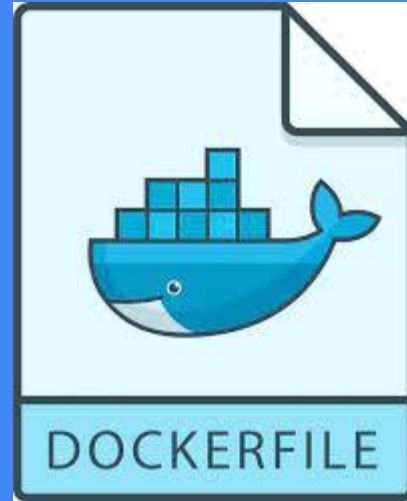


# Dockerfile

Taller Docker

Día 4



Por: Sandy Mérida

# Agenda

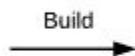
- Dockerfile
- Construcción de una imagen
- Eliminar/borrar una imagen
- Docker Hub
- Login Docker Hub
- Publicar image en Docker Hub
- Usemos nuestra image personalizada

# Dockerfile

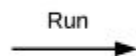
- Docker puede crear imágenes automáticamente leyendo las instrucciones de un archivo Dockerfile.
- Un Dockerfile es un archivo de texto
- Contiene los comandos que un usuario puede ejecutar en la terminal para ensamblar una imagen
- **docker build** puede crear una compilación automatizada que ejecute varias instrucciones de línea de comandos en sucesión.



Dockerfile



Docker  
Image



Docker  
Container

# Construcción de una imagen

```
$ docker build -t <NAME_IMAGE> .
```



```
$ docker build .
```

```
$ docker build -f /path/to/a/Dockerfile .
```

```
$ docker build -t <NAME_IMAGE> .
```

Fuente: <https://docs.docker.com/engine/reference/builder/>

# Eliminar/borrar una image

- No debe existir ningún contenedor creado con la imagen que se desea borrar
  - `docker stop <CONTAINER_NAME>`
  - `docker rm <CONTAINER_NAME>`
- Comando para eliminar image

**`$ docker rmi <REPOSITORY | IMAGE ID>`**

# Docker Hub

- Servicio de registro de repositorios
- Repositorio de imágenes de contenedores
- Proporcionado por Docker Inc.
- Acceso a repositorios públicos gratuitos
- Compartir imágenes
- Plan de suscripción para repositorios privados.



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





### Explore the world's largest container image repository

Easily search more than one million container images, including Certified and community-provided images.

[Explore](#)



### Share and store images in public or private repositories

Get access to free public repositories or choose a subscription plan for private repos.

[Get started](#)



### Become a Verified Publisher

A trusted way to run more technology in containers with certified infrastructure, containers and plugins.

[Sign up](#)

# Login Docker Hub

# Login Docker Hub desde terminal

**\$ docker login -u <USER>**

- USER es el que te registras en <https://hub.docker.com/>
- Luego te pide tu contraseña
- Al final veras el mensaje de inicio de sesión exitoso

```
Configure a credential helper to remove this warning. See  
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
```

```
Login Succeeded
```

# Publicar image en Docker Hub

# Publicar imagen en Docker Hub

1. Hacemos el login desde terminal
2. Preparamos nuestra imagen con la siguiente nomenclatura

**\$ docker tag <ID\_IMAGE> <USER>/<NAME>[tag]**

**o**

**\$ docker tag <ID\_IMAGE> <USER>/<NAME>[:v#]**

# Push image a Docker Hub

**\$ docker push <USER>/<NAME>[:v#]**

- Estamos empujando ó subiendo nuestra imagen personalizada a [Docker Hub](#)
- Si no haces pago de un plan, es de dominio público tu imagen
- OJO: ten cuidado con lo que almacena el código que agregaste, en especial con password, keys...

Usemos nuestra image personalizada

# Creación de contenedor

- Creamos un contenedor de nombre *servidor1* que se basa en una imagen personalizada
- Mapeamos los puertos, creamos un puente entre el puerto 3000 del contenedor con el puerto 8000 del host

```
$ docker run -it -d -p 8000:3000 --name=servidor1 <USER>/servidor_td:v1
```



# Anexo

- Repositorio oficial [COECYS202-TallerDocker](#)
- Puedes encontrar detalle documentado del contenido de este taller [aquí](#)