

I'm pretty sure many of you have successfully completed the Python on ACG course with lots of practice and lots of love. I'm confident it wouldn't hurt to re-quote some important parts of the Python course from the perspective of a student like me. I guess there are crucial places for cloud engineering, and it would not be a waste of time to repeat it and it would be useful to focus on it again.

Because Keith's videos are from 5 years ago, some versions are not compatible with each other. Although it updated the db\_setup.sh file 3 years ago, I have seen in my experience that there is a conflict between PostgreSQL versions and Linux servers. If you want to see the image that Keith SQL is processing on your own computer now, you could fulfill this desire by using the commands I have given below.

First, let's start our Azure Sandbox from the Cloud Playground page[[link](#)].

## Cloud Playground

Learn by doing in our real, live practice environments.

[CLOUD SANDBOXES](#) [INSTANT TERMINAL](#) [CLOUD SERVERS](#)



### Azure Sandbox

Access a Microsoft Azure sandbox, so you can learn by doing and cloud along with

Username

Password

URL

Auto Shutdown **12:21 am**  
The sandbox will be shut down and cleaned at this time.

[Open Sandbox](#)

[Delete Sandbox](#)

Remember to Right Click and "Open Link in incognito Window"  
or your browser's sandboxed mode

You could then use AWS and Google Cloud for practice.

## Cloud Playground

[Watch "What are Cloud Servers?"](#)

Learn by doing in our real, live practice environments.

[CLOUD SANDBOXES](#) [INSTANT TERMINAL](#) [CLOUD SERVERS](#)

There's nothing here!

[+ New Server](#)

After Open Sandbox, we first create a server for PostgreSQL. We choose Distribution CentOS 7 w/ Docker.

## Cloud Playground

[Watch "What are Cloud Servers?"](#)

Learn by doing in our real, live practice environments.

[CLOUD SANDBOXES](#) [INSTANT TERMINAL](#) [CLOUD SERVERS](#)

Units available: 8 Zone: **Australia**

[Learn more](#)

Distribution  
CentOS 7 w/ Docker

Tags  
PostgreSQL

Expires  
Aug 02, 2022

Units  
1

Ready

Quick Actions

[Open Terminal](#)

[+ New Server](#)

Open the terminal of the server we created and enter the following commands. And you will need to update your temp\_password. This password is for user sudo.

```
d394227ea41c login: cloud_user
Password:
You are required to change your password immediately (root enforced)
Changing password for cloud_user.
(current) UNIX password:
New password:
Retype new password:
Last failed login: Tue Jul 19 14:24:15 UTC 2022 from localhost on pts/0
There was 1 failed login attempt since the last successful login.
Last login: Tue Jun 14 20:18:37 on
[cloud_user@d394227ea41c ~]$ ls
Desktop  my-login.pp
[cloud_user@d394227ea41c ~]$
```

After reaching the terminal page, we will install the database on this server. The installation process continues as follows:

**download database ->>**

```
curl -o db_setup.sh https://raw.githubusercontent.com/linuxacademy/content-python3-sysadmin/master/helpers/db_setup.sh
```

```
[cloud_user@d394227ea41c ~]$
[cloud_user@d394227ea41c ~]$ curl -o db_setup.sh https://raw.githubusercontent.com/linuxacademy/content-python3-sysadmin/master/helpers/db_setup.sh
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 159k 100 159k 0 0 158k 0 0:00:01 0:00:01 --:--:-- 158k
[cloud_user@d394227ea41c ~]$ ls
db_setup.sh Desktop my-login.pp
[cloud_user@d394227ea41c ~]$
```

**then ->>**

```
chmod +x db_setup.sh
```

```
./db_setup.sh
```

```
100 159k 100 159k 0 0 158k 0 0:00:01 0:00:01 --:--:-- 158k
[cloud_user@d394227ea41c ~]$ ls
db_setup.sh Desktop my-login.pp
[cloud_user@d394227ea41c ~]$ chmod +x db_setup.sh
[cloud_user@d394227ea41c ~]$ ./db_setup.sh
Updating and installing Docker
[sudo] password for cloud_user:
```

Entering your sudo password then the db\_setup.sh file will automatically make the necessary settings for us. [Here is a good example of automation](#). As a cloud engineer, you will be able to create files in this way.

This may take some time.

```
[cloud_user@d394227ea41c ~]$ chmod +x db_setup.sh
[cloud_user@d394227ea41c ~]$ ./db_setup.sh
Updating and installing Docker
[sudo] password for cloud_user:
sudo: timed out reading password
[sudo] password for cloud_user:
Loaded plugins: fastestmirror
Determining fastest mirrors
epel/x86_64/metalink | 4.3 kB 00:00:00
 * base: download.cf.centos.org
 * epel: epel.mirror.digitalpacific.com.au
 * extras: download.cf.centos.org
 * nux-dextop: mirror.li.nux.ro
 * updates: download.cf.centos.org
base | 3.6 kB 00:00:00
epel | 4.7 kB 00:00:00
extras | 2.9 kB 00:00:00
nux-dextop | 2.9 kB 00:00:00
updates | 2.9 kB 00:00:00
xrdp | 2.9 kB 00:00:00
(1/9): epel/x86_64/updateinfo | 1.1 MB 00:00:00
(2/9): epel/x86_64/group_gz | 96 kB 00:00:00
(3/9): base/7/x86_64/group_gz | 153 kB 00:00:00
(4/9): updates/7/x86_64/primary_db | 16 MB 00:00:00
(5/9): extras/7/x86_64/primary_db | 247 kB 00:00:00
(6/9): base/7/x86_64/primary_db | 6.1 MB 00:00:01
(7/9): epel/x86_64/primary_db | 7.0 MB 00:00:01
(8/9): nux-dextop/x86_64/primary_db | 1.8 MB 00:00:02
(9/9): xrdp/primary_db | 1.8 MB 00:00:06
```

Still going on

```
(17/23): samba-common-4.10.16-19.el7_9.noarch.rpm | 216 kB 00:00:01
(18/23): samba-common-libs-4.10.16-19.el7_9.x86_64.rpm | 183 kB 00:00:01
(19/23): samba-libs-4.10.16-19.el7_9.x86_64.rpm | 271 kB 00:00:00
(20/23): samba-common-tools-4.10.16-19.el7_9.x86_64.rpm | 467 kB 00:00:00
(21/23): xz-5.2.2-2.el7_9.x86_64.rpm | 229 kB 00:00:00
(22/23): xz-libs-5.2.2-2.el7_9.x86_64.rpm | 103 kB 00:00:00
(23/23): zenity-3.28.1-2.el7_9.x86_64.rpm | 4.0 MB 00:00:00
-----
Total | 8.7 MB/s | 109 MB 00:00:12
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating : krb5-libs-1.15.1-54.el7_9.x86_64 | 1/46
  Updating : samba-common-4.10.16-19.el7_9.noarch | 2/46
  Updating : samba-client-libs-4.10.16-19.el7_9.x86_64 | 3/46
  Updating : libwbclient-4.10.16-19.el7_9.x86_64 | 4/46
  Updating : samba-common-libs-4.10.16-19.el7_9.x86_64 | 5/46
  Updating : python-libs-2.7.5-92.el7_9.x86_64 | 6/46
  Updating : 1:control-center-filesystem-3.28.1-8.el7_9.1.x86_64 | 7/46
  Updating : python-2.7.5-92.el7_9.x86_64 | 8/46
  Updating : samba-libs-4.10.16-19.el7_9.x86_64 | 9/46
  Updating : libsmbclient-4.10.16-19.el7_9.x86_64 | 10/46
  Updating : libkadm5-1.15.1-54.el7_9.x86_64 | 11/46
  Updating : zenity-3.28.1-2.el7_9.x86_64 | 12/46
  Updating : kernel-tools-libs-3.10.0-1160.71.1.el7.x86_64 | 13/46
```

We enter sudo password again.

```
Verifying : xz-libs-5.2.2-2.el7_9.x86_64 | 40/46
Verifying : python-libs-2.7.5-90.el7.x86_64 | 41/46
Verifying : samba-common-libs-4.10.16-18.el7_9.x86_64 | 42/46
Verifying : kernel-headers-3.10.0-1160.66.1.el7.x86_64 | 43/46
Verifying : samba-libs-4.10.16-18.el7_9.x86_64 | 44/46
Verifying : python-perf-3.10.0-1160.66.1.el7.x86_64 | 45/46
Verifying : libwbclient-4.10.16-18.el7_9.x86_64 | 46/46

Removed:
  kernel.x86_64 0:3.10.0-1160.42.2.el7

Installed:
  kernel.x86_64 0:3.10.0-1160.71.1.el7

Updated:
  control-center.x86_64 1:3.28.1-8.el7_9.1 | control-center-filesystem.x86_64 1:3.28.1-8.el7_9.1 | kernel-headers.x86_64 0:3.10.0-1160.71.1.el7
  kernel-tools.x86_64 0:3.10.0-1160.71.1.el7 | kernel-tools-libs.x86_64 0:3.10.0-1160.71.1.el7 | krb5-devel.x86_64 0:1.15.1-54.el7_9
  krb5-libs.x86_64 0:1.15.1-54.el7_9 | libkadm5.x86_64 0:1.15.1-54.el7_9 | libsmbclient.x86_64 0:4.10.16-19.el7_9
  libwbclient.x86_64 0:4.10.16-19.el7_9 | mutter.x86_64 0:3.28.3-31.el7_9 | python.x86_64 0:2.7.5-92.el7_9
  python-libs.x86_64 0:2.7.5-92.el7_9 | python-perf.x86_64 0:3.10.0-1160.71.1.el7 | samba-client-libs.x86_64 0:4.10.16-19.el7_9
  samba-common.noarch 0:4.10.16-19.el7_9 | samba-common-libs.x86_64 0:4.10.16-19.el7_9 | samba-common-tools.x86_64 0:4.10.16-19.el7_9
  samba-libs.x86_64 0:4.10.16-19.el7_9 | xz.x86_64 0:5.2.2-2.el7_9 | xz-libs.x86_64 0:5.2.2-2.el7_9
  zenity.x86_64 0:3.28.1-2.el7_9

Complete!
[sudo] password for cloud_user: []
```

The rpm file is automatically installing.

```
slirp4netns                                x86_64                                0.4.3-4.el7_8                                Extras                                81 K

Transaction Summary
=====
Install 2 Packages (+5 Dependent packages)
Upgrade 1 Package

Total download size: 97 M
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
(1/8): containerd.io-1.6.6-3.1.el7.x86_64.rpm                                | 33 MB  00:00:00
(2/8): docker-ce-20.10.17-3.el7.x86_64.rpm                                | 22 MB  00:00:00
(3/8): docker-ce-rootless-extras-20.10.17-3.el7.x86_64.rpm                | 8.2 MB  00:00:00
(4/8): docker-scan-plugin-0.17.0-3.el7.x86_64.rpm                          | 3.7 MB  00:00:00
(5/8): docker-ce-cli-20.10.17-3.el7.x86_64.rpm                            | 29 MB  00:00:00
(6/8): fuse-overlayfs-0.7.2-6.el7.x86_64.rpm                              | 54 kB  00:00:04
(7/8): fuse3-libs-3.6.1-4.el7.x86_64.rpm                                    | 82 kB  00:00:04
(8/8): slirp4netns-0.4.3-4.el7.x86_64.rpm                                  | 81 kB  00:00:04
-----
Total                                                                    14 MB/s | 97 MB  00:00:06
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : docker-scan-plugin-0.17.0-3.el7.x86_64                        1/9
  Installing : 1:docker-ce-cli-20.10.17-3.el7.x86_64                      2/9
  Installing : slirp4netns-0.4.3-4.el7.x86_64                              3/9
  Installing : containerd.io-1.6.6-3.1.el7.x86_64                          4/9
  Installing : fuse3-libs-3.6.1-4.el7.x86_64                              5/9
  Installing : fuse-overlayfs-0.7.2-6.el7.x86_64                          6/9
  Updating   : 3:docker-ce-20.10.17-3.el7.x86_64 [*****] 7/9]
```

When you see this screen, you need to create a user and password for the database. The user and password we will use here will be used to access the database from the Workstation server we will create shortly. Making this username and password different from sudo's will avoid confusion.

We can make it as Keith did.

Postgres username: demo

```
Verifying : fuse-overlayfs-0.7.2-6.el7.x86_64                                8/9
Verifying : docker-ce-18.06.0-ce-3.el7.x86_64                                9/9

Installed:
  containerd.io.x86_64 0:1.6.6-3.1.el7                                docker-ce-cli.x86_64 1:20.10.17-3.el7

Dependency Installed:
  docker-ce-rootless-extras.x86_64 0:20.10.17-3.el7  docker-scan-plugin.x86_64 0:0.17.0-3.el7  fuse-overlayfs.x86_64 0:0.7.2-6.el7_8  fuse3-libs.x86_64 0:3.6.1-4.el7
  slirp4netns.x86_64 0:0.4.3-4.el7_8

Updated:
  docker-ce.x86_64 3:20.10.17-3.el7

Complete!
Starting and enabling Docker
Configure database user
Postgres user name: ]
```

Password: password

```
Complete!
Starting and enabling Docker
Configure database user
Postgres user name: demo
Postgres user password: ]
```

Automation starts again

```
[sudo] password for cloud_user:
Error: No such container: postgres
Creating database container (and seed 'sample' database)
Unable to find image 'postgres:9.6.8-alpine' locally
9.6.8-alpine: Pulling from library/postgres
ff3a5c916c92: Pull complete
a503b44e1ce0: Pull complete
211706713093: Pull complete
222842002ceb: Extracting [=====] 7.864MB/13.27MB
2300d8bd75d6: Download complete
39a22e6e8a31: Download complete
19df431193ff: Download complete
314e26eea7a7: Download complete
93832602b89c: Download complete
]
```

Finally, our sample database has been completed by installing it on the Linux server we have created via Azure Sandbox

[illegible]

**Then ->> *exit* from PostgreSQL server**

```
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
[cloud_user@d394227ea41c ~]$ exit
```

Now we could go to azure sandbox cloud server again and set up a server for workstation. I made the sizes randomly because of it is tutorial.

Create New Server

Distribution

Fedora Workstation

Zone

Australia

Size

Medium - 3 unit(s)[~2 Virtual CPU, 4 GiB Memory]

Tag

Workstation

Cancel

Create Server

Here I chose fedora, another rpm distributor, because they made it a workstation functionally on the Azure Sandbox, and it installed PostgreSQL and plugins with a very high performance and without any problems.

Our Workstation server installation on Azure sandbox is completed

## Cloud Playground

Learn by doing in our real, live practice environments.

[CLOUD SANDBOXES](#) [INSTANT TERMINAL](#) [CLOUD SERVERS](#)

Units available: 5 Zone: Australia

[Learn more](#)

Distribution CentOS 7 w/ Docker	Tags PostgreSQL	Expires Aug 02, 2022	Units 1	Ready	Quick Actions	Open Terminal	
Distribution Fedora Workstation	Tags Workstation	Expires Aug 03, 2022	Units 3	Ready	Quick Actions	Open Terminal	

[+ New Server](#)

We need to update the password as in the server we set up for the database. We need to update password that every server we create it. I put the same sudo password, because it is tutorial .

```
d394227ea42c login: cloud_user
Password:
You are required to change your password immediately (administrator enforced).
Current password:
New password:
Retype new password:
Last failed login: Tue Jul 19 15:36:31 UTC 2022 from localhost on pts/0
There was 1 failed login attempt since the last successful login.
Last login: Tue Jun 14 15:57:57 on
[cloud_user@d394227ea42c ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos vncserver@.service
[cloud_user@d394227ea42c ~]$
```

we need to install PostgreSQL on this server. We need to apply the following commands respectively.

[\[link\]](#)

install PostgreSQL on fedora ->>

*sudo dnf install postgresql-server postgresql-contrib*

```
d394227ea42c login: cloud_user
Password:
Last login: Tue Jul 19 15:47:17 from localhost
[cloud_user@d394227ea42c ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos vncserver@.service
[cloud_user@d394227ea42c ~]$ pwd
/home/cloud_user
[cloud_user@d394227ea42c ~]$ sudo dnf install postgresql-server postgresql-contrib
```

Tap in Y

```
[cloud_user@d394227ea42c ~]$ sudo dnf install postgresql-server postgresql-contrib
[sudo] password for cloud_user:
Fedora 34 - x86_64                                12 kB/s | 3.6 kB  00:00
Fedora 34 openh264 (From Cisco) - x86_64          3.2 kB/s | 989 B  00:00
Fedora 34 - x86_64 - Updates                       7.9 kB/s | 3.6 kB  00:00
Dependencies resolved.
=====
Package                                Architecture      Version            Repository        Size
-----
Installing:
postgresql-contrib                     x86_64            13.4-1.fc34        updates           817 k
postgresql-server                      x86_64            13.4-1.fc34        updates           5.8 M
Installing dependencies:
libpq                                  x86_64            13.4-1.fc34        updates           203 k
postgresql                             x86_64            13.4-1.fc34        updates           1.5 M
uuid                                   x86_64            1.6.2-50.fc34      fedora            57 k
Transaction Summary
-----
Install 5 Packages
Total download size: 8.3 M
Installed size: 32 M
Is this ok [y/N]: 
```

It is completed then we could switch to second command

```
(1/5): uuid-1.6.2-50.fc34.x86_64.rpm                1.0 MB/s | 57 kB  00:00
(2/5): postgresql-13.4-1.fc34.x86_64.rpm           5.5 MB/s | 1.5 MB  00:00
(3/5): libpq-13.4-1.fc34.x86_64.rpm                 723 kB/s | 203 kB  00:00
(4/5): postgresql-contrib-13.4-1.fc34.x86_64.rpm    2.5 MB/s | 817 kB  00:00
(5/5): postgresql-server-13.4-1.fc34.x86_64.rpm     15 MB/s | 5.8 MB  00:00
-----
Total                                           5.1 MB/s | 8.3 MB  00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                : 1/5
  Installing               : libpq-13.4-1.fc34.x86_64      1/5
  Installing               : postgresql-13.4-1.fc34.x86_64 2/5
  Installing               : uuid-1.6.2-50.fc34.x86_64     3/5
  Installing               : postgresql-contrib-13.4-1.fc34.x86_64 4/5
  Running scriptlet: postgresql-server-13.4-1.fc34.x86_64 5/5
  Installing               : postgresql-server-13.4-1.fc34.x86_64 5/5
  Running scriptlet: postgresql-server-13.4-1.fc34.x86_64 5/5
  Verifying                : uuid-1.6.2-50.fc34.x86_64     1/5
  Verifying                : libpq-13.4-1.fc34.x86_64      2/5
  Verifying                : postgresql-13.4-1.fc34.x86_64 3/5
  Verifying                : postgresql-contrib-13.4-1.fc34.x86_64 4/5
  Verifying                : postgresql-server-13.4-1.fc34.x86_64 5/5
Installed:
libpq-13.4-1.fc34.x86_64      postgresql-13.4-1.fc34.x86_64      postgresql-contrib-13.4-1.fc34.x86_64      postgresql-server-13.4-1.fc34.x86_64      uuid-1.6.2-50.fc34.x86_64
complete!
[cloud_user@d394227ea42c ~]$ 
```

*sudo systemctl enable postgresql*

```
[cloud_user@d394227ea42c ~]$ sudo systemctl enable postgresql
Created symlink /etc/systemd/system/multi-user.target.wants/postgresql.service → /usr/lib/systemd/system/postgresql.service.
```

*sudo postgresql-setup --initdb --unit postgresql*

```
[cloud_user@d394227ea42c ~]$ sudo postgresql-setup --initdb --unit postgresql
* Initializing database in '/var/lib/pgsql/data'
* Initialized, logs are in /var/lib/pgsql/initdb_postgresql.log
[cloud_user@d394227ea42c ~]$ 
```

*sudo systemctl start postgresql*

```
[cloud_user@d394227ea42c ~]$ sudo systemctl start postgresql
[cloud_user@d394227ea42c ~]$ 
```

From here, we can connect to our database server from the workstation. We will use the username "demo" and the password "password" that we specified while creating the database server.

connect to PostgreSQL server(database) from fedora34-workstaion server and play it using by SQL. Public IPv4(dynamic) will swap every time you start it, username, and password(static)

## IP Address

### Public IPv4

3.25.204.223



`psql postgres://demo:password@3.25.204.223:80/sample -c "SELECT * FROM employees;"`

```
[cloud_user@d394227ea42c ~]$ sudo systemctl start postgresql
[cloud_user@d394227ea42c ~]$ psql postgres://demo:password@3.25.204.223:80/sample -c "SELECT * FROM employees;"
```

Here we can play on the database with simple or complex SQL commands. The exercises you will do with SQL will help you a lot in your python development. I did some practice with sql on the database for fun, it will be beneficial for you to do it too.

```
[cloud_user@d394227ea42c ~]$ psql postgres://demo:password@3.25.204.223:80/sample -c "SELECT * FROM employees;"
```

id	first_name	last_name	email	gender	favorite_color
1	Lauralee	Morkham	lmorkham0@example.com	Female	#878922
2	Hillery	Langland	hlangland1@example.com	Male	#6fd569
3	Regan	Kroger	rkroger2@example.com	Male	#d9c547
4	George	Treasaden	gtreasaden3@example.com	Male	#d5e6c2
5	Raddy	Curley	rcurley4@example.com	Male	#83974a
6	Waylen	Tott	wtott5@example.com	Male	#90532b
7	Filmore	Chartre	fchartre6@example.com	Male	#6a1fb5
8	Ulberto	Pimme	upimme7@example.com	Male	#7560c1
9	Sigfried	Lowre	slowre8@example.com	Male	#37c45b
10	Edwina	Henrichsen	ehenrichsen9@example.com	Female	#00ef5c
11	Emmeline	Harty	ehartya@example.com	Female	#004399
12	Nolan	Cansdall	ncansdallb@example.com	Male	#fff920
13	Chrystel	Wickey	cwickeyc@example.com	Female	#33b833
14	Ezequiel	McCart	emccartd@example.com	Male	#0437d5
15	Diarmid	Main	dmaine@example.com	Male	#f3f435
16	Jessamine	Jansik	jjansikf@example.com	Female	#db3da9
17	Linell	Brimicombe	lbrimicombeg@example.com	Female	#68e029
18	Faber	Netting	fnettingh@example.com	Male	#9c772e
19	Roz	Caple	rcaplei@example.com	Female	#cc5cb2
20	Caleb	Milch	cmilchj@example.com	Male	#8f1c39
21	Krystalle	Gibling	kgiblingk@example.com	Female	#78254d
22	Felipa	Pardy	fpardyl@example.com	Female	#b8b32e
23	Krystalle	Inkster	kinksterm@example.com	Female	#70144a
24	Loralyn	Hoofe	lhoofen@example.com	Female	#da0b31
25	Mirella	Sandars	msandarso@example.com	Female	#27a0ac
26	Stacee	Megahey	smegaheyp@example.com	Male	#8bcb37
27	Benetta	Olivelli	bolivelliq@example.com	Female	#0940f9
28	Ericka	Waylen	ewaylenr@example.com	Female	#3696d7
29	Virgie	Meiklam	vmeiklams@example.com	Female	#34fc78
30	Felecia	Tow	ftowt@example.com	Female	#bfd7db
31	Fanya	Elmhirst	felmhirstu@example.com	Female	#3c02a3
32	Rafi	Juschka	rjuschkav@example.com	Male	#e3ab10
33	Ruggiero	Buttriss	rbuttrissw@example.com	Male	#7ea766
34	Spenser	Shepton	ssheptonx@example.com	Male	#ca1ef4
35	Leann	Gooch	lgoochy@example.com	Female	#b96ca9



```
psql postgres://demo:password@3.25.204.223:80/sample -c "SELECT gender FROM employees
WHERE gender = 'Female';"
```

[illegible]

```
psql postgres://demo:password@3.25.204.223:80/sample -c "SELECT * FROM employees WHERE email LIKE '%tt%'";"
```

```
[cloud_user@id394227ea42c ~]$ psql postgres://demo:password@3.25.204.223:80/sample -c "SELECT * FROM employees WHERE email LIKE '%tt%';"
```

id	first_name	last_name	email	gender	favorite_color
6	Waylen	Tott	wtott5@example.com	Male	#90532b
18	Faber	Netting	fnettingh@example.com	Male	#9c772e
33	Ruggiero	Buttriss	rbuttrissw@example.com	Male	#7ea766
98	Beverlee	Butting	bbutting2p@example.com	Female	#b9bae1
120	Darci	Kettles	dkettles3b@example.com	Female	#ea1608
142	Desirae	Muffett	dmuffett3x@example.com	Female	#d77540
164	Christabel	Lovett	clovett4j@example.com	Female	#5fc98d
170	Joellyn	Pettisall	jpettisall4p@example.com	Female	#8b839d
190	Carma	Cottrell	ccottrell159@example.com	Female	#962cfb
227	Amby	Checketts	achecketts6a@example.com	Male	#de32e4
245	Asher	Fettis	afettis6s@example.com	Male	#cca6be
258	Lenette	Betteriss	lbetteriss75@example.com	Female	#9bb973
259	Bree	Muscott	bmuscott76@example.com	Female	#7b7b2c
264	Neall	Knappett	nknappett7b@example.com	Male	#5c5489
268	Alfy	Tottem	atottem7f@example.com	Female	#667637
277	Uta	Attawell	uattawell7o@example.com	Female	#426213
282	Mikkel	Beatty	mbeatty7t@example.com	Male	#9acd67

```
psql postgres://demo:password@3.25.204.223:80/sample -c "SELECT * FROM EMPLOYEES WHERE first_name NOT LIKE '%a%' AND id BETWEEN 200 AND 220;"
```

\*sql-SQL

```
885 | Cynthia | Lynnitt | clynnittok@example.com | Female | #6c2999
926 | Bertram | Waskett | bwaskettpp@example.com | Male | #0f139f
963 | Stafford | Romanetti | sromanettiq@example.com | Male | #ffea11
994 | Ragnar | Brewitt | rbrewittrl@example.com | Male | #1478c3
(59 rows)

[cloud_user@d394227ea42c ~]$ psql postgres://demo:password@3.25.204.223:80/sample -c "SELECT * FROM EMPLOYEES WHERE first_name NOT LIKE '%a%' AND id BETWEEN 200 AND 220;"
 id | first_name | last_name | email | gender | favorite_color
-----+-----+-----+-----+-----+-----
200 | Meridel | Jahner | mjahner5j@example.com | Female | #88632b
202 | Lem | Welden | lwelden5l@example.com | Male | #88553b
203 | Kristel | Worsell | kworsell15n@example.com | Female | #8d8193
204 | Christoforo | Ismail | cismail5n@example.com | Male | #83ff68
205 | Scotty | Hentze | shentze5o@example.com | Male | #ca777f
207 | Kinmy | Sabates | ksabates5q@example.com | Female | #aa08b0
209 | Kim | Custy | kcusty5s@example.com | Female | #7a009d
210 | Rochester | Foale | rfoale5t@example.com | Male | #80cf3f
212 | Ines | Nadin | inadin5v@example.com | Female | #f51e7e
214 | Joellie | McInoney | jmcloney5x@example.com | Female | #070e08
215 | Bekki | Farres | bfarres5y@example.com | Female | #404bbe
217 | Morgen | Toyer | mtoyer6a@example.com | Male | #a2f701
(12 rows)

[cloud_user@d394227ea42c ~]$
```

```
psql postgres://demo:password@3.25.204.223:80/sample -c "SELECT * FROM EMPLOYEES WHERE id>900 AND (last_name LIKE '%tt%' OR last_name LIKE '%ll%');"

```

```
[cloud_user@d394227ea42c ~]$
psql postgres://demo:password@3.25.204.223:80/sample -c "SELECT * FROM EMPLOYEES WHERE id>900 AND (last_name LIKE '%tt%' OR last_name LIKE '%ll%');"
 id | first_name | last_name | email | gender | favorite_color
-----+-----+-----+-----+-----+-----
904 | Corny | Stegell | cstegellp3@example.com | Male | #d27d0f
926 | Bertram | Waskett | bwaskettpp@example.com | Male | #0f139f
929 | Millisent | Lille | mlilleps@example.com | Female | #129790
934 | Joseph | Riall | jriallpx@example.com | Male | #27a216
944 | Rici | Fallowes | rfallowesq7@example.com | Female | #d97251
945 | Duky | Okill | dokillq8@example.com | Male | #12455b
958 | Georas | Whetnall | gwhetnallql@example.com | Male | #4975e3
963 | Stafford | Romanetti | sromanettiq@example.com | Male | #ffea11
976 | Irita | Willisich | iwillisichr3@example.com | Female | #b419c6
978 | Raphael | Gallyon | rgallyonr5@example.com | Male | #091048
985 | Hynda | Hall-Gough | hhallgoughrc@example.com | Female | #c5727b
994 | Ragnar | Brewitt | rbrewittrl@example.com | Male | #1478c3
998 | Malinde | Powell | mpowellrp@example.com | Female | #766951
1000 | Killie | Peperell | kpeperellrr@example.com | Male | #c523eb
(14 rows)

[cloud_user@d394227ea42c ~]$
```

```
psql postgres://demo:password@3.25.204.223:80/sample -c "SELECT last_name
FROM employees WHERE employees.email LIKE '%ll%' AND id <= 100;"
```

```
[cloud_user@d394227ea42c ~]$ psql postgres://demo:password@3.25.204.223:80/sample -c "SELECT last_name
FROM employees WHERE employees.email LIKE '%ll%' AND id <= 100;"
 last_name
-----
Cansdall
Olivelli
Grimsdell
Stillwell
Cripwell
Dyball
Mallord
Campbell-Dunlop
Currall
Yoodall
Collard
Golly
(12 rows)

[cloud_user@d394227ea42c ~]$
```

You could continue yourself to into SQL and do more exercise by using Cloud systems.

As you know, we are trying to create a web-based system. The first thing Keith does is set up a sample database server and manage it from another server running as Workstation. In our study, I used rmp, you can practice a lot by using dpkg(Debian-Ubuntu) and changing the variations by making Cloud servers AWS or Google Cloud.

We have just installed the workstation server and database now and there are many components such as front-end & back-end in cloud computing architecture.

As you know, In this way, instead of reloading the whole system by creating a server for the front-end, we can divide the front-end or back-end into parts using microservices. As a cloud engineer, we do the development, deployment and operations(DevOps) of the images of these services, which are update by the developers, and other contributor assigned in task manager.

As an example cloud servers that we can use for the website we will create

## Cloud Playground

Learn by doing in our real, live practice environments.

[Watch "What are Cloud Servers?"](#)

[CLOUD SANDBOXES](#) [INSTANT TERMINAL](#) [CLOUD SERVERS](#)

Units available: 0 Zone: Australia

[Learn more](#)

Distribution CentOS 7 w/ Docker	Tags PostgreSQL	Expires Aug 02, 2022	Units 1	Ready	Quick Actions	Open Terminal		
Distribution Fedora Workstation	Tags Workstation	Expires Aug 03, 2022	Units 3	Ready	Quick Actions	Open Terminal		
Distribution CentOS w/ code-server	Tags Back-End	Expires Aug 03, 2022	Units 4	Ready	Quick Actions	Open Terminal		
Distribution CentOS 7 w/ Docker	Tags Front-End	Expires Aug 03, 2022	Units 1	Ready	Quick Actions	Open Terminal		

⚠ We actively monitor Cloud Playground for abusive, prohibited, or otherwise un-awesome behavior. Cloud Playground is for educational purposes only.

Having issues? Contact [Cloud Playground Support](#).

More power to your elbow.

Author : Mehmet A. Kir

Source : [\[Link\]](#)