

PS C:\dev> docker network ls				
NETWORK ID	NAME	DRIVER	SCOPE	
d4cd8b5ba890	bridge	bridge	local	
b326b2ae67bd	host	host	local	
179faac8beb7	none	null	local	

docker network Is

Crear una nueva red docker network create MI_RED

Crear el primer contenedor y asignarlo a "MI_RED" docker run --name MI_CONTENEDOR --network MI_RED nginx

Supongamos que deseamos que "contenedor1" haga una solicitud HTTP a "contenedor2". Podemos hacerlo utilizando el nombre del contenedor:

Desde contenedor1, hacemos una solicitud HTTP a contenedor2

docker exec -it contenedor1 curl http://contenedor2

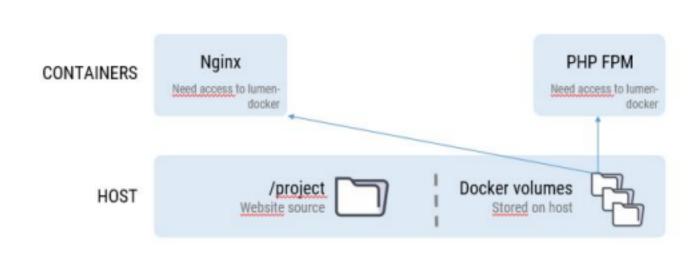
Esto permite que los contenedores se comuniquen entre sí en la misma red personalizada.

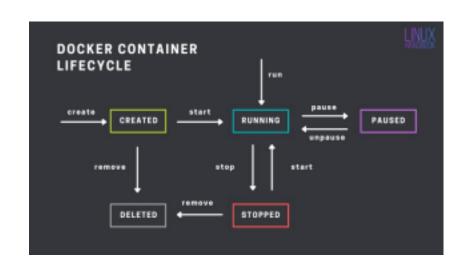
Supongamos que deseamos exponer un servidor web en el puerto 80 de un contenedor al puerto 8080 del host. Para ello, utilizamos el mapeo de puertos al iniciar el contenedor:

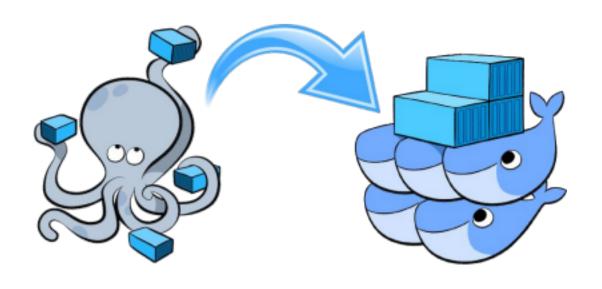
docker run -d -p 8080:80 nginx

Ahora, cualquier solicitud que llegue al puerto 8080 del host será redirigida al puerto 80 del contenedor nginx, permitiendo que las solicitudes HTTP lleguen al contenedor a través del puerto del host









```
docker-compose.yml
```

```
services:
     service1:
       image: image1
       container_name: container1
       ports:
         - 1234:1234
       environment:
         - PUID=1000
         - PGID=100
       volumes:
         - /volume1/docker/service1:/app/config
     service2:
       image: image2
       container_name: container2
       ports:
         - 5678:5678
       environment:
networks:
  default:
    name: network1
```

ipam:

config:

```
sidekiq:
    build:
      context: ./spooky-rails.api
      dockerfile: Dockerfile.dev
    depends_on:
      - webapi
      - redis
    volumes:
      - ./spooky-rails.api:/app
      - gem_cache:/usr/local/bundle/gems
      - node_modules:/app/node_modules
    env_file: ./spooky-rails.api/.env
    environment:
      RAILS_ENV: development
    entrypoint: ["sh", "/app/sidekiq-entrypoint.sh"]
volumes:
  gem_cache:
 node modules.
```

```
You, seconds ago | 1 author (You)
         image: postgres:12.5
           POSTGRES_USER: root
           POSTGRES_PASSWORD: password
9
           - "5432:5432"
           - ./tmp/db:/var/lib/postgresgl/data
         command: bash -c "rm -f tmp/pids/server.pid && bundle exec rails s -p 3000 -b '0.0.0.0"
14
          - .:/app
           - gem_data:/usr/local/bundle
         ports:
19
           - 3000:3000
         depends_on:
           - db
       gen_data:
```

docker-compose.ymi