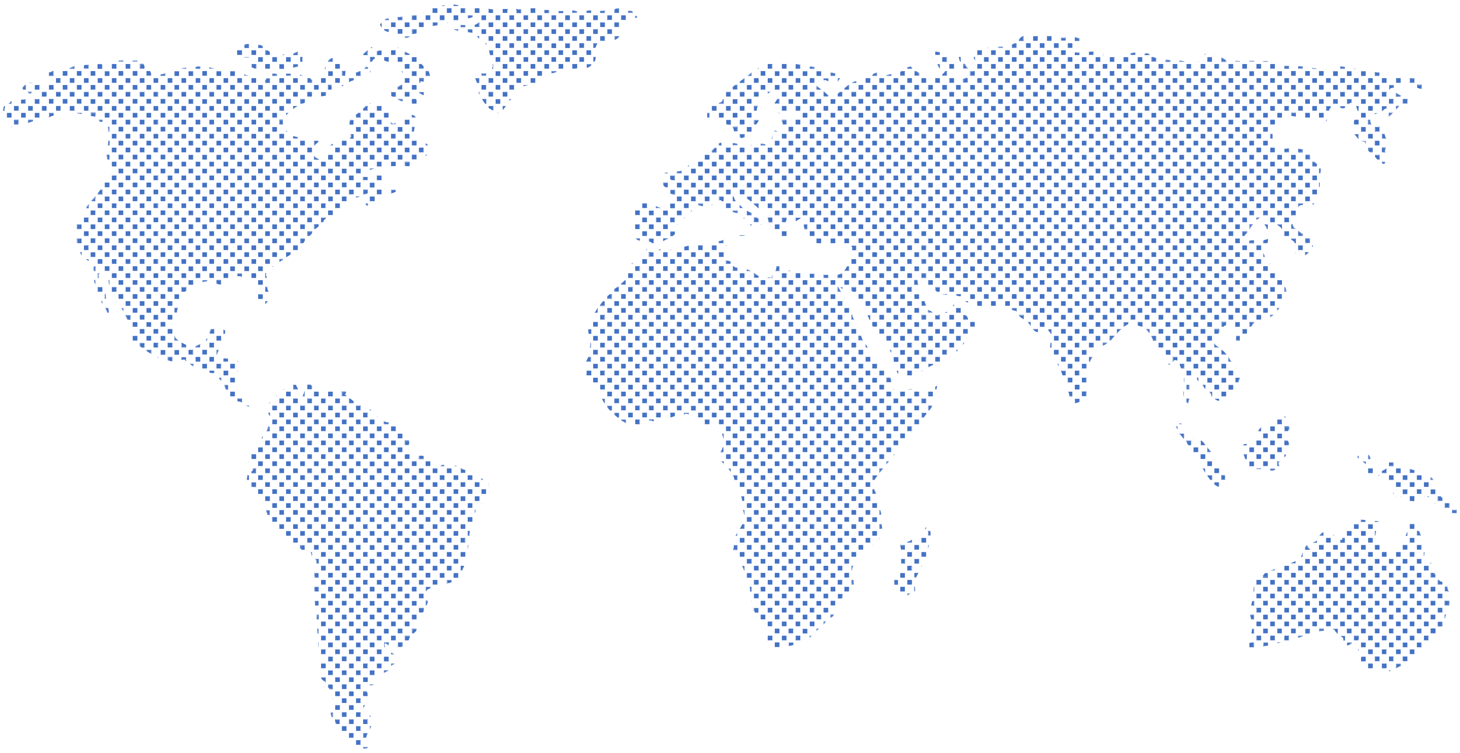
Pega Docker
Postgres/Tomcat Container



SRINIVASA KANDRU

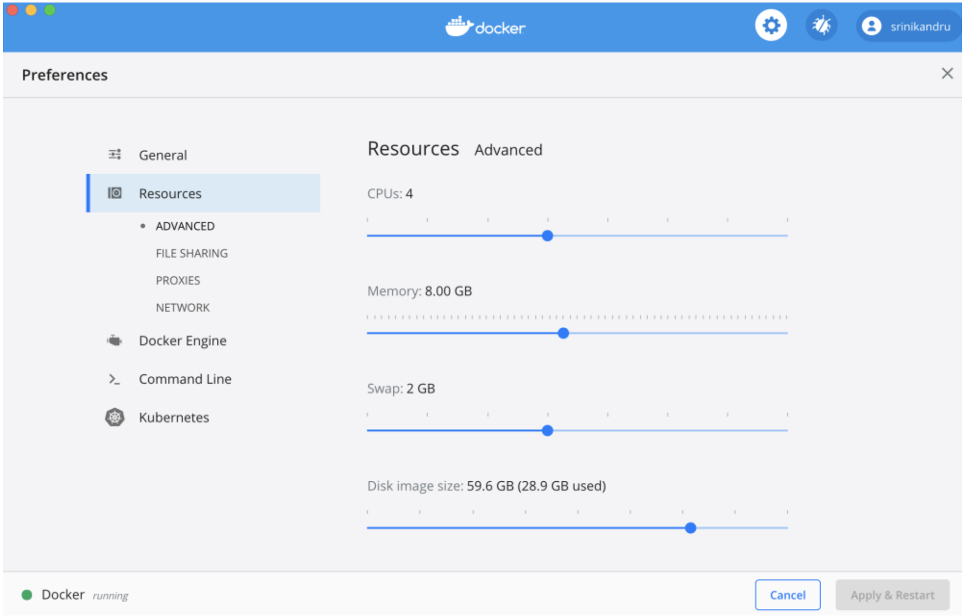
Sreesoft Solutions

Contents

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.



CPUs: 4

Memory: 8.00 GB

Swap: 2 GB

Disk image size: 59.6 GB (28.9 GB used)

Docker *running*

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
5. Install pgAdmin 4 client.

Build Docker Image(s)

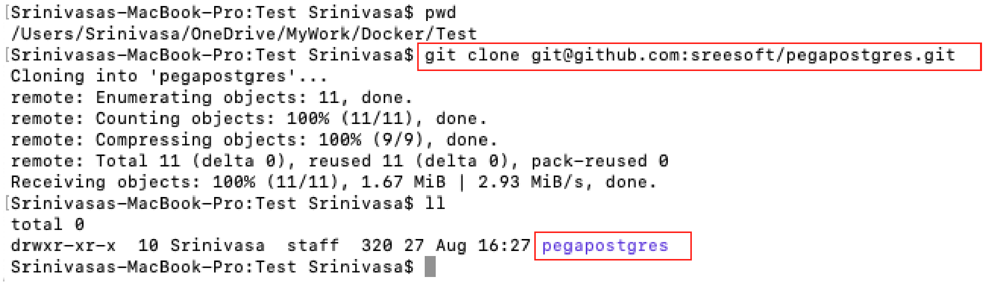
All necessary files and dependencies are made available at below GitHub repository.
https://github.com/sreesoft/pegapostgres.

Follow step-by-step guide to clone this repository 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.

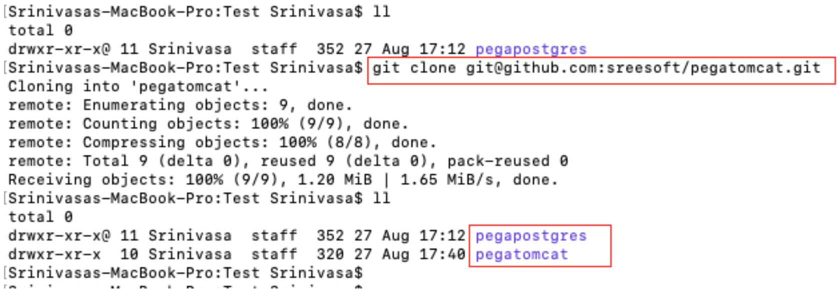
git clone git@github.com:sreesoft/pegapostgres.git



```
[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$
```

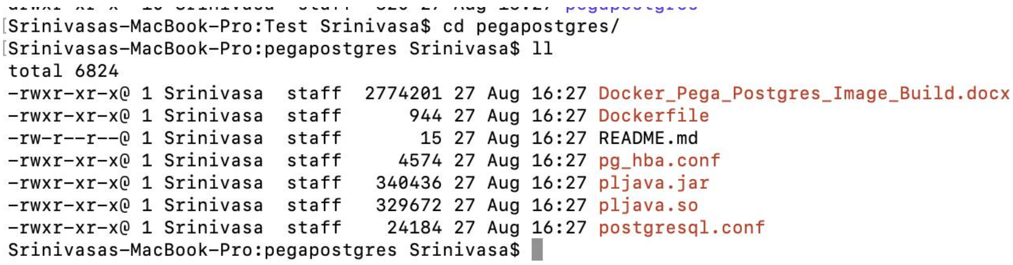
4. Now clone other repository ‘pegatomcat’ as shown below.

git clone 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$
```

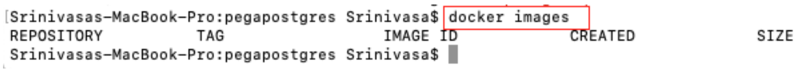
5. There should be 2 folders created after these clone commands executed successfully. Go into ‘pegapostgres’ folder and inspect the contents.



```
[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$
```

6. 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.

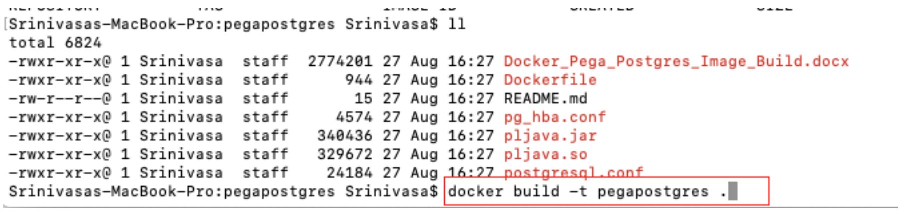
docker images



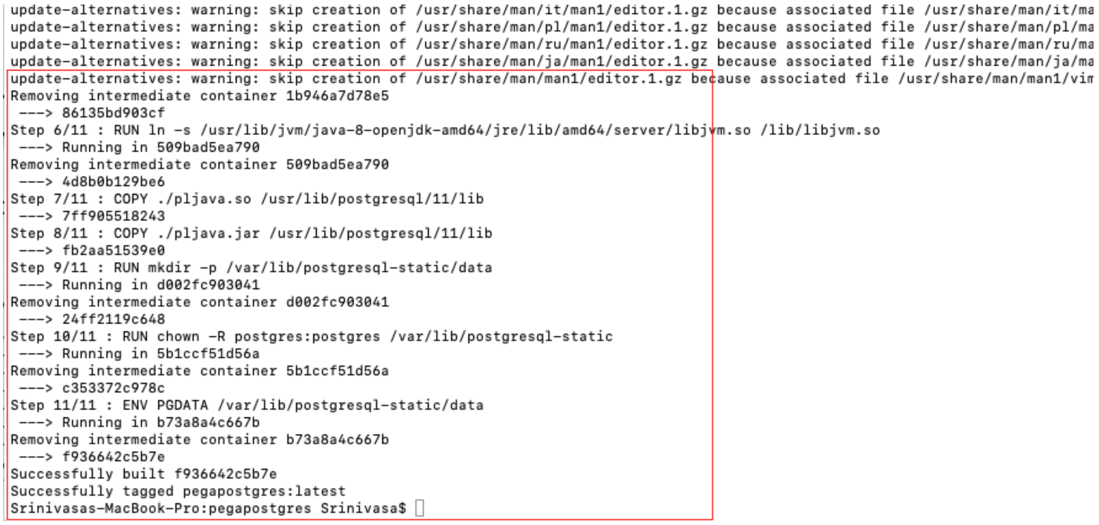
```
[Srinivasas-MacBook-Pro:pegapostgres Srinivasa$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
Srinivasas-MacBook-Pro:pegapostgres Srinivasa$
```

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

docker build -t pegapostgres .



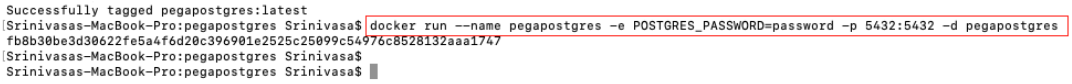
```
Srinivasas-MacBook-Pro:pegapostgres Srinivasa$ ll
total 6824
-rwxr-xr-x@ 1 Srinivasa  staff  2774201 27 Aug 16:27 Docker_Pega_Postgres_Image_Build.docx
-rwxr-xr-x@ 1 Srinivasa  staff      944 27 Aug 16:27 Dockerfile
-rw-r--r--@ 1 Srinivasa  staff       15 27 Aug 16:27 README.md
-rwxr-xr-x@ 1 Srinivasa  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/ma
update-alternatives: warning: skip creation of /usr/share/man/pl/man1/editor.1.gz because associated file /usr/share/man/pl/ma
update-alternatives: warning: skip creation of /usr/share/man/ru/man1/editor.1.gz because associated file /usr/share/man/ru/ma
update-alternatives: warning: skip creation of /usr/share/man/ja/man1/editor.1.gz because associated file /usr/share/man/ja/ma
update-alternatives: warning: skip creation of /usr/share/man/man1/editor.1.gz because associated file /usr/share/man/man1/vim
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$ 
```

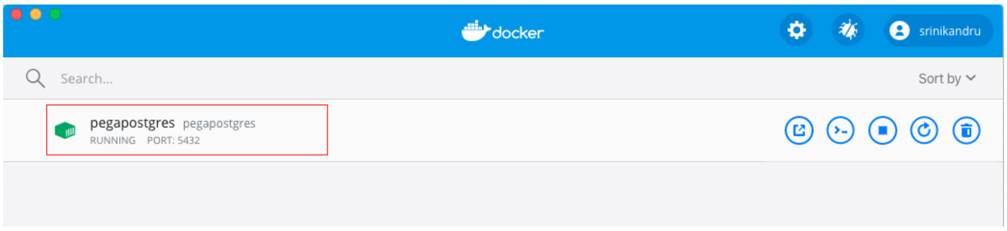
8. Once successfully built the image, run it using the below command

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



```
Successfully tagged pegapostgres:latest
Srinivasas-MacBook-Pro:pegapostgres Srinivasa$ docker run --name pegapostgres -e POSTGRES_PASSWORD=password -p 5432:5432 -d pegapostgres
fb8b30be3d30622fe5a4f6d20c396901e2525c25099c54976c8528132aaa1747
Srinivasas-MacBook-Pro:pegapostgres Srinivasa$
Srinivasas-MacBook-Pro:pegapostgres Srinivasa$ 
```

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



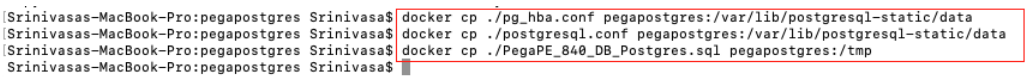
Configure Pega Postgres image

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 and prepared and available in this repository. Follow below steps to amend your local image.

1. This docker container does not have any Pega database installed yet. For this purpose, Postgres DB is backed-up & extracted from Pega 8.4.0 personal edition. This extracted copy is not available in GitHub repository as size is too big for GitHub. (More details on how to extract this file detailed later in this document.) (Alternatively, request this backup file)

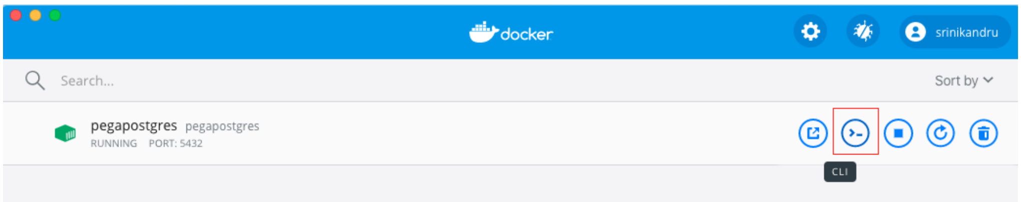
2. Copy config files into container using below. Please note, PegaPE_840_DB_Postgres is the DB backup file & it is only working name for it.

```
docker cp ./pg_hba.conf pegapostgres:/var/lib/postgresql-static/data
docker cp ./postgresql.conf pegapostgres:/var/lib/postgresql-static/data
docker cp ./PegaPE_840_DB_Postgres.sql pegapostgres:/tmp
```

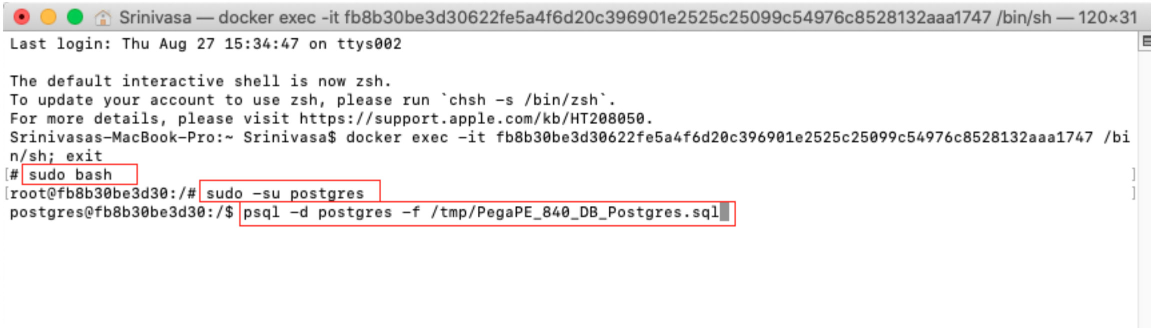


```
Srinivasas-MacBook-Pro:pegapostgres Srinivasa$ docker cp ./pg_hba.conf pegapostgres:/var/lib/postgresql-static/data
Srinivasas-MacBook-Pro:pegapostgres Srinivasa$ docker cp ./postgresql.conf pegapostgres:/var/lib/postgresql-static/data
Srinivasas-MacBook-Pro:pegapostgres Srinivasa$ docker cp ./PegaPE_840_DB_Postgres.sql pegapostgres:/tmp
Srinivasas-MacBook-Pro:pegapostgres Srinivasa$ 
```

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



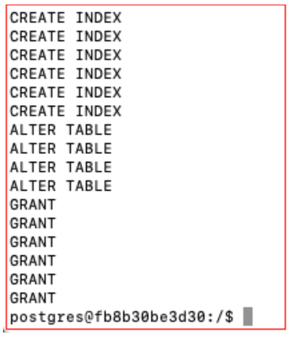
```
sudo bash
sudo -su postgres
psql -d postgres -f /tmp/PegaPE_840_DB_Postgres.sql
```



Srinivasa — docker exec -it fb8b30be3d30622fe5a4f6d20c396901e2525c25099c54976c8528132aaa1747 /bin/sh — 120×31

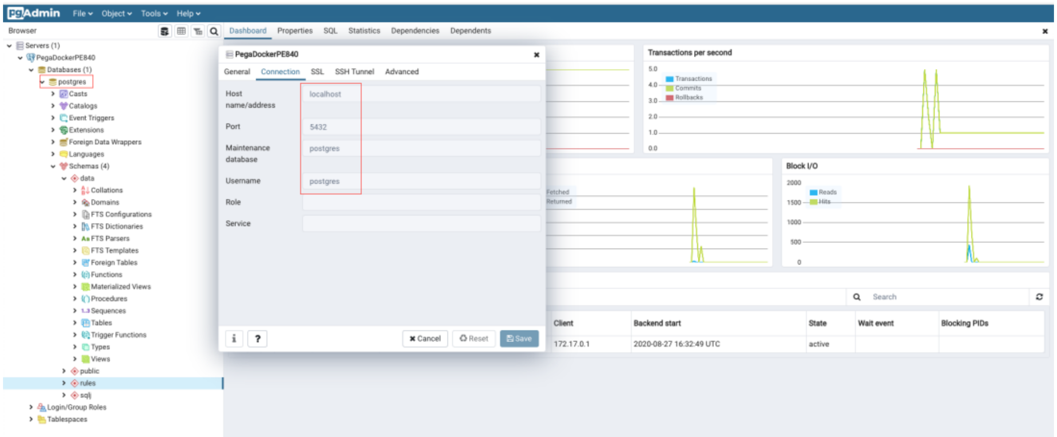
```
Last login: Thu Aug 27 15:34:47 on ttys002

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 /bi
n/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:/$ 
```

4. Once, completed, Test Postgres connection via PGAdmin4 or SQL Developer
   Use connection details as shown in screenshot.



PegaDockerPE840

Host name/address: localhost

Port: 5432

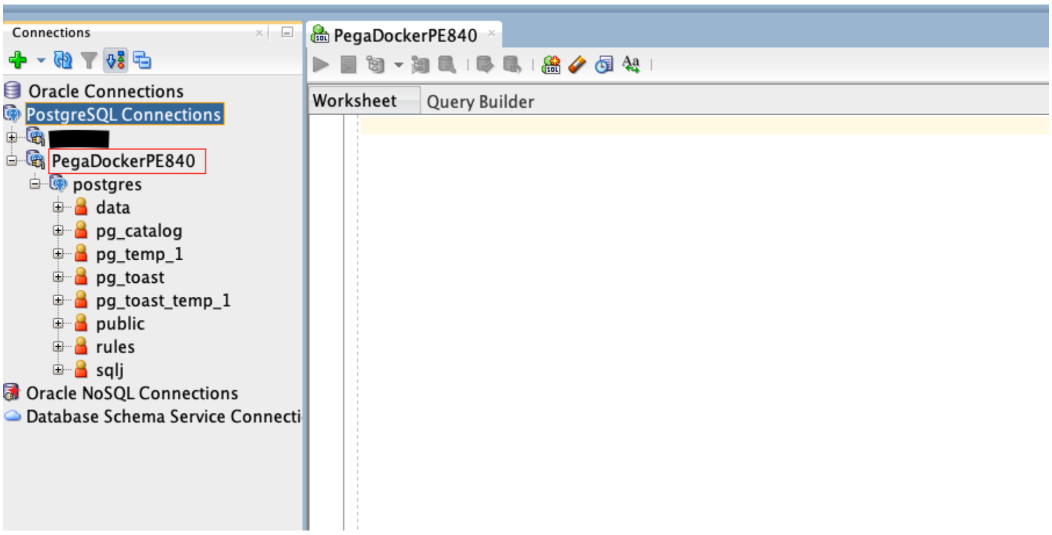
Maintenance database: postgres

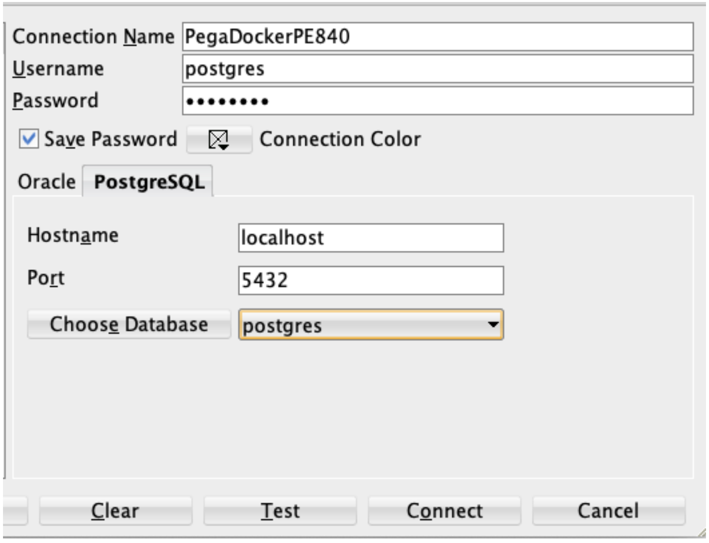
Username: postgres

Role:

Service:

Client	Backend start	State	Wait event	Blocking PIDs
172.17.0.1	2020-08-27 16:32:49 UTC	active		





Connection Name: PegaDockerPE840

Username: postgres

Password: ••••••••

☑ Save Password

Connection Color

Oracle | PostgreSQL

Hostname: localhost

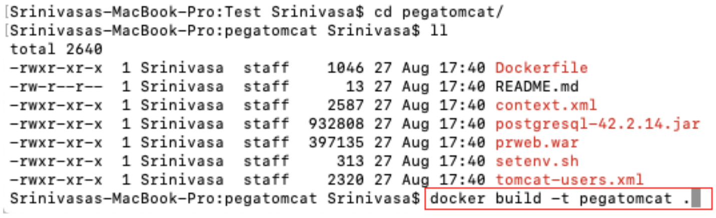
Port: 5432

Choose Database: postgres

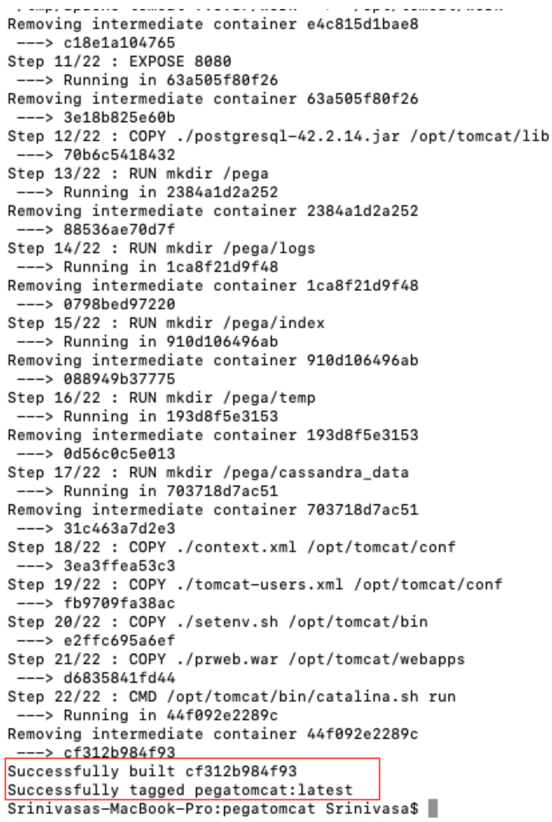
Configure Pega Tomcat image

1. Go to 'pegatomcat' folder created when cloned GitHub 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 .
```

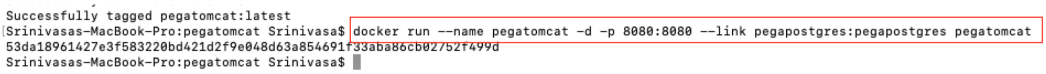


```
Removing intermediate container e4c815d1bae8
 ---> c18e1a104765
Step 11/22 : EXPOSE 8080
 ---> Running in 63a505f80f26
Removing intermediate container 63a505f80f26
 ---> 3e18b825e60b
Step 12/22 : COPY ./postgresql-42.2.14.jar /opt/tomcat/lib
 ---> 70b6c5418432
Step 13/22 : RUN mkdir /pega
 ---> Running 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$
```

Running Pega Instance(s)

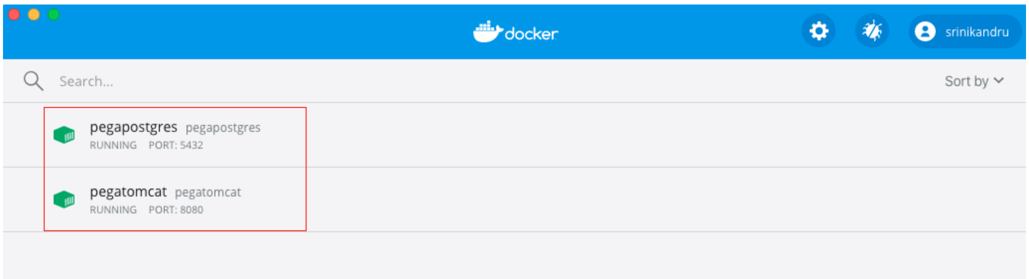
1. Once pegatomcat image is build, it is required to run this image by linking pegapostgres container. Use the below command to run it.

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



```
Successfully tagged pegatomcat:latest
[Srinivasas-MacBook-Pro:pegatomcat Srinivasa$ docker run --name pegatomcat -d -p 8080:8080 --link pegapostgres:pegapostgres pegatomcat
53da18961427e3f583220bd421d2f9e048d63a854691f33aba86cb02752f499d
Srinivasas-MacBook-Pro:pegatomcat Srinivasa$
```

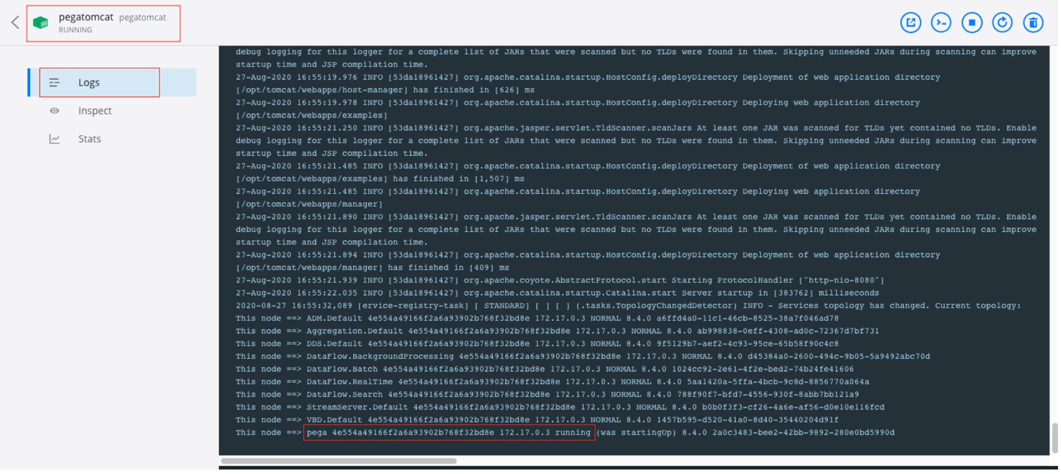
2. It should have 2 running instances as shown below.



pegapostgres pegapostgres
RUNNING PORT: 5432

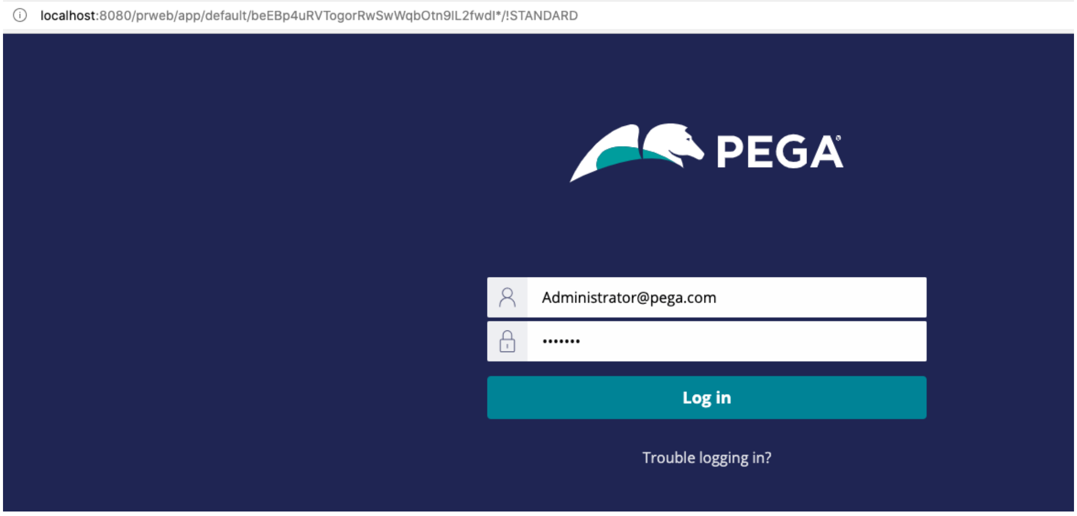
pegatomcat pegatomcat
RUNNING PORT: 8080

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.



```
debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve
startup time and JSP compilation time.
27-Aug-2020 16:55:19.976 INFO [53da18961427] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory
[/opt/tomcat/webapps/host-manager] has finished in [626] ms
27-Aug-2020 16:55:19.978 INFO [53da18961427] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory
[/opt/tomcat/webapps/examples]
27-Aug-2020 16:55:21.250 INFO [53da18961427] org.apache.jasper.servlet.TldScanner.scanJars At least one JAR was scanned for TLDs yet contained no TLDs. Enable
debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve
startup time and JSP compilation time.
27-Aug-2020 16:55:21.485 INFO [53da18961427] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory
[/opt/tomcat/webapps/examples] has finished in [1,507] ms
27-Aug-2020 16:55:21.485 INFO [53da18961427] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory
[/opt/tomcat/webapps/manager]
27-Aug-2020 16:55:21.890 INFO [53da18961427] org.apache.jasper.servlet.TldScanner.scanJars At least one JAR was scanned for TLDs yet contained no TLDs. Enable
debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve
startup time and JSP compilation time.
27-Aug-2020 16:55:21.894 INFO [53da18961427] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory
[/opt/tomcat/webapps/manager] has finished in [409] ms
27-Aug-2020 16:55:21.939 INFO [53da18961427] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8080"]
27-Aug-2020 16:55:22.035 INFO [53da18961427] org.apache.catalina.startup.Catalina.start Server startup in [383762] milliseconds
2020-08-27 16:55:32,089 [ervice-registry-task] [ STANDARD] [ ] [ ] (.tasks.TopologyChangedDetector) INFO - Services topology has changed. Current topology:
This node ==> ADM.Default 4e554a49166f2a6a93902b768f32bd8e 172.17.0.3 NORMAL 8.4.0 a6ffd4a0-11c1-46cb-8525-38a7f046ad78
This node ==> Aggregation.Default 4e554a49166f2a6a93902b768f32bd8e 172.17.0.3 NORMAL 8.4.0 ab998838-0eff-4308-ad0c-72367d7bf731
This node ==> DDS.Default 4e554a49166f2a6a93902b768f32bd8e 172.17.0.3 NORMAL 8.4.0 9f5129b7-aef2-4c93-95ce-65b58f90c4c8
This node ==> DataFlow.BackgroundProcessing 4e554a49166f2a6a93902b768f32bd8e 172.17.0.3 NORMAL 8.4.0 d45384a0-2600-494c-9b05-5a9492abc70d
This node ==> DataFlow.Batch 4e554a49166f2a6a93902b768f32bd8e 172.17.0.3 NORMAL 8.4.0 1024cc92-2e61-4f2e-bed2-74b24fe41606
This node ==> DataFlow.RealTime 4e554a49166f2a6a93902b768f32bd8e 172.17.0.3 NORMAL 8.4.0 5aa1420a-5ffa-4bcb-9c8d-8856770a064a
This node ==> DataFlow.Search 4e554a49166f2a6a93902b768f32bd8e 172.17.0.3 NORMAL 8.4.0 788f90f7-bfd7-4556-930f-8abb7bb121a9
This node ==> StreamServer.Default 4e554a49166f2a6a93902b768f32bd8e 172.17.0.3 NORMAL 8.4.0 b0b0f3f3-cf26-4a6e-af56-d0e10e116fcd
This node ==> VBD.Default 4e554a49166f2a6a93902b768f32bd8e 172.17.0.3 NORMAL 8.4.0 1457b595-d520-41a0-8d40-35440204d91f
This node ==> pega 4e554a49166f2a6a93902b768f32bd8e 172.17.0.3 running (was startingUp) 8.4.0 2a0c3483-bee2-42bb-9892-280e0bd5990d
```

4. Access Pega using http://localhost:8080/prweb



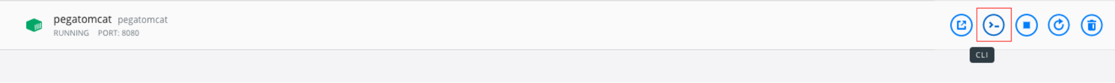
localhost:8080/prweb/app/default/beEBp4uRVTogorRwSwWqbOtn9IL2fwdI*/!STANDARD

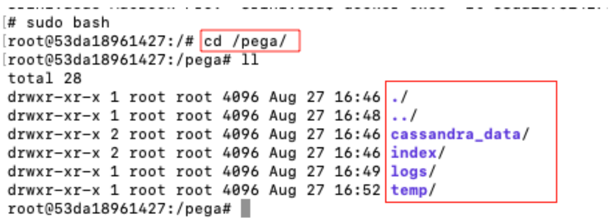
Administrator@pega.com

•••••••

Administrator@pega.com / install

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





```
[# sudo bash
[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#
```

6. Happy adventure… 😊

Extracting Pega Postgres DB from Personal Edition

1. As explained earlier in this document, Postgres DB is extracted from Personal Edition. For this purpose, pg_dump tool is used. (this tool comes with pgAdmin 4)

2. Below example is taken from Windows, for other OS, go through the postgres recommendations.

3. Run below command to extract postgres DB to a folder "D:\Share\PegaPE_840_DB_Postgres.sql"

   C:\Program Files\pgAdmin 4\v4\runtime> .\pg_dump.exe -f "D:\Share\PegaPE_840_DB_Postgres.sql" -U postgres -W postgres