

Iniciación con docker

Instalamos docker en un servidor con IP 192.168.1.200 que llamaremos host

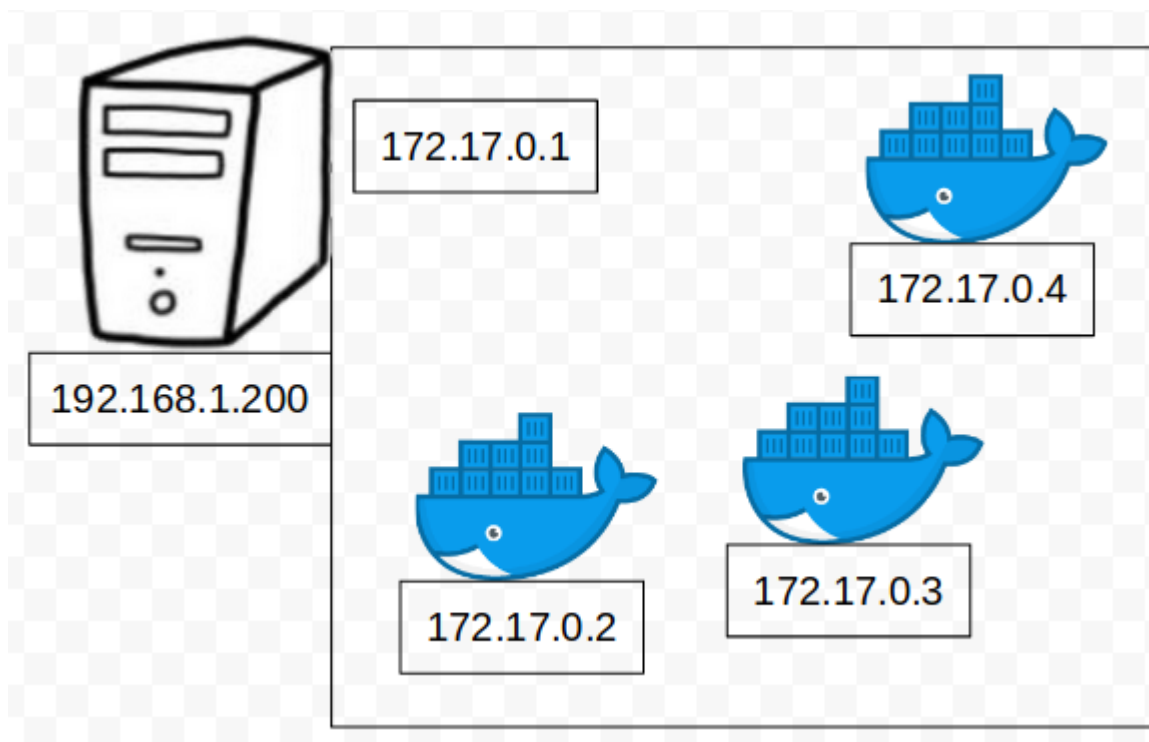
Internamente crea una red 172.17.0.0/16 donde el host tiene ip **172.17.0.1** y los contenedores que vamos creando:

172.17.0.2

172.17.0.3

172.17.0.4

.....



Ejecutamos un docker con nginx. En este caso la imagen de nginx levanta el puerto 80, se tendría que mirar la documentación de cada docker:

```
docker run -ti nginx
```

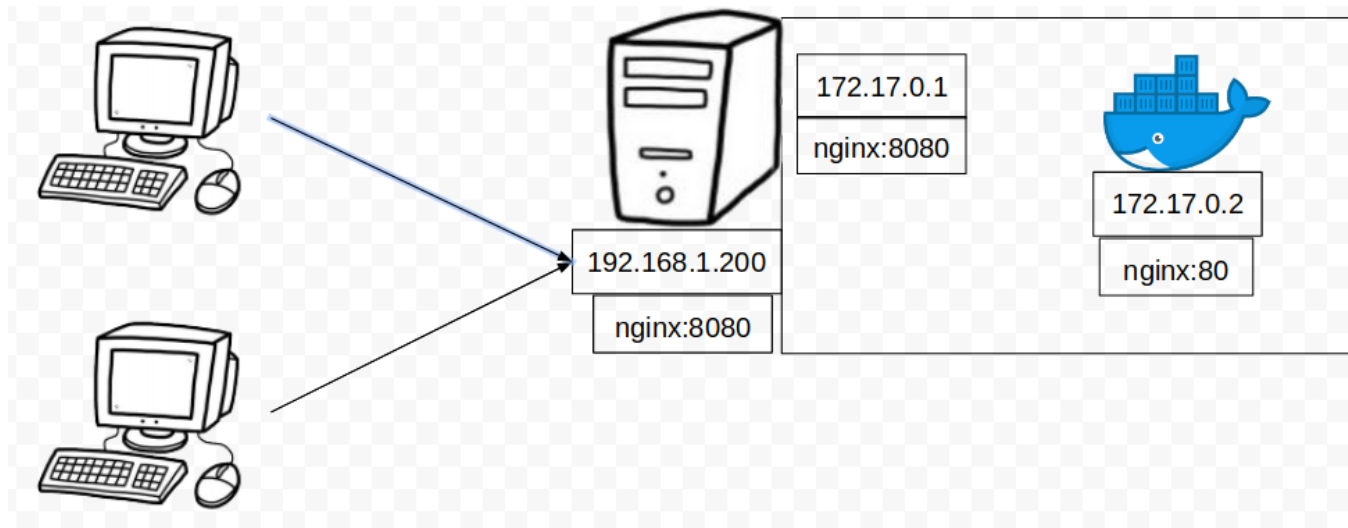
Podemos acceder al docker en la ip 172.0.0.2:80 desde el servidor pero no desde fuera.



Para acceder desde fuera, podemos mapear el puerto al puerto que queramos, por ejemplo al 8080 con el parámetro p:

```
-p <puerto externo>:<puerto interno>
```

```
docker run -ti -p 8080:80 nginx
```

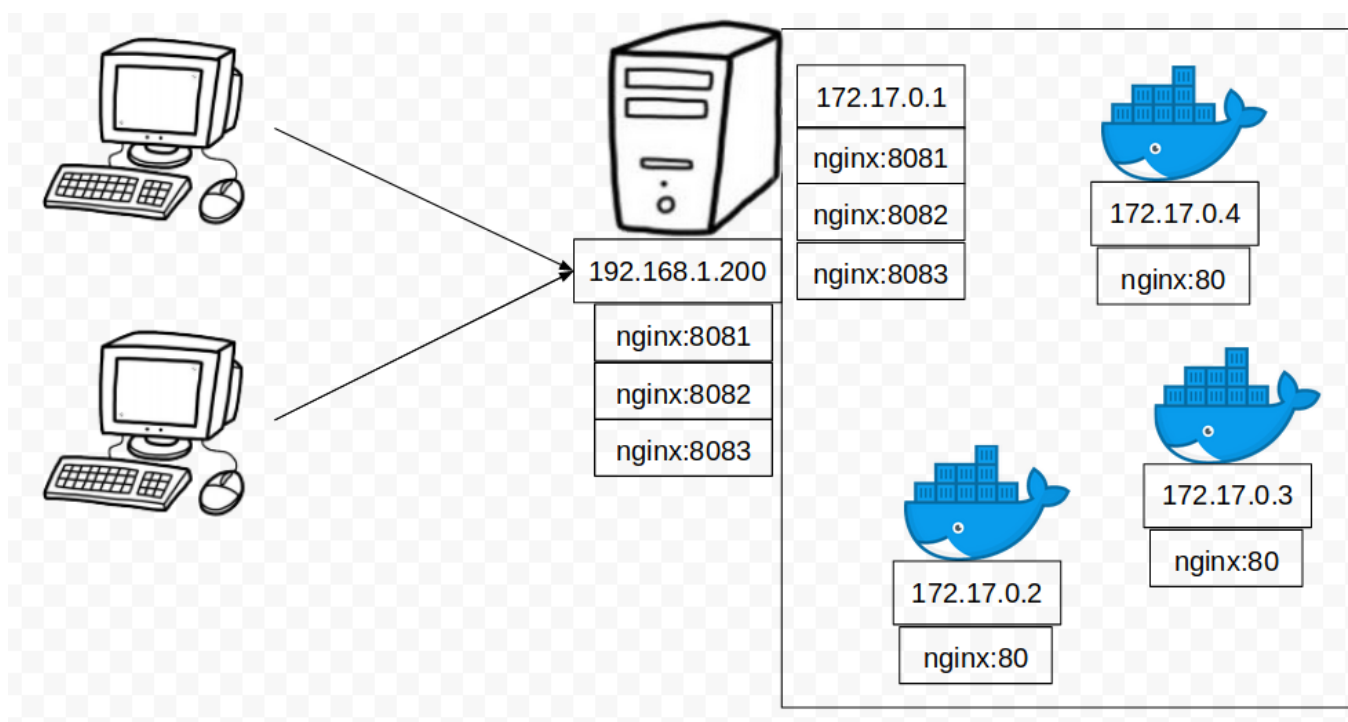


Ahora podemos acceder a nginx en:

```
http://192.168.1.200:8080
```

Podemos levantar mas servicios y mapear a puertos diferentes, nos creará varios contenedores.

```
docker run -ti -p 8081:80 nginx
docker run -ti -p 8082:80 nginx
docker run -ti -p 8083:80 nginx
```



Volúmenes

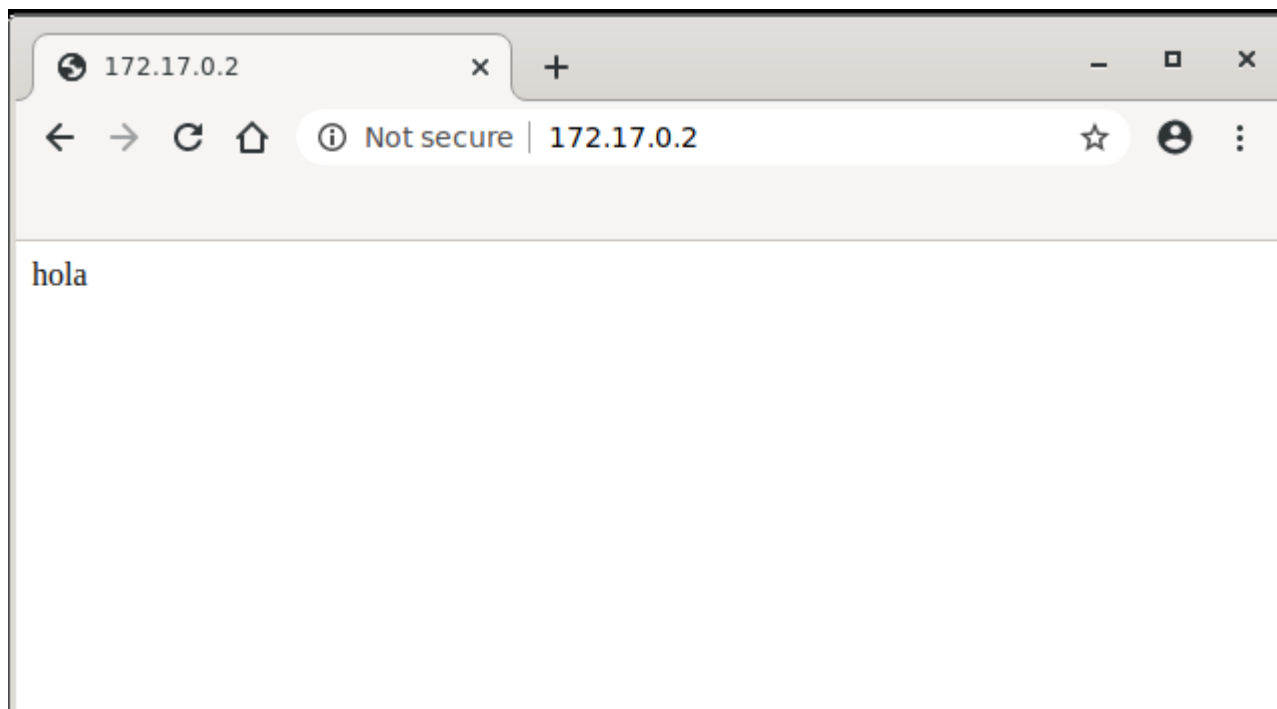
Podemos hacer que la información de los docker sea persistente mapeando volúmenes, para contenido, ficheros de configuración....

Por ejemplo podemos cambiar el contenido de nginx a una carpeta local, Es con el parámetro v:

```
-v <directorio local>:<directorio del docker>
```

Tenemos que conocer la configuración del docker para saber la ruta de los html. Creamos un fichero index.html y lo ponemos en nuestro directorio en local /mi_html/index.html:

```
docker run -v /mi_html:/usr/share/nginx/html nginx
```



Así, guardando el directorio /mi_html podemos levantar el mismo nginx varias veces o en diferentes servidores.

Podemos hacer lo mismo con los ficheros de configuración, en este caso un fichero:

```
docker run -ti /mi_conf/nginx.conf:/etc/nginx/nginx.conf
```

Caso práctico, podemos levantar muchos servicios con un nginx por delante:

Creamos el fichero nginx.conf que haga de proxy inverso:

```
server {  
    listen      80;  
    server_name miweb.com;  
  
    location / {  
        proxy_pass http://192.168.1.200:8081;  
    }  
}  
  
server {  
    listen      80;
```

```
server_name  miwordpress.com;

location / {
    proxy_pass http://192.168.1.200:8082;
}

}

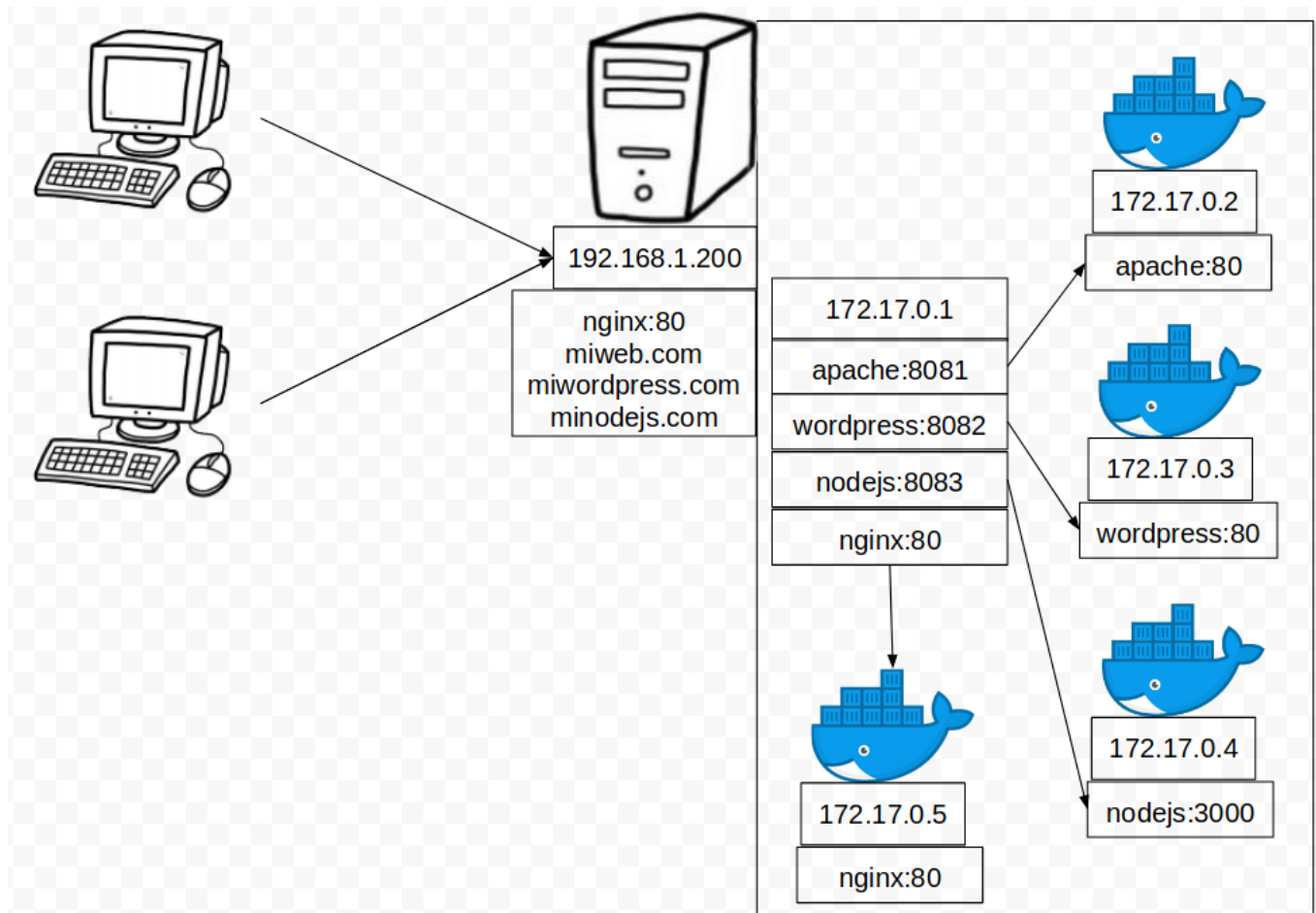
server {
    listen      80;
    server_name minodejs.com;

    location / {
        proxy_pass http://192.168.1.200:8083;
    }
}
```

Levantamos 3 dockers con los servicios y un docker con nginx que estará por delante mapeando todos:

```
docker run -ti -p 8081:80 apache
docker run -ti -p 8082:80 wordpress
docker run -ti -p 8083:3000 nodejs

docker run -ti -p 80:80 nginx
```



Si quiero añadir un nuevo servicio, simplemente tengo que modificar el fichero nginx.conf de mi servidor host y reiniciar el docker de nginx. Por ejemplo un servidor de redmine, levanto el docker con redmine, que por defecto es el puerto 3000, lo cambio al 8084 para tenerlo mas ordenado:

```
docker run -ti -p 8084:3000 redmine
```

Añado la entrada al fichero nginx.conf que lo tengo en local:

```
server {
    listen      80;
    server_name miredmine.com;

    location / {
        proxy_pass http://192.168.1.200:8084;
    }
}
```

Y desde el servidor host reiniciar el docker de nginx:

```
docker restart nginx
```

Mas ejemplos

Hay servicios mas complejos que tenemos que mirar en la documentación que puertos sirve, ficheros de configuración, variables.....

Por ejemplo un controlador unifi:

```
docker run --name=unifi-controller -e PUID=1000 -e PGID=1000 -e MEM_LIMIT=1024M `#optional` -p 3478:3478/udp -p 10001:10001/udp -p 8080:8080 -p 8081:8081 -p 8443:8443 -p 8843:8843 -p 8880:8880 -p 6789:6789 -v ${PWD}/unifi-controller:/config --restart unless-stopped linuxserver/unifi-controller
```

Luego levantamos un servidor de radius mapeando el fichero con los usuarios en formato json:

```
docker run --rm -ti -p 1812:1812/udp -p 1813:1813/udp -v usuariosfreeradius
```

Y tengo en pocos segundos un servidor unifi corriendo con una base de datos freeradius

From:
<http://wiki.legido.com/> - **Legido Wiki**

Permanent link:
<http://wiki.legido.com/doku.php?id=informatica:linux:docker:ejemplos>

Last update: **2020/04/23 18:18**



