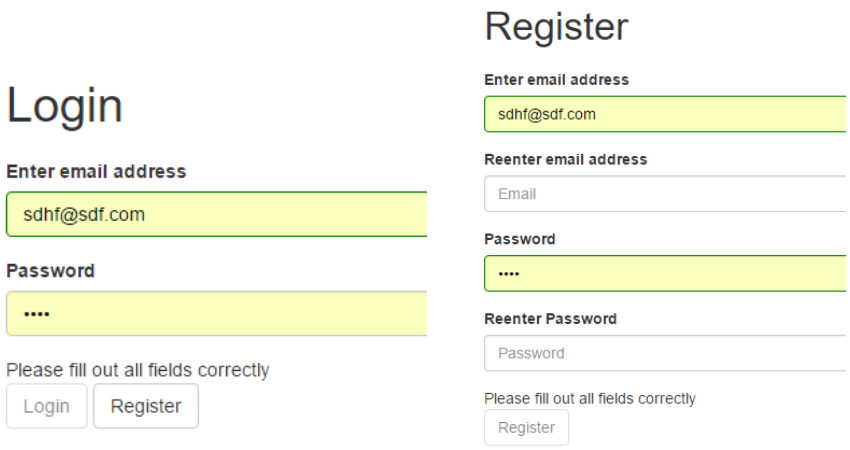
1 Introduction

The initial web application I was hoping to build would build on my previous project and add some data visualisations. I intended to use D3 to create the visualisations as it provides a lot of advanced features. I also intended to create an elastic search cluster that I would host on an Azure cloud VM this proved to be a lot of effort and I eventually had to abandon this idea due to time constraints. I expanded my database to include games consoles that would be used as the basis of my data visualisations.

2 Design

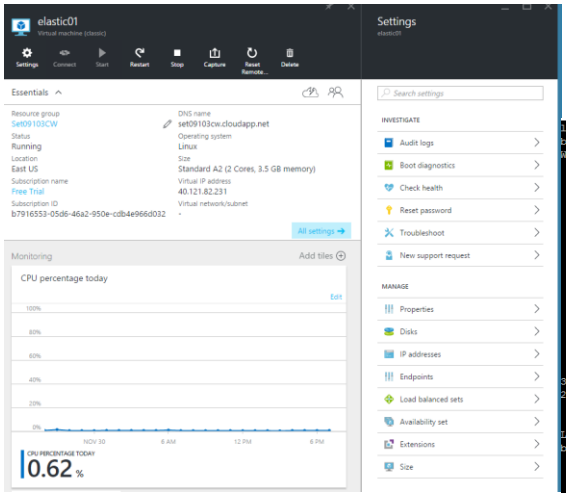
The structure of the web application revolves around a RESTful API, which provides a JSON data source for the client side application to consume and render it to the clients browser. The client side application makes requests to the python flask application running on the server and the server returns the collection or single entity that was requested.

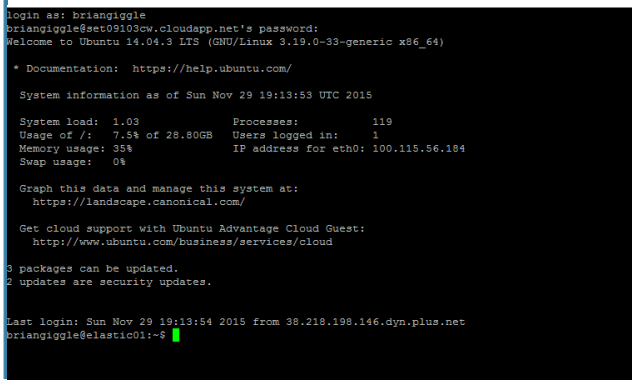
I also added the ability to create accounts and login with existing ones, the email address and password that the user chose to use to create are stored in a database after first hashing the password with a salt using bcrypt. If the user is not logged in they are redirected to the login page where they can choose to log in or create a new account.



Once logged in the user is redirected to the main list of games, from here the user can navigate to the visualisations page (Platforms). This gives the option to choose from 3 different data visualisations. Due to (the unfortunately) static data I decided to try to use browsers localstorage to save the data so that it only needs to be downloaded once, then the users browser is used to render the data.

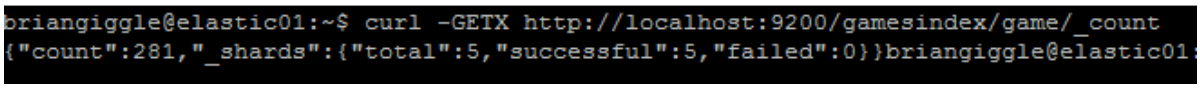
A large portion of the time I spent on the project was used trying to set up an elastic cluster that could be accessed by web app due to the large amount of data that I would have liked to use to create more visualisations/improvements in the future. I used azure to set up and host an Ubuntu virtual machine to use as a host for the elastic cluster.





I also managed to set up elastic search on this machine,

Where my project fell through was that I could not access the port 9200 on the machine I had set up from outside of the machine, this would have been fine if I did not need to demo the web app or allow it to run standalone. My next plan was to set up a flask application on this VM that would act as a gateway to the elastic cluster, this would be better than opening up port 9200 to the internet as then anyone could alter the data, after my first test seemed to work fine. My next challenge was to index the data I wished to make accessible, this is where I met another hurdle after indexing the 47000 games, I checked the elastic index to find only 281 documents had been indexed, it was at this point that I had to abandon this idea due to time constraints.



3. Enhancements

3.1 SQL security

Currently my web application will execute any sql that is in the filters which is a big security concern, this would have been fixed by using elastic search with a service in front of it to only allow data to the correct users. It would also only expose get methods with no way for a user to inject anything malicious

3.2 Error handling

I would also like to add error handling that would handle errors and a meaningful way and present them to the user.

3.3 More Visualisations

I would also have liked to add different visualisations that would display different sets of data, in more varied ways, along with giving the user more control over what data is visualised.

4. Critical Evaluation

I am a little disappointed with my final result as I had planned to create something more elaborate, but due to the issues detailed above I wasn't able to achieve. I also would have liked to create some more advanced visualisations but I found the learning curve for D3 to be quite high, I feel given some more time I could have produced a much better web application.

5. Personal evaluation

Although I am disappointed with the outcome I have learnt a lot about using a cloud service to host a VM that could be used as a server, I also used SSH to control the setup process of an elastic cluster on this server. I also managed to host the site on this server using nginx and Gunicorn, it can be accessed: http://set09103cw.cloudapp.net/

I am very happy with what I learnt about actually deploying a web application to a cloud service throughout the course of this course work.

6. Resources

Flask: http://flask.pocoo.org/

Levinux: http://mikelev.in/ux/

SQLite3: https://www.sqlite.org/

AngularJS: https://angularjs.org/

JQuery: https://jquery.com/

Bootstrap: http://getbootstrap.com/

Angular Bootstrap: https://angular-ui.github.io/bootstrap/

GiantBomb Api: http://www.giantbomb.com/api/

Azure: https://azure.microsoft.com/en-gb/

Elastic: https://www.elastic.co/

7. References Dr Simon Wells, Notes and workbook:
http://moodle.napier.ac.uk/pluginfile.php/955615/mod_resource/content/10/workbook.pdf

SQLite Documentation:

https://www.sqlite.org/docs.html

AngularJS Documentation:

https://docs.angularjs.org/api

Angular Bootstrap Documentation:

https://angular-ui.github.io/bootstrap/

GiantBomb api Documentation:

http://www.giantbomb.com/api/documentation