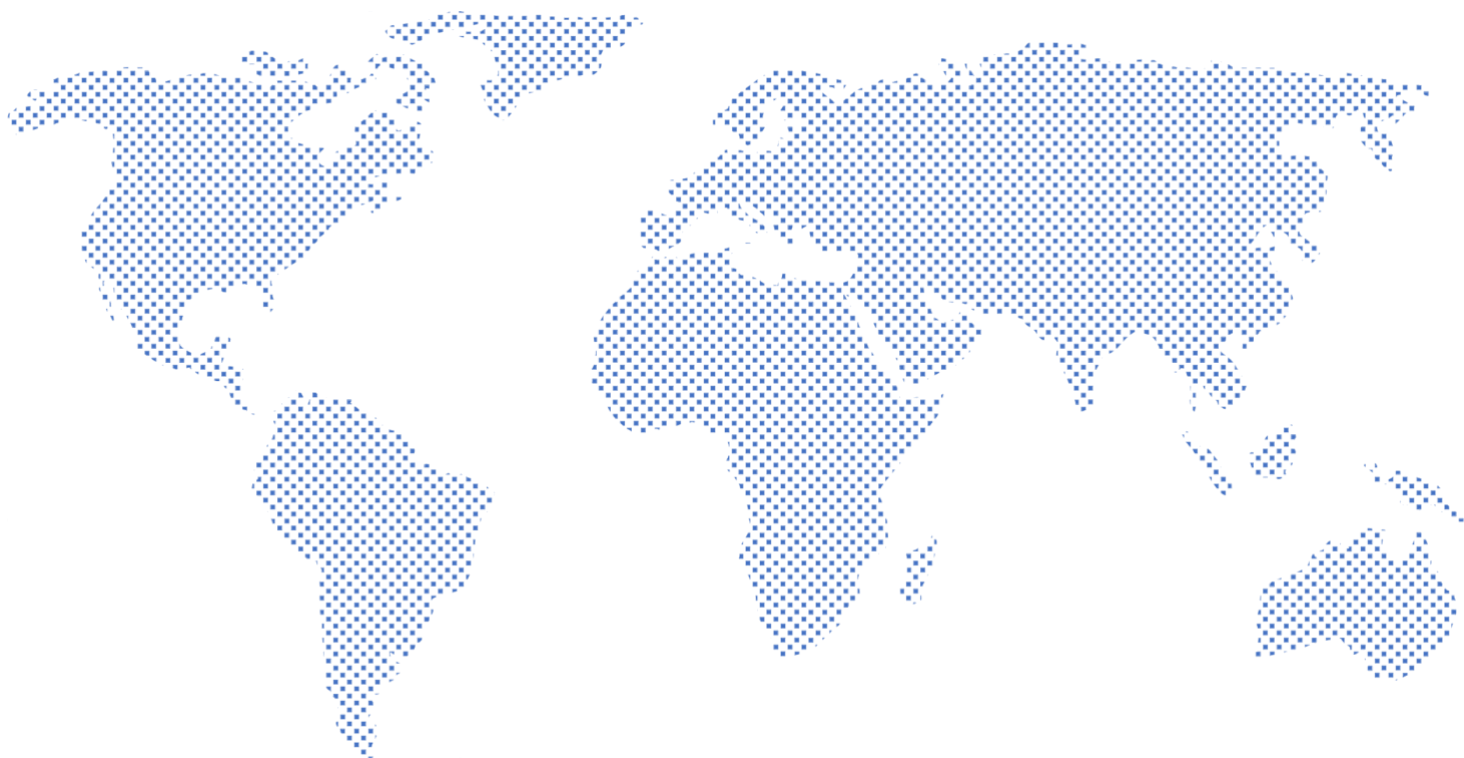


Pega Docker Postgres/Tomcat Container



SRINIVASA KANDRU

Sreesoft Solutions



Contents

| | |
|---|----|
| About this document..... | 3 |
| Disclaimer & Licensing Terms..... | 3 |
| Check Prerequisites..... | 3 |
| 1. Extract Pega Database Backup & prweb from PE (Personal Edition)..... | 5 |
| 1.1 Extract/copy prweb war file from PE folder | 6 |
| 2. Build Docker Image(s)..... | 7 |
| 3. Configure Pega Postgres image..... | 9 |
| 3.1 Export Backup file via host ‘psql’ utility. | 10 |
| 3.2 Upload backup file into container and load..... | 11 |
| 3.3 Test Postgres Connection | 12 |
| 4. Configure Pega Tomcat image | 14 |
| 5. Running Pega Instance(s) | 14 |

About this document

Currently Pega is available for learning in the form of personal edition (PE) or personal virtual machine (PVM). PE works only on windows & PVS does work on various platforms but eat up many resources. Also, in general tech world, for ease of development & deployment to cloud environments, Docker is used widely. There is no official docker version of Pega available. There seems a docker version just running tomcat but not any database which is key part of Pega installation. This document is to detail the steps needed to run Pega in docker containers.

This docker image can be configured in any of the following OS's

- Windows 10
- Mac OS X
- Linux

Disclaimer & Licensing Terms

This document is only for educational purposes. Whoever following it, must understand Pega's Personal Edition licencing terms and adhere to it. This document does not distribute Pega PE software, this document is only for who have Pega PE authorised software and wants to learn how to fit that into multiple work environments. It's purely for educational purposes.

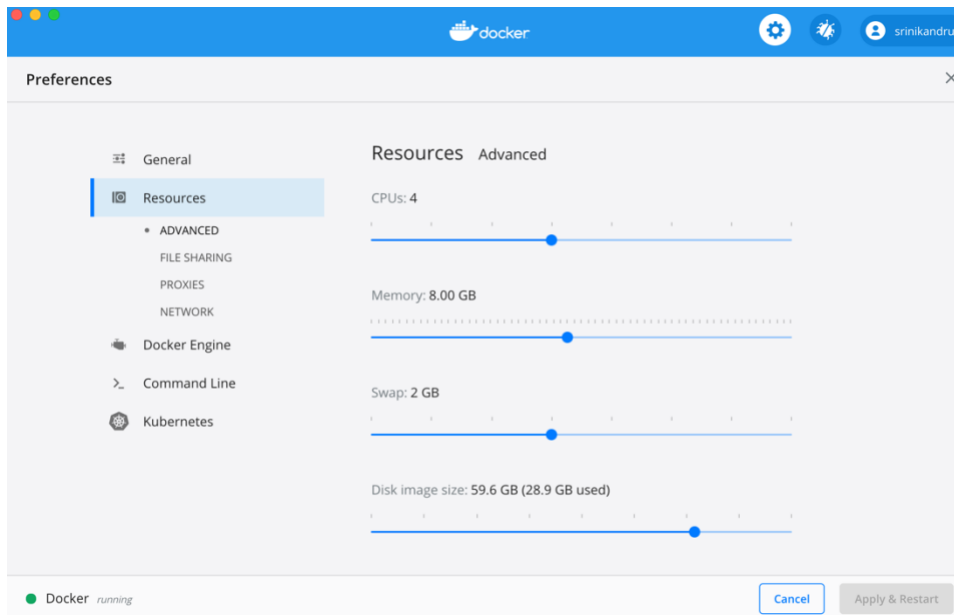
There is no guarantee the below steps work in a specific environment. Please use caution while following the instructions.

Check Prerequisites

Before installing Pega Docker images, ensure all the prerequisites are met and available in the system.

1. Ensure Docker software installed and running in the target system. Docker can be downloaded from here. <https://www.docker.com/products/docker-desktop>
2. Once installed, keep docker resource settings optimal for Pega. It is recommended to have 8GB of memory allocated. It may vary depends on target system capacity.

In Windows or Mac OS X, docker preferences can be viewed graphically as below.



In Linux distros, Docker share host OS capacities usually. It can be configured following respective Linux distro CLI commands. To see how the resources allocated to Docker, run the below command and see.

```
docker info
```

3. Have a GitHub account and familiarise how to check-out repositories.
4. Install git command line utility as described here.

<https://www.atlassian.com/git/tutorials/install-git>

It may be required to authenticate local system with GitHub to allow check-out a repository. Please follow the below articles to resolve any issues if they arise.

<https://docs.github.com/en/free-pro-team@latest/github/authenticating-to-github/adding-a-new-ssh-key-to-your-github-account>

<https://docs.github.com/en/free-pro-team@latest/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

```
srinivasa@srinivasa:~$ ssh-keygen -t rsa -b 4096 -C "Srinivasa .com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/srinivasa/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/srinivasa/.ssh/id_rsa
Your public key has been saved in /home/srinivasa/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vHL04q9Bz/r02EWG1HB5rzXLs+keSFUQesLd/z4ANls Srinivasa .com
The key's randomart image is:
+---[RSA 4096]-----+
|  . .00. |
|  =.00. |
|  =.00. |
|  . . =E..+ |
|  S O B..o+ |
|  . o . +.O+. |
|  o *   ....= |
|  .O + .   =. |
|  .+=* o   ooo |
+---[SHA256]-----+
srinivasa@srinivasa:~$
```

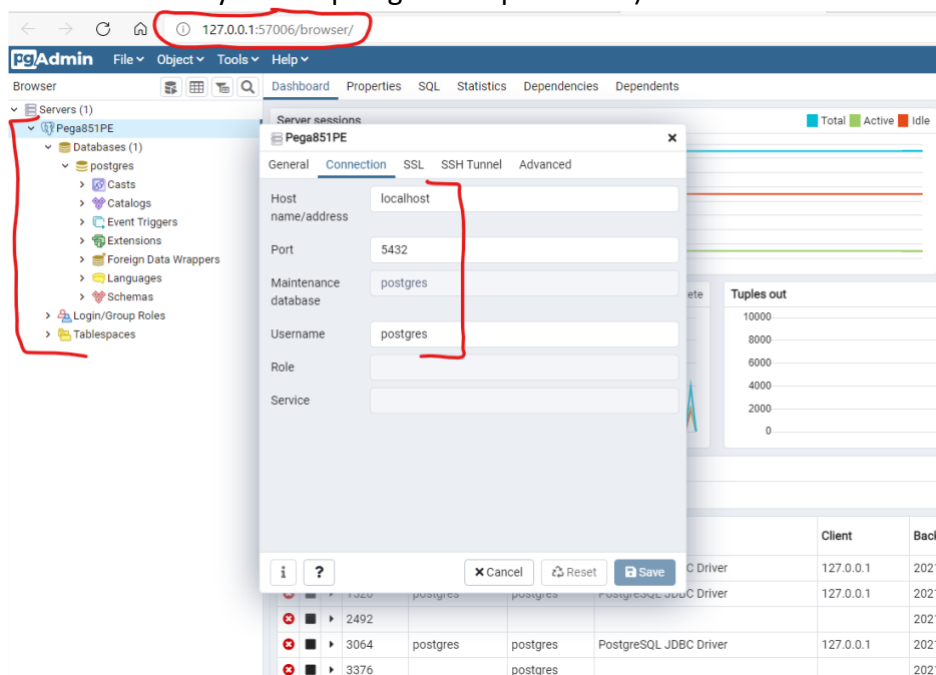
5. Install pgAdmin 4 client.

6. Keep/Extract Pega Database backup from Personal Edition. To extract, follow the related section in this document.

1. Extract Pega Database Backup & prweb from PE (Personal Edition)

To run Pega in Docker containers, we are relying on postgres database backup extracted from a running Pega instance (Either from PE or other available Pega instance). In this case, it is going to be Personal Edition of Pega. (Following steps can be applied to any other pega instance running on postgres)

1. To extract database backup Postgres DB, we are going to use **pg_dump** tool. (this tool comes with pgAdmin 4 standard installation)
2. Install Pega personal edition and ensure it is working normally.
3. Check/Configure DB access with pre-installed pgadmin-4 client as shown in below screenshot. (password is usually either 'postgres' or 'password')



4. Below example is taken from Windows, for other OS, go through the postgres recommendations.
5. Run below command (or similar as per your environment) to extract postgres DB backup to a chosen folder (For ex : "C:\Users\Srinivasa\Temp")

In Windows :

```
C:\Program Files\pgAdmin 4\v4\runtime> .\pg_dump.exe -f
```

```
"C:\Users\Srinivasa\Temp\PegaPE_851_DB_Postgres.sql" -U postgres -W postgres
```

```

PS C:\Program Files\pgAdmin 4\4\runtime> pwd
Path
C:\Program Files\pgAdmin 4\4\runtime

PS C:\Program Files\pgAdmin 4\4\runtime> ls

Directory: C:\Program Files\pgAdmin 4\4\runtime

Mode                LastWriteTime         Length Name
----                -
d-----          07/02/2021    20:55          imageformats
d-----          07/02/2021    20:55          platforms
a-----          18/01/2021    15:03         15360 comerr64.dll
a-----          18/01/2021    15:03        384512 gssapi64.dll
a-----          18/01/2021    15:03         64000 k5sprt64.dll
a-----          18/01/2021    15:05         27136 kinit.exe
a-----          18/01/2021    15:03       1210368 krsn_64.dll
a-----          01/10/2020    13:34       2840064 libcrypto-1_1-x64.dll
a-----          26/11/2020    15:38        301056 libpq.dll
a-----          01/10/2020    13:34       678840 libssl-1_1-x64.dll
a-----          25/01/2021    14:39        588288 pgAdmin4.exe
a-----          26/11/2020    15:39        456704 pg_dump.exe
a-----          26/11/2020    15:39       1361192 pg_dumpall.exe
a-----          26/11/2020    15:39       213504 pg_restore.exe
a-----          26/11/2020    15:39       499200 psql.exe
a-----          23/09/2020    17:00       100368 python.exe
a-----          23/09/2020    17:00       4285564 python3.dll
a-----          23/09/2020    17:00        98832 pythons.exe
a-----          25/01/2021    14:39         28 qt.conf
a-----          27/03/2020    13:18       6193272 Qt5Core.dll
a-----          27/03/2020    13:18       7129720 Qt5Gui.dll
a-----          27/03/2020    13:18       1381496 Qt5Network.dll
a-----          27/03/2020    18:41       337520 Qt5Svg.dll
a-----          27/03/2020    13:18       5596280 Qt5Widgets.dll
a-----          29/05/2020    13:24         84992 zlib.dll

PS C:\Program Files\pgAdmin 4\4\runtime> .\pg_dump.exe -f "C:\Users\Srinivasa\Temp\PegaPE_851_DB_Postgres.sql" -U postgres -W postgres
Password:
PS C:\Program Files\pgAdmin 4\4\runtime>

```

In Linux (with remote postgres) :

```
/usr/bin/pg_dump --host "192.168.2.111" --port "5432" --username "postgres" --password --dbname "postgres" --file "/home/srinivasa/Downloads/PegaPE_851_DB_Postgres.sql"
```

6. Keep the backed-up copy handy for future steps.

To restore the backup copy into fresh postgres image, below/similar commands are used. (These commands are useful in future steps too)

pg_restore if binary backup taken

```
/usr/bin/pg_restore --host "192.168.2.71" --port "5432" --username "postgres" --no-password --dbname "postgres" --verbose "/share/samba/PegaPE_851_DB_Postgres.sql"
```

psql if sql/text backup taken

```
/usr/bin/psql --host "192.168.2.111" --port "5432" --username "postgres" --password --dbname "postgres" --file "/home/srinivasa/Downloads/PegaPE_851_DB_Postgres.sql"
```

1.1 Extract/copy prweb war file from PE folder

prweb.war file is required to deploy it in tomcat container. **Copy this file and keep it handy** for further steps. Please see the relevant location in below screenshot.

| Documents | PRPCPersonalEdition > tomcat > webapps | | | |
|--------------|--|-------------|-----------|--|
| Name | Date modified | Type | Size | |
| docs | 15/04/2021 15:19 | File folder | | |
| examples | 15/04/2021 15:20 | File folder | | |
| host-manager | 15/04/2021 15:20 | File folder | | |
| manager | 15/04/2021 15:20 | File folder | | |
| prhelp | 15/04/2021 15:23 | File folder | | |
| prweb | 15/04/2021 15:23 | File folder | | |
| ROOT | 15/04/2021 15:19 | File folder | | |
| prhelp.war | 24/02/2020 02:16 | WAR File | 80,721 KB | |
| prweb.war | 24/02/2020 02:15 | WAR File | 389 KB | |

2. Build Docker Image(s)

To install Pega, it is required to run a Database container and an App server container. Key details are as below.

1. A postgres container will be configured first and a backup DB file will be dumped into it to call it Pega.
2. A tomcat container will be configured next and linked to postgres container to read from Database.
3. This tomcat container will act as app/web server.
4. All necessary files and dependencies are made available at respective GitHub repositories.

<https://github.com/sreesoft/pegapostgres>

<https://github.com/sreesoft/pegatomcat>

Follow step-by-step guide to clone these repositories and build your own local copy.

1. First, choose a base folder to keep all files cloned from GitHub. (Example folder is visible in the below screenshot. 'Test')
2. Use Terminal/Command Line/Power shell (based on operating system) to go to this folder.
3. Once in this folder, use git clone command to clone 'pegapostgres' repo as shown in below screenshot.

```
git clone git@github.com:sreesoft/pegapostgres.git
```

(OR, if line endings are a problem in Windows, use the below command)

```
git clone -c core.autocrlf=false git@github.com:sreesoft/pegapostgres.git
```

```
[Srinivasas-MacBook-Pro:Test Srinivasa$ pwd
/Users/Srinivasa/OneDrive/MyWork/Docker/Test
[Srinivasas-MacBook-Pro:Test Srinivasa$ git clone git@github.com:sreesoft/pegapostgres.git
Cloning into 'pegapostgres'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 11 (delta 0), reused 11 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), 1.67 MiB | 2.93 MiB/s, done.
[Srinivasas-MacBook-Pro:Test Srinivasa$ ll
total 0
drwxr-xr-x 10 Srinivasa staff 320 27 Aug 16:27 pegapostgres
[Srinivasas-MacBook-Pro:Test Srinivasa$
```

1. Now clone other repository 'pegatomcat' as shown below.

```
git clone git@github.com:sreesoft/pegatomcat.git
```

(OR, if line endings are a problem in Windows, use the below command)
 git clone -c core.autocrlf=false git@github.com:sreesoft/pegatomcat.git

```
[Srinivasas-MacBook-Pro:Test Srinivasa$ ll
total 0
drwxr-xr-x@ 11 Srinivasa staff 352 27 Aug 17:12 pegapostgres
[Srinivasas-MacBook-Pro:Test Srinivasa$ git clone git@github.com:sreesoft/pegatomcat.git
Cloning into 'pegatomcat'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 0), reused 9 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), 1.20 MiB | 1.65 MiB/s, done.
[Srinivasas-MacBook-Pro:Test Srinivasa$ ll
total 0
drwxr-xr-x@ 11 Srinivasa staff 352 27 Aug 17:12 pegapostgres
drwxr-xr-x 10 Srinivasa staff 320 27 Aug 17:40 pegatomcat
[Srinivasas-MacBook-Pro:Test Srinivasa$
```

2. There should be 2 folders created after these clone commands executed successfully.
3. Go into 'pegatomcat' folder and copy 'prweb.war' file extracted from PE to this location.
4. Go into 'pegapostgres' folder and inspect the contents.

```
[Srinivasas-MacBook-Pro:Test Srinivasa$ cd pegapostgres/
[Srinivasas-MacBook-Pro:pegapostgres Srinivasa$ ll
total 6824
-rwxr-xr-x@ 1 Srinivasa staff 2774201 27 Aug 16:27 Docker_Pega_Postgres_Image_Build.docx
-rwxr-xr-x@ 1 Srinivasa staff 944 27 Aug 16:27 Dockerfile
-rw-r--r--@ 1 Srinivasa staff 15 27 Aug 16:27 README.md
-rwxr-xr-x@ 1 Srinivasa staff 4574 27 Aug 16:27 pg_hba.conf
-rwxr-xr-x@ 1 Srinivasa staff 340436 27 Aug 16:27 pljava.jar
-rwxr-xr-x@ 1 Srinivasa staff 329672 27 Aug 16:27 pljava.so
-rwxr-xr-x@ 1 Srinivasa staff 24184 27 Aug 16:27 postgresql.conf
[Srinivasas-MacBook-Pro:pegapostgres Srinivasa$
```

5. Check if you have any docker images existing. Please verify if same named image exists. If exists, be mindful this build will overwrite it. To check existing docker images, use this command.

docker images

```
[Srinivasas-MacBook-Pro:pegapostgres Srinivasa$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
Srinivasas-MacBook-Pro:pegapostgres Srinivasa$
```

6. Now, build this image as shown below. It may take few minutes based on internet bandwidth.

```
docker build -t pegapostgres .
```



```

Srinivasas-MacBook-Pro:pegapostgres Srinivasa$ ll
total 6824
-rwxr-xr-x@ 1 Srinivasa  staff  2774201 27 Aug 16:27 Docker_Pega_Postgres_Image_Build.docx
-rwxr-xr-x@ 1 Srinivasa  staff    944 27 Aug 16:27 Dockerfile
-rw-r--r--@ 1 Srinivasa  staff    15 27 Aug 16:27 README.md
-rwxr-xr-x@ 1 Srinivasa  staff   4574 27 Aug 16:27 pg_hba.conf
-rwxr-xr-x@ 1 Srinivasa  staff  340436 27 Aug 16:27 pljava.jar
-rwxr-xr-x@ 1 Srinivasa  staff  329672 27 Aug 16:27 pljava.so
-rwxr-xr-x@ 1 Srinivasa  staff   24184 27 Aug 16:27 postgresql.conf
Srinivasas-MacBook-Pro:pegapostgres Srinivasa$ docker build -t pegapostgres .
update-alternatives: warning: skip creation of /usr/share/man/it/man1/editor.1.gz because associated file /usr/share/man/it/man1/editor.1.gz (not found)
update-alternatives: warning: skip creation of /usr/share/man/pl/man1/editor.1.gz because associated file /usr/share/man/pl/man1/editor.1.gz (not found)
update-alternatives: warning: skip creation of /usr/share/man/ru/man1/editor.1.gz because associated file /usr/share/man/ru/man1/editor.1.gz (not found)
update-alternatives: warning: skip creation of /usr/share/man/ja/man1/editor.1.gz because associated file /usr/share/man/ja/man1/editor.1.gz (not found)
update-alternatives: warning: skip creation of /usr/share/man/man1/editor.1.gz because associated file /usr/share/man/man1/editor.1.gz (not found)
Removing intermediate container 1b946a7d78e5
--> 86135bd903cf
Step 6/11 : RUN ln -s /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server/libjvm.so /lib/libjvm.so
--> Running in 509bad5ea790
Removing intermediate container 509bad5ea790
--> 4d8b0b129be6
Step 7/11 : COPY ./pljava.so /usr/lib/postgresql/11/lib
--> 7ff905518243
Step 8/11 : COPY ./pljava.jar /usr/lib/postgresql/11/lib
--> fb2aa51539e0
Step 9/11 : RUN mkdir -p /var/lib/postgresql-static/data
--> Running in d002fc903041
Removing intermediate container d002fc903041
--> 24ff2119c648
Step 10/11 : RUN chown -R postgres:postgres /var/lib/postgresql-static
--> Running in 5b1ccf51d56a
Removing intermediate container 5b1ccf51d56a
--> c353372c978c
Step 11/11 : ENV PGDATA /var/lib/postgresql-static/data
--> Running in b73a8a4c667b
Removing intermediate container b73a8a4c667b
--> f936642c5b7e
Successfully built f936642c5b7e
Successfully tagged pegapostgres:latest
Srinivasas-MacBook-Pro:pegapostgres Srinivasa$

```

- Once successfully built the image, run it using the below command. Note the database password is used as 'postgres'.

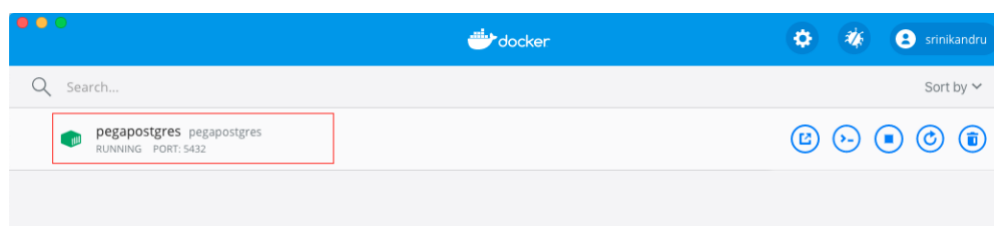
```
docker run --name pegapostgres -e POSTGRES_PASSWORD=postgres -p 5432:5432 -d pegapostgres
```

```

postgres 11.8 31022a1b540T 8 minutes ago 242MB
Srinivasas-MBP:pegapostgres Srinivasa$ docker run --name pegapostgres -e POSTGRES_PASSWORD=postgres -p 5432:5432 -d pegapostgres
7ba9f64186defc69694555542a7ba5b4ec95a2d95162b6dfe3f2bd16e2bbd4e2
Srinivasas-MBP:pegapostgres Srinivasa$ ll
total 6824
-rwxr-xr-x@ 1 Srinivasa  staff  2774201 27 Aug 16:27 Docker_Pega_Postgres_Image_Build.docx
-rwxr-xr-x@ 1 Srinivasa  staff    944 27 Aug 16:27 Dockerfile
-rw-r--r--@ 1 Srinivasa  staff    15 27 Aug 16:27 README.md
-rwxr-xr-x@ 1 Srinivasa  staff   4574 27 Aug 16:27 pg_hba.conf
-rwxr-xr-x@ 1 Srinivasa  staff  340436 27 Aug 16:27 pljava.jar
-rwxr-xr-x@ 1 Srinivasa  staff  329672 27 Aug 16:27 pljava.so
-rwxr-xr-x@ 1 Srinivasa  staff   24184 27 Aug 16:27 postgresql.conf
Srinivasas-MacBook-Pro:pegapostgres Srinivasa$

```

- On success, docker dashboard will show running container as below.



3. Configure Pega Postgres image

Postgres image maintains its configuration files & data at a default volume (think as volume mount) which may not be persisted upon committing docker containers.

It is required to update few configurations to make it work for Pega. All required configuration files are prepared and available in this repository. Follow below steps to amend your local image.

- Copy config files into container as below.

```
docker cp ./pg_hba.conf pegapostgres:/var/lib/postgresql-static/data
docker cp ./postgresql.conf pegapostgres:/var/lib/postgresql-static/data
```

```
/ba9f64186de pegapostgres "docker-entrypoint.s..." About a minute ago Up About a minute 0.0.0.0:5432->5432/
Srinivasas-MBP:pegapostgres Srinivasas$ docker cp ./pg_hba.conf pegapostgres:/var/lib/postgresql-static/data
Srinivasas-MBP:pegapostgres Srinivasas$ docker cp ./postgresql.conf pegapostgres:/var/lib/postgresql-static/data
Srinivasas-MBP:pegapostgres Srinivasas$
```

- This docker container does not have any Pega database installed yet. We are going to utilise the backed-up copy made earlier in this document.

Please note, PegaPE_851_DB_Postgres is the filename of DB backup extracted earlier. It is only working name for it, different name can be used.

There are 2 ways to export the DB Backup into postgres docker container. First one is to use 'psql' utility provided as part of PGAdmin4 tool. Second one is to upload this backup file into Docker container and install. Both options are documented below. Use according to the convenience.

3.1 Export Backup file via host 'psql' utility.

Find 'psql' utility in host machine. If its Mac, it usually available in the shown path in below image.

Once found, run the below command as shown below. File paths may vary.

```
./psql --host "localhost" --port "5432" --username "postgres" --password --dbname "postgres" --file
"/Users/Srinivasa/Downloads/PegaPE_851_DB_Postgres.sql"
```

```
Srinivas-MBP:SharedSupport Srinivas$ pwd
/Applications/pgAdmin 4.app/Contents/SharedSupport
Srinivas-MBP:SharedSupport Srinivas$ ll
total 3384
-rwxr-xr-x@ 1 Srinivasa admin 551456 23 Mar 14:01 pg_dump
-rwxr-xr-x@ 1 Srinivasa admin 147408 23 Mar 14:01 pg_dumpall
-rwxr-xr-x@ 1 Srinivasa admin 259312 23 Mar 14:01 pg_restore
-rwxr-xr-x@ 1 Srinivasa admin 768224 23 Mar 14:01 psql
Srinivas-MBP:SharedSupport Srinivas$
Srinivas-MBP:SharedSupport Srinivas$
Srinivas-MBP:SharedSupport Srinivas$ /psql --host "localhost" --port "5432" --username "postgres" --password --dbname "postgres" --file "/Users/Srinivasa/Downloads/PegaPE_851_DB_Postgres.sql"
Password: 
```

Password to use here is 'postgres'

Once password is accepted, it will start loading all tables and objects into Postgres container. It may take a while depends on target system capacity. Once completed, it will show similar as below.

```
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
CREATE INDEX  
ALTER TABLE  
ALTER TABLE  
ALTER TABLE  
GRANT  
GRANT  
GRANT  
GRANT  
GRANT  
Srinivasas-MBP:SharedSupport Srinivasas
```

If no errors are presented, that's it. DB is successfully loaded. Then skip section 3.2 and go to 3.3.

3.2 Upload backup file into container and load

When 'psql' utility not available in host machine or not interested to run from host, these steps to be followed.

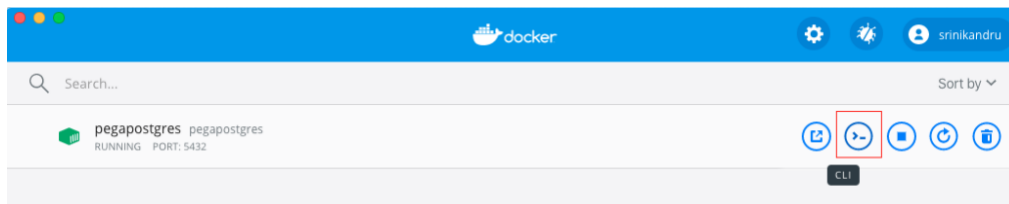
First load the DB backup file into containers temp location as shown below.

```
docker cp ./PegaPE_851_DB_Postgres.sql pegapostgres:/tmp
```

```
Srinivasas-MacBook-Pro:pegapostgres Srinivasas$ docker cp ./PegaPE_840_DB_Postgres.sql pegapostgres:/tmp
```

Now, go to docker dashboard and open CLI (Command Line Interface) utility to run below commands.





```
sudo bash
sudo -su postgres
psql -d postgres -f /tmp/PegaPE_851_DB_Postgres.sql
```

```
Srinivasa — docker exec -it fb8b30be3d30622fe5a4f6d20c396901e2525c25099c54976c8528132aaa1747 /bin/sh — 120x31
Last login: Thu Aug 27 15:34:47 on ttys002

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Srinivasas-MacBook-Pro:~ Srinivasa$ docker exec -it fb8b30be3d30622fe5a4f6d20c396901e2525c25099c54976c8528132aaa1747 /bin/sh; exit
# sudo bash
root@fb8b30be3d30:/# sudo -su postgres
postgres@fb8b30be3d30:/# psql -d postgres -f /tmp/PegaPE_840_DB_Postgres.sql
```

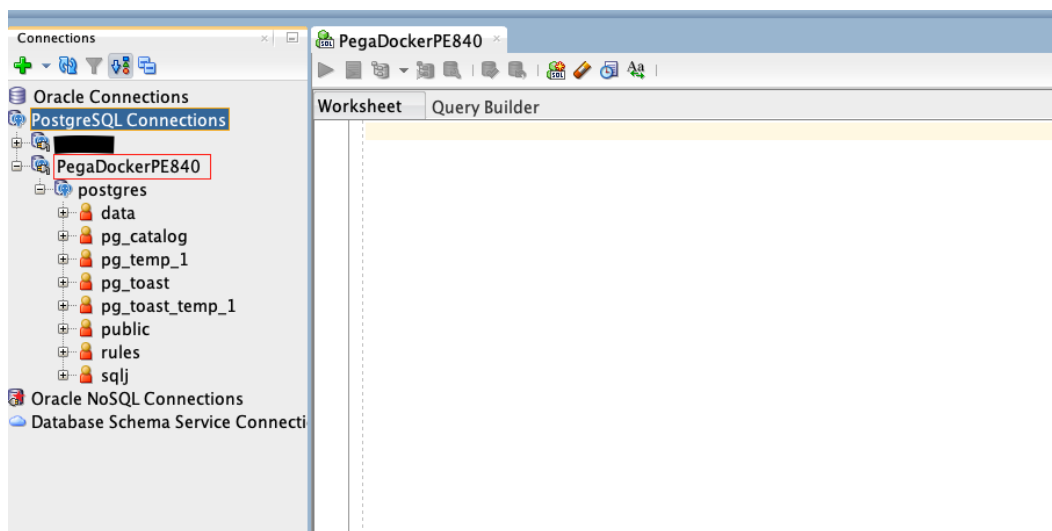
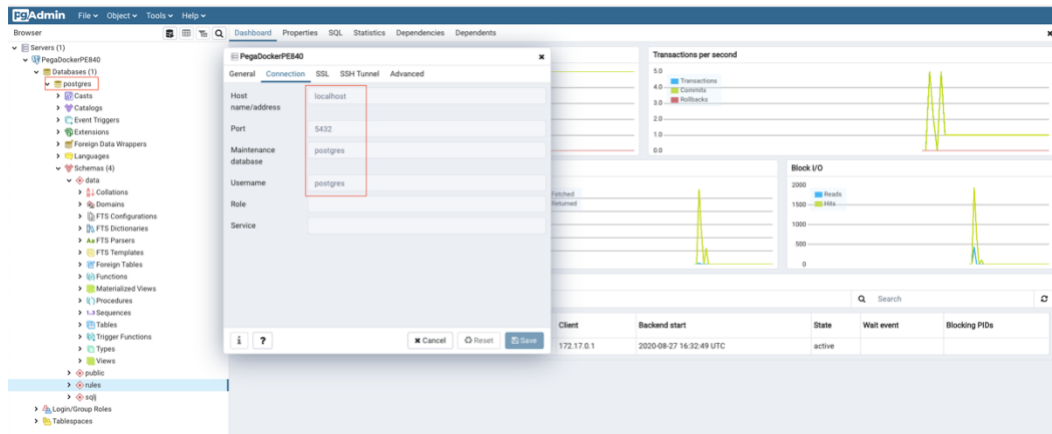
```
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
postgres@fb8b30be3d30:/#
```

3.3 Test Postgres Connection

When backup file loaded successfully, It is time to test Postgres connection via PGAdmin4 or SQL Developer

Use connection details as shown in screenshot.

Use password as 'postgres'.



Connection Name: PegaDockerPE840

Username: postgres

Password:

☒ Save Password ☒ Connection Color

Oracle ☐ PostgreSQL ☒

Hostname: localhost

Port: 5432

Choose Database: postgres

Clear Test Connect Cancel

4. Configure Pega Tomcat image

Go to 'pegatomcat' folder created earlier and build image as shown below.

`docker build -t pegatomcat .`

```
[Srinivasas-MacBook-Pro:Test Srinivasa$ cd pegatomcat/
[Srinivasas-MacBook-Pro:pegatomcat Srinivasa$ ll
total 2640
-rwxr-xr-x  1 Srinivasa  staff   1046 27 Aug 17:40 Dockerfile
-rw-r--r--  1 Srinivasa  staff    13 27 Aug 17:40 README.md
-rwxr-xr-x  1 Srinivasa  staff   2587 27 Aug 17:40 context.xml
-rwxr-xr-x  1 Srinivasa  staff  932808 27 Aug 17:40 postgresql-42.2.14.jar
-rwxr-xr-x  1 Srinivasa  staff   397135 27 Aug 17:40 prweb.war
-rwxr-xr-x  1 Srinivasa  staff    313 27 Aug 17:40 setenv.sh
-rwxr-xr-x  1 Srinivasa  staff   2320 27 Aug 17:40 tomcat-users.xml
[Srinivasas-MacBook-Pro:pegatomcat Srinivasa$ docker build -t pegatomcat .
```

```
Removing intermediate container e4c815d1bae8
----> c18e1a104765
Step 11/22 : EXPOSE 8080
----> Running in 63a505f80f26
Removing intermediate container 63a505f80f26
----> 3e18b825e60b
Step 12/22 : COPY ./postgresql-42.2.14.jar /opt/tomcat/lib
----> 70b6c5418432
Step 13/22 : RUN mkdir /pega
----> Running in 2384a1d2a252
Removing intermediate container 2384a1d2a252
----> 88536ae70d7f
Step 14/22 : RUN mkdir /pega/logs
----> Running in 1ca8f21d9f48
Removing intermediate container 1ca8f21d9f48
----> 0798bed97220
Step 15/22 : RUN mkdir /pega/index
----> Running in 910d106496ab
Removing intermediate container 910d106496ab
----> 088949b37775
Step 16/22 : RUN mkdir /pega/temp
----> Running in 193d8f5e3153
Removing intermediate container 193d8f5e3153
----> 0d56c0c5e013
Step 17/22 : RUN mkdir /pega/cassandra_data
----> Running in 703718d7ac51
Removing intermediate container 703718d7ac51
----> 31c463a7d2e3
Step 18/22 : COPY ./context.xml /opt/tomcat/conf
----> 3ea3ffea53c3
Step 19/22 : COPY ./tomcat-users.xml /opt/tomcat/conf
----> fb9709fa38ac
Step 20/22 : COPY ./setenv.sh /opt/tomcat/bin
----> e2ffc695a6ef
Step 21/22 : COPY ./prweb.war /opt/tomcat/webapps
----> d6835841fd44
Step 22/22 : CMD /opt/tomcat/bin/catalina.sh run
----> Running in 44f092e2289c
Removing intermediate container 44f092e2289c
----> cf312b984f93
Successfully built cf312b984f93
Successfully tagged pegatomcat:latest
[Srinivasas-MacBook-Pro:pegatomcat Srinivasa$
```

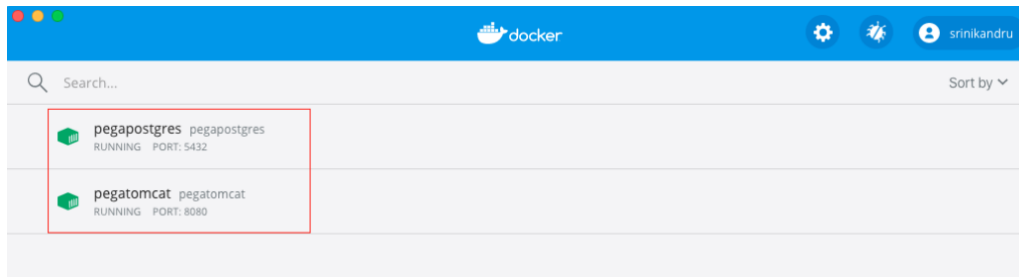
5. Running Pega Instance(s)

1. Once pegatomcat image is build, it is required to run this image by linking pegapostgres container. Use the below command to run it.

docker run --name pegatomcat -d -p 8080:8080 --link pegapostgres:pegapostgres pegatomcat

```
Successfully tagged pegatomcat:latest
Srinivasas-MacBook-Pro:pegatomcat Srinivasas$ docker run --name pegatomcat -d -p 8080:8080 --link pegapostgres:pegapostgres pegatomcat
53da18961427e3f583220bd421d2f9e048d63a854691f33aba86cb027b2f499d
Srinivasas-MacBook-Pro:pegatomcat Srinivasas$
```

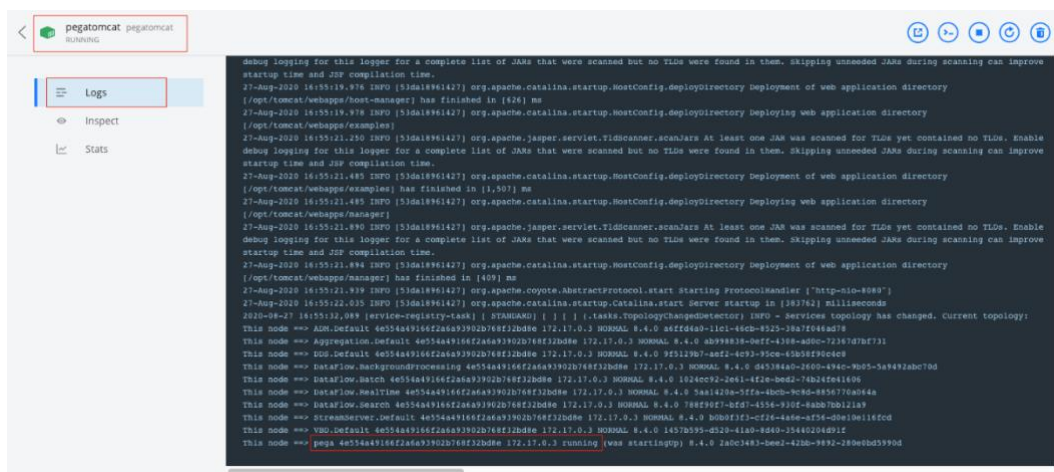
2. It should have below 2 running instances as shown.



If using Linux, run the below command to see docker running instances.

```
docker ps
skandru@SrinivasUbuntu:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
65245723b156   pegatomcat    "/bin/sh -c '/usr/lo..." 3 days ago    Up 3 days    0.0.0.0:8080->8080/tcp             pegatomcat
464d1be09ea0   pegapostgres  "docker-entrypoint.s..." 4 days ago    Up 4 days    0.0.0.0:5432->5432/tcp             pegapostgres
skandru@SrinivasUbuntu:~$
```

3. Check its logs and observe the highlighted log message to know pega is up & running. It may take few minutes to start up this container.

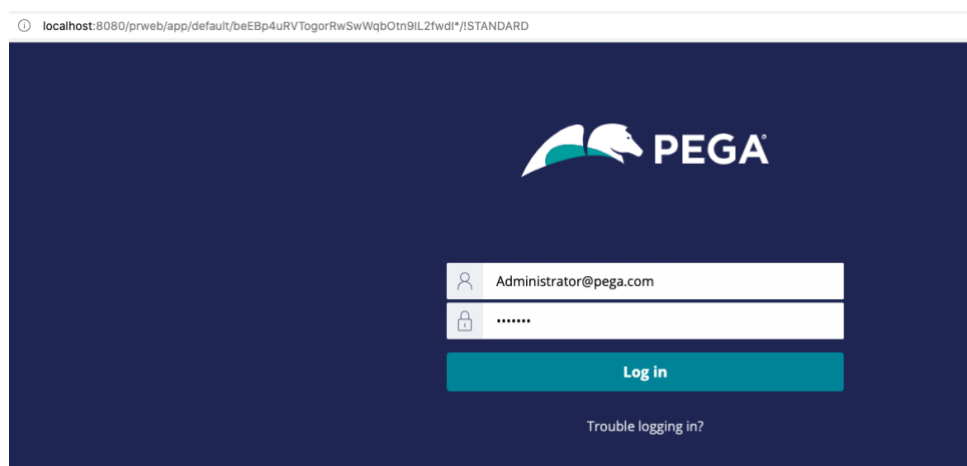


4. To open terminal of docker container from command line, use the below and check logs & other configs.

```
docker exec -it pegatomcat /bin/bash
```

```
[skandru@SrinivasaUbuntu:~]$ docker exec -it pegatomcat /bin/bash
[root@65245723b156:/usr/local/tomcat# ll
total 184
-rw-r--r-- 1 root root 6898 Dec 3 11:48 RELEASE-NOTES
-rw-r--r-- 1 root root 3257 Dec 3 11:48 README.md
-rw-r--r-- 1 root root 2333 Dec 3 11:48 NOTICE
-rw-r--r-- 1 root root 57092 Dec 3 11:48 LICENSE
-rw-r--r-- 1 root root 5409 Dec 3 11:48 CONTRIBUTING.md
-rw-r--r-- 1 root root 18982 Dec 3 11:48 BUILDING.txt
-rw-r--r-- 1 root root 16507 Dec 3 11:48 RUNNING.txt
drwxr-xr-x 2 root root 4096 Dec 18 08:59 native-jni-lib
drwxr-xr-x 1 root root 4096 Dec 23 20:13 lib
drwxr-xr-x 1 root root 4096 Dec 23 20:13 bin
drwxr-xr-x 1 root root 4096 Dec 23 20:13 webapps.dist
drwxr-xr-x 1 root root 4096 Dec 23 20:14 conf
drwxr-xr-x 1 root root 4096 Dec 23 20:14 webapps
drwxrwxrwx 1 root root 4096 Dec 23 20:14 work
drwxr-xr-x 7 root root 4096 Dec 23 20:14 kafka-1.1.0.4
drwxr-xr-x 10 root root 4096 Dec 23 20:15 cassandra
drwxrwxrwx 1 root root 4096 Dec 23 20:15 temp
drwxrwxrwx 1 root root 4096 Dec 27 07:57 logs
drwxr-xr-x 387 root root 20480 Dec 27 19:04 kafka-data
[root@65245723b156:/usr/local/tomcat# pwd
/usr/local/tomcat
root@65245723b156:/usr/local/tomcat# █
```

5. Access Pega using <http://localhost:8080/prweb>



Administrator@pega.com / install (or password@1)

6. All required folders for logs, temp... etc are available as below.



```

pegatomcat pegatomcat
RUNNING PORT: 8080

root@53da18961427:/# cd /pega/
root@53da18961427:/pega# ll
total 28
drwxr-xr-x 1 root root 4096 Aug 27 16:46 ./
drwxr-xr-x 1 root root 4096 Aug 27 16:48 ../
drwxr-xr-x 2 root root 4096 Aug 27 16:46 cassandra_data/
drwxr-xr-x 2 root root 4096 Aug 27 16:46 index/
drwxr-xr-x 1 root root 4096 Aug 27 16:49 logs/
drwxr-xr-x 1 root root 4096 Aug 27 16:52 temp/
root@53da18961427:/pega#

```

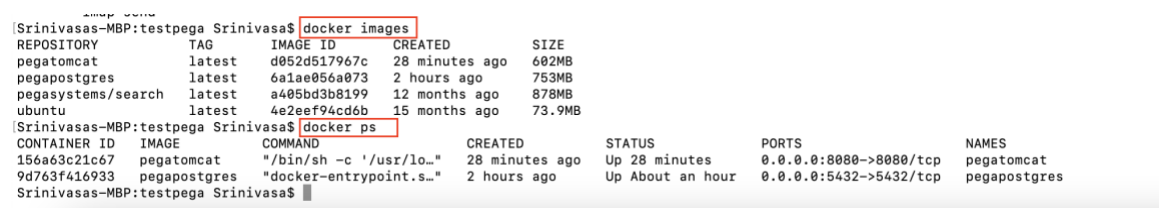
7. Happy adventure... 😊

6. Docker Image Export/Import

To export and import docker image, use below commands. For any advanced options, please refer docker help documents.

To display docker images available...

docker image



```

Srinivasas-MBP:testpega Srinivasas$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
pegatomcat           latest     d052d517967c 28 minutes ago 602MB
pegapostgres         latest     6a1ae056a073 2 hours ago   753MB
pegasystems/search   latest     a405bd3b8199 12 months ago 878MB
ubuntu               latest     4e2eef94cd6b 15 months ago 73.9MB
Srinivasas-MBP:testpega Srinivasas$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS      PORTS                               NAMES
156a63c21c67   pegatomcat "/bin/sh -c '/usr/lo..." 28 minutes ago Up 28 minutes 0.0.0.0:8080->8080/tcp   pegatomcat
9d763f416933   pegapostgres "docker-entrypoint.s..." 2 hours ago Up About an hour 0.0.0.0:5432->5432/tcp   pegapostgres
Srinivasas-MBP:testpega Srinivasas$

```

Before exporting an image, it needs to be committed.

docker commit 9d763f416933 pegapostgres:withpegadb

docker commit 156a63c21c67 pegatomcat:withpega86

```
Srinivasas-MBP:testpega Srinivasa$ docker commit 9d763f416933 pegapostgres:withpegadb
sha256:70a93cd845cae94f30a842823d2f89cde3a78b8b729cacb1feafe90b3a669c38
Srinivasas-MBP:testpega Srinivasa$ docker images
```

| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|--------------------|------------|--------------|---------------|--------|
| pegapostgres | withpegadb | 70a93cd845ca | 2 minutes ago | 7.88GB |
| pegatomcat | latest | d052d517967c | 2 hours ago | 602MB |
| pegapostgres | latest | 6a1ae056a073 | 3 hours ago | 753MB |
| pegasystems/search | latest | a405bd3b8199 | 12 months ago | 878MB |
| ubuntu | latest | 4e2eef94cd6b | 15 months ago | 73.9MB |

```
Srinivasas-MBP:testpega Srinivasa$
```

```
Srinivasas-MBP:testpega Srinivasa$ docker images
```

| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|--------------------|------------|--------------|---------------|--------|
| pegatomcat | withpega86 | ba98f182a35b | 3 minutes ago | 8.65GB |
| pegapostgres | withpegadb | 70a93cd845ca | 8 minutes ago | 7.88GB |
| pegatomcat | latest | d052d517967c | 2 hours ago | 602MB |
| pegapostgres | latest | 6a1ae056a073 | 3 hours ago | 753MB |
| pegasystems/search | latest | a405bd3b8199 | 12 months ago | 878MB |
| ubuntu | latest | 4e2eef94cd6b | 15 months ago | 73.9MB |

```
Srinivasas-MBP:testpega Srinivasa$
```

After commit, save/export the image into a file...

```
docker save -o ~/Downloads/PegaPostgresDB.tar pegapostgres:withpegadb
docker save -o ~/Downloads/PegaTomcat.tar pegatomcat:withpega86
```

Use the exported files for loading at different docker containers.

Use docker load commands to import an image...

```
docker load -i ~/Downloads/PegaPostgres.tar
docker load -i ~/Downloads/PegaTomcat.tar
```

Once loaded images successfully, use below commands to run those images.

```
docker run --name pegapostgres -e POSTGRES_PASSWORD=postgres -p 5432:5432 -d pegapostgres
docker run --name pegatomcatdev -d -p 8080:8080 --link pegapostgres:pegapostgres pegatomcat
```

7. Issues Fixed

1. When Tomcat admin portal is not accessible, it is more likely that privileges/permissions are not set for Admin app. Followed the solutions outlined at below URL to solve it.

[tomcat8 - 403 Access Denied on Tomcat 8 Manager App without prompting for user/password - Stack Overflow](#)

2. After cloning git repository, if the Line ending characters causing any trouble, alternative way of cloning is used. This is specially an issue in Windows OS. Below article is followed to solve this issue.

[git - Trying to clone a repository but the line endings are wrong - Stack Overflow](#)