

Christian David Lopez Morcillo  
Luisa Maria Vivas

## # sd-docker-assignment

### Objetivo:

En este repositorio incluya los pasos y las evidencias para el despliegue de un mirror con postgresql ó mysql y el empleo del mirror desde un contenedor de docker.

### Procedimiento:

Para la realización de esta tarea, se procede a instalar un mirror con la herramienta aptly sobre una maquina aprovisionada con vagrant. Luego se aprovisiona un contenedor de docker con postgres que consumirá los servicios del mirror.

### Pruebas:

#### 1. Instalación de aptly:

Se adiciona el repositorio de aptly a archivo sources.list.

```
* echo deb http://repo.aptly.info/ squeeze main >> /etc/apt/sources.list
```

Se adiciona la key:

```
*sudo apt-key adv --keyserver keys.gnupg.net --recv-keys 9E3E53F19C7DE460
```

Se realiza un apt-get update

```
*apt-get update
```

Se instala el paquete de aptly

```
*apt-get install aptly
```

#### 2. Creacion del mirror en aptly:

Se genera una key RSA para crear los repositorios de aptly

```
*gpg --gen-key
```

Generamos entropia

```
*cat /dev/urandom
```

Importamos la key creada

```
*gpg --no-default-keyring --keyring /usr/share/keyrings/ubuntu-archive-keyring.gpg --export | gpg
--no-default-keyring --keyring trustedkeys.gpg --import
```

Creamos un repositorio llamado xenial-main-python3, con los siguientes paquetes mysql-server, python3, postgresql, postgresql-contrib.

```
*aptly mirror create -architectures=amd64 -filter='Priority (required) | Priority (important) | Priority
(standard) | mysql-server | python3 | postgresql | postgresql-contrib' -filter-with-deps xenial-main-
python3 http://mirror.upb.edu.co/ubuntu/ xenial main
```

Actualizamos el mirror para hacer un cache de los paquetes a guardar.

```
*aptly mirror update xenial-main-python3
```

Creamos un snapshot del mirror para publicarlo.

```
*aptly snapshot create xenial-snapshot-python3 from mirror xenial-main-python3
```

Publicamos el mirror.

```
*aptly publish snapshot xenial-snapshot-python3
```

Exportamos la key para acceder a los repositorios del mirror.

```
*gpg --export --armor > my_key.pub
```

Corremos el servicio:

```
*aptly serve
```

### 3. Aprovisionamiento del contenedor docker con postgres:

Creamos un dockerfile con la siguiente configuración:

```
FROM ubuntu:16.04
MAINTAINER cdlopezmorcillo@gmail.com

ADD config/my_key.pub /tmp

#Configure repository
RUN apt-key add /tmp/my_key.pub
RUN rm -f /tmp/my_key.pub
RUN echo "deb http://192.168.131.121:8080/ xenial main" > sources.list
RUN chmod 777 /tmp

#Install packages
RUN apt-get clean
```

RUN apt-get update -y  
RUN apt-get install postgresql -y  
RUN apt-get install postgresql-contrib -y

#Giving permissions  
EXPOSE 5432

Creamos una subinterfaz que nos permita acceder al mirror y sacar el my\_key.pub. Luego hace ping al mirror

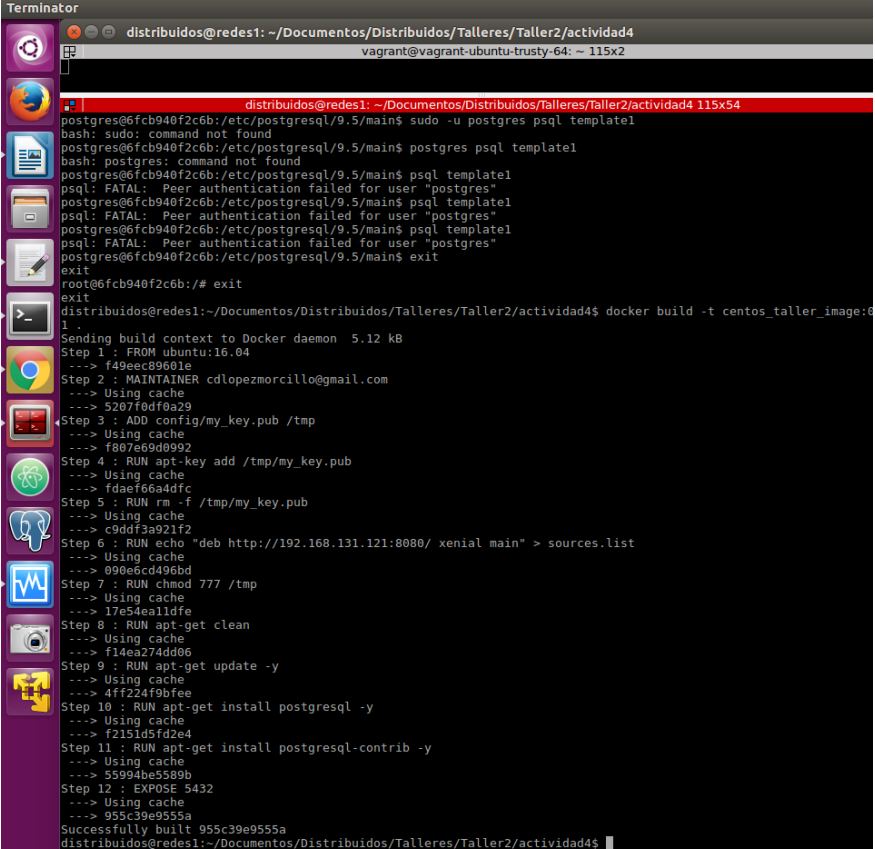
\*sudo ifconfig enp5s0:0 192.168.131.129

\*ping 192.168.131.121

\*Sacamos la key del mirror y lo guardamos en la carpeta config

A partir del archivo dockerfile creamos una imagen docker llamada centos\_taller\_image.

\*docker build -t centos\_taller\_image:0.0.1 .

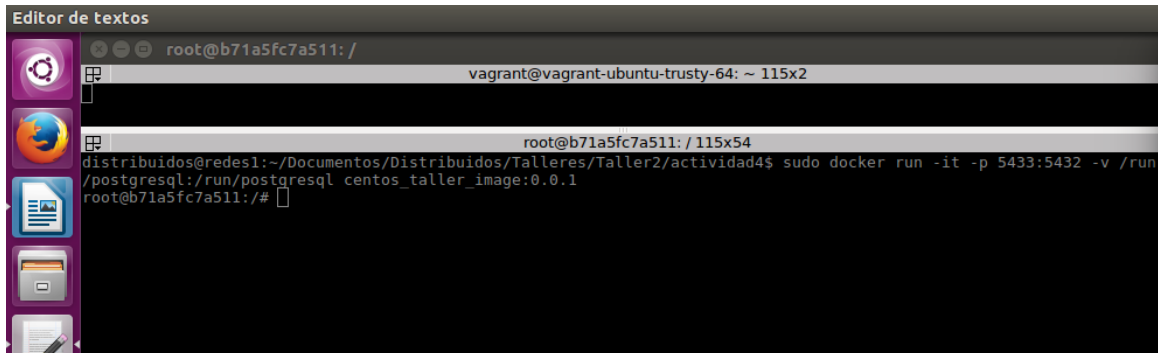


```
Terminator
distribuidos@redes1: ~/Documentos/Distribuidos/Talleres/Taller2/actividad4
vagrant@vagrant-ubuntu-trusty-64: ~ 115x2

distribuidos@redes1: ~/Documentos/Distribuidos/Talleres/Taller2/actividad4 115x54
postgres@6fcb940f2c6b:/etc/postgresql/9.5/main$ sudo -u postgres psql template1
bash: sudo: command not found
postgres@6fcb940f2c6b:/etc/postgresql/9.5/main$ postgres psql template1
bash: postgres: command not found
postgres@6fcb940f2c6b:/etc/postgresql/9.5/main$ psql template1
psql: FATAL: Peer authentication failed for user "postgres"
postgres@6fcb940f2c6b:/etc/postgresql/9.5/main$ psql template1
psql: FATAL: Peer authentication failed for user "postgres"
postgres@6fcb940f2c6b:/etc/postgresql/9.5/main$ psql template1
psql: FATAL: Peer authentication failed for user "postgres"
postgres@6fcb940f2c6b:/etc/postgresql/9.5/main$ exit
exit
root@6fcb940f2c6b:/# exit
exit
distribuidos@redes1:~/Documentos/Distribuidos/Talleres/Taller2/actividad4$ docker build -t centos_taller_image:0.0.1 .
Sending build context to Docker daemon 5.12 kB
Step 1 : FROM ubuntu:16.04
--> f49eec89601e
Step 2 : MAINTAINER cdlopezmorcillo@gmail.com
--> Using cache
--> 5207f8df0a29
Step 3 : ADD config/my_key.pub /tmp
--> Using cache
--> f807e69d0992
Step 4 : RUN apt-key add /tmp/my_key.pub
--> Using cache
--> fd8ef66a4dfc
Step 5 : RUN rm -f /tmp/my_key.pub
--> Using cache
--> c9ddf3a921f2
Step 6 : RUN echo "deb http://192.168.131.121:8080/ xenial main" > sources.list
--> Using cache
--> 090e6cd496bd
Step 7 : RUN chmod 777 /tmp
--> Using cache
--> 17e54ealldfe
Step 8 : RUN apt-get clean
--> Using cache
--> f14ea274dd06
Step 9 : RUN apt-get update -y
--> Using cache
--> 4ff224f9bfee
Step 10 : RUN apt-get install postgresql -y
--> Using cache
--> f2151d5fd2e4
Step 11 : RUN apt-get install postgresql-contrib -y
--> Using cache
--> 55994be5589b
Step 12 : EXPOSE 5432
--> Using cache
--> 955c39e9555a
Successfully built 955c39e9555a
distribuidos@redes1:~/Documentos/Distribuidos/Talleres/Taller2/actividad4$
```

A partir de la imagen centos\_taller\_image:0.0.1 creamos un contenedor. Además mapeamos el puerto 5433 del contenedor de al 5432 del host.

\*sudo docker run -it -p 5433:5432 --rm centos\_taller\_image:0.0.1 /bin/bash

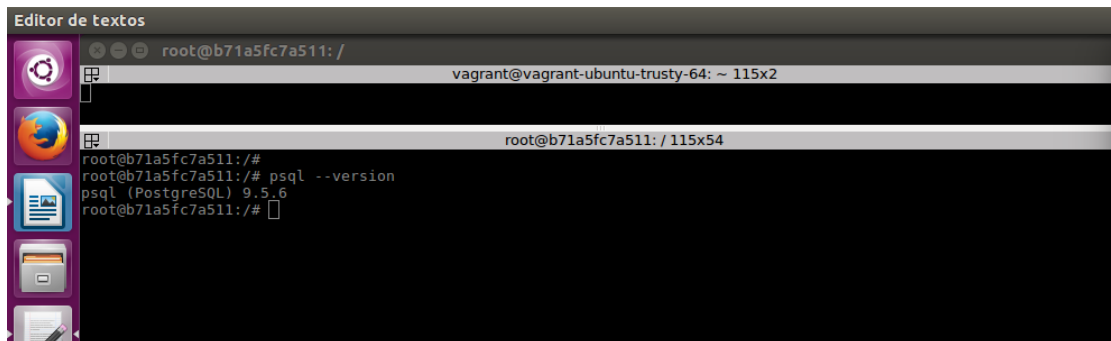


```
Editor de textos
root@b71a5fc7a511: /
vagrant@vagrant-ubuntu-trusty-64: ~ 115x2

root@b71a5fc7a511: / 115x54
distribuidos@redes1:~/Documentos/Distribuidos/Talleres/Taller2/actividad4$ sudo docker run -it -p 5433:5432 -v /run
/postgresql:/run/postgresql centos_taller_image:0.0.1
root@b71a5fc7a511:/#
```

Finalmente obtenemos el contenedor con el servidor postgres instalado. Entramos por consola y verificamos que este instalado verificando la versión instalada.

\*psql --version



```
Editor de textos
root@b71a5fc7a511: /
vagrant@vagrant-ubuntu-trusty-64: ~ 115x2

root@b71a5fc7a511: / 115x54
root@b71a5fc7a511:/#
root@b71a5fc7a511:/# psql --version
psql (PostgreSQL) 9.5.6
root@b71a5fc7a511:/#
```

Luego iniciamos el servicio postgres

\*/etc/init.d/postgresql restart

Entonces cambiamos de usuario a postgres

\*su - postgres

Ejecutamos la consola

\*psql

```
Terminator
root@0deacd47a04c: /
vagrant@vagrant-ubuntu-trusty-64: ~ 115x2

root@0deacd47a04c: /115x31
You are now connected to database "template1" as user "postgres".
template1=# \t
tuples only is off.
template1=# \d
No relations found.
template1=# \dt
No relations found.
template1=# \
Invalid command \. Try \? for help.
template1=# \d
No relations found.
template1=# \d
No relations found.
template1=# create table distribuidos (nombre varchar(100), edad int)
template1=# CREATE TABLE distribuidos (nombre varchar(100), edad int)
template1=# \d
No relations found.
template1=# create table distribuidos (nombre varchar(100), edad int)
CREATE TABLE distribuidos (nombre varchar(100), edad int);
ERROR:  syntax error at or near "/"
LINE 1: /set template1
        ^
template1=# create table distribuidos (nombre varchar(100), edad int);
CREATE TABLE
template1=# \d
List of relations
-----
Schema | Name          | Type  | Owner
-----
public | distribuidos  | table | postgres
(1 row)
```

Creamos en el esquema template1 , creamos una tabla distribuidos , introducimos unos registros y luego hacemos consultas de ellos:

- \* \c template1
- \* create table distribuidos (nombre varchar(100), edad int);
- \* insert into distribuidos (nombre,edad) values ('christian', 23)
- \* select \* from distribuidos

```
Terminator
root@0deacd47a04c: /
vagrant@vagrant-ubuntu-trusty-64: ~ 115x2

root@0deacd47a04c: /115x31
edad | integer |
Table "public.distribuidos"
Column | Type          | Modifiers
-----
nombre | character varying(100) |
edad   | integer       |
template1=# insert into distribuidos values (christian, 23)
template1=# ;
ERROR:  column "christian" does not exist
LINE 1: insert into distribuidos values (christian, 23)
                                         ^
template1=# insert into distribuidos (nombre,edad) values (christian, 23)
;
ERROR:  column "christian" does not exist
LINE 1: insert into distribuidos (nombre,edad) values (christian, 23...
                                         ^
template1=# insert into distribuidos (nombre,edad) values ('christian', 23)
INSERT 0 1
template1=# ;
template1=# select * from distribuidos
template1=# ;
 nombre | edad
-----
christian | 23
(1 row)
template1=#
```