# AWS EC2 Instance

* i-0abbfb7a4aca6ab01
* t2.micro
* us-west-2a

## Project location:-

/home/ec2-user/my-service

## Installation of nodejs, npm and angular cli

<https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/setting-up-node-on-ec2-instance.html>

**Versions:-**

C:\Users\Dell>npm -v

6.4.1

C:\Users\Dell>node -v

v10.11.0

C:\Users\Dell>ng -v

Angular CLI: 6.2.4

## Tools used

PuTTY, WinSCP, Visual Studio Code

# Tasks

## Angular Tutorial

1. Page with text, icon, links
2. Login page with username, password

## Fetch and show json content from files

**Websites:-**

* udemy.com
* edx.org
* coursera.org

## AWS configuration

config\_AWS() {

AWS.config.update({ accessKeyId: 'AKIAJMMFSVVRZC2KGDXA', secretAccessKey: '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' });

AWS.config.update({ region: 'us-west-2' }); // say US West (Oregon) us-west-2

}

## S3 bucket API integration

AWS S3 bucket :- appsolzone

Contents:- coursera.json, edx.json, udemy.json

**Code snippet:-**

// Call S3 to create the bucket

awss3.listObjects(bucketParams, function (err, data) {

if (err) {

console.log("Error", err);

} else {

console.log("Success", data);

}

});

## Cloudsearch API integration

<https://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/CloudSearch.html>

AWS Cloudsearch domain:- [coursera-courses](https://us-west-2.console.aws.amazon.com/cloudsearch/home?region=us-west-2)

**Normal search**

location field type:- text

Search: English

Filter Query: location:['49.05830,-83.40570','51.05830,-85.40570']

Query Parser: Simple

**Haversin (Sorting Results by Distance)**

location field type:- latlon

expression:- haversin(50.05830,-84.40570,location.latitude,location.longitude)

**Data:-**

[ {

"type" : "add",

"id" : "tt2226417",

"fields" : {

"level" : "Intermediate",

"ratings" : "4.9",

"location" : "50.05830,-84.40570",

"language" : "English",

"title" : "Single Page Web Applications with AngularJS"

}

}, {

"type" : "add",

"id" : "tt0816711",

"fields" : {

"level" : "Intermediate",

"ratings" : "Intermediate",

"location" : "51.05830,-85.40570",

"language" : "English",

"title" : "Front-End Javascript Frameworks:Angular"

}

} ]

**Code snippet:-**

//end point :- cloudsearch.us-west-2.amazonaws.com - Oregon

var cloudsearch = new CloudSearch({apiVersion: '2013-01-01'});

var params = {

DomainNames: [

'coursera-courses'

]

};

cloudsearch.describeDomains(params, function(err, data) {

if (err) console.log(err, err.stack); // an error occurred

else console.log(data); // successful response

});

### Execute:-

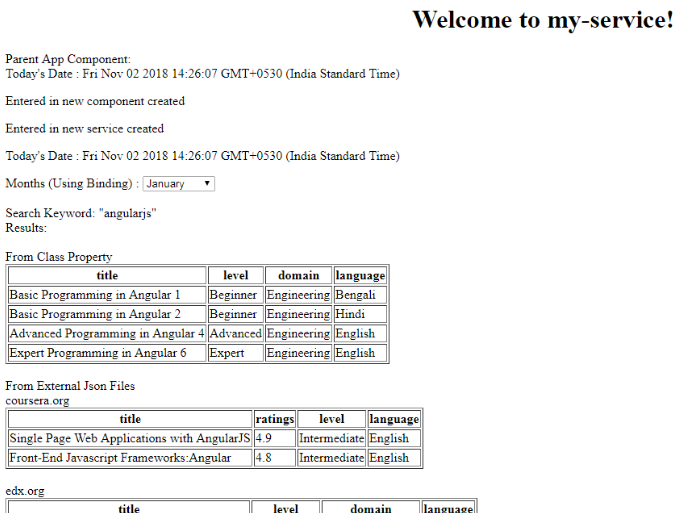
private-ip of ec2 machine

[ec2-user@ip-172-31-33-53 my-service]$ ng serve --port 4200 --host 172.31.33.53

### Output:-

Public-ip of ec2 machine

Browser:- <http://34.221.108.202:4200/>



# Issue

Cloudsearch api in AWSJavascriptSDK failed to execute. The reason is likely to be related to CORS access policy. **Cloudsearch doesn’t support cross-domain requests.**

## Fix:-

Tried various ways suggested in the internet but found no luck. Finally, added cors-anywhere package in the code and resolved the issue. Direct http call in angularjs with the aws cloudsearch url also worked by adding cors plugin in chrome browser.

# Radius based search implementation

## Data:-

2 points -> "location":"50.0583,-84.4057" , "location":"51.0583,-85.4057"}

## Inputs:-

Centre -> (0,0)

Radius -> 9650 units

Bounding Rectangle Points -> '59.83,83.39','-59.83,-83.39'

## Query:-

[http://search-coursera-courses-ttta3hjjy7jepitjpzvrt64ug4.us-west-2.cloudsearch.amazonaws.com/2013-01-01/search?q=%27English%27&expr.distance=haversin(0,-0,location.latitude,location.longitude)&expr.radius=9650&expr.offset=(distance-9650)&fq=location:[%2759.83460030720133,83.39684826785447%27,%27-59.83460030720133,-83.39684826785447%27]&sort=distance%20asc&return=offset,radius,distance,\_all\_fields&format=xml](http://search-coursera-courses-ttta3hjjy7jepitjpzvrt64ug4.us-west-2.cloudsearch.amazonaws.com/2013-01-01/search?q=%27English%27&expr.distance=haversin(0,-0,location.latitude,location.longitude)&expr.radius=9650&expr.offset=(distance-9650)&fq=location:%5b%2759.83460030720133,83.39684826785447%27,%27-59.83460030720133,-83.39684826785447%27%5d&sort=distance%20asc&return=offset,radius,distance,_all_fields&format=xml)

## Response:-

<results>

<status rid="8cbZ0/MshUYKxbGI" time-ms="20"/>

<hits found="2" start="0">

<hit id="tt2226417">

<field name="ratings">4.9</field>

<field name="language">English</field>

<field name="level">Intermediate</field>

<field name="title">Single Page Web Applications with AngularJS</field>

<field name="location">50.0583,-84.4057</field>

<expr name="offset">-36.42170802293549</expr>

<expr name="radius">9650.0</expr>

<expr name="distance">9613.578291977065</expr>

</hit>

<hit id="tt0816711">

<field name="ratings">Intermediate</field>

<field name="language">English</field>

<field name="level">Intermediate</field>

<field name="title">Front-End Javascript Frameworks:Angular</field>

<field name="location">51.0583,-85.4057</field>

<expr name="offset">41.536342067578516</expr>

<expr name="radius">9650.0</expr>

<expr name="distance">9691.536342067579</expr>

</hit>

</hits>

</results>

Two manual calculations are done.

* Finding bounding co-ordinates given center and radius.
* Filtering out points outside circle

# Steps to deploy radius-based-search angular application in aws ec2

1. Create AWS EC2 instance with ssh security group.

2. Install Putty in windows - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html

3. Login to ec2 via putty using ppk file.

4. Run following in ec2 to install node,npm and angular cli:-

* curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.32.0/install.sh | bash
* . ~/.nvm/nvm.sh
* nvm install 10.11.0
* node -e "console.log('Running Node.js ' + process.version)"
* npm install -g @angular/cli@6.2.4

5. Zip angular project to my-service.zip

6. Transfer zip file to aws ec2 machine using Winscp.

7. Unzip the file.

* unzip my-service.zip

8. Go to folder my-service and deploy. (with private ip and default port 4200)

* ng serve --host 172.31.38.224

9. Open the url in the browser with public ip, lat, lon and radius.

* http://34.222.114.106:4200/search?lat=22.58796393&lon=88.42647098&radius=3

# Book Listing and Searching

Two form applications are made:-

1. Book Listing – User inputs (author, title, year, cover, address) are taken to add book information. This information is stored in json format in S3 bucket and uploaded to cloudsearch domain.
2. Book Searching – User inputs (lat/lon, radius) are taken to find and show nearest books based on radius search on cloudsearch domain.