Taller: Despliegue de Aplicaciones con Docker (2/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

Parte 2: Ejemplos usando el comando Docker

En esta segunda guía aprenderá a instalar docker y a ejecutar el comando docker para usar imágenes existentes, crear, detener, arrancar e inspeccionar contenedores.

Instalar y Configurar: Descargue el repositorio del taller desde Github. El repositorio tiene todos los documentos y comandos requeridos, para facilitar y agilizar el taller. Para descargarlo use el siguiente comando:

```
git clone https://github.com/ingsistemascloud/seiiis2019.git
```

Espere a que descargue y ubíquese en el directorio seiiis2019. Use el siguiente comando:

```
cd seiiis2019/
```

Encontrará un SCRIPT llamado install-docker.sh, que permitirá instalar docker y docker-compose, las dos herramientas a usar en el taller. Ejecute ese SCRIPT con el siguiente comando:

```
./install-docker.sh
```

Espere a que instale y ubíquese en el directorio ejemplosDockerCommand. Use el siguiente comando:

```
cd ejemplosDockerCommand/
```

En ese directorio encontrará seis (6) ejemplos. Por cada ejemplo hay un fichero de texto plano (TXT) que contiene los comandos. Puede copiarlos y pegarlos, de manera que no se presenten errores mientras prueba y aprende. Si usa putty, para copiar simplemente seleccione el texto y para pegar haga click derecho. Para ver el contenido de los ficheros pruebe a usar el comando more y el nombre del fichero, por ejemplo:

```
more ejemplo0.txt
```

Taller: Despliegue de Aplicaciones con Docker (2/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

0. Docker Hola Mundo. Para crear un docker se usa el comando "sudo docker run IMAGEN", donde IMAGEN corresponde al nombre de una imagen existente. Si la imagen no existe en el servidor, se descarga desde dockerhub. Para realizar la prueba típica de Hola Mundo, ejecute los siguientes comandos:

```
sudo docker images
```

Si todo funciona correctamente, no aparecerá ninguna imagen, pues acaba de instalar docker y es la primera vez que lo usa. Ahora ejecute el siguiente comando:

```
sudo docker run hello-world
```

Si todo funciona correctamente, aparecerá una pantalla como la siguiente:

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:c3b4ada4687bbaa170745b3e4dd8ac3f194ca95b2d0518b417fb47e5879d9b5f
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.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Lo que sucedió es que se descargó la imagen del docker hello-world y se ejecutó en el servidor. Pruebe de nuevo el primer comando y ahora aparecerá la imagen que se descargó.

```
sudo docker images
```

```
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hello-world         latest              fce289e99eb9        9 months ago        1.84kB
```

Taller: Despliegue de Aplicaciones con Docker (2/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

1. Docker jenkins. Ahora se va a crear un docker usando una imagen de la herramienta jenkins, usada para automatizar la integración contínua (devops). Use el siguiente comando (como el comando no cabe en una sóla línea, las lineas se separan usando el carácter backslash \):

```
sudo docker run -d --name myjenkins \
-p 8080:8080 -p 50000:50000 jenkins/jenkins:lts
```

Espere a que finalice y aparecerá una imagen como la siguiente:

```
Unable to find image 'jenkins/jenkins:lts' locally
lts: Pulling from jenkins/jenkins
092586df9206: Pull complete
ef599477fae0: Pull complete
4530c6472b5d: Pull complete
d34d61487075: Pull complete
272f46008219: Pull complete
12ff6ccfe7a6: Pull complete
f26b99e1adb1: Pull complete
b5a7230f28ac: Pull complete
d19ccd039899: Pull complete
0cd916c5652e: Pull complete
e7f33481e423: Pull complete
3c8f6834c187: Pull complete
aed304377df9: Pull complete
841ddf9b4483: Pull complete
84beffd862a8: Pull complete
2643324effeb: Pull complete
2b8c3739eb59: Pull complete
42423eff2b9e: Pull complete
ce3c0ae47fcb: Pull complete
Digest: sha256:bf741663dff2c5c163bd16706d4d1b9e6f86fa7b92e365257e170239d0bfb24f
Status: Downloaded newer image for jenkins/jenkins:lts
4c18a4b59d18be7240929a59aa4d576e4808eaf5faf5a23e6cc91b26df46fab6
```

Ahora pruebe el comandos docker container ps, que lista los contenedores en ejecución. Aparecerá su docker jenkins ejecutándose y recibiendo peticiones HTTP en el puerto 8080. Observe que el docker se llama myjenkins, según el comando usado para crearlo.

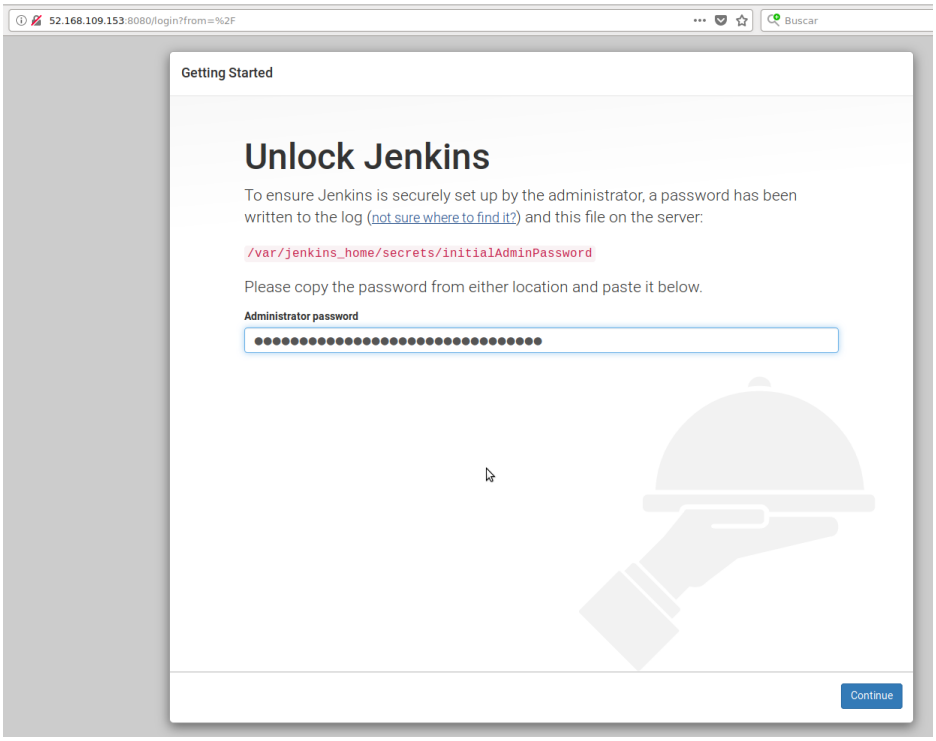
```
sudo docker container ps
```

```
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS              PORT
S                                          NAMES
4c18a4b59d18        jenkins/jenkins:lts    "/sbin/tini -- /usr/…"   29 seconds ago      Up 26 seconds       0.0.
0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp   myjenkins
```

Taller: Despliegue de Aplicaciones con Docker (2/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

Abra el navegador e ingrese a la URL http://DIRECCION_IP_PUBLICA:8080/ donde la DIRECCION_IP_PUBLICA corresponde a la dirección de su máquina virtual, la cual debe revisar en la consola de AWS. Aparecerá una ventana como la siguiente, pidiéndole una clave e informándole la ruta donde se encuentra esa clave.



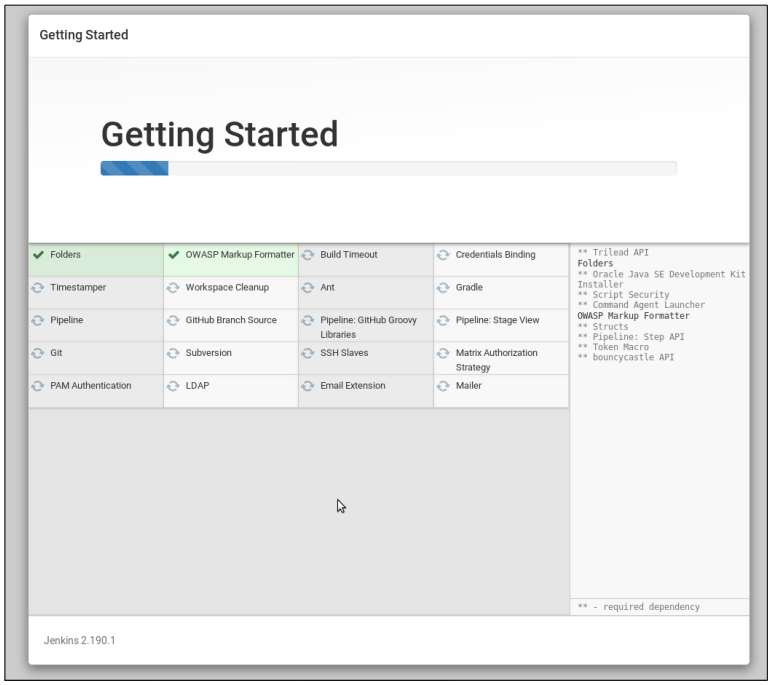
Ejecute el siguiente comando para acceder a la clave, copie la clave y péguela en la ventana del navegador.

```
sudo docker exec -it myjenkins \
cat /var/jenkins_home/secrets/initialAdminPassword
```

Felicitaciones. Acaba de desplegara una aplicación usando dockers, algo muy sencillo y ágil. Ahora siga el asistente de instalación web de la aplicación, paso a paso.

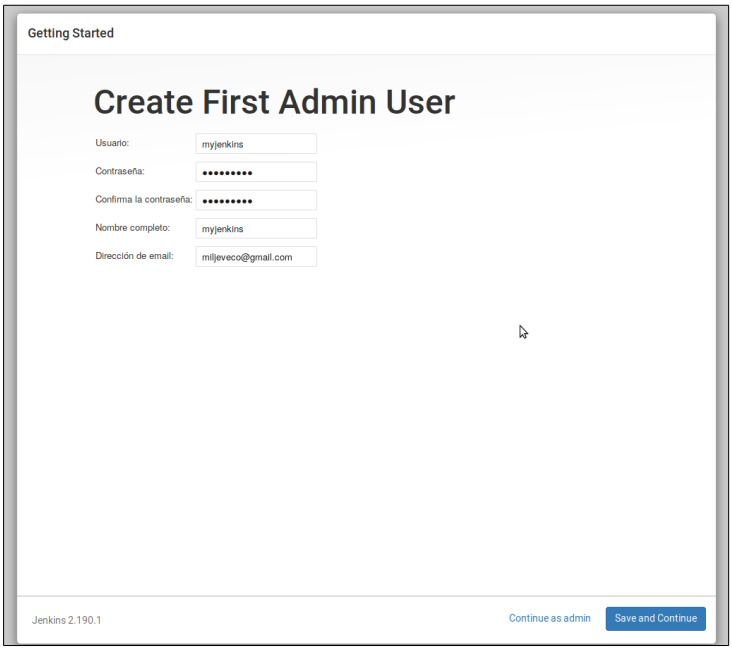
Taller: Despliegue de Aplicaciones con Docker (2/5)

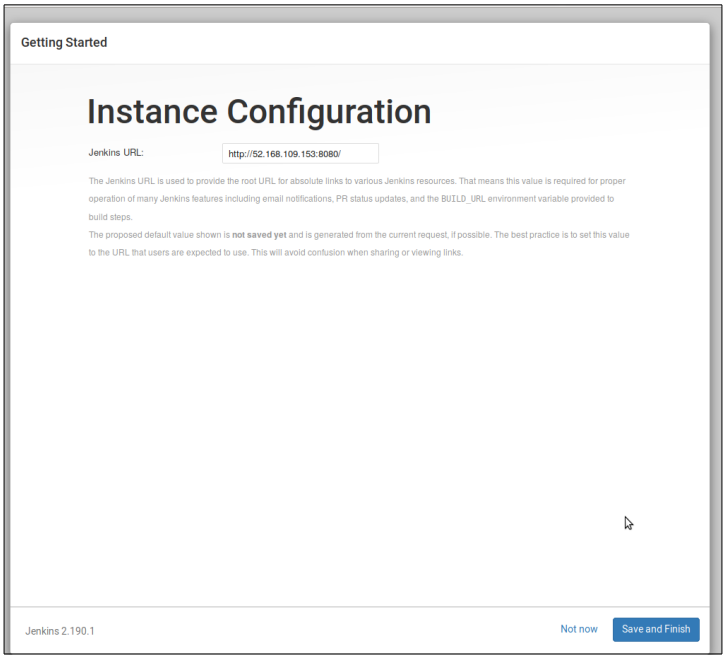
Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17



Taller: Despliegue de Aplicaciones con Docker (2/5)

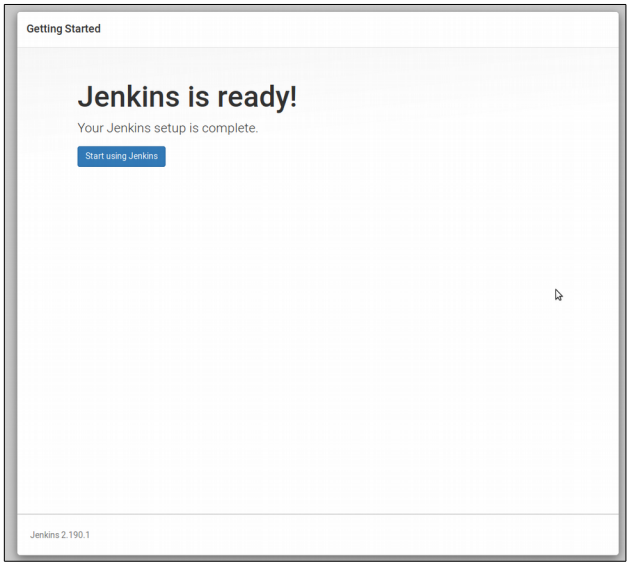
Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

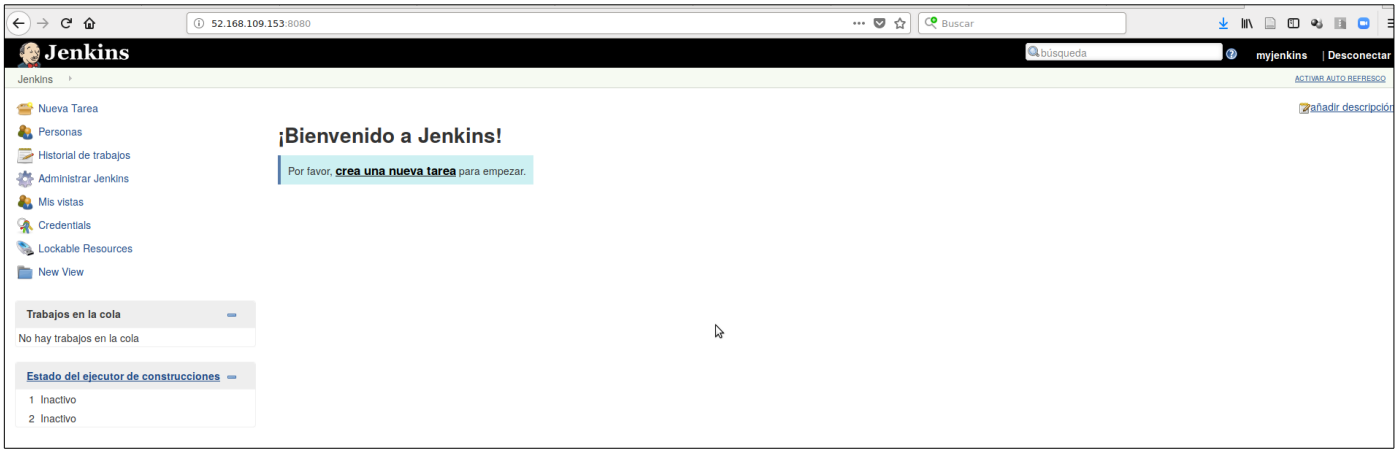




Taller: Despliegue de Aplicaciones con Docker (2/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17





La aplicación lógicamente maneja datos, por lo cual el docker crea un volumen en el cual se almacenan esos datos. Pruebe el comando docker volume ls para ver ese volumen.

```
sudo docker volume ls
```

Aparecerá una imagen como la siguiente.

```
DRIVER              VOLUME NAME
local               6edf4912b75299d39abfff52c1b639f12ebc53bda81bc9d86e0cec1c2c58da2d
```

Taller: Despliegue de Aplicaciones con Docker (2/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

Por defecto, los volúmenes creados usan un indentificador alfanumérico hexadecimal, que docker gestiona internamente. Si se desea, se puede indicar la ruta donde se quiere crear un volumen, para poder realizar copias de respaldo o acceder de manera directa. En el siguiente ejemplo se indicará la ruta del volumen.

Detenga el docker de jenkins usando el siguiente comando:

```
sudo docker stop myjenkins
```

Taller: Despliegue de Aplicaciones con Docker (2/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

2. Docker Base de Datos mariadb. Ahora se va a crear un docker usando una imagen del motor de bases de datos mariadb. Use el siguiente comando

```
sudo docker run -d --name mymariadb \
    -p 3306:3306 \
    --volume=/taller-docker/volumenes/mariadb:/var/lib/mysql \
    -e MYSQL_ROOT_PASSWORD=R00tMySQL% \
    -e MYSQL_DATABASE=wordpressdb \
    -e MYSQL_USER=usrwordpress \
-e MYSQL_PASSWORD=pswdwordpress mariadb:10.3
```

Espere a que finalice, aparecerá una ventana como la siguiente:

```
Unable to find image 'mariadb:10.3' locally
10.3: Pulling from library/mariadb
5667fdb72017: Pull complete
d83811f270d5: Pull complete
ee671aafb583: Pull complete
7fc152dfb3a6: Pull complete
9f669c535a8b: Pull complete
a6de1092ee4e: Pull complete
ee37a2c88dd9: Pull complete
d927a3dd356c: Pull complete
d83c9d39c64f: Pull complete
1b0644883413: Pull complete
dc9f06d5fe30: Pull complete
2c0b65692be7: Pull complete
2643377bb0e1: Pull complete
0a628bd38198: Pull complete
Digest: sha256:fbd0431ea2eb80e0291f92334ff0870e1230bd8d2a6b6bd40fe2a74e53091635
Status: Downloaded newer image for mariadb:10.3
1815d49a3497f4e22063ae062486beb3cf3460e8a6fadd790360a421397cbc4a
```

Ahora ejecute los comandos docker images y docker container ps y observe que hay tres imágenes: la imagen de Hola Mundo, la de jenkins y la de mariadb. Además, hay sólo un contenedor, el de mariadb, pues el de jenkins se detuvo previamente. Use los mismos comandos de los ejemplos anteriores:

```
sudo docker images
```

```
sudo docker container ps
```

Taller: Despliegue de Aplicaciones con Docker (2/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

Observará imágenes como la siguiente:

```
ubuntu@ejemplo2:~/seiiis2019/util$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
jenkins/jenkins     lts                 5721a6cac43c        3 weeks ago         572MB
mariadb             10.3                47dff68107c4        3 weeks ago         343MB
hello-world         latest              fce289e99eb9        9 months ago        1.84kB
ubuntu@ejemplo2:~/seiiis2019/util$ sudo docker container ps
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS              PORTS
                    NAMES
1815d49a3497        mariadb:10.3        "docker-entrypoint.s…"   About a minute ago   Up About a minute   0.0.0
.0:3306->3306/tcp   mymariadb
```

Verifique que se creó el volumen en la ruta indicada en el comando de creación de docker. Use el siguiente comando.

```
sudo ls /taller-docker/volumenes/mariadb/
```

Ahora pruebe el comando para conectarse a la base de datos de mariadb, usando el docker. Use la contraseña que aparece en el comando de creación del docker.

```
sudo docker exec -it mymariadb mysql -u root -p
```

Observará imágenes como la siguiente. Pruebe, según la imagen, listar las bases de datos del docker mariadb.

```
ubuntu@ejemplo2:~/seiiis2019/util$ sudo docker exec -it mymariadb mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.3.18-MariaDB-1:10.3.18+maria~bionic mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpressdb        |
+--------------------+
4 rows in set (0.001 sec)

MariaDB [(none)]>
```

Taller: Despliegue de Aplicaciones con Docker (2/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

Al crear el docker se asigna una dirección IP en una red privada especial para dockers. Pruebe el comando docker inspect para ver la información del docker de mariadb. Los siguientes comandos le ayudarán a identificar fácilmente la dirección IP y otros detalles:

```
sudo docker inspect mymariadb
```

```
sudo docker inspect  --format='{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'  mymariadb
```

```
sudo docker inspect  --format='{{json .Config}}'  mymariadb
```

Usualmente la dirección es 172.17.0.2, pues sólo hay un docker y la dirección 1 corresponde a la subred de dockers. Recuerde la dirección del docker para el siguiente ejemplo.

```
ubuntu@ejemplo2:~/seiiis2019/util$ sudo docker inspect --format='{{range .NetworkSettings.Networks}}{{.IPAddres
s}}{{end}}'  mymariadb
172.17.0.2
ubuntu@ejemplo2:~/seiiis2019/util$ sudo docker inspect --format='{{json .Config}}'   mymariadb
{"Hostname":"1815d49a3497","Domainname":"","User":"","AttachStdin":false,"AttachStdout":false,"AttachStderr":fa
lse,"ExposedPorts":{"3306/tcp":{}},"Tty":false,"OpenStdin":false,"StdinOnce":false,"Env":["MYSQL_ROOT_PASSWORD=
R00tMySQL%","MYSQL_DATABASE=wordpressdb","MYSQL_USER=usrwordpress","MYSQL_PASSWORD=pswdwordpress","PATH=/usr/lo
cal/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin","GOSU_VERSION=1.10","GPG_KEYS=177F4010FE56CA3336300305F1
656F24C74CD1D8","MARIADB_MAJOR=10.3","MARIADB_VERSION=1:10.3.18+maria~bionic"],"Cmd":["mysqld"],"Image":"mariad
b:10.3","Volumes":{"/var/lib/mysql":{}},"WorkingDir":"","Entrypoint":["docker-entrypoint.sh"],"OnBuild":null,"L
abels":{}}
```

Taller: Despliegue de Aplicaciones con Docker (2/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

3. Docker aplicación phpmyadmin. Ahora se va a crear un docker usando una imagen de la aplicación web PHP phpmyadmin, que permite administrar a través de la web la base de datos de mariadb. Use el siguiente comando

```
sudo docker run -d --name myphpadmin \
    -p 80:80 \
    -e PMA_HOST=172.17.0.2 \
    --volume=/taller-docker/volumenes/phpmyadmin:/var/www/html \
phpmyadmin/phpmyadmin
```

Espere a que finalice y pruebe todos los comandos de los ejemplos anteriores.

```
sudo docker images
sudo docker container ps
sudo docker inspect myphpadmin
sudo docker inspect  --format='{{range .NetworkSettings.Networks}}
{{.IPAddress}}{{end}}'  myphpadmin
sudo docker inspect  --format='{{json .Config}}'  myphpadmin
sudo ls /taller-docker/volumenes/phpmyadmin
```

Luego ingrese al navegador y verifique que la aplicación funciona correctamente. Use la contraseña que aparece en el comando de creación del docker de mariadb.

Puesto que los docker de mariadb y de phpmyadmin se crearon usando volúmenes, los datos quedarán respaldados allí incluso cuando se eliminan los docker.

Taller: Despliegue de Aplicaciones con Docker (2/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

4. Docker aplicación wordpress. Ahora se va a crear un docker usando una imagen del popular Gestor de Contenidos (CMS) wordpress, que usa apache, PHP y mariadb. Use el siguiente comando

```
sudo docker run -d --name mywordpress \
    --volume=/taller-docker/volumenes/wordpress:/var/www/html \
    -e WORDPRESS_DB_HOST=172.17.0.2 \
    -e WORDPRESS_DB_USER=usrwordpress \
    -e WORDPRESS_DB_PASSWORD=pswdwordpress \
    -e WORDPRESS_DB_NAME=wordpressdb \
    -p 80:80 \
wordpress
```

Espere a que finalice y pruebe todos los comandos de los ejemplos anteriores.

```
sudo docker images
sudo docker container ps
sudo docker inspect myphpadmin
sudo docker inspect  --format='{{range .NetworkSettings.Networks}}
{{.IPAddress}}{{end}}'  myphpadmin
sudo docker inspect  --format='{{json .Config}}'  myphpadmin
sudo ls /taller-docker/volumenes/phpmyadmin
```

Luego ingrese al navegador y verifique que la aplicación funciona correctamente. Use la contraseña que aparece en el comando de creación del docker. Termine el proceso de instalación y configuración.

Felicitaciones, ya ha usado cinco (5) imágenes de contenedores y ha desplegado cinco (5) contenedores: hola mundo, mariadb, jenkins, phpmyadmin y wordpress. Además, los dockers de phpmyadmin y wordpress se comunican y usan el docker de mariadb.

Taller: Despliegue de Aplicaciones con Docker (2/5)

Milton Jesús Vera Contreras - miltonjesusvc@ufps.edu.co
Seminario Internacional de Investigación en Ingeniería del Software
Medellín, 2019/10/17

5. Docker SO Oracle linux. Ahora se va a crear un docker usando una imagen del popular Sistema Operativo Oracle Linux, en el cual podría desplegar bases de datos y aplicaciones usando los marcos de trabajo y herramientas de Oracle (aunque sólo para aprendizaje por temas de licencias).

```
sudo docker run -it oraclelinux /bin/bash
```

En este caso el docker no se ejecuta como un proceso persistente o demonio (dameon) sino que se ejedcuta de manera interactiva. Por lo tanto, ahora no está en su servidor sino en un servidor "dockerizado", una abstracción o virtualización del sistema operativo que se maneja con docker. Pruebe dentro del contenedor los siguientes comandos:

```
uname -a
cat /etc/oracle-release
cat /etc/redhat-release
```

Pruebe todos los comandos de los ejemplos anteriores para revisar todas las imágenes y contenedores:

```
ubuntu@ejemplo2:~/seiiis2019/util$ sudo docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
wordpress               latest              fc466e7eaa8e        17 hours ago        542MB
jenkins/jenkins         lts                 5721a6cac43c        3 weeks ago         572MB
phpmyadmin/phpmyadmin   latest              8ad87a8e95c4        3 weeks ago         458MB
mariadb                 10.3                47dff68107c4        3 weeks ago         343MB
oraclelinux             latest              5f993b1aafe5        2 months ago        235MB
hello-world             latest              fce289e99eb9        9 months ago        1.84kB
ubuntu@ejemplo2:~/seiiis2019/util$ sudo docker container ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
                    NAMES
d50b67e846d0        wordpress           "docker-entrypoint.s…"   6 minutes ago       Up 5 minutes        0.0.0
0:80->80/tcp        mywordpress
1815d49a3497        mariadb:10.3        "docker-entrypoint.s…"   23 minutes ago      Up 23 minutes       0.0.0
0:3306->3306/tcp    mymariadb
```

Detenga todos los contenedores, usando el comando docker stop, seguido de los nombres de todos los contenedores.