

# Workshop 07 - Deploy a Web App Using Docker

---

## Creating a Docker Workspace

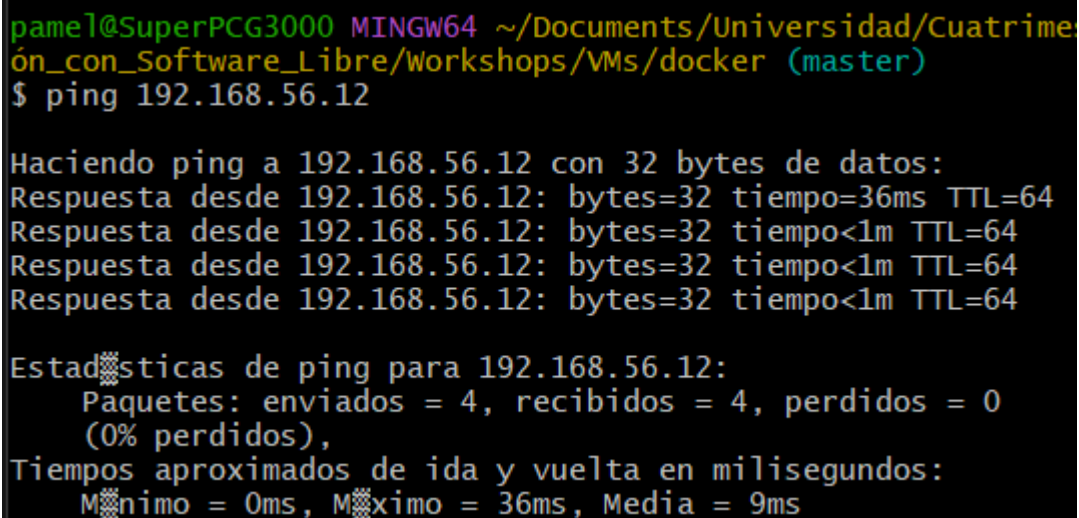
To generate a new virtual Docker we position ourselves into our VMs folder and write the following commands.

```
mkdir docker
cd docker
vagrant init debian/bookworm64
code Vagrantfile
```

Inside our file, we're gonna go to line 35 and change our virtual routers IP "192.168.56.12", making it look like this `config.vm.network "private_network", ip: "192.168.56.12"`. We also have to de-comment lines 59, 64, 65 and for a little more space we're gonna assign a different value to the memory on line 64:

```
vb.memory = "2048"
```

This way we can try and ping it to see if it works:



```
pamel@SuperPCG3000 MINGW64 ~/Documents/Universidad/Cuatrimestre/Trabajo/Trabajo_07/Workshops/VMs/docker (master)
$ ping 192.168.56.12

Haciendo ping a 192.168.56.12 con 32 bytes de datos:
Respuesta desde 192.168.56.12: bytes=32 tiempo=36ms TTL=64
Respuesta desde 192.168.56.12: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.56.12: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.56.12: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.56.12:
  Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
  Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 36ms, Media = 9ms
```

## Connecting to Virtual Machine

Now, to alter the machine is the same process as every other, a simple `vagrant ssh` will work. Additionally, we should rename the VM.

```
vagrant ssh
sudo hostnamectl set-hostname docker
sudo nano /etc/hosts
## After changes
exit
vagrant ssh
```

```

vagrant@bookworm: ~
GNU nano 7.2 /etc/hosts *
127.0.0.1 localhost
127.0.0.2 docker
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

pamel@SuperPCG3000 MINGW64 ~/Documents/Univer
on_con_Software_Libre/Workshops/VMs/docker (r
$ vagrant ssh
Linux docker 6.1.0-18-amd64 #1 SMP PREEMPT_D
x86_64

The programs included with the Debian GNU/Li
the exact distribution terms for each progr
individual files in /usr/share/doc/*/copyrig

Debian GNU/Linux comes with ABSOLUTELY NO WA
permitted by applicable law.
Last login: Thu Aug 1 21:20:07 2024 from 10
vagrant@docker:~$ hostname
docker
vagrant@docker:~$ |

```

## Installing Docker

Docker has a very usefull guide on how to install it on Debian, it contains the following commands.

```

sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin

```

And with that we should be ready to start creating containers.

## Authorize Users

A good practice in programming is to keep users authorized for certain tasks, in this instance, creating a container should not be an admin only job.

To authorize another user to create a container, we use `sudo gpasswd -a $USER docker`. And then we only have to restart the session.

```
vagrant@docker:~$ sudo gpasswd -a $USER docker
Adding user vagrant to group docker
vagrant@docker:~$ exit
logout

pamel@SuperPCG3000 MINGW64 ~/Documents/Universidad/CuatrimestreII2024/Programación_con_Software_Libre/Workshops/VMs/docker (master)
$ vagrant ssh
Linux docker 6.1.0-18-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.76-1 (2024-07-16) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Aug  1 21:22:42 2024 from 10.0.2.2
vagrant@docker:~$ docker container ls
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS         PORTS          NAMES
vagrant@docker:~$
```

We can even do a little hello world with docker `docker run hello-world`

```
vagrant@docker: ~
permitted by applicable law.
Last login: Thu Aug  1 21:22:42 2024 from 10.0.2.2
vagrant@docker:~$ docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
vagrant@docker:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:1408fec50309afee38f3535383f5b09419e6dc0925bc69891e79d84cc4cdcec6
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.
```

## Conditioning our Workspace

In our host machine we'll position ourselves in the docker folder and run the following commands:

```
mkdir DockerWS
cd DockerWS
mkdir sites
mkdir proxy
cd ../
code Vagrantfile
```

And in the Vagrantfile we will add a line on the blank space in the 47 and write `config.vm.synced_folder` `"/home/vagrant/sites", owner: "www-data", group: "www-data"`

To apply the changes, we halt and the up the machine.

## Creating a Site inside Docker

We might as well use a site we already have, so we're gonna copy and paste lospatitos to here. To do that, we position ourselves in `../docker/DockerWS/sites` of our host machine and write:

```
cp -r ../../../../webserver/sites/lospatitos.isw811.xyz/ .
mkdir public
mv * public
touch docker-compose.yml
code docker-compose.yml
```

Inside this file we're gonna declare all information necessary to start the container:

```
services:

  web_lospatitos:
    image: php:7.4-apache
    container_name: web_lospatitos
    hostname: web_lospatitos
    restart: always
    environment:
      DB_HOST: db_lospatitos
      DB_USER: $MYSQL_USER
      DB_PASSWORD: $MYSQL_PASSWORD
      DB_NAME: $MYSQL_DATABASE
    volumes:
      - ./public:/var/www/html
    networks:
      - net_isw811

  db_lospatitos:
    image: mysql:8.0
    container_name: db_lospatitos
    hostname: db_lospatitos
    restart: always
    env_file: .env
    volumes:
      - vol_lospatitos:/var/lib/mysql
    command: '--default-authentication-plugin=mysql_native_password'
    networks:
      - net_isw811

volumes:
  vol_lospatitos:

networks:
  net_isw811:
    external: true
```

## Creating an .env file

Pretty self-explanatory, we just create a .env file in the same folder as our docker-composer.yml and declare the constants.

```
MYSQL_USER=isw811
MYSQL_PASSWORD=secret
MYSQL_ROOT_PASSWORD=secret
MYSQL_DATABASE=lospatitos
```

## Create the Network

Connected to our virtual machine we write

```
docker network create net_isw811`
docker compose up
```

```
pamel@SuperPCG3000 MINGW64 ~/Documents/Universidad/CuatrimestreII2024/Programaci
on_con_Software_Libre/Workshops/VMs/docker/DockerWS/sites/lospatitos.isw811.xyz
(master)
$ vagrant ssh
Linux docker 6.1.0-18-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.76-1 (2024-02-01)
x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Aug  1 22:03:29 2024 from 10.0.2.2
vagrant@docker:~$ cd /vagrant/DockerWS/sites/lospatitos.isw811.xyz/
vagrant@docker:/vagrant/DockerWS/sites/lospatitos.isw811.xyz$ docker network cre
ate net_isw811
b31ecd1fffb520d0b35c124963296274f1cab49f56d077a2cb1e2775c6f63d9cd
vagrant@docker:/vagrant/DockerWS/sites/lospatitos.isw811.xyz$ docker network ls

```

NETWORK ID	NAME	DRIVER	SCOPE
365807f5b69d	bridge	bridge	local
d64348be985a	host	host	local
b31ecd1fffb52	net_isw811	bridge	local
9a3270863c81	none	null	local

```
o_lospatitos | 2024-08-01T22:45:01.942291Z 0 [Warning] [MY-010910] [Server] '
default_authentication_plugin' is deprecated and will be removed in a future re
lease. Please use authentication_policy instead.
o_lospatitos | 2024-08-01T22:45:01.942291Z 0 [System] [MY-010116] [Server] /
r/sbin/mysqld (mysqld 8.0.39) starting as process 1
o_lospatitos | 2024-08-01T22:45:01.950632Z 1 [System] [MY-013576] [InnoDB] I
noDB initialization has started.
o_lospatitos | 2024-08-01T22:45:02.399161Z 1 [System] [MY-013577] [InnoDB] I
noDB initialization has ended.
o_lospatitos | 2024-08-01T22:45:02.628034Z 0 [Warning] [MY-010068] [Server] c
ertificate ca.pem is self signed.
o_lospatitos | 2024-08-01T22:45:02.628375Z 0 [System] [MY-013602] [Server] C
hannel mysql_main configured to support TLS. Encrypted connections are now supp
orted for this channel.
o_lospatitos | 2024-08-01T22:45:02.632888Z 0 [Warning] [MY-011810] [Server] I
nsecure configuration for --pid-file: Location '/var/run/mysqld' in the path is
accessible to all OS users. Consider choosing a different directory.
o_lospatitos | 2024-08-01T22:45:02.653342Z 0 [System] [MY-011323] [Server] X
plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/
mysqld/mysqlx.sock
o_lospatitos | 2024-08-01T22:45:02.653888Z 0 [System] [MY-010931] [Server] /
r/sbin/mysqld: ready for connections. Version: '8.0.39' socket: '/var/run/mys
qld/mysqld.sock' port: 3306 MySQL Community Server - GPL.
```

## Connect to an Apache Container

Now that we've done all that we can go into the docker virtual machine and open a command line in our apache server using `docker exec -it web_lospatitos bash`

```

vagrant@docker:~$ docker exec -it web_lospatitos bash
root@web_lospatito
vagrant@docker:~$ docker container ls
CONTAINER ID   IMAGE                                COMMAND                                CREATED        STA
TS            NAMES
736b9c8fe022   php:7.4-apache                     "docker-php-entrypoi..."  34 seconds ago Up
tcp          web_lospatitos
24c4c4f62313   mysql:8.0                          "docker-entrypoint.s..."  10 minutes ago Up
6/tcp, 33060/tcp  db_lospatitos
vagrant@docker:~$ docker exec -it web_lospatitos bash
root@web_lospatitos:/var/www/html# set | grep DB
DB_HOST=db_lospatitos
DB_NAME=lospatitos
DB_PASSWORD=secret
DB_USER=isw811
root@web_lospatitos:/var/www/html#

```

We can also access the database, changing 'web\_lospatitos' to 'db\_lospatitos'

```

root@web_lospatitos:/var/www/html# exit
exit
vagrant@docker:~$ docker exec -it db_lospatitos bash
bash-5.1# set | grep MYSQL
MYSQL_DATABASE=lospatitos
MYSQL_MAJOR=8.0
MYSQL_PASSWORD=secret
MYSQL_ROOT_PASSWORD=secret
MYSQL_SHELL_VERSION=8.0.38-1.el9
MYSQL_USER=isw811
MYSQL_VERSION=8.0.39-1.el9
bash-5.1# |

```

And create a database...

```
mysql -u root --password=$MYSQL_ROOT_PASSWORD
```



```
MYSQL_USER=ism811
MYSQL_VERSION=8.0.39-1.el9
bash-5.1# mysql -u root --password=$MYSQL_ROOT_PASSWORD
mysql: [Warning] Using a password on the command line interface can be insecure
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.39 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

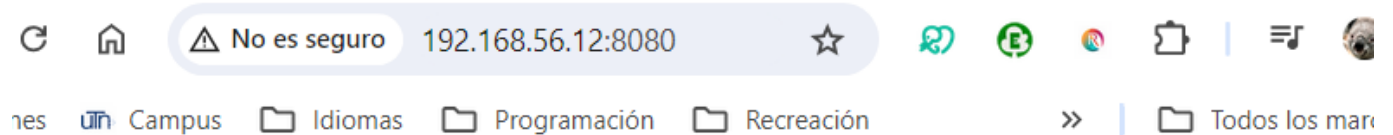
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

## Access the page

Every app has its own port, in this case, our web can be temporarily placed on the port 8080 so that's what we're gonna look for in the search.





# Los Patitos

Here is duck



By Proxy

Let's get in the proxy folder and run the following commands:

```
mkdir nginx_configs
touch nginx_configs/default.conf
touch nginx_configs/lospatitos.isw811.xyz.conf
code .
```

In the lospatitos.isw811.xyz.conf file we'll write:

```
upstream web_lospatitos {
    server web_lospatitos:80;
}

server {
    listen 443 ssl;
    server_name lospatitos.isw811.xyz;

    ssl_certificate /etc/ssl/certs/nginx/lospatitos.isw811.xyz/cert.pem;
    ssl_certificate_key /etc/ssl/certs/nginx/lospatitos.isw811.xyz/privkey.pem;

    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-Proto $scheme;

    location / {
        proxy_pass http://web_lospatitos;
    }
}

server {
    listen 80;
    server_name lospatitos.isw811.xyz;
    return 301 https://$server_name$request_uri;
}
```

And in the default.conf

```
server {

    listen 80;

    location / {
        root    /usr/share/nginx/html;

        index   index.html index.htm;
    }

}
```

In that same level of nginx\_configs folder we'll create another two folders 'ssl' and 'default\_site', as well as another docker-compose.yml.

### Docker-compose.yml

```
services:
  proxy:
    image: nginx:1-alpine
    container_name: web_proxy
```

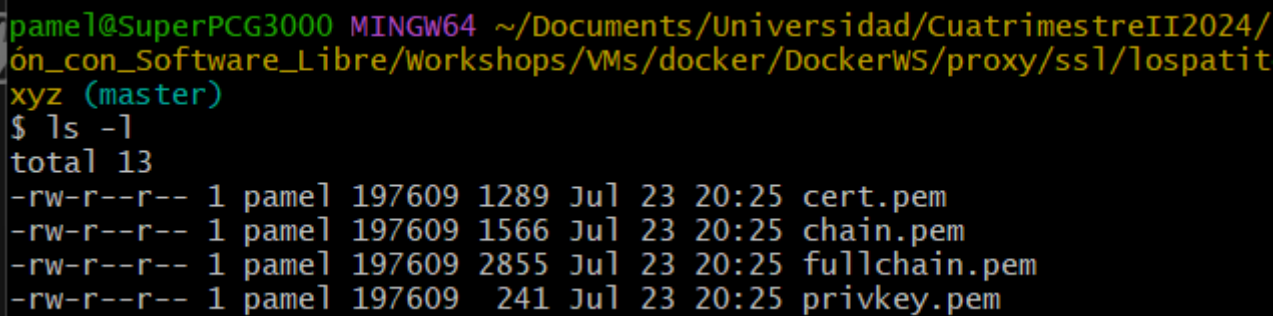
```
hostname: web_proxy
volumes:
  - ./nginx_configs:/etc/nginx/conf.d:ro
  - ./ssl:/etc/ssl/certs/nginx:ro
  - ./default_site:/usr/share/nginx/html:ro
ports:
  - "443:443"
  - "80:80"
networks:
  - net_isw811

networks:
  net_isw811:
    external: true
```

## SSL

```
mkdir
mkdir -p ssl/lospatitos.isw811.xyz
```

And inside of it we should have all our ssl files and keys.

A terminal window screenshot showing a directory listing of SSL files. The prompt is 'pamel@SuperPCG3000 MINGW64 ~/Documents/Universidad/CuatrimestreII2024/ón\_con\_Software\_Libre/Workshops/VMs/docker/DockerWS/proxy/ssl/lospatit xyz (master)'. The command '\$ ls -l' is executed, showing a total of 13. The files listed are: cert.pem, chain.pem, fullchain.pem, and privkey.pem, all with permissions -rw-r--r-- and owned by pamel.

```
pamel@SuperPCG3000 MINGW64 ~/Documents/Universidad/CuatrimestreII2024/
ón_con_Software_Libre/Workshops/VMs/docker/DockerWS/proxy/ssl/lospatit
xyz (master)
$ ls -l
total 13
-rw-r--r-- 1 pamel 197609 1289 Jul 23 20:25 cert.pem
-rw-r--r-- 1 pamel 197609 1566 Jul 23 20:25 chain.pem
-rw-r--r-- 1 pamel 197609 2855 Jul 23 20:25 fullchain.pem
-rw-r--r-- 1 pamel 197609 241 Jul 23 20:25 privkey.pem
```

## Default\_site

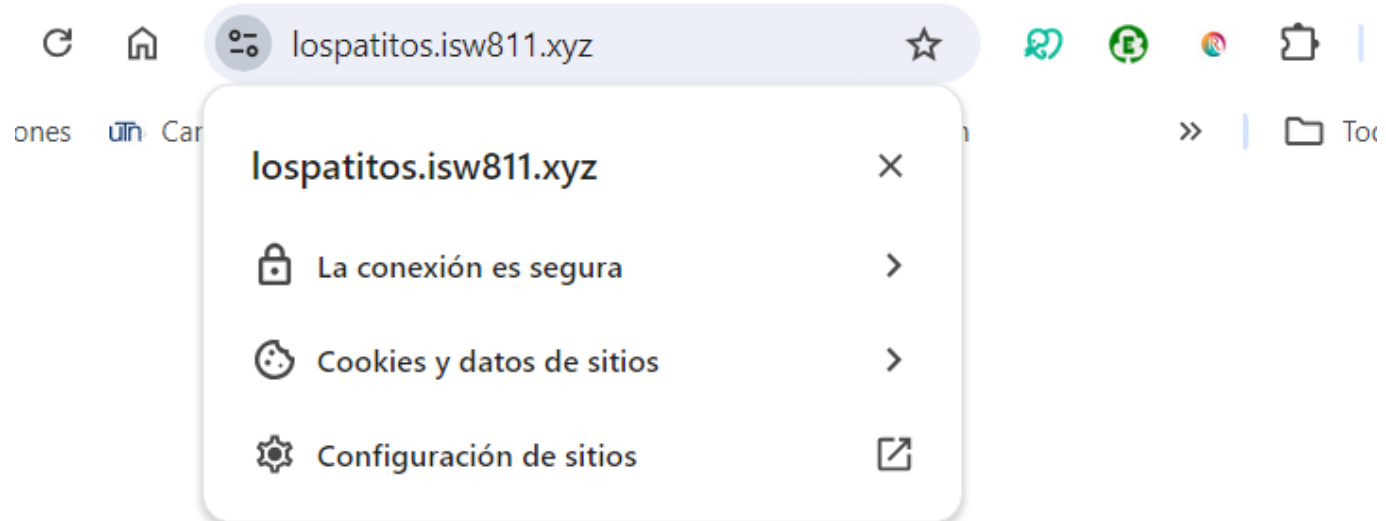
This will only have an inde.html file with nothing much on it.

## Including the Page

To see the page by its domain we add the IP into our hosts file

Archivo	Editar	Ver
<pre># The IP address and the host name should be separated by at least one # space. # # Additionally, comments (such as these) may be inserted on individual # lines or following the machine name denoted by a '#' symbol. # # For example: # #      102.54.94.97      rhino.acme.com      # source server #      38.25.63.10      x.acme.com         # x client host  # localhost name resolution is handled within DNS itself. 192.168.56.10 pamelamurillo.isw811.xyz 192.168.56.10 lfts.isw811.xyz 192.168.56.12 <u>lospatitos.isw811.xyz</u> 192.168.56.10 socialhubmanager.pma.com  #      127.0.0.1      localhost #      ::1           localhost</pre>		
Ln 14, Col 15	948 caracteres.	100% Windows (CRLF,

And see it in the web



Created by Pamela Murillo