# Developing apps quickly with Azure Database for PostgreSQL

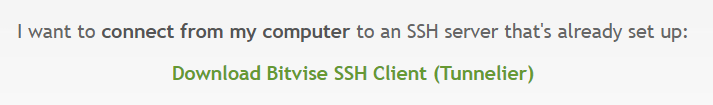
## Overview

Azure Database for PostgreSQL is a PostgreSQL database service built on Microsoft’s scalable cloud infrastructure for application developers. Leverage your existing open-source PostgreSQL skills and tools and, scale on-the-fly without downtime to efficiently deliver existing and new applications with reduced operational overhead. Built-in features maximize performance, availability, and security. Azure Database for PostgreSQL empowers developers to focus on application innovation instead of database management tasks.

## Pre-requisite for this code-challenge

1. Valid Azure Subscription
2. Install bitvise client on your local machines

<https://www.bitvise.com/download-area>



### Scenario Overview

This hands-on lab (code-challenge) will step you through the following:

Note- **For this lab, you need a valid Azure subscription**

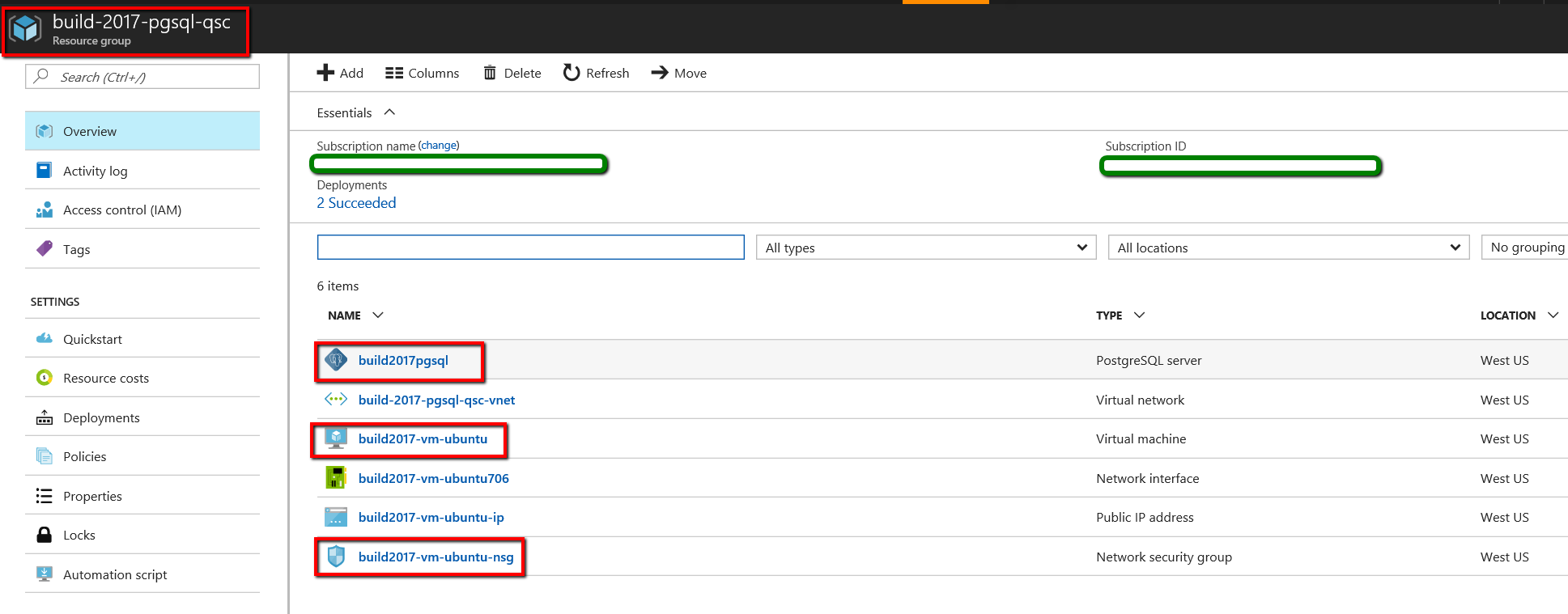
* Create an Ubuntu Azure VM
* Enable firewall
* Create database bootcamp through pgadmin in Azure PostgreSQL database instance
* Login into to ubuntu VM
* Download app
  + git clone <https://github.com/vitorfs/bootcamp.git>
  + cd bootcamp
* Install pre-reqs
  + pip install -U -r requirements.txt
* Change your connection string
  + vim .env
  + Paste this in the .env file (change database credentials to yours)
    - DEBUG=True
    - SECRET\_KEY='mys3cr3tk3y'
    - DATABASE\_URL='postgres://pgdemouser@pgdemo:<passwd>@pgdemo.database.windows.net:5432/bootcamp'
* Run migration and serve the app
  + Python manage.py migrate
  + Python manage.py runserver 0.0.0.0:8000
* Go to chrome/firefox and go to http://<appidaddress>
* Play around with the app, go to PgAdmin to show that it works

### About the code challenge

Wish to learn how to use the new Azure Database for PostgreSQL to build apps rapidly? This exercise will walk you through the steps to deploy a Python/Djano app and connect to PostgreSQL database service. Part of the lab, you will experience the capabilities of PostgreSQL database service using PostgreSQL tools.

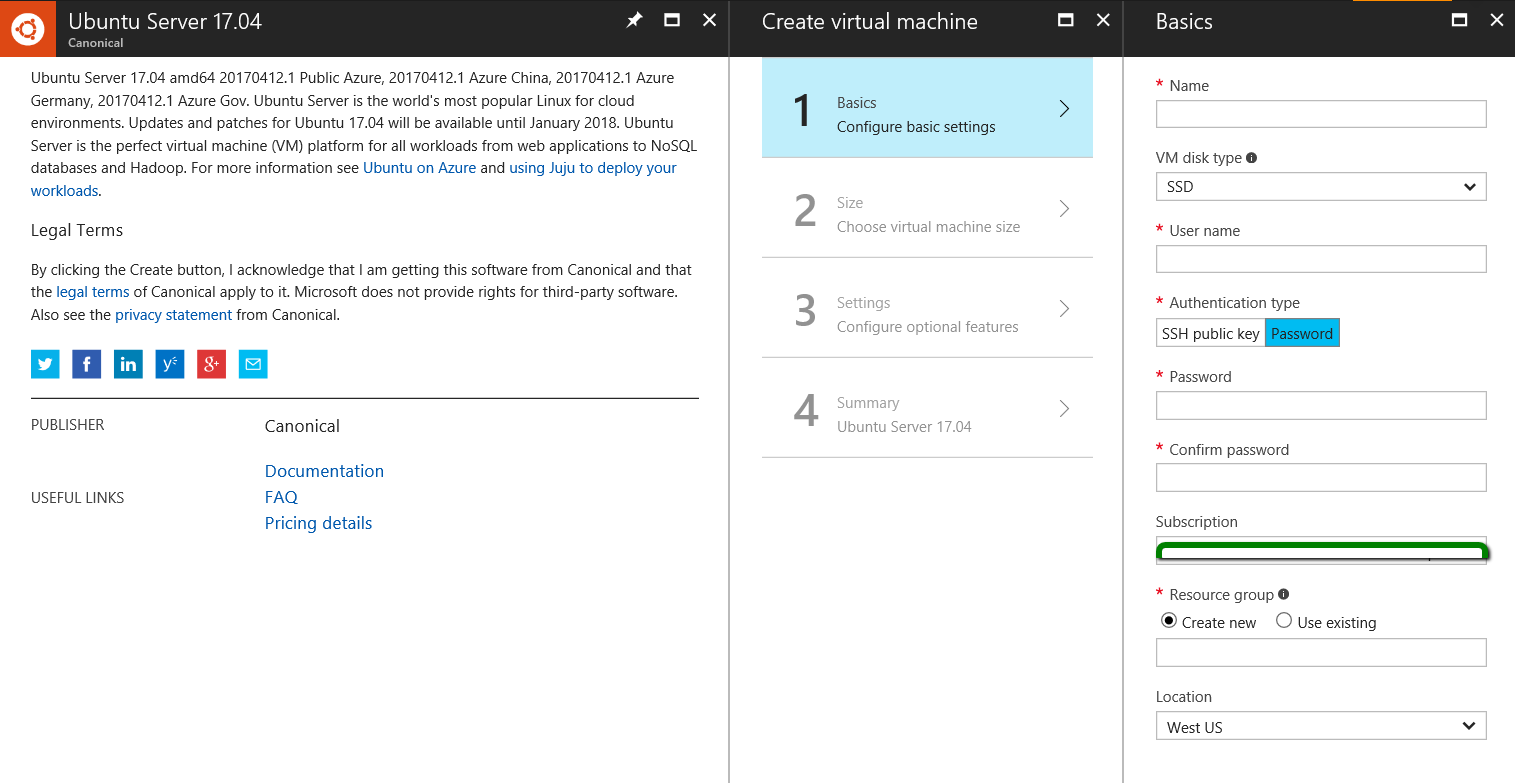
In this lab, we are cloning a Python app running on Django framework and PostgreSQL named bootcamp from GitHub, and then by just flipping a connection string to “PostgreSQL as a Service in Azure” making the app run in Azure.

**Note-** Here's what it will look like when you have created all the services in Azure needed for this lab. Here’s how it looks from my subscription.

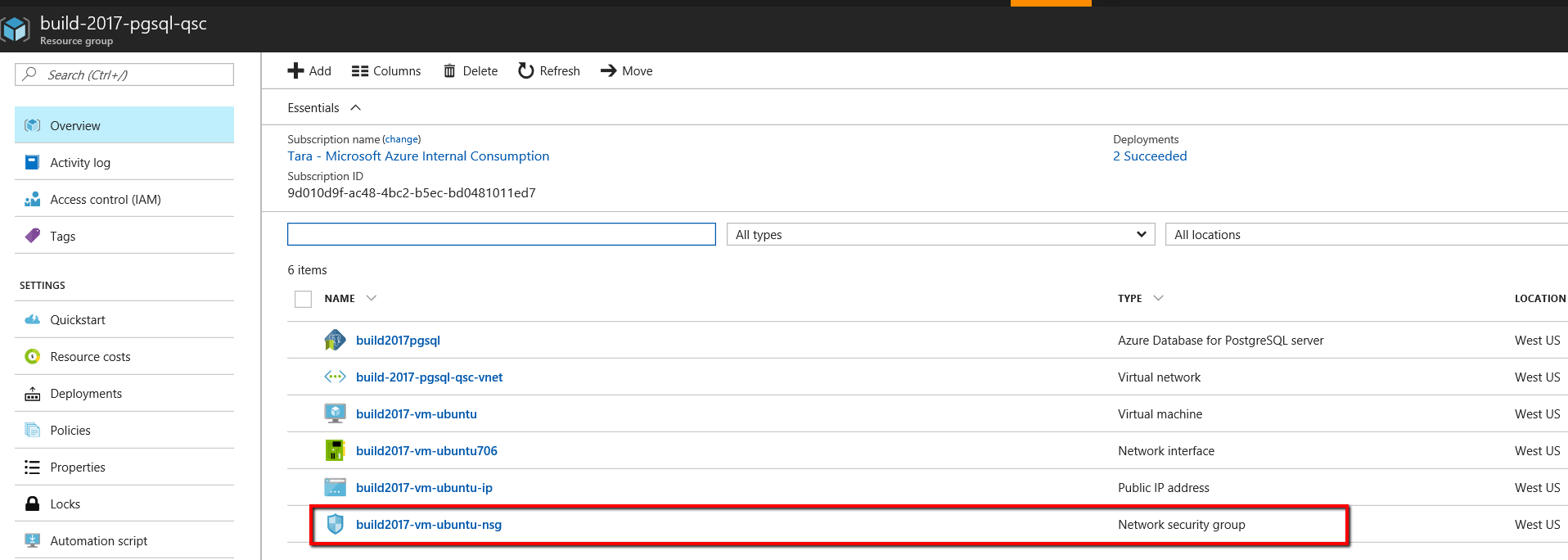


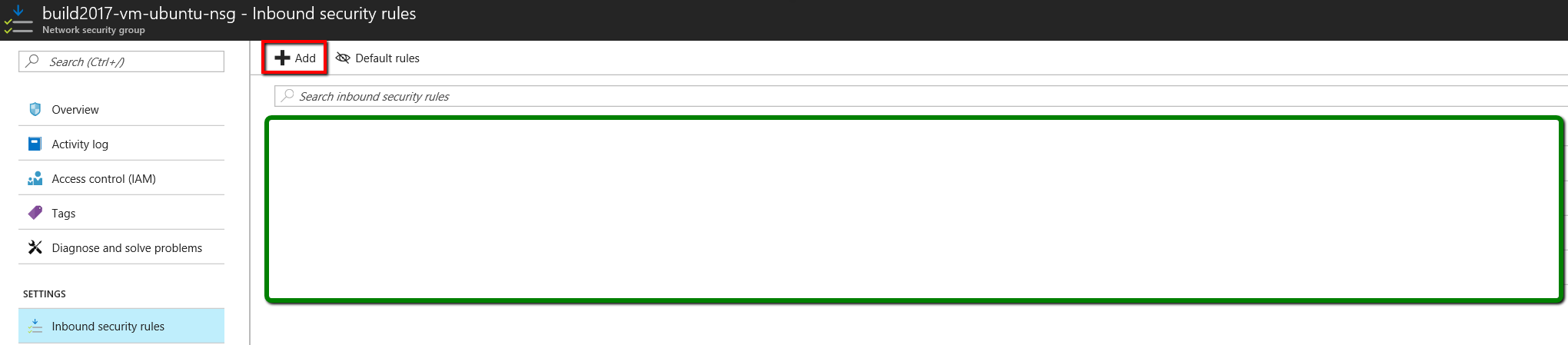
### Start of the Lab

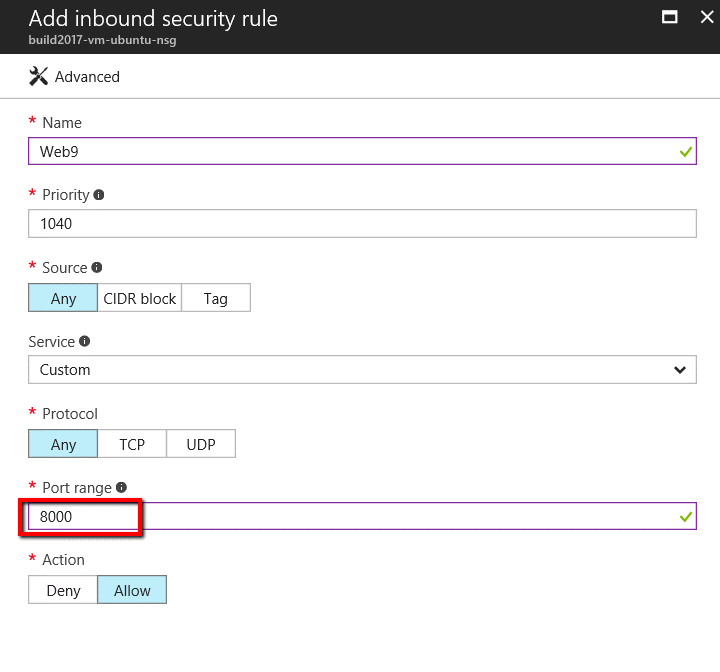
1. Given the fact you have an Azure credential, please create an Ubuntu VM from the VM Gallery, and take a note of the host IP Address, username and password. The VM provisioning will ask for a resource group, please provide a name.



1. Go to the NSG you created in step one (this is created by default)- and add port 8000 to the Inbound security rules







1. Click on Bitvise Client

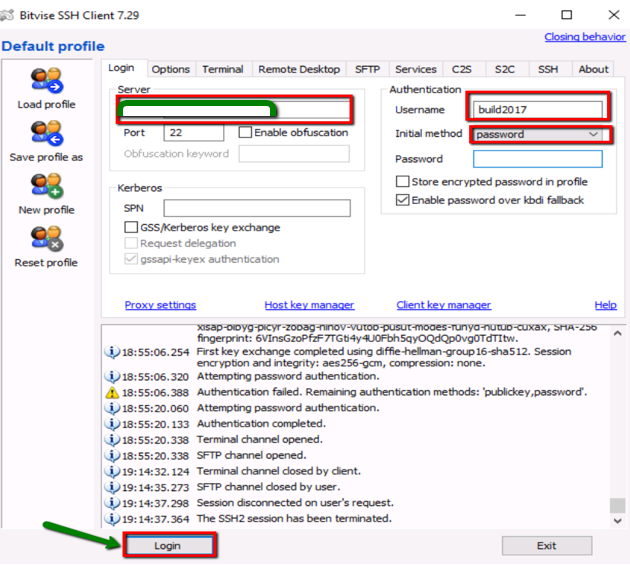


1. This will open up a window to help you logon to Ubuntu VM, enter the details as mentioned here, and hit Login

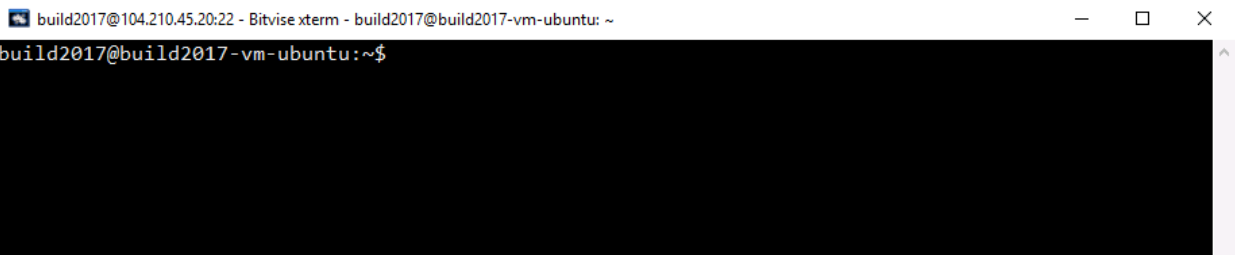
**Host- 104.xxx.xx.xx**

**Username- “you provided while provisioning the Ubuntu VM in Azure”**

**Password- “you provided while provisioning the Ubuntu VM in Azure”**

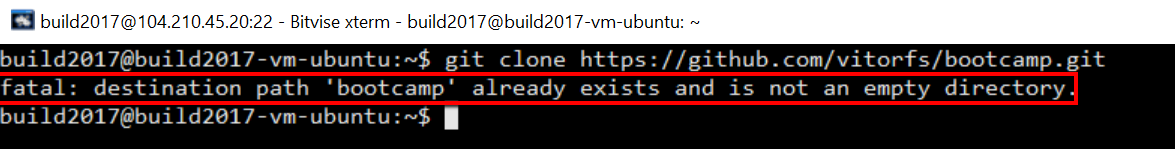


1. **This action should open a new window with the terminal to Ubuntu VM**



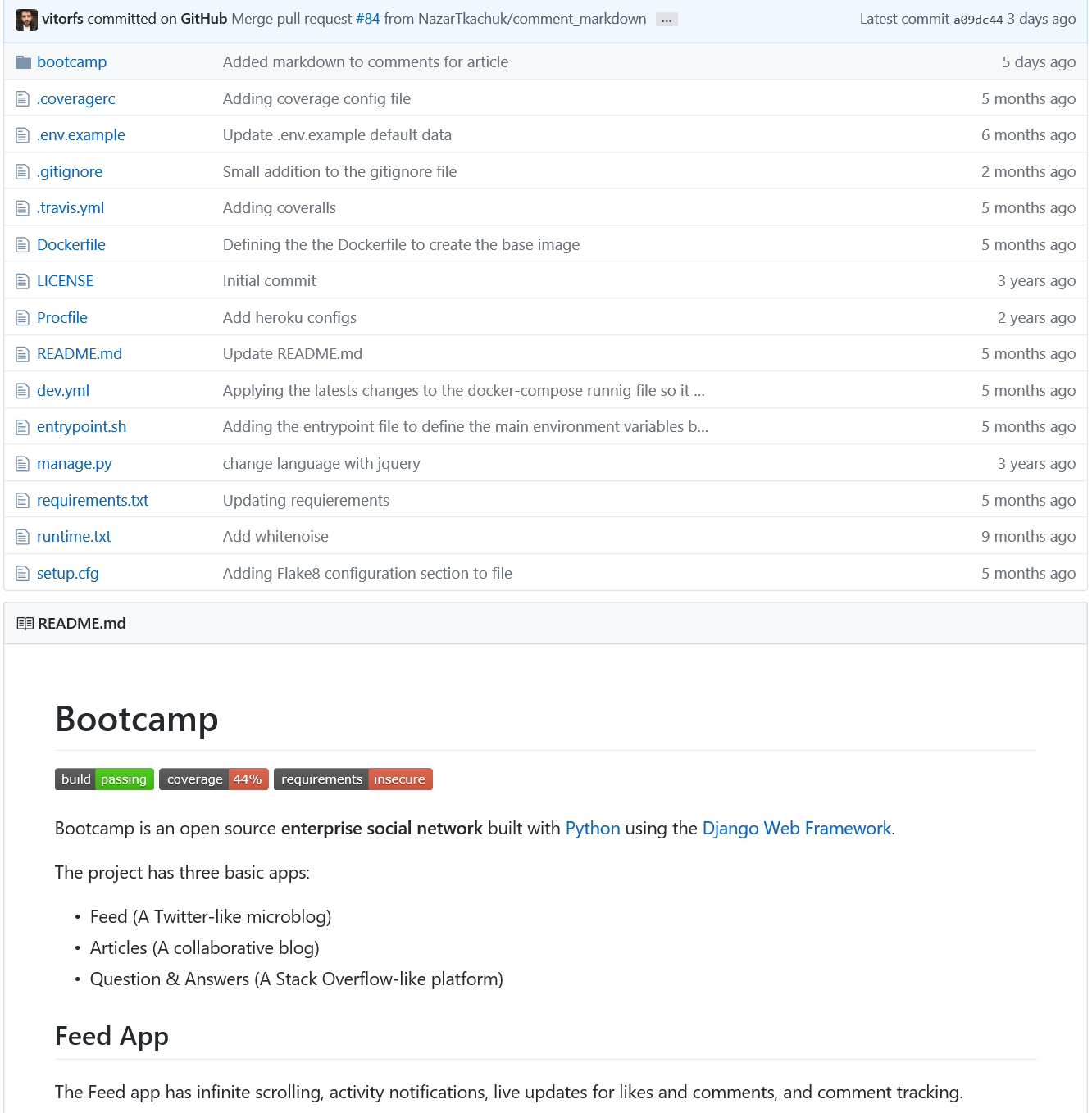
1. **We have already gitcloned the app, though you can always re-run the command, and you would see that the destination path exists**

git clone <https://github.com/vitorfs/bootcamp.git>



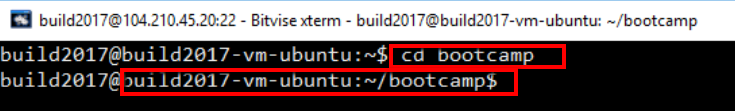
**Here’s a look at the GitHub Repo- You can go to the browser and have a look at the repo if you like (optional Step)**

[**https://github.com/vitorfs/bootcamp.git**](https://github.com/vitorfs/bootcamp.git)



1. **In the same terminal run**-

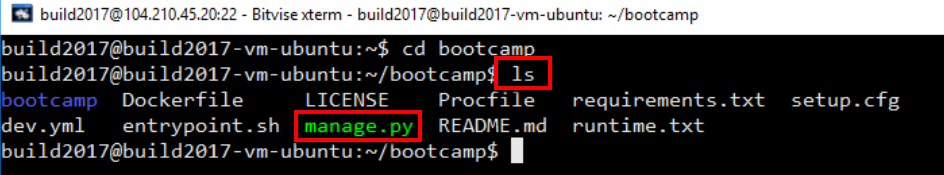
**cd bootcamp**



Change your connection string (this is a read-only operation we have already modified the connection string, to reduce any friction and updates on the same VM)

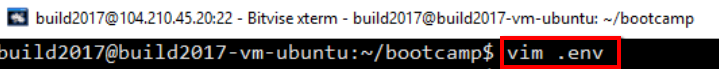
1. **Run ls (It’s LS in smaller case) to list the files, note the manage.py. That’s our installed script from the repo clone.**

**ls**

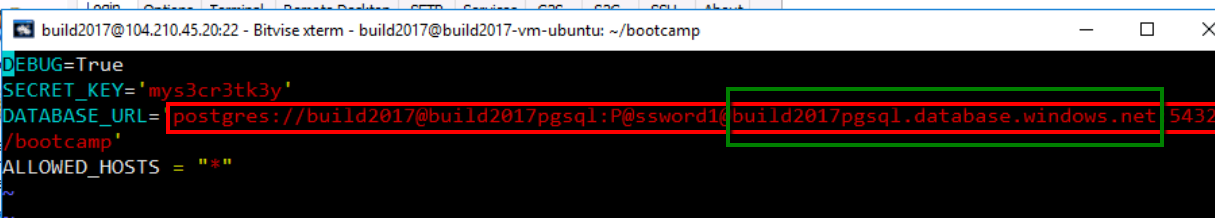


1. **Run the following command from the terminal (remaining in the folder bootcamp)**

**vim .env**



1. **The vim .env will show you the following screen**



**Note-** Check the credentials and if you see closely, we are using the flashy new Azure Database for PostgreSQL service to connect to this app. We have pre-populated the connection string.

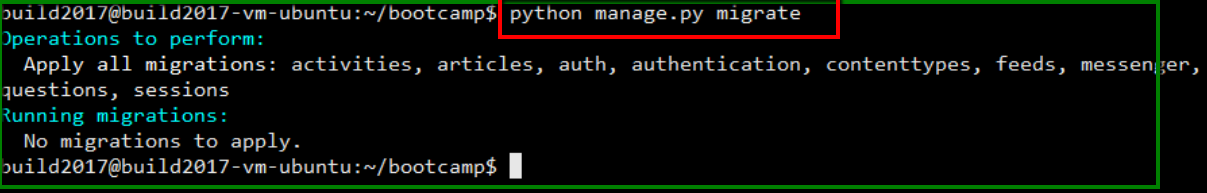
1. **Careful here- Please do not edit the .env file. On the Terminal, to come out of the editing mode-**

* Press **ESC** key
* Write “**:q**”

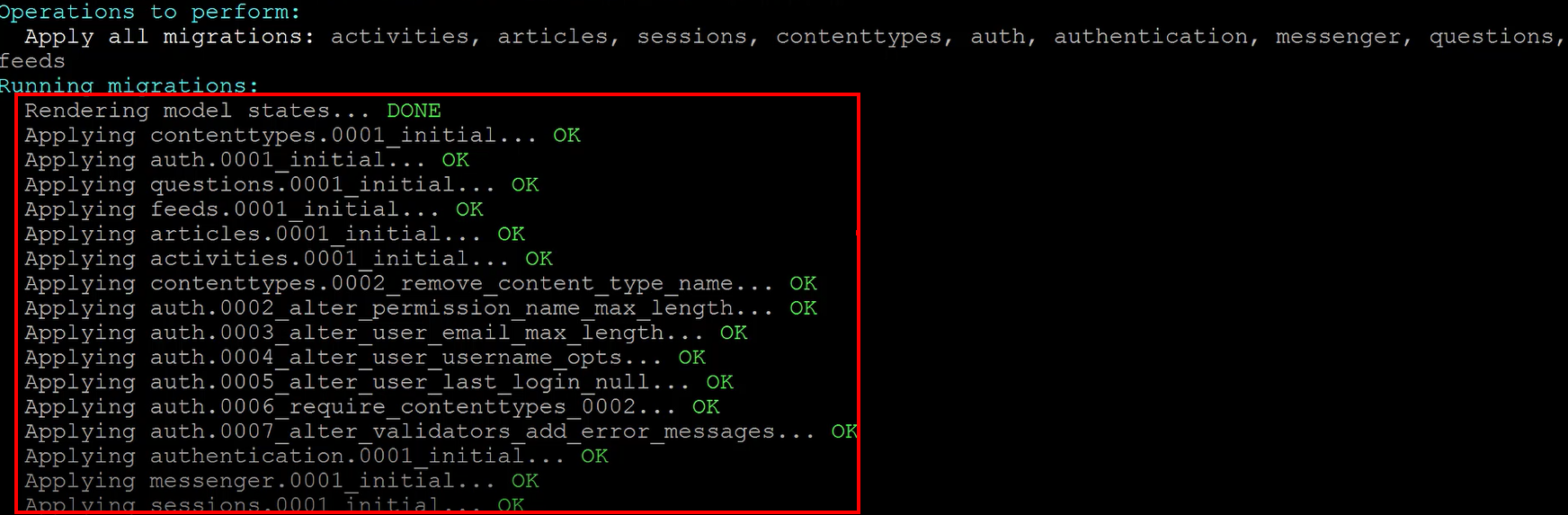


1. **Run migration and serve the app. Run the following command**

**python manage.py migrate**



***Example****-Here’s what it would have looked if we would have run this command for the first time. Though, as mentioned we have already run the manage.py migrate to keep moving on with the lab*

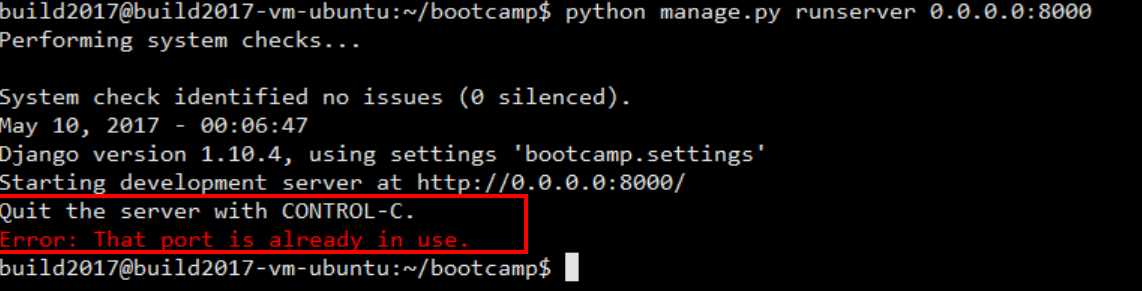


1. **Let’s get the server up- Run the following command**

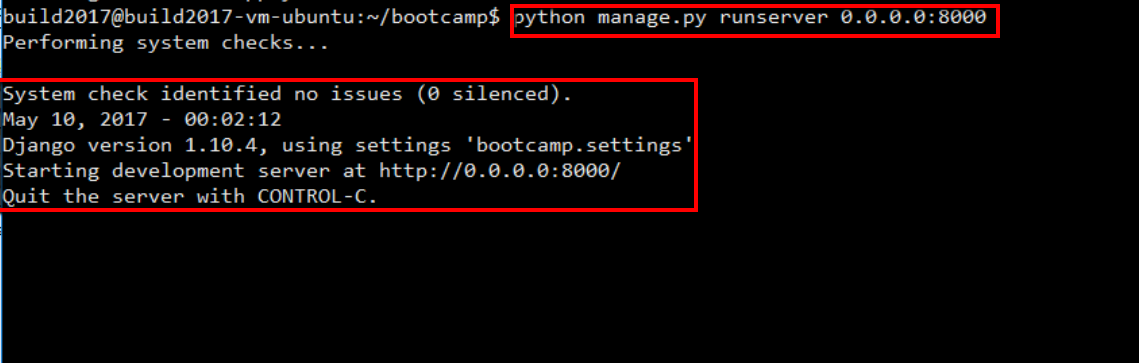
**Python manage.py runserver 0.0.0.0:8000**

**Note-** The port numbers from **8000 to 65536** are open on the network security group inbound rules of the Ubuntu VM. Please choose one port of your choice. Ranges of 8000 is good, though feel free to use any port number.

If you have chosen a port number already in use while other attendees at BUILD are using, you would get the following error- **“The port is already in use”- Please use another port.**



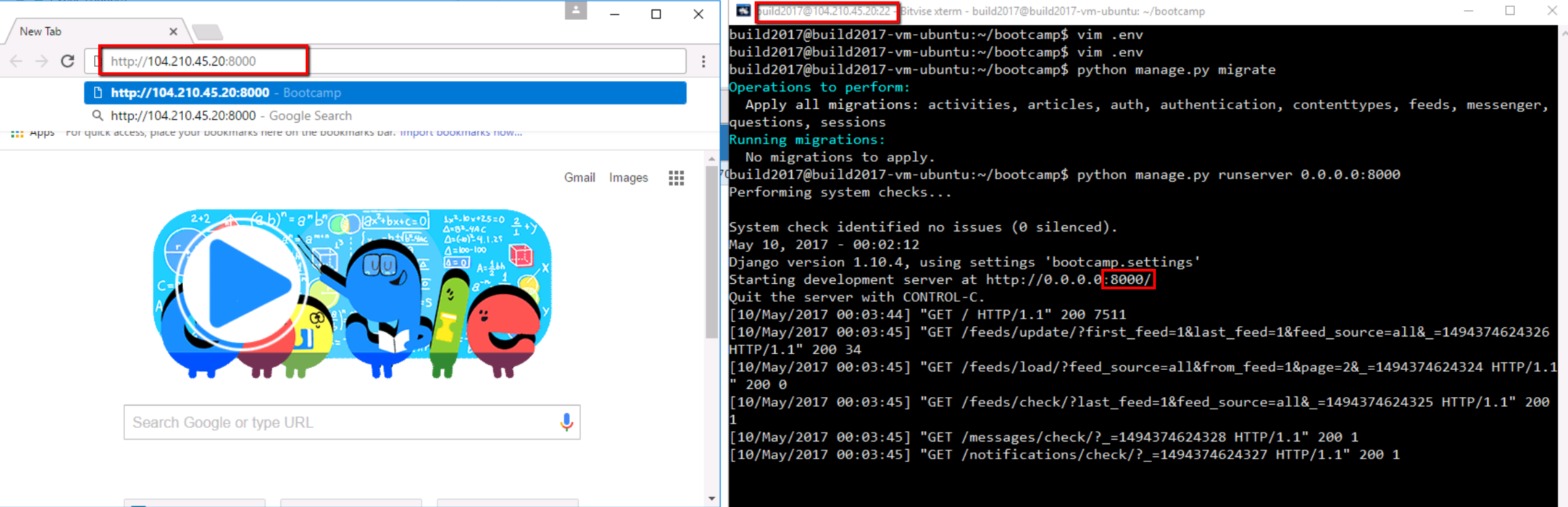
**Note- So if you got a good port the application should run fine, and should look like this**

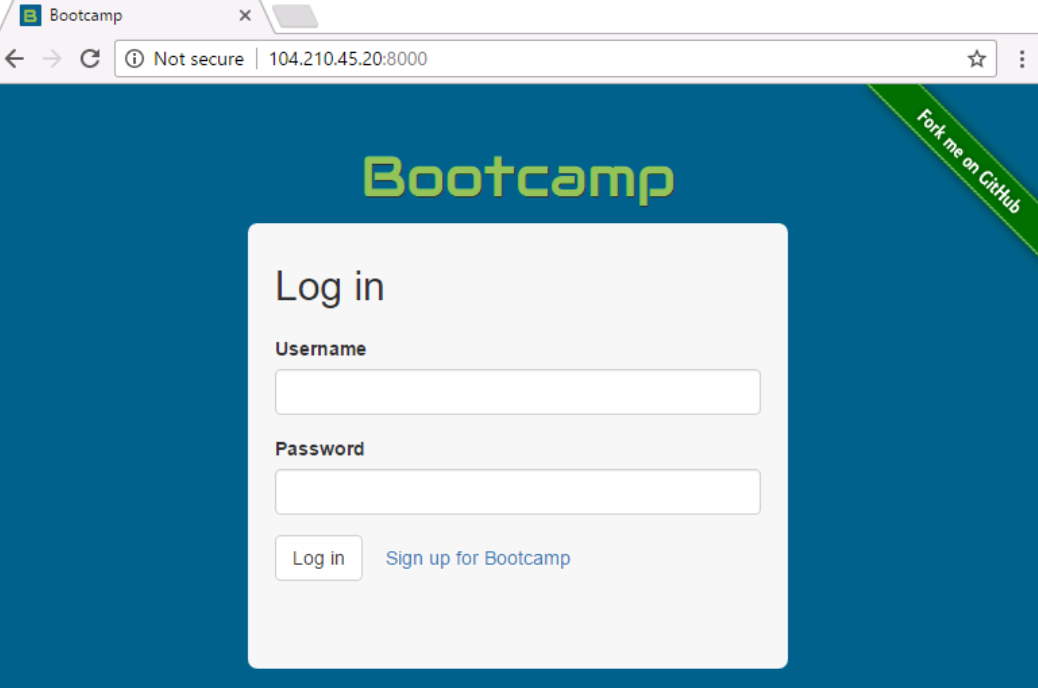


1. **Now go to chrome/firefox on the Windows 10 VM, and open the following URL**

[**http://104.210.45.20:8000/**](http://104.210.45.20:8000/)

**Replace the Port with the port number you used for running the server app in the previous step. If all goes well, you should now see the Bootcamp app.**



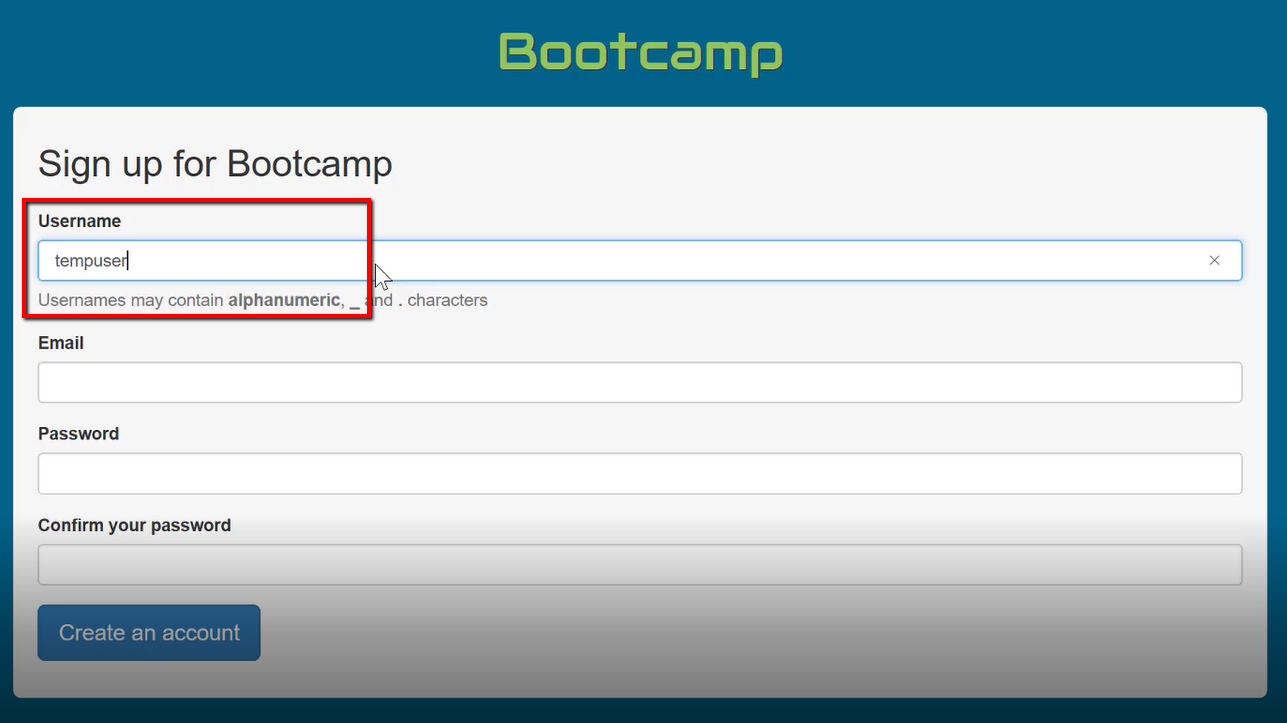


1. Let’s play around with the app. On the bootcamp app, please sign up for the bootcamp.

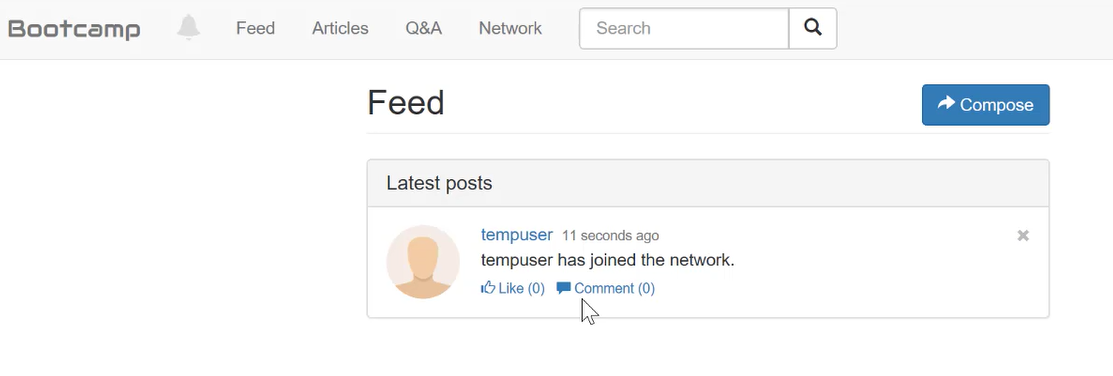


1. **Create any username. Easier- tempuser with your initials and date – “tempuser0511tj” something like this;**

**Provide your personal email address, and create a password, and then “Create an Account”**



**Note-** The next screen you would see after successful creation of the account is this:



1. **Let’s go to PgAdmin to show that how everything is tied up in the backend, with Azure and Azure Database for PostgreSQL.**

* **Go to PgAdmin 4.0**



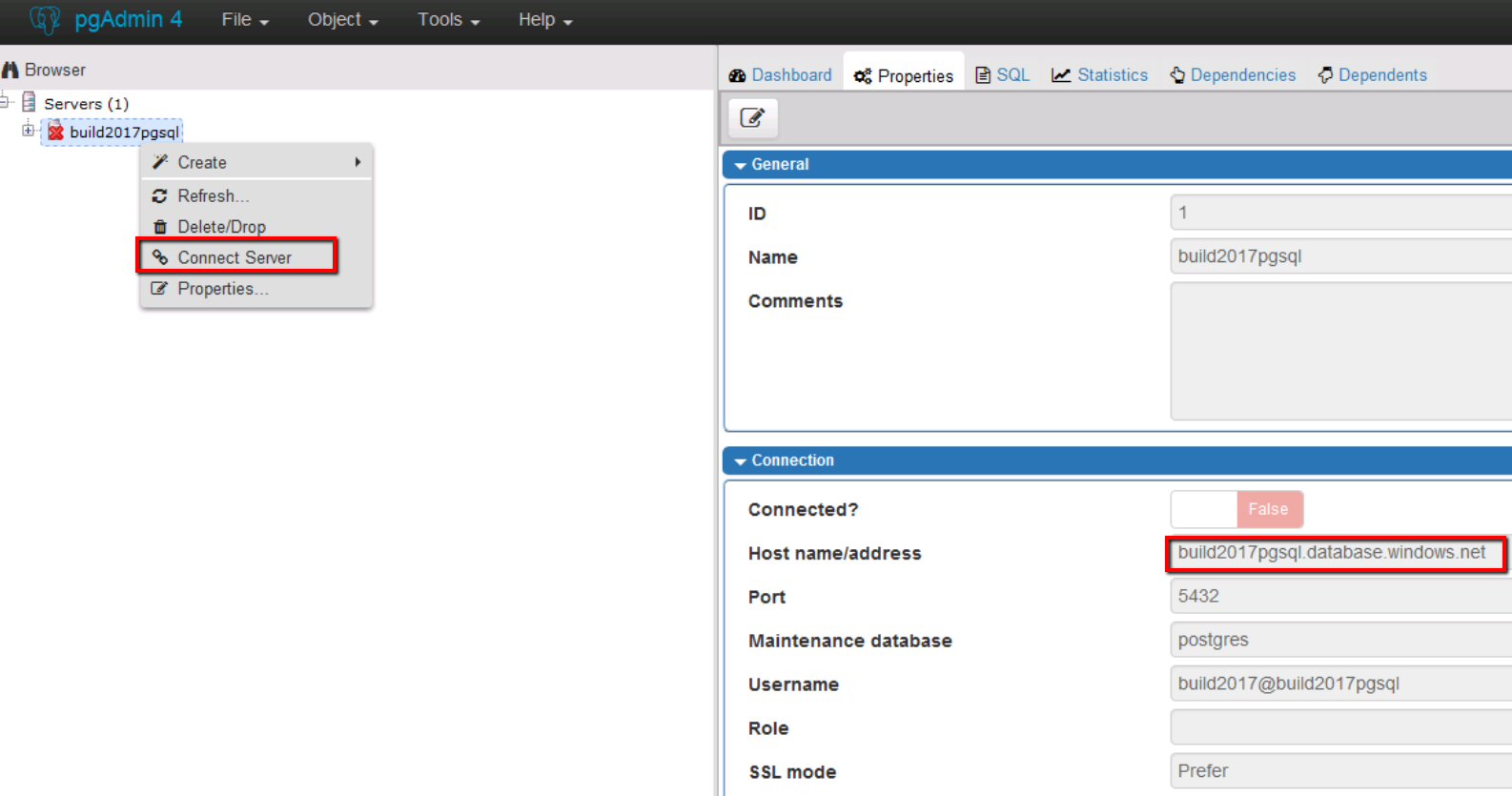
* **This will take a bit of a moment to open up**



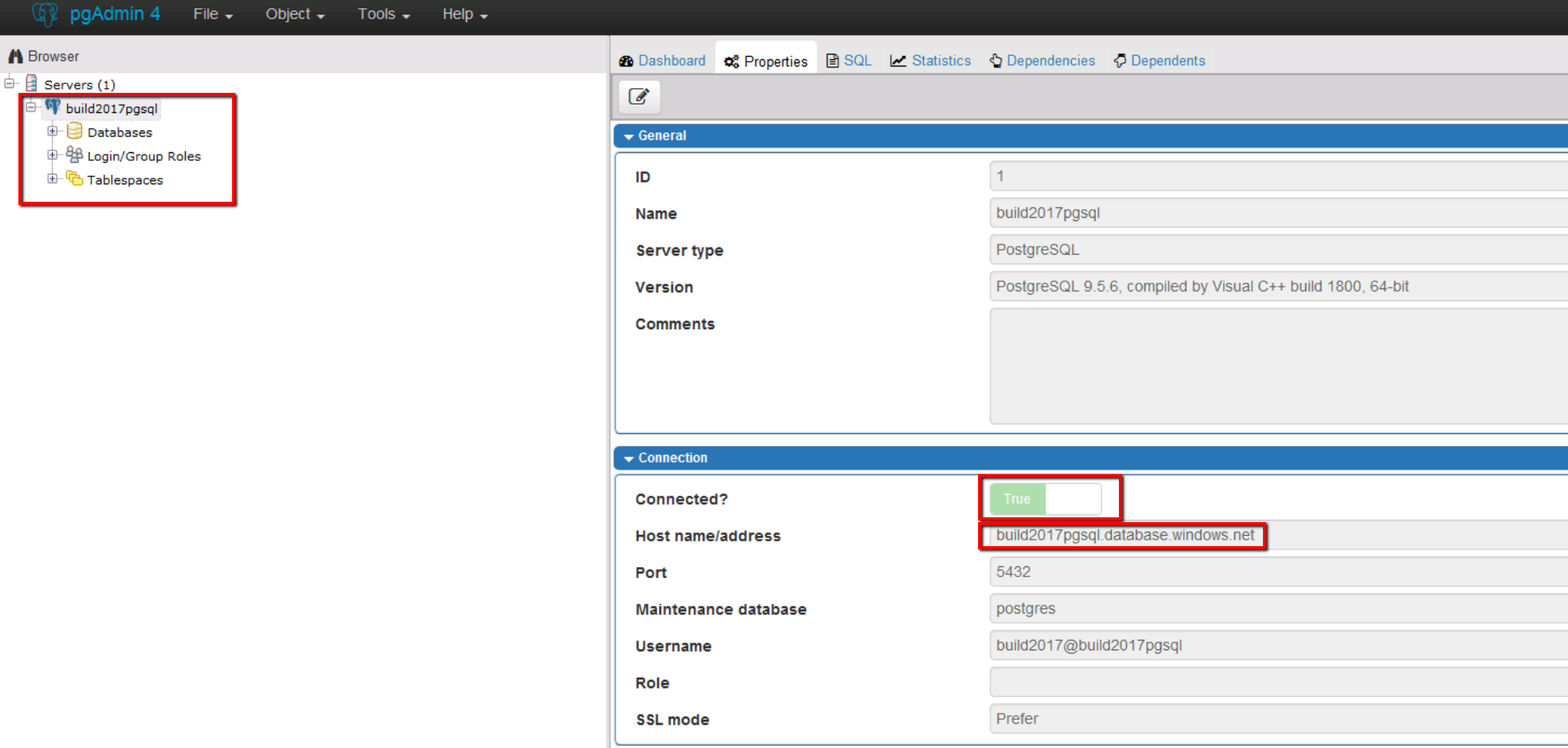
* **Once it is up, you would see the following**



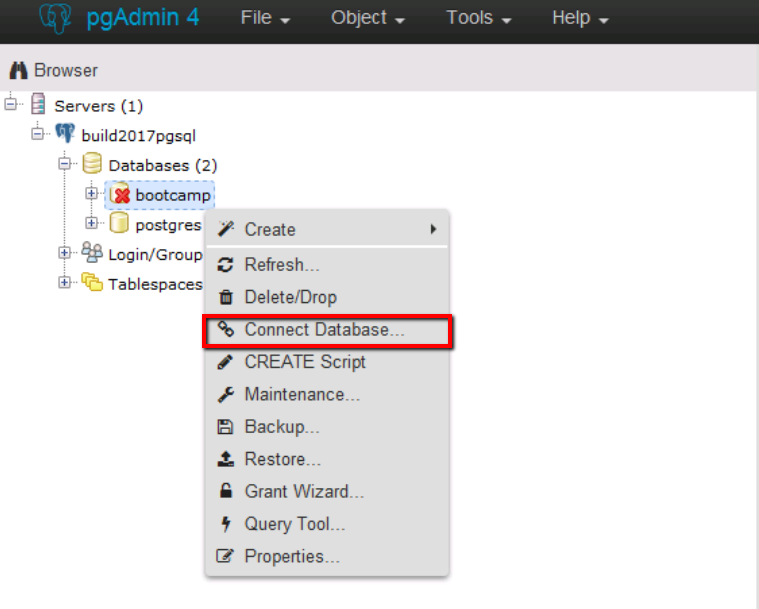
* **Right-Click Connect Server, and keep a note of the hostname/address. You would see it’s pointing to the flashy new “Azure Database for PostgreSQL in the cloud”**



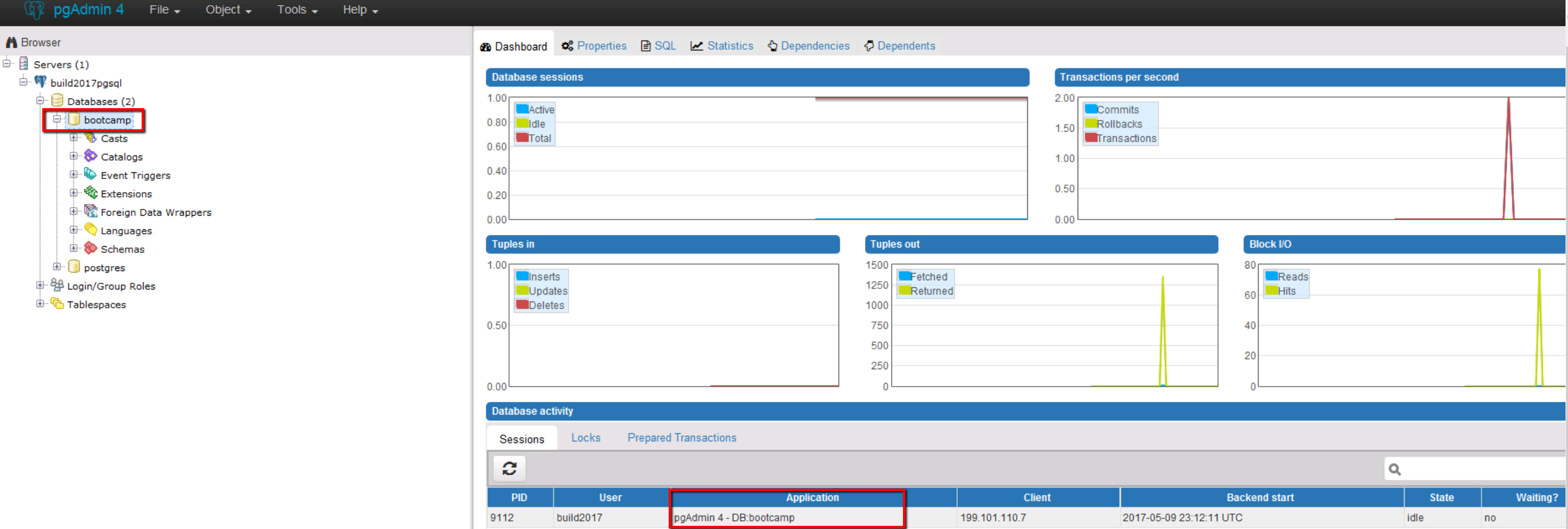
* **If all goes well, you will see the connection changing to True, and the server now shows the databases, and other artifacts of PostgreSQL instance in Azure (as a managed service)**



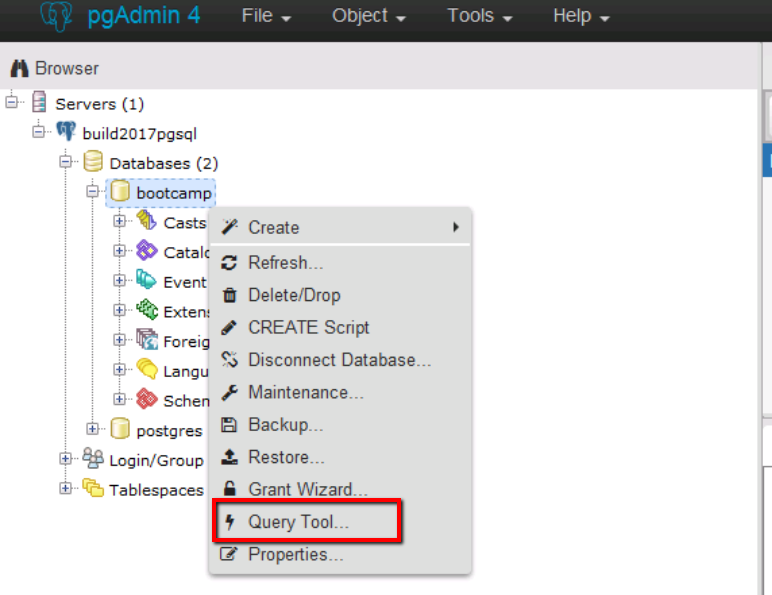
* **Expand Databases tab, and then click “Connect Database”- Bootcamp**



* **Now if the connection to the database succeeds- here’s what you should see. Just to mention again, the service is already pre-hosted in Azure, and the database is created.**

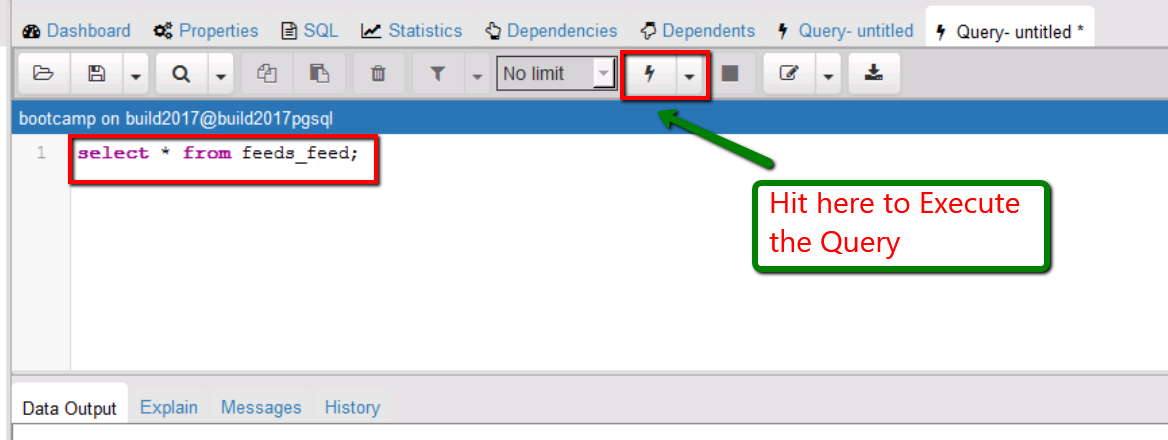


* **Right Click- Database “Bootcamp” and select “Query Tool”**

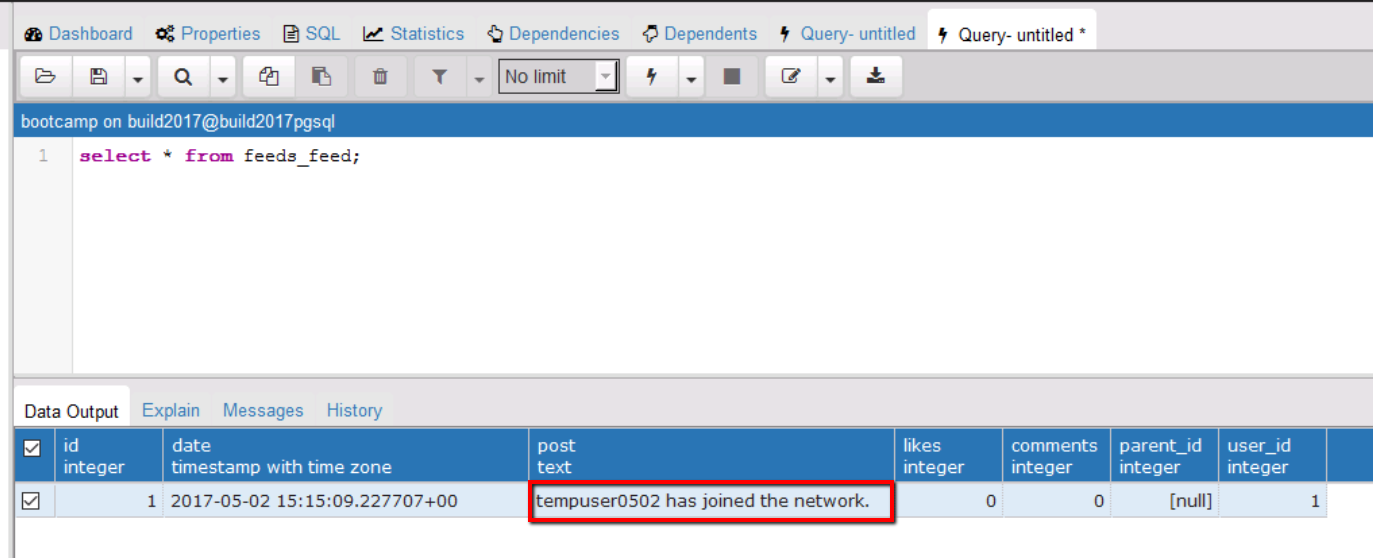


* **Run the following query**

**select \* from feeds\_feed;**



* **You should now see your username and in this table; there are chances that you may see more than one user, as many may be doing the lab**



**Conclusion-** In a short while, we git cloned and app running on python and Django framework, over PostgreSQL backend, from Github. Changed the connection string to a provisioned PostgreSQL Service in Azure, and got the app working against PostgreSQL in Azure. Post this, you can use any feature of PostgreSQL you can name of and that will work, PostGIS, or pgScale, etc.