



#Xalvy

XALDIGITAL

Challenge Data Engineer

Canseco García Edgar Jesús

Agosto de 2022

Solution

Environment setup

For environment setup, I create a VM on GCP (a budget topic), to deploy an environment from scratch.

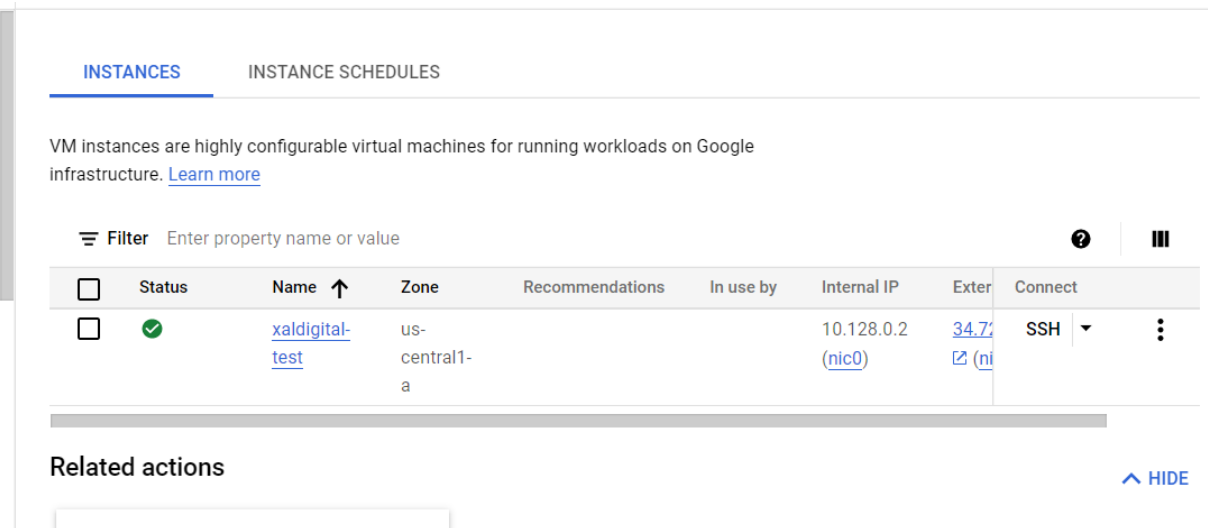


Figura 1- GCP VM

The image used is Ubuntu, I installed the necessary components to run a GUI environment, python, and docker for this challenge.

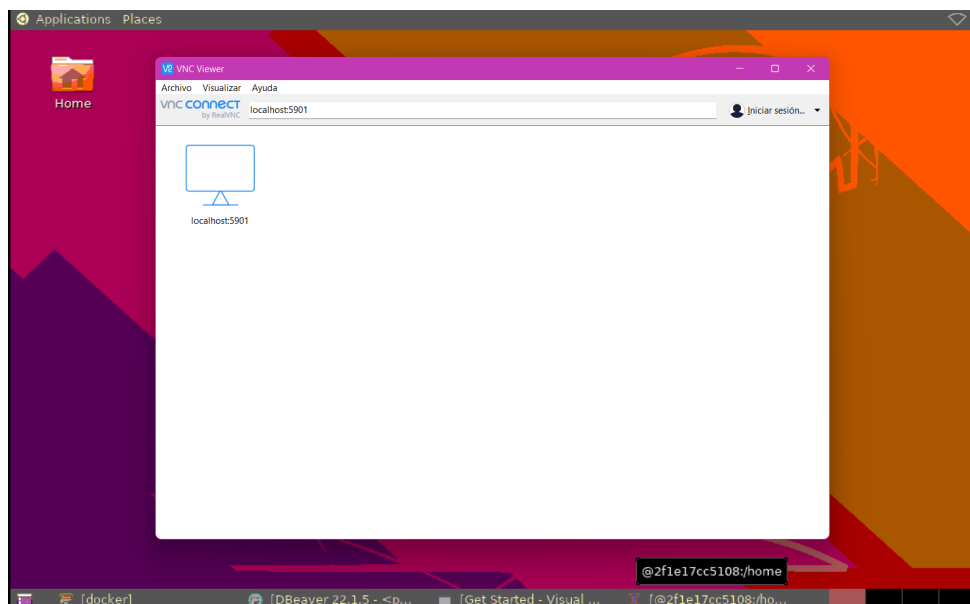
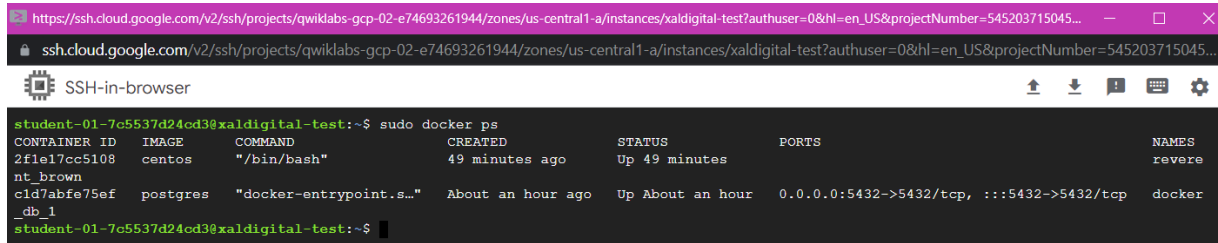


Figura 2- GUI Configuration

With the yml granted for the challenge, a Postgres database and a centos server are configured and launched with docker compose.



```
student-01-7c5537d24cd3@xaldigital-test:~$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2fle17cc5108	centos	"/bin/bash"	49 minutes ago	Up 49 minutes		revere
nt_brown						
cld7abfe75ef	postgres	"docker-entrypoint.s..."	About an hour ago	Up About an hour	0.0.0.0:5432->5432/tcp, :::5432->5432/tcp	docker_db_1

```
student-01-7c5537d24cd3@xaldigital-test:~$
```

Figura 3- Containers

Design the E-R

Once the connection to PostgreSQL is made, I design the E-R from the database and create the table schema where data will be ingested.

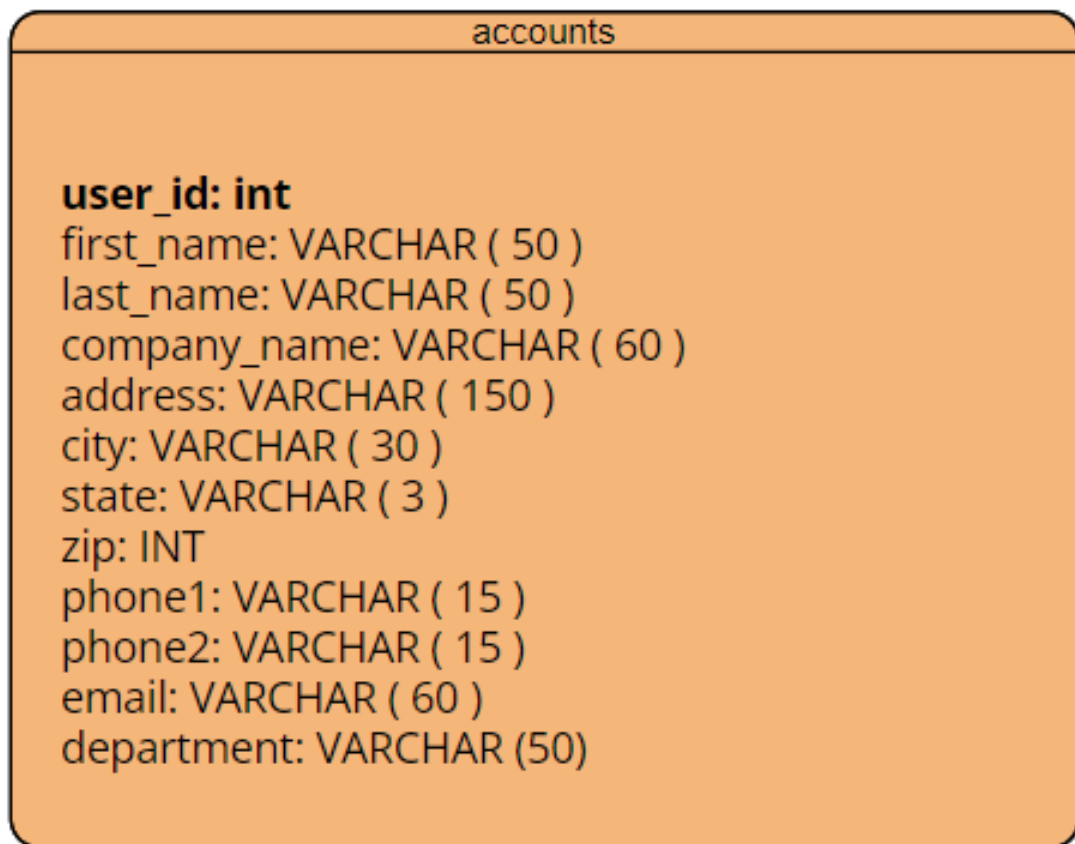


Figura 4- E-R

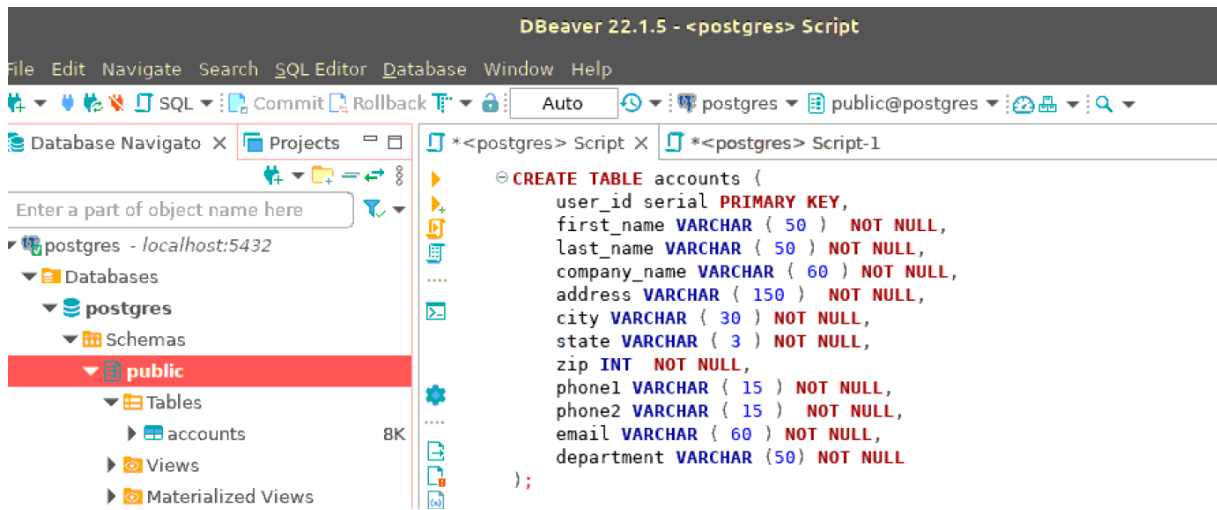


Figura 5- Table creation

Ingest the data

With the 'docker cp' command I copy the file from my ubuntu environment to centos server.



Figura 6- CSV file

I tried to make the connection from the centos server to PostgreSQL, I installed the necessary libraries but I got a bit confused with the network configurations and communication between containers. I still have the work of learning how the containers need to be configured with more time.



```

Get Started - Visual Studio Code

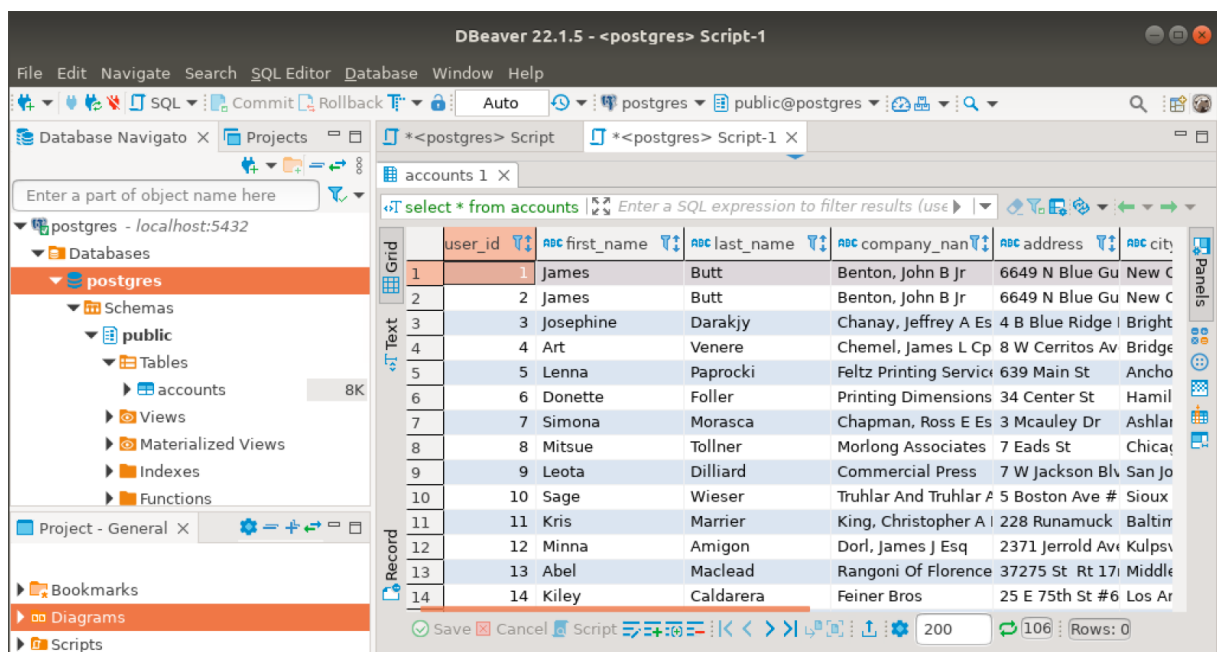
@2f1e17cc5108:/home

KeyboardInterrupt
>>>
[root@2f1e17cc5108 home]# listen_addresses = '*'
bash: listen_addresses: command not found
[root@2f1e17cc5108 home]# python3
Python 3.6.8 (default, Sep 10 2021, 09:13:53)
[GCC 8.5.0 20210514 (Red Hat 8.5.0-3)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import psycopg2
>>> conn = psycopg2.connect(host='172.18.0.2',
... dbname='postgres',
... user='postgres',
... password='example',
... port='5432')
Traceback (most recent call last):
  File "<stdin>", line 5, in <module>
  File "/usr/local/lib64/python3.6/site-packages/psycopg2/__init__.py", line 122
, in connect
    conn = _connect(dsn, connection_factory=connection_factory, **kwargs)
psycopg2.OperationalError: could not connect to server: Connection timed out
Is the server running on host "172.18.0.2" and accepting
TCP/IP connections on port 5432?
>>>

```

Figura 7- Failed connection

So to continue with the challenge, I will use my ubuntu server instead of centos. With this, I create the first version of the code that validates the state column and inserts the data into PostgreSQL



DBeaver 22.1.5 - <postgres> Script-1

File Edit Navigate Search SQL Editor Database Window Help

Database Navigator x Projects x

Enter a part of object name here

postgres - localhost:5432

- Databases
 - postgres
 - Schemas
 - public
 - Tables
 - accounts 8K
 - Views
 - Materialized Views
 - Indexes
 - Functions

Project - General x

Bookmarks

Diagrams

Scripts

*<postgres> Script

*<postgres> Script-1 x

accounts 1 x

select * from accounts

	user_id	first_name	last_name	company_name	address	city
1	1	James	Butt	Benton, John B Jr	6649 N Blue Gu	New C
2	2	James	Butt	Benton, John B Jr	6649 N Blue Gu	New C
3	3	Josephine	Darakjy	Chanay, Jeffrey A Es	4 B Blue Ridge I	Bright
4	4	Art	Venere	Chemel, James L Cp	8 W Cerritos Av	Bridge
5	5	Lenna	Paprocki	Feltz Printing Servi	639 Main St	Ancho
6	6	Donette	Foller	Printing Dimensions	34 Center St	Hamil
7	7	Simona	Morasca	Chapman, Ross E Es	3 McAuley Dr	Ashlar
8	8	Mitsue	Tollner	Morlong Associates	7 Eads St	Chica
9	9	Leota	Dilliard	Commercial Press	7 W Jackson Bl	San Jo
10	10	Sage	Wieser	Truhlar And Truhlar	5 Boston Ave #	Sioux
11	11	Kris	Marrier	King, Christopher A	228 Runamuck	Baltir
12	12	Minna	Amigon	Dorl, James J Esq	2371 Jerrold Av	Kulps
13	13	Abel	Maclead	Rangoni Of Florence	37275 St Rt 17	Middle
14	14	Kiley	Caldarera	Feiner Bros	25 E 75th St #6	Los Ar

Save Cancel Script

200 106 Rows: 0

Figura 8- Table Result

Api for read

Finally, I implement the API with flask for reading the first 25 elements by the user_id of the table. The API address is: <http://localhost:5000/api/accounts/>

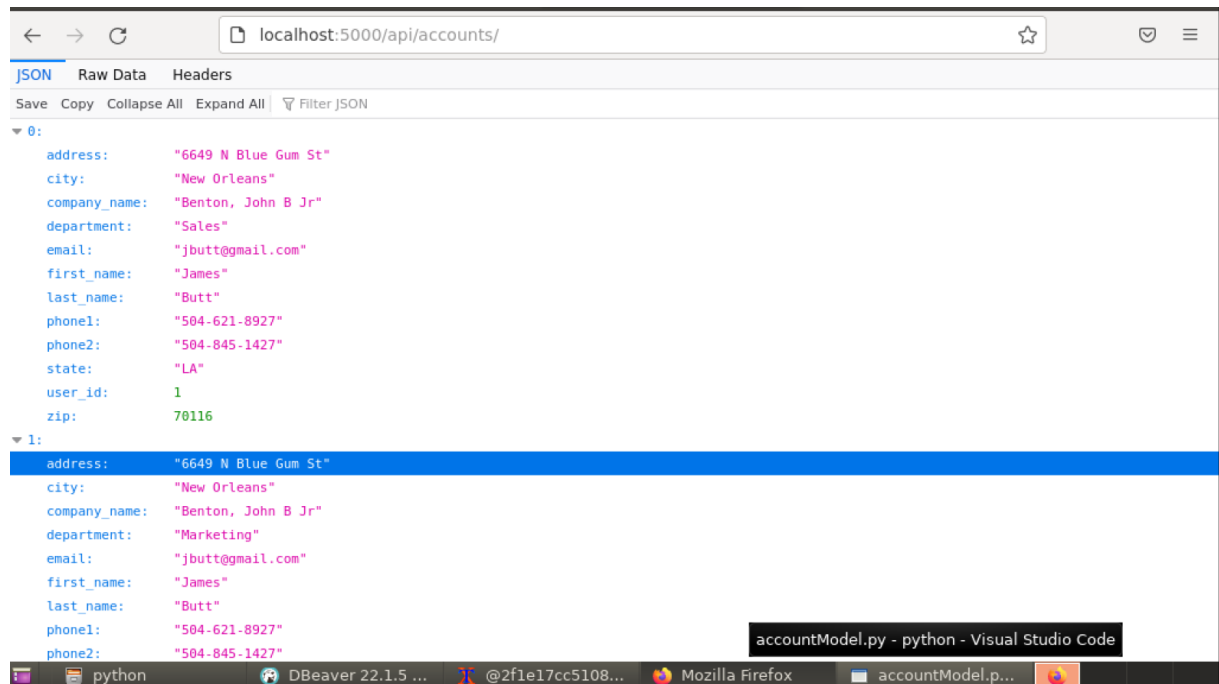


Figura 9- Table Result

Conclusion

It was fun to implement the challenge. I learned a lot implementing my first API, and also I have the work to search the Docker configurations better.