

# Laboratorio contenedores:

---

## Prerequisitos

Si no creasteis la máquina virtual ya en la anterior práctica, en [este enlace](#) tenéis los pasos a seguir para hacerlo.

---

## DOCKER

---

### Crear un Dockerfile:

- que tenga de imagen base la penúltima versión oficial de Node.js
- arranca node como proceso principal de la imagen
- añade el metadato maintainer con tu e-mail
- copia el Dockerfile en la ruta `/tmp`
- inserta el código que se os proporciona en el siguiente [enlace](#)
- instalar las dependencias del package.json
- cambia el puerto por defecto de nodejs por el puerto 2000. [Variable de entorno APP\_PORT]
- expón el puerto

Graba con asciinema los siguientes pasos:

```
asciinema rec <USER>-docker.cat
```

1. Construye la imagen docker con el nombre `<USER>_node`.
2. Lista todas las imágenes.
3. Ejecuta el comando que elimina todas las imagenes que están a "none".
4. Arranca un contenedor en modo detached usando la imagen que acabamos de crear, mapeando al puerto 8080 y asignándole el nombre `<USER>_container`
5. Valida que se ha desplegado la aplicación y es accesible usando el comando `wget -qO- <URL>/hostname`
6. Muestra el log del contenedor.
7. Entra al contenedor y ejecuta el comando `cat /tmp/Dockerfile`.
8. Saca todos los detalles del contenedor.
9. Lista todos los contenedores que hay en ejecución.
10. Elimina la imagen que acabamos de crear.

**Para verificar que se han grabado correctamente los pasos:**

```
asciinema play <USER>-docker.cat
```

## DOCKER-COMPOSE

---

Crea un fichero de configuración de docker-compose con los siguientes dos servicios:

- `<USER>_mysql:`
  - usar la última imagen oficial de MYSQL (especificando el número)
  - guardar todos los datos de `/var/lib/mysql` en un volumen llamado `<USER>-data`
  - inicializa la BBDD: en el directorio `/docker/app/sql` tenéis el script `sql`.
- `<USER>_app:`
  - usar la imagen generada o generarla con `compose`.
  - definir las variables de entorno necesarias para conectar a la BBDD MYSQL con usuario `ROOT`
  - definir el puerto `4000` para levantar el proceso `nodejs`
  - exponer el servicio en el rango de puertos `8080` y `8090`

Graba con `asciinema` los siguientes pasos:

```
asciinema rec <USER>-docker-compose.cat
```

1. Arranca los servicios en modo `detached`
2. Muestra solo las trazas del contenedor que contiene la aplicación `<USER>_app`
3. Demuestra que se ha desplegado la aplicación y es accesible usando el comando `wget -qO- <URL>/hostname`
4. Muestra el contenido de `docker-compose.yaml` usando el comando `cat`
5. Escala para tener dos contenedores `<USER>_app`
6. Demuestra que se ha desplegado una segunda instancia de la aplicación `wget -qO- <URL>/hostname`

## ENTREGA:

---

- `Dockerfile`
- `docker-compose.yaml`
- `<USER>-docker.cat`
- `<USER>-docker-compose.cat`