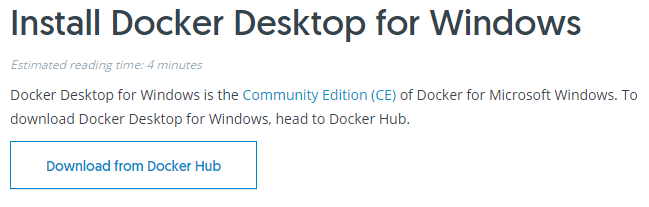
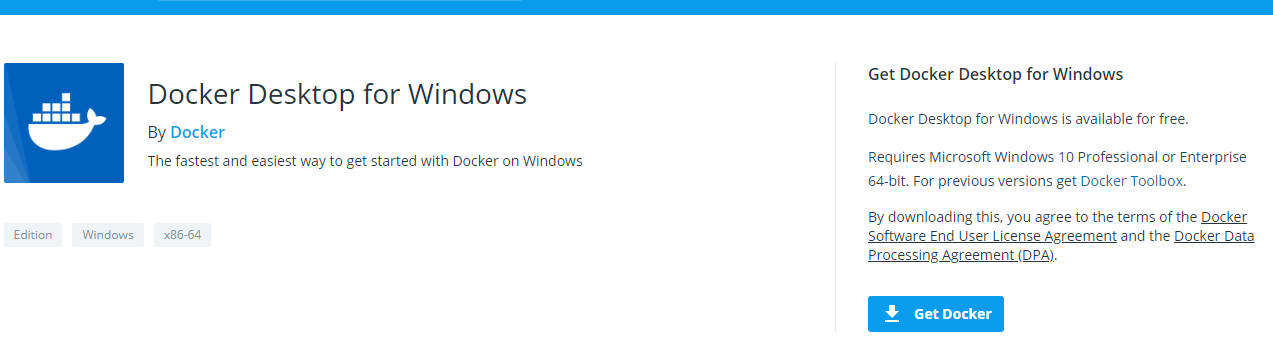
**Mobile and Internet Computing**

**To Download docker desktop for the windows**

* Go to this link(<https://docs.docker.com/docker-for-windows/install/>)



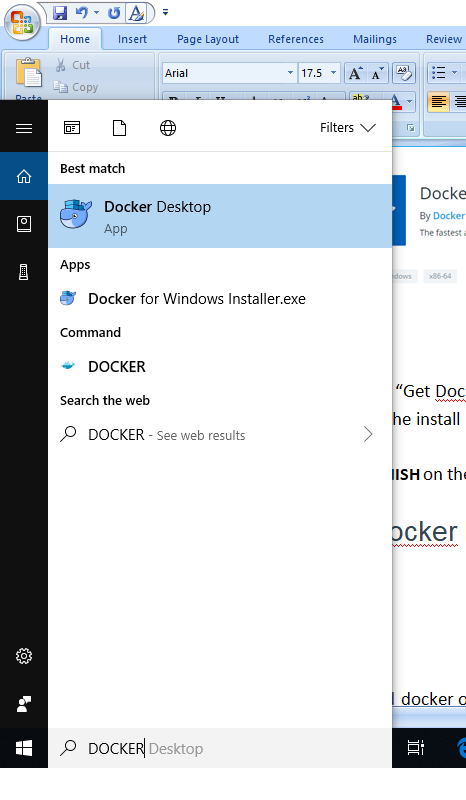
* Click on “Download from Docker Hub”, there you will find



* Click on “Get Docker”. Then docker desktop.exe will be downloaded.
* Follow the install wizard to accept the license, authorize the installer, and proceed with the install.
* Click **FINISH** on the setup complete dialog to launch Docker.

Start Docker Desktop for Windows

* Docker doesn’t start automatically after installation. You need to search for it in search box. Select Docker desktop for windows and click it



* Installed docker on my computer.
* Tested the docker environment.
* Test your installation of docker in command prompt.

1. Open a terminal window (Command Prompt or PowerShell, but not PowerShell ISE).
2. Run docker --version to ensure that you have a supported version of Docker:
3. > docker --version
4. Docker version 18.03.0-ce, build 0520e24
5. Pull the [hello-world image](https://hub.docker.com/r/library/hello-world/) from Docker Hub and run a container:
6. > docker run hello-world
7. docker : Unable to find image 'hello-world:latest' locally
8. ...
9. latest:
10. Pulling from library/hello-world
11. ca4f61b1923c:
12. Pulling fs layer
13. ca4f61b1923c:
14. Download complete
15. ca4f61b1923c:
16. Pull complete
17. Digest: sha256:97ce6fa4b6cdc0790cda65fe7290b74cfebd9fa0c9b8c38e979330d547d22ce1
18. Status: Downloaded newer image for hello-world:latest
19. Hello from Docker!
20. This message shows that your installation appears to be working correctly.
21. ...
22. List the hello-world image that was downloaded from Docker Hub:
23. > docker image ls
24. List the hello-world container (that exited after displaying “Hello from Docker!”):
25. > docker container ls --all
26. Explore the Docker help pages by running some help commands:
27. > docker --help
28. > docker container --help
29. > docker container ls --help

> docker run --help

# How to use PostgreSQL client applications

This article describes how to use PostgreSQL client applications to access and manage your PostgreSQL databases.

Table of Contents

* [PostgreSQL client applications](https://www.a2hosting.com/kb/developer-corner/postgresql/postgresql-client-applications#PostgreSQL-client-applications)
* [Using pgAdmin III](https://www.a2hosting.com/kb/developer-corner/postgresql/postgresql-client-applications#Using-pgAdmin-III)
* [More Information](https://www.a2hosting.com/kb/developer-corner/postgresql/postgresql-client-applications#More-Information)
* [Related Articles](https://www.a2hosting.com/kb/developer-corner/postgresql/postgresql-client-applications#Related-Articles)

#### POSTGRESQL CLIENT APPLICATIONS

There are numerous GUI (graphical user interface) client applications that you can use to manage PostgreSQL databases. These client applications enable you to view databases, run SQL queries, and more.

One of the most popular and widely-used PostgreSQL client applications is pgAdmin III. This application is free to download and use, and there are versions available for several operating systems, including Microsoft Windows, Mac OS X, and Linux.

#### USING PGADMIN III

To download pgAdmin III, visit <http://www.pgadmin.org/download>. Follow the instructions for your computer's operating system to install the application.

After you have downloaded and installed pgAdmin III, you can use it to manage your PostgreSQL databases. To do this, follow these steps:

1. You must first set up a remote PostgreSQL connection, either by using a SSH tunnel, or by using a direct connection to the A2 Hosting server.

For detailed information about how to set up a remote PostgreSQL connection, please see [this article](https://www.a2hosting.com/kb/developer-corner/postgresql/remote-postgresql-connections).

1. After you have established a remote connection to the server, start pgAdmin III.
2. On the File menu, click Add Server.
3. On the Properties tab, in the Name text box, type a name for the server connection. The name can be whatever you want.
4. In the Host text box, type localhost.
5. Confirm that the Port text box value is set to **5432**.
6. In the Maintenance DB text box, type the name of the PostgreSQL database that you want to access.
7. In the Username text box, type the username for the PostgreSQL database.
8. In the Password text box, type the username's password.

To have pgAdmin III remember the password, select the Store passwordcheck box.

1. Click OK. pgAdmin III connects to the server.
2. In the Object browser pane, click the + icon next to the connection name you specified in step 4. The tree expands to show Databases, Tablespaces, Group Roles, and Login Roles.
3. Click Databases, and then click the name of the database you specified in step 7.
   * To access the PostgreSQL console for the database, on the Plugins menu, click PSQL Console.
   * To run the SQL query tool, on the Tools menu, click Query tool.

To setup the database on docker

* Pull the image of postgres from docker hub

docker pull postgres

* If it’s running on a specific server, use your server IP. (For Windows docker-machine you probably need to use 192.168.99.100).

Commands

C:\Users\Deepak>psql -h localhost -p 5432 -U postgres -W

Password:

psql (11.2)

WARNING: Console code page (437) differs from Windows code page (1252)

8-bit characters might not work correctly. See psql reference

page "Notes for Windows users" for details.

Type "help" for help.

postgres=# \l

server

Success. You can now start the database server using:

pg\_ctl -D /var/lib/postgresql/data -l logfile start

#### METHOD #2: SET UP A DIRECT CONNECTION

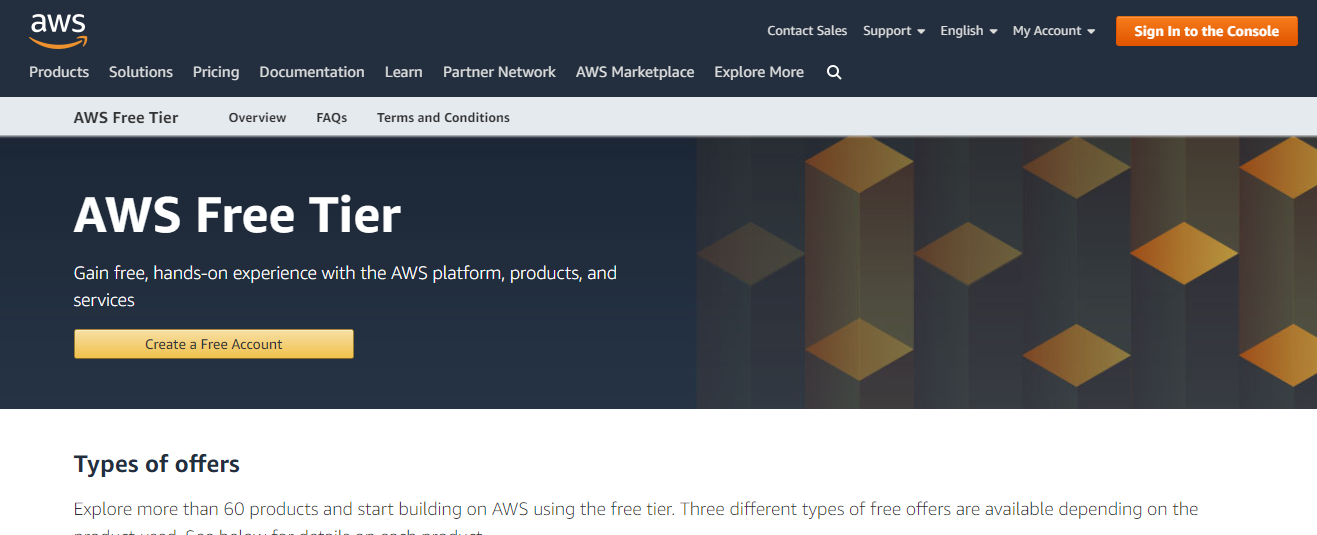
To set up a direct connection between your local site and the PostgreSQL server, you must configure a client application. There are several PostgreSQL client applications available, but for all of them, you must provide the following information to establish a remote connection:

* The name of the remote server (for example, a2ss42.a2hosting.com).
* The port of the remote server (this is always 5432).
* The PostgreSQL database name.
* The PostgreSQL database username.
* The PostgreSQL database password.

Steps to create an AWS free account are:

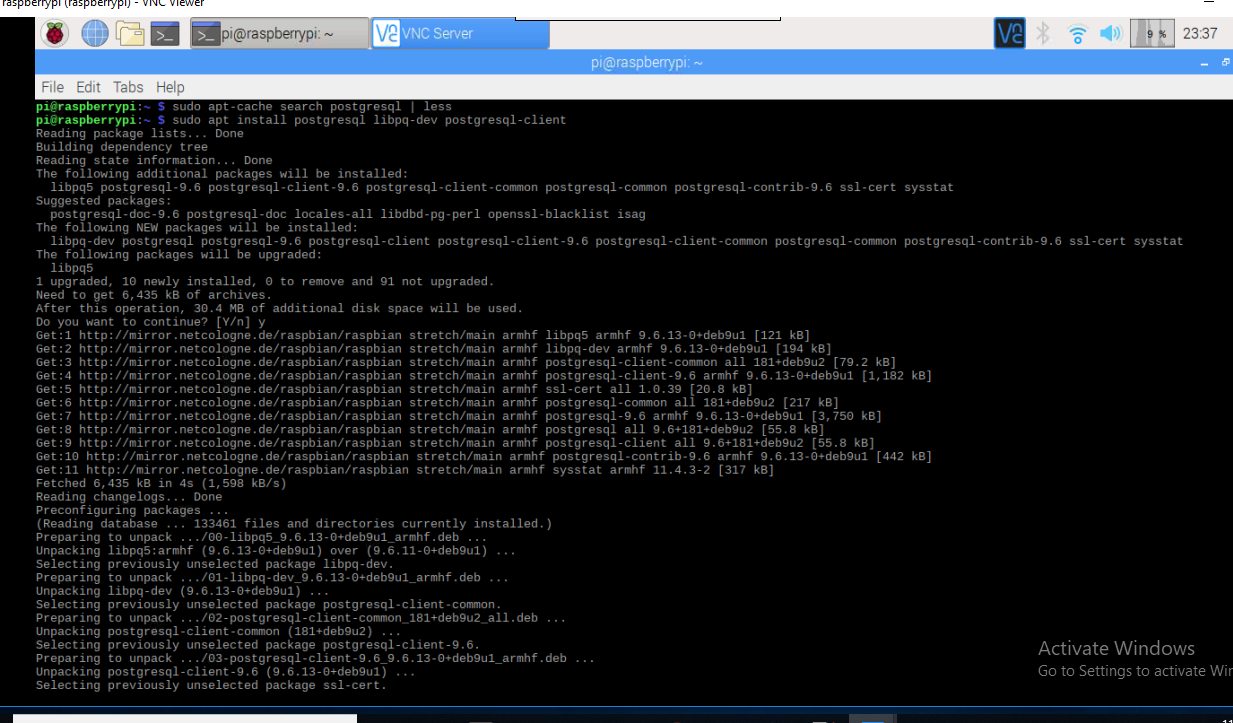
Open your web browser and navigate the link <https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=categories%23featured>

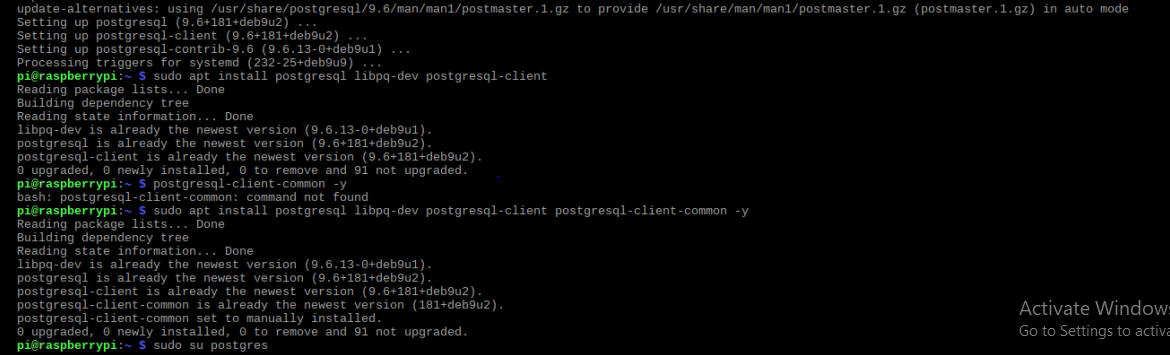
Click on the**“Sign Up Now” or “Create an AWS Account”** button.

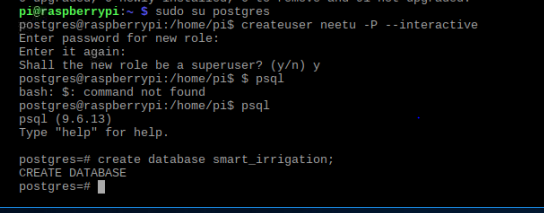


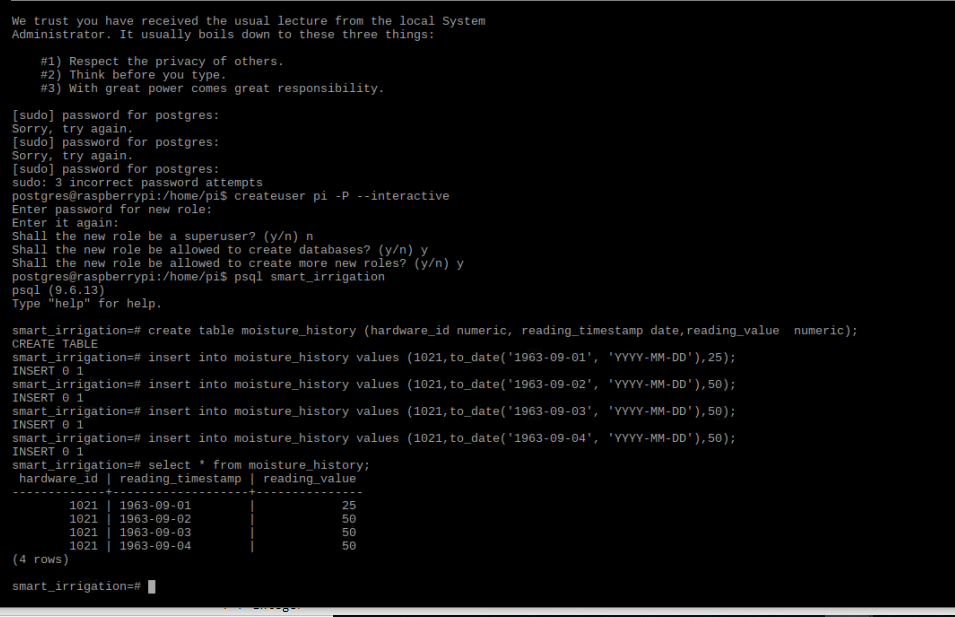
Installation of postgres database on raspberry pi:

<https://opensource.com/article/17/10/set-postgres-database-your-raspberry-pi>









Steps to install a keyboard in raspberry pi without a physical keyboard

1. Write the password for wifi and commands to install keyboard in a text document on a USB drive or SD card

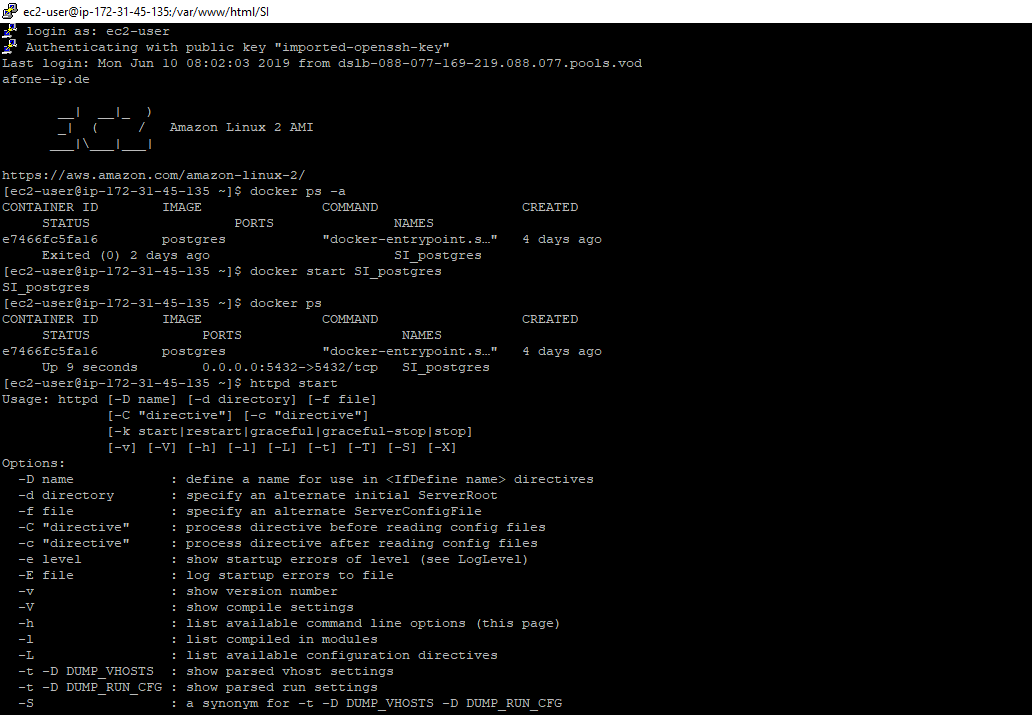
2. Open the text document , copy and paste the commands as per requirement.

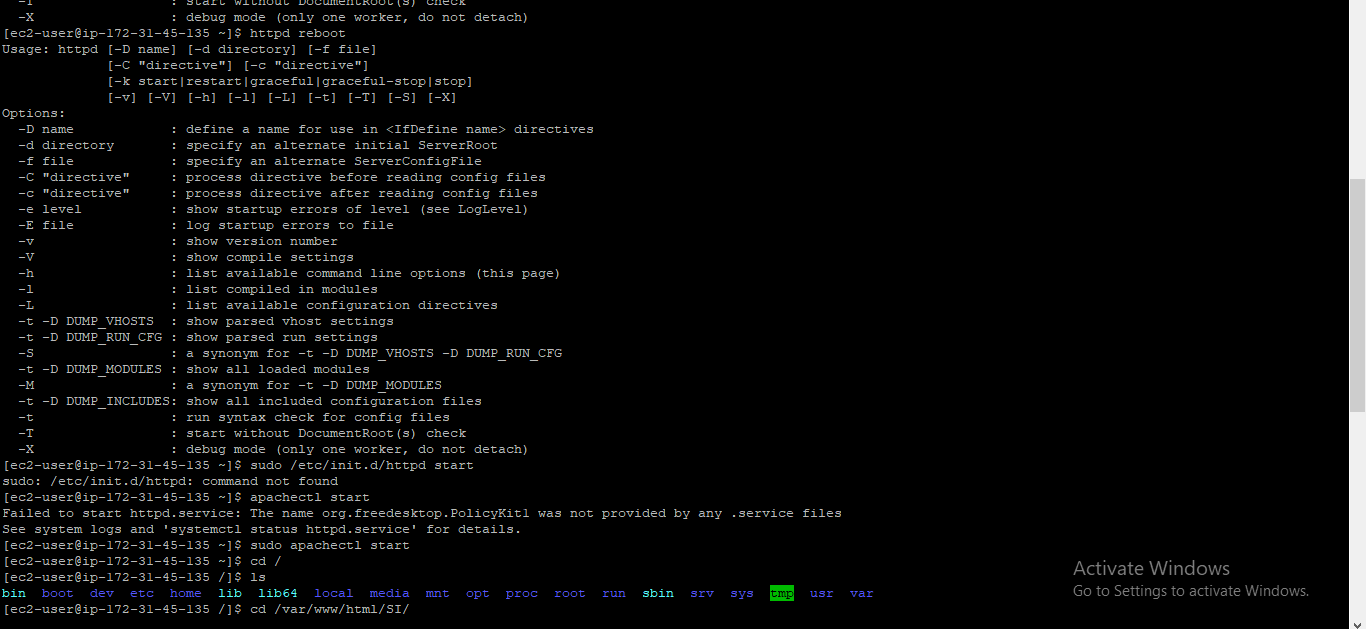
3. In my case, I copied and pasted the password for the wifi so that I can use Internet to install the keyboard.

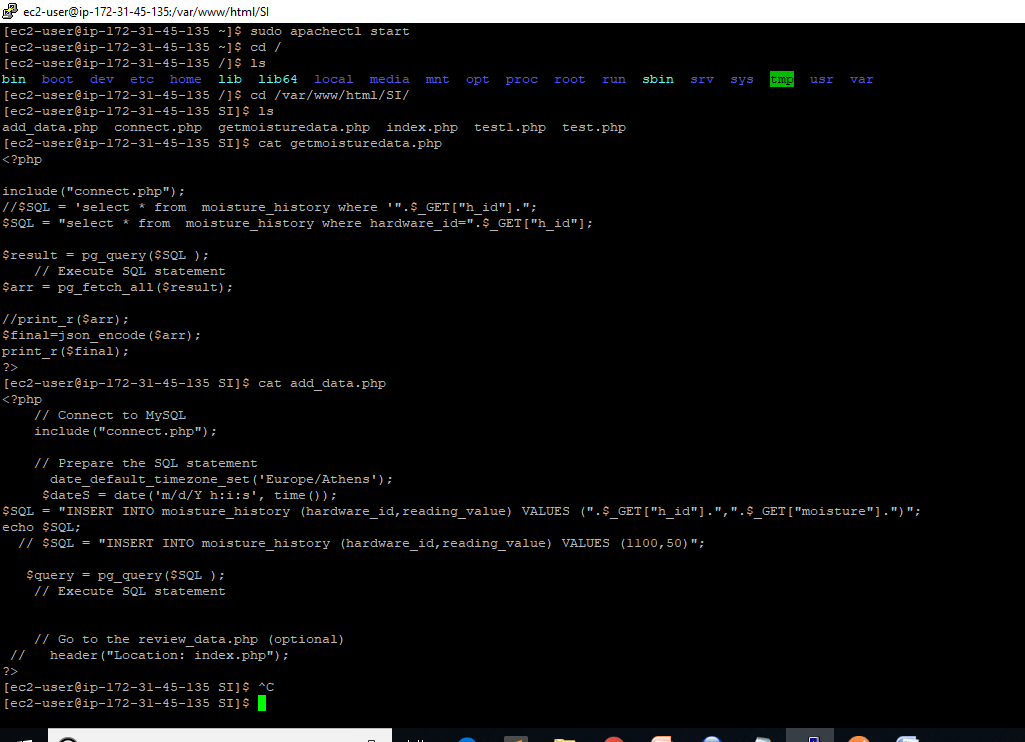
4. Write the following commands in terminal to install virtual keyboard.

Matchbox-keyboard

Open Terminal







**Steps to setup AWS central database in Docker**

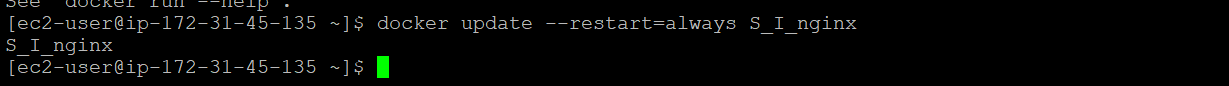
1. Create postgres container using postgres image.

docker run --name SI\_postgres -p 5432:5432 -v $HOME/docker/volumes/postgres:/var/lib/postgresql/data postgres

1. Setup to run the Docker on system startup.

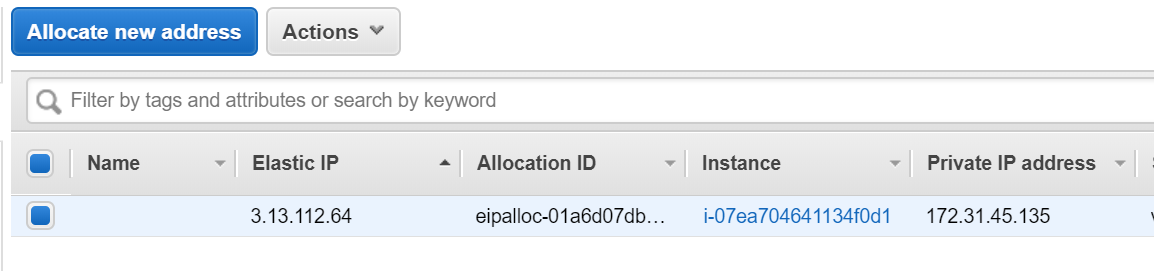
$ sudosystemctlenable docker

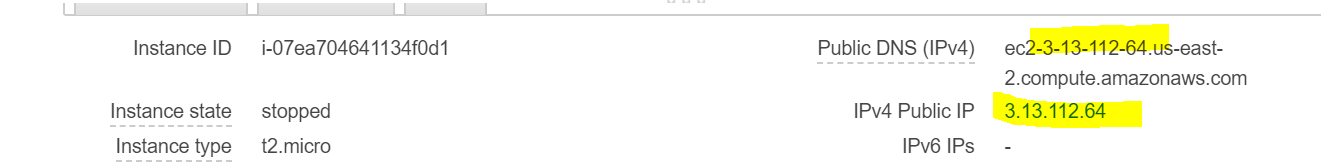
1. Setup to run the docker container on system startup.



1. Public DNS in AWS changes after every reboot so it can not be used in the API as the address will keep on changing.

To solve this problem Elastic IP is implemented in AWS to keep the IP address same after reboot.

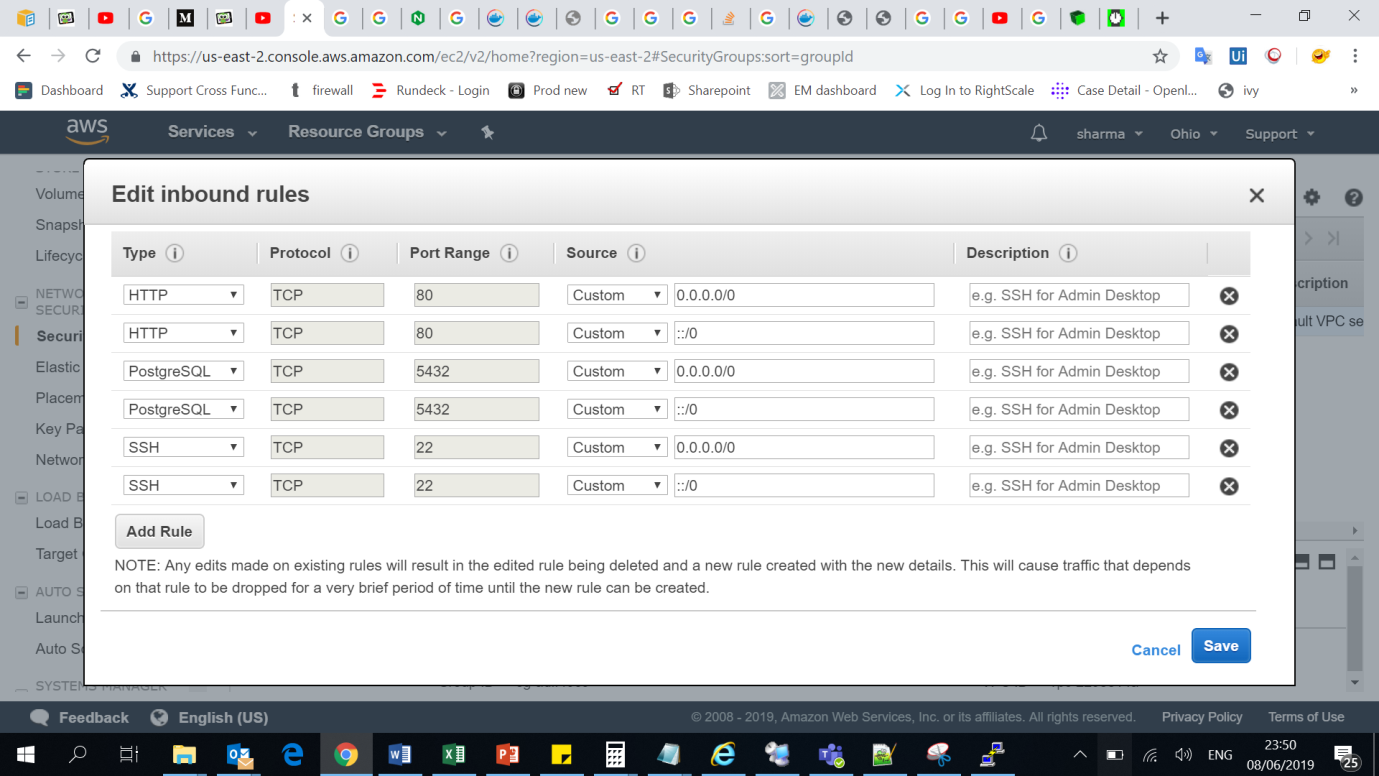




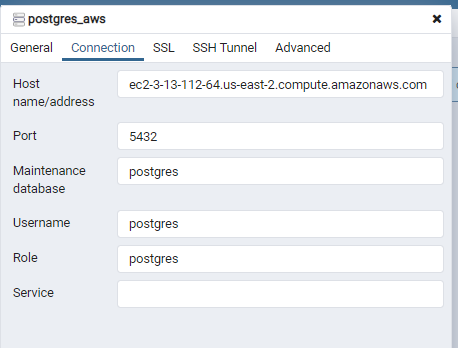
1. How to run command inside container?

docker exec -it S\_I\_nginx /bin/bash

1. Open port 5432 to connect pgadmin to postgres database created on Docker container inside AWS Instance.



1. Connect to Online database from pgadmin using details below.



1. Create the Tables in database using pgadmin and sql.

**Steps to setup the web server to enable DB to be accessible from anywhere using http request**

1. Install Apache server on AWS Linux instance

[ec2-user ~]$ **sudo yum update –y**

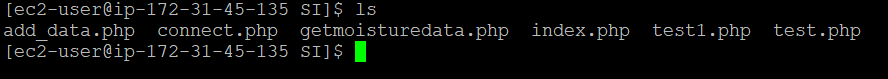
[ec2-user ~]$ **sudo yum install -y httpd24 php56 php56-mysqlnd**

1. **Start Apache server using below command**

$ sudo apachectl start

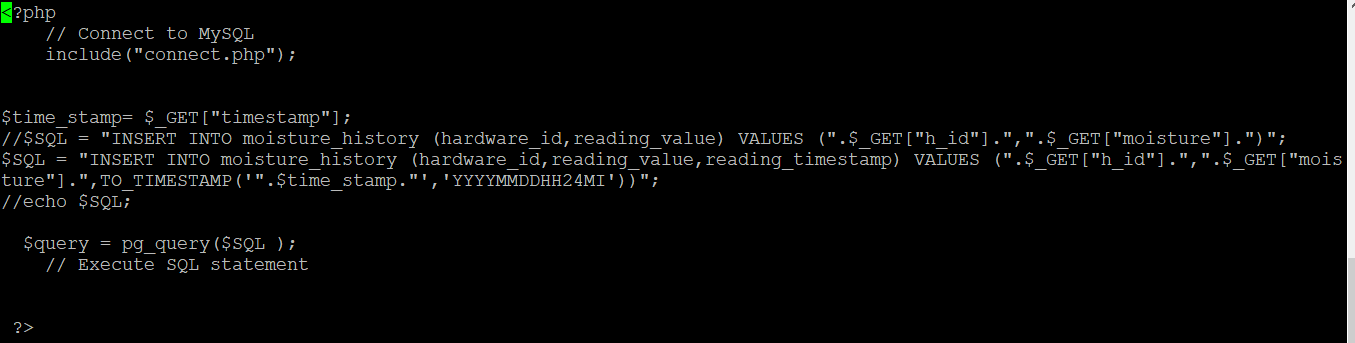
1. Create PHP scripts to access database using url in location

cd /var/www/html/SI/

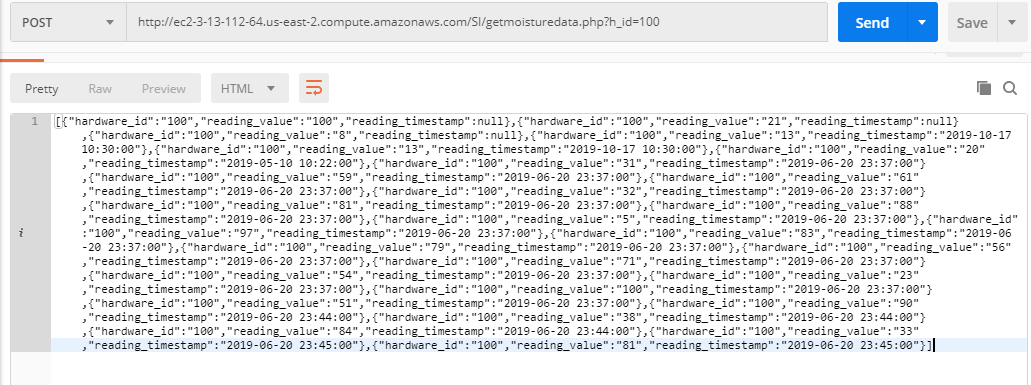


1. Create script to update table using http get request which could be used from Raspberry pi to save data into the AWS database using below url.

<http://ec2-3-13-112-64.us-east-2.compute.amazonaws.com/SI/add_data.php?h_id=100&moisture=20&timestamp=201905101022>

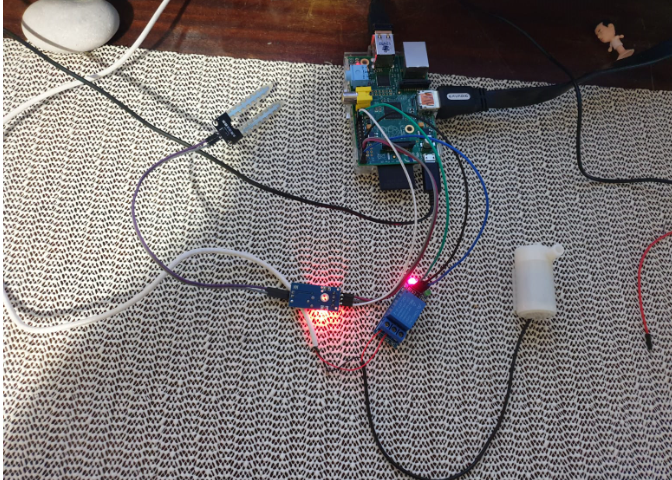


1. Create script to retrieve table data using http request from front end in json format.

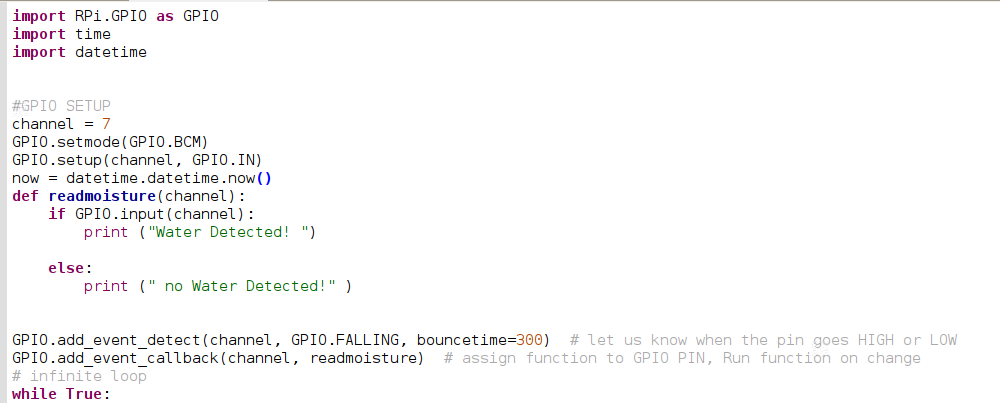


Python program in Raspberry pi to gather sensor reading and send to the AWS database created above

1. Connect the moisture sensor to GPIO7 and Motor control on GPIO8



1. Detect water using the water sensor by connecting it to GPIO 7 pin on Raspberry pi



1. Send the collected data to the AWS database using below script (ADC converter not available hence sending random values)



1. Control motor on GPIO pin with below python code.

