### world map graphic

Pega Docker

Postgres Container

Srinivasa Kandru

**Sreesoft Solutions**

##### [Click here to enter a date]

Contents

[1](#_Toc49359872)

[Check Prerequisites 3](#_Toc49359873)

[Build Docker Image 3](#_Toc49359874)

[Configure Pega Postgres image 4](#_Toc49359875)

[Running Pega Instance(s) 4](#_Toc49359876)

# 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.

A screenshot of a cell phone

Description automatically generated

1. Have a github account and familiarise how to check-out repositories.

# Build Docker Image

Docker file is available at Github repository <https://github.com/sreesoft/pegapostgres>.

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

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated



Check out this repository into a folder of choosing ($PPGS\_HOME) and build docker image using the below command. (via Terminal or Powershell)

“docker build -t pegapostgres .”

A screenshot of a cell phone

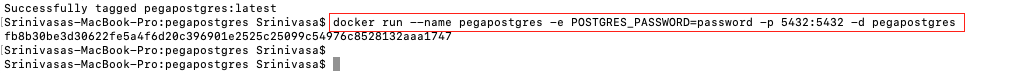
Description automatically generated

A screenshot of text

Description automatically generated

Once docker image created, run it using the below command

docker run --name pegapostgres -e POSTGRES\_PASSWORD=password -p 5432:5432 -d pegapostgres



A screenshot of a cell phone

Description automatically generated

# Configure Pega Postgres image

Postgres image uses a data volume and configuration files created upon start of postgres service. To change the postgres default config to adopt Pega specifications, copy prepared conf files into data folder as shown in below commands.

docker cp ./pg\_hba.conf pegapostgres:/var/lib/postgresql-static/data

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

Pega postgres DB backup is extracted from Personal Edition. Used pg\_dump tool to create a dump of all postgres DB.

In windows, command looks as below. For other OS, go through the postgres recommendations.

C:\Program Files\pgAdmin 4\v4\runtime> .\pg\_dump.exe -f "D:\Share\PegaPE\_840\_DB\_Postgres.sql" -U postgres -W postgres

For ubuntu

pg\_dump -U postgres -W pega > PegaPE\_840\_DB\_Postgres.sql

Copy this dump file into docker postgres container.

docker cp ./PegaPE\_840\_DB\_Postgres.sql pegapostgres:/tmp



Inside Docker container, run the below commands to restore the Pega DB.

sudo bash

sudo -su postgres

psql -d postgres -f /tmp/PegaPE\_840\_DB\_Postgres.sql

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Test Postgres connection via PGAdmin4 or SQL Developer

A screenshot of a computer

Description automatically generated

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Go to pegatomcat folder created when cloned github.

A close up of text on a white background

Description automatically generated

A close up of text on a white background

Description automatically generated

# Running Pega Instance(s)

with link to postgres

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



A screenshot of a cell phone

Description automatically generated

A screenshot of a computer

Description automatically generated

<http://localhost:8080/prweb>

A screenshot of a cell phone

Description automatically generated

[Administrator@pega.com](mailto:Administrator@pega.com) / install



A screenshot of a cell phone

Description automatically generated