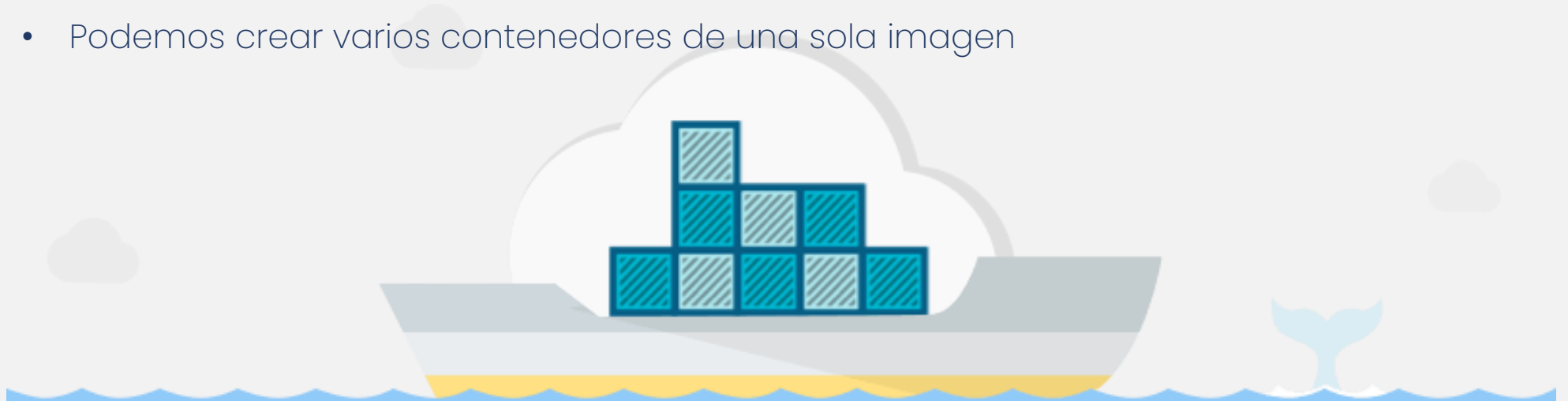
Departamento de Tecnología Informática y Computación

2022 - 2023

4. Docker Container

- 1. Introducción
- 2. Instalación
- 3. Docker Images
- 4. **Docker Container**
- 5. Docker Volumes
- 6. Docker Network
- 7. Docker Compose
- 8. Docker Registry

- Docker Containers
- Instancia en ejecución de una imagen
- Son capas temporales
- Instancias con capa de lectura y escritura
- Podemos crear varios contenedores de una sola imagen



4. Docker Container

- 1. Introducción
- 2. Instalación
- 3. Docker Images
- 4. **Docker Container**
- 5. Docker Volumes
- 6. Docker Network
- 7. Docker Compose
- 8. Docker Registry

- Listar contenedores: `docker ps` / `docker ps -a`
- Opciones mas usuales run:
 - `-d` / Background
 - `-p 8080:8080` / Mapeo de puertos de la máquina física al contenedor
 - Jenkins // `docker run -p 8080:8080 -p 50000:50000 --restart=on-failure jenkins/jenkins`



4. Docker Container

- 1. Introducción
- 2. Instalación
- 3. Docker Images
- 4. **Docker Container**
- 5. Docker Volumes
- 6. Docker Network
- 7. Docker Compose
- 8. Docker Registry

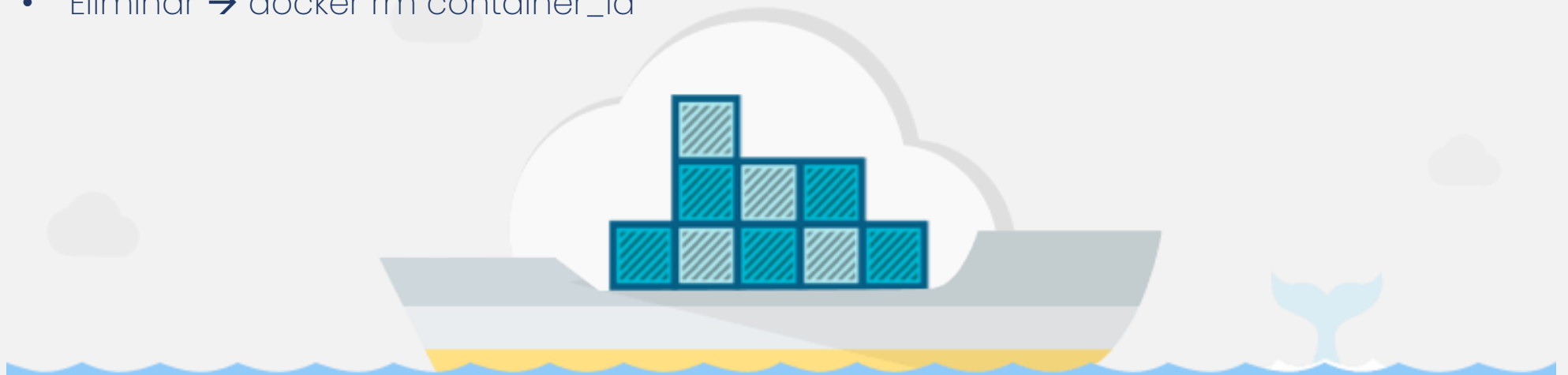
- Ejercicio:
 - *Install MySQL client and test*



4. Docker Container

- 1. Introducción
- 2. Instalación
- 3. Docker Images
- 4. **Docker Container**
- 5. Docker Volumes
- 6. Docker Network
- 7. Docker Compose
- 8. Docker Registry

- Administración de contenedores de Docker
- Construir → `docker build`
- Arrancar un container → `docker run`
- Parar → `docker stop container_id`
- Eliminar → `docker rm container_id`



5. Docker Volumes

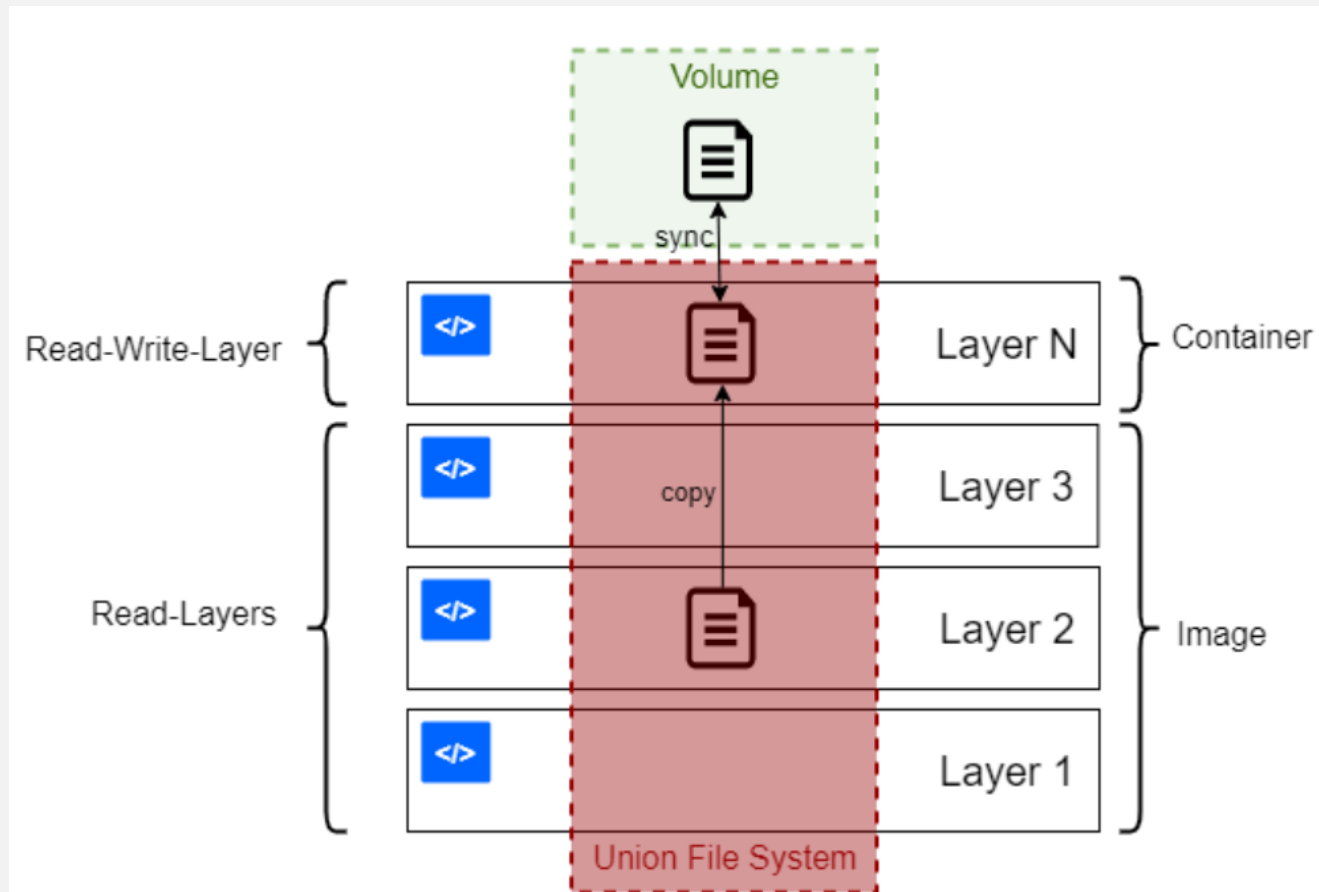
- 1. Introducción
- 2. Instalación
- 3. Docker Images
- 4. Docker Container
- 5. Docker Volumes**
- 6. Docker Network
- 7. Docker Compose
- 8. Docker Registry

- Los contenedores de Docker son usados para ejecutar apps en entornos aislados
- Todos los cambios en el contenedor son perdidos al pararlo
- Si queremos mantener datos podemos utilizar Docker Volumes



5. Docker Volumes

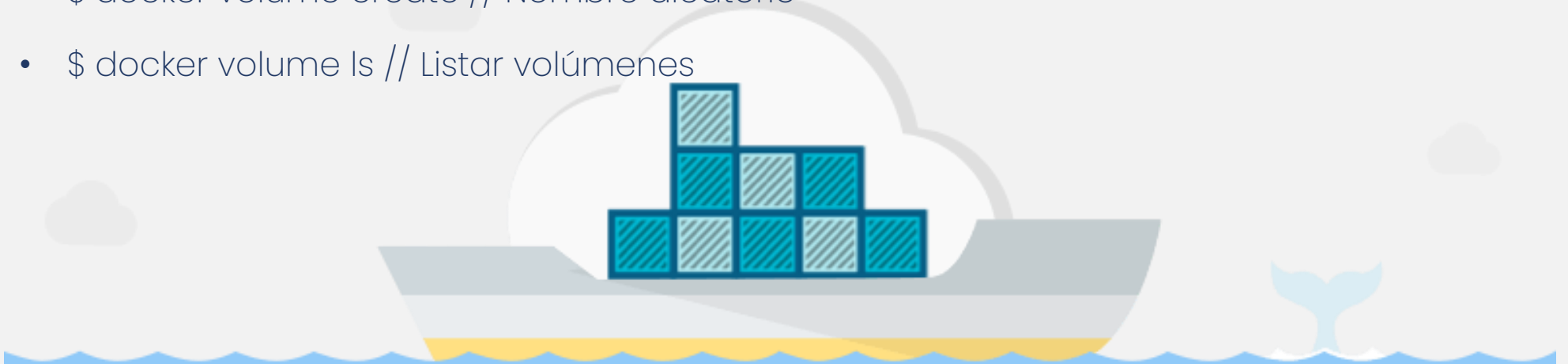
- 1. Introducción
- 2. Instalación
- 3. Docker Images
- 4. Docker Container
- 5. Docker Volumes
- 6. Docker Network
- 7. Docker Compose
- 8. Docker Registry



5. Docker Volumes

- 1. Introducción
- 2. Instalación
- 3. Docker Images
- 4. Docker Container
- 5. Docker Volumes**
- 6. Docker Network
- 7. Docker Compose
- 8. Docker Registry

- Ciclo de vida más largo que los contenedores
- Se pueden compartir entre contenedores
- Podemos crear un volumen con un id específico o indicarlo
- `$ docker volume create id_name_volume // Nombre especificado`
- `$ docker volume create // Nombre aleatorio`
- `$ docker volume ls // Listar volúmenes`



5. Docker Volumes

- 1. Introducción
- 2. Instalación
- 3. Docker Images
- 4. Docker Container
- 5. Docker Volumes**
- 6. Docker Network
- 7. Docker Compose
- 8. Docker Registry

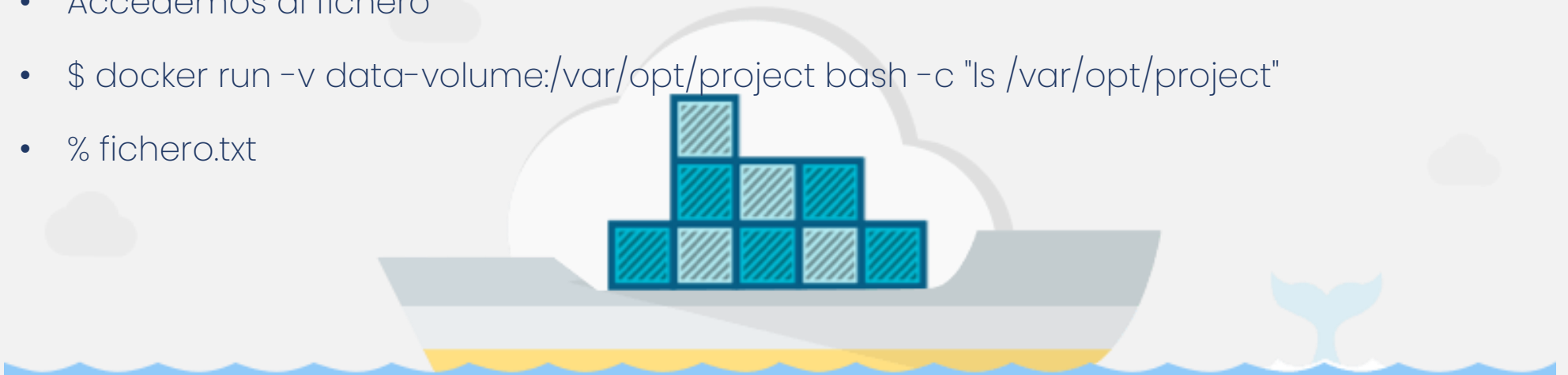
- `$ docker volume inspect id_name_volume //` Inspeccionar los datos
- `$ docker volume rm id_name_volume //` Eliminar volumen id_name_volume
- `$ docker volume prune //` Eliminar volúmenes



5. Docker Volumes

- 1. Introducción
- 2. Instalación
- 3. Docker Images
- 4. Docker Container
- 5. Docker Volumes
- 6. Docker Network
- 7. Docker Compose
- 8. Docker Registry

- Montar un volumen
- `$ docker run -v data-volume:/var/opt/project bash:latest \`
- `bash -c "ls /var/opt/project"`
- Generamos un fichero.txt en la ruta especificada
- Accedemos al fichero
- `$ docker run -v data-volume:/var/opt/project bash -c "ls /var/opt/project"`
- `% fichero.txt`



5. Docker Volumes

1. Introducción

2. Instalación

3. Docker Images

4. Docker Container

5. Docker Volumes

6. Docker Network

7. Docker Compose

8. Docker Registry

The input to `-mount` is a string of key-value pairs, separated by commas. Here we've set:

- **type** – as volume to indicate a volume mount
- **src** – to the name of the volume, though this could have been a source directory if we'd been making a bind mount
- **dst** – as the destination mount point in the container
- **volume-driver** – the local driver in this case
- **readonly** – to make this mount read-only; we could have chosen `rw` for read/write



5. Docker Volumes

- 1. Introducción
- 2. Instalación
- 3. Docker Images
- 4. Docker Container
- 5. Docker Volumes
- 6. Docker Network
- 7. Docker Compose
- 8. Docker Registry

- `$ docker run --mount \`
- `'type=volume,src=data-volume,\`
- `dst=/var/opt/project,volume-driver=local,\`
- `readonly' \`
- `bash -c "ls /var/opt/project"`

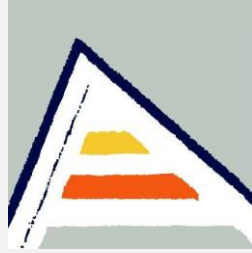


5. Docker Volumes

- 1. Introducción
- 2. Instalación
- 3. Docker Images
- 4. Docker Container
- 5. Docker Volumes**
- 6. Docker Network
- 7. Docker Compose
- 8. Docker Registry

- Conectar a un volumen crea una conexión a largo plazo entre el contenedor y el volumen
- Incluso cuando se cierra el contenedor
- Podemos vincular volúmenes entre contenedores en ejecución
 - `$ -volumes-from`





Grado en Ingeniería Informática

Sistemas Industriales

Docker

Pablo Casado

pcasado@dtic.ua.es

Departamento de Tecnología Informática y Computación

2022 - 2023