2/6/2019

AIR QUALITY INDEX IN CHINA(PM 2.5, PM10)

Cloud Computing group assignment

Group name: Master Chef

Group members:

Ye Lyu(s3710372): Share load

Yi Weng(s3642928): Share load

Xinhong Chen(s3710646): Share load

Links:

URL of application: https://s3.amazonaws.com/s3642928-rmit-storage/html/HomeContent.html

Repository url: https://github.com/Ivylyu/cloudcomputing.git

Public dataset link of application:
https://api.openaq.org/v1/latest?country=CN&&parameter=pm10&&limit=1500

Summary:

This application allows people to check Air Quality Index of PM2.5 and PM10 in China, so people can take precautions to avoid getting sick.
This application utilizes a number of cloud services including Openaq API,Amazon S3, Amazon EC2, Amazon DynamoDB, Google Map API, Google Chart API and IAM.

Introduction:

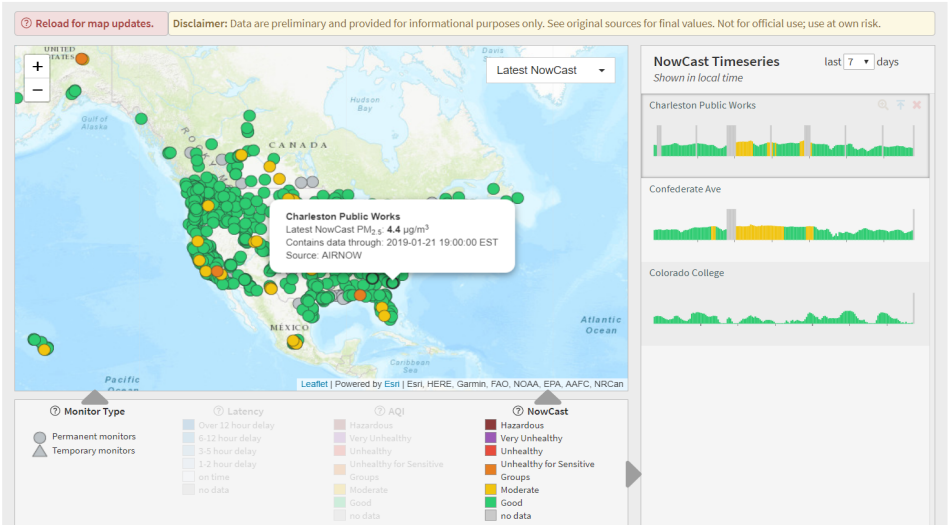
Air Pollution is a server problem in China. With accelerated industry development, air pollution is worsening. There are lots of particulate matter(a mixture of solid particles and liquid droplets found in the air), such as PM 10 and PM 2.5, which will harm human health. Therefore, people can check real-time PM index for different locations(all stations) in google map in our application and take precautions to take care of themselves. In addition, people can also check previous 6-day daily average PM index to compare on google chart. It can be useful for data analysis. In order to achieve these functions, we use appropriate cloud services.

Related work:

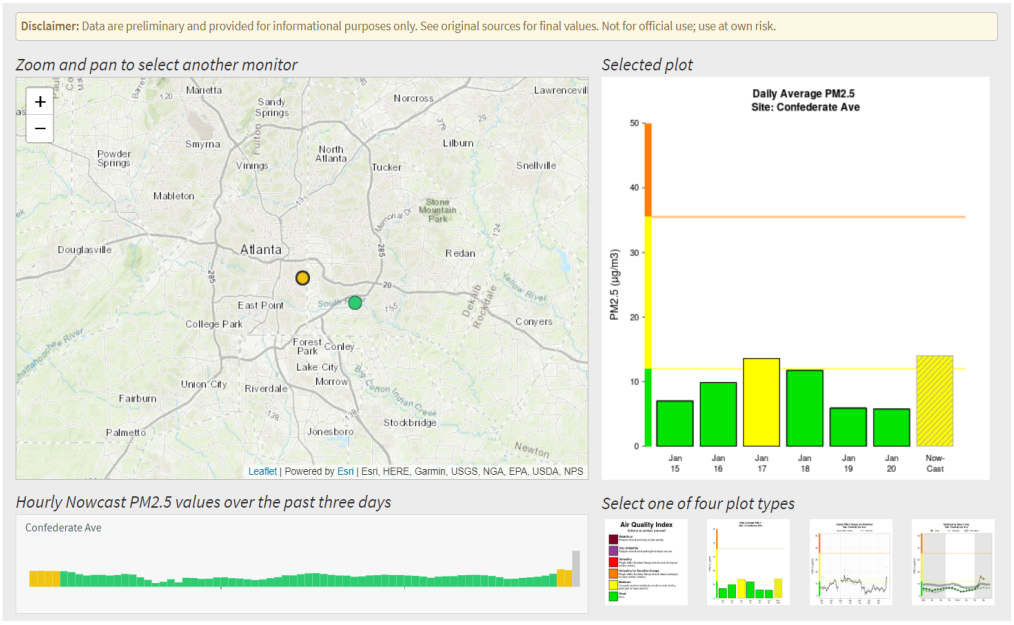
We found a similar website which monitor PM 2.5 in the United States. Our design is similar to this website, especially using google map API and google chart API.

https://tools.airfire.org/monitoring/v4#!/?category=PM2.5_nowcast&centerlat=44.2893&centerlon=-95.4492&zoom=3&monitors=450190049_01&monitors=131210055_01&monitors=080410017_01

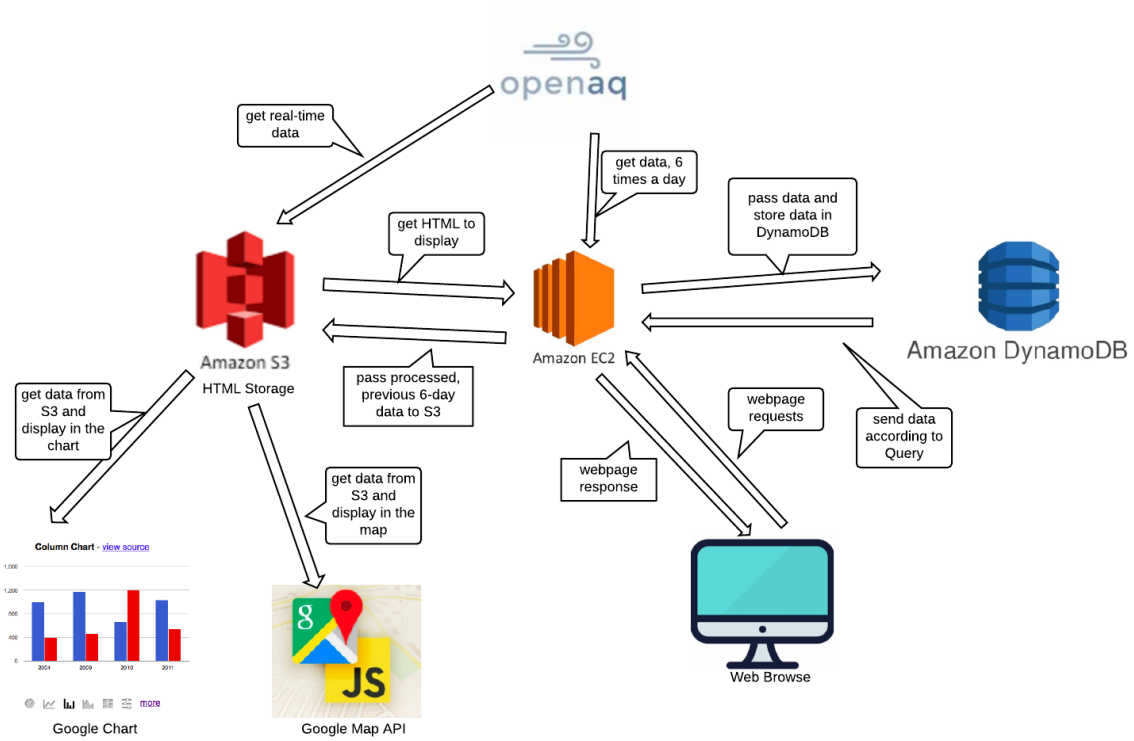
Map Page



Analysis Page



Software Design/Architecture:



Openaq API: API source document. OpenAQ platform makes air quality data available through Public Datasets Program.

Amazon S3: Store html, icons and temporary json files.

Amazon EC2: As Sever, virtual machine, administrators' keys.

Amazon DynamoDB: stores all Tables.

Google Map API: This is what creates the map.

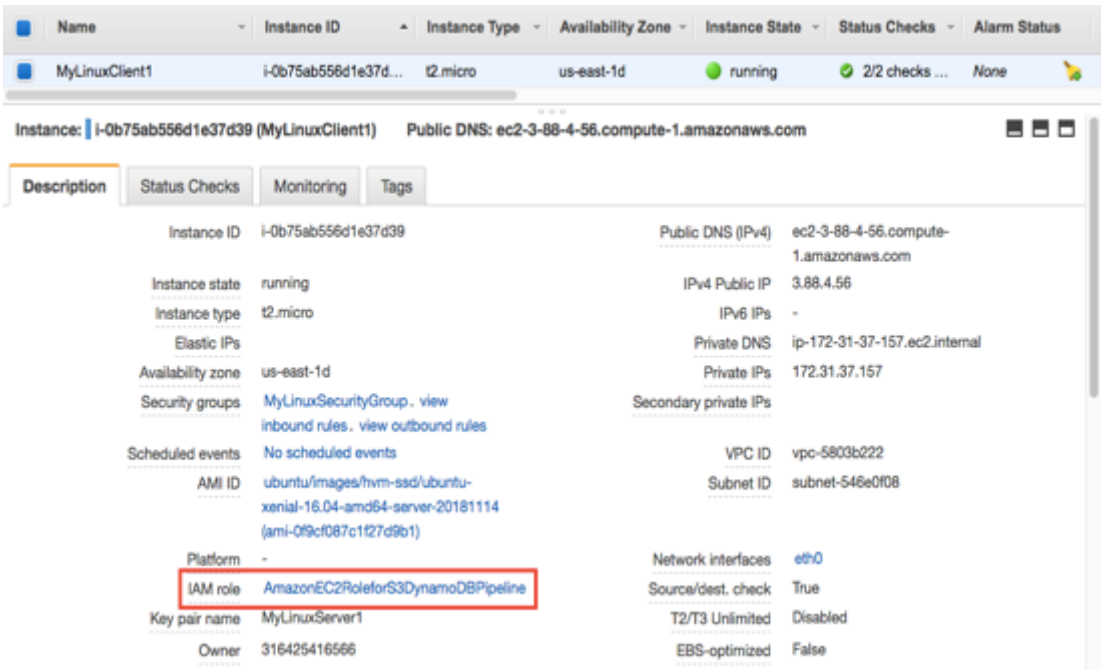
Google Chart API: This is what creates the chart.

IAM: gives the role to EC2.

Implementation:

Step 1:

We followed tutorial sheet in Lab5;Amazon Web Service (AWS), created the EC2 instance. The following picture is our Cloud Server and Environment setup:



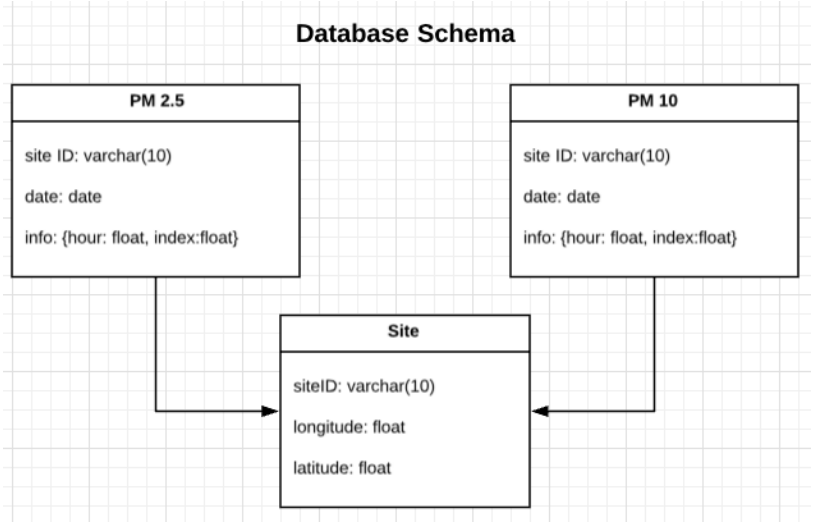

```
ubuntu@ip-172-31-37-157:~/tmp/Java$ javac -version
javac 1.8.0_191
```

That's the terminal or Putty access to virtual machine.

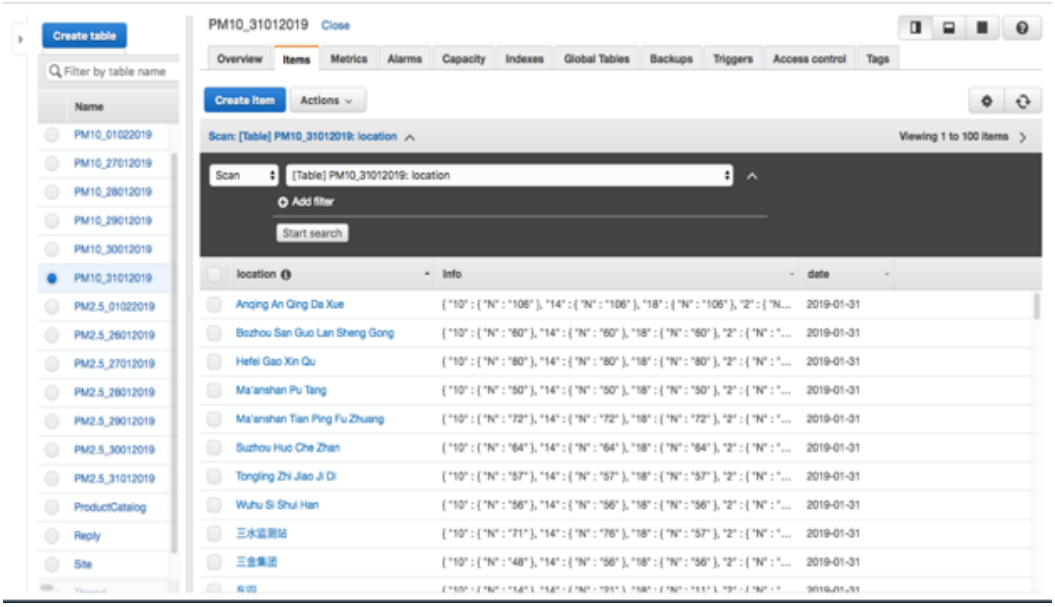
Step 2:

We followed tutorial sheet in Lab 8 and 9, Cloud Database-DynamoDB. We used Java to work with DynamoDB from local environment. We created tables and load sample in JSON format. Here's the link for Java code: https://github.com/Ivylyu/cloudcomputing.git

The following picture shows our Database Schema:

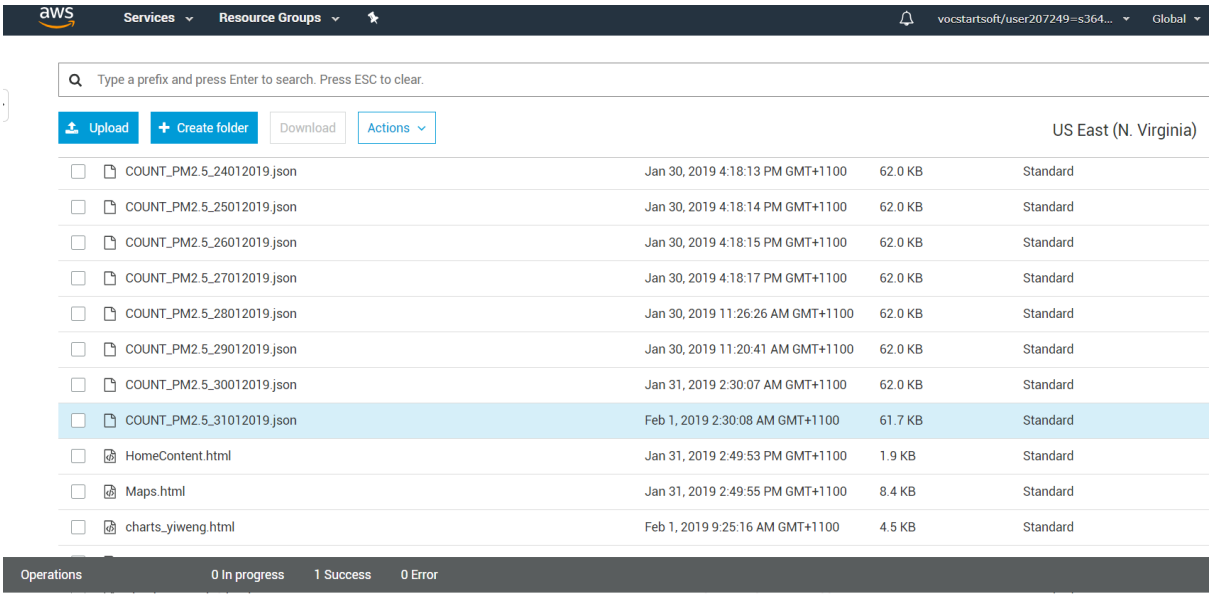


In addition, the following picture shows our Database structure in DynamoDB:

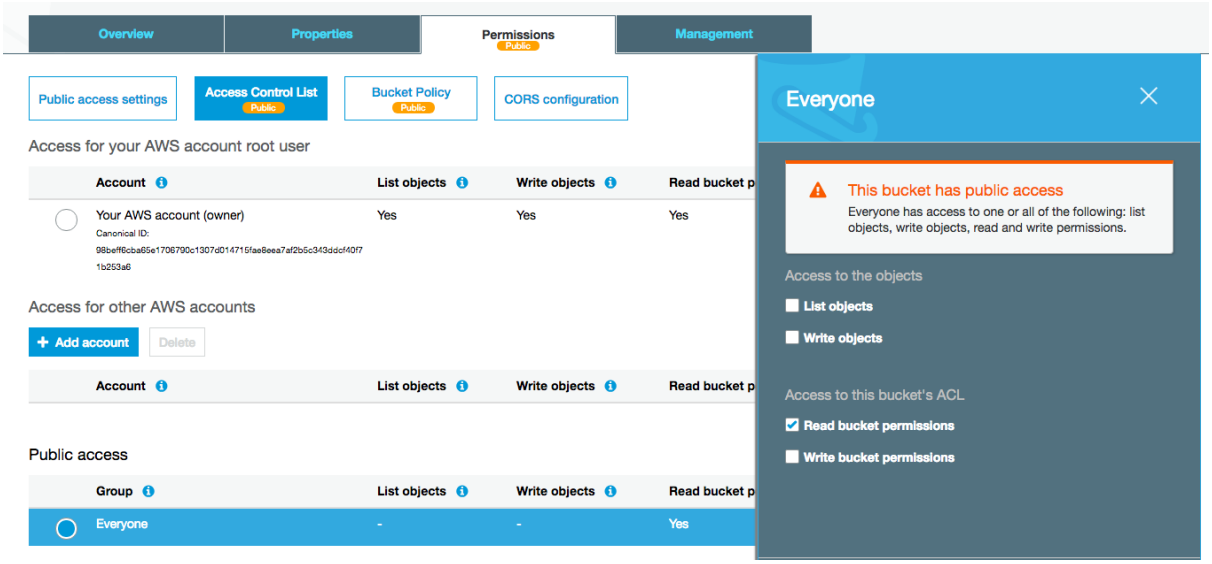


Step 3:

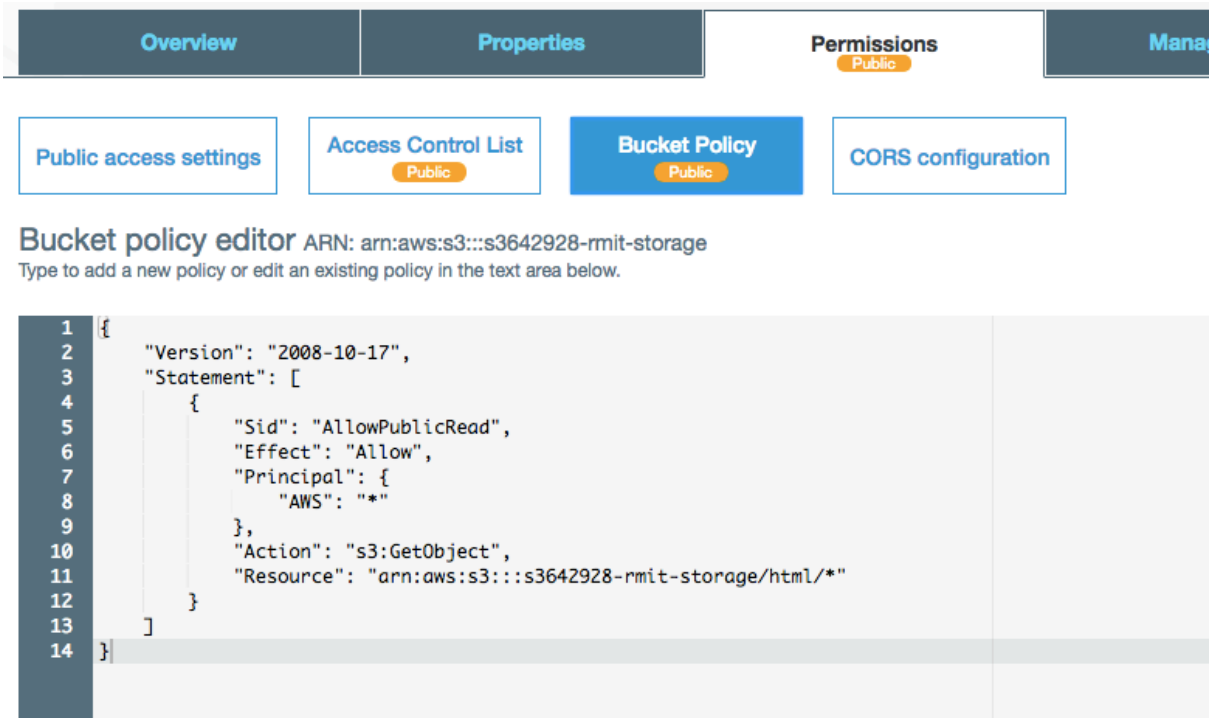
We use S3 to publish our html static page file and deposit temporary data files generated by our EC2 server(which store the daily average of PM2.5 and PM10 index).



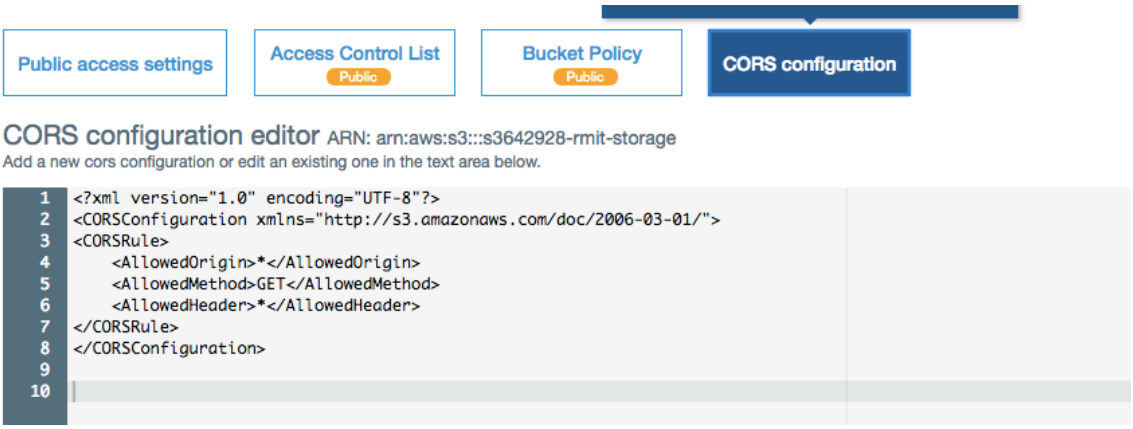
We did S3 permission setup as well, everyone can read the page files from the bucket.



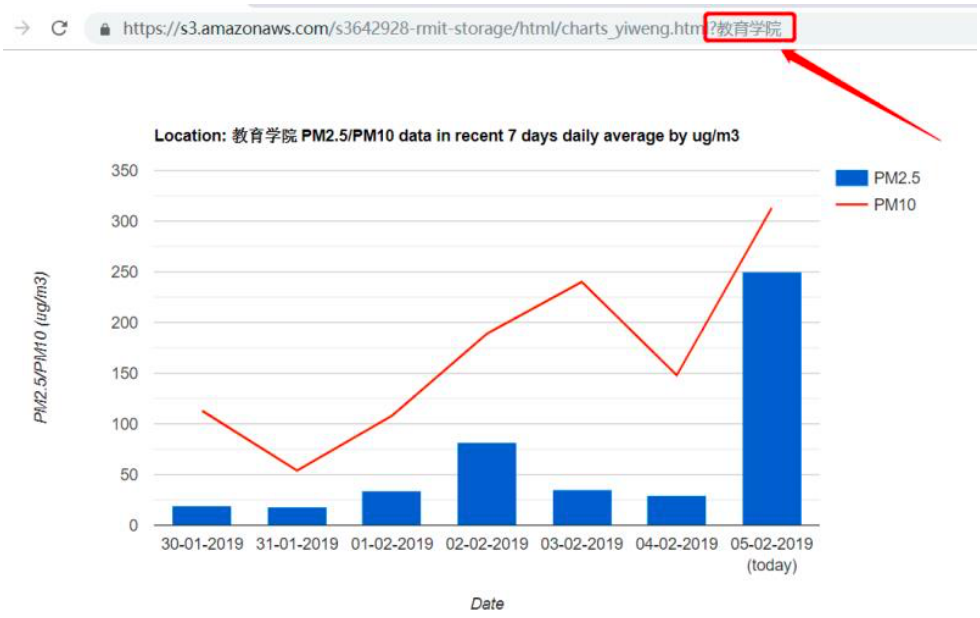
Furthermore, we change Permissions so that all files are public.



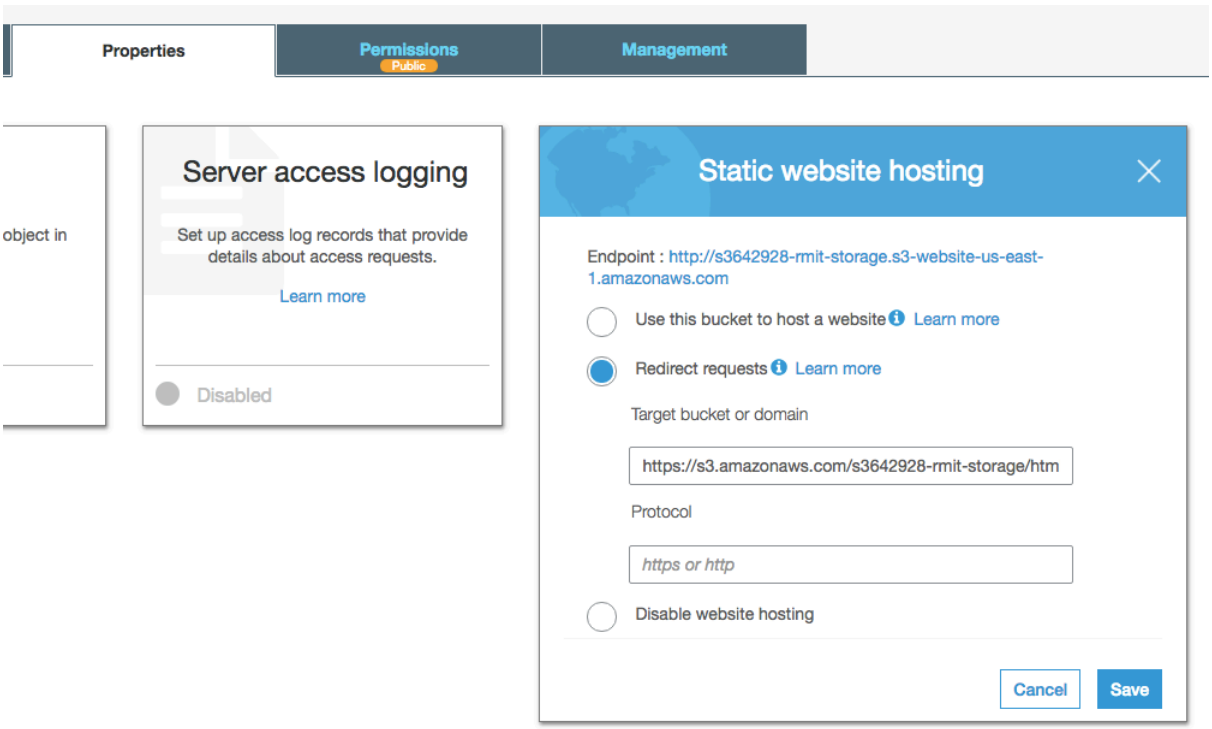
We also modified CORS configuration based on the learning material from AWS website. The link is here(https://docs.aws.amazon.com/AmazonS3/latest/dev/cors.html#how-do-i-enable-cors). It can handle different types of HTTP requests, therefore html can read the content in the file directly.



When user click on the certain location, we pass the location to next page.



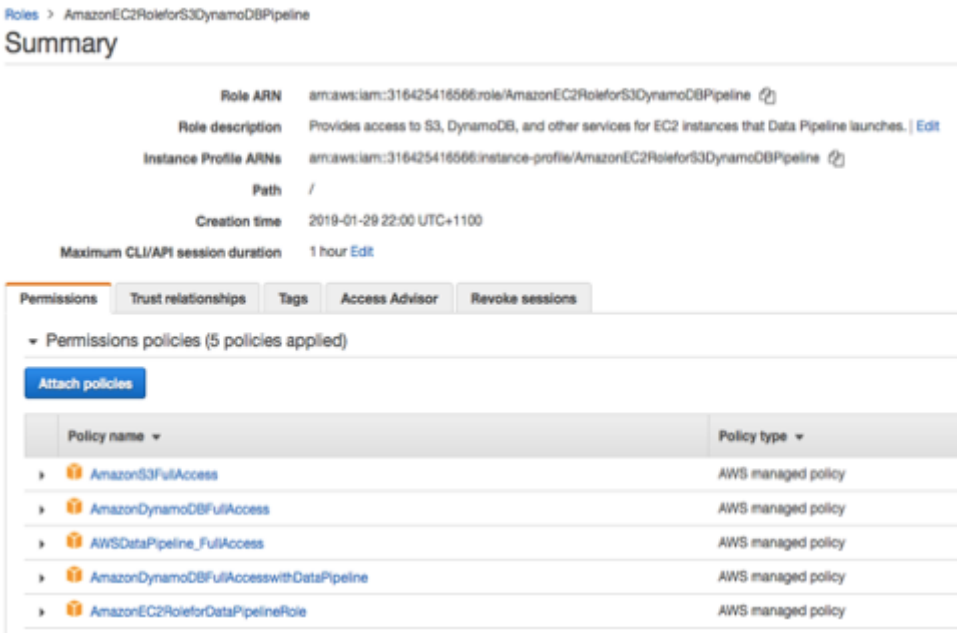
Through Properties setting, we added redirect requests on Static website hosting. So the information can be passed successfully.



Step 4:

We set IAM role for EC2 access to DynamoDB and S3, we followed steps on AWS Identity and Access Management website about “Creating a Role to Delegate Permissions to an IAM User”, the link is here:
https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-user.html

The following pictures show all policies. In addition, you can see the IAM role section on the EC2 instance has been delegated.



Finally, these are our Daily scripts.

A. Update database 6 times a day, and create daily table in DynamoDB.

B. Calculate average PM2.5 and PM10 by every location and date into S3 temporary file by java programming from DynamoDB data.

```
30 00      * * *   sudo sh /home/ubuntu/tmp/script/CreateTablesDaily.bash
30 01      * * *   sudo sh /home/ubuntu/tmp/script/PM25collectionScript.bash
30 05      * * *   sudo sh /home/ubuntu/tmp/script/PM25collectionScript.bash
30 09      * * *   sudo sh /home/ubuntu/tmp/script/PM25collectionScript.bash
30 13      * * *   sudo sh /home/ubuntu/tmp/script/PM25collectionScript.bash
30 17      * * *   sudo sh /home/ubuntu/tmp/script/PM25collectionScript.bash
30 21      * * *   sudo sh /home/ubuntu/tmp/script/PM25collectionScript.bash
30 23      * * *   sudo sh /home/ubuntu/tmp/script/CreateJsonS3Daily.bash
ubuntu@ip-172-31-37-157:~/tmp/Java$ ls -l
total 945204
-rw-rw-r-- 1 ubuntu ubuntu 120945088 Jan 26 14:05 CreateAQITable.jar
-rw-rw-r-- 1 ubuntu ubuntu 120950867 Jan 27 07:20 CreatePM10Table.jar
-rw-rw-r-- 1 ubuntu ubuntu 120945089 Jan 26 14:06 CreatePM25Table.jar
-rw-rw-r-- 1 ubuntu ubuntu 120955568 Jan 26 13:42 LocationLoadData.jar
-rw-rw-r-- 1 ubuntu ubuntu 121020113 Jan 30 07:57 PM10DataCountToS3.jar
-rw-rw-r-- 1 ubuntu ubuntu 121017979 Jan 30 05:47 PM10LoadData.jar
-rw-rw-r-- 1 ubuntu ubuntu 121020041 Jan 30 08:03 PM25DataCountToS3.jar
-rw-rw-r-- 1 ubuntu ubuntu 121017980 Jan 30 05:46 PM25LoadData.jar
ubuntu@ip-172-31-37-157:~/tmp/Java$
```

C. Log record generate by java programming

```
ubuntu@ip-172-31-37-157:~/tmp/log$ ls
COUNT_PM10_30012019.json                    PM10_2019013021.log
COUNT_PM2.5_30012019.json                   PM10_2019013101.log
CreationAQI_20190126.log                    PM10_2019013105.log
CreationAQI_20190127.log                    PM10_2019013109.log
CreationPM10_20190127.log                   PM10_2019013113.log
CreationPM10_20190128.log                   PM10_2019013117.log
CreationPM10_20190129.log                   PM10_2019013121.log
CreationPM10_20190130.log                   PM10_2019020101.log
CreationPM10_20190131.log                   PM10_2019020105.log
CreationPM10_20190201.log                   PM10_2019020109.log
CreationPM25_20190126.log                   PM25_2019012620.log
CreationPM25_20190127.log                   PM25_2019012621.log
CreationPM25_20190128.log                   PM25_2019012701.log
CreationPM25_20190129.log                   PM25_2019012705.log
CreationPM25_20190130.log                   PM25_2019012707.log
CreationPM25_20190131.log                   PM25_2019012709.log
CreationPM25_20190201.log                   PM25_2019012713.log
JsonS3PM10_20190129?.log                    PM25_2019012717.log
JsonS3PM10_Thu Jan 31 23:30:01 CST 2019.log PM25_2019012721.log
JsonS3PM10_Wed Jan 30 08:13:40 CST 2019.log PM25_2019012801.log
JsonS3PM10_Wed Jan 30 23:30:01 CST 2019.log PM25_2019012805.log
```

D. HTML using google map, chart and OpenAQ api

This is our OpenAQ API data source.

User manual:

Our application is very easy to use. Please see the instruction below.

https://youtu.be/ORwO_TUzAmA

Discussion:

Due to the time limitation, we could make the webpage nicer. In addition, we can optimize our java code and JavaScript ajax code to improve efficiency, such as, using multi-threading.

References:

Amazon DynamoDB Developer Guide API Version 2012-08-10:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/dynamodb-dg.pdf
Amazon Simple Storage Service Developer Guide API Version 2006-03-01:
https://docs.aws.amazon.com/AmazonS3/latest/dev/cors.html
AWS Identity and Access Management User Guide 2019:
https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html
Open AQ Platform API
https://docs.openaq.org/