CentOS 7 – PostgreSQL 12 & PGAdmin4

Version:	1.0.0
Created by:	cloudimg

Table of Contents

1.) Overview

This document is provided as a user guide for the CentOS 7 – PostgreSQL 12 & PGAdmin4 product offering on the Azure Marketplace. Please reach out to support@cloudimg.co.uk if any issues are encountered following this user guide for the chosen product offering.

cloudimg
(+44) 02045382725
3rd Floor 86-90 Paul Street London EC2A 4NE
support@cloudimg.co.uk
https://cloudimg.co.uk

2.)Access & Security

Please update the security group of the target instance to allow the below ports and protocols for access and connectivity.

Protocol	Type	Port	Description
SSH	TCP	22	SSH connectivity
HTTP	HTTP	80	PGAdmin4 Console
Custom TCP	TCP	5432	Postgres Database listener port for remote access

3.)System Requirements

The minimum system requirements for the chosen product offering can be found below

Minimum CPU	Minimum RAM	Required Disk Space
1	1 GB	20 GB

4.) Connecting to the Instance

Once launched in the Azure Virtual Machines Service, please connect to the instance via an SSH client using the **azureuser** with the key pair associated at launch. Once connected as the **azureuser**, you will be able to **sudo** to the **root** user by issuing the below command.

Switch to the root user

```
sudo su -
```

5.) On Startup

An OS package update script has been configured to run on boot to ensure the image is fully up to date at first use. You can disable this feature by removing the script from /stage/scripts/ and deleting the entry in crontab for the root user.

Disable the OS update script from running on reboot

```
rm -f /stage/scripts/initial_boot_update.sh

crontab -e
```

```
#DELETE THE BELOW LINE. SAVE AND EXIT THE FILE.
@reboot /stage/scripts/initial_boot_update.sh
```

6.) Filesystem Configuration

Please see below for a screenshot of the server disk configuration and specific mount point mappings for software locations.

```
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        472M     0  472M   0% /dev
tmpfs           482M   16K  482M   1% /dev/shm
tmpfs           482M  6.5M  475M   2% /run
tmpfs           482M     0  482M   0% /sys/fs/cgroup
/dev/nvme0n1p2   38G  3.1G   33G   9% /
/dev/nvme1n1    9.8G   76M  9.2G   1% /var/lib/pgsql
/dev/nvme0n1p1  2.0G   93M  1.7G   6% /boot
tmpfs            97M     0   97M   0% /run/user/1002
```

Mount Point	Description
/boot	Operating System Kernel files
/var/lib/postgres	Postgres installation directory

7.) Server Components

Please see below for a list of installed server components and their respective installation paths. The below versions are subject to change on initial boot based on the initial_boot_update.sh script finding new versions of the software in the systems package repositories.

Component	Version	Software Home
Cloud-Init	19.4	/etc/cloud
Apache HTTP Sever	2.4.6	/etc/httpd
Postgres Database Engine	12.11	/var/lib/pgsql
PGAdmin4	4	/usr/pgadmin4

Azure CLI	2.53.1	/lib64/az

8.) Scripts and Log Files

The below table provides a breakdown of any scripts & log files created to enhance the useability of the chosen offering.

Script/Log	Path	Description
Initial_boot_update.sh	/stage/scripts	Update the Operating System with the latest updates available.
Initial_boot_update.log	/stage/scripts	Provides output for initial_boot_update.sh
postgresql_root_database_password.log	/stage/scripts/	Postgres database user password

9.) Using System Components

Instructions can be found below for using each component of the server build mentioned in section 7 of this user guide document.

Azure CLI

Using Azure CLI – as any OS user.

```
az
```

Cloud-Init

Edit the /etc/cloud/cloud.cfg file to reflect your desired configuration. A link to the cloud-init official documentation can be found below for referencing best practise for your use case.

https://cloudinit.readthedocs.io/en/latest/

```
vi /etc/cloud/cloud.cfg
```

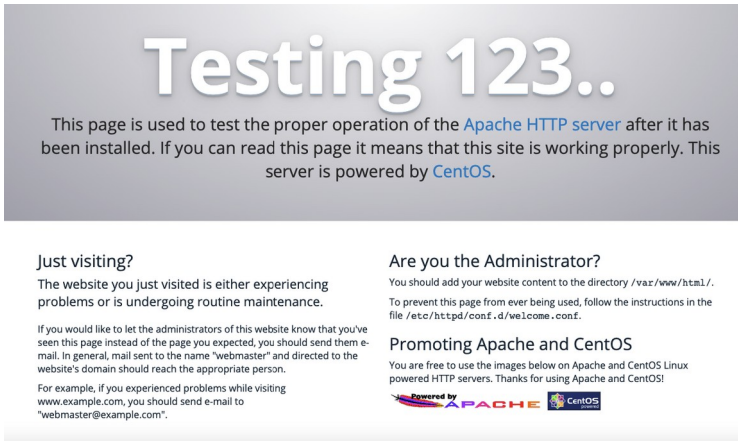
Apache HTTP Server

The Apache HTTP Server has been configured to start on boot, please use the below commands to start, stop and check the status of the service.

```
#Check the HTTP Server is running
systemctl status httpd

#Stop the HTTP Server
systemctl stop httpd

#Start the HTTP Server
systemctl start httpd
```

Once the HTTP Server status has started, you will be able to access the Apache front end via the below URL exchanging the values between <> to match that of your own instance.

<PRIVATE/PUBLICIP>:80



PGAdmin4

pgAdmin is designed to monitor and manage multiple PostgreSQL and EDB Advanced Server database servers, both local and remote, through a single graphical interface that allows the easy creation and management of database objects, as well as a number of other tools for managing your databases.

Upon initial use of the AMI, run the below script as the **root** user to configure the credentials required to access the PGAdmin4 Console running locally on the target server.

```
/usr/pgadmin4/bin/setup-web.sh
```

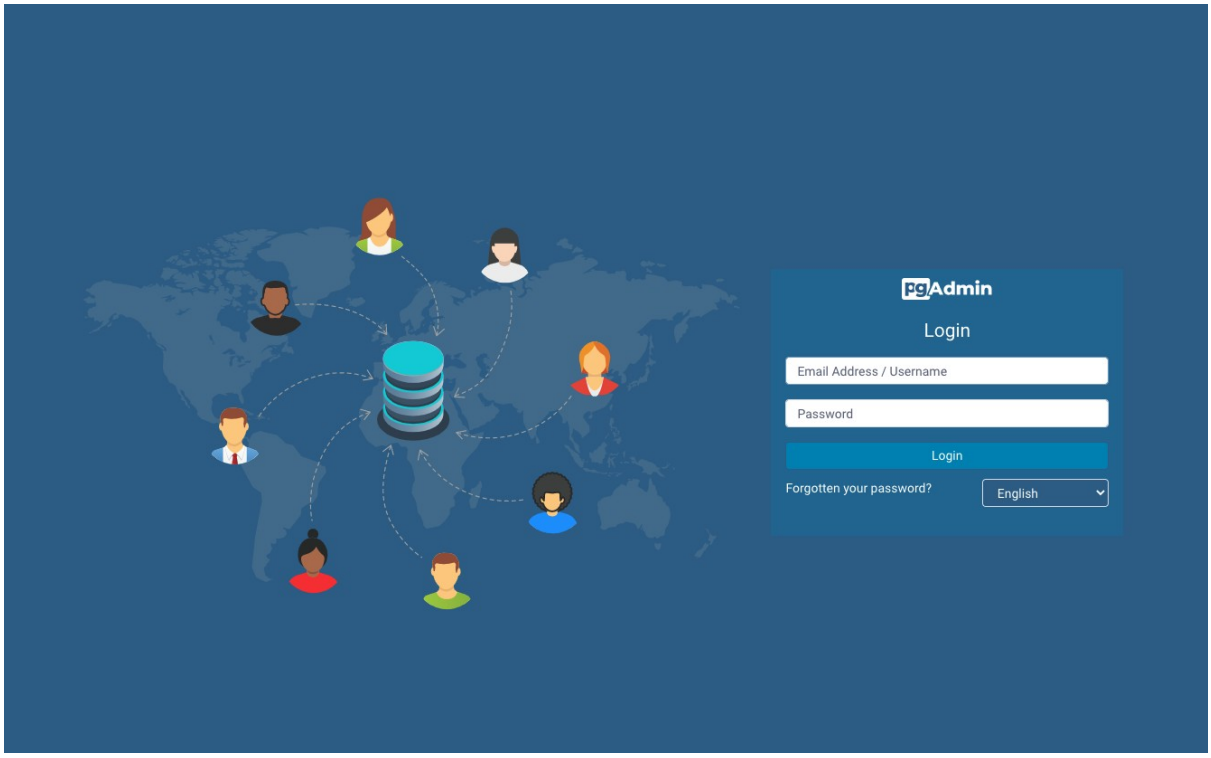
OUTPUT & RESPOND

```
[root@ip-172-31-93-13 ~]# /usr/pgadmin4/bin/setup-web.sh
Setting up pgAdmin 4 in web mode on a Redhat based platform...
Creating configuration database...
NOTE: Configuring authentication for SERVER mode.

Enter the email address and password to use for the initial pgAdmin user account:

Email address: example@cloudimg.co.uk   (This is an example, enter an email address of choice)
Password:      (Enter a password of choice)
Retype password: (Enter a password of choice)

pgAdmin 4 - Application Initialisation
======================================

Creating storage and log directories...
Configuring SELinux...
setsebool:  SELinux is disabled.
setsebool:  SELinux is disabled.
The Apache web server is running and must be restarted for the pgAdmin 4 installation to
complete. Continue (y/n)? y           (Enter y to restart Apache)
Apache successfully restarted. You can now start using pgAdmin 4 in web mode at
http://127.0.0.1/pgadmin4
[root@ip-172-31-93-13 ~]#
```

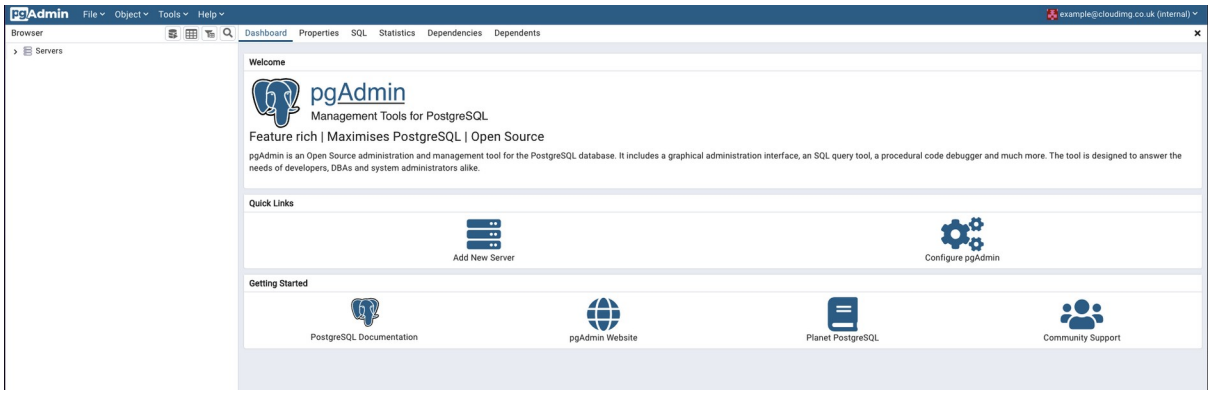
You will now be able to access the PGAdmin4 Front End via the below URL exchanging the values between the <> to match that of your instance.

<PUBLIC/PRIVATEIP>/pgadmin4

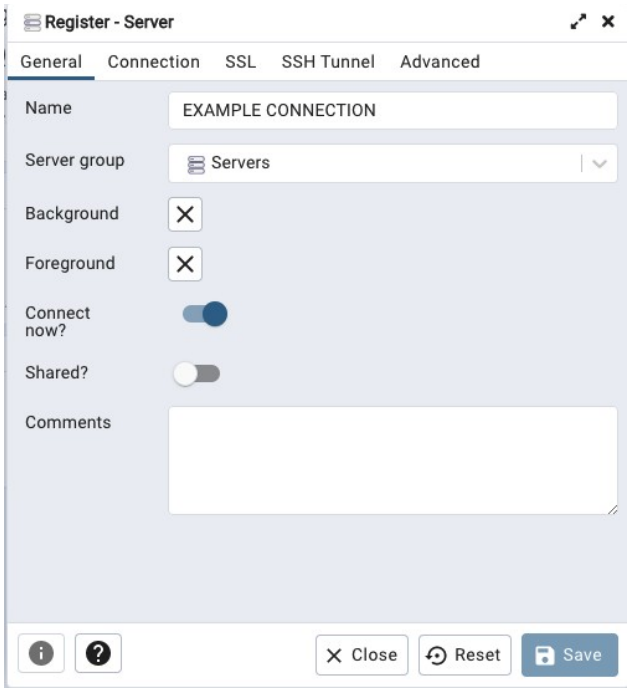
Enter the credentials created in the above usr/pgadmin4/bin/setup-web.sh script.



Click Login

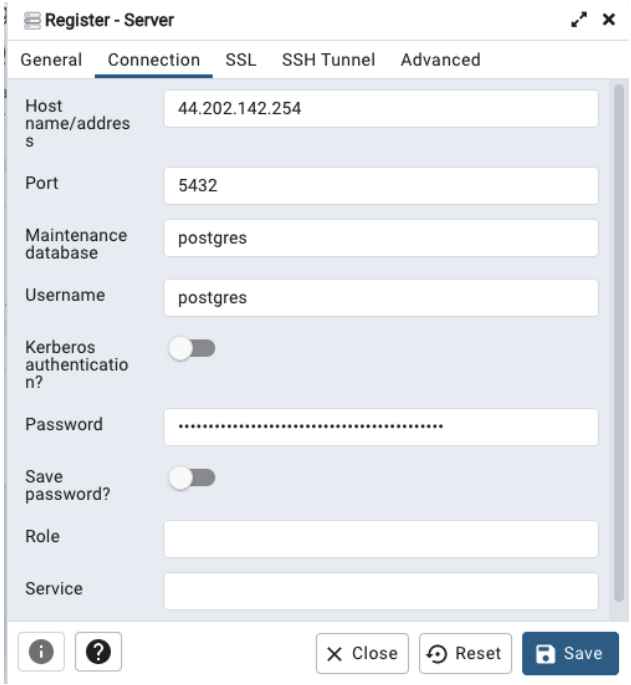


Select Add New Server to connect to the Postgres Database.



Enter a name for the connection, this example will use ‘EXAMPLE CONNECTION’

Select Connection



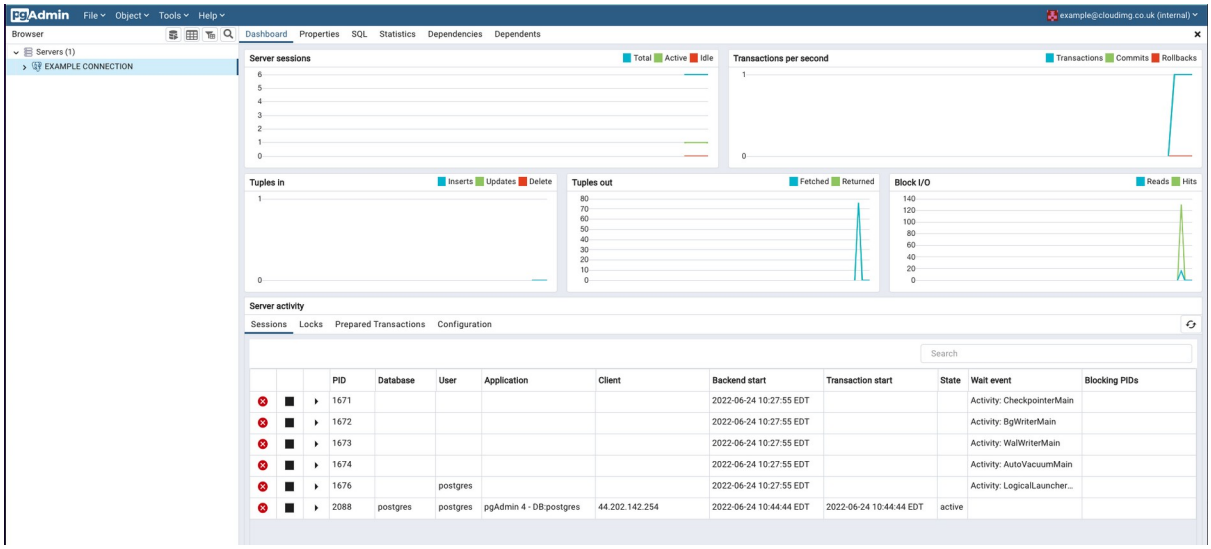
Enter in the Host field the IP address of your instance.

Enter Username: postgres

cloudimg
(+44) 02045382725
3rd Floor 86-90 Paul Street London EC2A 4NE
support@cloudimg.co.uk
https://cloudimg.co.uk

For Password enter the value found in the /stage/scripts/ postgresql_root_database_password.log file located on the instance.

Click Save



You will now have a remote connection to the Postgres database running on the instance.

Postgres Database Engine

For local administration and access to the postgres database, follow the below commands for checking the service, stopping the service & starting the service along with how to log into the database from the instance command line interface.

As the root user, run the below commands to check, stop or start the postgres database service. By default, the service is configured to start on boot.

```
#Check the postgres server status

systemctl status postgresql-12

#Stop the postgres database engine

systemctl stop postgresql-12

#Start the postgres database engine
```

```
systemctl start postgresql-12
```

Run the below commands as the **root** user to switch to the postgres OS user for accessing the postgres database engine command line interface from the instance locally.

```
sudo su - postgres

psql
```


cloudimg
(+44) 02045382725
3rd Floor 86-90 Paul Street London EC2A 4NE
support@cloudimg.co.uk
https://cloudimg.co.uk