

Search engine for indoor environment data using
ElasticSearch and front end search UI using React.

CS410 FINAL PROJECT

NETID: NADIAW2

Table of Contents

<u>ABSTRACT</u>	<u>2</u>
<u>DEMO APP</u>	<u>2</u>
<u>TECHNICAL ARCHITECTURE</u>	<u>3</u>
<u>SETTING UP ELASTIC SEARCH IN GOOGLE CLOUD.</u>	<u>3</u>
<u>DATA PIPELINE.....</u>	<u>10</u>
<u>FINE TUNING THE ENGINE</u>	<u>11</u>
<u>CREATING A SEARCH UI TO SEARCH DATA.....</u>	<u>12</u>
<u>SETTING A DEVELOPMENT ENVIRONMENT LOCALLY ON YOUR COMPUTER.....</u>	<u>12</u>
<u>CREATING A UI FOR SEARCH EXPERIENCE BY USING APP SEARCH PACKAGES.....</u>	<u>14</u>
<u>CREATING A SEARCH EXPERIENCE</u>	<u>14</u>
<u>DEPLOYING THE REACT APP TO GOOGLE CLOUD PLATFORM.....</u>	<u>17</u>
<u>CREATE THE APP ON THE APP ENGINE</u>	<u>17</u>
<u>DEPLOY THE APP</u>	<u>21</u>

Abstract

In the age of sensors, devices and platforms collecting millions of datapoints every second, it comes necessary to be able to sift through all the data to develop insights efficiently Big data offers the solution for analyzing large amount of data and using the technique of Elasticsearch, access to data can be made faster.¹

I will be creating a web application to use ElasticSearch to search content from a set of documents of environmental datapoints collected by sensors (indoor air, humidity, temp etc). Currently, it is difficult to search for data in a RDMS database and it takes significant time using traditional SQL queries. The project will take the data transfer it to ElasticSearch server. The front end written in React will allow users to search for data. Measurable outcomes are going to be the amount of time it takes to run a query against a traditional RDMS database vs. using ElasticSearch. The planned architecture is shown in Figure1.

Demo App

https://cs410-env-search-app.uc.r.appspot.com/?size=n_20_n

¹ Gujarat, India, Darshita Kalyani, and Dr. Devarshi Mehta, "Paper on Searching and Indexing Using Elasticsearch," *International Journal Of Engineering And Computer Science*, June 30, 2017, <https://doi.org/10.18535/ijecs/v6i6.45>.

Technical Architecture

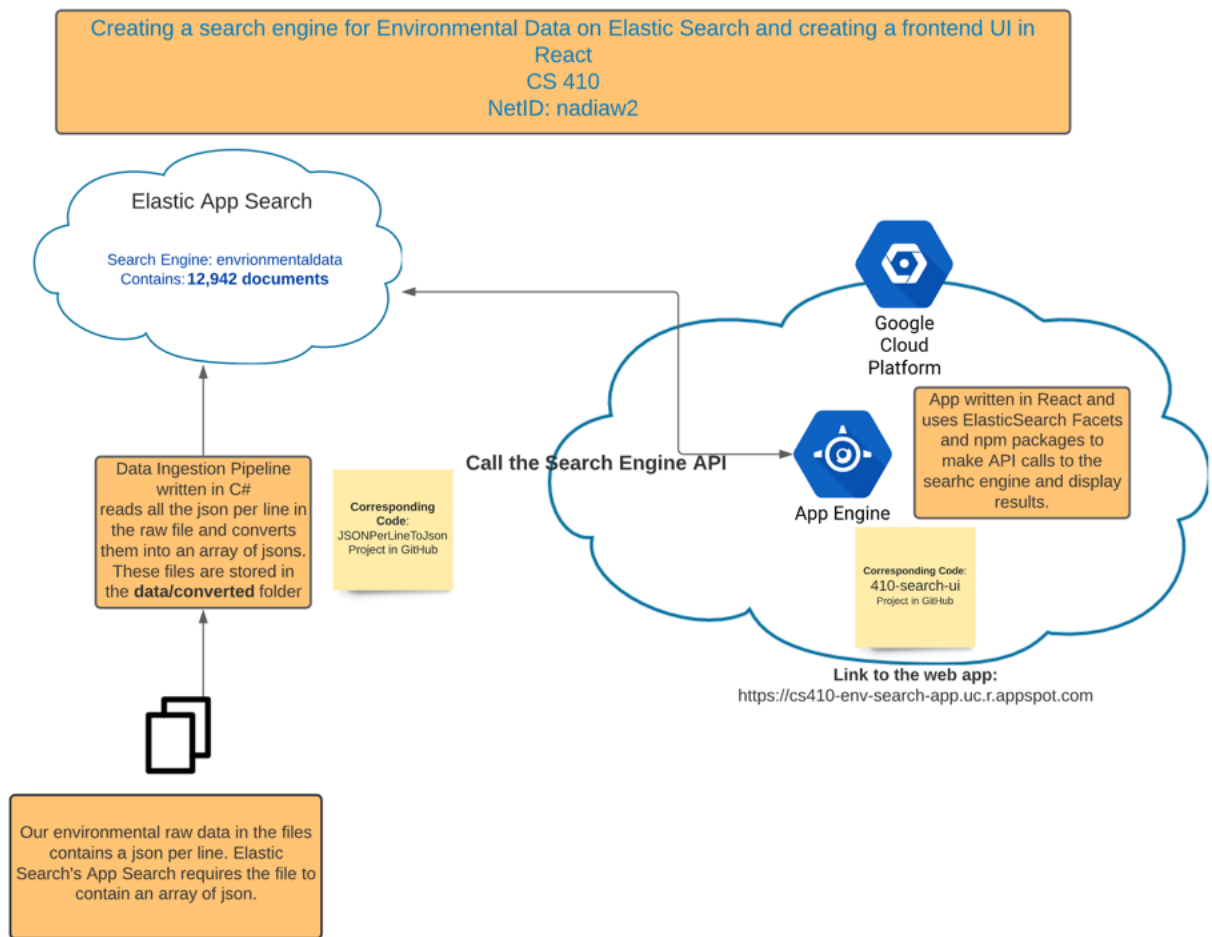
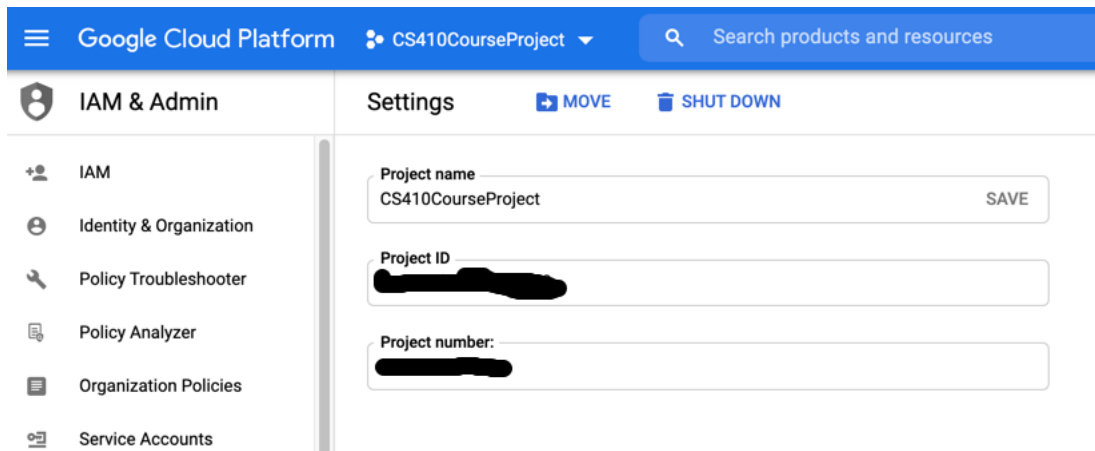


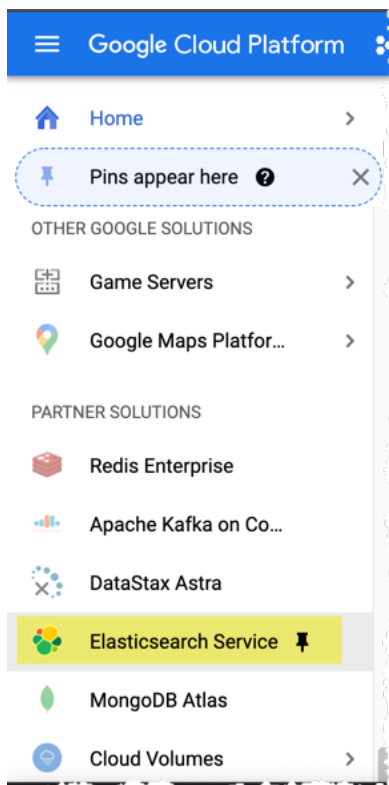
Figure 1

Setting up Elastic Search in Google Cloud.

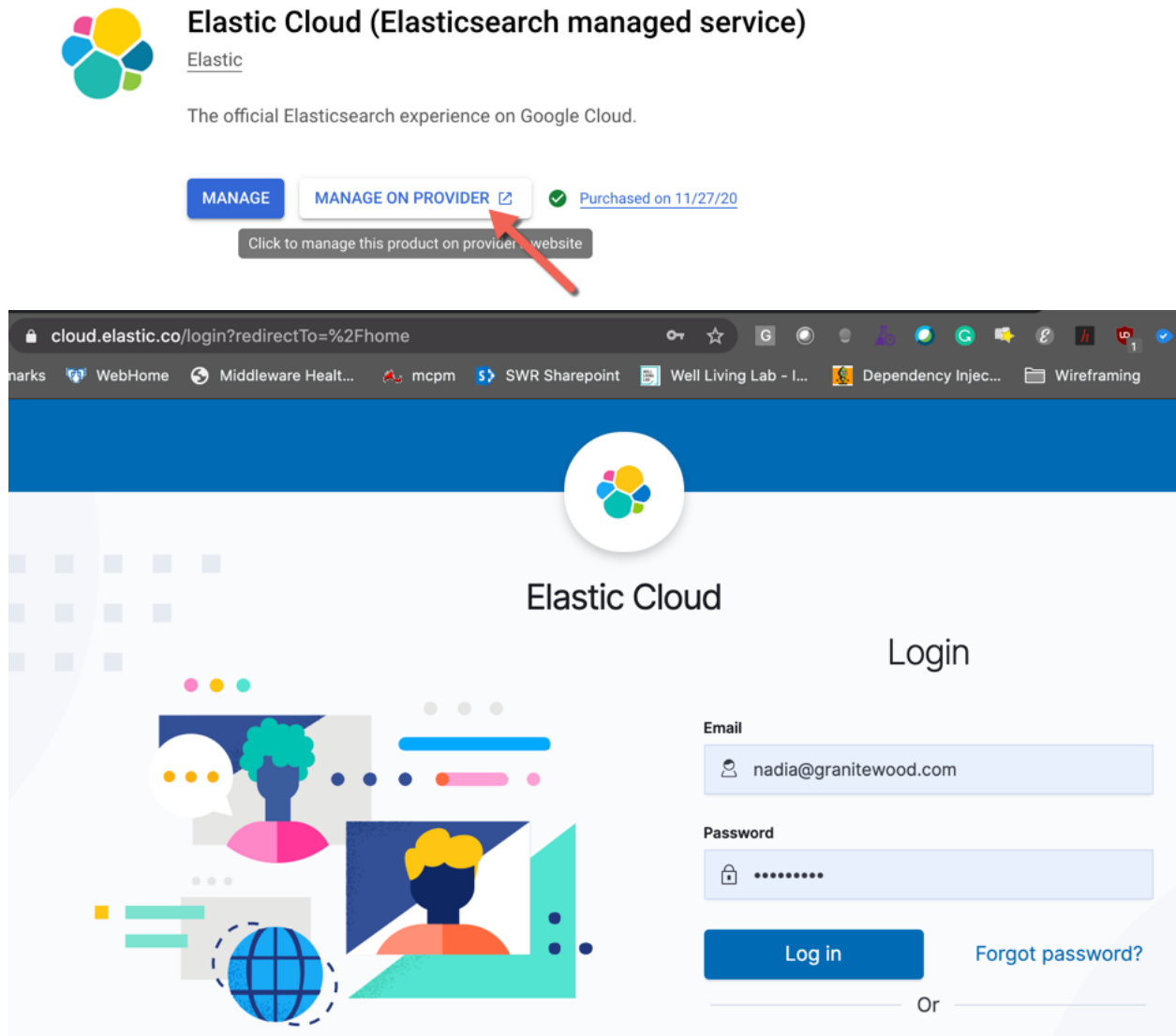
1. Go to <https://console.cloud.google.com/> and create a project on Google Cloud.



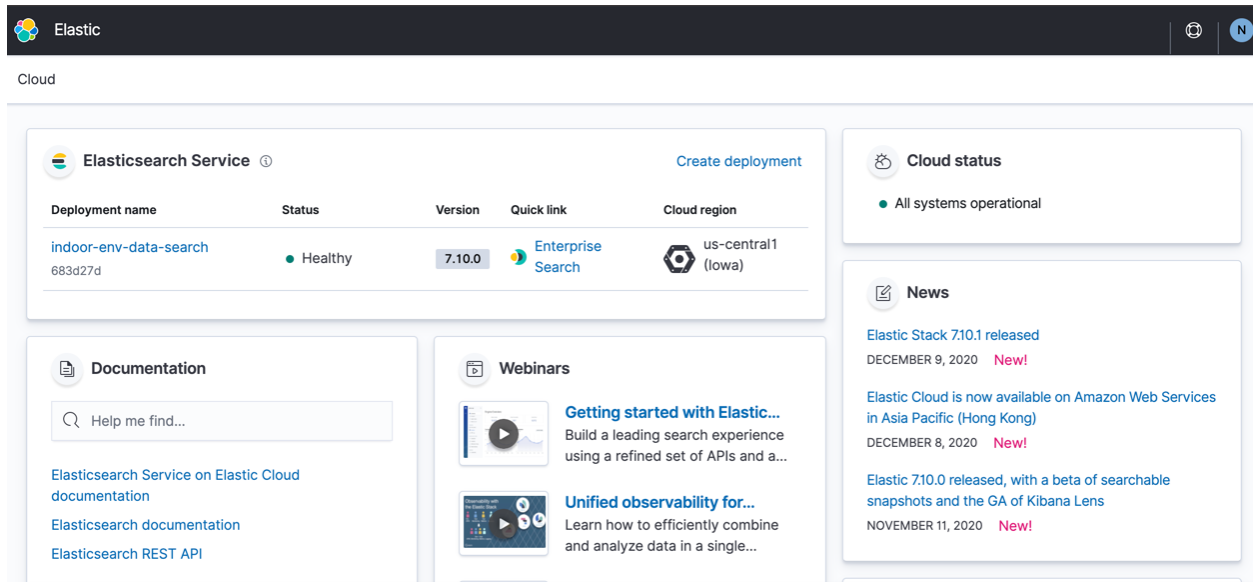
2. Choose Elasticsearch Service. The only reason I chose this is to get some free credit to do my project work. You can create a separate account on Elasticsearch if you want to but the trial only lasts for 14 days.
3. A little bit about, Elasticsearch Service on Google Cloud: The service offers seamless integrated billing through your Google Cloud account for simple management and powerful customization.



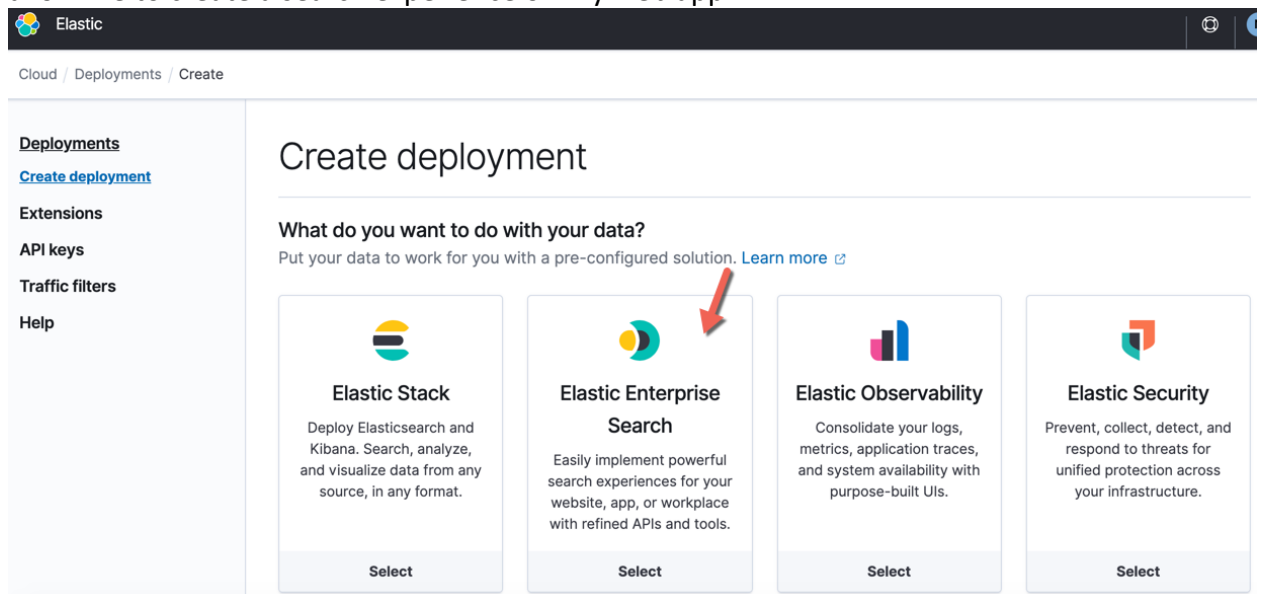
- Once the service is setup, you can click on manage on provider, to go directly to Elastic Cloud, to manage and create a search deployment on the cloud.
<https://cloud.elastic.co/>



- Once you login, you will be taken to the Elastic Cloud dashboard. Here you can create your “deployments” .

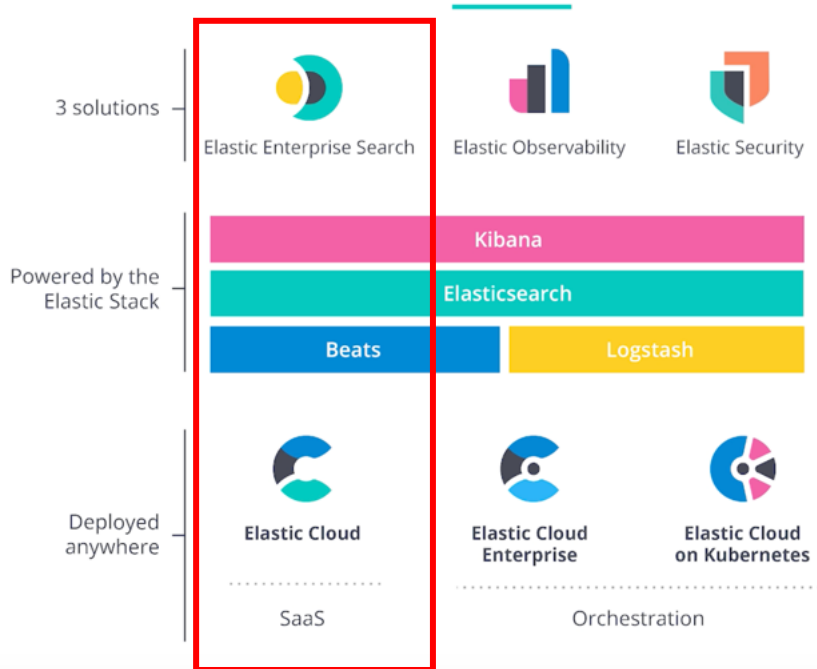


- When you create a deployment, you are given a choice of selecting from pre-configured environments for your need. In my case, I chose the Elastic Enterprise Search solution to allow me to create a search experience on my web app.



Elastic technology provides the following stack options. For my project I am using the stack outlines in red. This stack gives me the Elastic Cloud, which gives me the ability to make RESTFUL API calls to my search engine.

Elastic Technology



7. Once your deployment is created you will be taken to the deployment dashboard. In this project, we will be focusing on using Enterprise search capability.

Cloud / Deployments / indoor-env-data-search

Deployments

- indoor-env-data-search
- Edit
- Elasticsearch
 - Snapshots
 - API console
- Kibana
- APM
- Enterprise Search
- Logs and metrics
- Activity
- Security
- Performance

Extensions

API keys

Traffic filters

Help

indoor-env-data-search

us-central1 (Iowa)

Creating your deployment

Your deployment will be ready in a couple of minutes.

Get started with your deployment

The next step is to customize your search experience.

[Open Enterprise Search](#)

Ready in a few minutes

Forgot to save your credentials?
[Reset your deployment password](#)

Deployments

indoor-env-data-search

Edit
Elasticsearch
Snapshots
API console
Kibana
APM
Enterprise Search
Logs and metrics
Activity
Security
Performance

Extensions

API keys

Traffic filters

Help

indoor-env-data-search

us-central1 (Iowa)

Deployment name

indoor-env-data-search

Edit

Deployment status

● Healthy

Open Kibana

Manage

Deployment ID: 683d27d

Deployment version

v7.10.0

Upgrade

Applications

Elasticsearch / Copy endpoint

Kibana / Launch / Copy endpoint

Enterprise Search / Launch / Copy endpoint

Cloud ID

```
indoor-env-data-  
search:dXmtY2VudHJhbDEuZ2NwLmNsb3VkLmVzLm1vJGI4ZmQzOT  
U0NTQxZjQ3ZDVhYjc3MzQyZWUwMjJhM2ZjJGFhZTc5NGFjOGVhZTQ  
5NjJiNzFkZWZjMzd1MzZmODcx
```

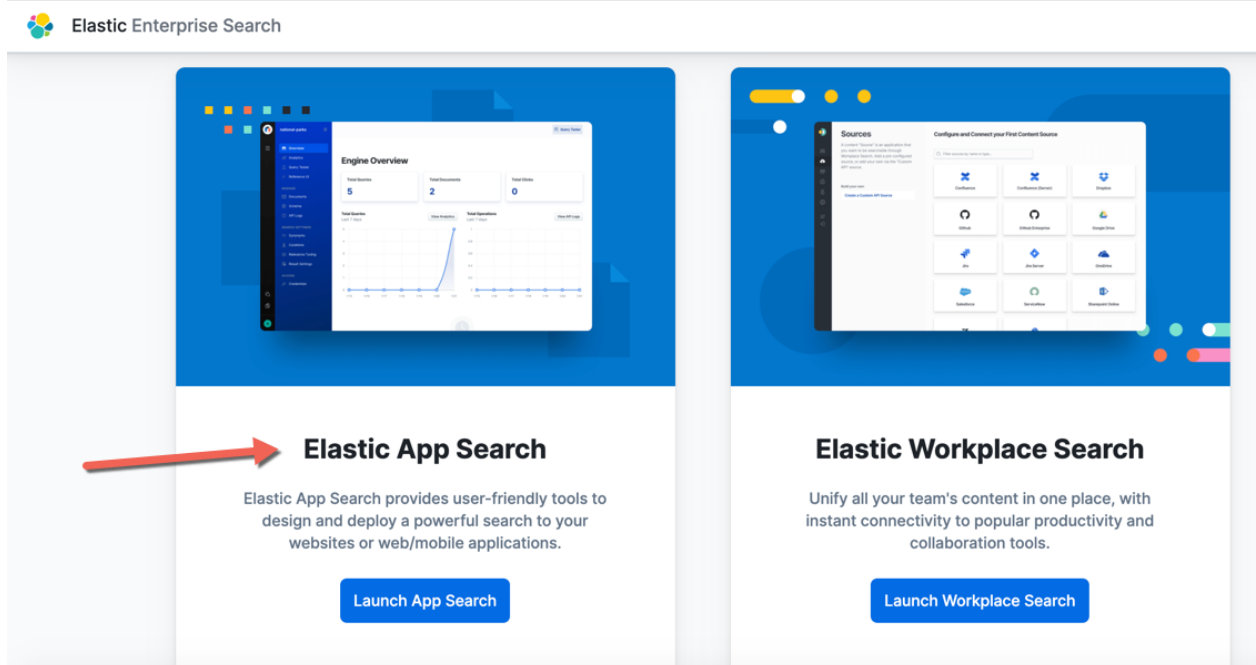
Instances

Instance configuration

Health

8. Once you launch Enterprise Search, it will give you an option to select a product. For this project, I used App Search.

Elastic Enterprise Search



Elastic App Search

Elastic App Search provides user-friendly tools to design and deploy a powerful search to your websites or web/mobile applications.

Launch App Search

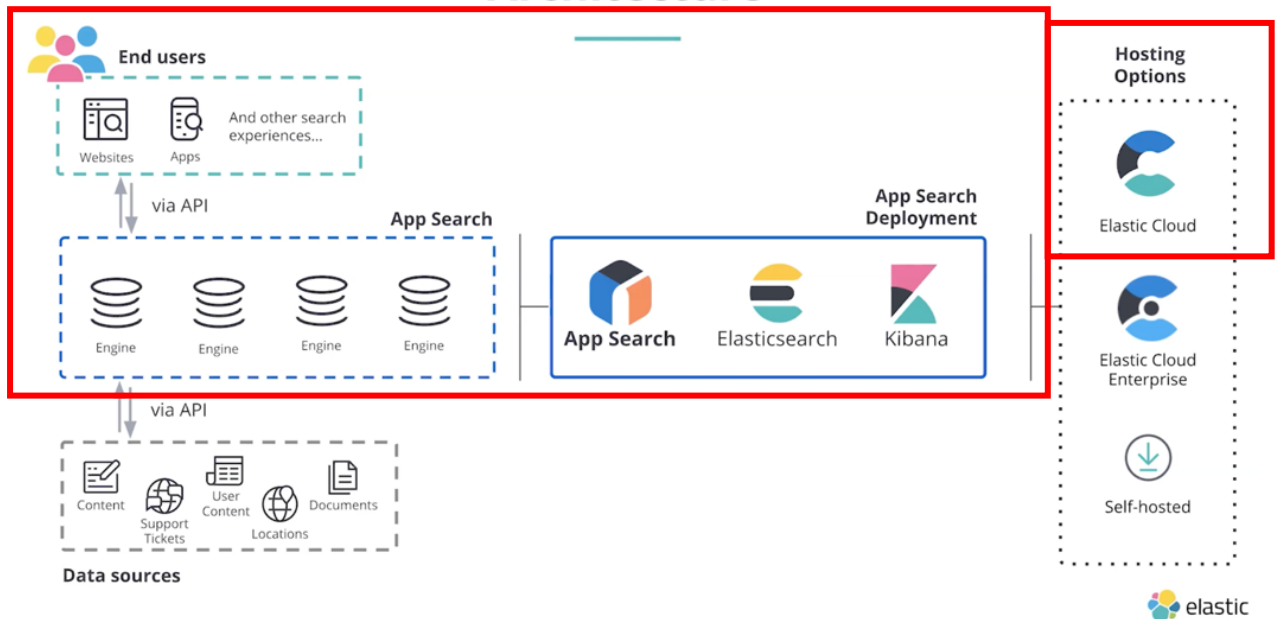
Elastic Workplace Search

Unify all your team's content in one place, with instant connectivity to popular productivity and collaboration tools.

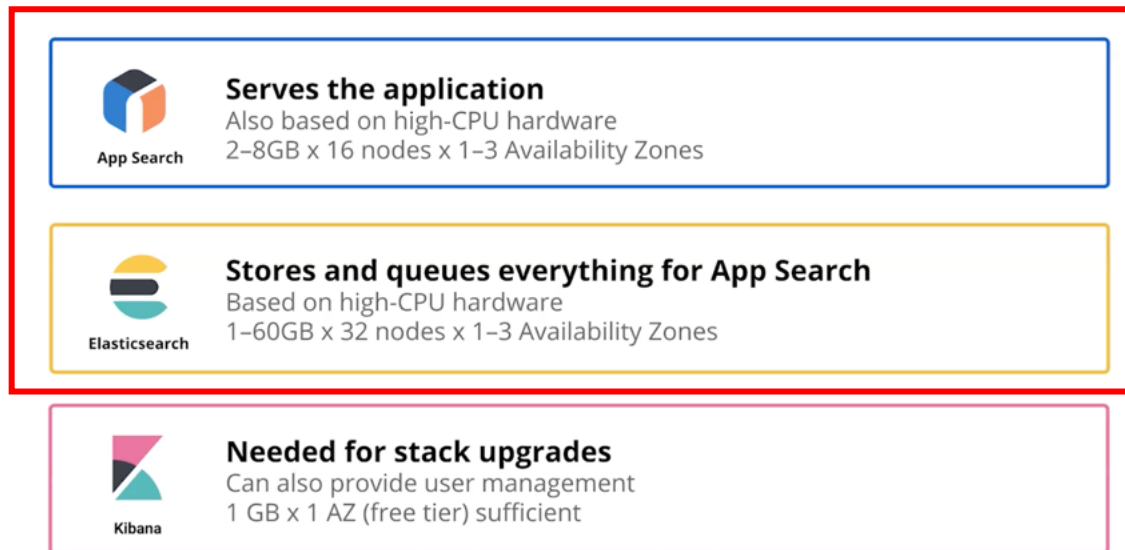
Launch Workplace Search

9. An overview of the architecture of App Search stack is below. I have highlighted the architectural components which are being utilized for this project.


Architecture



Architecture




10. Once the App Search is launch, it gives you an option to create a search engine.

 App Search
v7.10.0

[Engines](#)
[Meta Engines](#)
[Credentials](#)
[Users & roles](#)
[Settings](#)
[Sign Out](#)

Engines

Create an Engine


Name	Created	Documents	Fields	
environmentaldata	November 27, 2020	12942	30	Manage 


11. For this project, I created an engine called `environmentaldata`. This search engine allows a user to search through documents which contain sensor data e.g. humidity, temperature, light, battery info etc. Currently the engine contains xxxx documents. The documents were loaded into the engine by uploading JSON file to the engine.

Add new documents to `environmentaldata`

There are three ways to send documents to your Engine for indexing. You can paste raw JSON, upload a .json file, or POST to the [Documents API](#) endpoint.

Click on your choice below or see [Indexing by API](#).


Paste JSON


Upload a JSON file

12. The engine requires a specific formatting for the json files to adhere to. The JSON field names have to be all lowercase or be separated by underscore. This created a need to automate the conversion of existing json files to be converted to the required format.

Data Pipeline

13. **Data Ingestion program in C#:** In order to quickly load json files to the engine, I created a program in C# to convert existing files to a proper json file so that it can be imported into App Search. This code can be run if you have Visual Studio free community version installed. The program requires to have a “data” folder where the files needed to be converted need to be stored. The converted files are stored in the “data/converted” folder. I have included some converted file in the repo as well:
<https://github.com/nadiawoodninja/CourseProject/tree/main/data/converted>

Build successful.

Search: Press 'F3' to search

Solution: JSONPerLineToJson (main)

- JSONPerLineToJson
 - Connected Services
 - Dependencies
 - Program.cs

Program.cs

```

1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 using System.Linq;
5
6 namespace JSONPerLineToJson
7 {
8     class Program
9     {
10         static void Main(string[] args)
11         {
12             foreach (var f in Directory.EnumerateFiles("/Users/nadiawood/Documents/GitHub/CourseProject/da
13             {
14                 if (f.StartsWith(".")) continue;
15                 if (!f.EndsWith(".json")) continue;
16
17                 var lines = File.ReadAllLines(f);
18                 var output = new List<string>();
19                 output.Add("[");
20                 output.AddRange(lines.Select(line => line.ToLowerInvariant() + ","));
21
22                 string last = output[output.Count - 1];
23                 last = last.Substring(0, last.Length - 1);
24                 output[output.Count - 1] = last;
25
26                 output.Add("]");
27                 File.WriteAllLines(f.Replace("/data", "/data/converted"), output);
28             }
29         }
30     }
31 }
32

```

main CourseProject / JSONPerLineToJson /

Go to file Add file

This branch is 20 commits ahead of CS410Fall2020:main. Pull request Compare

nadiawoodninja final project submission 19505fd 15 hours ago History

File	Commit	Message	Time
..	19505fd	final project submission	15 hours ago
./JSONPerLineToJson/xs	19505fd	final project submission	15 hours ago
JSONPerLineToJson	19505fd	final project submission	15 hours ago
JSONPerLineToJson.sln	19505fd	Progress Report	14 days ago

Fine Tuning the Engine

- Once the documents are loaded into then engine, you can index any JSON object. The json object will become a search-optimized document within your Engine. A schema is created for you when you index your data - you do not need to specify any schema or alter your data before uploading. You can alter your schema later to set the appropriate data types.

Manage Engine Schema

Add new fields or change the types of existing ones.

Create a Schema Field

Update Types

id

• batterycurrentvoltage	Recently Added	text	▼
• coord_phi	Recently Added	text	▼
• coord_theta	Recently Added	text	▼
• dataunits	Recently Added	text	▼
• batteryminimumvoltage	Recently Added	text	▼
• studyname	Recently Added	text	▼
• datatype	Recently Added	text	▼
• datavalue	Recently Added	text	▼
• ...	Recently Added	text	▼

15. You also have the option to refine search by using features like, Relevance Tuning, Synonyms & Curations. For this project I utilized the Synonym feature as we may have data from different sensors and the same datapoint maybe spelt differently or represented differently.

Synonyms

slope

≈ clope

Manage

Creating a search UI to search data

Setting a development Environment locally on your computer.

16. Download and install Node.js from <https://nodejs.org/en/>
17. Once installation is complete run this command. We are going to use this to create a react app.
- ```
npm i -g create-react-app
```

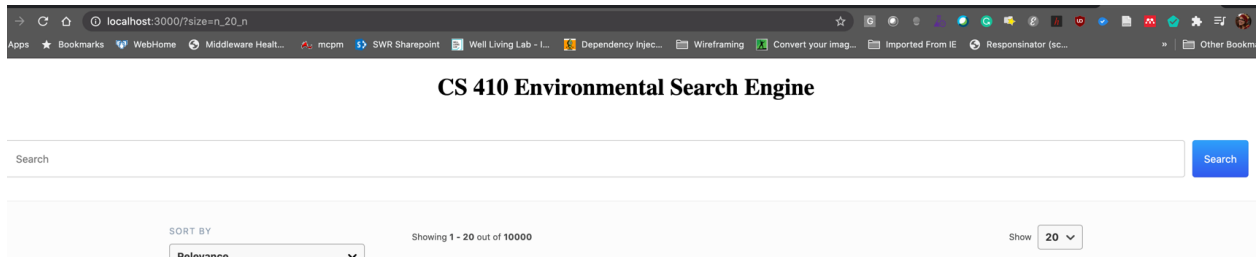
```
CourseProject — -bash — 113x27
CS410ProjectDocumentation
CS410ProjectDocumentation.docx
CS410ProjectProgressReport.docx
CS410ProjectProgressReport.pdf
EnvData_ElasticSearch_ProjectProposal.docx
EnvData_ElasticSearch_ProjectProposal.pdf
JSONPerLineToJson
README.md
architecture.png
data
~$410ProjectDocumentation.docx
(base) nadias-mbp:CourseProject nadiawood$ ls -l
total 4520
drwxr-xr-x@ 3 nadiawood staff 96 Dec 12 18:57 CS410ProjectDocumentation
-rw-r--r--@ 1 nadiawood staff 1772487 Dec 12 18:57 CS410ProjectDocumentation.docx
-rw-r--r--@ 1 nadiawood staff 72380 Nov 30 09:18 CS410ProjectProgressReport.docx
-rw-r--r--@ 1 nadiawood staff 167650 Nov 29 14:45 CS410ProjectProgressReport.pdf
-rw-r--r--@ 1 nadiawood staff 79792 Nov 29 14:45 EnvData_ElasticSearch_ProjectProposal.docx
-rw-r--r--@ 1 nadiawood staff 151819 Nov 29 14:45 EnvData_ElasticSearch_ProjectProposal.pdf
drwxr-xr-x 5 nadiawood staff 160 Nov 27 20:42 JSONPerLineToJson
-rw-r--r--@ 1 nadiawood staff 1887 Nov 29 14:45 README.md
-rw-r--r--@ 1 nadiawood staff 51601 Nov 29 14:40 architecture.png
drwxr-xr-x 33 nadiawood staff 1056 Dec 12 17:23 data
-rw-r--r--@ 1 nadiawood staff 162 Dec 12 16:57 ~$410ProjectDocumentation.docx
(base) nadias-mbp:CourseProject nadiawood$ create-react-app 410-search-ui
-bash: create-react-app: command not found
(base) nadias-mbp:CourseProject nadiawood$ npm i -g create-react-app
```

18. Once the package is installed create the react app by running the command below.  
`create-react-app 410-search-ui`

This command installs a light weight web server, webpack to bundle our code for deployment and Babel for compiling our JavaScript code.

```
CourseProject — -bash — 113x27
(base) nadias-mbp:GitHub nadiawood$ cd CourseProject
(base) nadias-mbp:CourseProject nadiawood$ ls
CS410ProjectDocumentation
CS410ProjectDocumentation.docx
CS410ProjectProgressReport.docx
CS410ProjectProgressReport.pdf
EnvData_ElasticSearch_ProjectProposal.docx
EnvData_ElasticSearch_ProjectProposal.pdf
JSONPerLineToJson
README.md
architecture.png
data
~$410ProjectDocumentation.docx
(base) nadias-mbp:CourseProject nadiawood$ ls -l
total 4520
drwxr-xr-x@ 3 nadiawood staff 96 Dec 12 18:57 CS410ProjectDocumentation
-rw-r--r--@ 1 nadiawood staff 1772487 Dec 12 18:57 CS410ProjectDocumentation.docx
-rw-r--r--@ 1 nadiawood staff 72380 Nov 30 09:18 CS410ProjectProgressReport.docx
-rw-r--r--@ 1 nadiawood staff 167650 Nov 29 14:45 CS410ProjectProgressReport.pdf
-rw-r--r--@ 1 nadiawood staff 79792 Nov 29 14:45 EnvData_ElasticSearch_ProjectProposal.docx
-rw-r--r--@ 1 nadiawood staff 151819 Nov 29 14:45 EnvData_ElasticSearch_ProjectProposal.pdf
drwxr-xr-x 5 nadiawood staff 160 Nov 27 20:42 JSONPerLineToJson
-rw-r--r--@ 1 nadiawood staff 1887 Nov 29 14:45 README.md
-rw-r--r--@ 1 nadiawood staff 51601 Nov 29 14:40 architecture.png
drwxr-xr-x 33 nadiawood staff 1056 Dec 12 17:23 data
-rw-r--r--@ 1 nadiawood staff 162 Dec 12 16:57 ~$410ProjectDocumentation.docx
(base) nadias-mbp:CourseProject nadiawood$ create-react-app 410-search-ui
```

Once the app is created go to folder 410-search-ui and run this command. This will launch our development server on localhost:3000  
`npm start`



### Creating a UI for search experience by using App Search packages

19. Install **React Search UI** and the **App Search** connector by running these commands

```
npm install --save @elastic/react-search-ui @elastic/search-ui-app-search-connector
```

### Creating a search experience

20. I use Atom as my editor for React apps. The app folder contains **src** folder which contains all the source code. **App.js** is the main file where the program starts execution.
21. The src folder also has a config folder which contains engine.json. This file contains all the configuration needed to configure your search UI. In this file you can define your “facets”, the fields which will be displayed on your results page, your sort fields etc.

```

1 {
2 "engineName": "environmentaldata",
3 "endpointBase": "https://0189dc168aee4f4c83d3371d52e6b812.ent-search.us-central1.gcp.cloud.es.io",
4 "searchKey": "search-3vksxzxixi64t1jg9gcd9ums",
5 "resultFields": [
6 "batterycurrentvoltage",
7 "coord_phi",
8 "coord_theta",
9 "dataunits",
10 "batteryminimumvoltage",
11 "studyname",
12 "datatype",
13 "datavalue",
14 "studyid",
15 "vendordata",
16 "batterymaximumvoltage",
17 "coord_x",
18 "coord_y",
19 "slope",
20 "eventtimestamp",
21 "datavaluecalibratedsi",
22 "wllid",
23 "datavaluecalibratedq",
24 "coord_z",
25 "dataid",
26 "intercept",
27 "datasource",
28 "location",
29 "dataconnection",
30 "quadratic_c",
31 "placement",
32 "quadratic_b",
33 "quadratic_a",
34 "clope",
35 "id"
36],
37 "sortFields": [
38

```

Figure 2: engine.json

The ability to define these configurations are provided by the packages which were installed above.



```
1 //all your imports for the search experience
2 import React from "react";
3 import AppSearchAPIConnector from "@elastic/search-ui-app-search-connector";
4
5 //import components needed for the Results Page.
6 import {
7 ErrorBoundary,
8 Facet,
9 SearchProvider,
10 SearchBox,
11 Results,
12 PagingInfo,
13 ResultsPerPage,
14 Paging,
15 Sorting,
16 WithSearch
17 } from "@elastic/react-search-ui";
18 import { Layout } from "@elastic/react-search-ui-views";
19
20 //Import the default css
21 import "@elastic/react-search-ui-views/lib/styles/styles.css";
22
23 //Import any configuration helpers
24 import {
25 buildAutocompleteQueryConfig,
26 buildFacetConfigFromConfig,
27 buildSearchOptionsFromConfig,
28 buildSortOptionsFromConfig,
29 getConfig,
30 getFacetFields
31 } from "../config/config-helper";
32
33
```

Figure 3: App.js



Select from Google Cloud Platform NEW PROJECT ⋮

Search projects and folders

RECENT ALL


| Name                                | ID                   |
|-------------------------------------|----------------------|
| ✓  CS410CourseProject ?             | serious-house-296922 |
| CS410-Env-Search-App ?              | cs410-env-search-app |
| <span>████████████████████</span> ? | 337074490840         |

CANCEL OPEN

23. Once the project is created, create an App Engine application.

Google Cloud Platform My Project 73752 Search products and resources

App Engine

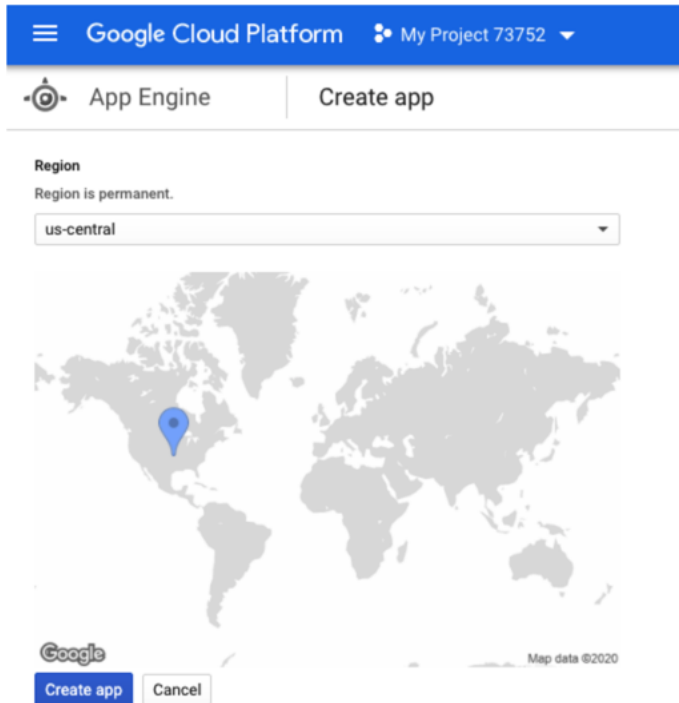


### Welcome to App Engine

Build scalable apps in any language on Google's infrastructure

Create Application

## 24. Select a region



Google Cloud Platform My Project 73752

App Engine Create app

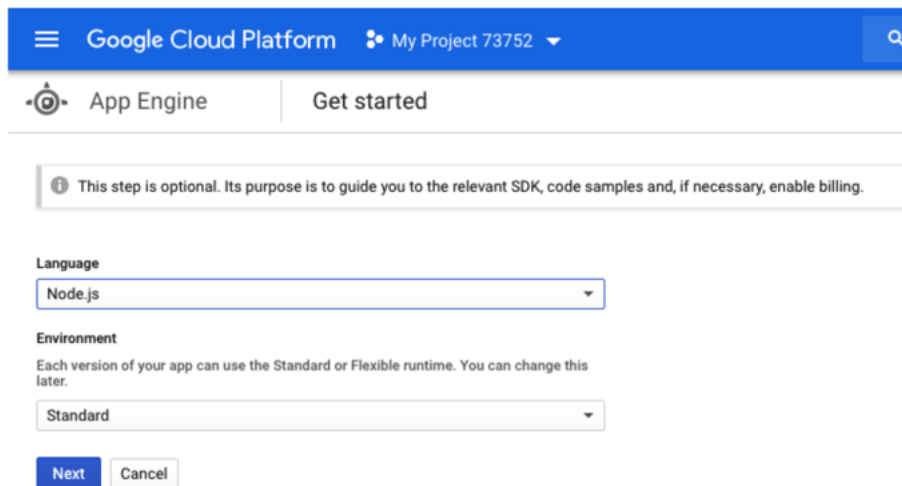
Region  
Region is permanent.

us-central

Google Map data ©2020

Create app Cancel

## 25. Select Node.js and standard environment



Google Cloud Platform My Project 73752

App Engine Get started

*This step is optional. Its purpose is to guide you to the relevant SDK, code samples and, if necessary, enable billing.*

Language  
Node.js

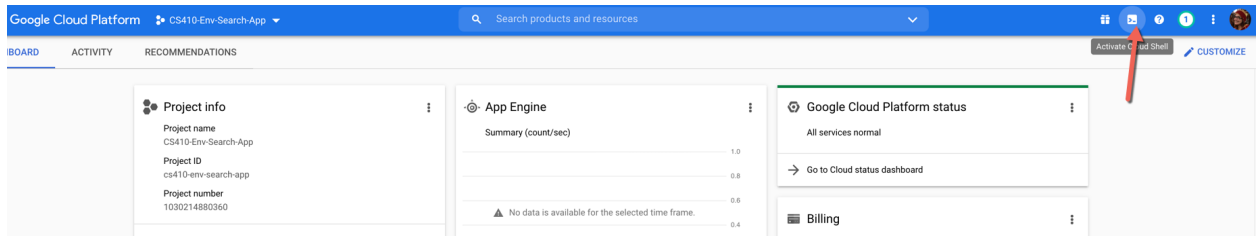
Environment  
Each version of your app can use the Standard or Flexible runtime. You can change this later.

Standard

Next Cancel

## 26. Clone our app's source code from GitHub

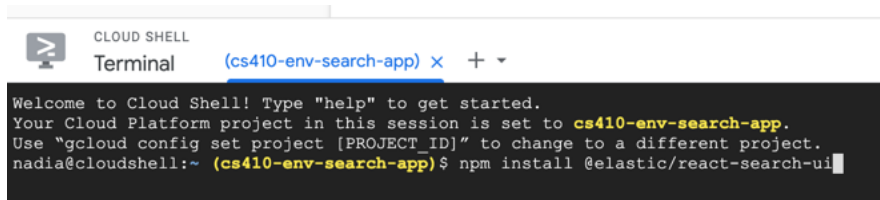
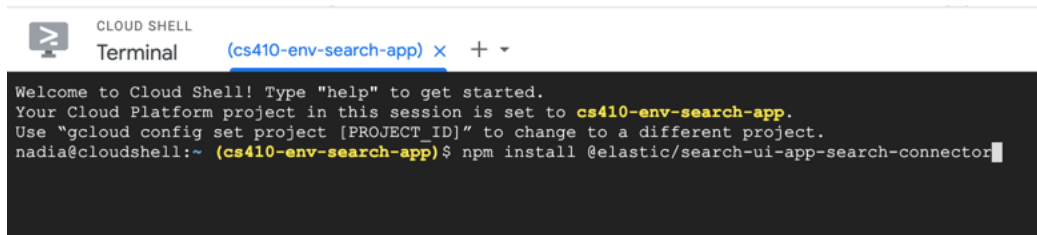
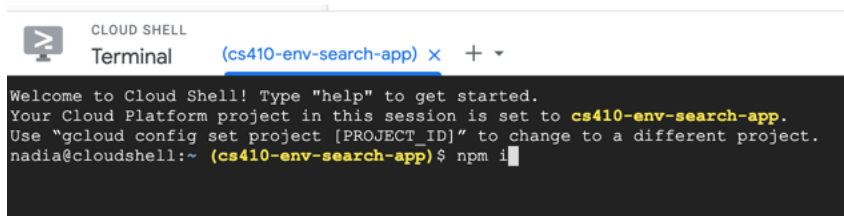
## 27. Activate the shell by clicking



git clone <https://github.com/nadiawoodninja/CourseProject.git>

28. Install npm by running and install other elastic search packages

```
npm i
npm install @elastic/search-ui-app-search-connector
npm install @elastic/react-search-ui
```



29. Build our app for deployment

To do this, simply go into your app's root folder (where your "src" folder is),  
cd CourseProject  
cd 410-search-ui

And type the following command:

```
npm i
npm run build
```

This creates a folder named "build" in our root directory.

30. Delete every thing else besides the **build** folder. Get rid of everything else, except for the build folder.

Use these commands to remove files and folders

```
rm <file-to-remove>
rm -r <remove-recursively-like-directories-inside-directories>
```

31. Add an app.yaml and deploy

In the same folder where we have our “build” folder, create a new file named app.yaml. By the end of this step, the only things left should be the “build” folder and “app.yaml”. That’s all the App Engine will need to run our app.

```
touch app.yaml
nano app.yaml
```

And add the following to its content:

```
runtime: nodejs12
handlers:
Serve all static files with url ending with a file extension
- url: /(.*\..+)$
 static_files: build/\1
 upload: build/(.*\..+)$
Catch all handler to index.html
- url: /*
 static_files: build/index.html
 upload: build/index.html
```

## Deploy the app

32. Deploy the app using the following command

```
gcloud app deploy
```

33. The app is running here

[https://cs410-env-search-app.uc.r.appspot.com/?size=n\\_20\\_n](https://cs410-env-search-app.uc.r.appspot.com/?size=n_20_n)



CLOUD SHELL

Terminal

(cs410-env-search-app) x + ▾

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to cs410-env-search-app.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
nadia@cloudshell:~ (cs410-env-search-app) $ ls
cloudshell_open CourseProject README-cloudshell.txt
nadia@cloudshell:~ (cs410-env-search-app) $ cd CourseProject/
nadia@cloudshell:~/CourseProject (cs410-env-search-app) $ cd 410-search-ui/
nadia@cloudshell:~/CourseProject/410-search-ui (cs410-env-search-app) $ gcloud app deploy
```