



LABORATORIO PWEB 2

GRUPO - G

LAB 01: “Docker”

ELABORADO POR:

- Huamani Condori Jeanpiero Sixto

DOCENTE:

Pinto Oppe Lino Jose

**AREQUIPA – PERÚ
2024**

Especificaciones del Lab 01

- Instale el servidor web Apache HTTP server 2.x
- Instale cualquiera de estos lenguajes de programación: PHP, Perl, Python.
- Configure el servidor web para que interprete uno de los lenguajes de programación.
- Instale cualquiera de los servidores de base de datos: MySQL, MariaDB, PostgreSQL.
- Instale el servidor Open SSH Server. Envíe archivos al servidor: imágenes, css, js, etc.
- Cree un usuario pw2 con contraseña: 12345678.
- Otorgue permisos al usuario para acceder a la aplicación web. (Read/Write)
- Finalmente implemente el trabajo final del curso de pw1 en ese contenedor.
- Elabore un informe paso a paso para donde explique funcionalmente el proyecto demostrando que se trata de un contenedor docker.
- Adjunte la URL de un video donde muestre que se trata de un contenedor Docker.

Entregables:

- Informe de laboratorio
- URL: Video Youtube.
- INVESTIGACIÓN: Suba el proyecto a Docker Hub (Público).
- URL: Docker Hub

Resolucion del laboratorio

Instalación de requerimientos y configuración del servidor

En esta sección trataremos los siguientes puntos:

- Instale el servidor web Apache HTTP server 2.x
- Instale cualquiera de estos lenguajes de programación: PHP, Perl, Python.
- Instale cualquiera de los servidores de base de datos: MySQL, MariaDB, PostgreSQL.
- Cree un usuario pw2 con contraseña: 12345678.

Para instalar todos los requerimientos del servidor se preparó un archivo dockerfile en donde se pondrá el código para instalar y descargar los programas necesarios a la imagen base UBUNTU.

-Archivo dockerfile

```

FROM ubuntu:latest

# Actualiza los repositorios e instala los paquetes necesarios
RUN apt-get update && apt-get install -y \
    libapache2-mod-perl2 \
    cpanminus \
    apache2 \
    vim \
    openssh-server \
    mysql-server \
    gcc \
    libdbd-mysql-perl \
    perl \
    libapache2-mod-fcgid

# Inicia los servicios necesarios
RUN /etc/init.d/apache2 start && \
    /etc/init.d/ssh start && \
    /etc/init.d/mysql start

# Habilita los módulos necesarios de Apache
RUN cpanm CGI

# Crea un usuario y establece una contraseña
RUN useradd -m pw2 && echo "pw2:12345678" | chpasswd
RUN a2enmod cgid
# Reinicia Apache
RUN /etc/init.d/apache2 restart

# Copia el script SQL al contenedor
COPY createData.sql /tmp/createData.sql

# Ejecuta el script SQL para inicializar la base de datos
RUN service mysql start && \
    mysql -u root -e "CREATE USER 'pw2'@'localhost' IDENTIFIED BY '12345678';" && \
    mysql -u root -e "GRANT ALL PRIVILEGES ON mydatabase.* TO 'pw2'@'localhost';" && \
    mysql -u root -e "FLUSH PRIVILEGES;" && \
    mysql -u root -e "CREATE DATABASE IF NOT EXISTS mydatabase;" && \
    mysql -u root mydatabase < /tmp/createData.sql && \
    service mysql stop

# Cambia los permisos del directorio HTML
RUN chown -R pw2:www-data /var/www/html/

# Exponer el puerto 80 para Apache
EXPOSE 80

CMD ["bash"]

```

-creación de la imagen a partir del dockerfile

```

PS C:\Users\User\Downloads\inventory-docker-main> docker build -t lab01_imagen -f ubuntu.dockerfile .
[+] Building 0.0s (0/0) docker:default
[+] Building 58.4s (15/15) FINISHED
=> [internal] load build definition from ubuntu.dockerfile
=> => transferring dockerfile: 1.43kB
=> [internal] load metadata for docker.io/library/ubuntu:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [ 1/10] FROM docker.io/library/ubuntu:latest@sha256:3e5be01163158540853063947aeb6166507f4841b5b9d7c09804621674e9c7c1
=> [internal] load build context
=> => transferring context: 199B
=> CACHED [ 2/10] RUN apt-get update && apt-get install -y libapache2-mod-perl2 cpanminus apache2 vim openssh-
=> CACHED [ 3/10] RUN /etc/init.d/apache2 start && /etc/init.d/ssh start && /etc/init.d/mysql start
=> CACHED [ 4/10] RUN cpanm CGI
=> CACHED [ 5/10] RUN useradd -m pw2 && echo "pw2:12345678" | chpasswd
=> [ 6/10] RUN a2enmod cgi
=> [ 7/10] RUN /etc/init.d/apache2 restart
=> [ 8/10] COPY createData.sql /tmp/createData.sql
=> [ 9/10] RUN service mysql start && mysql -u root -e "CREATE USER 'pw2'@'localhost' IDENTIFIED BY '12345678';" && mysql
=> [10/10] RUN chown -R pw2:www-data /var/www/html/
=> exporting to image
=> => exporting layers
=> => writing image sha256:8c3c3db28fe069799726e4830aec1106e1ebefa2d58edc0a1cc04b8fb355e7f56
=> => naming to docker.io/library/lab01_imagen
What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview

```

-creación del contenedor

```

PS C:\Users\User\Downloads\inventory-docker-main> docker run --name pw2_lab01 -p 8084:80 -p 8085:3306 -p 8086:22 -it lab01_imagen /bin/bash
root@72030844b3fd:/# exit
exit
PS C:\Users\User\Downloads\inventory-docker-main> docker ps
CONTAINER ID   IMAGE          COMMAND                  STATUS    PORTS          NAMES
72030844b3fd   lab01_imagen  "/bin/bash"             Exited (0) 4 seconds ago

```

Enviar el proyecto pweb 1 al servidor creado con docker

En esta sección trataremos los siguientes puntos:

- Instale el servidor Open SSH Server. Envíe archivos al servidor: imágenes, css, js, etc.
- Finalmente implemente el trabajo final del curso de pw1 en ese contenedor.

Para enviar las carpetas y archivos del proyecto usaremos sftp

-Moviendo archivos estáticos a /var/www/html

```

PS C:\WINDOWS\system32> sftp -P 8086 pw2@127.0.0.1
pw2@127.0.0.1's password:
Connected to 127.0.0.1.
sftp> ls
sftp> ls -la
sftp> cd /var/www/html
sftp> ls -la
sftp> put -r C:\Users\User\Downloads\inventory-docker-main\inventario\html
Uploading C:/Users/User/Downloads/inventory-docker-main/inventario/html/ to /var/www/html/
Entering C:/Users/User/Downloads/inventory-docker-main/inventario/html/
C:/Users/User/Downloads/inventory-docker-main/inventario/html/estilos
C:/Users/User/Downloads/inventory-docker-main/inventario/html/estilos/style.css
C:/Users/User/Downloads/inventory-docker-main/inventario/html/formulario.html
Entering C:/Users/User/Downloads/inventory-docker-main/inventario/html/img
Entering C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/alpaca
C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/alpaca/chalina.jpg
C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/alpaca/chompa.jpg
C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/alpaca/img1.jpg
C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/alpaca/medias.jpg
C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/alpaca/Nuevo Archivo WinRAR ZIP.zip
Entering C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/index
C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/index/banner.jpg
C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/index/banner2.png
C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/index/banner3.jpg
C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/index/banner4.webp
C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/index/banner5.png
C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/index/banner6.png
C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/index/banner7.png
C:/Users/User/Downloads/inventory-docker-main/inventario/html/img/index/fondo.gif

```

-Moviendo archivos perl a /usr/lib/cgi-bin

```
sftp> cd /usr/lib/cgi-bin/
sftp> put -r C:\Users\User\Downloads\inventory-docker-main\inventario\cgi-bin/
Uploading C:\Users\User\Downloads\inventory-docker-main\inventario\cgi-bin/ to /usr/lib/cgi-bin/
Entering C:\Users\User\Downloads\inventory-docker-main\inventario\cgi-bin/
C:\Users\User\Downloads\inventory-docker-main\inventario\cgi-bin\agregar.pl          100% 2945   260.6KB/s   00:00
C:\Users\User\Downloads\inventory-docker-main\inventario\cgi-bin\prueba.pl         100%  874   125.8KB/s   00:00
C:\Users\User\Downloads\inventory-docker-main\inventario\cgi-bin\retirar.pl        100% 2097   262.5KB/s   00:00
C:\Users\User\Downloads\inventory-docker-main\inventario\cgi-bin\verLista.pl       100%  838    82.2KB/s   00:00
sftp> root -e "FLUSH PRIVILEGES;" && \
```

Dar permisos para ejecutar la web

En esta sección trataremos los siguientes puntos:

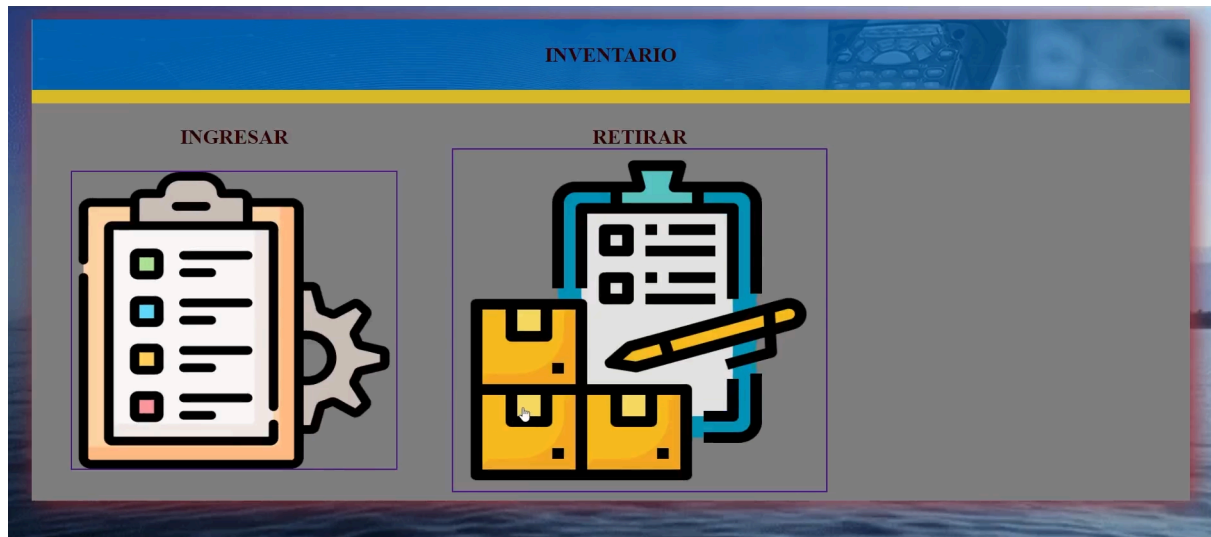
- Otorgue permisos al usuario para acceder a la aplicación web. (Read/Write)
- Configure el servidor web para que interprete uno de los lenguajes de programación.

A las carpetas le daremos permiso 755 , para los archivos estáticos 644, y 755 para los archivos perl.

-Dando permisos

```
root@72030844b3fd:/var/www/html# ls -la
total 40
drwx----- 1 pw2 www-data 4096 May 10 18:55 .
drwxr-xr-x 1 root root    4096 May 10 00:36 ..
drwx----- 2 pw2 www-data 4096 May 10 18:55 estilos
-rw-rw-r-- 1 pw2 www-data 1328 May 10 18:55 formulario.html
drwx----- 7 pw2 www-data 4096 May 10 18:55 img
-rw-r--r-- 1 pw2 www-data  912 May 10 18:55 index.html
drwx----- 2 pw2 www-data 4096 May 10 18:55 javascript
-rw-rw-r-- 1 pw2 www-data 1433 May 10 18:55 retirar.html
root@72030844b3fd:/var/www/html# chmod 755 estilos
root@72030844b3fd:/var/www/html# chmod 755 img
root@72030844b3fd:/var/www/html# chmod 755 javascript
root@72030844b3fd:/var/www/html# chmod 644 formulario.html
root@72030844b3fd:/var/www/html# chmod 644 index.html
root@72030844b3fd:/var/www/html# chmod 644 retirar.html
root@72030844b3fd:/var/www/html# chmod 644 retirar.html
```

-Habilitando la ejecución de cgi-bin perl



Url del video

En esta sección trataremos los siguientes puntos:

- Adjunte la URL de un video donde muestre que se trata de un contenedor Docker.

<https://youtu.be/8AbtHp7N-V8>

Url del docker hub

En esta sección trataremos los siguientes puntos:

- URL: Docker Hub

https://hub.docker.com/r/jeanphc/jhuamani_pw2_lab01