***Dissertation:***

***(Stephen’s section)***

**1st way:**

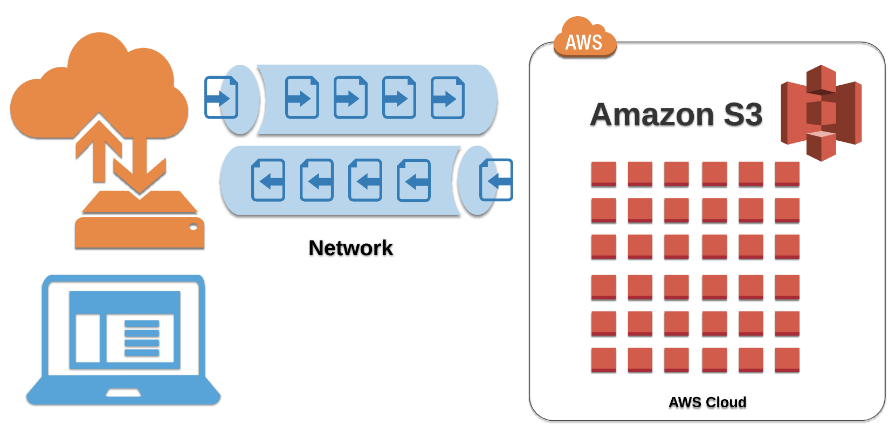
HTML & CSS files to aws amazon:

When setting up the application to be hosted by aws amazon I had no prior knowledge behind the hosting service. This which had me doing extensive research into the hosting providers.

I Had to research the following before uploading anything: S3, route53, Ec2. These which I narrowed it down to after looking up what I would need from amazon’s web services.

To begin I researched S3 buckets. This after a few posts and videos I became aware that it is amazon’s way of storing files. My understanding at the beginning was limited to seeing just files but after I could see that you could restrict access to these files. This which I learned is to protect the access of the files which can be retrieved from anywhere on any device through the web if needed.

Below Showing a Diagram which I believe describes the process.



Following this research, I began looking at Route 53 which I could understand I would need for the files I would be uploading. Route 53 being amazons widely available and scalable cloud domain name system (DNS) service. This which I learned helps host your files to the web for the public to see on any browser worldwide.

**NEXT TO EXPLAIN:**

* Two issues as was not able to see website after uploading all the files :

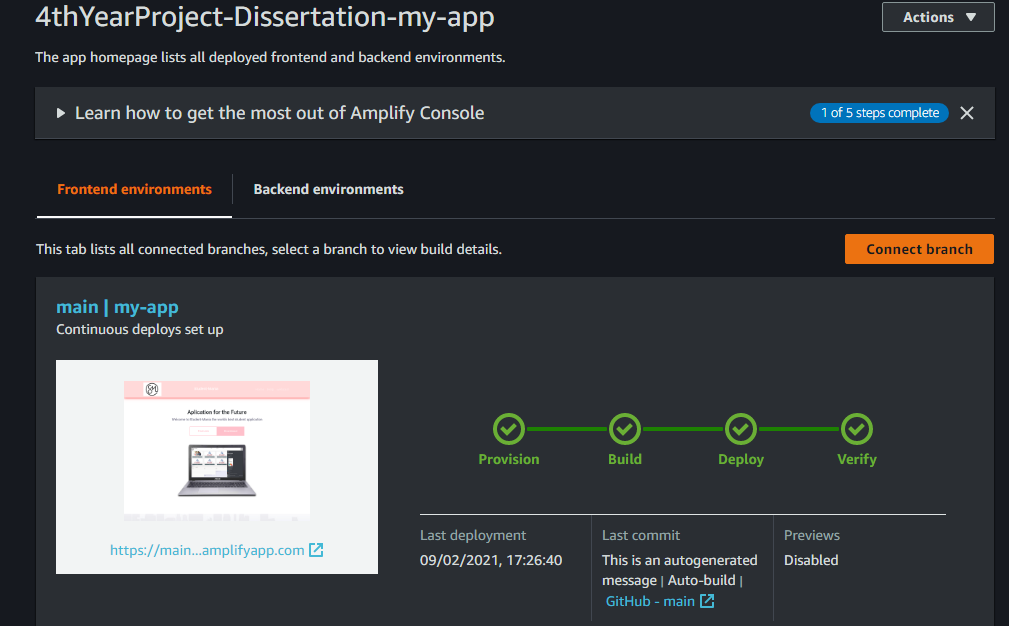
Permissions tab in the s3 bucket– make files static to be able seen on web

Also have to unblock public access to allow public to see website.

* Route53
* record A being for an alias which connects the bucket with html files to the URL
* Record (SOA) and name of server(NS) being provided by amazon
* CNAME setup when directing the bucket to the new [www.student-mania.com](http://www.student-mania.com) domain

**2nd way:**

HTML & CSS & JavaScript & react to amazon aws:



Uploading to amplify I discovered that I can only host the application to amplify if I was the admin holder of the repository of the project. I soon then uploaded the project to amplify but came into some issues. Here where I wanted to be able contact customer support but due to them removing the live chat functionality unless you pay for a plan, I was unable to access. I then had to research why I was getting the error “page cannot be found error 404”.This which I knew to be a linking issue as it was there but couldn’t be located. This which after extensive research I didn’t find the solution. I found that this was not covered enough on online forums or anything.

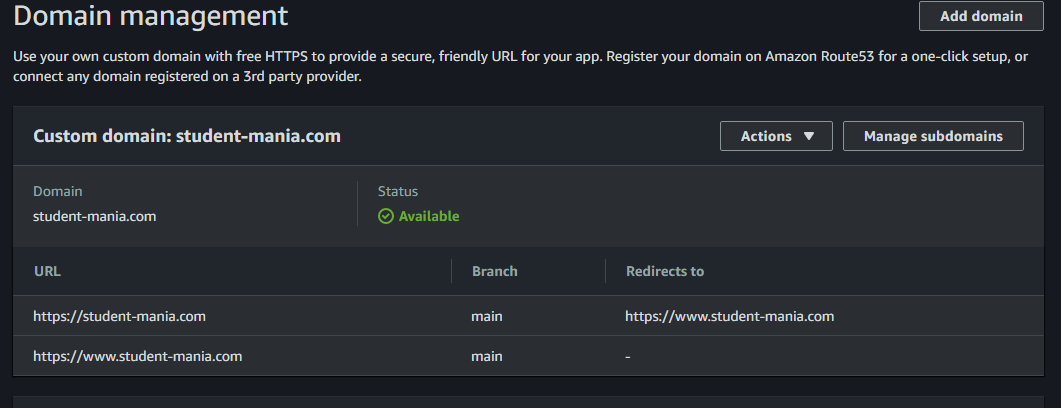
The solution I found was that when you are uploading to aws amplify and linking your repository to it you must click on the small box at the end that allows the admin to specify the exact folder of the application within your repository folder. This which is not covered or explain to be “needed”. This which I thought was an extra feature if wanted. This which I also put two and two together as when I was researching, I came across that when you are uploading the repo to amplify that when amplify discovers your app it looks for your package. JSON file. This which tells amplify all the commands it needs to build the application.

Once I got round the issue of error 404, I had the application hosted by aws amplify ass seen running through the deploying stage of amplify.

During the second way It was unfortunate that amazon started to set up plans for users to have to pay for help off customer service if you want help in other ways than live chat. This feature which I used at the end of the first way of going about uploading the application website to aws amazon. As after I uploaded everything I was still unable to see the website but on live chat I was helped within minutes .Being explained to me that it can take up to 48hrs to fully upload and be visible on the web with the URL they provide.

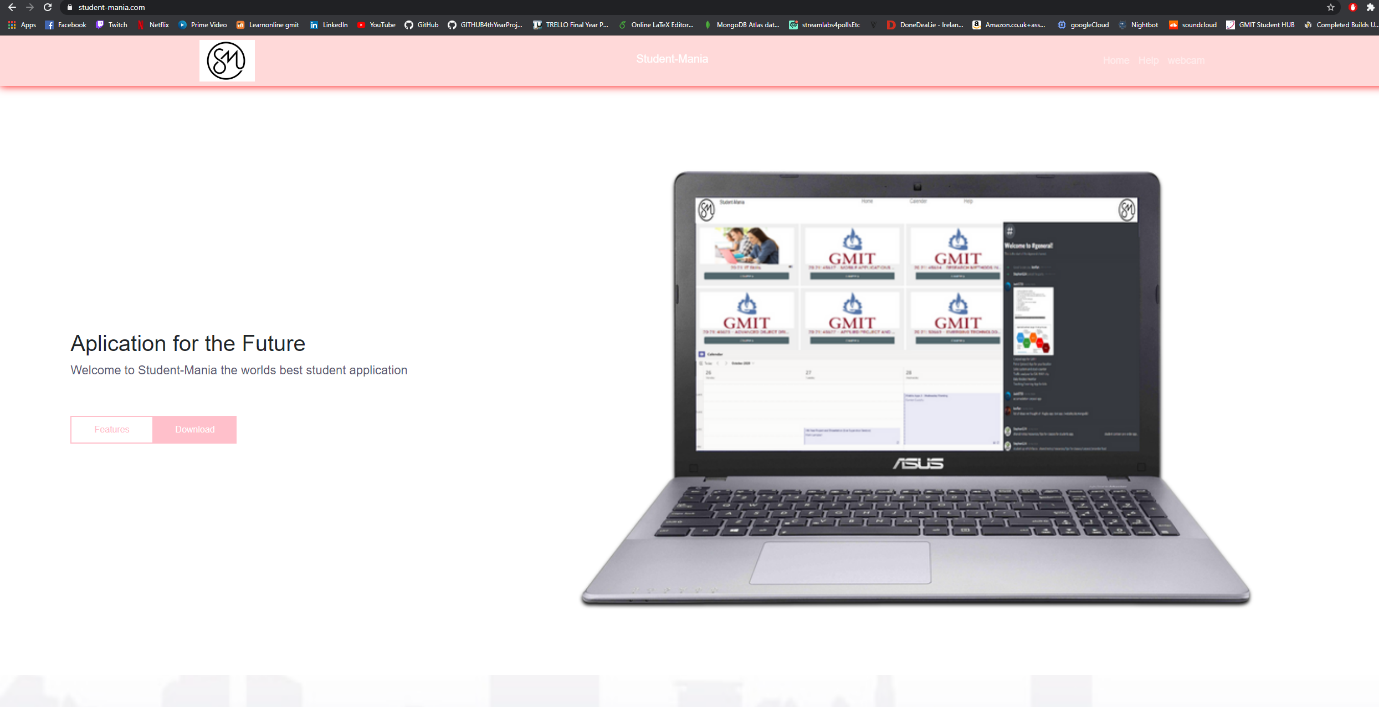
**Linking aws amplify with custom domain –**

In my learning of linking your domain to amplify I learned that when you setup a route53 it provides you with the 4 name servers and SOA .But when you go to your “Domain management” on your aws amplify and select the route53 you setup to link the two it generates the “CNAME” and “ALIAS(A)” for you. This which had to be done manually when not using aws amplify.



This which since I have done both ways, I know what amplify does for its users, so they don’t have to setup everything themselves.

In the process of adding my custom servers to my GoDaddy domain I ran into an error when I clicked the save button. This error which was “an unexpected error”. This when I looked up came back to be a common error. This which suggested to access it on a incognito window or clear cache. This which did not work for me and ended up having to contact customer service via live chat.

I was happy to hear they had live chat but unfortunately the customer service person did not provide me with a reason behind the error. The only solution was that he entered the name servers into the account for me and save it then. This which I found strange as I got my other group teammate to try the same but he also experienced the same error.After all the issues by the end of the day I was happy to say I got the website up and running and searchable through google for the second time using a different approach.

|  |  |
| --- | --- |
| Pros: | Cons |
| Updates online App when you commit to GitHub | If Errors in code wont build. |
| Don’t have to do the linking between files and servers, does it for you |  |
| Makes you app public by default |  |
| Can add tests to application when uploading if you wished for in the build process |  |
|  |  |
|  |  |
|  |  |

Amplify is very useful when making changes to your application as you don’t need to reupload your files to your s3 bucket to update your online application. It simply automatically rebuilds your application when you make a change/commit to your linked GitHub repository. This which saves time and helps in faster development of applications.

**AMPLIFY**