

Docker Básico

Arturo Silvelo

Try New Roads

Soluciones Redes

Descargar imágenes

```
docker pull ghcr.io/trynewroads/course-frontend:1.0.0  
docker pull ghcr.io/trynewroads/course-backend:1.0.0
```

Verificar que las imágenes se han descargado

```
docker image ls
```

1. Crea una red personalizada llamada `course-network` que permita la comunicación entre los contenedores.

```
docker network create course-network
```

Verifica que se ha creado y el rango de la subred creada

```
docker network ls  
docker inspect course-network
```

```
{  
  "Subnet": "172.22.0.0/16",  
  "Gateway": "172.22.0.1"  
}
```

2. Inicia un contenedor basado en la imagen `course-backend`, conéctalo a la red `course-network`, dale un nombre de host y al contenedor.

```
docker run
-d # Segundo plano
-p 3000:3000 # Conectar el puerto 3000 del host con el 3000 del contenedor
--network course-network # Asignar la red
--hostname course-backend # Nombre en la red
--name cb # Nombre del contenedor
ghcr.io/trynewroads/course-backend:1.0.0 # imagen usada
```

Comprobar la creación

```
docker ps
```

Comprobar la red

```
docker inspect course-network
```

```
"Networks": {  
  "course-network": {  
    "Gateway": "172.22.0.1",  
    "IPAddress": "172.22.0.2",  
    "DNSNames": [  
      "cb", # Resolución por el nombre del contenedor  
      "43bf280a0797", # Resolución por el id del contenedor  
      "course-backend" # Resolución por el hostname  
    ]  
  }  
}
```

Acceder servidor <http://localhost:3000/docs>

TODO API 1.0 OAS 3.0		
API para manejar tareas		
App ^		
GET	/api/healthcheck	Healthcheck de la API
tasks ^		
GET	/api/tasks	Obtener todas las tareas
POST	/api/tasks	Crear una nueva tarea
GET	/api/tasks/{id}	Obtener una tarea por ID
PUT	/api/tasks/{id}	Actualizar una tarea por ID
DELETE	/api/tasks/{id}	Eliminar una tarea por ID
PUT	/api/tasks/{id}/status	Actualizar el estado de una tarea por ID
PATCH	/api/tasks/{id}/status	Actualizar el estado de una tarea por ID
auth ^		
POST	/api/auth/login	Login de usuario
GET	/api/auth/me	Obtener el usuario logueado

3. Inicia un contenedor basado en la imagen `course-frontend` y conéctalo también a la red `course-network`.

Crear fichero de configuración nginx `default.conf.template`

```
server {
    listen 80;
    server_name localhost;
    root /usr/share/nginx/html;
    index index.html;

    location / {
        try_files $uri $uri/ /index.html;
    }

    location /api {
        proxy_pass http://course-backend:3000/api;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        # Headers para CORS si es necesario
        add_header Access-Control-Allow-Origin *;
        add_header Access-Control-Allow-Methods "GET, POST, PUT, DELETE, OPTIONS";
        add_header Access-Control-Allow-Headers "Content-Type, Authorization";
    }
}
```



```
docker run
-d # Segundo plano
-p 80:80 # Conectar el puerto 3000 del host con el 3000 del contenedor
-v ./default.conf.template:/etc/nginx/templates/default.conf.template # Configuración de nginx
--network course-network # Asignar la red
--hostname course-frontend # Nombre en la red
--name cf # Nombre del contenedor
ghcr.io/trynewroads/course-frontend:1.0.0 # imagen usada
```


Comprobar la creación

```
docker ps
```

Comprobar la red

```
docker inspect course-network
```

Acceder web <http://localhost> con el usuario `admin` y contraseña `12345678`


×

Nueva Tarea

Crear Tarea

Lista de Tareas

[Crear Tarea](#)

Id: #001

Tarea 1

Descripción tarea 1

☐ Pendiente



Id: #002

Título 2

Descripcion 2

☐ Pendiente



4. Asegúrate de que la aplicación frontend pueda acceder al backend a través del nombre del servicio (`course-backend`) en lugar de una dirección IP.

- Usuario `admin`, contraseña: `12345678`
- Acceder servidor <http://localhost:3000/docs>
- Acceder cliente <http://localhost:80>
- Crear una tarea
- Ver el listado de tareas