Install and test Bitnami Stack and pgAdmin

1. Download the Bitnami software

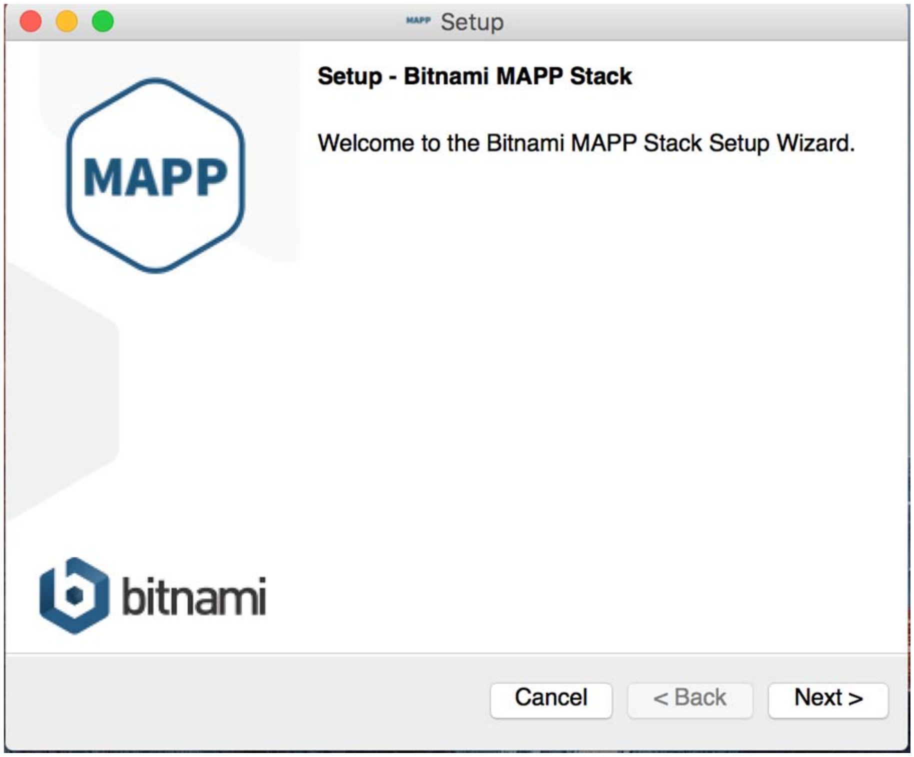
Sign in to Bitnami and download the M/W/LAAP stack according to your system: MAPP for Mac, WAPP for Windows and LAPP for Linux.

i. MAPP 7.0.22-1 (64-bit): https://bitnami.com/stack/mapp/installer
i. WAPP 7.0.22-1 (64-bit): https://bitnami.com/stack/wapp/installer
ii. LAPP 7.0.22-1 (64-bit): https://bitnami.com/stack/lapp/installer

2. Install WAPP/LAPP/MAPP

i. MAPP

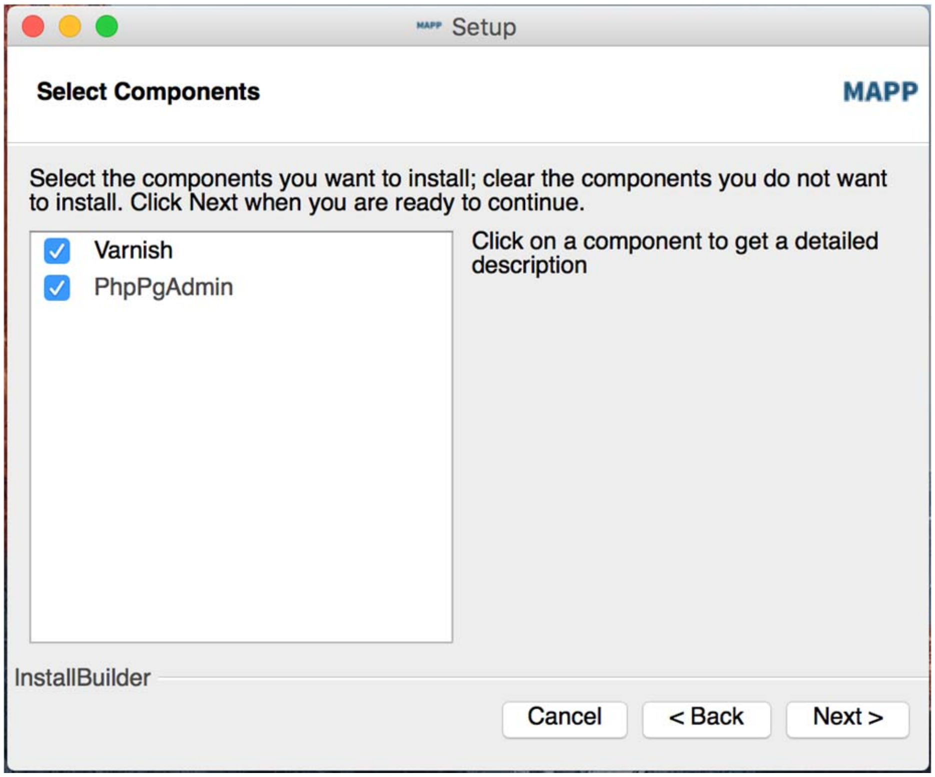
a) Activate MAPP installer dmg:



Setup - Bitnami MAPP Stack

Welcome to the Bitnami MAPP Stack Setup Wizard.

Click Next:



Select Components

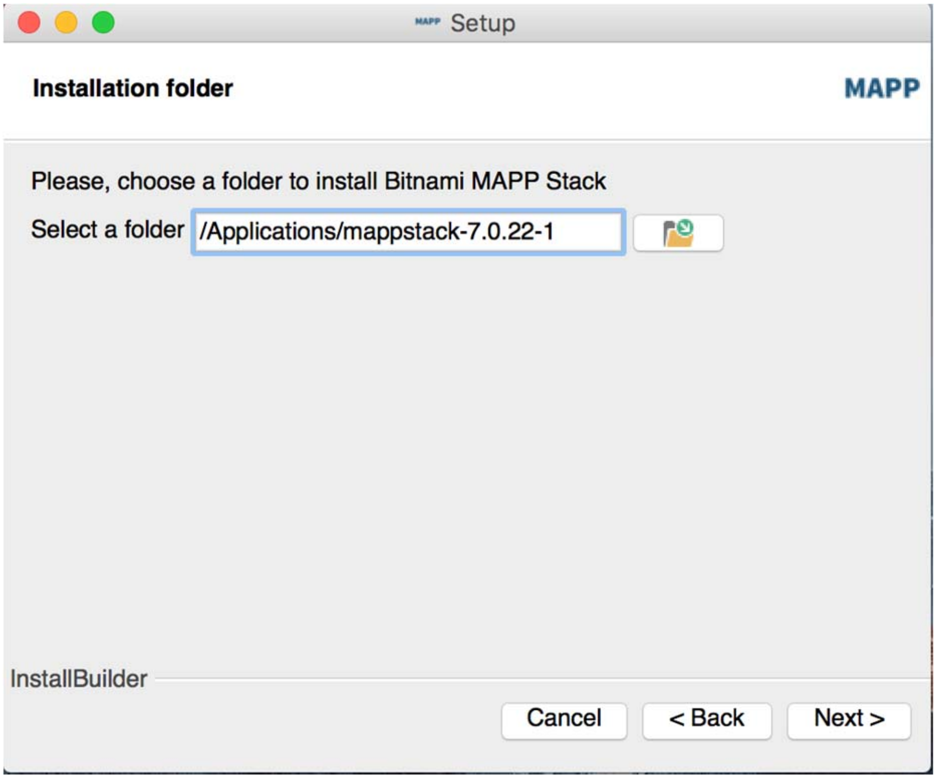
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

☑ Varnish
☑ PhpPgAdmin

Click on a component to get a detailed description

Click Next:

Note that, the folder here is your installation directory later referred to as INSTALLDIR.

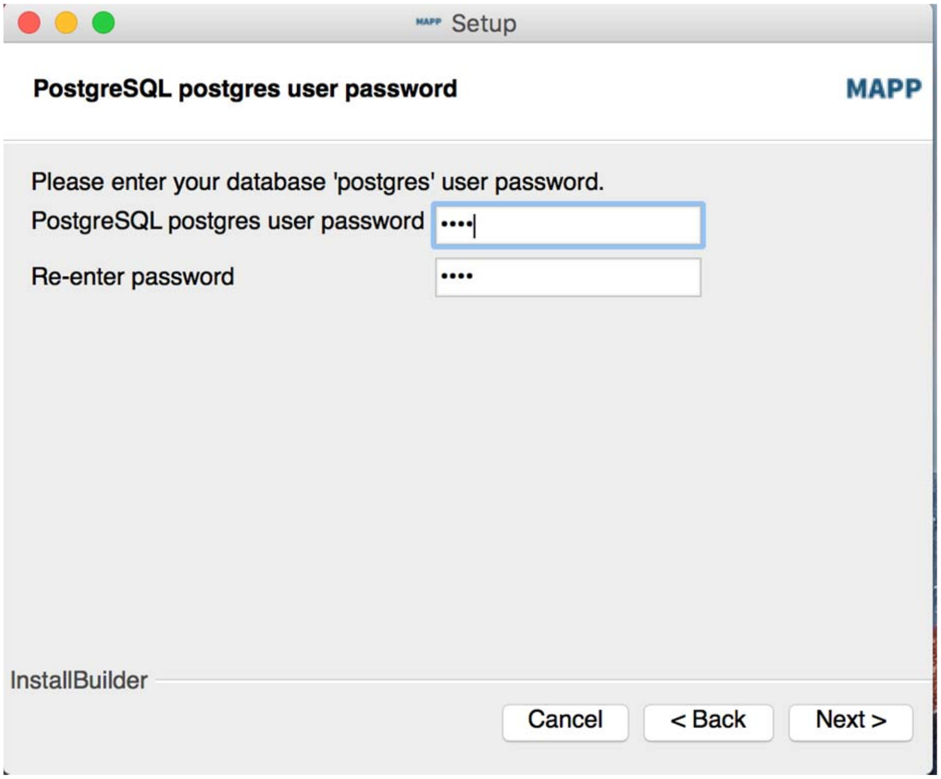


Installation folder

Please, choose a folder to install Bitnami MAPP Stack

Select a folder /Applications/mappstack-7.0.22-1

b) Input your database password (Remember it):



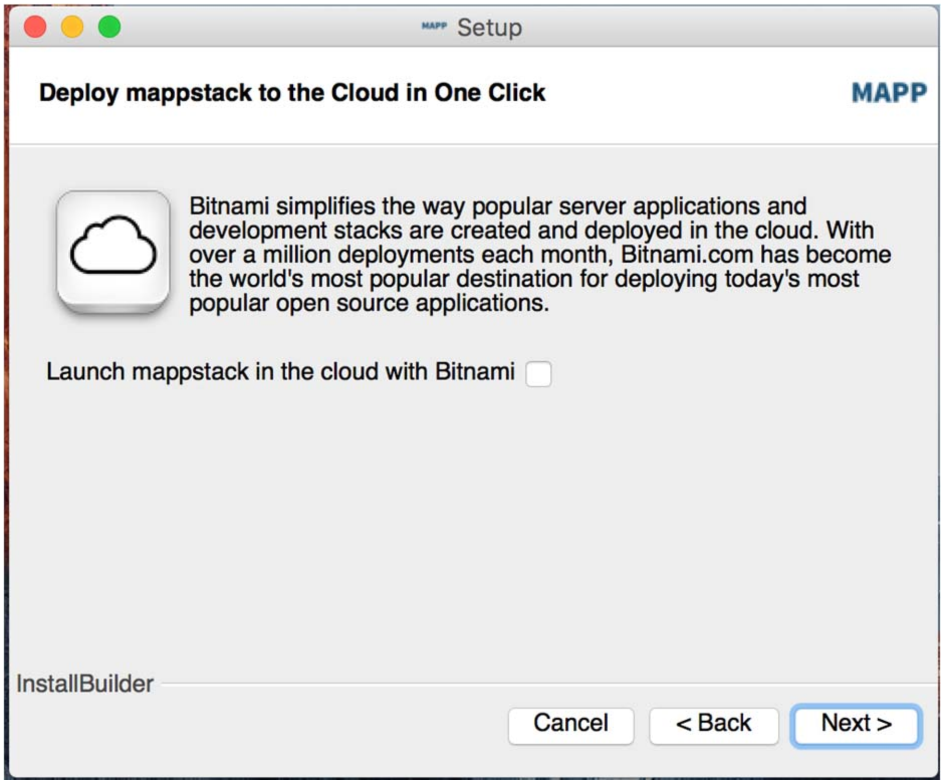
PostgreSQL postgres user password

Please enter your database 'postgres' user password.

PostgreSQL postgres user password ••••

Re-enter password ••••

c) Disable the cloud launch

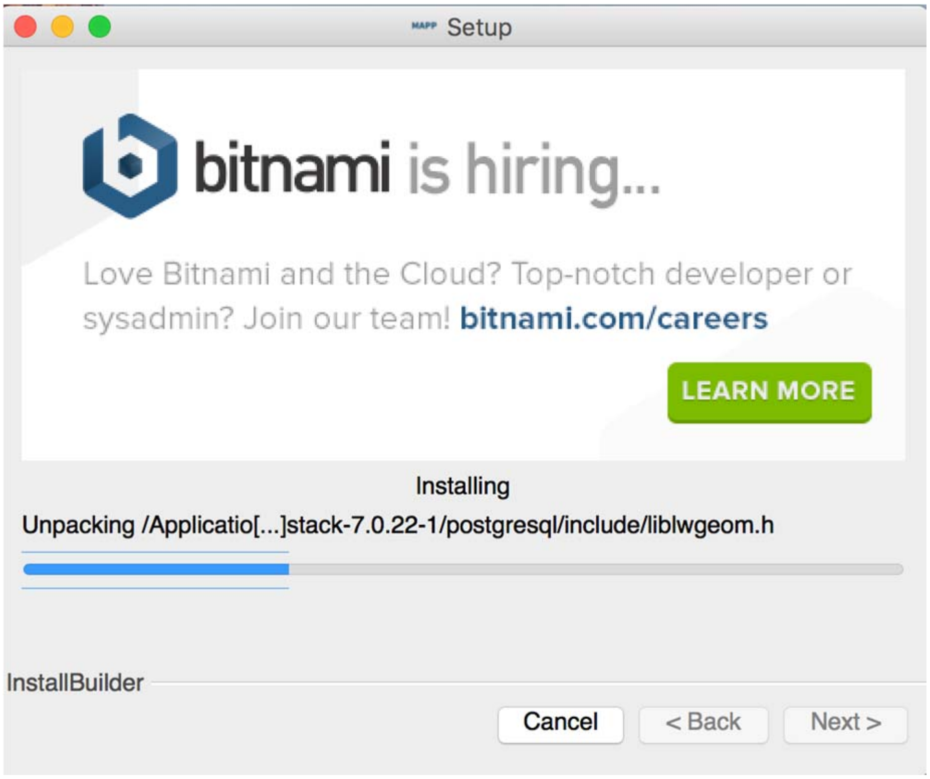


Deploy mappstack to the Cloud in One Click

Bitnami simplifies the way popular server applications and development stacks are created and deployed in the cloud. With over a million deployments each month, Bitnami.com has become the world's most popular destination for deploying today's most popular open source applications.

Launch mappstack in the cloud with Bitnami □

d) Wait till the installation completes:


MAPP Setup
bitnami is hiring...
Love Bitnami and the Cloud? Top-notch developer or sysadmin? Join our team! bitnami.com/careers
LEARN MORE
InstallBuilder
Cancel
< Back
Next >

Installing

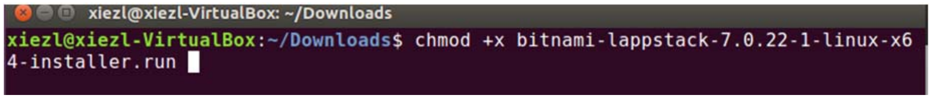
Unpacking /Applicatio[...]stack-7.0.22-1/postgresql/include/liblwgeom.h

e) Done.

ii. WAPP (Similar)

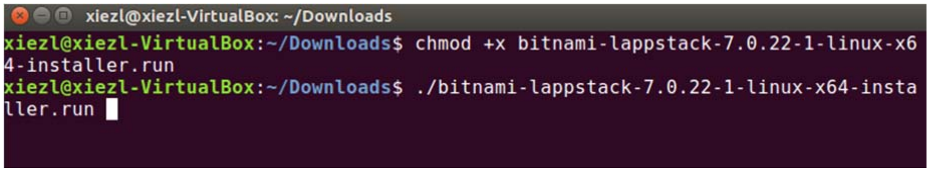
iii. LAPP:

a) Open a terminal, go to the folder where you downloaded LAPP installer and add execution permission to the downloaded installation script:



```
xiezl@xiezl-VirtualBox:~/Downloads$ chmod +x bitnami-lappstack-7.0.22-1-linux-x6
4-installer.run
```

Then execute the script:



```
xiezl@xiezl-VirtualBox:~/Downloads$ chmod +x bitnami-lappstack-7.0.22-1-linux-x6
4-installer.run
xiezl@xiezl-VirtualBox:~/Downloads$ ./bitnami-lappstack-7.0.22-1-linux-x64-insta
ller.run
```

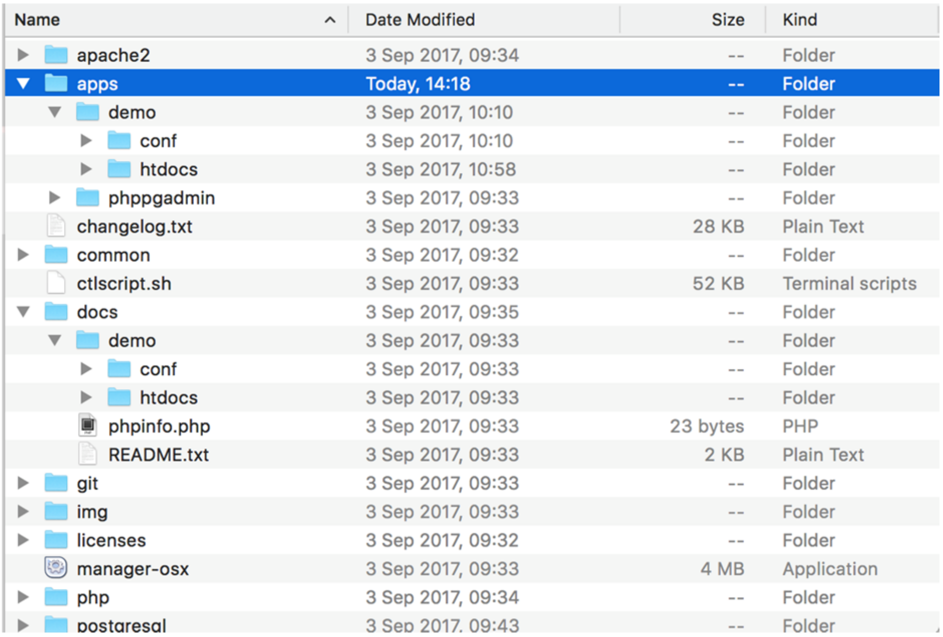
b) The rest steps are similar to MAPP installation

3. Test – Create the demo php site.

i. Open MAPP/WAPP/LAPP application (Note for Mac users: You might find it as ‘manager-osx’ in your Launchpad)

ii. Copy the entire folder INSTALLDIR/docs/demo under INSTALLDIR/apps/. So, under your apps directory you will find something like this:

(INSTALLDIR is the folder that WAPP/LAPP/MAPP is installed)



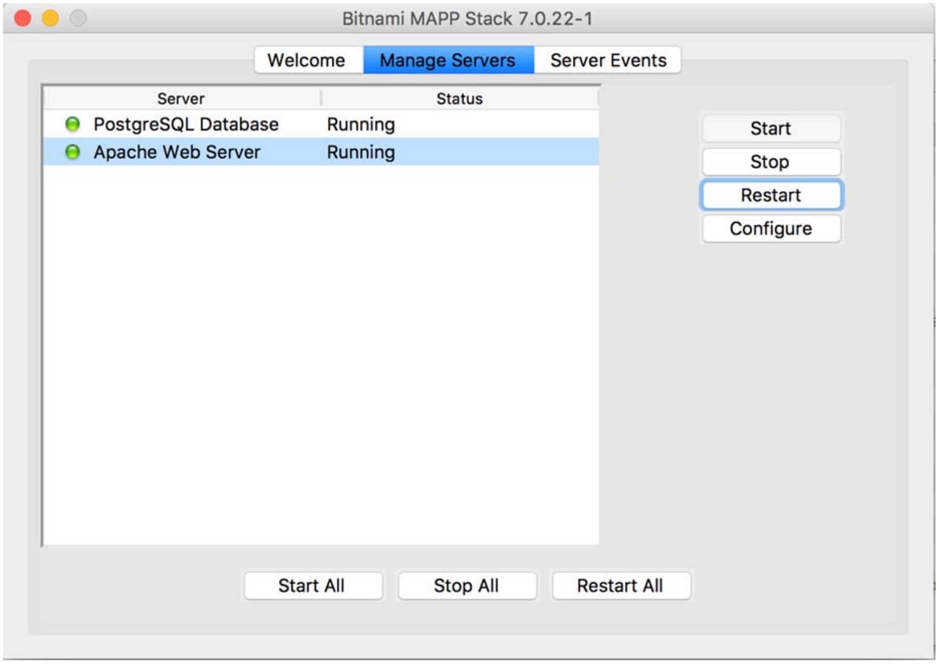
Name	Date Modified	Size	Kind
apache2	3 Sep 2017, 09:34	--	Folder
apps	Today, 14:18	--	Folder
demo	3 Sep 2017, 10:10	--	Folder
conf	3 Sep 2017, 10:10	--	Folder
htdocs	3 Sep 2017, 10:58	--	Folder
phppgadmin	3 Sep 2017, 09:33	--	Folder
changelog.txt	3 Sep 2017, 09:33	28 KB	Plain Text
common	3 Sep 2017, 09:32	--	Folder
ctlscript.sh	3 Sep 2017, 09:33	52 KB	Terminal scripts
docs	3 Sep 2017, 09:35	--	Folder
demo	3 Sep 2017, 09:33	--	Folder
conf	3 Sep 2017, 09:33	--	Folder
htdocs	3 Sep 2017, 09:33	--	Folder
phpinfo.php	3 Sep 2017, 09:33	23 bytes	PHP
README.txt	3 Sep 2017, 09:33	2 KB	Plain Text
git	3 Sep 2017, 09:33	--	Folder
img	3 Sep 2017, 09:33	--	Folder
licenses	3 Sep 2017, 09:32	--	Folder
manager-osx	3 Sep 2017, 09:33	4 MB	Application
php	3 Sep 2017, 09:34	--	Folder
postgresql	3 Sep 2017, 09:43	--	Folder

iii. Decompress the demo.zip file and replace the index.php file into folder INSTALLDIR/apps/demo/htdocs/.

iv. Add the following line to the end of the INSTALLDIR /apache2/conf/bitnami/bitnami-apps-prefix.conf file:

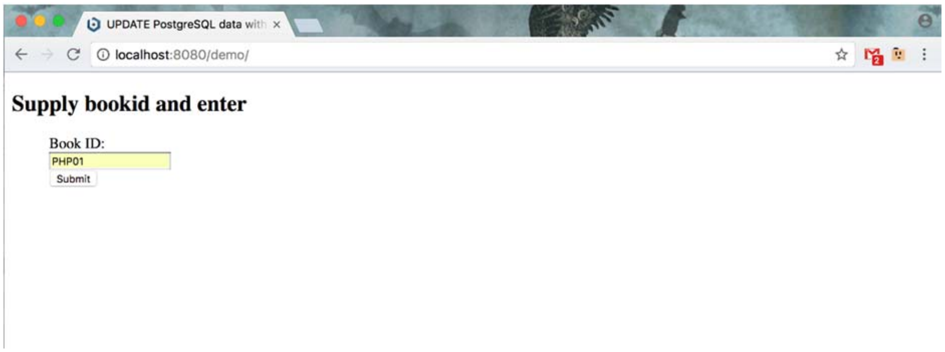
```
Include "installdir/apps/demo/conf/httpd-prefix.conf"
```

v. Go back to your MAPP/WAPP/LAPP application's "Manage Servers" panel to restart Apache Web Server.



Server	Status
PostgreSQL Database	Running
Apache Web Server	Running

vi. Open your browser and input the address http://localhost:8080/demo (Linux/mac) or http://localhost:80/demo (windows) and you should see the following page:



Supply bookid and enter

Book ID: PHP01

vii. Done.

4. Install pgAdmin 4 on windows/mac

i. Install pgAdmin:

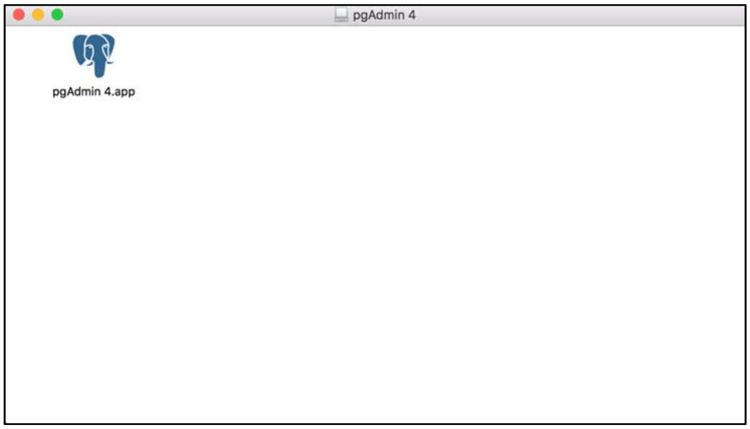
a) **Mac:**

i. Download pgAdmin4-1.6.dmg from https://www.postgresql.org/ftp/pgadmin/pgadmin4/v1.6/macos/

ii. Activate the installation dmg

iii. Click "Agree"

iv. Keep clicking next till you get the following window:



v. Open finder window

vi. Go to Applications folder

vii. Drag and Drop pgAdmin 4.app file to this folder

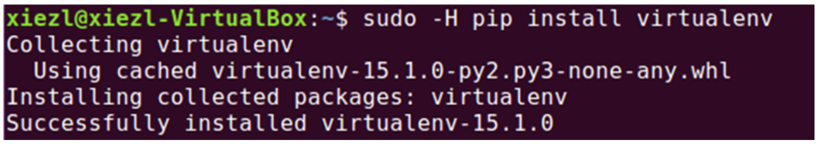
b) Windows:

i. Download pgAdmin4-1.6-x86.exe from https://www.postgresql.org/ftp/pgadmin/pgadmin4/v1.6/windows/

ii. Open the exe file and keep clicking agree or next to finish the installation

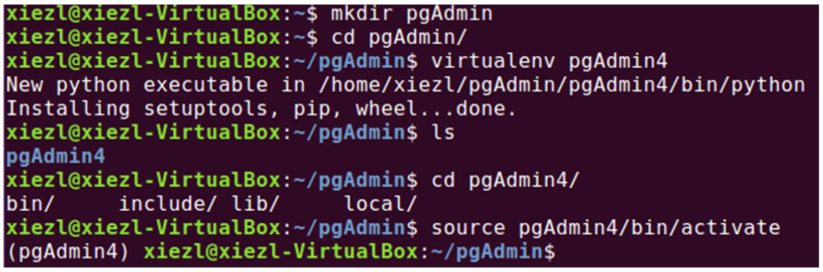
c) Linux:

i. Install virtual environment for python:



```
xiezl@xiezl-VirtualBox:~$ sudo -H pip install virtualenv
Collecting virtualenv
  Using cached virtualenv-15.1.0-py2.py3-none-any.whl
Installing collected packages: virtualenv
Successfully installed virtualenv-15.1.0
```

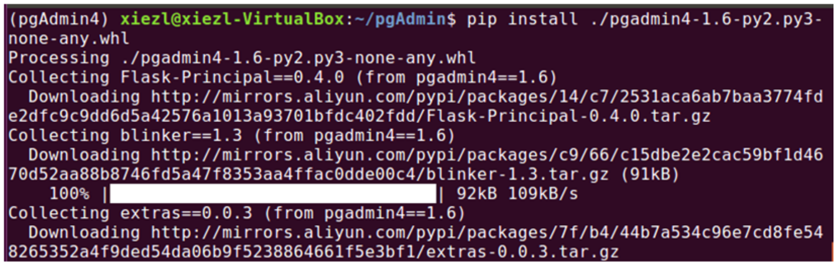
ii. Create the virtual environment under ~/pgAdmin folder:



```
xiezl@xiezl-VirtualBox:~$ mkdir pgAdmin
xiezl@xiezl-VirtualBox:~$ cd pgAdmin/
xiezl@xiezl-VirtualBox:~/pgAdmin$ virtualenv pgAdmin4
New python executable in /home/xiezl/pgAdmin/pgAdmin4/bin/python
Installing setuptools, pip, wheel...done.
xiezl@xiezl-VirtualBox:~/pgAdmin$ ls
pgAdmin4
xiezl@xiezl-VirtualBox:~/pgAdmin$ cd pgAdmin4/
bin/     include/ lib/      local/
xiezl@xiezl-VirtualBox:~/pgAdmin$ source pgAdmin4/bin/activate
(pgAdmin4) xiezl@xiezl-VirtualBox:~/pgAdmin$
```

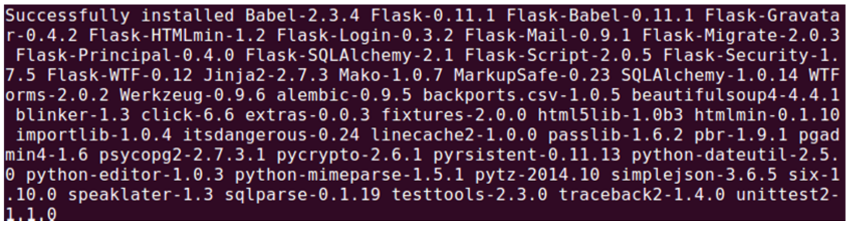
iii. Download pgadmin4-1.6-py2.py3-none-any.whl from https://www.postgresql.org/ftp/pgadmin/pgadmin4/v1.6/pip/ and copy it into ~/pgAdmin folder.

iv. Install the wheel file: (Long wait)



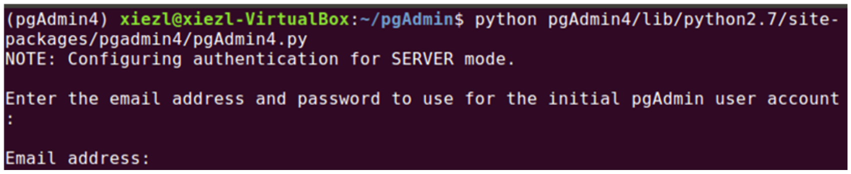
```
(pgAdmin4) xiezl@xiezl-VirtualBox:~/pgAdmin$ pip install ./pgadmin4-1.6-py2.py3-
none-any.whl
Processing ./pgadmin4-1.6-py2.py3-none-any.whl
Collecting Flask-Principal==0.4.0 (from pgadmin4==1.6)
  Downloading http://mirrors.aliyun.com/pypi/packages/14/c7/2531aca6ab7baa3774fd
e2dfc9c9dd6d5a42576a1013a93701bfdc402fdd/Flask-Principal-0.4.0.tar.gz
Collecting blinker==1.3 (from pgadmin4==1.6)
  Downloading http://mirrors.aliyun.com/pypi/packages/c9/66/c15dbe2e2cac59bf1d46
70d52aa88b8746fd5a47f8353aa4ffac0dde00c4/blinker-1.3.tar.gz (91kB)
    100% |████████████████████████████████| 92kB 109kB/s
Collecting extras==0.0.3 (from pgadmin4==1.6)
  Downloading http://mirrors.aliyun.com/pypi/packages/7f/b4/44b7a534c96e7cd8fe54
8265352a4f9ded54da06b9f5238864661f5e3bf1/extras-0.0.3.tar.gz
```

The success message:



```
Successfully installed Babel-2.3.4 Flask-0.11.1 Flask-Babel-0.11.1 Flask-Gravata
r-0.4.2 Flask-HTMLmin-1.2 Flask-Login-0.3.2 Flask-Mail-0.9.1 Flask-Migrate-2.0.3
 Flask-Principal-0.4.0 Flask-SQLAlchemy-2.1 Flask-Script-2.0.5 Flask-Security-1.
7.5 Flask-WTF-0.12 Jinja2-2.7.3 Mako-1.0.7 MarkupSafe-0.23 SQLAlchemy-1.0.14 WTF
orms-2.0.2 Werkzeug-0.9.6 alembic-0.9.5 backports.csv-1.0.5 beautifulsoup4-4.4.1
 blinker-1.3 click-6.6 extras-0.0.3 fixtures-2.0.0 html5lib-1.0b3 htmlmin-0.1.10
 importlib-1.0.4 itsdangerous-0.24 linecache2-1.0.0 passlib-1.6.2 pbr-1.9.1 pgad
min4-1.6 psycopg2-2.7.3.1 pycrypto-2.6.1 pyrsistent-0.11.13 python-dateutil-2.5.
0 python-editor-1.0.3 python-mimeparse-1.5.1 pytz-2014.10 simplejson-3.6.5 six-1
.10.0 speaklater-1.3 sqlparse-0.1.19 testtools-2.3.0 traceback2-1.4.0 unittest2-
1.1.0
```

v. Start the python file:

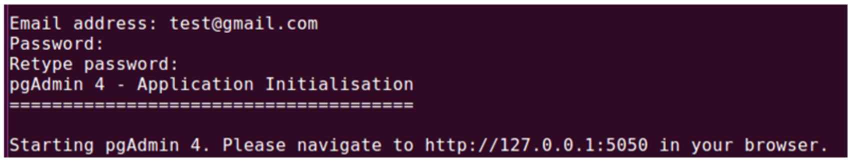


```
(pgAdmin4) xiezl@xiezl-VirtualBox:~/pgAdmin$ python pgAdmin4/lib/python2.7/site-
packages/pgadmin4/pgAdmin4.py
NOTE: Configuring authentication for SERVER mode.

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

Email address:
```

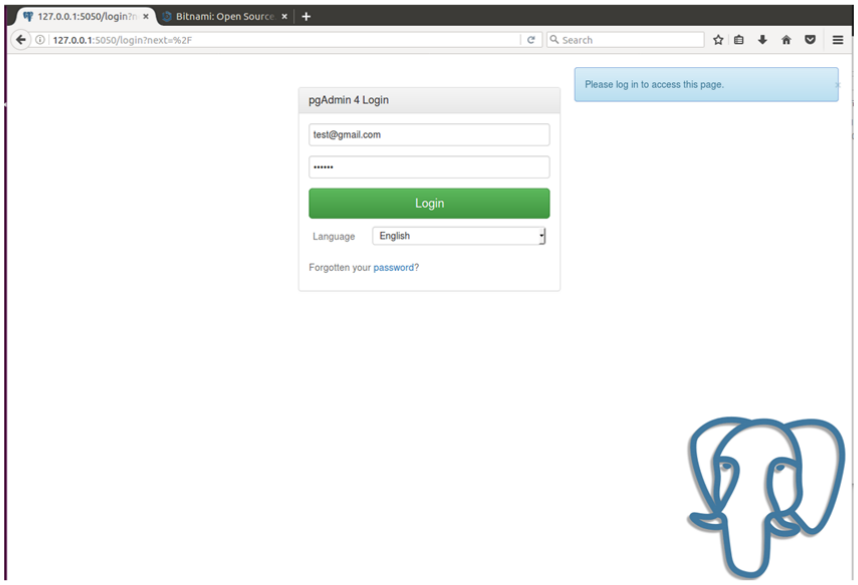
Input the login email address and login password:



```
Email address: test@gmail.com
Password:
Retype password:
pgAdmin 4 - Application Initialisation
======================================

Starting pgAdmin 4. Please navigate to http://127.0.0.1:5050 in your browser.
```

vi. Visit the pgAdmin application on your browser by http://127.0.0.1:5050 with the email address and password just created.

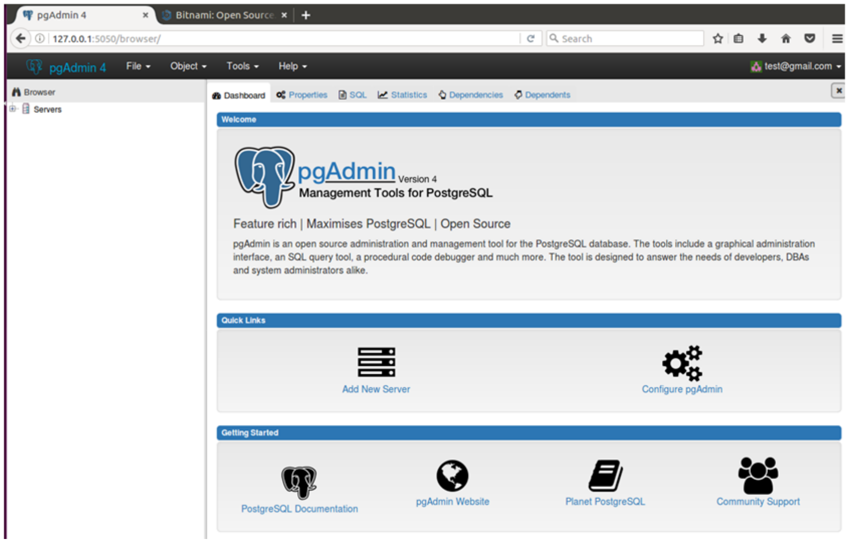


Please log in to access this page.

test@gmail.com

••••••

Success page:



Feature rich | Maximises PostgreSQL | Open Source

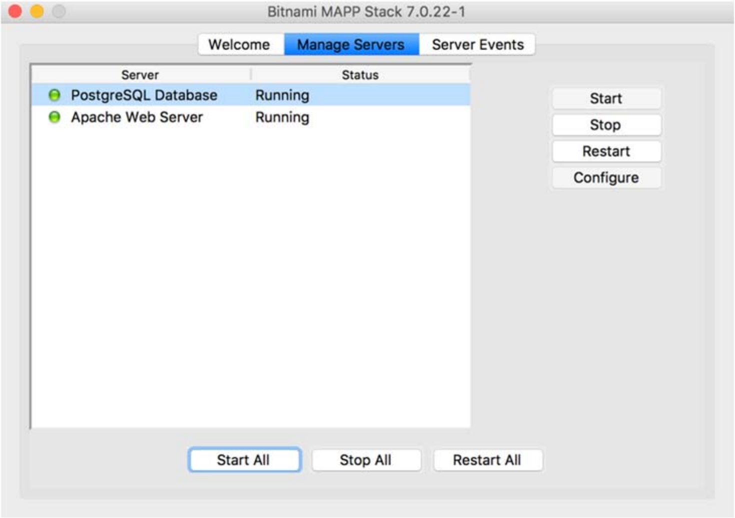
pgAdmin is an open source administration and management tool for the PostgreSQL database. The tools include a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.

vii. Done.

5. Test – Connect to database and create table via pgAdmin.

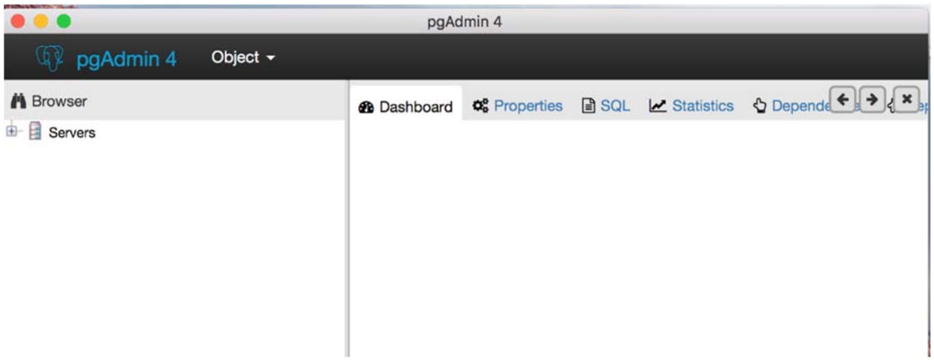
i. Open MAPP/WAPP/LAPP application (to auto start the Postgresql database server)

a) In case the server is not alive: Go to “Manage Servers”, Select “PostgreSQL Database” and Start it. Alternatively, you can just click “Start All” to start everything including the Apache server.

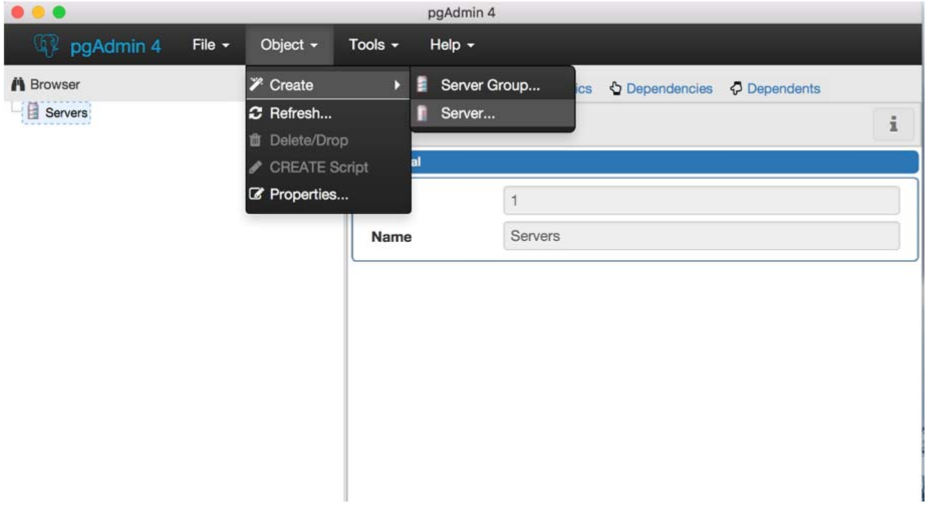


Server	Status
PostgreSQL Database	Running
Apache Web Server	Running

b) Open pgAdmin4



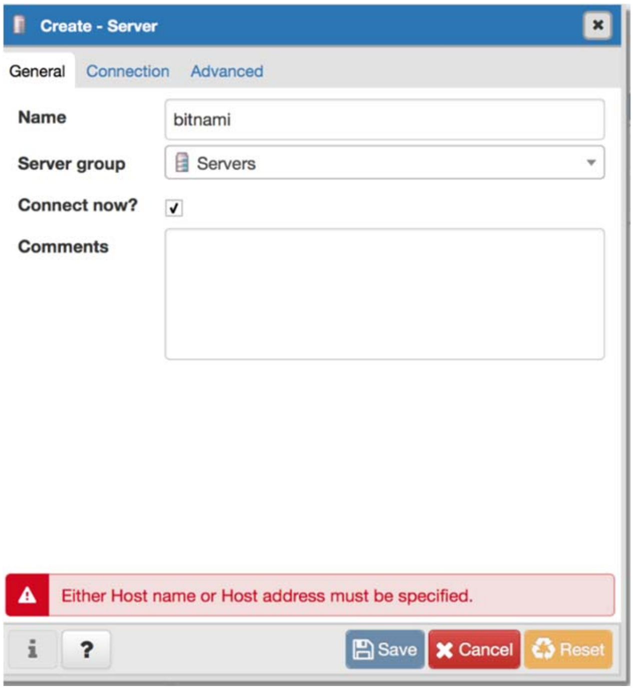
c) Click Object-Create-Server:



ID: 1

Name: Servers

d) Input the server name as “bitnami”:



Name: bitnami

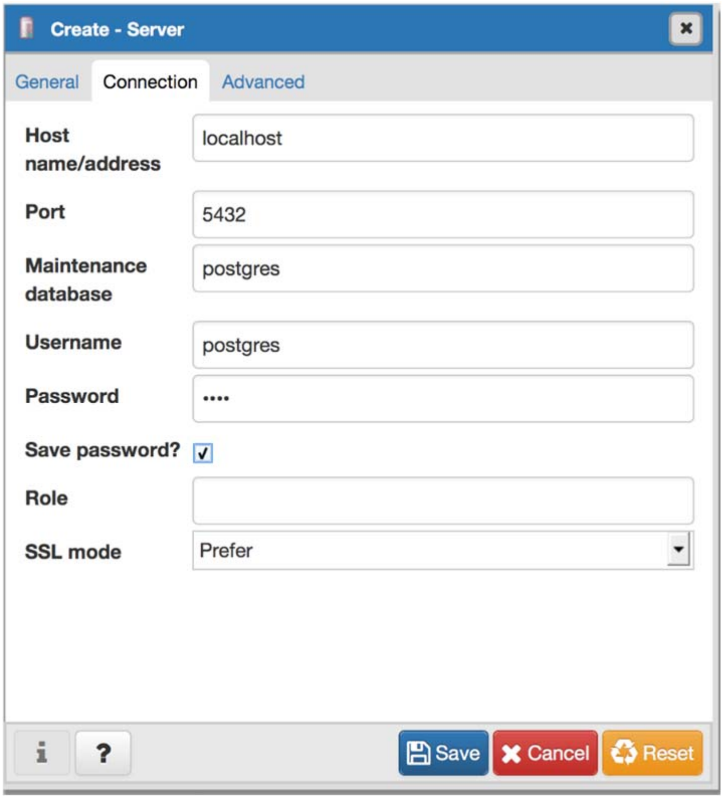
Server group: Servers

Connect now? ☑

Comments:

Either Host name or Host address must be specified.

e) Input host name and password under connection panel:



Host name/address: localhost

Port: 5432

Maintenance database: postgres

Username: postgres

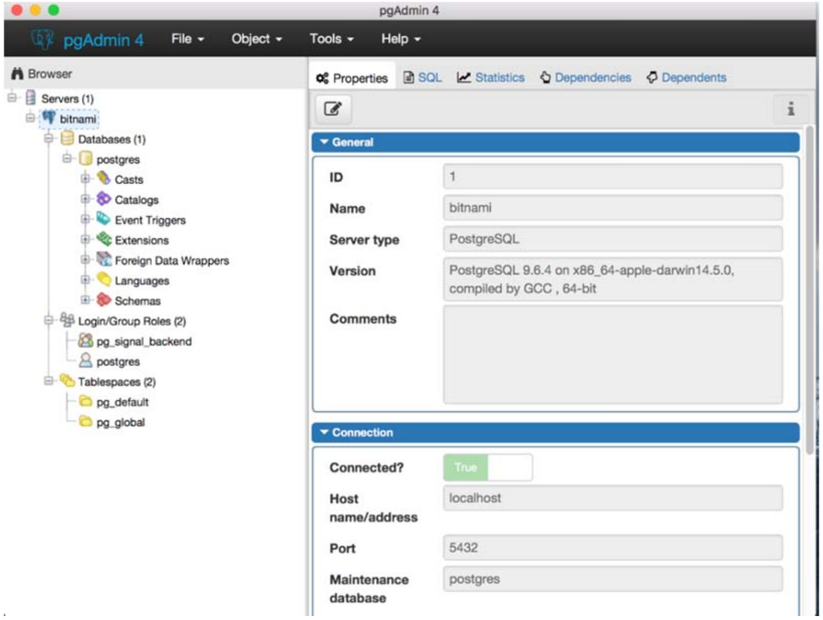
Password: ••••

Save password? ☑

Role:

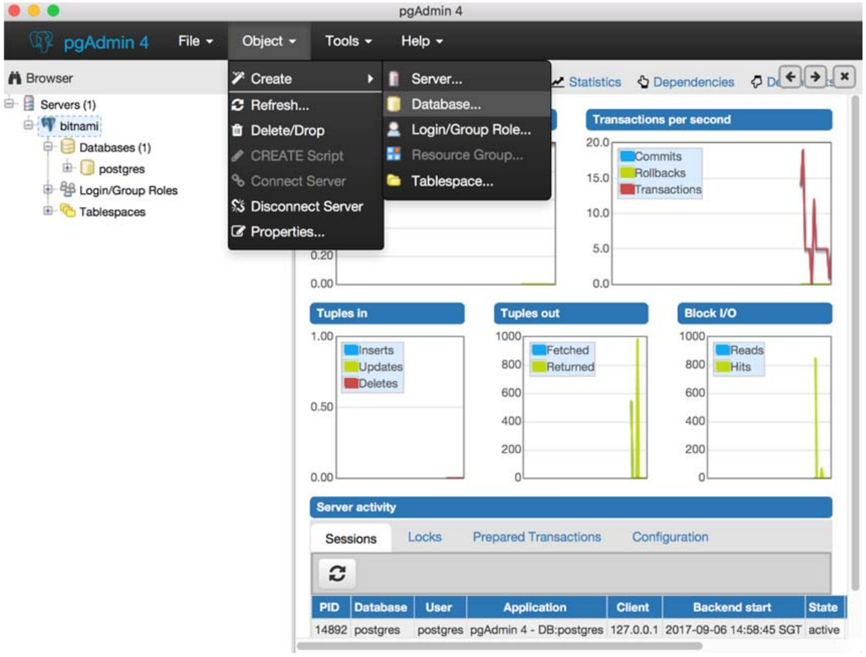
SSL mode: Prefer

f) Click save:



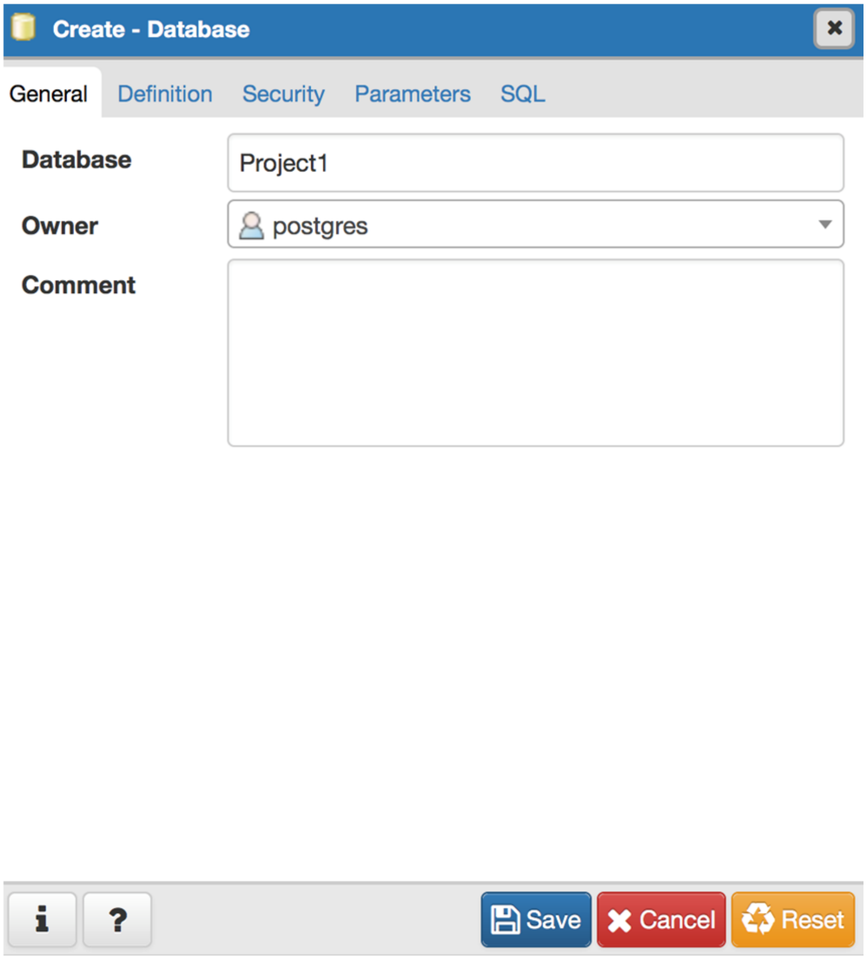
ID	1
Name	bitnami
Server type	PostgreSQL
Version	PostgreSQL 9.6.4 on x86_64-apple-darwin14.5.0, compiled by GCC , 64-bit
Comments	
Connected?	True
Host name/address	localhost
Port	5432
Maintenance database	postgres

g) Click on Object-Create-Database to create a database named “Project1”



PID	Database	User	Application	Client	Backend start	State
14892	postgres	postgres	pgAdmin 4 - DB:postgres	127.0.0.1	2017-09-06 14:58:45 SGT	active

Click Save:

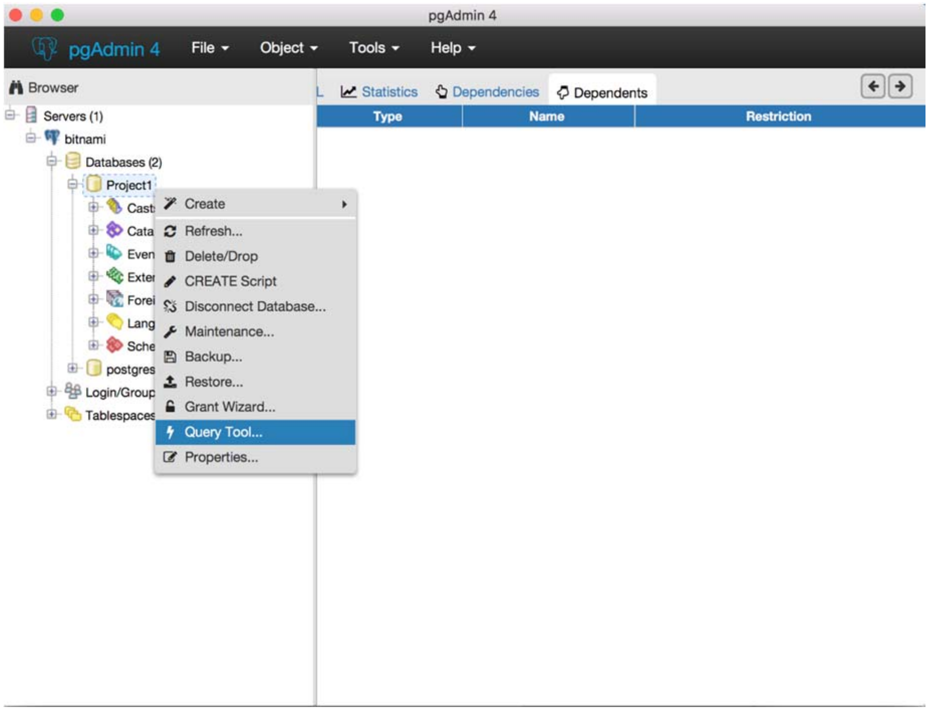


Database: Project1

Owner: postgres

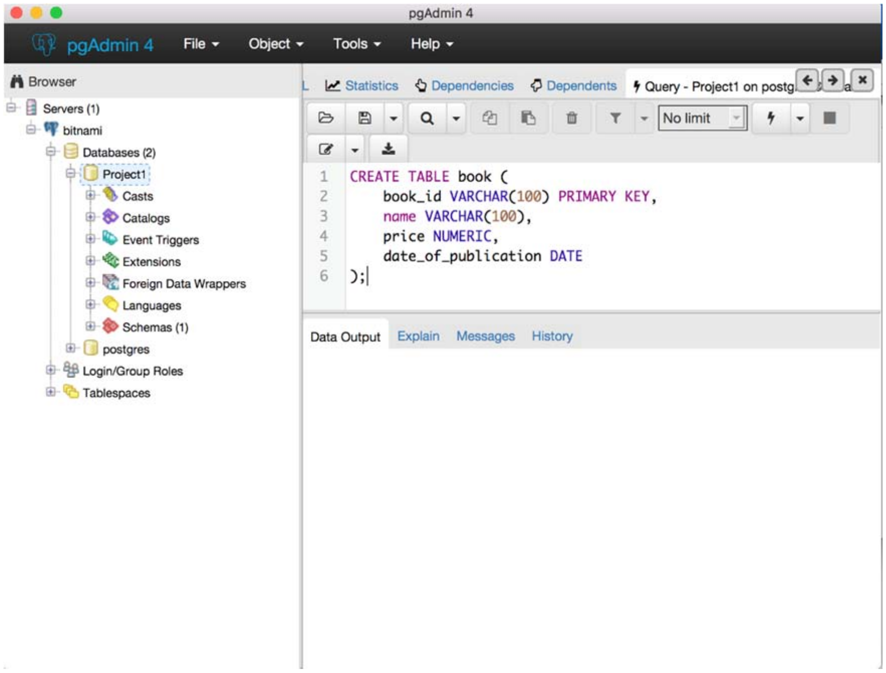
Comment:

h) Right click on Servers-bitnami-Databases-Project1, select Query Tool:



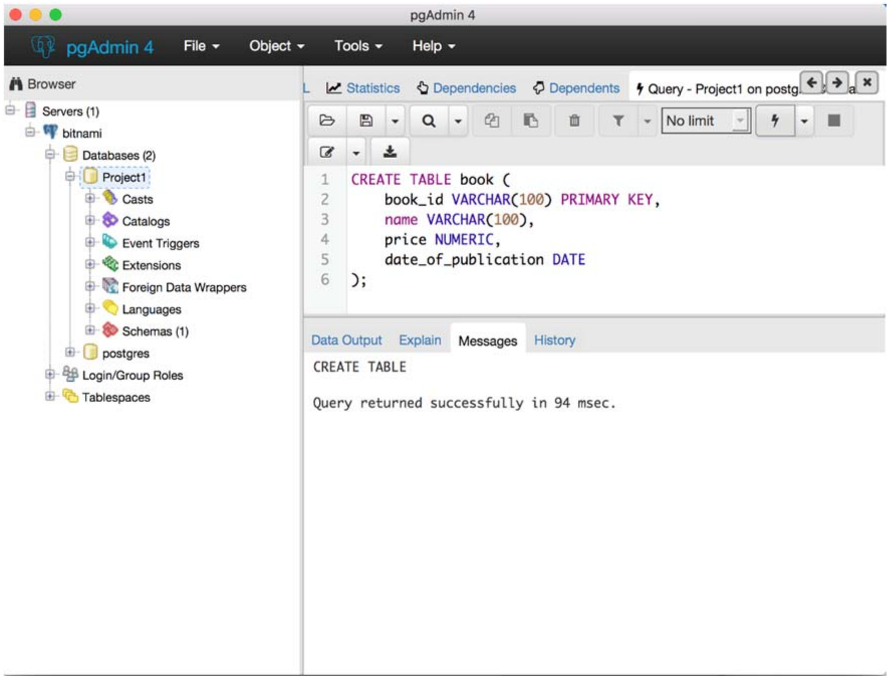
i) Input following commands to create a table named “book”:

```
CREATE TABLE book (
    book_id VARCHAR(100) PRIMARY KEY,
    name VARCHAR(100),
    price NUMERIC,
    date_of_publication DATE
);
```



```
CREATE TABLE book (
    book_id VARCHAR(100) PRIMARY KEY,
    name VARCHAR(100),
    price NUMERIC,
    date_of_publication DATE
);
```

j) Click “Execute” (The lightening button) and check the result:



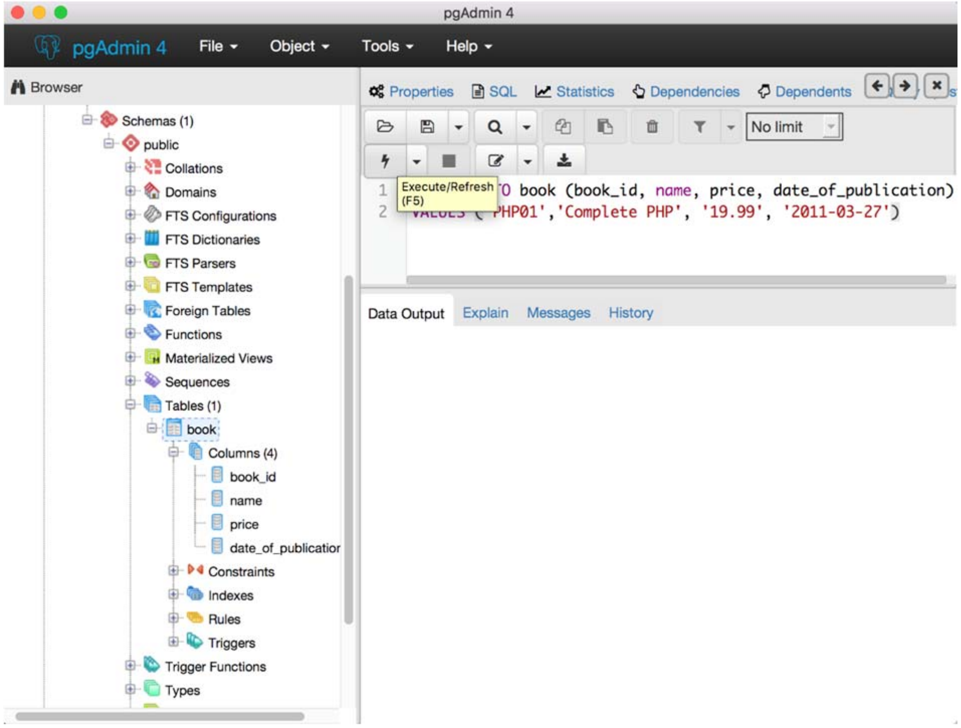
```
CREATE TABLE book (
    book_id VARCHAR(100) PRIMARY KEY,
    name VARCHAR(100),
    price NUMERIC,
    date_of_publication DATE
);
```

```
CREATE TABLE

Query returned successfully in 94 msec.
```

k) Input following Insert Command into SQL area then click execute:

```
INSERT INTO book (book_id, name, price, date_of_publication)
VALUES ('PHP01','Complete PHP', 19.99, '2011-03-27')
```

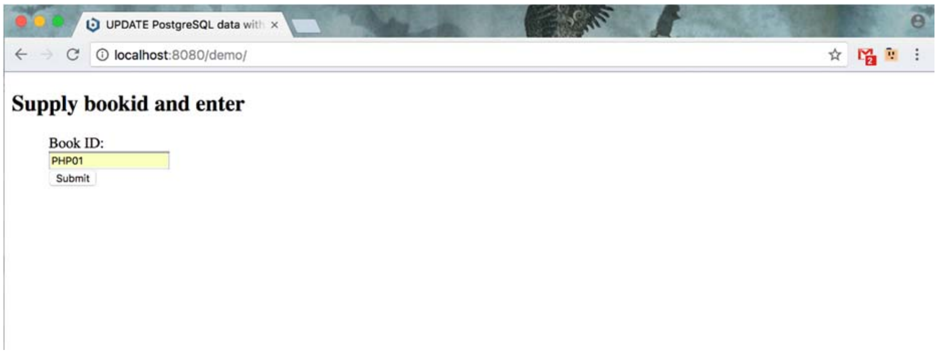


```
O book (book_id, name, price, date_of_publication)
PHP01','Complete PHP', '19.99', '2011-03-27')
```

l) You should see the insertion is successful from the prompted message.

6. Test – Query and update the "book" table via demo page:

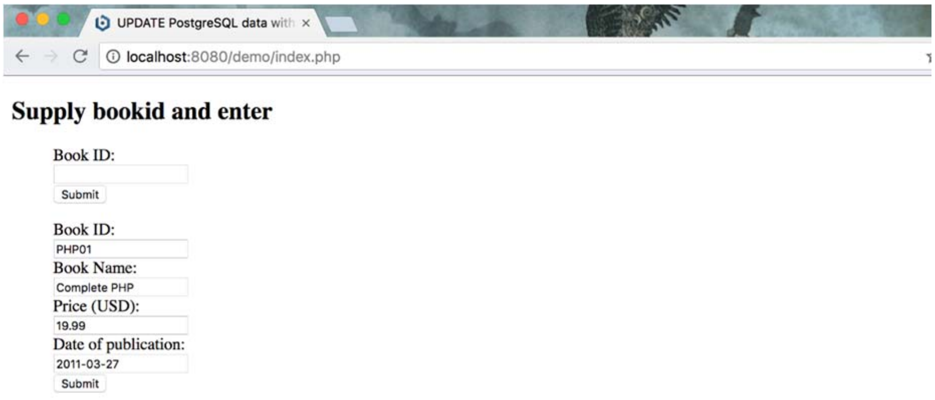
i. Visit the demo page, input primary key "PHP01" and click "submit":
Windows: http://localhost:80/demo
Linux/mac: http://localhost:8080/demo



Supply bookid and enter

Book ID:
PHP01
Submit

ii. Check whether the values are same as what you insert in previous test:

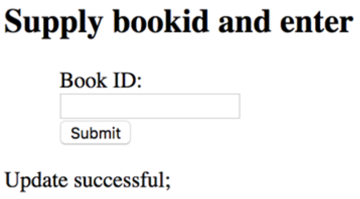


Supply bookid and enter

Book ID:
Submit

Book ID:
PHP01
Book Name:
Complete PHP
Price (USD):
19.99
Date of publication:
2011-03-27
Submit

iii. Change the value of price from 19.99 to 21.99 and click submit, you should see "update successful" page:

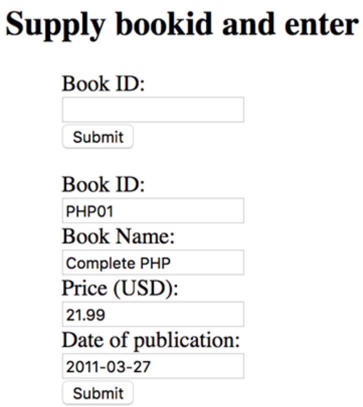


Supply bookid and enter

Book ID:
Submit

Update successful;

iv. Check again whether the value of price is changed:



Supply bookid and enter

Book ID:

Submit

Book ID:

PHP01

Book Name:

Complete PHP

Price (USD):

21.99

Date of publication:

2011-03-27

Submit

v. Done.

7. Explaination of the php Code

The code used in the demo is shown as following.

Note: you need to change the password (shown as "test" in the code) of the database account accordingly as mentioned in the comments.

```
<!DOCTYPE html>
<head>
  <title>UPDATE PostgreSQL data with PHP</title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <style>li {list-style: none;}</style>
</head>
<body>
  <h2>Supply bookid and enter</h2>
  <ul>
    <form name="display" action="index.php" method="POST" >
      <li>Book ID:</li>
      <li><input type="text" name="bookid" /></li>
      <li><input type="submit" name="submit" /></li>
    </form>
  </ul>
  <?php
        // Connect to the database. Please change the password in the
following line accordingly
    $db     = pg_connect("host=localhost port=5432 dbname=Project1
user=postgres password=test");
    $result = pg_query($db, "SELECT * FROM book where book_id =
'$_POST[bookid]'");             // Query template
```

```
    $row    = pg_fetch_assoc($result);          // To store the result row
    if (isset($_POST['submit'])) {
        echo "<ul><form name='update' action='index.php' method='POST' >
        <li>Book ID:</li>
        <li><input type='text' name='bookid_updated' value='$row[book_id]'
/></li>
        <li>Book Name:</li>
        <li><input type='text' name='book_name_updated' value='$row[name]'
/></li>
        <li>Price (USD):</li><li><input type='text' name='price_updated'
value='$row[price]' /></li>
        <li>Date of publication:</li>
        <li><input type='text' name='dop_updated'
value='$row[date_of_publication]' /></li>
        <li><input type='submit' name='new' /></li>
        </form>
        </ul>";
    }
    if (isset($_POST['new'])) { // Submit the update SQL command
        $result = pg_query($db, "UPDATE book SET book_id =
'$_POST[bookid_updated]',
    name = '$_POST[book_name_updated]',price = '$_POST[price_updated]',
    date_of_publication = '$_POST[dop_updated]'");
        if (!$result) {
            echo "Update failed!!";
        } else {
            echo "Update successful!";
        }
    }
    ?>
</body>
</html>
```