Lab 2

Overview

I created a Google Cloud instance, as I was familiar with it and had credits in my account to use. I set up a modified personal version of TLE, a Discord bot specialised in competitive programming statistics, which I have been contributing to since 2019. My modified code can be found here.

TLE stores users linked to the bot on Discord to their corresponding Codeforces accounts and connects to the Codeforces API. They can then access the past contests, solved problems, rating history of a user through a set of commands and more statistics related to the website.

Conclusion

Undertaking this laboratory encouraged me to compare different types of cloud providers, allowing me to increase my understanding of cloud computing. I mainly experimented with AWS and Google Cloud for this lab, and I found Google Cloud more straightforward to use. I also felt that Google Cloud console provided the user with more freedom, as I was able to SSH into a VM instance I created very simply, while I was not able to understand how to do so for AWS. I also enjoyed the larger range of options of locations provided by Google Cloud, as well as the indication of low-CO2 regions.

In comparison to Lab 1, where we had to investigate virtual machines, I found that the VM provided by Google Cloud (SSH-in-Browser) was somewhat restrictive, as we are limited to only using the Linux Terminal, however applications that I previously ran on the other VM (Oracle VirtualBox) ran much faster on the Google Cloud VM.

Screenshots

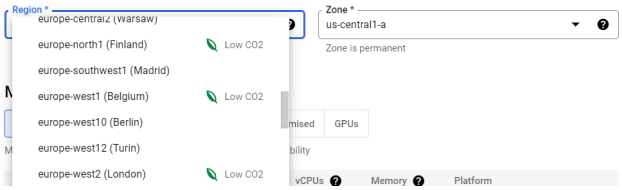

Figure 1: Google Cloud Portal Instance Zone Options

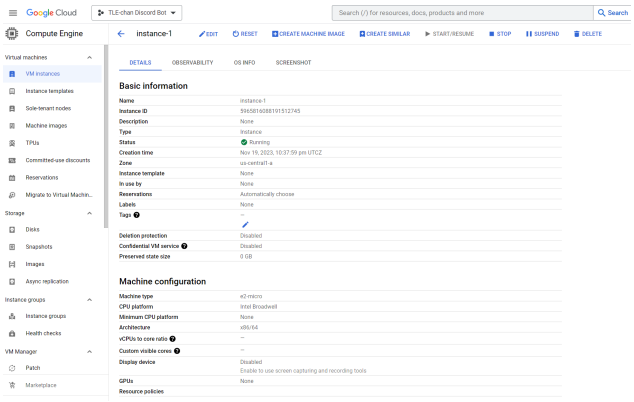

Figure 2: Google Cloud Portal Created Instance Information

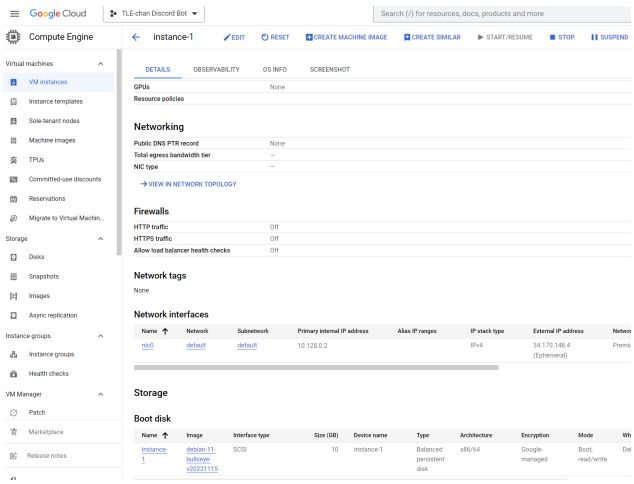

Figure 3: Google Cloud Portal Created Instance Information (contd.)

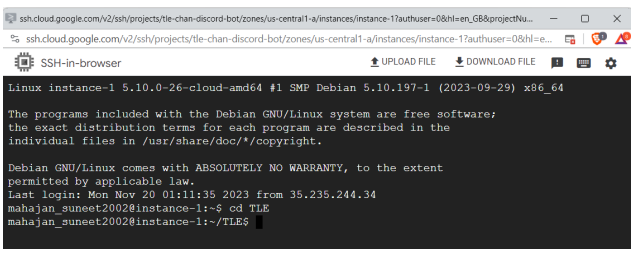

Figure 4: Google Cloud SSH-in-Browser

```
35
36  def nice_sub_type(types):
37      nice_map = {'CONTESTANT':'Contest: {}',
38                  'OUT_OF_COMPETITION':'Unofficial: {}',
39                  'VIRTUAL':'Virtual: {}',
40                  'PRACTICE':'Practice: {}'}
41      return [nice_map[t] for t in types]
42
43  def _plot_rating(plot_data, mark):
44      for ratings, when in plot_data:
45          plt.plot(when,
46                   ratings,
47                   linestyle='-',
48                   marker=mark,
49                   markersize=3,
50                   markerfacecolor='white',
51                   markeredgewidth=0.5)
52      gc.plot_rating_bg(cf.RATED_RANKS)
```

Figure 5: Code Excerpt

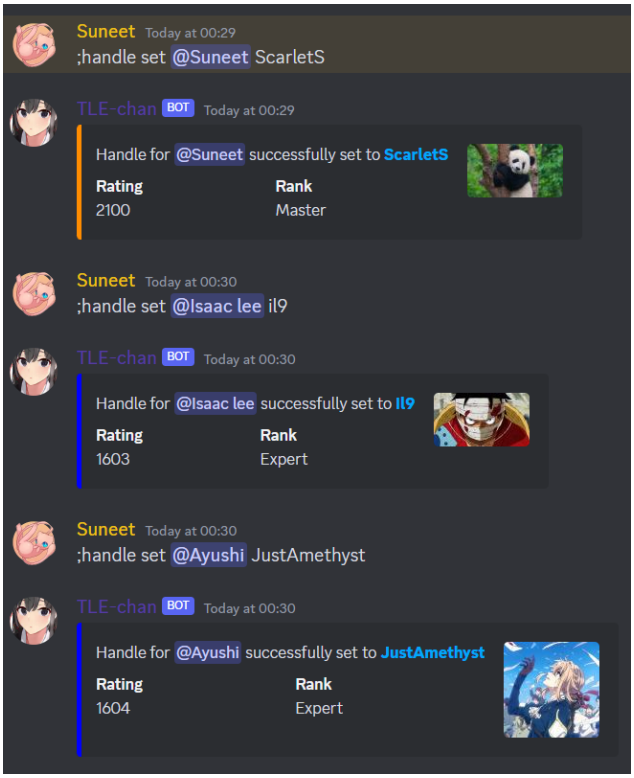

Figure 6: Adding data (linking users with accounts) to the database

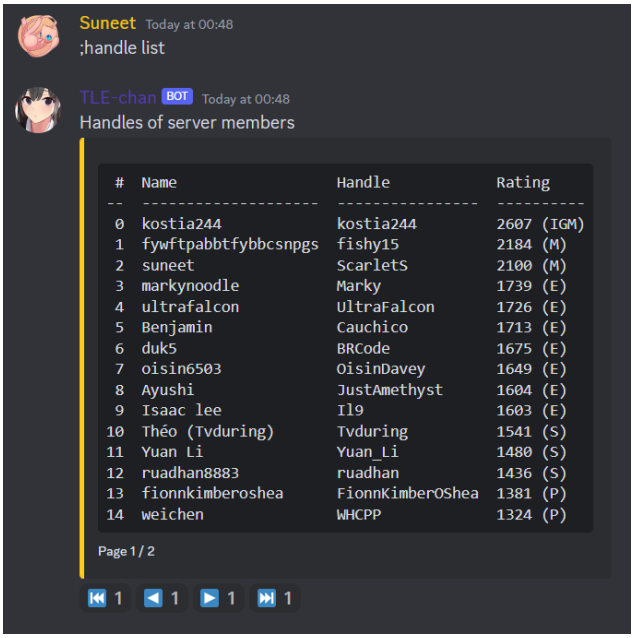

Figure 7: Querying all the data in the database (in order of descending rating), supporting pagination for larger datasets

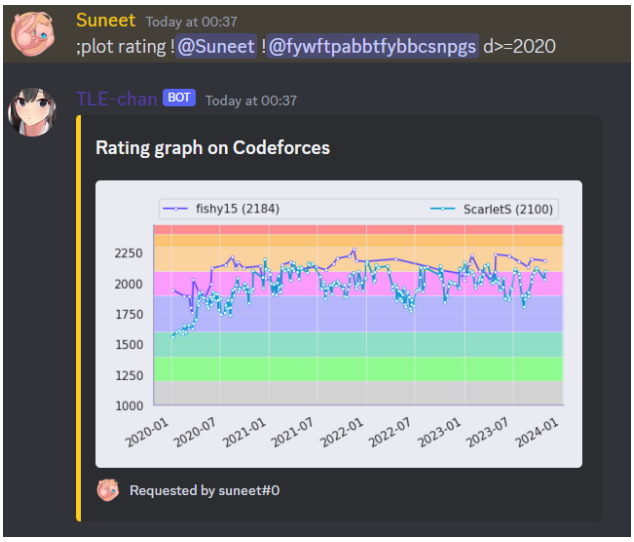

Figure 8: Comparing the ratings of two users over a defined time period

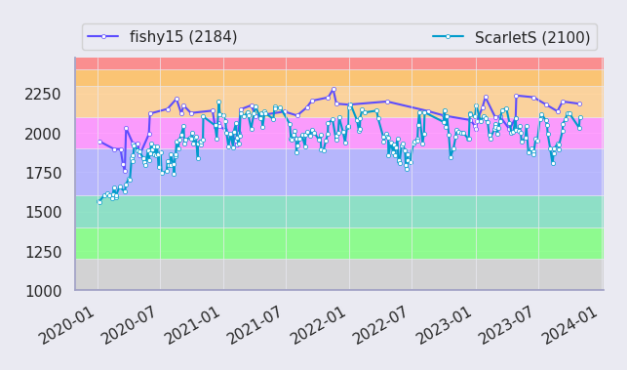

Figure 9: Zoomed-in generated rating graph requested above, comparing the ratings of the myself and the Discord user named fywftpabbtfybbcsnpgs