

Software and Applications Needed

1. Amazon Web Service account (AWS)
2. EC2 Ubuntu instance hosted on AWS
3. Firebase
4. Notepad++ and Sublime text editors
5. GitHub
6. Google Docs / Google Drive
7. FileZilla
8. Google Cloud Storage Bucket
9. Apache Ubuntu instance

Deployment

To begin we used an Amazon Web Service account (AWS) to host our web service. We had originally made an EC2 Microsoft instance, but eventually changed to an Ubuntu instance, because it is easier to debug log files from Linux through terminal as opposed to a Microsoft service. We had also begun with an Elastic File System to hold our JSON file in the database, but we found we didn't need this because both MongoDB and Firebase hold JSON files. We then used FileZilla and WinSCP to upload our web page files to our AWS instance. We decided to change from a MongoDB to a Firebase database, because it is easier to handle state across the site with JSON and JavaScript as opposed to PHP. Firebase is a NoSQL database like MongoDB. It is hosted on its own server through Google. We used GitHub to collaborate on code within our branches to push and pull code, and to create our Markdown document within the readme file. We also used Google Docs and Google Drive to share our requirements documents and sprint information. We decided to use a Google Cloud Storage Bucket to hold the JSON files for the database.

Deployment

1. Create an Amazon Web Service Account (AWS)
 - Choose 'Services' from the top dropdown menu
 - Click 'Launch Instance'
 - Scroll down the page, and choose the Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-01f05461 (this one is easier to debug issues with a MAC on terminal)
 - Choose the t2.micro (free tier)
 - Choose 'create a new key pair' from the dropdown menu
 - Name the key pair
 - Download, and save the key pair to your desktop
 - Click 'Launch Instance'

- Navigate to the 'EC2 Dashboard', and choose 'Running Instances' to ensure that the instance is running correctly (may need to log out of the AWS account and back in to refresh the page)
2. Then, use a file sharing system (we used both FileZilla and WinSCP) to host our website files to the root folder of the EC2 instance these files are located on GitHub in the DML folder

- You will need the credentials from the AWS Ubuntu instance to login to the instance through the file sharing system
- Download our files from the DML folder on GitHub to your desktop
- To be able to get all of our file to connect to your new Firebase database you will need to change the key at the top of file in app.js to your credentials that are located within Firebase here is a picture(this will be explained in setting up the database) #

```
(function () {  
  
    //init firebase  
    var config = {  
        apiKey: "AIzaSyDhctdgHv0NFpEVrvF9FyFkmLHr2z_bajg",  
        authDomain: "software2016-9e4fa.firebaseio.com",  
        databaseURL: "https://software2016-9e4fa.firebaseio.com",  
        storageBucket: "software2016-9e4fa.appspot.com",  
        messagingSenderId: "971715155403"  
    };  
    firebase.initializeApp(config);  
  
    console.log("in here");  
  
    //getting elements
```

- Drag and drop them from your desktop to the root folder of the EC2 instance on the file sharing system
3. Next, set up the Firebase database by creating a Firebase account (you will need a Gmail email account)
- Click 'Create New Project' on the main page: Name the project and choose the United States from the dropdown box as the region, and then click 'Create Project'
 - Change the Authentication by going under 'Authentication' from the side bar: Choose the 'Sign-In Method' from the top menu, and then enable the 'Email/Password' option since this is what we use for our login
 - Click 'Storage' on the menu to the left: Choose 'Rules' from the top menu, then change the allow section to 'allow read, write;', and finally click 'Public'
 - Click 'Overview' in the side menu: Click 'Add Firebase to your App', and then copy and paste it to the app.js file
 - Click Database on the side menu bar: Click Data at the top menu bar, Click the green '+', Click the next green '+', give the first one the name of 'uploads', name the next one whatever you want and add a value of whatever you want, and click 'Add'

