The Idea

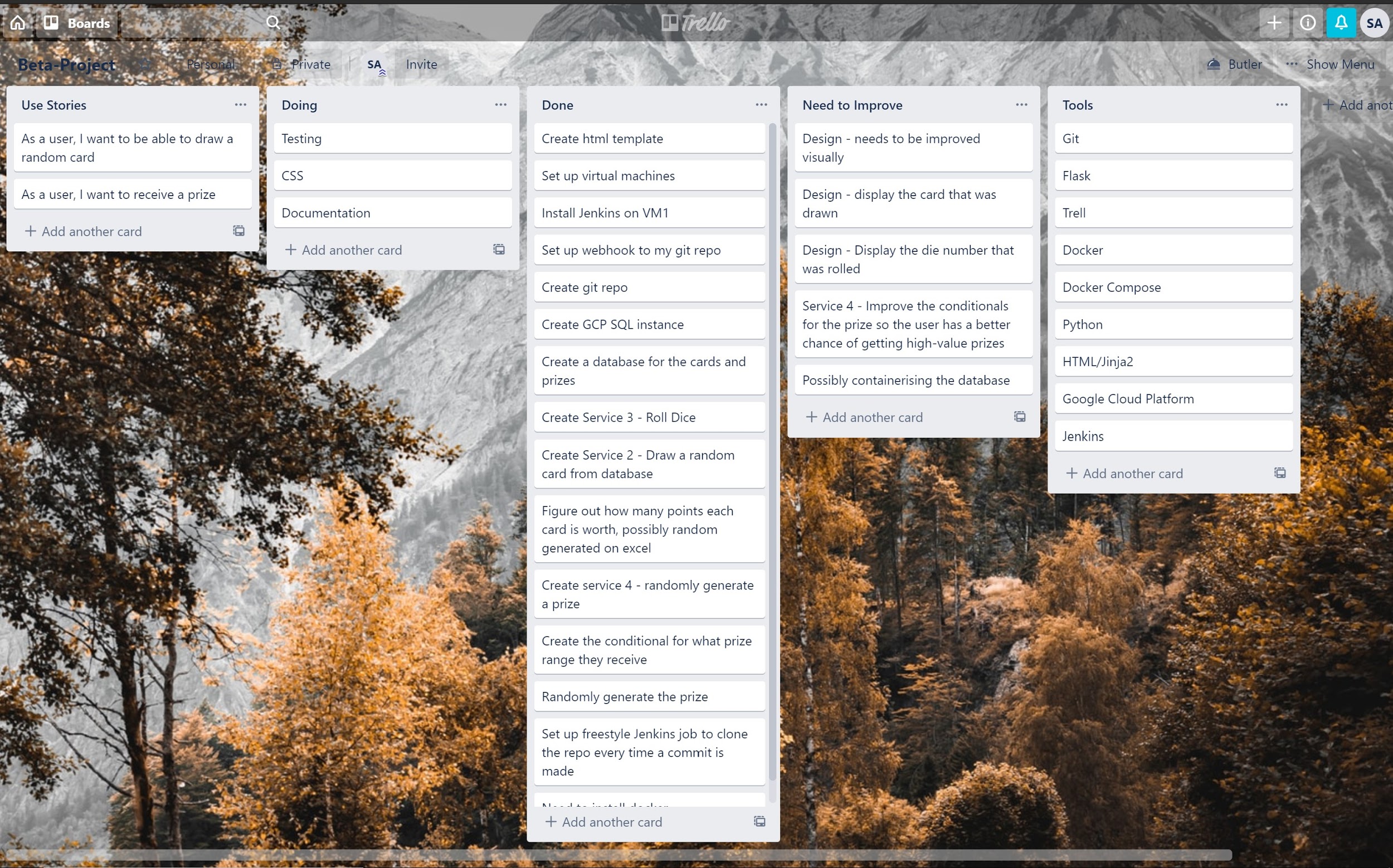
The idea of this project is to create a flask app game that will draw a random card from a database which has a random point assigned to it and to roll of a dice. The two values are then multiplied together to reach a final point. Depending on how many points you get will determine how good your prize is, which is also randomly generated from a database.

Tools Used

**Google Cloud Platform** - This is for a virtual machine to be used so the application and jenkins back be hosted separately on different machines.

**Git** - This is a version control system which will help save/track your save progress and ensure the code that is being written isn’t lost. Git also gives the opportunity for possible collaboration with others

**Trello** - Trello is used to help keep track of overall progress of development. The Kanban board can assist with planning for the development of the project and setting personal goals and deadlines throughout the development of the project

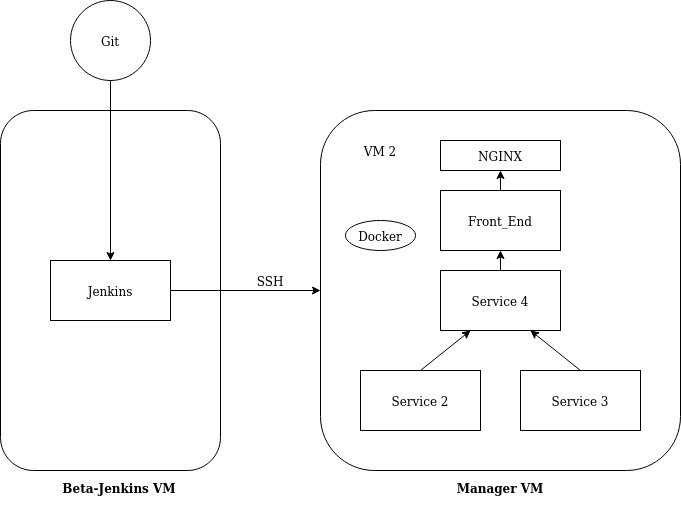


**Flask** - the application was a flask application. Flask is micro web service which allowed the application to be written in python, with a front end design and python functions for each service.

**Docker** - this tool helps containing the 4 services which were create individually

**Jenkins** - To assist with getting the latest version of the app and deploying the application, Jenkins was helpful. It was configured to ssh into the second virtual machines clone the git repo, and then build the application.

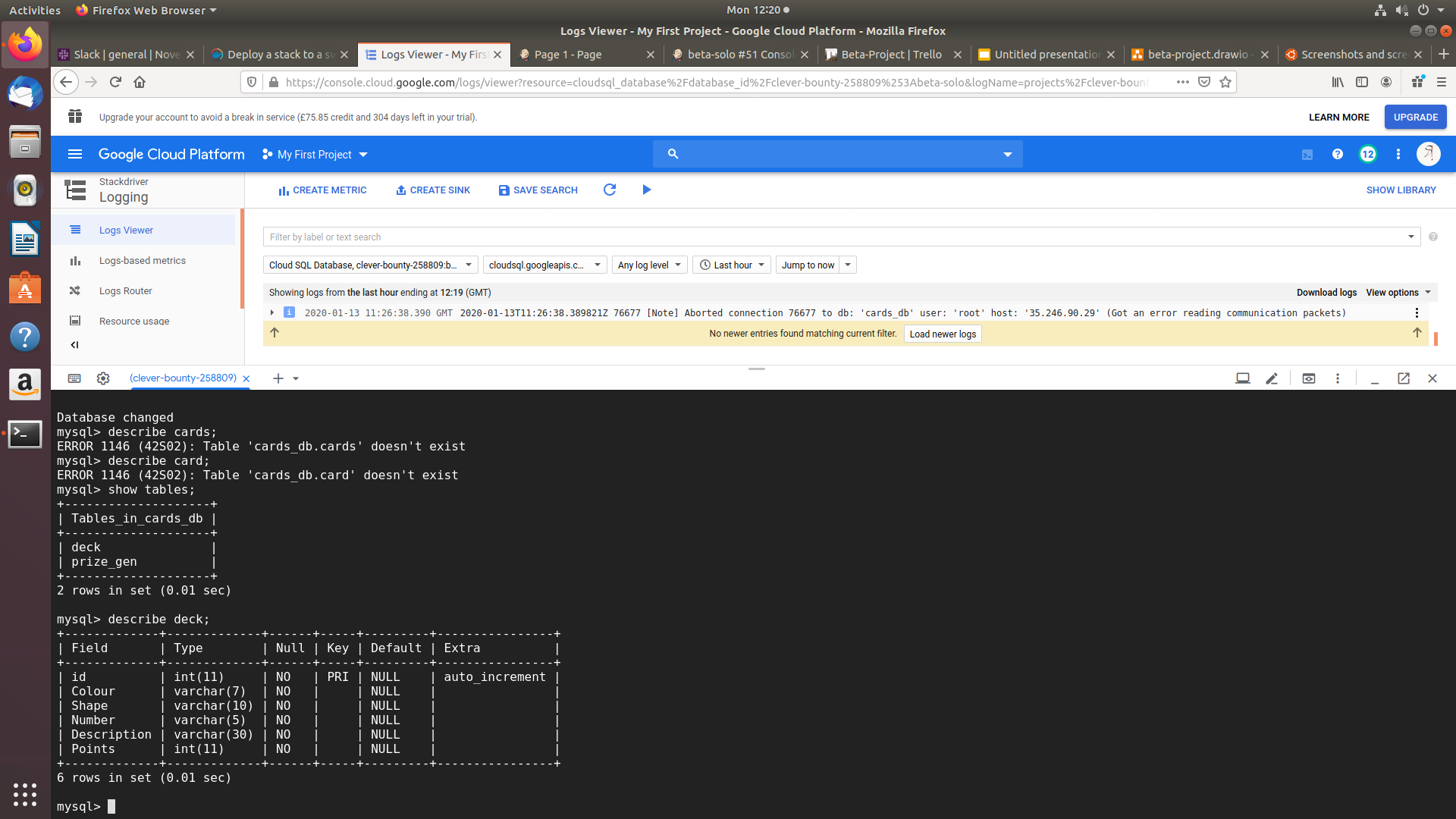
Structure



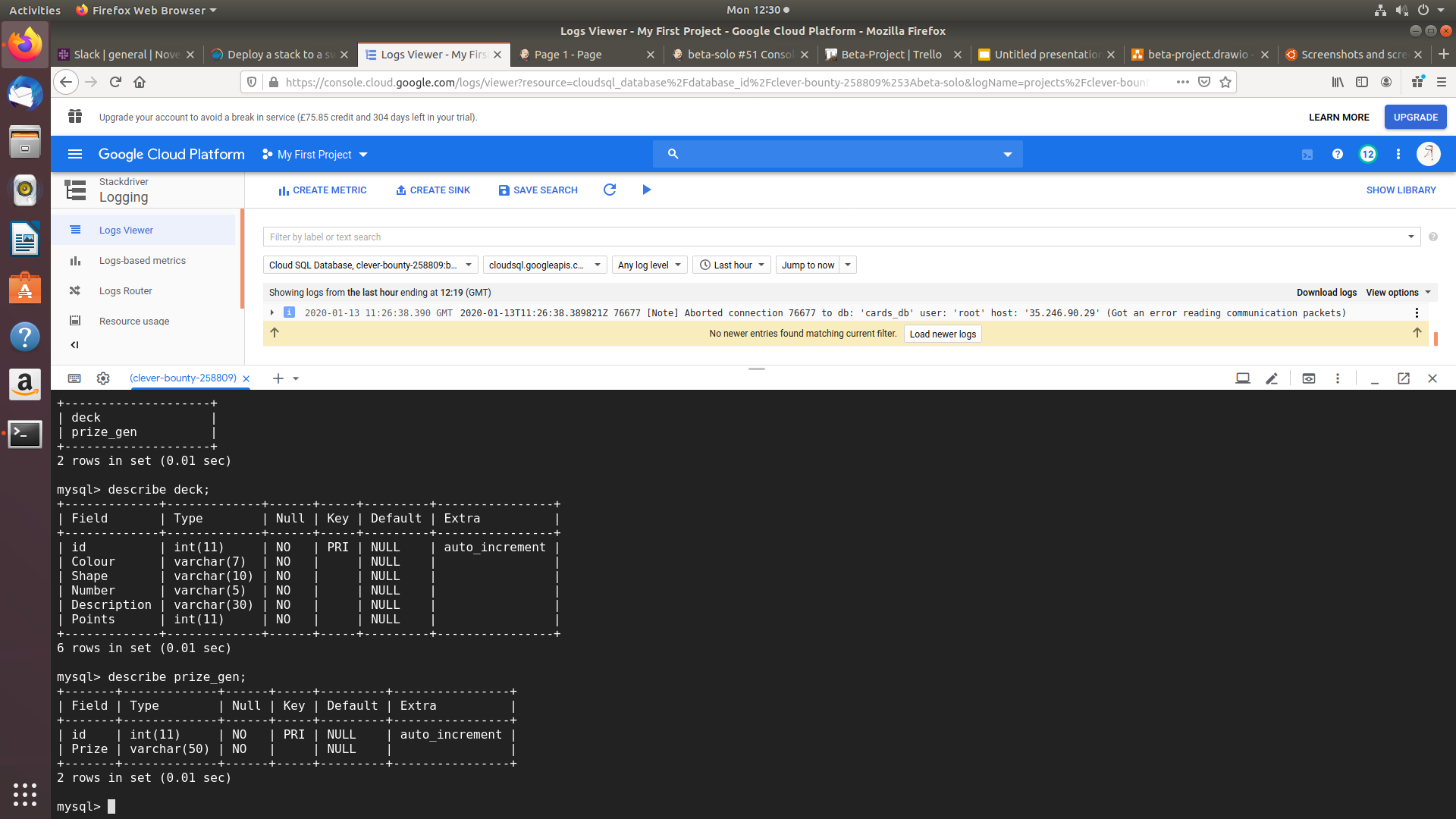
There are two VM’s, one VM that has jenkins install on it which is also webhooked to the git repository for the application. Within the second VM, the application is hosted and stores service images in the docker registry. The application VM is also the Swarm master so can push the images to the swarm worker, this is essential when updating the application.

The way the application is structured, the front end of the application talks Service 4 by receiving the prize from a dictionary within it , and the other two services are communicating to service 4.

Within service 4, the value that has been stored in a dictionary from service 2 and service 3 will be called and stored as an integer in service 4. Both values are then multiplied in service 4 and once the total is made, the prize database is queried based on the scores.



Services 2 and Services 4 both need databases as one is querying the database for cards and the other is querying the database for prizes.



At the base of the application, there is an NGIX web server which acts as a load balancer and handles reverse proxies. When the user attempts to access the application, they will be connected to the app through port 80 where nginx itself is communicating with front end of the application. Having nginx at the top of communication will limit the user so they would never talk directly to the flask application but would be redirected there instead through nginx.

Possible Improvements

The design of the application can be vastly improved by adding CSS or picture of the cards after the user has drawn their card, along with how many points the card is worth. Another improvement includes to show the dice number that they have rolled. All these improvements can greatly make the application appealing and improve the experience for the users.

Dockerizing the database can also be an aspect of improvement so that it can be deployed along with the application.

Risk Assessment

|  |  |  |
| --- | --- | --- |
| Risk | Likelihood | Solution |
| Lack of understanding with assignment brief | Low | When faced with the issue of not understanding the briefing, one of the best things to do is to read the assignment brief a few more time. This will allow familiarity with the brief and then the individual who set the assignment brief can be asked for clarification and understanding of the task |
| Planning | Medium | It is possible to start developing an application without planning but this is not advised as it will lead to human error and can lead to accidentally developing away from the users need and the scope of the project as a whole. The solution can be to use a kanban board software such as Trello in order to keep track of use cases,software design, completed tasks and uncompleted tasks. |
| No online connection | Medium | Online connection is not immediately because it is possible to write/code a whole application locally. But being only is very important especially if a cloud service or version control system is being used. |
| GCP going down | Low | The threat of GCP doing down is very low as GCP are owned by google and are very trustworthy but ever in rare occasion GCP ever goes down, other cloud based virtual machines such as Microsoft Azure or Amazon Cloud Services can be used |
| Loss of work | High | It is very easy to lose the environment or the folder in which the application is written in. One of the best solutions to protect against the loss of work is to use a version control system such as GIT. A version control system will also allow the retrieval of past saved versions of the application |
| Hardware | High | Hardware where the application is being made can face unexpected damages which will hinder the create of application. A solution to avoid the hardware from failing on you is to possibly do regular hardware maintenance. Preferably relying on the hardware working 27/7 is not advised. |
| Application not working | Mediam | After the creation sometime it is expected for the application not work the first time round so when it doesn’t work, reading error logs will help with debugging error messages. Another solution to help will be to write tests to check multiple aspects of the code. |
| Poor time management | High | Naturally time can become an issue as unexpected event can occur which will delay the production of the application. The solution for time management will be to plan your time and give extra for unexpected delays. Time management tools such as Gantt Charts can also be used |