Grado en Ingeniería Informática

Sistemas Industriales

Docker

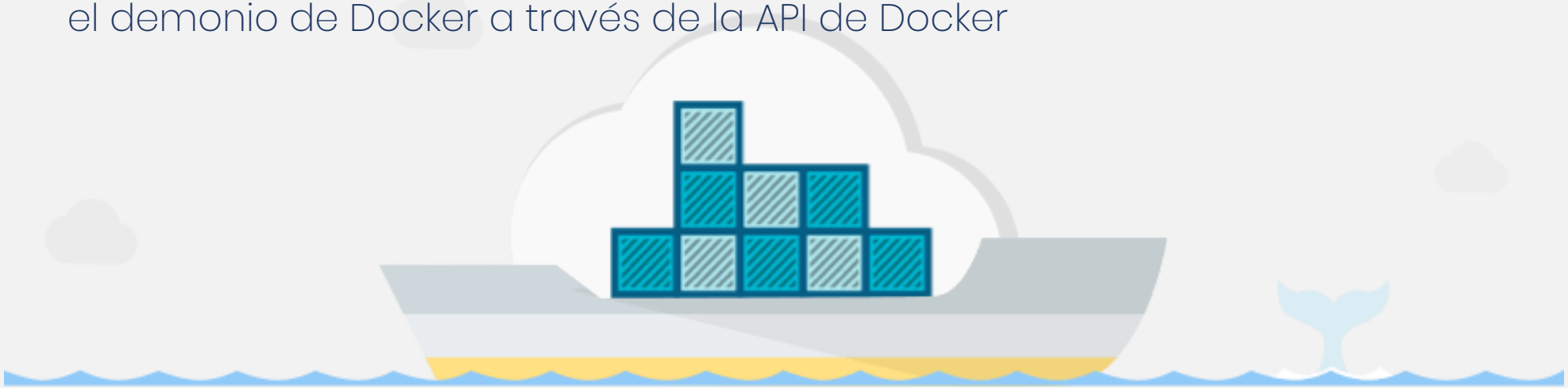
Pablo Casado
pcasado@dtic.ua.es

Departamento de Tecnología Informática y Computación

2022 - 2023

7. Docker Compose

- Creada por el equipo de Docker para facilitar la tarea de crear y configurar varios contenedores en un entorno de desarrollo
- Toma como entrada un fichero de configuración YAML
- Crea los recursos (contenedores, volúmenes, redes, etc.) comunicándose con el demonio de Docker a través de la API de Docker



7. Docker Compose

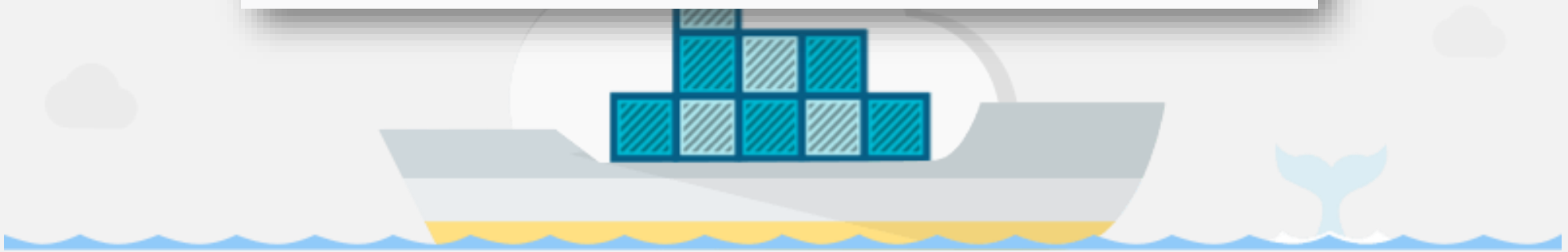
- Proyecto de código abierto https://github.com/docker/compose
- Responsable de la orquestación rápida de los clústeres de contenedores de Docker
- Dockerfile → contenedor de aplicación independiente
- Docker-Compose → conjunto de contenedores asociados como un proyecto

7. Docker Compose

- Conceptos:
- **Service**: contenedor para una aplicación que en realidad puede incluir varias instancias de contenedor que ejecutan la misma imagen
- **Project**: unidad de negocio completa que consta de un conjunto de contenedores de aplicaciones asociados, definido en el archivo docker-compose.yml

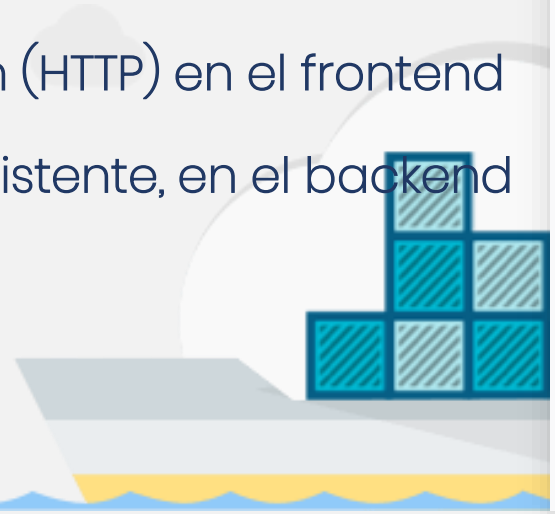
7. Docker Compose


```
(External user) --> 443 [frontend network]
                          |
                +--------------------+
                |  frontend service  |...ro...<HTTP configuration>
                |      "webapp"      |...ro...<server certificate> #secured
                +--------------------+
                          |
                     [backend network]
                          |
                +--------------------+
                |  backend service   |  r+w   ____________________
                |     "database"     |=======( persistent volume )
                +--------------------+        \__________________/
```



7. Docker Compose

- 2 services: imágenes de Docker
  - webapp
  - database
- 1 secret (HTTPS) en el frontend
- 1 configuración (HTTP) en el frontend
- 1 volumen persistente, en el backend
- 2 networks



```yaml
services:
  frontend:
    image: awesome/webapp
    ports:
      - "443:8043"
    networks:
      - front-tier
      - back-tier
    configs:
      - httpd-config
    secrets:
      - server-certificate

  backend:
    image: awesome/database
    volumes:
      - db-data:/etc/data
    networks:
      - back-tier

volumes:
  db-data:
    driver: flocker
    driver_opts:
      size: "10GiB"

configs:
  httpd-config:
    external: true

secrets:
  server-certificate:
    external: true

networks:
  # The presence of these objects is sufficient to define them
  front-tier: {}
  back-tier: {}
```

7. Docker Compose

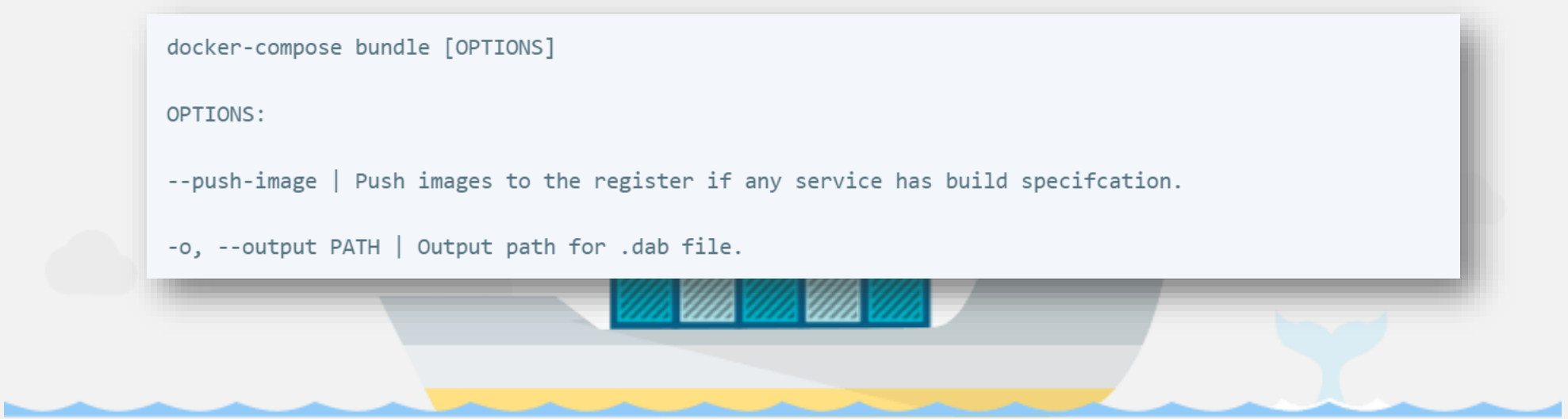
- Principales opciones:
- **Build**: crear servicios especificados en el archivo docker-compose.yml

```
OPTIONS:

--compress | Command line flag to compress the build context, Build context is nothing but a directory

--force-rm | Remove any intermediate container while building.

--no-cache | Build images without using any cached layers from previoud builds.

--pull | Allways pull newer version of the base image.

-m, --memory | Set memory limit for the container used for building the image.

--parallel | Exploit go routines to parallely build images, As docker daemon is written in go.

--build-arg key=val | Pass any varaible to the dockerfile from the command line.
```

7. Docker Compose

- Principales opciones:
- **Bundle**: usado para crear paquetes de aplicaciones distribuidas

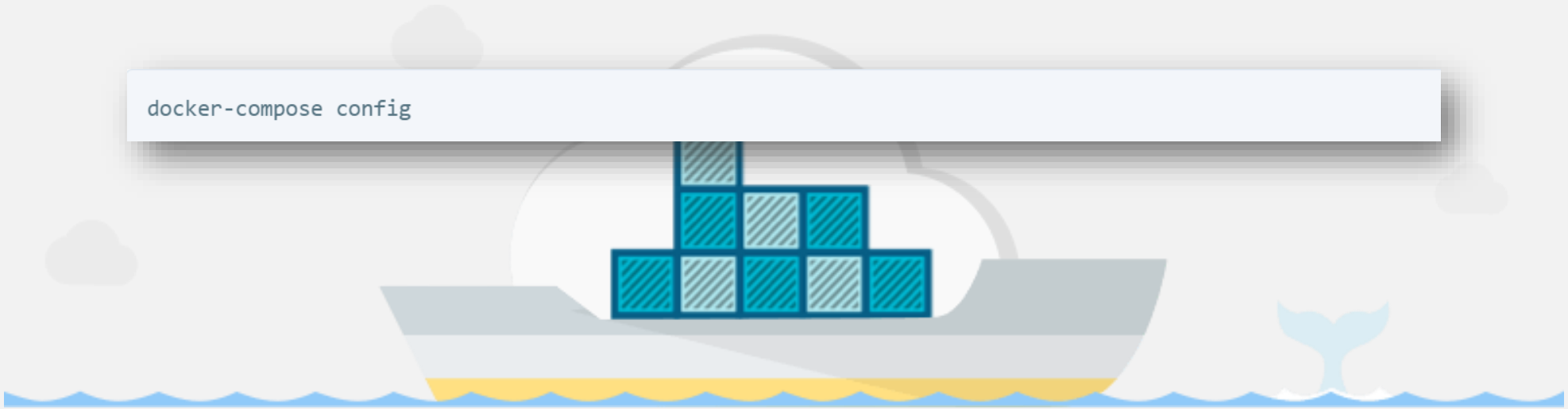
```
docker-compose bundle [OPTIONS]

OPTIONS:

--push-image | Push images to the register if any service has build specifcation.

-o, --output PATH | Output path for .dab file.
```



7. Docker Compose

- Principales opciones:
- **Config**: usado para validar el fichero docker-compose.yml
- Debe ser ejecutado donde se encuentre el fichero

```
docker-compose config
```



7. Docker Compose

- Principales opciones:
- Up: crea e inicia los recursos especificados

```
docker-compose up [OPTIONS] [SERVICE...]

OPTIONS:

-d, --detach | Run containers in background.

--build | Always build images even if it exists.

--no-deps | Avoid creating any linked services.

--force-recreate | Force recreating containers even if specification is not changed.

--no-recreate | Do not recreate containers.

--no-build | Do not build any image even if it is missing.

--no-start | Just create the containers without starting them.

--scale SERVICE=NUM | Create multiple containers for a service.

-V, --renew-anon-volumes | Recreate anonymous volumes instead of getting data from previous ones.

Example:

docker-compose up -d        # Will run service containers in background
docker-compose up web       # Will start service web and server because of 'depends_on' field
docker-compose up server    # will start server service only.
```

7. Docker Compose

- Principales opciones:
- **Down**: detener y borrar los recursos especificados

```
docker-compose down [OPTIONS]

--rmi type | Remove images Type = all (Remove every image in the compose file), local (Remove images w

-v, --volumes | Remove named volumes except the external ones and also remove anonymous volumes

-t, --timeout TIMEOUT | Speficy shutdown time in seconds. (default = 10)

Example:

docker-compose down         # Will delete all containers of both web and server and no volume will be

docker-compose down -v      # Will also delete anonymous and data volumes.
```

7. Docker Compose

- **Scale**: escalar servicios especificados

```
docker-compose scale [SERVICE=NUM...]

Example:

docker-compose scale server=3 web=2
```

- **Start**: arranca contenedores creados

```
docker-compose stop [SERVICE...]

Example:

docker-compose stop         # Stop containers for every service.
docker-compose stop web     # Stop containers only for service web.
```

7. Docker Compose

- **Stop**: parada de contenedores corriendo.

```
docker-compose stop [SERVICE...]

Example:

docker-compose stop         # Stop containers for every service.
docker-compose stop web     # Stop containers only for service web.
```

Grado en Ingeniería Informática

Sistemas Industriales

Docker

Pablo Casado
pcasado@dtic.ua.es

Departamento de Tecnología Informática y Computación

2022 - 2023