

ASSIGNMENT - 3

PART A: Build an event-driven serverless Application

SUBMITTED TO: Prof. SAURABH DEY

SUBMITTED BY: DHRUV DOSHI

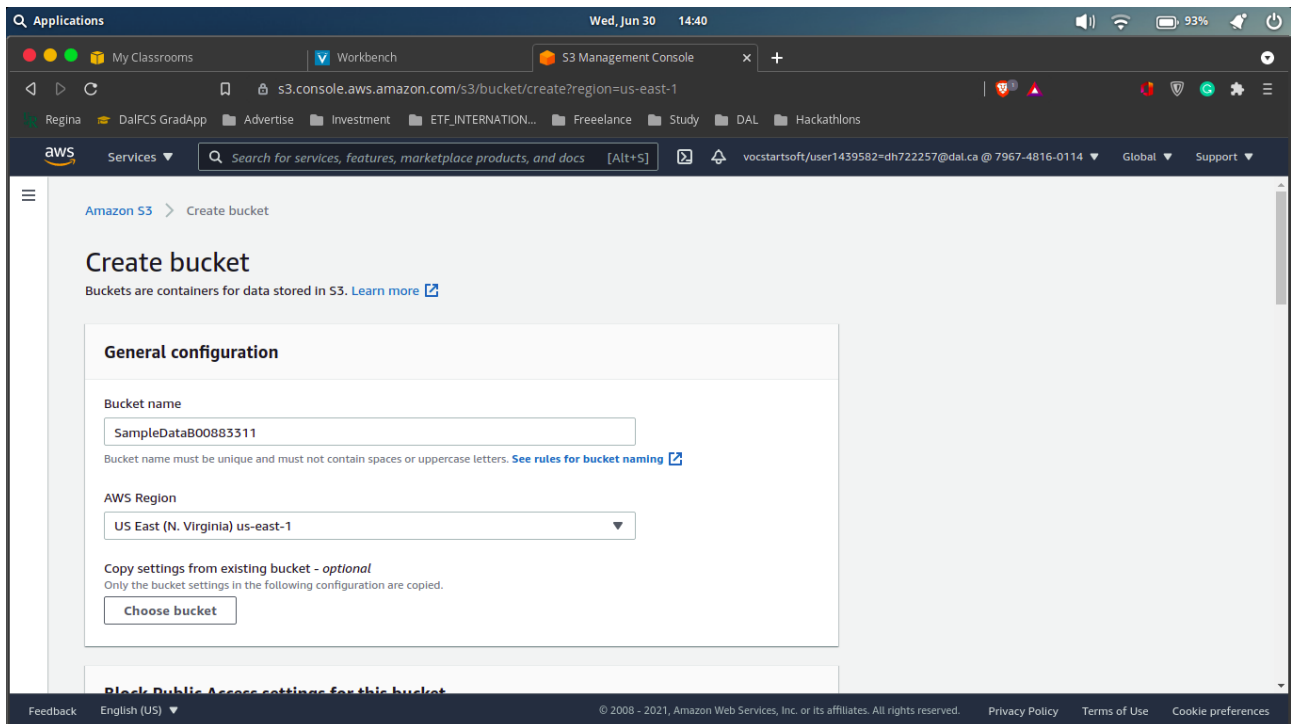
dh722257@dal.ca

FIND WHOLE CODE AT GITLAB:

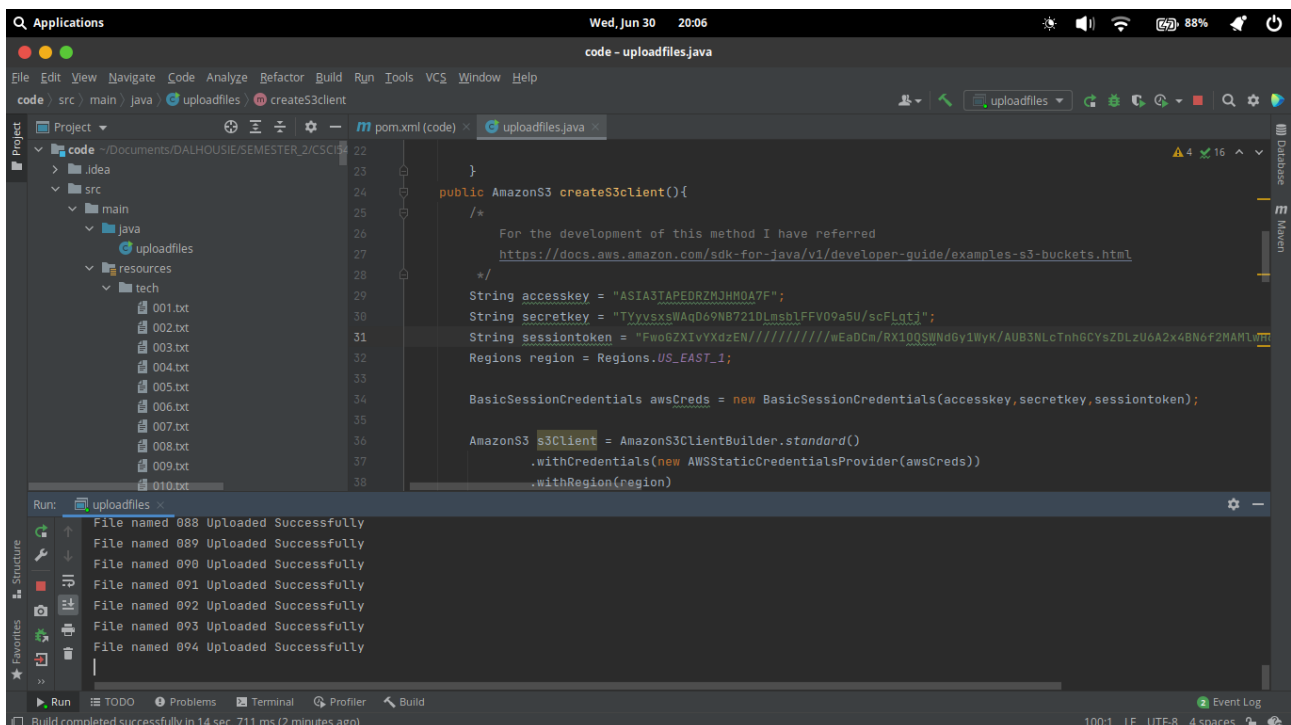
https://git.cs.dal.ca/doshi/dhruv-csci5410-summer/-/tree/master/assignment_3

There is a series of Screenshots which explains the whole process.

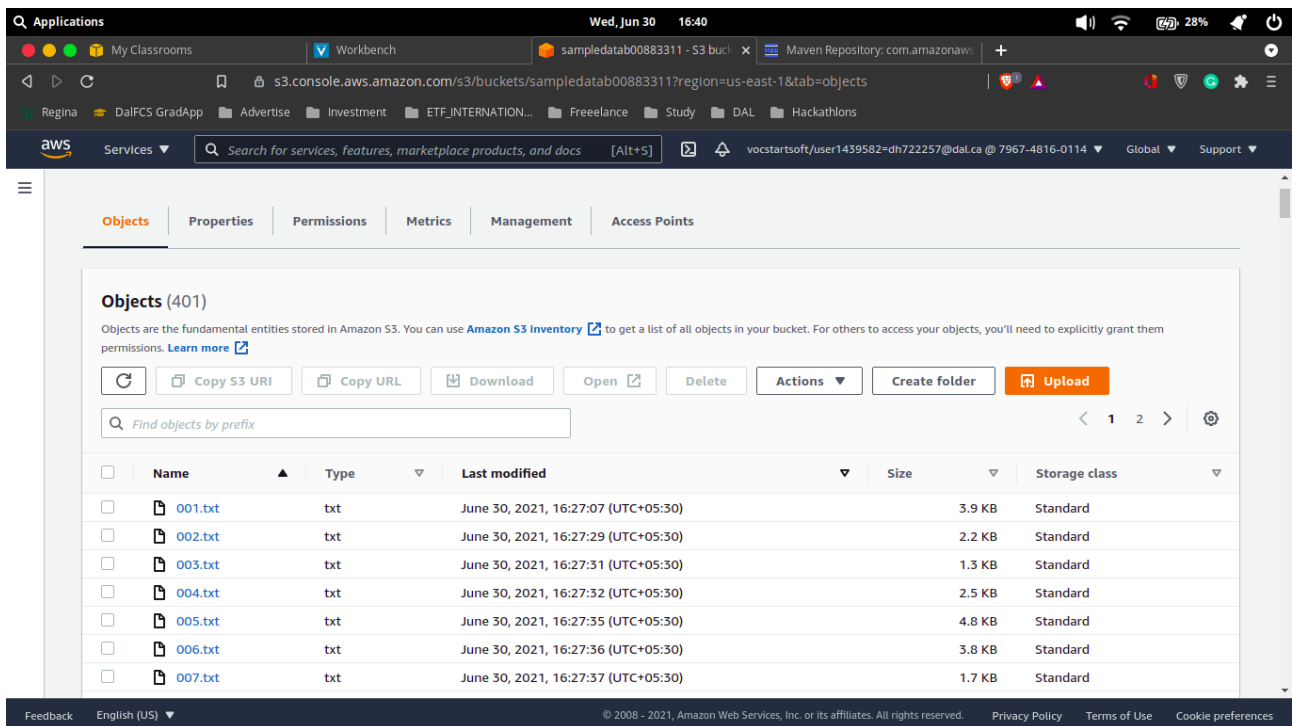
Create S3 Bucket to save data 401 extracted files



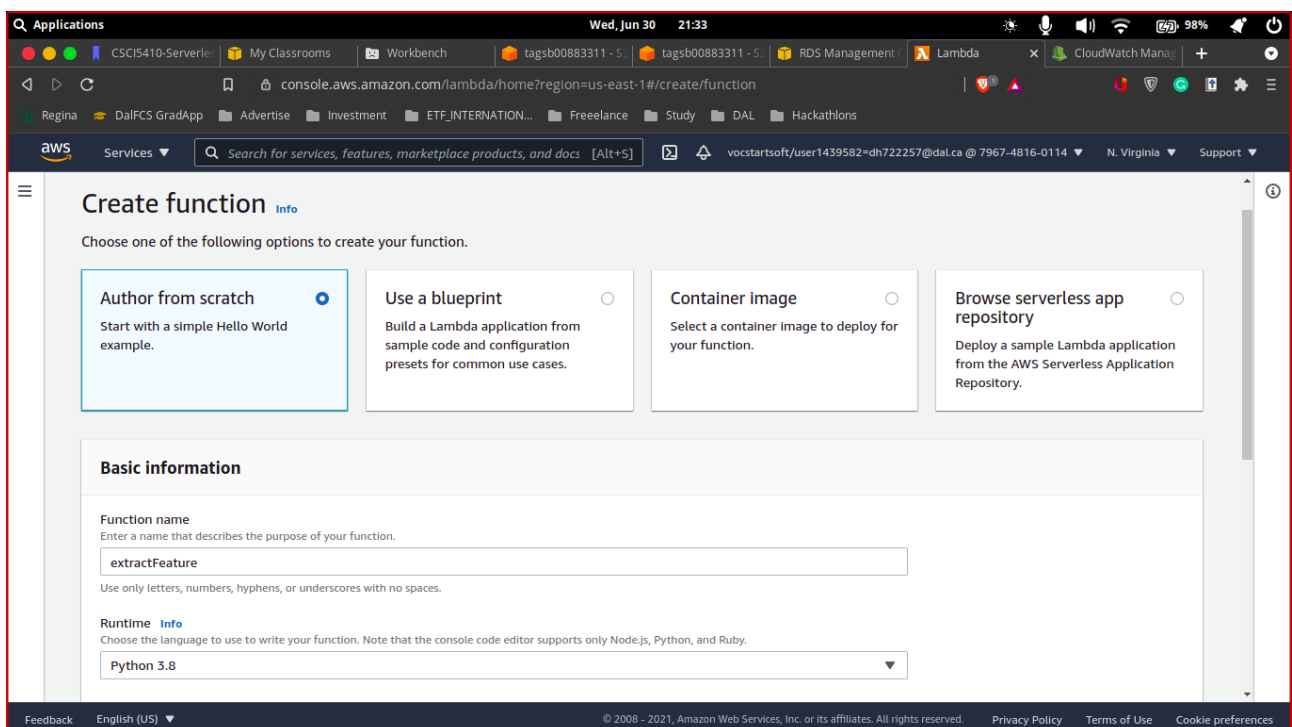
Java code for pushing files to S3 Bucket



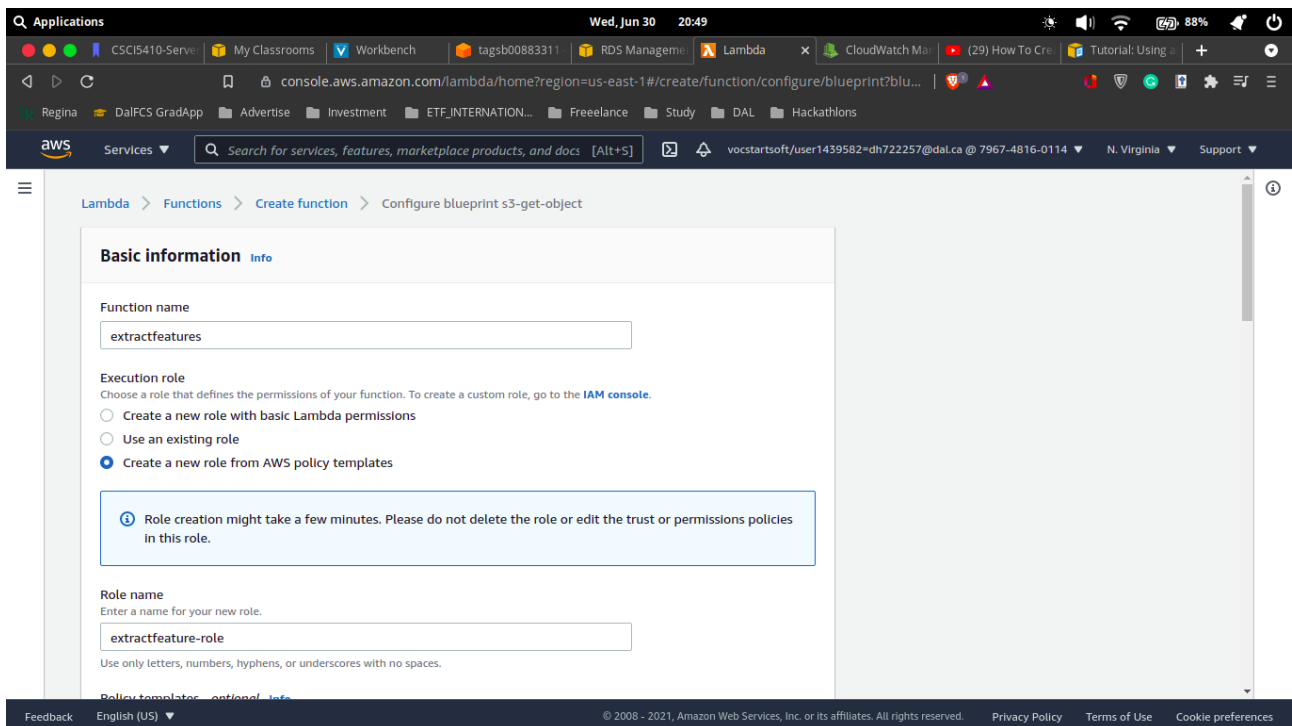
After Uploading data to S3



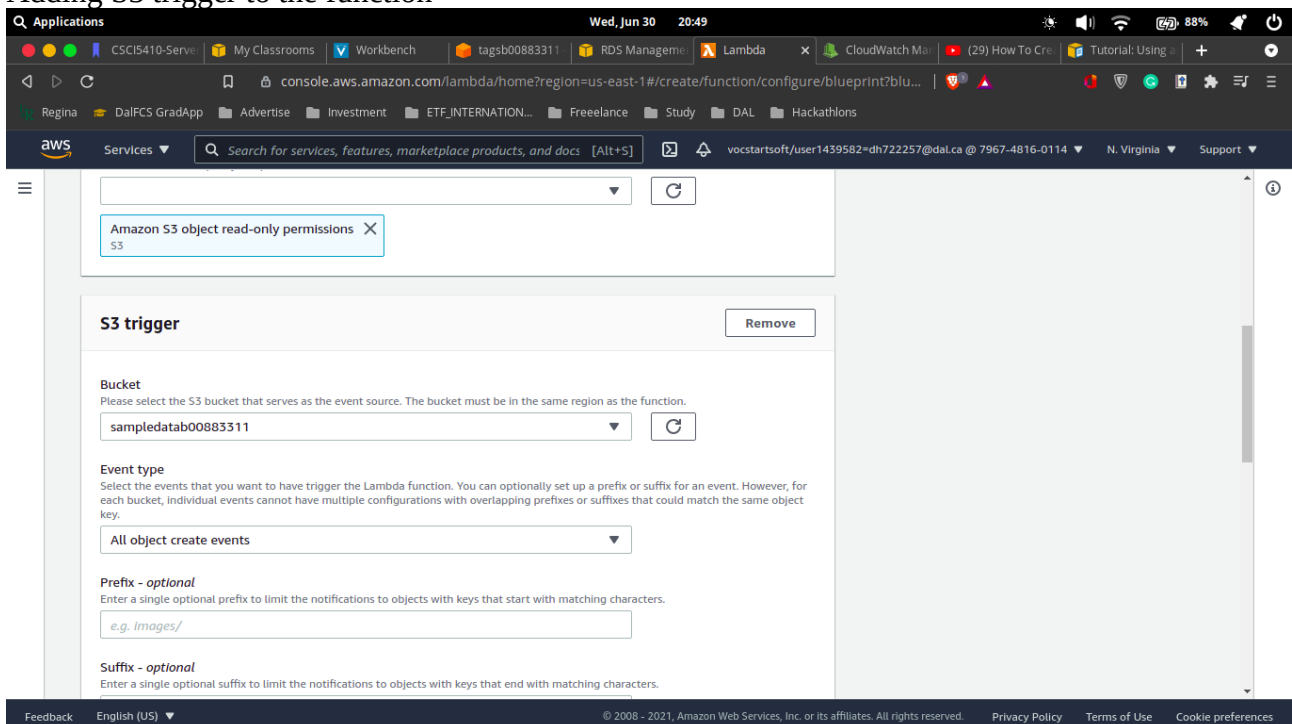
Creating new lambda function



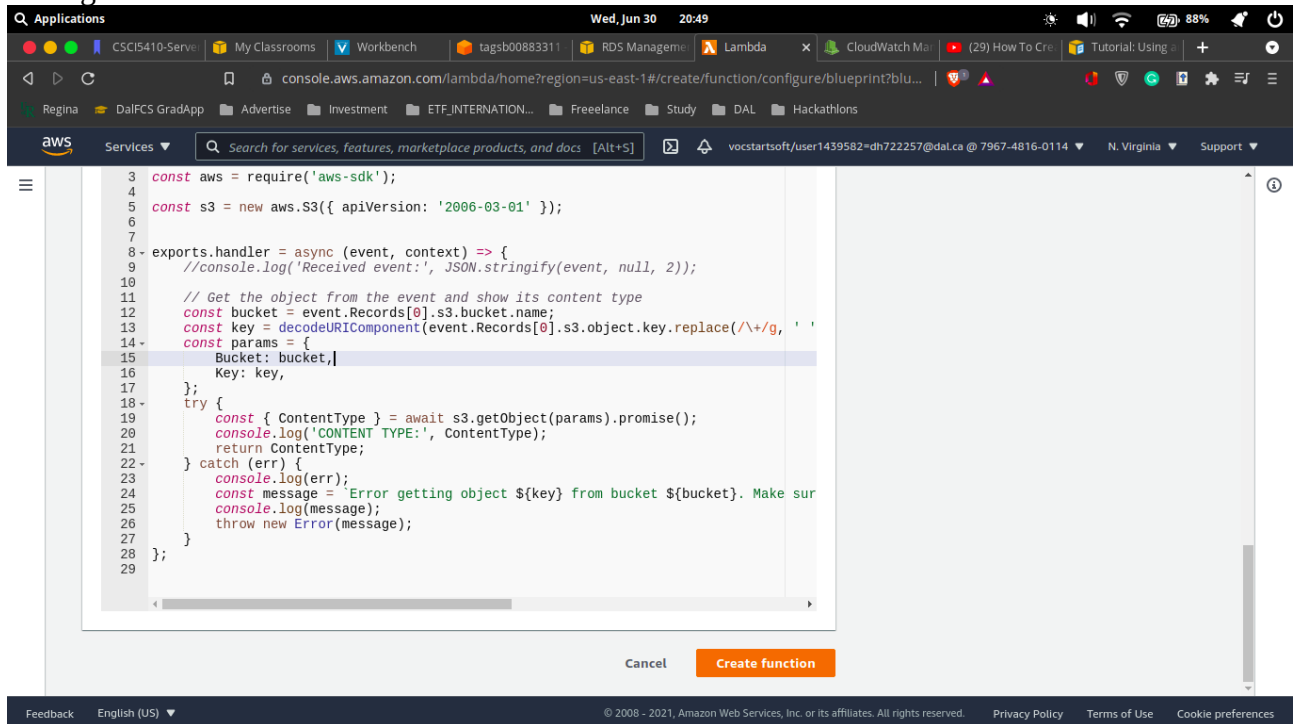
Creating a function



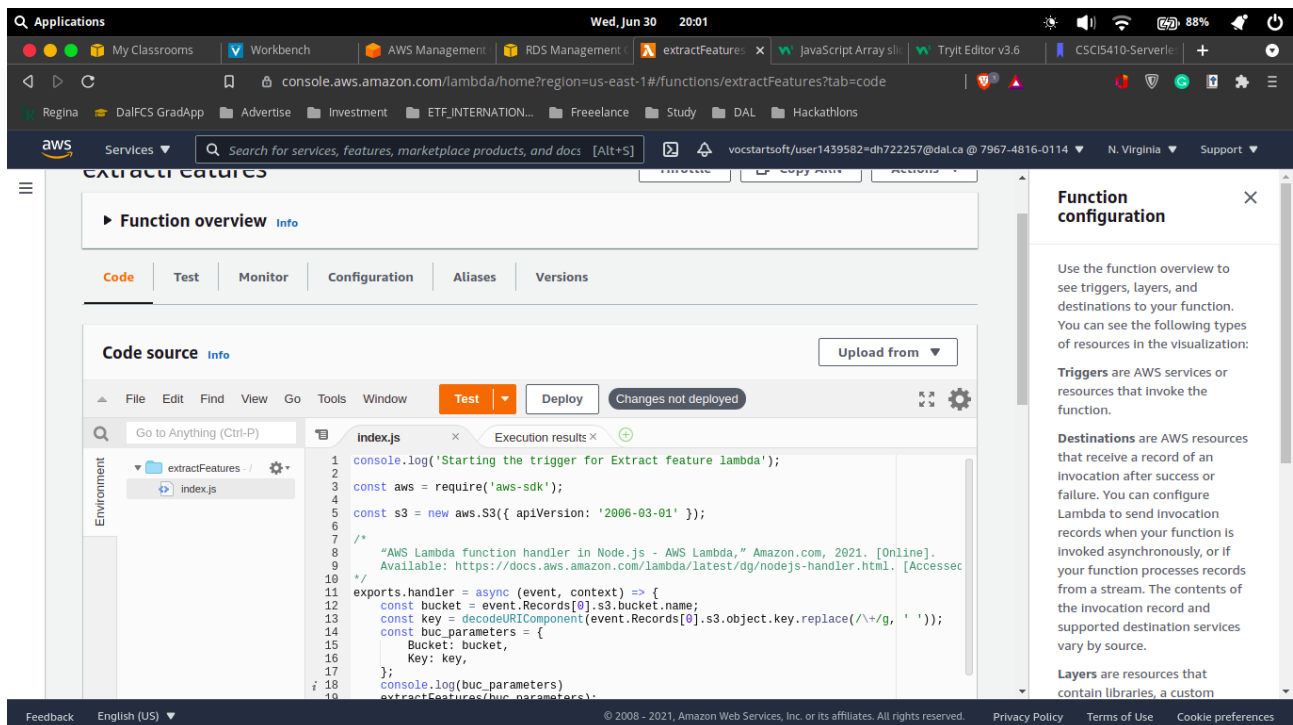
Adding S3 trigger to the function



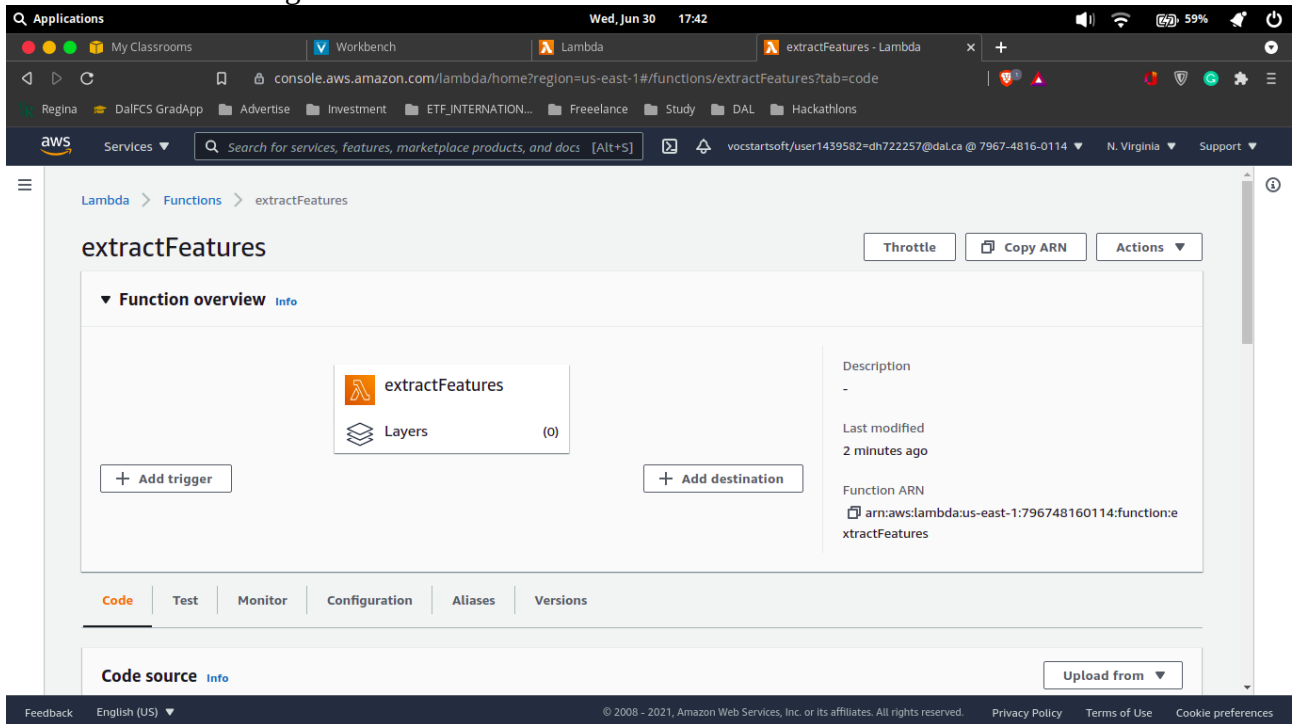
Auto generated code from lambda



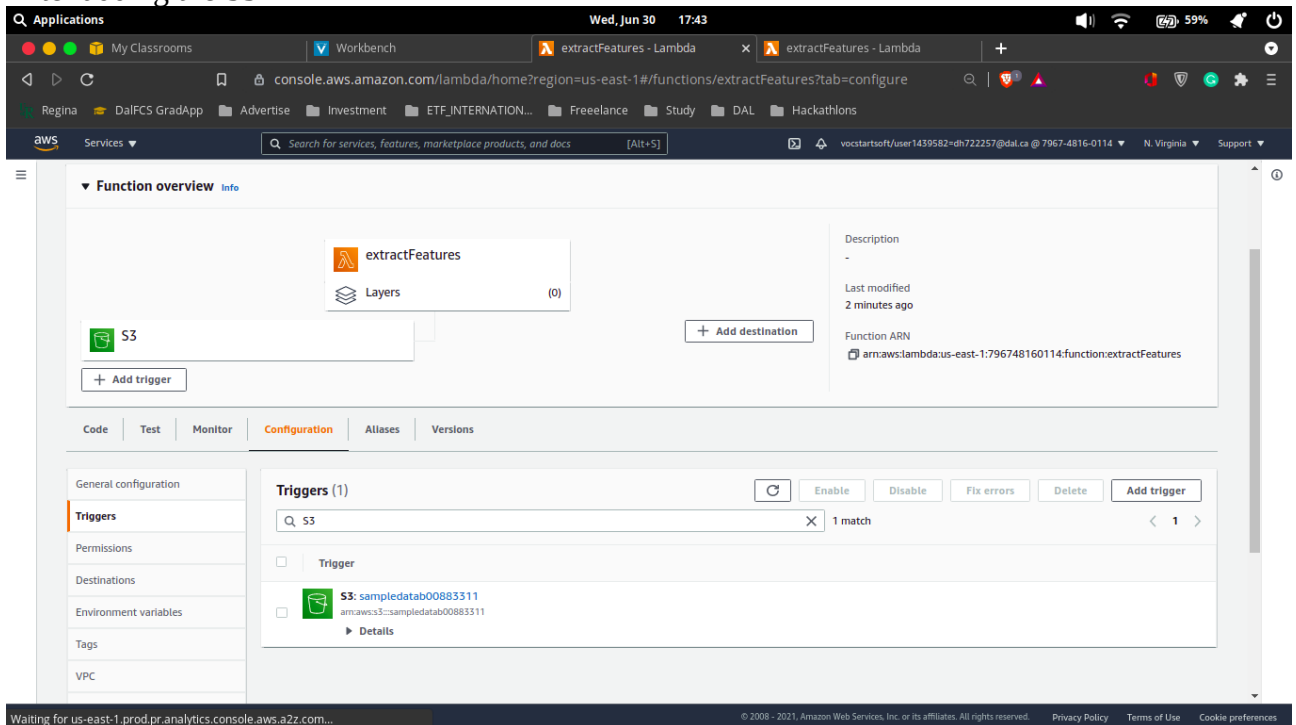
Extractfeature code uploading



Before after the setting structure



After adding the S3



New Object S3 data writing from lambda function

The screenshot shows the Amazon S3 console interface. The left sidebar contains navigation options: Buckets, Access Points, Object Lambda Access Points, Batch Operations, Access analyzer for S3, Block Public Access settings for this account, Storage Lens (Dashboards, AWS Organizations settings), Feature spotlight (3), and AWS Marketplace for S3. The main content area is titled 'Objects (47)' and includes a description of objects and a list of actions: Copy S3 URI, Copy URL, Download, Open, Delete, and Actions. Below these are buttons for 'Create folder' and 'Upload'. A search bar is present with the text 'Find objects by prefix'. A table lists the objects with columns for Name, Type, Last modified, Size, and Storage class. The objects are named 045ne through 058ne, all of type '-', last modified on June 30, 2021, and stored in the 'Standard' storage class.

Name	Type	Last modified	Size	Storage class
045ne	-	June 30, 2021, 22:16:20 (UTC+05:30)	782.0 B	Standard
046ne	-	June 30, 2021, 22:16:20 (UTC+05:30)	1.5 KB	Standard
047ne	-	June 30, 2021, 22:16:22 (UTC+05:30)	765.0 B	Standard
048ne	-	June 30, 2021, 22:16:22 (UTC+05:30)	1.1 KB	Standard
049ne	-	June 30, 2021, 22:16:23 (UTC+05:30)	1.3 KB	Standard
050ne	-	June 30, 2021, 22:16:24 (UTC+05:30)	867.0 B	Standard
056ne	-	June 30, 2021, 22:16:29 (UTC+05:30)	1.1 KB	Standard
058ne	-	June 30, 2021, 22:16:30 (UTC+05:30)	962.0 B	Standard

Completion of data adding by lambda function (401 files for 401 input files)

The screenshot shows the Amazon S3 console interface with the title 'Objects (401)'. The main content area displays a table of objects. The objects are named 0258ne through 0267ne, all of type '-', last modified on June 30, 2021, and stored in the 'Standard' storage class. The table is paginated, showing objects 1 through 2.

Name	Type	Last modified	Size	Storage class
0258ne	-	June 30, 2021, 22:19:56 (UTC+05:30)	718.0 B	Standard
0259ne	-	June 30, 2021, 22:19:43 (UTC+05:30)	953.0 B	Standard
025ne	-	June 30, 2021, 22:16:00 (UTC+05:30)	735.0 B	Standard
0260ne	-	June 30, 2021, 22:19:46 (UTC+05:30)	1.1 KB	Standard
0261ne	-	June 30, 2021, 22:19:46 (UTC+05:30)	1.1 KB	Standard
0262ne	-	June 30, 2021, 22:19:46 (UTC+05:30)	696.0 B	Standard
0263ne	-	June 30, 2021, 22:19:46 (UTC+05:30)	766.0 B	Standard
0264ne	-	June 30, 2021, 22:19:47 (UTC+05:30)	1.2 KB	Standard
0265ne	-	June 30, 2021, 22:19:48 (UTC+05:30)	939.0 B	Standard
0266ne	-	June 30, 2021, 22:19:48 (UTC+05:30)	552.0 B	Standard
0267ne	-	June 30, 2021, 22:19:50 (UTC+05:30)	1.2 KB	Standard

Final Structure of S3

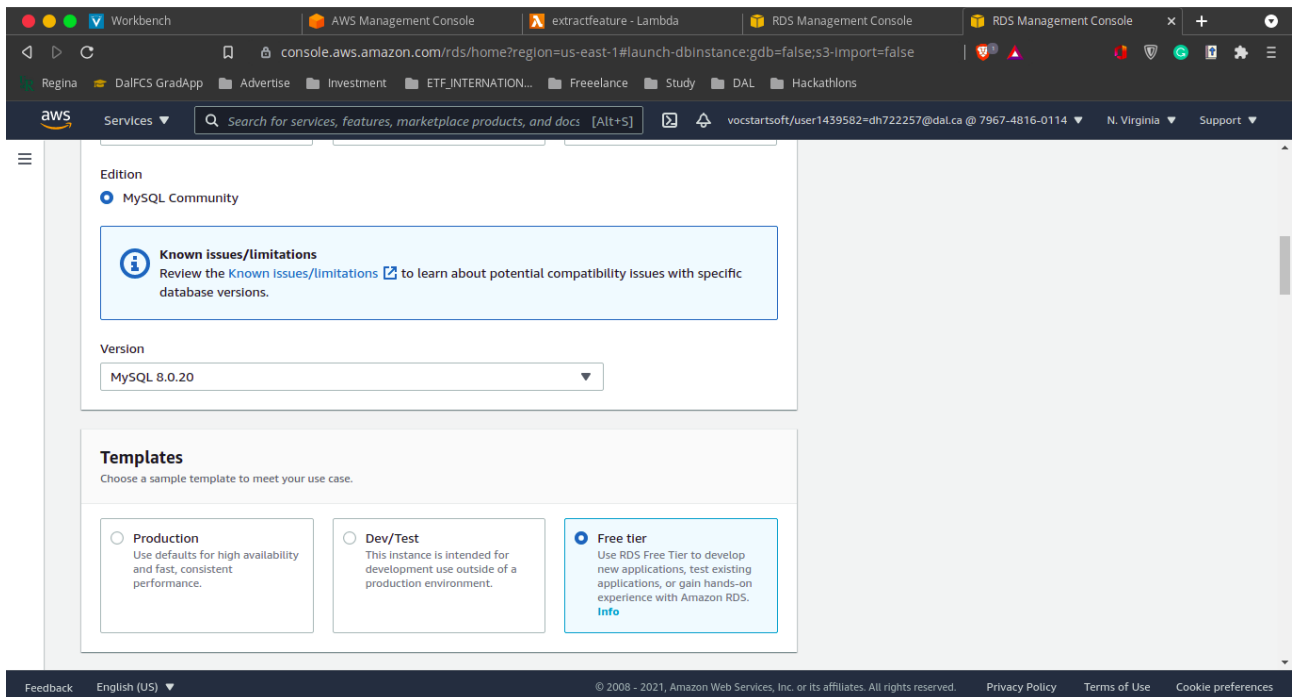
The screenshot displays the Amazon S3 console interface. On the left, a sidebar menu includes 'Buckets', 'Access Points', 'Object Lambda Access Points', 'Batch Operations', 'Access analyzer for S3', 'Block Public Access settings for this account', 'Storage Lens', 'Dashboards', 'AWS Organizations settings', 'Feature spotlight', and 'AWS Marketplace for S3'. The main content area is titled 'Amazon S3' and features an 'Account snapshot' section with a 'View Storage Lens dashboard' button. Below this, a 'Buckets (2)' section shows a table of buckets. The table has columns for Name, AWS Region, Access, and Creation date. Two buckets are listed: 'sampledatab0883311' and 'tagsb00883311', both in the 'US East (N. Virginia) us-east-1' region with public access. A 'Create bucket' button is visible in the top right of the buckets section.

Name	AWS Region	Access	Creation date
sampledatab0883311	US East (N. Virginia) us-east-1	Objects can be public	June 30, 2021, 21:46:26 (UTC+05:30)
tagsb00883311	US East (N. Virginia) us-east-1	Objects can be public	July 1, 2021, 00:26:16 (UTC+05:30)

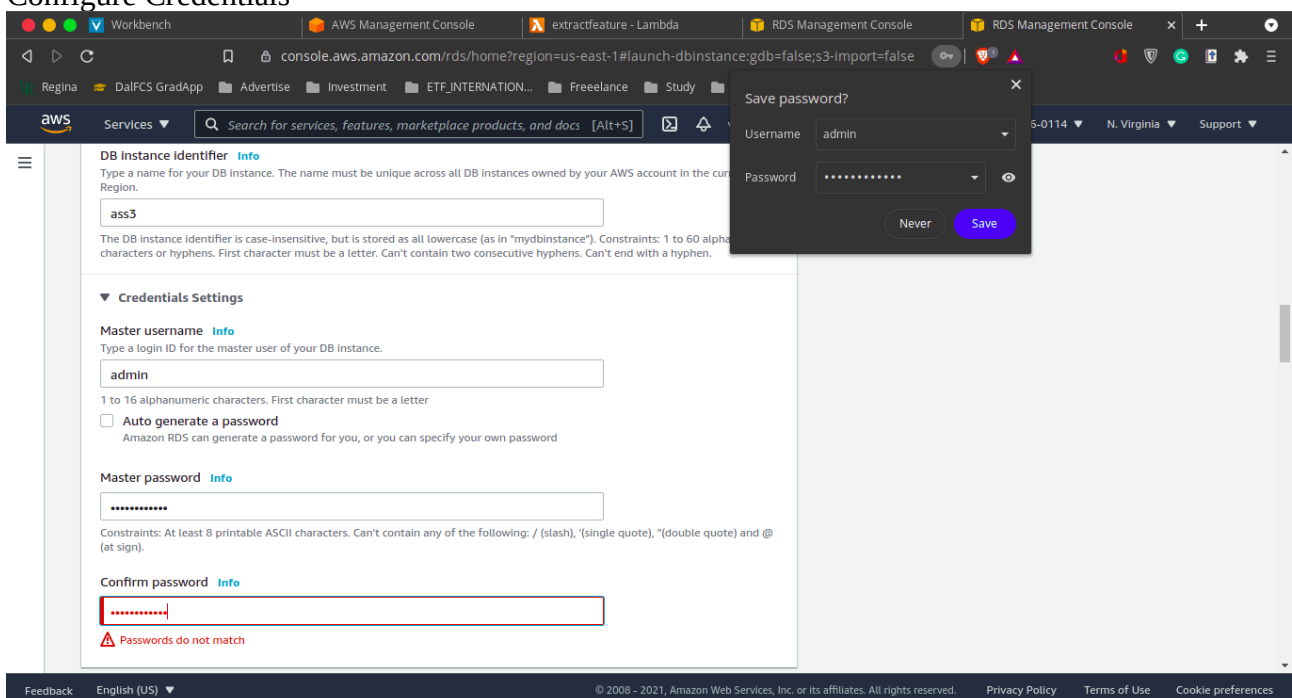
RDS DB CREATION

The screenshot shows the 'Create database' page in the Amazon RDS console. It features two main sections: 'Choose a database creation method' and 'Engine options'. In the 'Choose a database creation method' section, 'Standard create' is selected, with a description: 'You set all of the configuration options, including ones for availability, security, backups, and maintenance.' The 'Easy create' option is also visible, described as 'Use recommended best-practice configurations. Some configuration options can be changed after the database is created.' The 'Engine options' section displays six database engines: Amazon Aurora, MySQL (selected), MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server. Each engine is represented by an icon and a brief description.

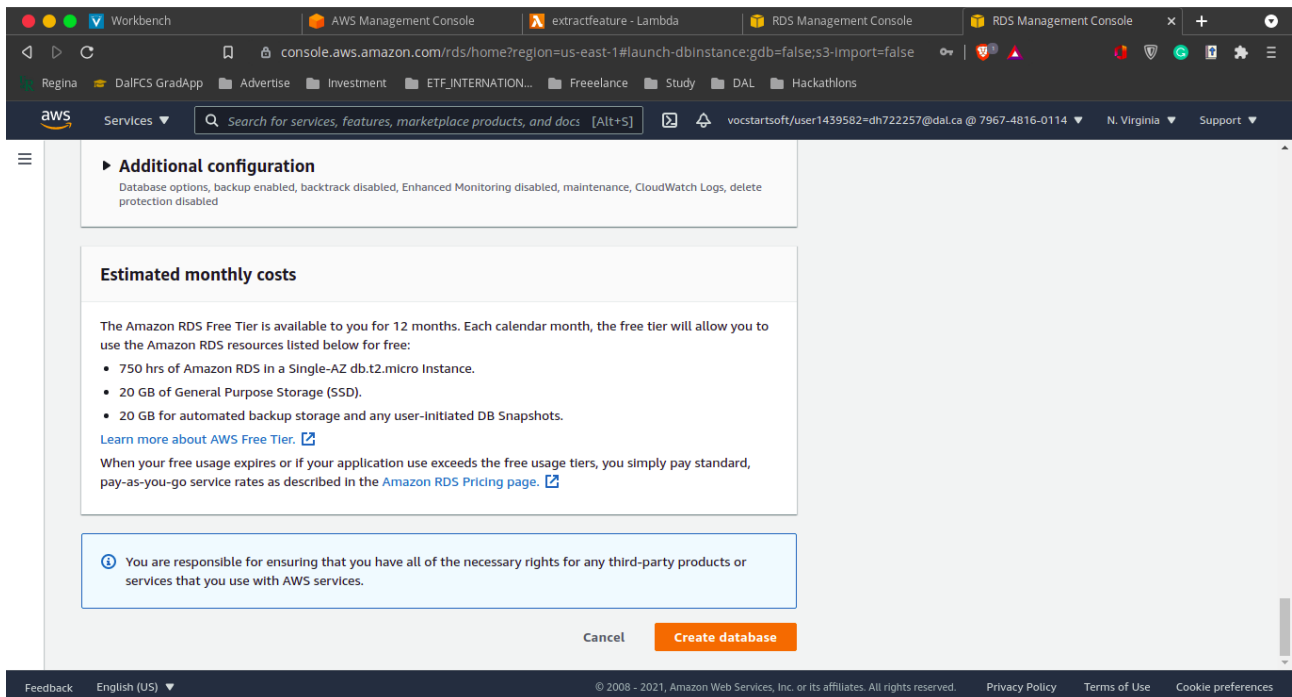
Free tier selection



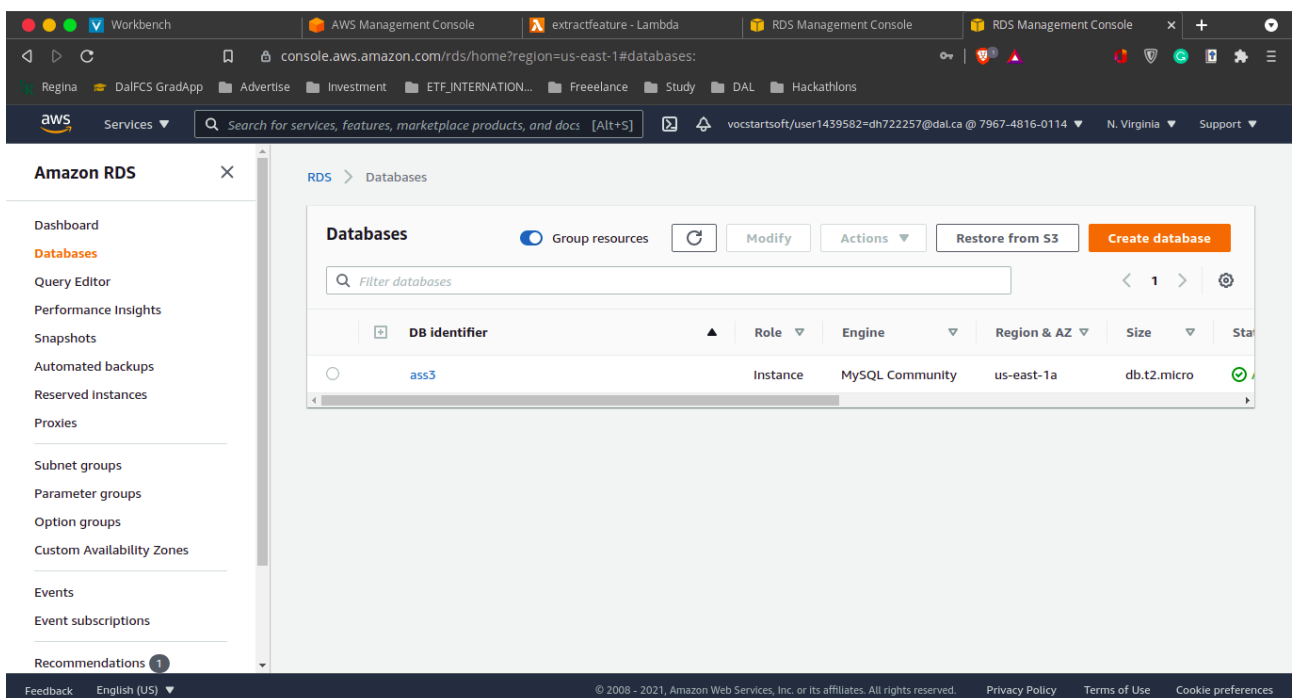
Configure Credentials



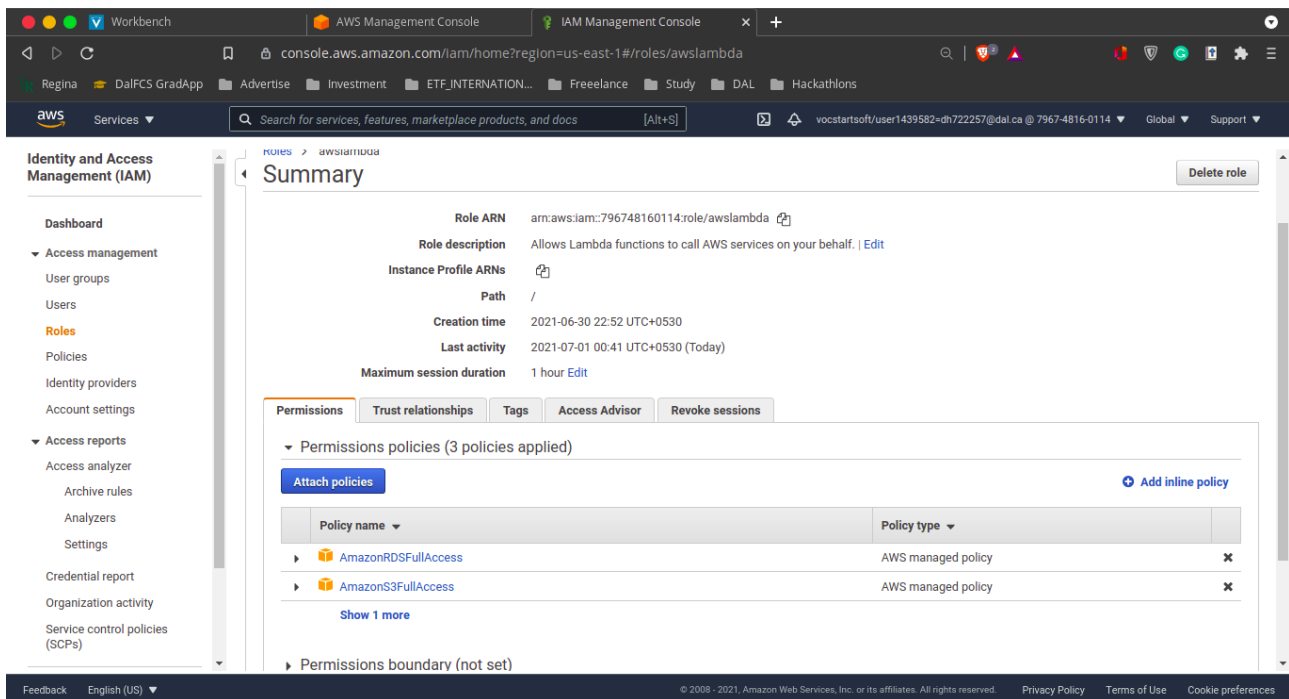
Click and RDS is ready



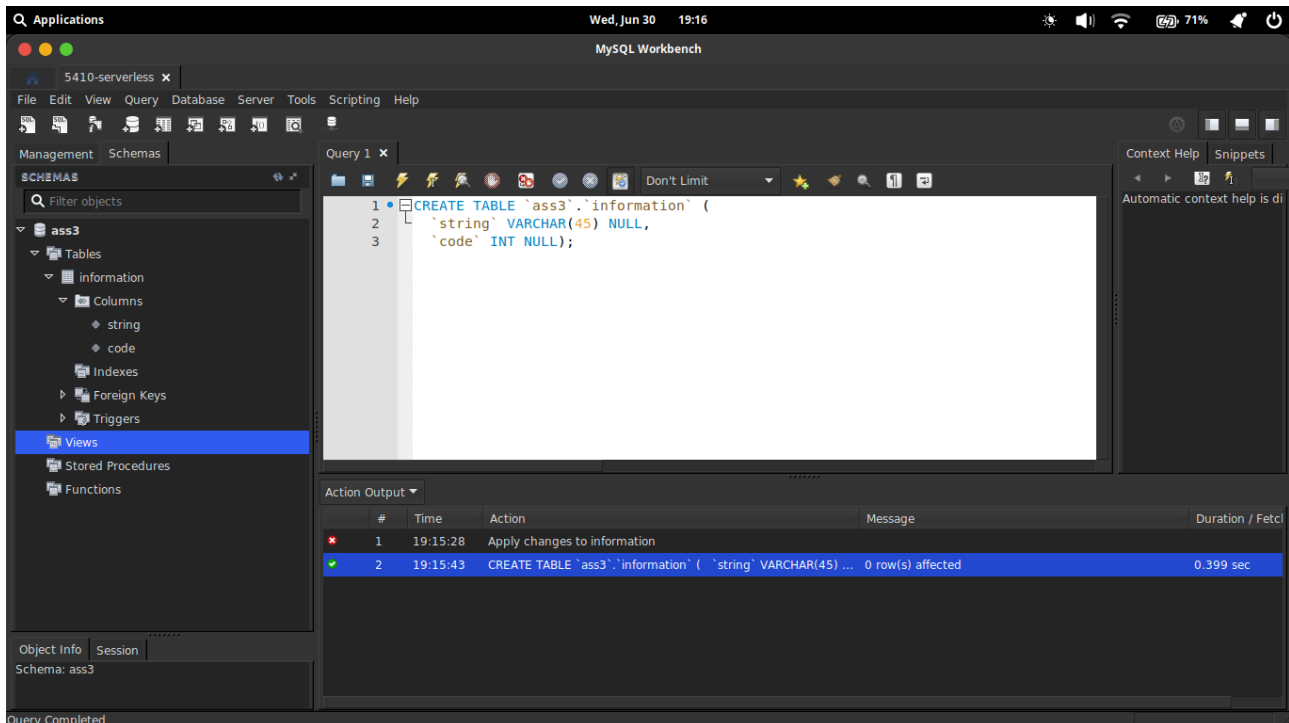
RDS is up and running



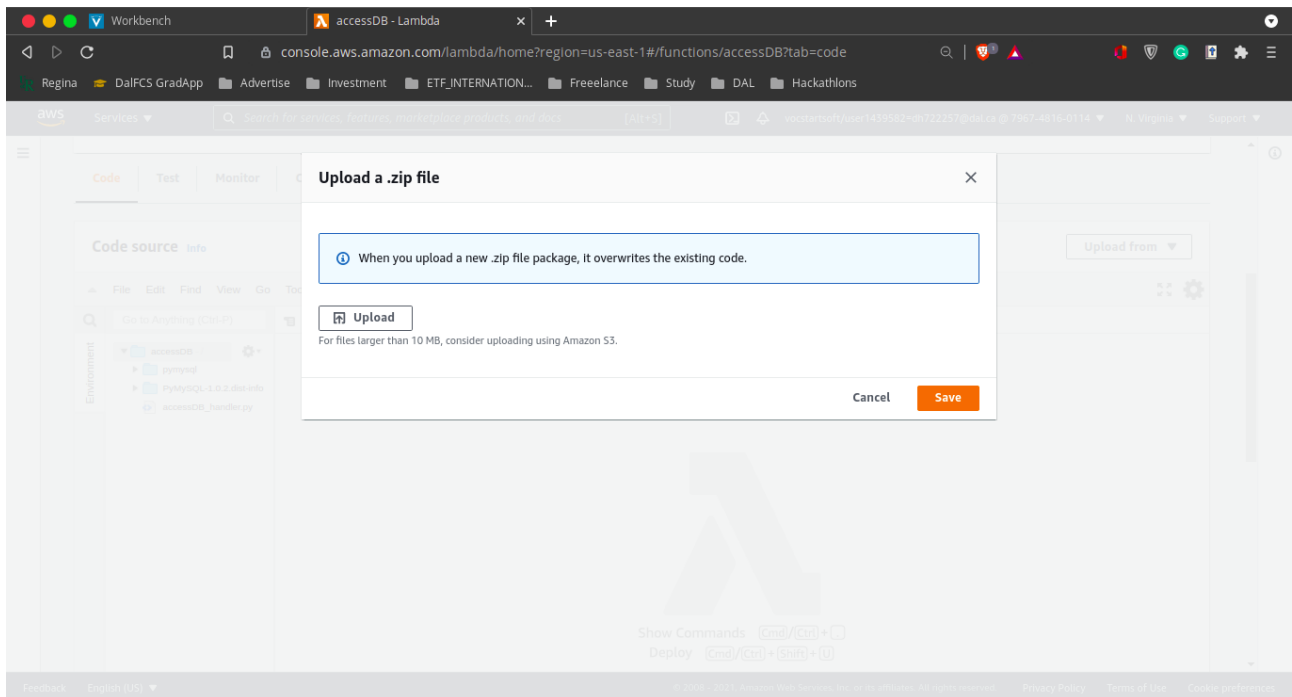
IAM: RDS rule for Lambda



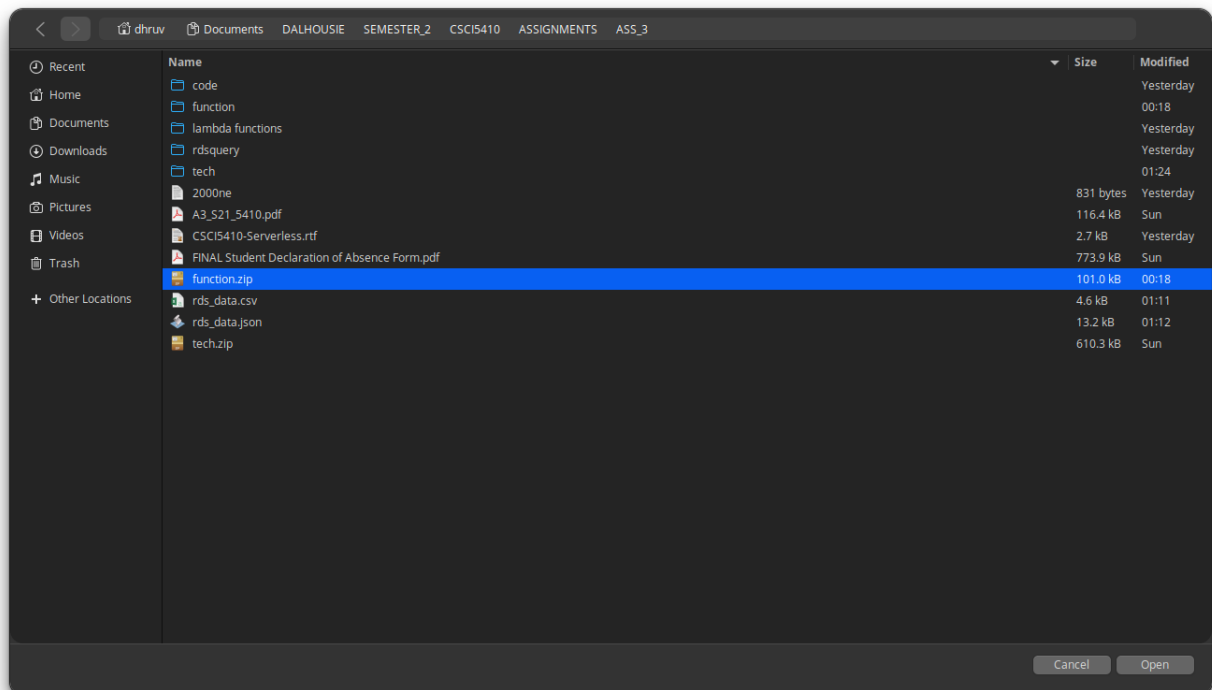
Create table with workbench



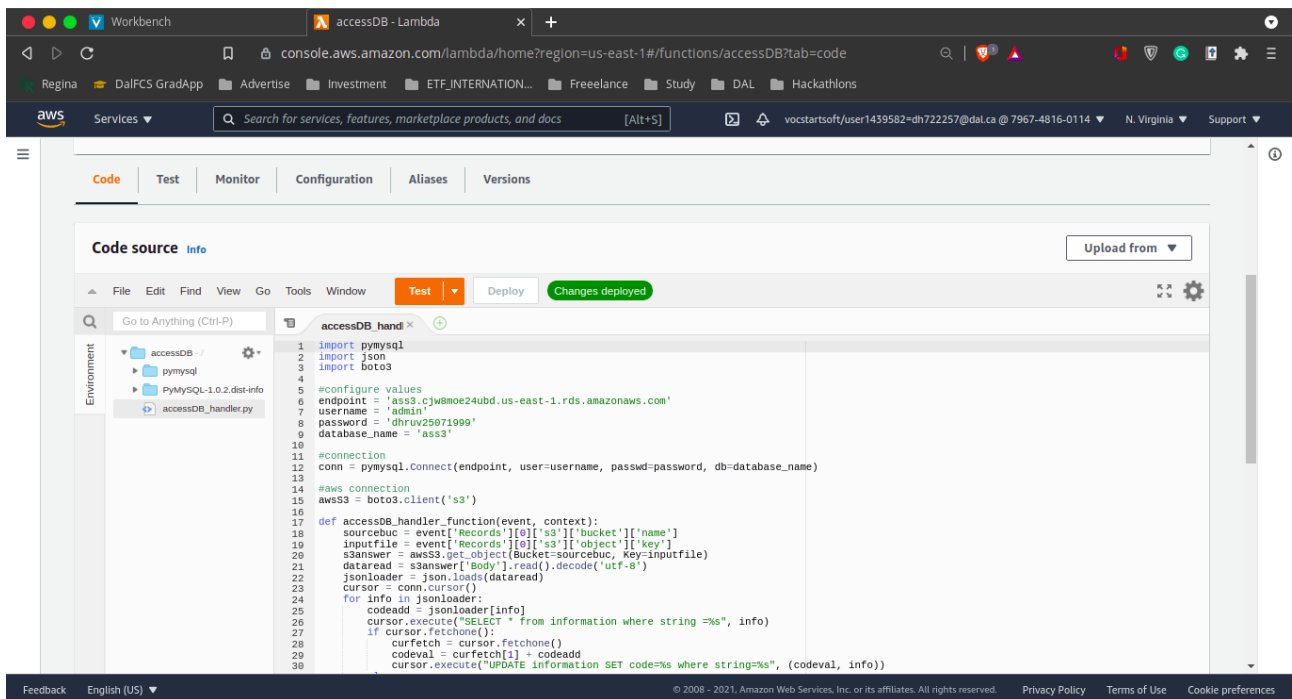
Another Lambda function upload by ZIP for pthon



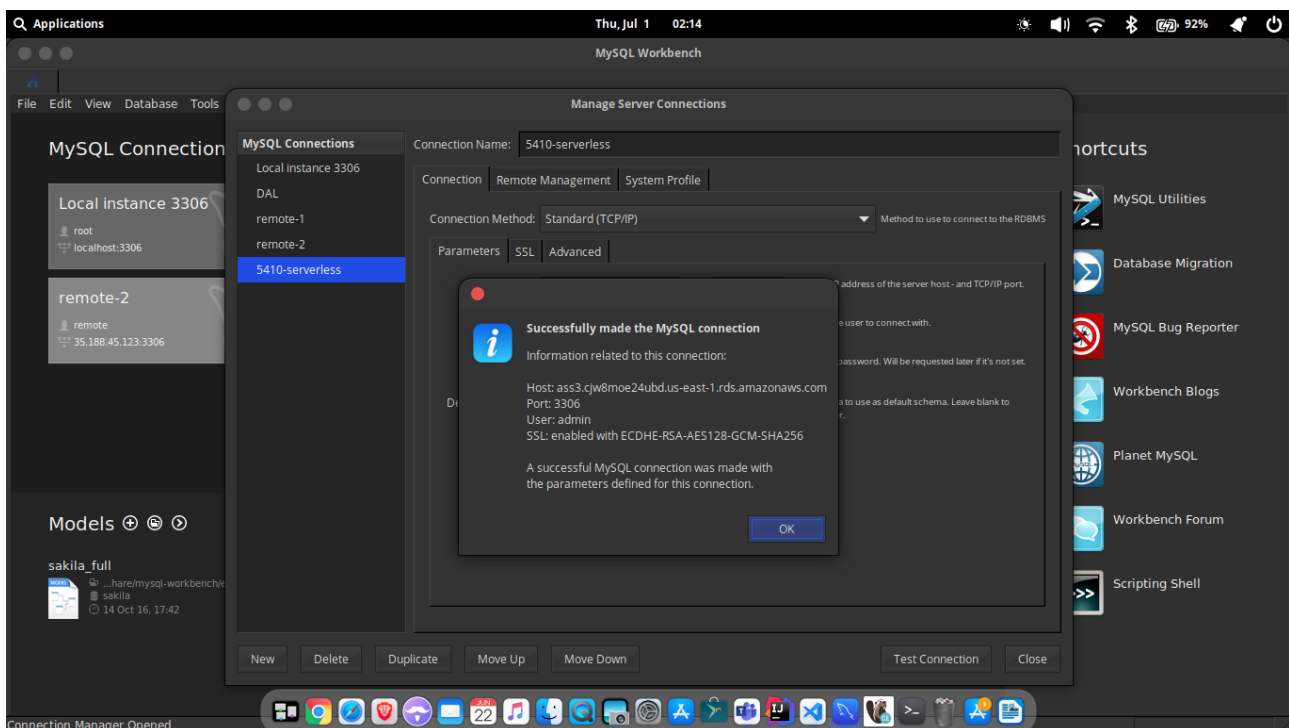
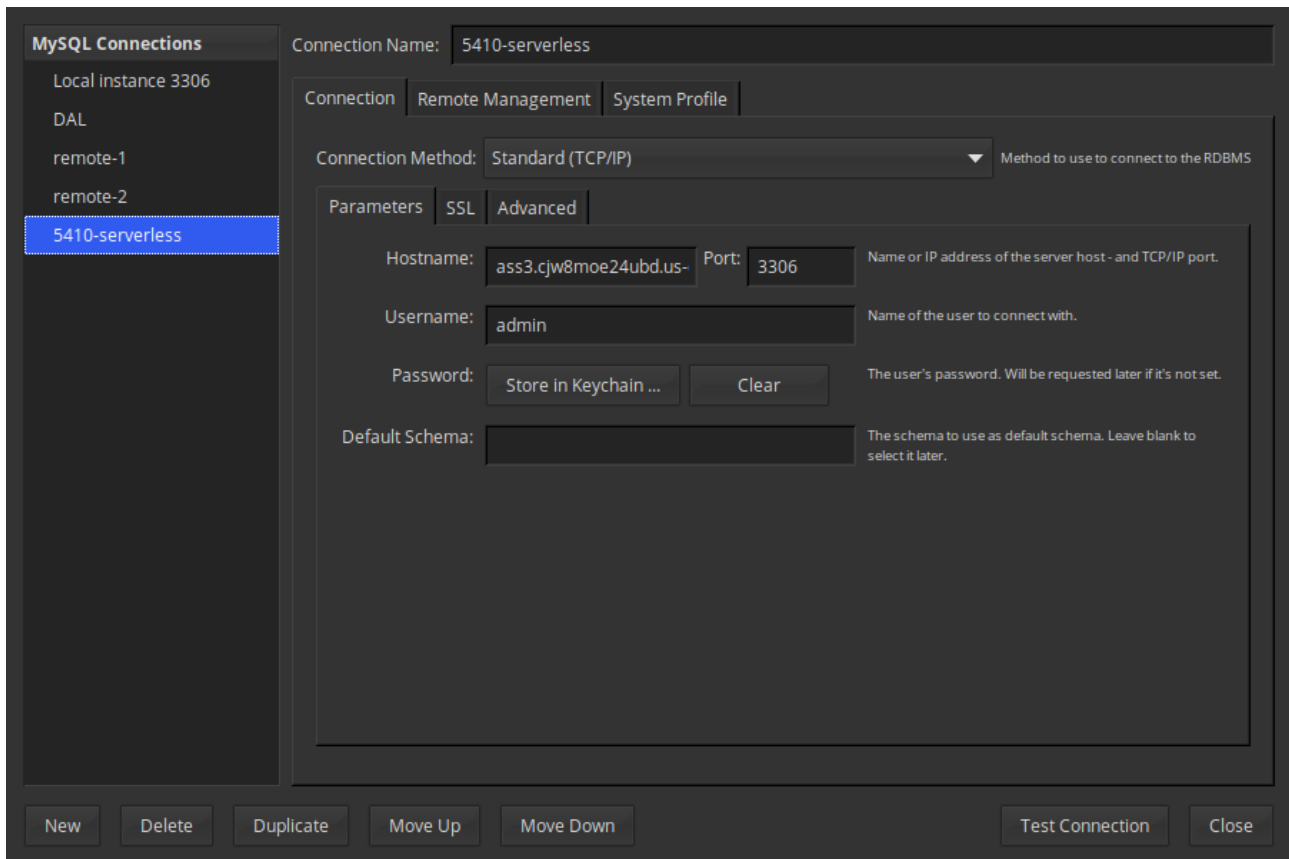
Taking the ZIP file and uploading the same



After Upload of all the files



After whole setup and lambda triggering the insert or update method in SQL Now let's check SQL Workbench.



After testing connect to db with ass-3 as schema

The screenshot displays the MySQL Workbench application window. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the 'SCHEMAS' tab with a search filter and a tree view containing 'ass3', 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'ass3' schema is selected. The central 'Query 1' editor contains the SQL statement: `1 • select * from information;`. Below the editor, the 'Result Grid' shows the query results in a table with columns '#', 'string', and 'code'. The results are as follows:

#	string	code
	Ink	1
	Asia	7
	The	11
	Kyrgyz	12
	Republic,	7
	Soviet	3
	This	1
	In	3

Below the result grid, the 'Action Output' tab shows the execution details:

#	Time	Action	Message	Duration / Fetch
1	02:15:29	select * from information	411 row(s) returned	0.268 sec / 0.0

The status bar at the bottom indicates 'Query Completed'.

CODE FOR ALL PART

EXTRACT FEATURE

```
console.log('Starting the trigger for Extract feature lambda');

const aws = require('aws-sdk');

const s3 = new aws.S3({ apiVersion: '2006-03-01' });

/*
  "AWS Lambda function handler in Node.js - AWS Lambda," Amazon.com, 2021. [Online].
  Available: https://docs.aws.amazon.com/lambda/latest/dg/nodejs-handler.html. [Accessed: 30-
  Jun-2021]
*/
exports.handler = async (event, context) => {
  const bucket = event.Records[0].s3.bucket.name;
  const key = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, ' '));
  const buc_parameters = {
    Bucket: bucket,
    Key: key,
  };
  console.log(buc_parameters)
  extractFeatures(buc_parameters);
};

const extractFeatures = (buc_parameters) => {
  var uploadParams = {
    Bucket: 'tagsb00883311',
    Key: "",
    Body: ""
  }
  var counter = '{ \n "' + buc_parameters.Key + 'ne" : [ ' ;
  s3.getObject(buc_parameters, function (err, metadata) {
    // Using === because we need to match objects strictly
    if (err && err.code === 'NotFound') {
      console.log("The object on which we are working is missing")
    }
    else {
      var file_info = metadata.Body.toString('utf-8');
      // Split with each new line
      file_info = file_info.split("\n")
      for (var i = 0; i < file_info.length; i++) {
        // splitting by space to get each word
        var identical_word = file_info[i].split(' ');
        for (var k = 0; k < identical_word.length; k++) {
          // when null continue
          if (identical_word[k] === "") {
            continue;
          }
        }
      }
    }
  });
}
```



```

// change to null
identical_word[k] = identical_word[k].replace(/[^a-zA-Z ]/g, "");
// check for lower and upper case
if (identical_word[k].charAt(0) === identical_word[k].charAt(0).toUpperCase()) {
    // when null continue
    if (identical_word[k] === "") {
        continue;
    }
    var format_string = "\n\t{ "
    format_string = format_string + "" + identical_word[k] + " : 1} ,";
    counter = counter + format_string;
}
}
}
// removing last element
counter = counter.slice(0, -1);
counter = counter + " \n]}";
// as required in assignment
var filename = buc_parameters.Key + ".ne"
uploadParams.Key = filename;
uploadParams.Body = counter;
s3.upload(uploadParams, function (s3Err, data) {
    if (s3Err) {
        console.log(s3Err);
    }
    console.log(`The uploading of file is done successfully at ${data.Location}`)
});
console.log(counter);
var countjsonarray = JSON.parse(counter);
}
})
}

```

ACCESS DB NODE

```
console.log('Starting accessDb lambda trigger');

const aws = require('aws-sdk');

const s3 = new aws.S3({ apiVersion: '2006-03-01' });

/*
  "AWS Lambda function handler in Node.js - AWS Lambda," Amazon.com, 2021. [Online].
  Available: https://docs.aws.amazon.com/lambda/latest/dg/nodejs-handler.html. [Accessed: 30-
  Jun-2021]
  */
exports.handler = async (event, context) => {
  const bucket = event.Records[0].s3.bucket.name;
  const key = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, ' '));
  const buc_parameters = {
    Bucket: bucket,
    Key: key,
  };
  var fdata
  s3.getObject(buc_parameters, function (err, metadata) {
    if (err && err.code === 'NotFound') {
      console.log("The object on which we are working is missing")
    } else {
      fdata = metadata.Body.toString('utf-8');
    }
  })
  accessDB(fdata, buc_parameters.Key)
};

const accessDB = async (arrayodstrings, k1) => {
  var pool = mysql.createPool({
    host: "ass3.cjw8moe24ubd.us-east-1.rds.amazonaws.com",
    user: "admin",
    password: "dhruv25071999",
    port: "3306",
    database: "ass3"
  });

  console.log(arrayodstrings[k1]);
  for (var a = 0; a < arrayodstrings[k1].length; a++) {
    for (var key in arrayodstrings[k1][a]) {
      await addtoDB(key, pool);
    }
  }
  pool.end();
}
```

```

function addtoDB(key, pool) {
  return new Promise(function (resolve) {
    var ch = "SELECT * FROM information where string= '" + key + "'";
    pool.query(ch, function (err, result) {
      if (result.length < 1) {
        var ins = "INSERT INTO information (string,code) VALUES ('" + key + "',1)";
        pool.query(ins, function (err, result) {
          if (err) {
            console.log(err);
          }
          resolve("Inserted completely");
        })
      } else {
        var up = "UPDATE information SET code = code + 1 WHERE string =' " + key + "'";
        pool.query(up, function (err, result) {
          if (err) {
            console.log(err);
          }
          resolve("updatation completed");
        })
      }
    })
  })
}

```

ACCESS DB PYTHON

```
import pymysql
import json
import boto3

#configure values
endpoint = 'ass3.cjw8moe24ubd.us-east-1.rds.amazonaws.com'
username = 'admin'
password = 'dhruv25071999'
database_name = 'ass3'

#connection
conn = pymysql.Connect(endpoint, user=username, passwd=password, db=database_name)

#aws connection
awsS3 = boto3.client('s3')

def accessDB_handler(event, context):
    sourcebuc = event['Records'][0]['s3']['bucket']['name']
    inputfile = event['Records'][0]['s3']['object']['key']
    s3answer = awsS3.get_object(Bucket=sourcebuc, Key=inputfile)
    dataread = s3answer['Body'].read().decode('utf-8')
    jsonloader = json.loads(dataread)
    cursor = conn.cursor()
    for info in jsonloader:
        codeadd = jsonloader[info]
        cursor.execute("SELECT * from information where string =%s", info)
        if cursor.fetchone():
            curfetch = cursor.fetchone()
            codeval = curfetch[1] + codeadd
            cursor.execute("UPDATE information SET code=%s where string=%s", (codeval, info))
        else:
            cursor.execute("INSERT INTO information values (%s,%s) ", (info, codeadd))
    conn.commit()
```

OUTPUT JSON FILE OF RDS

```
[{"string":"Ink", "code":1},
{"string":"Asia", "code":7},
{"string":"The", "code":11},
{"string":"Kyrgyz", "code":12},
{"string":"Republic,", "code":7},
{"string":"Soviet", "code":3},
{"string":"This", "code":1},
{"string":"In", "code":3},
{"string":"President,", "code":3},
{"string":"Askar", "code":3},
{"string":"Akaev,", "code":3},
{"string":"Parliamentary", "code":3},
{"string":"Presidential", "code":3},
{"string":"US", "code":3},
{"string":"Republic", "code":3},
{"string":"German", "code":3},
{"string":"Embassy,", "code":3},
{"string":"Soros", "code":3},
{"string":"Foundation", "code":3},
{"string":"It", "code":3},
{"string":"However,", "code":1},
{"string":"At", "code":3},
{"string":"UV", "code":1},
{"string":"If", "code":3},
{"string":"Likewise,", "code":3},
{"string":"These", "code":3},
{"string":"Autumn", "code":3},
{"string":"Republics,", "code":3},
{"string":"Ukraine", "code":3},
{"string":"Georgia.", "code":3},
{"string":"Widely", "code":3},
{"string":"Local", "code":3},
{"string":"Others,", "code":3},
{"string":"Coalition", "code":3},
{"string":"Non-governmental", "code":3},
{"string":"Organizations,", "code":3},
{"string":"Serbia,", "code":1},
{"string":"South", "code":3},
{"string":"Africa,", "code":3},
{"string":"Indonesia", "code":3},
{"string":"Turkey.", "code":3},
{"string":"Afghanistan", "code":3},
{"string":"Christian", "code":3},
{"string":"Islamic", "code":3},
{"string":"Other", "code":3},
{"string":"February.", "code":3},
{"string":"David", "code":3},
{"string":"Mikosz", "code":3},
{"string":"IFES,", "code":3},
{"string":"China", "code":7},
```

```
{ "string": "Chinese", "code": 12 },
{ "string": "According", "code": 3 },
{ "string": "Microsoft", "code": 7 },
{ "string": "Digital", "code": 1 },
{ "string": "PC", "code": 1 },
{ "string": "Nicholas", "code": 3 },
{ "string": "Negroponte", "code": 3 },
{ "string": "MIT's", "code": 3 },
{ "string": "Media", "code": 3 },
{ "string": "Labs", "code": 3 },
{ "string": "He", "code": 4 },
{ "string": "BBC", "code": 3 },
{ "string": "World", "code": 3 },
{ "string": "Service", "code": 3 },
{ "string": "Go", "code": 3 },
{ "string": "Linux-based", "code": 3 },
{ "string": "Technology", "code": 1 },
{ "string": "Wi-fi", "code": 3 },
{ "string": "Peru", "code": 1 },
{ "string": "A", "code": 3 },
{ "string": "Peru.", "code": 1 },
{ "string": "Virus", "code": 3 },
{ "string": "Christmas", "code": 7 },
{ "string": "Security", "code": 3 },
{ "string": "Windows", "code": 1 },
{ "string": "Apple", "code": 4 },
{ "string": "Google's", "code": 7 },
{ "string": "Search", "code": 3 },
{ "string": "Google", "code": 7 },
{ "string": "UK", "code": 4 },
{ "string": "TV", "code": 12 },
{ "string": "British", "code": 3 },
{ "string": "IBM", "code": 12 },
{ "string": "Linux", "code": 7 },
{ "string": "Xbox", "code": 7 },
{ "string": "EU", "code": 3 },
{ "string": "Global", "code": 3 },
{ "string": "PlayStation", "code": 7 },
{ "string": "Details", "code": 1 },
{ "string": "Sony's", "code": 3 },
{ "string": "San", "code": 1 },
{ "string": "Francisco", "code": 3 },
{ "string": "Monday.", "code": 3 },
{ "string": "Sony", "code": 1 },
{ "string": "Finding", "code": 3 },
{ "string": "Re-using", "code": 3 },
{ "string": "Research", "code": 3 },
{ "string": "Intel", "code": 4 },
{ "string": "Scientists", "code": 3 },
{ "string": "Britons", "code": 1 },
{ "string": "Sun", "code": 12 },
{ "string": "Microsystems", "code": 3 },
```

```
{"string":"Grid", "code":1},
{"string":"Sun.", "code":3},
{"string":"So-called", "code":3},
{"string":"Game", "code":3},
{"string":"Video", "code":3},
{"string":"Bioware", "code":1},
{"string":"Lasers", "code":3},
{"string":"An", "code":3},
{"string":"Indian", "code":3},
{"string":"Tata", "code":7},
{"string":"Teleservices", "code":3},
{"string":"Sony", "code":4},
{"string":"PSP", "code":1},
{"string":"Warnings", "code":3},
{"string":"Warning", "code":3},
{"string":"Net", "code":3},
{"string":"Piero", "code":12},
{"string":"Open", "code":3},
{"string":"Reboot", "code":3},
{"string":"Solutions", "code":3},
{"string":"Fake", "code":3},
{"string":"ID", "code":3},
{"string":"Trust", "code":3},
{"string":"Almost", "code":3},
{"string":"Mobile", "code":3},
{"string":"Third-generation", "code":3},
{"string":"That", "code":3},
{"string":"Iran", "code":1},
{"string":"Movie", "code":3},
{"string":"T-Mobile", "code":4},
{"string":"Unlike", "code":3},
{"string":"DVD", "code":12},
{"string":"DVDs", "code":7},
{"string":"Macrovision.", "code":3},
{"string":"Hotspot", "code":3},
{"string":"People", "code":3},
{"string":"Wireless", "code":3},
{"string":"Broadreach", "code":3},
{"string":"Skype", "code":7},
{"string":"Millions", "code":3},
{"string":"MP3", "code":7},
{"string":"Cheaper", "code":3},
{"string":"Domain", "code":4},
{"string":"Blind", "code":3},
{"string":"Moving", "code":3},
{"string":"Junk", "code":3},
{"string":"Spam", "code":3},
{"string":"Sporting", "code":3},
{"string":"Half-Life", "code":1},
{"string":"Bafta", "code":7},
{"string":"BT", "code":4},
{"string":"Mobiles", "code":1},
```

```
{"string":"Despite", "code":3},
{"string":"Jupiter.", "code":3},
{"string":"Concern", "code":3},
{"string":"RFID", "code":12},
{"string":"Consumers", "code":1},
{"string":"Ask", "code":12},
{"string":"Jeeves", "code":7},
{"string":"Bloglines", "code":7},
{"string":"Looks", "code":3},
{"string":"Kenyan", "code":3},
{"string":"Yahoo", "code":3},
{"string":"Yahoo,", "code":3},
{"string":"Tough", "code":3},
{"string":"Firms", "code":3},
{"string":"Creator", "code":3},
{"string":"Telewest", "code":4},
{"string":"Sky", "code":7},
{"string":"Plus", "code":7},
{"string":"Cable", "code":3},
{"string":"Plus.", "code":3},
{"string":"UK,", "code":3},
{"string":"PVRs", "code":7},
{"string":"Games", "code":1},
{"string":"Gadget", "code":3},
{"string":"Rich", "code":3},
{"string":"Viruses,", "code":3},
{"string":"Blog", "code":1},
{"string":"America", "code":3},
{"string":"Americans", "code":3},
{"string":"Software", "code":3},
{"string":"Savvy", "code":3},
{"string":"Internet", "code":3},
{"string":"New", "code":3},
{"string":"Nintendo", "code":7},
{"string":"Euro", "code":3},
{"string":"Nintendo's", "code":1},
{"string":"DS,", "code":7},
{"string":"Europe", "code":3},
{"string":"March,", "code":3},
{"string":"Smart", "code":3},
{"string":"Voters", "code":3},
{"string":"Voting", "code":3},
{"string":"Bloggies", "code":4},
{"string":"Nominations", "code":1},
{"string":"Sunday,", "code":3},
{"string":"Weblogs", "code":3},
{"string":"Blogs", "code":3},
{"string":"Latest", "code":3},
{"string":"Opera", "code":7},
{"string":"Web", "code":3},
{"string":"Spanish", "code":12},
{"string":"Spin", "code":3},
```



```
{ "string": "Norway", "code": 1 },
{ "string": "What", "code": 1 },
{ "string": "California", "code": 3 },
{ "string": "Honour", "code": 3 },
{ "string": "Bush", "code": 3 },
{ "string": "Blogger", "code": 3 },
{ "string": "Camera", "code": 1 },
{ "string": "Four", "code": 3 },
{ "string": "Nuclear", "code": 3 },
{ "string": "Pandas", "code": 3 },
{ "string": "Home", "code": 1 },
{ "string": "Podcasts", "code": 3 },
{ "string": "DIY", "code": 3 },
{ "string": "Portable", "code": 1 },
{ "string": "Viewers", "code": 1 },
{ "string": "Pompeii", "code": 1 },
{ "string": "Fast", "code": 3 },
{ "string": "Joke", "code": 3 },
{ "string": "Progress", "code": 3 },
{ "string": "By", "code": 3 },
{ "string": "Argonaut", "code": 7 },
{ "string": "Jez", "code": 1 },
{ "string": "San,", "code": 3 },
{ "string": "Dozens", "code": 3 },
{ "string": "Concerns", "code": 3 },
{ "string": "Robots", "code": 1 },
{ "string": "Attack", "code": 3 },
{ "string": "Loyalty", "code": 3 },
{ "string": "Doom", "code": 4 },
{ "string": "Sci-fi", "code": 3 },
{ "string": "Golden", "code": 3 },
{ "string": "Joystick", "code": 3 },
{ "string": "Commodore", "code": 12 },
{ "string": "Broadband", "code": 3 },
{ "string": "Slim", "code": 3 },
{ "string": "Seamen", "code": 3 },
{ "string": "Gates", "code": 12 },
{ "string": "Bill", "code": 3 },
{ "string": "Consumer", "code": 1 },
{ "string": "Electronics", "code": 3 },
{ "string": "Show", "code": 3 },
{ "string": "Las", "code": 3 },
{ "string": "Vegas,", "code": 3 },
{ "string": "Mr", "code": 4 },
{ "string": "GTA", "code": 3 },
{ "string": "Putting", "code": 3 },
{ "string": "Brother'", "code": 3 },
{ "string": "Literally", "code": 3 },
{ "string": "Some", "code": 3 },
{ "string": "London's", "code": 3 },
{ "string": "Science", "code": 3 },
{ "string": "Museum,", "code": 3 },
```

```
{"string":"Future", "code":3},
{"string":"Face", "code":3},
{"string":"One", "code":3},
{"string":"Jeremiah.", "code":3},
{"string":"Football", "code":4},
{"string":"Manager", "code":4},
{"string":"For", "code":3},
{"string":"Championship", "code":3},
{"string":"Indeed,", "code":3},
{"string":"CM", "code":4},
{"string":"PCs.", "code":3},
{"string":"Musicians", "code":1},
{"string":"ITunes", "code":3},
{"string":"Freeze", "code":3},
{"string":"DS", "code":12},
{"string":"No", "code":3},
{"string":"File-swappers", "code":3},
{"string":"Legal", "code":3},
{"string":"Online", "code":3},
{"string":"After", "code":3},
{"string":"Doors", "code":3},
{"string":"Thousands", "code":3},
{"string":"Who", "code":3},
{"string":"Metal", "code":7},
{"string":"Slug", "code":7},
{"string":"Like", "code":3},
{"string":"With", "code":3},
{"string":"But", "code":1},
{"string":"And", "code":1},
{"string":"Casual", "code":1},
{"string":"Supercomputer", "code":1},
{"string":"First", "code":3},
{"string":"Swap", "code":3},
{"string":"Ban", "code":3},
{"string":"Smartphones", "code":3},
{"string":"Skulls", "code":12},
{"string":"Owners", "code":3},
{"string":"Nokia", "code":3},
{"string":"Those", "code":3},
{"string":"Musical", "code":3},
{"string":"Analyst", "code":3},
{"string":"Multi-purpose", "code":3},
{"string":"Computer", "code":3},
{"string":"Your", "code":3},
{"string":"Launched", "code":3},
{"string":"How", "code":3},
{"string":"Remote", "code":3},
{"string":"Soon", "code":3},
{"string":"Gangsters", "code":3},
{"string":"Row", "code":3},
{"string":"Music", "code":3},
{"string":"Man", "code":3},
```

```
{ "string": "Toxic", "code": 3 },
{ "string": "Lifestyle", "code": 3 },
{ "string": "Faster", "code": 3 },
{ "string": "Instead", "code": 3 },
{ "string": "Anti-spam", "code": 3 },
{ "string": "Cyber", "code": 3 },
{ "string": "Gritty", "code": 3 },
{ "string": "Prince", "code": 3 },
{ "string": "Persia", "code": 3 },
{ "string": "Disney", "code": 7 },
{ "string": "Within", "code": 3 },
{ "string": "Satellite", "code": 3 },
{ "string": "Darfur", "code": 3 },
{ "string": "Aid", "code": 1 },
{ "string": "Sudanese", "code": 3 },
{ "string": "More", "code": 3 },
{ "string": "HP", "code": 12 },
{ "string": "Robotic", "code": 3 },
{ "string": "Hi-tech", "code": 3 },
{ "string": "Interactive", "code": 3 },
{ "string": "Londoners", "code": 3 },
{ "string": "Parents", "code": 3 },
{ "string": "Ways", "code": 3 },
{ "string": "Gamers", "code": 1 },
{ "string": "Set", "code": 3 },
{ "string": "Go-ahead", "code": 3 },
{ "string": "Television", "code": 3 },
{ "string": "Then", "code": 3 },
{ "string": "Now", "code": 3 },
{ "string": "Hollywood", "code": 3 },
{ "string": "Bond", "code": 7 },
{ "string": "Humanoid", "code": 3 },
{ "string": "Car-maker", "code": 3 },
{ "string": "Honda's", "code": 3 },
{ "string": "Asimo", "code": 7 },
{ "string": "Gamer", "code": 3 },
{ "string": "Poles", "code": 3 },
{ "string": "GameBoy", "code": 4 },
{ "string": "Santy", "code": 7 },
{ "string": "Speech", "code": 3 },
{ "string": "Napster", "code": 4 },
{ "string": "Half", "code": 3 },
{ "string": "UK's", "code": 3 },
{ "string": "Multimedia", "code": 1 },
{ "string": "Blinx", "code": 7 },
{ "string": "Rings", "code": 3 },
{ "string": "Gambling", "code": 3 },
{ "string": "Many", "code": 3 },
{ "string": "EA", "code": 7 },
{ "string": "Souped-up", "code": 3 },
{ "string": "Super", "code": 3 },
{ "string": "UK.", "code": 3 },
```

```
{"string":"Tech", "code":3},
{"string":"Speak", "code":3},
{"string":"Dublin", "code":3},
{"string":"Dublin's", "code":3},
{"string":"Hacker", "code":3},
{"string":"Once", "code":1},
{"string":"P2P", "code":19},
{"string":"Big", "code":3},
{"string":"Rivals", "code":3},
{"string":"Apple...", "code":3},
{"string":"What's", "code":3},
{"string":"Why", "code":3},
{"string":"Cell", "code":13},
{"string":"Mac", "code":19},
{"string":"Can", "code":3},
{"string":"Mini", "code":15},
{"string":"Ultra", "code":1},
{"string":"Thursday", "code":1},
{"string":"Virgin", "code":11},
{"string":"Radio", "code":12},
{"string":"Cebit", "code":14},
{"string":"Cebit,", "code":3},
{"string":"Hanover", "code":3},
{"string":"There", "code":7},
{"string":"Third", "code":3},
{"string":"Chip", "code":3},
{"string":"Slow", "code":3},
{"string":"Faster", "code":3},
{"string":"Anti-tremor", "code":3},
{"string":"Hitachi", "code":4},
{"string":"Japanese", "code":3},
{"string":"Emiew,", "code":3},
{"string":"Have", "code":3},
{"string":"PC?", "code":3},
{"string":"Gizmondo", "code":7},
{"string":"Confusion", "code":3},
{"string":"Bad", "code":3},
{"string":"Nasa", "code":3},
{"string":"Rolling", "code":3},
{"string":"Long", "code":3},
{"string":"PCs", "code":3},
{"string":"Court", "code":12},
{"string":"Judges", "code":3},
{"string":"France", "code":3},
{"string":"TV's", "code":3},
{"string":"Cebit", "code":14},
{"string":"Be", "code":3},
{"string":"Losing", "code":3},
{"string":"Be", "code":3}]
```

REFERENCES:

- [1]“AWS Lambda function handler in Node.js - AWS Lambda,” *Amazon.com*, 2021. [Online]. Available: <https://docs.aws.amazon.com/lambda/latest/dg/nodejs-handler.html>. [Accessed: 30-Jun-2021]
- [2]“Deploy Node.js Lambda functions with .zip file archives - AWS Lambda,” *Amazon.com*, 2021. [Online]. Available: <https://docs.aws.amazon.com/lambda/latest/dg/nodejs-package.html>. [Accessed: 30-Jun-2021]
- [3]“AWS Lambda function logging in Node.js - AWS Lambda,” *Amazon.com*, 2019. [Online]. Available: <https://docs.aws.amazon.com/lambda/latest/dg/nodejs-logging.html>. [Accessed: 30-Jun-2021]
- [4]“AWS Lambda function errors in Node.js - AWS Lambda,” *Amazon.com*, 2021. [Online]. Available: <https://docs.aws.amazon.com/lambda/latest/dg/nodejs-exceptions.html>. [Accessed: 30-Jun-2021]
- [5]edureka, “AWS Lambda Tutorial | AWS Tutorial for Beginners | AWS Cloud | AWS Lambda | AWS Training | Edureka,” *YouTube*. 24-Apr-2017 [Online]. Available: <https://www.youtube.com/watch?v=XZggsCITQdY>. [Accessed: 30-Jun-2021]
- [6]Amazon Web Services, “Introduction to AWS Lambda & Serverless Applications,” *YouTube*. 31-Jan-2019 [Online]. Available: <https://www.youtube.com/watch?v=EBSdyoO3goc>. [Accessed: 30-Jun-2021]
- [7]“AWS Lambda with Node.js: A Complete Getting Started Guide,” *Stackify*, 05-Jul-2019. [Online]. Available: <https://stackify.com/aws-lambda-with-node-js-a-complete-getting-started-guide/>. [Accessed: 30-Jun-2021]
- [8] “MySQL :: MySQL Workbench,” *Mysql.com*, 2019. [Online]. Available: <https://www.mysql.com/products/workbench/>