

# Containerization and Big Data: Exercise 2-2

Brandon Hosley

June 19, 2020

## 2 Dockerizing PostgreSQL Database

### 2.5 Custom Dockerfile for PostgreSQL

```
bhos12@us20041ts: ~/myPostgreSQL
bhos12@us20041ts:~/myPostgreSQL$ cat > Dockerfile
#
# example Dockerfile for https://docs.docker.com/engine/examples/postgresql_service/
#
FROM ubuntu:16.04

# Add the PostgreSQL GPG key to verify their Debian packages.
# It should be the same key as https://www.postgresql.org/media/keys/ACCC4CF8.asc
RUN apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-keys B97B0AFCA1A47F044F244A07FCC7D46ACCC4CF8

# Add PostgreSQL's repository. It contains the most recent stable release
# of PostgreSQL, ``9.3``.
RUN echo "deb http://apt.postgresql.org/pub/repos/apt/ precise-pgdg main" > /etc
/apr/sources.list.d/pgdg.list

# Install ``python-software-properties``, ``software-properties-common`` and Pos
tgreSQL 9.3
# There are some warnings (in red) that show up during the build. You can hide
# them by prefixing each apt-get statement with DEBIAN_FRONTEND=noninteractive
RUN apt-get update && apt-get install -y python-software-properties software-pro
perties-common postgresql-9.3 postgresql-client-9.3 postgresql-contrib-9.3

# Note: The official Debian and Ubuntu images automatically ``apt-get clean``
# after each ``apt-get``

# Run the rest of the commands as the ``postgres`` user created by the ``postgre
s-9.3`` package when it was ``apt-get installed``
USER postgres

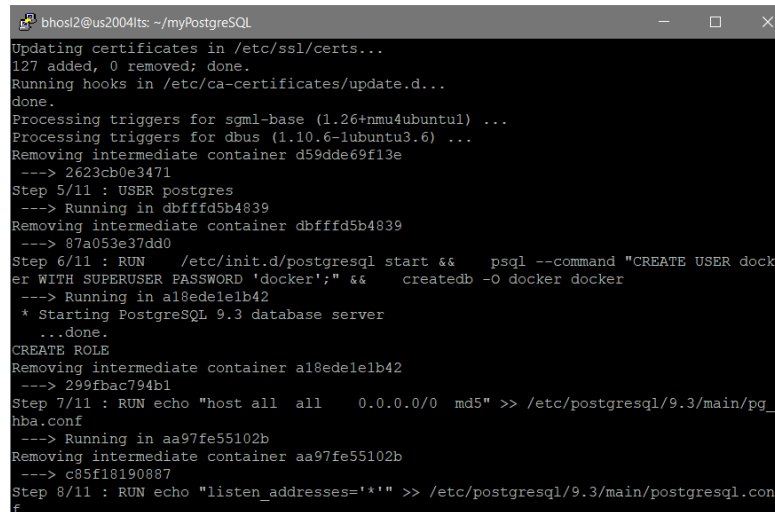
# Create a PostgreSQL role named ``docker`` with ``docker`` as the password and
# then create a database ``docker`` owned by the ``docker`` role.
# Note: here we use ``&&`` to run commands one after the other - the ``\``
# allows the RUN command to span multiple lines.
RUN /etc/init.d/postgresql start && \
    psql --command "CREATE USER docker WITH SUPERUSER PASSWORD 'docker';" && \
    createdb -O docker docker

# Adjust PostgreSQL configuration so that remote connections to the
# database are possible.
RUN echo "host all all 0.0.0.0/0 md5" >> /etc/postgresql/9.3/main/pg_hba.co
nf
```

Dockerfile example taken from

[https://docs.docker.com/engine/examples/postgresql\\_service/](https://docs.docker.com/engine/examples/postgresql_service/)

## 2.6 Building Custom PostgreSQL Docker Image



```
bhosl2@us2004lts: ~/myPostgreSQL
Updating certificates in /etc/ssl/certs...
127 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for sgml-base (1.26+nmu4ubuntu1) ...
Processing triggers for dbus (1.10.6-1ubuntu3.6) ...
Removing intermediate container d59dde69f13e
--> 2623cb0e3471
Step 5/11 : USER postgres
--> Running in dbfffd5b4839
Removing intermediate container dbfffd5b4839
--> 87a053e37dd0
Step 6/11 : RUN /etc/init.d/postgresql start && psql --command "CREATE USER dock
er WITH SUPERUSER PASSWORD 'docker';" && createdb -O docker docker
--> Running in a18ede1e1b42
* Starting PostgreSQL 9.3 database server
...done.
CREATE ROLE
Removing intermediate container a18ede1e1b42
--> 299fbac794b1
Step 7/11 : RUN echo "host all all 0.0.0.0/0 md5" >> /etc/postgresql/9.3/main/pg_
hba.conf
--> Running in aa97fe55102b
Removing intermediate container aa97fe55102b
--> c85f18190887
Step 8/11 : RUN echo "listen_addresses='*'" >> /etc/postgresql/9.3/main/postgresql.con
f
```

The Dockerfile instructs Docker to build the container:

1. Use Ubuntu as a base
2. Set PostgreSQL PGP public key to ensure connection to the correct source
3. Add address to the PostgreSQL repository
4. Apply updates to:
  - 4.1. OS
  - 4.2. Python software
  - 4.3. PPA updates
  - 4.4. PostgreSQL
  - 4.5. PostgreSQL Client
  - 4.6. PostgreSQL Additional Features
5. Change to a specific user
6.
  - 6.1. Start PostgreSQL
  - 6.2. Create a Docker user with a password
  - 6.3. Create a Database with Docker as the owner
7. Set the authentication method to allow all connections
8. Set a listening address for the Docker
9. Expose port 5432 (Default for PostgreSQL)
10. Add volumes to persist data beyond single container
11. CMD runs to complete building the image

## 2.7 Container Creation and Verification

```
bhosl2@us2004lts: ~/myPostgreSQL
---> 32cfe76a9682
Step 10/11 : VOLUME ["etc/postgresql", "/var/log/postgresql", "/var/lib/postgresql"]
---> Running in 2e3684406272
Removing intermediate container 2e3684406272
---> f33f97f4517b
Step 11/11 : CMD ["/usr/lib/postgresql/9.3/bin/postgres", "-D", "/var/lib/postgresql/9.3/main", "-c", "config_file=/etc/postgresql/9.3/main/postgresql.conf"]
---> Running in 1350593a47ac
Removing intermediate container 1350593a47ac
---> a9fbdf45f42e
Successfully built a9fbdf45f42e
Successfully tagged mypostgresql:latest
bhosl2@us2004lts:~/myPostgreSQL$ docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
mypostgresql         latest             a9fbdf45f42e       2 hours ago        395MB
ubuntu              16.04             330ae480cb85       2 days ago         125MB
postgres            latest             b97bae343e06       9 days ago         313MB
hello-world         latest             bf756fblae65       5 months ago       13.3kB
bhosl2@us2004lts:~/myPostgreSQL$
```

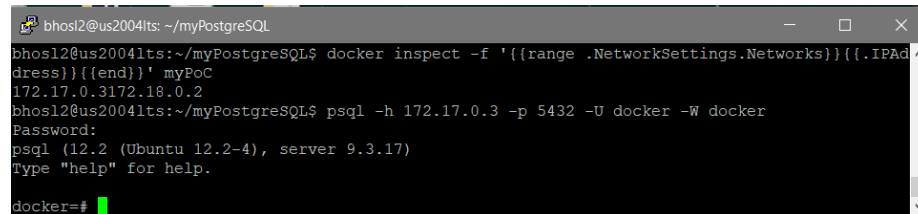
```
bhosl2@us2004lts:~/myPostgreSQL$ docker container ls -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
02bb2f5e5698       mypostgresql       "/usr/lib/postgres..." 41 seconds ago      Up 40 seconds      5432/tcp           mypoc
79d0df25a321       postgres:latest    "docker-entrypoint.s..." 5 days ago          Up 5 days          5432/tcp           mypostgresqlname
6e7d54b2c659       hello-world        "/hello"            2 weeks ago         Exited (0) 2 weeks ago                               nostalgic_lehmann

bhosl2@us2004lts:~/myPostgreSQL$ docker exec -it mypoc bash
postgres@02bb2f5e5698:/# service postgresql status
9.3/main (port 5432): online
postgres@02bb2f5e5698:/# psql -h localhost -p 5432 -U docker -W docker
Password for user docker:
psql (9.3.17)
SSL connection (cipher: DHE-RSA-AES256-GCM-SHA384, bits: 256)
Type "help" for help.

docker-# \q
postgres@02bb2f5e5698:/#
```

## 2.8 Connection from ...

### 2.8.1 Host System

A terminal window titled 'bhosl2@us2004lts: ~/myPostgreSQL' showing the following commands and output:

```
bhosl2@us2004lts:~/myPostgreSQL$ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}' myPoC
172.17.0.3172.18.0.2
bhosl2@us2004lts:~/myPostgreSQL$ psql -h 172.17.0.3 -p 5432 -U docker -W docker
Password:
psql (12.2 (Ubuntu 12.2-4), server 9.3.17)
Type "help" for help.

docker=#
```

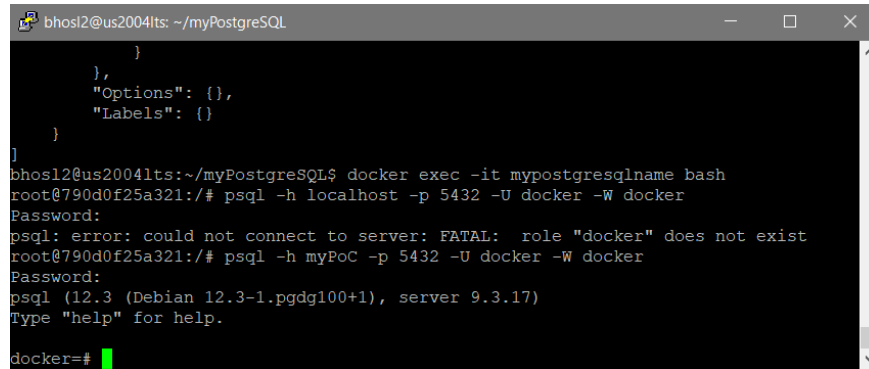
```
bhosl2@us2004lts:~/myPostgreSQL$ sudo sh -c 'echo "deb
→ http://apt.postgresql.org/pub/repos/apt $(lsb_release
→ -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
```

```
bhosl2@us2004lts:~/myPostgreSQL$ sudo apt install -y
→ postgresql-client
```

```
bhosl2@us2004lts:~/myPostgreSQL$ docker inspect -f
→ '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
→ myPoC 172.17.0.3172.18.0.2
bhosl2@us2004lts:~/myPostgreSQL$ psql -h 172.17.0.3 -p 5432 -U
→ docker -W docker
Password:
psql (12.2 (Ubuntu 12.2-4), server 9.3.17)
Type "help" for help.

docker=#
```

## 2.8.2 Another Container



```
bhosl2@us2004lts: ~/myPostgreSQL
}
},
"Options": {},
"Labels": {}
}
]
bhosl2@us2004lts:~/myPostgreSQL$ docker exec -it mypostgresqlname bash
root@790d0f25a321:/# psql -h localhost -p 5432 -U docker -W docker
Password:
psql: error: could not connect to server: FATAL:  role "docker" does not exist
root@790d0f25a321:/# psql -h myPoC -p 5432 -U docker -W docker
Password:
psql (12.3 (Debian 12.3-1.pgdg100+1), server 9.3.17)
Type "help" for help.

docker=#
```

```
bhosl2@us2004lts:~/myPostgreSQL$ docker container ls -a
CONTAINER ID        IMAGE               COMMAND
→ CREATED
02bb2f5e5698        mypostgresql       "/usr/lib/postgresql..."
→ 12 minutes ago
790d0f25a321        postgres:latest    "docker-entrypoint.s..."
→ 5 days ago
6e7d54b2c659        hello-world        "/hello"
→ 2 weeks ago

bhosl2@us2004lts:~/myPostgreSQL$ docker network create
→ myPostgresqlNet
3a7aabb38c3c5dbe203eefae75ee7654e9657124796eee71451d4688e8227b81

bhosl2@us2004lts:~/myPostgreSQL$ docker network connect
→ myPostgresqlNet myPoC
bhosl2@us2004lts:~/myPostgreSQL$ docker network connect
→ myPostgresqlNet mypostgresql
bhosl2@us2004lts:~/myPostgreSQL$ docker network inspect
→ myPostgresqlNet

root@790d0f25a321:/# psql -h myPoC -p 5432 -U docker -W docker
Password:
psql (12.3 (Debian 12.3-1.pgdg100+1), server 9.3.17)
Type "help" for help.

docker=#
```