COMP4442 Service and Cloud Computing
Lab 4

Zuchao MA

Mar. 23, 2022

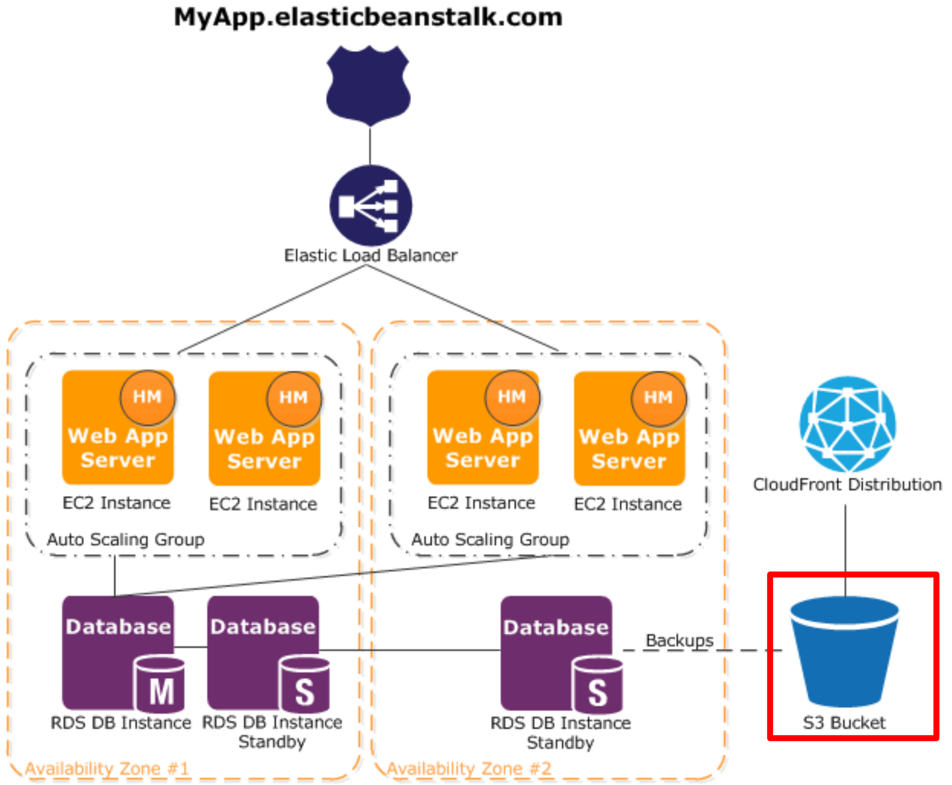
Outline

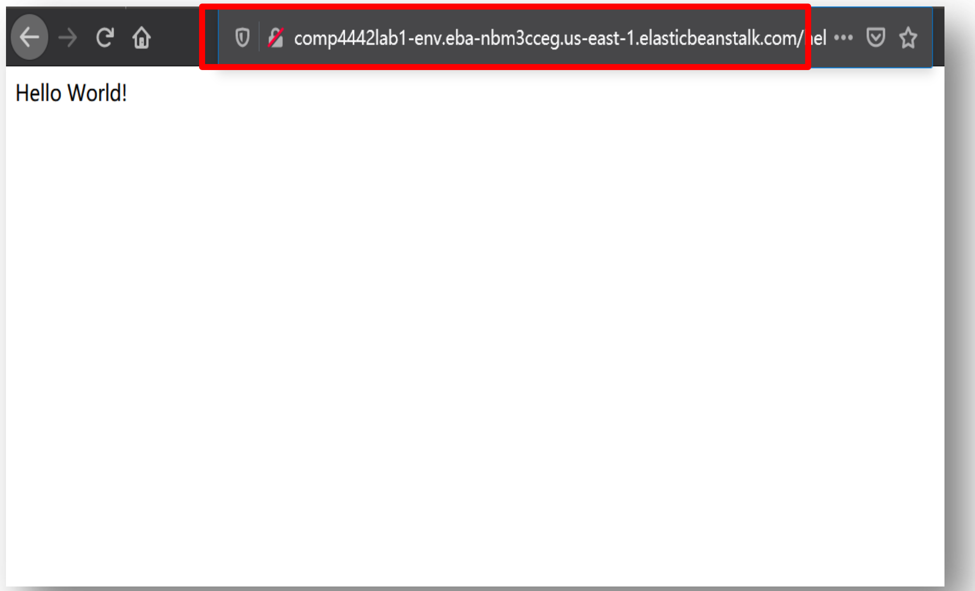
- Review of Labs

- How to develop dynamic web app with AWS Beanstalk
  - Structure of flask project
  - Create static webpage with flask
  - Create dynamic web app with flask
  - Deploy code in AWS Beanstalk

Review of Labs

- We learned how to use AWS services to develop a simple web application.
  - Use Elastic Beanstalk with simple configuration

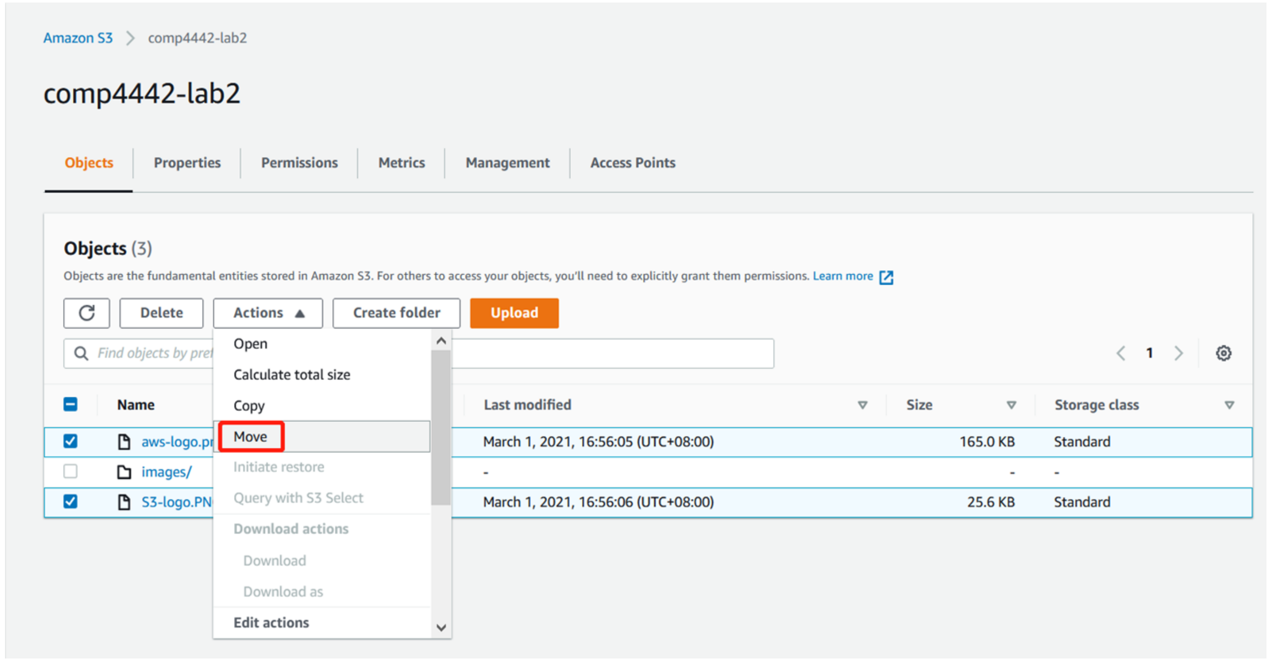



comp4442lab1-env.eba-nbm3cceg.us-east-1.elasticbeanstalk.com/

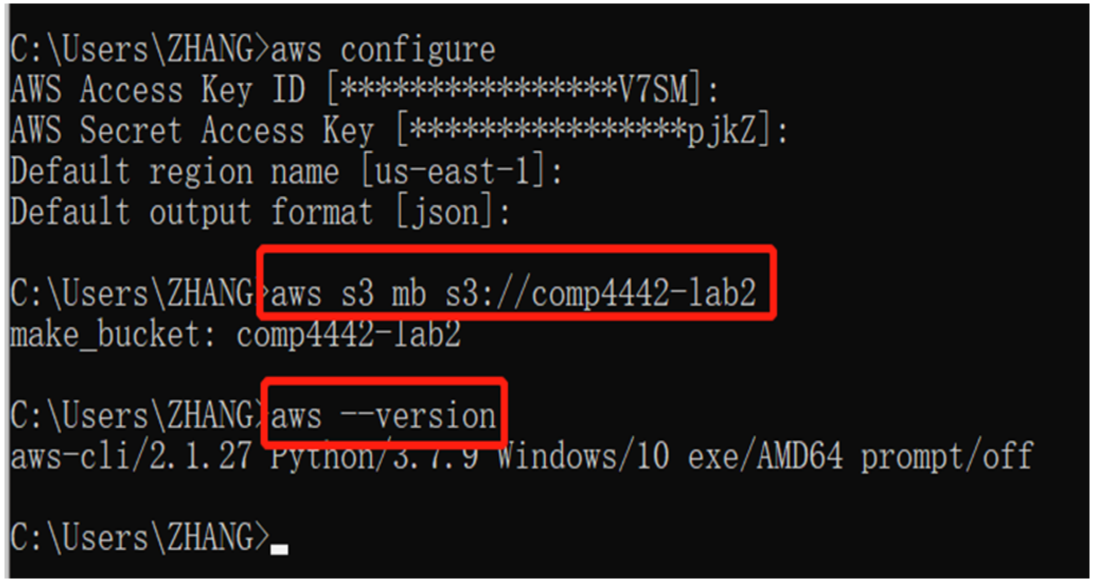
Hello World!

Review of Labs

- We learned how to use AWS S3 to manage files (upload, delete, move, etc.)
  – Via AWS console
  – Via AWS CLI



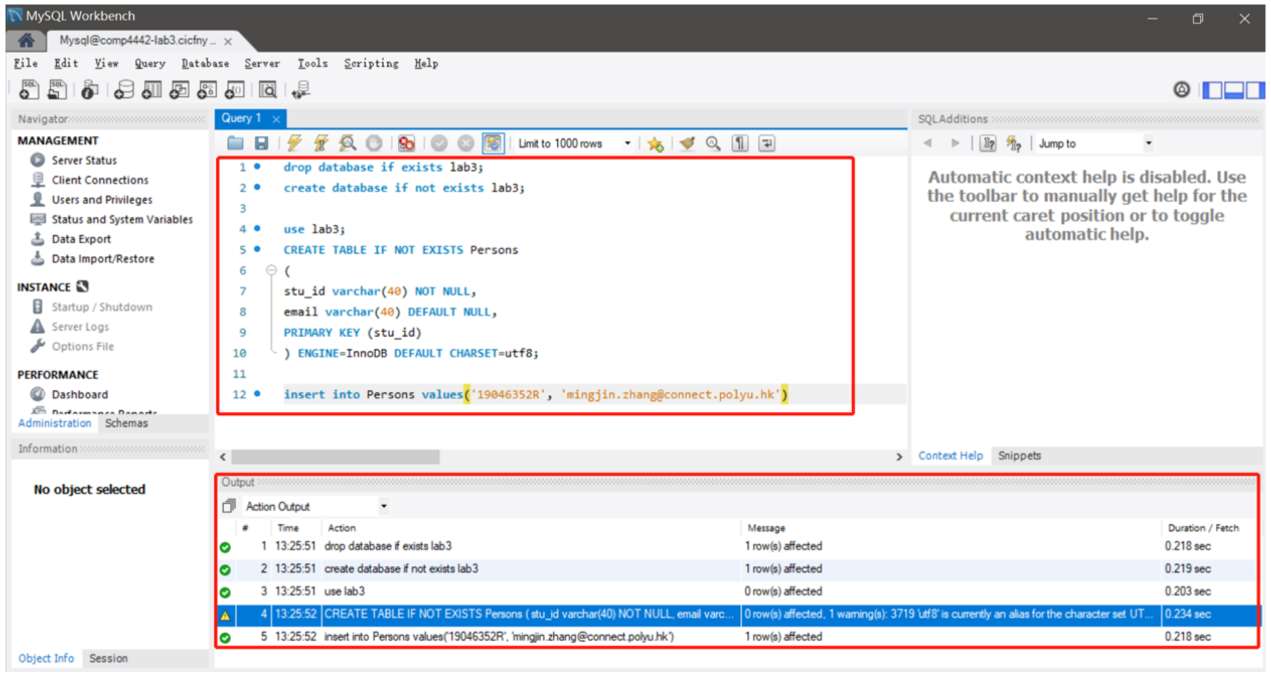
Name	Last modified	Size	Storage class
aws-logo.pn	March 1, 2021, 16:56:05 (UTC+08:00)	165.0 KB	Standard
images/	-	-	-
S3-logo.PN	March 1, 2021, 16:56:06 (UTC+08:00)	25.6 KB	Standard



```
C:\Users\ZHANG>aws configure
AWS Access Key ID [****************V7SM]:
AWS Secret Access Key [****************pjkZ]:
Default region name [us-east-1]:
Default output format [json]:

C:\Users\ZHANG>aws s3 mb s3://comp4442-lab2
make_bucket: comp4442-lab2

C:\Users\ZHANG>aws --version
aws-cli/2.1.27 Python/3.7.9 Windows/10 exe/AMD64 prompt/off

C:\Users\ZHANG>_
```

Review of Labs

- We learned how to use AWS RDS to manage structural data
  - Via MySQL Workbench
  - Via python



```sql
drop database if exists lab3;
create database if not exists lab3;

use lab3;
CREATE TABLE IF NOT EXISTS Persons
(
stu_id varchar(40) NOT NULL,
email varchar(40) DEFAULT NULL,
PRIMARY KEY (stu_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

insert into Persons values('19046352R', 'mingjin.zhang@connect.polyu.hk')
```

#	Time	Action	Message	Duration / Fetch
1	13:25:51	drop database if exists lab3	1 row(s) affected	0.218 sec
2	13:25:51	create database if not exists lab3	1 row(s) affected	0.219 sec
3	13:25:51	use lab3	0 row(s) affected	0.203 sec
4	13:25:52	CREATE TABLE IF NOT EXISTS Persons (stu_id varchar(40) NOT NULL, email varc...	0 row(s) affected, 1 warning(s): 3719 'utf8' is currently an alias for the character set UT...	0.234 sec
5	13:25:52	insert into Persons values('19046352R', 'mingjin.zhang@connect.polyu.hk')	1 row(s) affected	0.218 sec

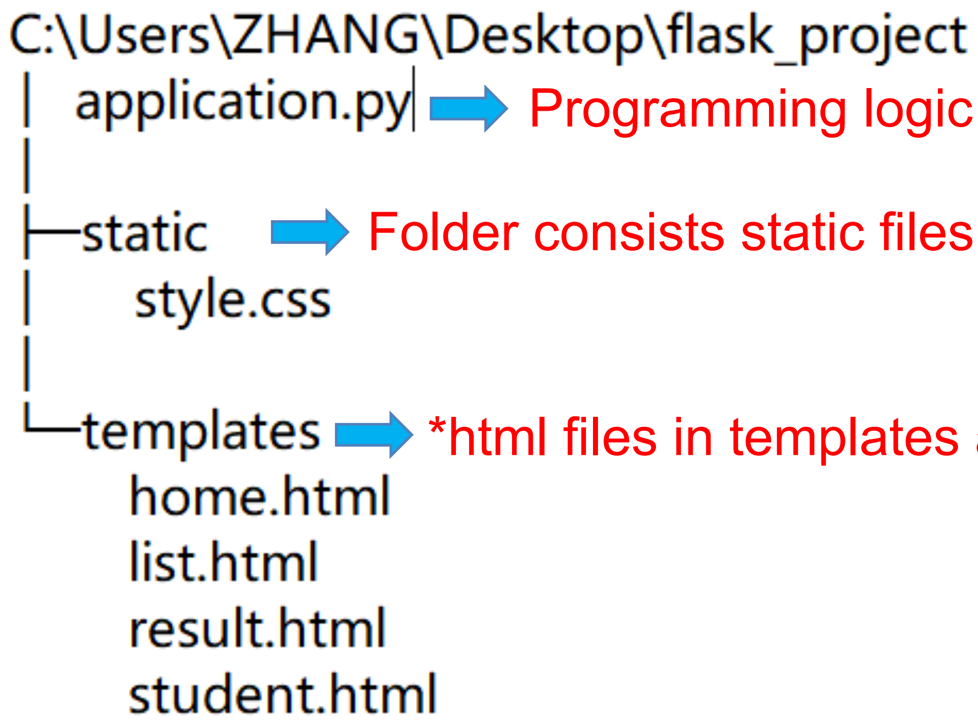
```python
import mysql.connector

mydb = mysql.connector.connect(
host = '<hostname>',
user = '<master username>',
port = '3306',
database = 'lab3',
passwd = '<master password>'

)

mycursor = mydb.cursor()
mycursor.execute("select * from Persons")
myresult = mycursor.fetchall()

for x in myresult:
    print(x)
```

Python Flask

- Flask is a popular lightweight web application framework. It is designed to make getting started quick and easy.
- Lab Preparation
  - Create an empty directory *lab4* and go to the directory `cd lab4`
  - Install the virtual environment `pip install virtualenv`
  - Create a virtual environment `virtualenv lab4`
  - Activate the virtual environment
    - `lab4\Scripts\activate` (windows)
    - `Source lab4/bin/activate` (macOS)
  - Install flask `pip install flask`
  - Install mysql-connector `pip install mysql-connector`

Python Flask

- Structure of a flask project



```
C:\Users\ZHANG\Desktop\flask_project
|   application.py  ➡ Programming logic control
|
├─static            ➡ Folder consists static files, e.g., images, css, js files (optional)
|       style.css
|
└─templates         ➡ *html files in templates are for front-end view (optional)
      home.html
      list.html
      result.html
      student.html
```

Create Static Webpage with Flask

- Create the *application.py* file with the following code

```
from flask import Flask

application = Flask(__name__)

@application.route("/")
def index():
    return "Your Flask App Works!"

@application.route("/hello")
def hello():
    return "Hello World!"


if __name__ == "__main__":
    application.run(port=5000, debug=True)
```

→ From flask package import Flask class

→ Instantiate the class to create the application object

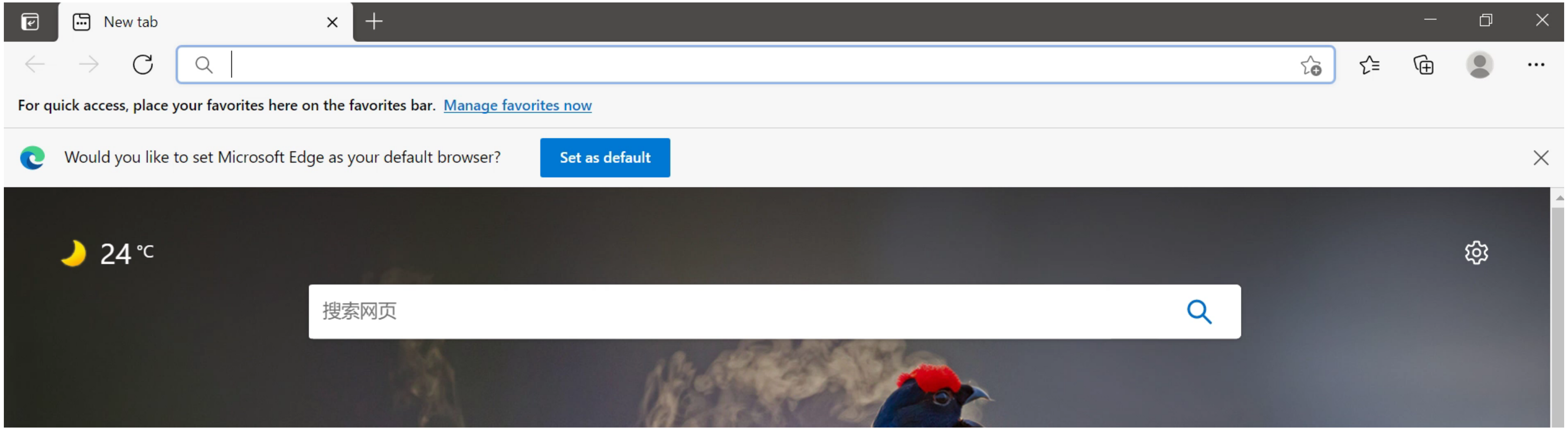
→ Server will call the index() to response to http://127.0.0.1/5000

→ Server will call the hello() to response to http://127.0.0.1/hello

→ Program entry

Create Dynamic Web App with Flask

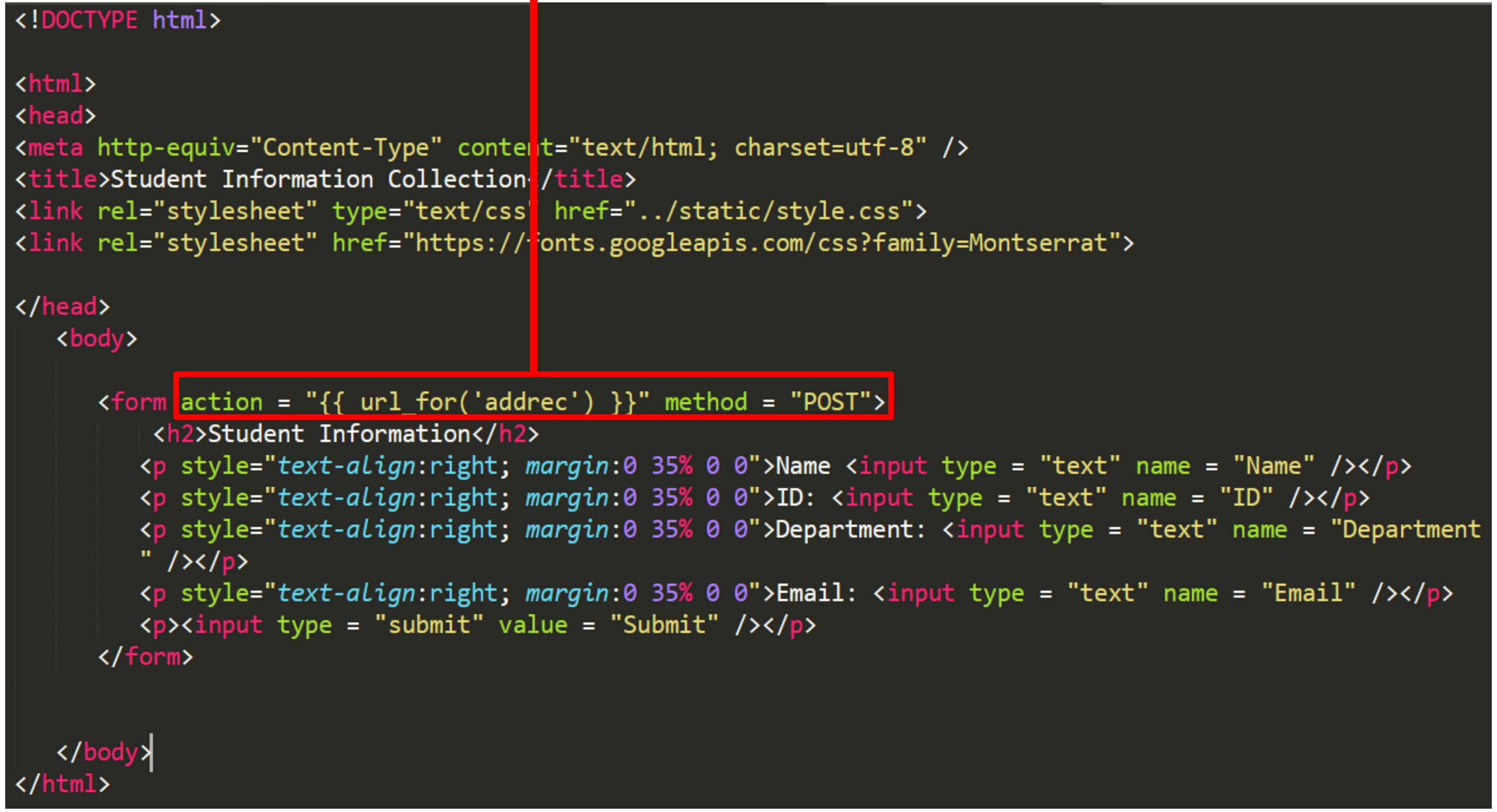
- We will develop a student information collection web application
  - A HTML form to enable students input information
  - A database to store the input information
  - A HTML table to query and display the input information


New tab
For quick access, place your favorites here on the favorites bar. Manage favorites now
Would you like to set Microsoft Edge as your default browser?
Set as default
24 °C
搜索网页

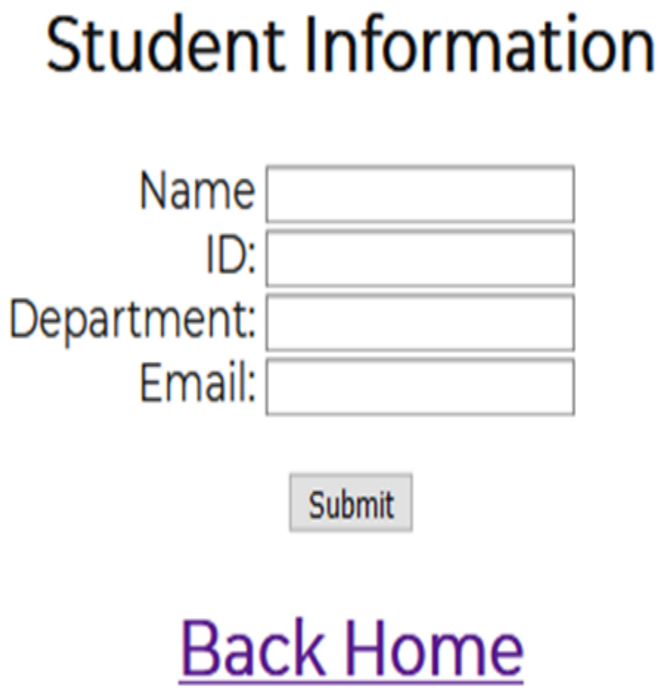
Create Dynamic Web App with Flask

- First, we create the student information form

/addrec to handle the post request

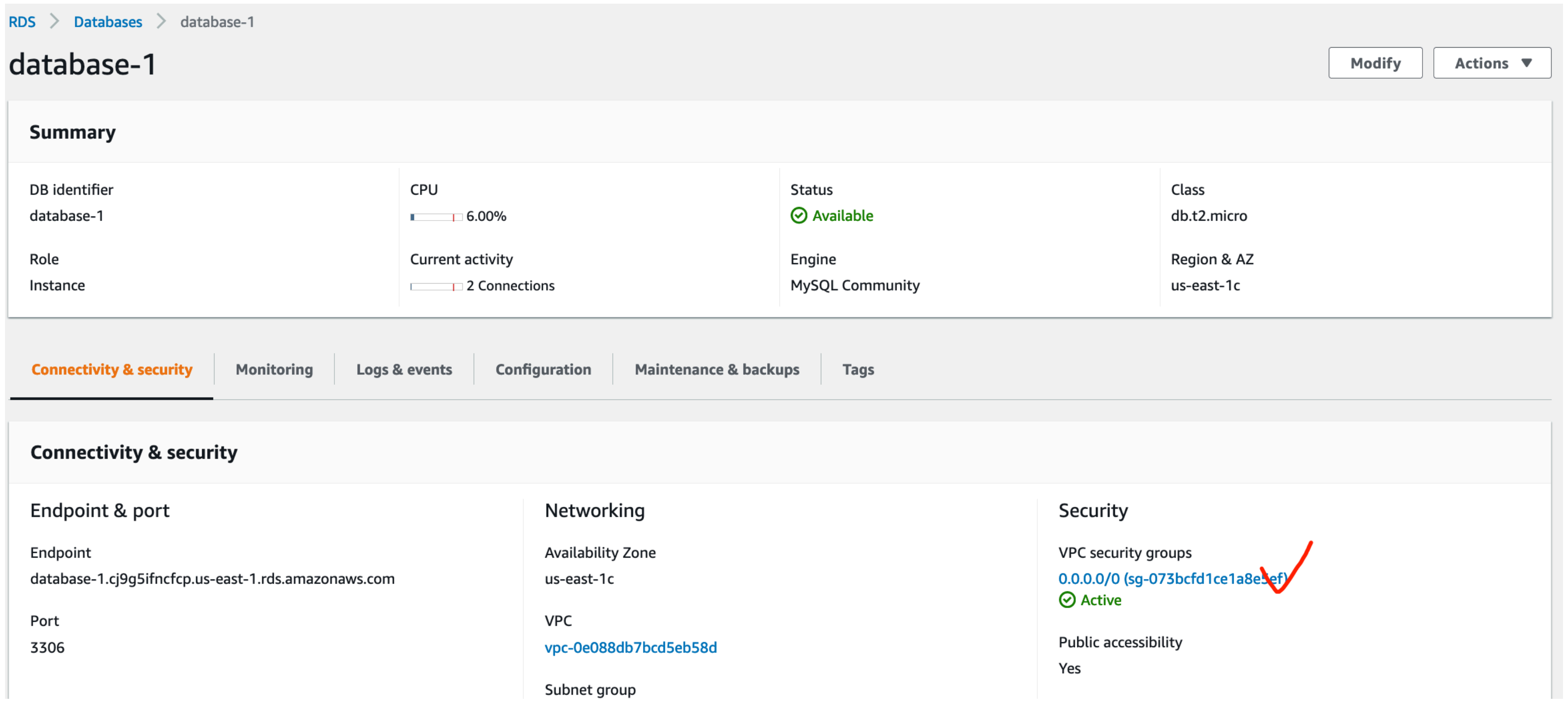


```
<!DOCTYPE html>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Student Information Collection</title>
<link rel="stylesheet" type="text/css" href="../static/style.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Montserrat">

</head>
   <body>

      <form action = "{{ url_for('addrec') }}" method = "POST">
          <h2>Student Information</h2>
         <p style="text-align:right; margin:0 35% 0 0">Name <input type = "text" name = "Name" /></p>
         <p style="text-align:right; margin:0 35% 0 0">ID: <input type = "text" name = "ID" /></p>
         <p style="text-align:right; margin:0 35% 0 0">Department: <input type = "text" name = "Department
         " /></p>
         <p style="text-align:right; margin:0 35% 0 0">Email: <input type = "text" name = "Email" /></p>
         <p><input type = "submit" value = "Submit" /></p>
      </form>


   </body>
</html>
```



Prepare the database

- Check the database connection setting



RDS > Databases > database-1

database-1

Modify | Actions ▼

Summary

DB identifier	CPU	Status	Class
database-1	6.00%	Available	db.t2.micro
Role	Current activity	Engine	Region & AZ
Instance	2 Connections	MySQL Community	us-east-1c

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | Tags

Connectivity & security

Endpoint & port

Endpoint
database-1.cj9g5ifncfcp.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1c

VPC
vpc-0e088db7bcd5eb58d

Subnet group

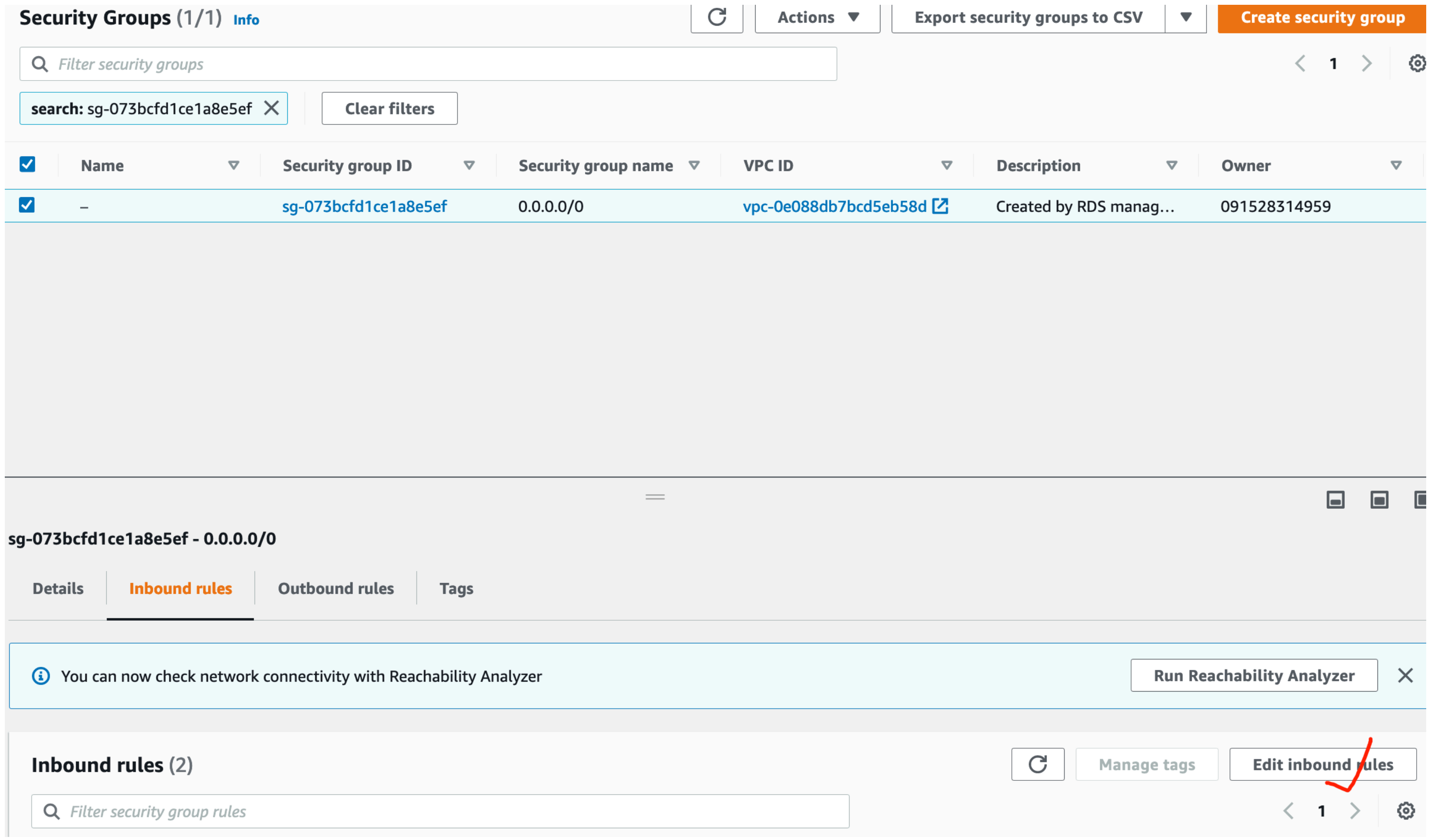
Security

VPC security groups
0.0.0.0/0 (sg-073bcfd1ce1a8e5ef) ✓
Active

Public accessibility
Yes

Prepare the database

- Check the database connection setting


Security Groups (1/1) Info

Actions ▼ | Export security groups to CSV ▼ | Create security group

Filter security groups

search: sg-073bcfd1ce1a8e5ef | Clear filters

☑	Name	Security group ID	Security group name	VPC ID	Description	Owner
☑	–	sg-073bcfd1ce1a8e5ef	0.0.0.0/0	vpc-0e088db7bcd5eb58d	Created by RDS manag…	091528314959

sg-073bcfd1ce1a8e5ef - 0.0.0.0/0

Details | Inbound rules | Outbound rules | Tags

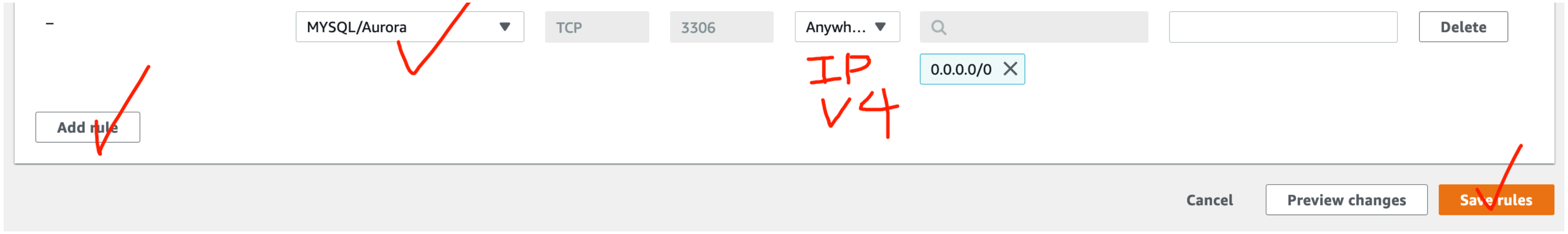
You can now check network connectivity with Reachability Analyzer | Run Reachability Analyzer

Inbound rules (2) | Manage tags | Edit inbound rules ✓

Filter security group rules

Prepare the database

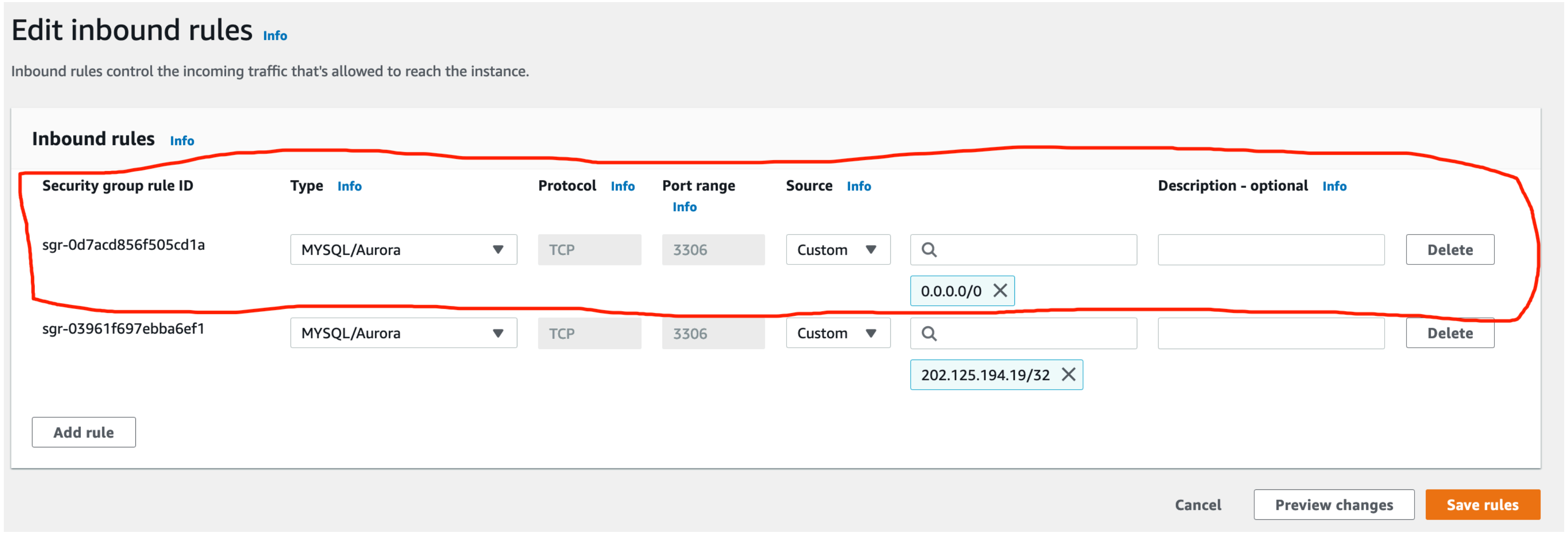
- Check the database connection setting



–	MYSQL/Aurora ✓	TCP	3306	Anywh...			Delete
				IP v4	0.0.0.0/0		

Add rule ✓

Cancel | Preview changes | Save rules ✓



Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0d7acd856f505cd1a	MYSQL/Aurora	TCP	3306	Custom	0.0.0.0/0		Delete
sgr-03961f697ebba6ef1	MYSQL/Aurora	TCP	3306	Custom	202.125.194.19/32		Delete

Add rule

Cancel | Preview changes | Save rules

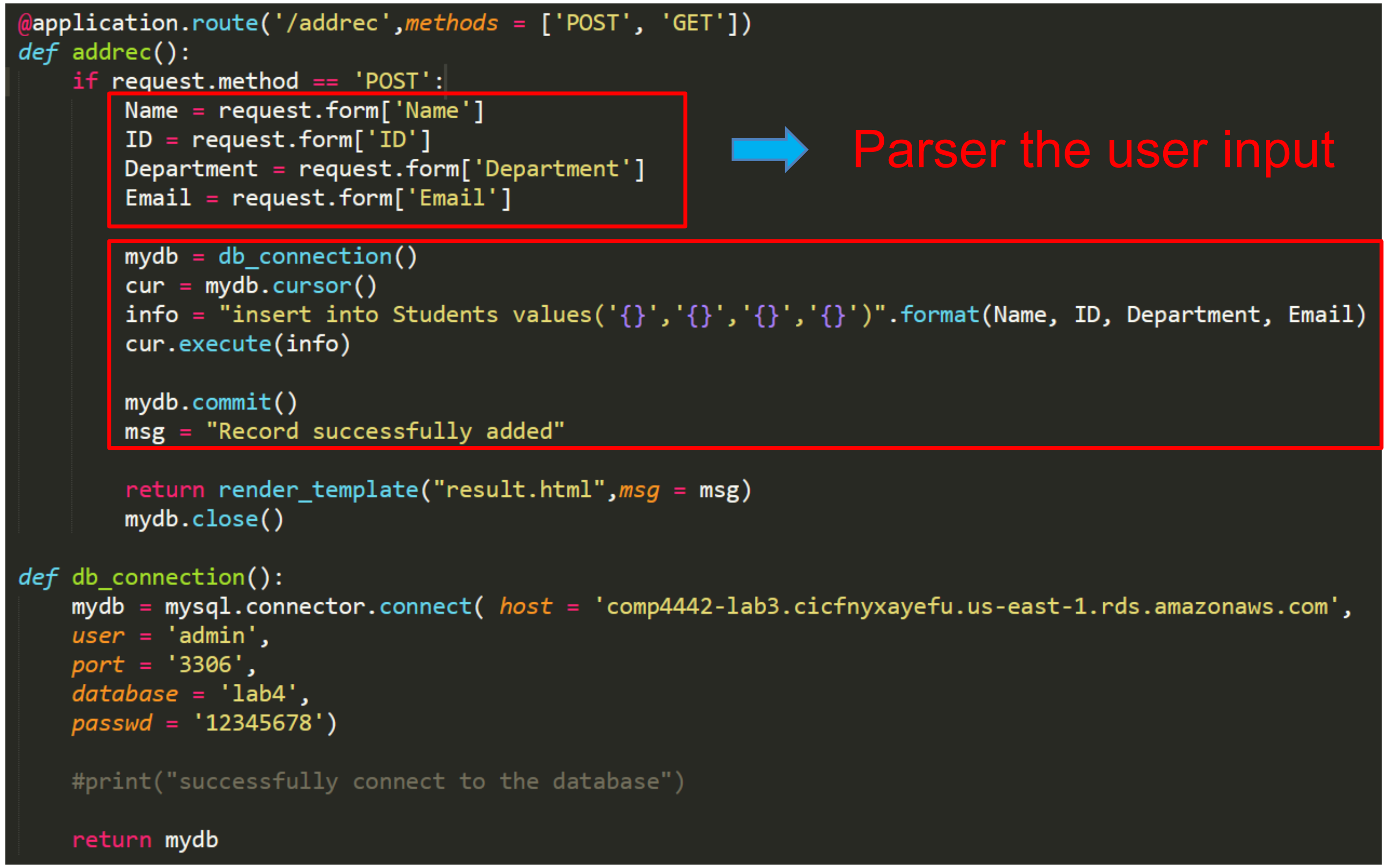
Create Dynamic Web App with Flask

- Create a new table in the database we created in lab-03

```
drop database if exists lab4;
create database if not exists lab4;

use lab4;
CREATE TABLE IF NOT EXISTS Students
(
Name varchar(40) NOT NULL,
ID varchar(40) NOT NULL,
Department varchar(40) NOT NULL,
Email varchar(40) DEFAULT NULL,
PRIMARY KEY (ID)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Create Dynamic Web App with Flask

- Implement the response function to insert student information to the AWS RDS database



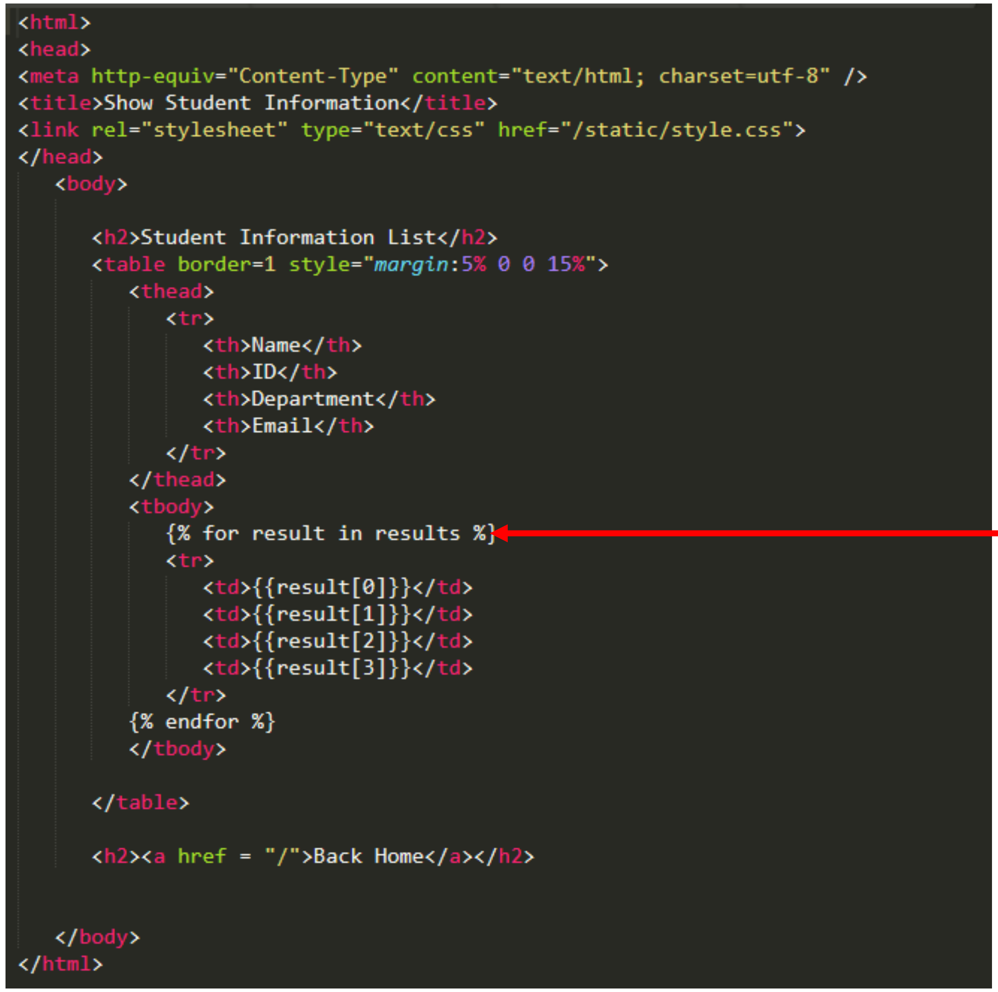
```
@application.route('/addrec',methods = ['POST', 'GET'])
def addrec():
    if request.method == 'POST':
        Name = request.form['Name']
        ID = request.form['ID']
        Department = request.form['Department']
        Email = request.form['Email']

        mydb = db_connection()
        cur = mydb.cursor()
        info = "insert into Students values('{}','{}','{}','{}')".format(Name, ID, Department, Email)
        cur.execute(info)

        mydb.commit()
        msg = "Record successfully added"

        return render_template("result.html",msg = msg)
        mydb.close()

def db_connection():
    mydb = mysql.connector.connect( host = 'comp4442-lab3.cicfnyxayefu.us-east-1.rds.amazonaws.com',
    user = 'admin',
    port = '3306',
    database = 'lab4',
    passwd = '12345678')

    #print("successfully connect to the database")

    return mydb
```

Parser the user input

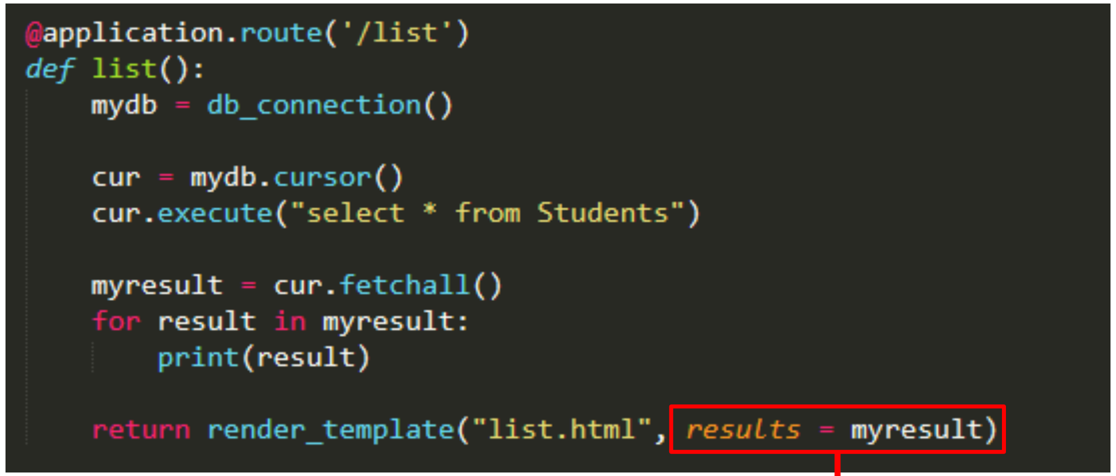
Add information to database

Create Dynamic Web App with Flask

- Query the information on the RDS database



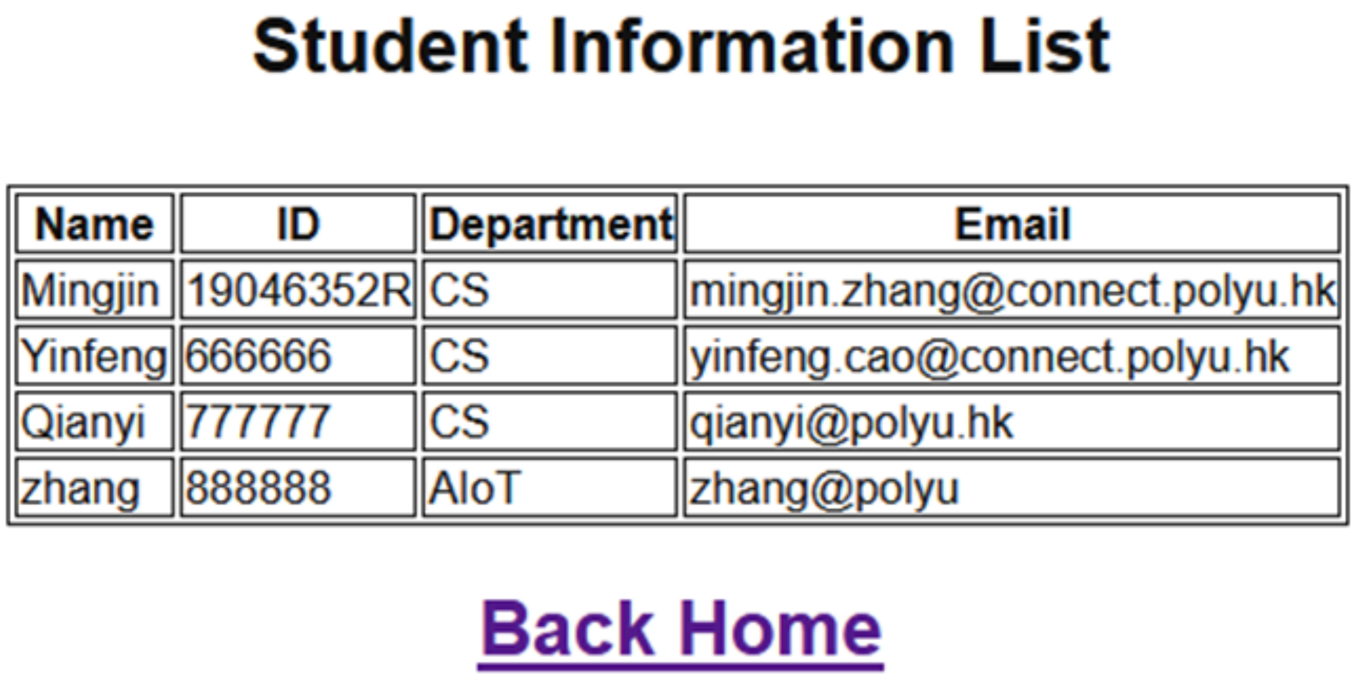
```html
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Show Student Information</title>
<link rel="stylesheet" type="text/css" href="/static/style.css">
</head>
   <body>

      <h2>Student Information List</h2>
      <table border=1 style="margin:5% 0 0 15%">
         <thead>
            <tr>
               <th>Name</th>
               <th>ID</th>
               <th>Department</th>
               <th>Email</th>
            </tr>
         </thead>
         <tbody>
            {% for result in results %}
            <tr>
               <td>{{result[0]}}</td>
               <td>{{result[1]}}</td>
               <td>{{result[2]}}</td>
               <td>{{result[3]}}</td>
            </tr>
         {% endfor %}
         </tbody>

      </table>

      <h2><a href = "/">Back Home</a></h2>


   </body>
</html>
```



```python
@application.route('/list')
def list():
    mydb = db_connection()

    cur = mydb.cursor()
    cur.execute("select * from Students")

    myresult = cur.fetchall()
    for result in myresult:
        print(result)

    return render_template("list.html", results = myresult)
```

Create Dynamic Web App with Flask

- Check the student information at "*http://127.0.0.1:5000/list*".



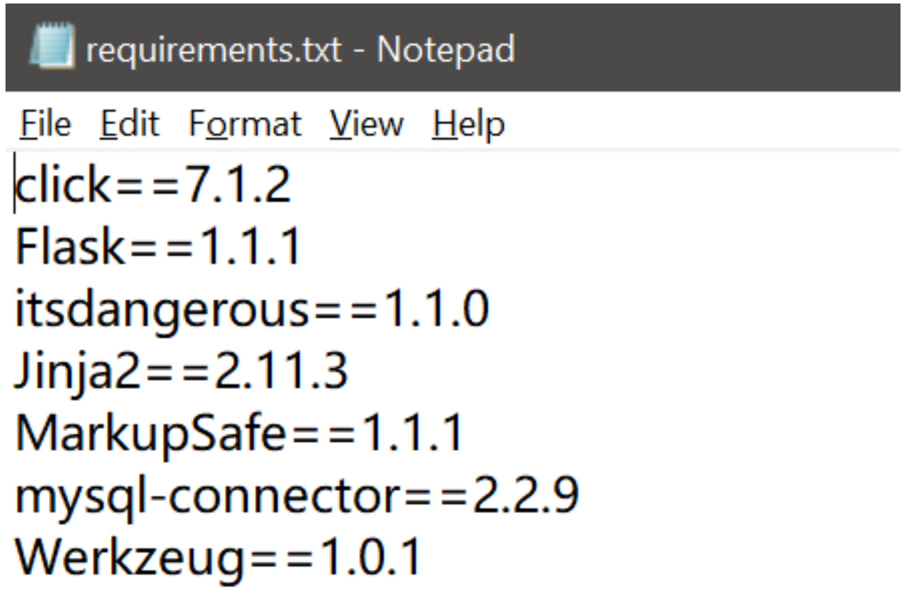
Student Information List

Name	ID	Department	Email
Mingjin	19046352R	CS	mingjin.zhang@connect.polyu.hk
Yinfeng	666666	CS	yinfeng.cao@connect.polyu.hk
Qianyi	777777	CS	qianyi@polyu.hk
zhang	888888	AIoT	zhang@polyu

Back Home

Deploy Code in AWS Beanstalk

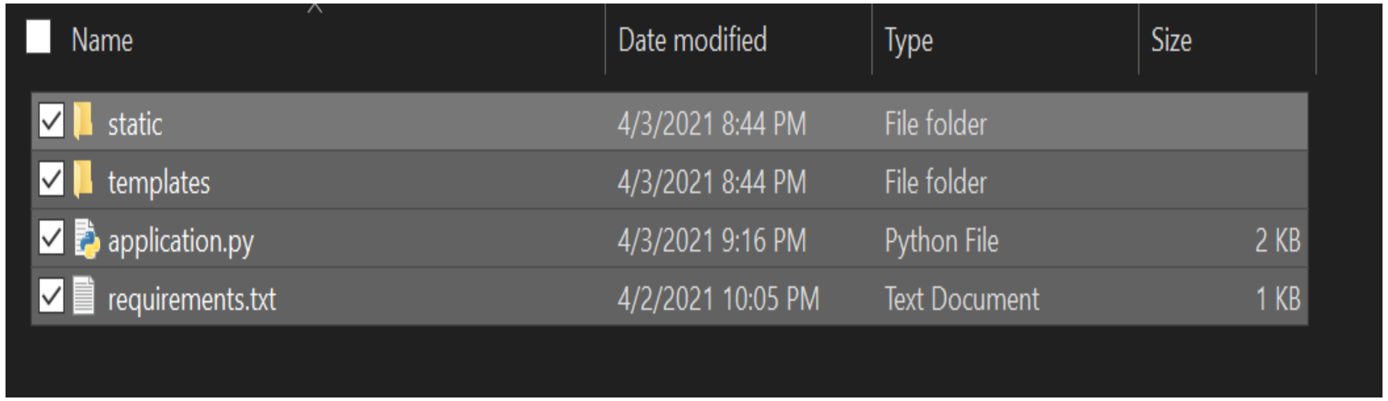
- To deploy the code, we need to configure the requirements.txt.
  - `pip freeze > requirement.txt`
- Pack the following files and directories into a ZIP file.



requirements.txt - Notepad

File Edit Format View Help

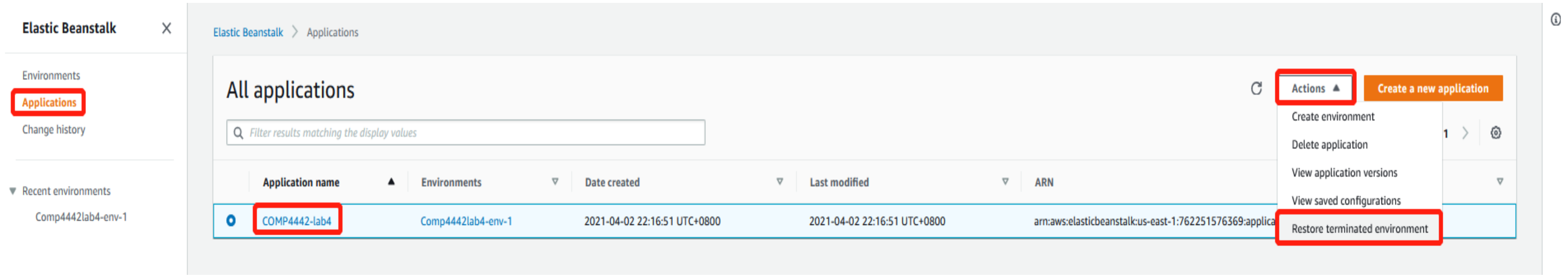
```
click==7.1.2
Flask==1.1.1
itsdangerous==1.1.0
Jinja2==2.11.3
MarkupSafe==1.1.1
mysql-connector==2.2.9
Werkzeug==1.0.1
```



Name	Date modified	Type	Size
✓ static	4/3/2021 8:44 PM	File folder	
✓ templates	4/3/2021 8:44 PM	File folder	
✓ application.py	4/3/2021 9:16 PM	Python File	2 KB
✓ requirements.txt	4/2/2021 10:05 PM	Text Document	1 KB

Deploy Code in AWS Beanstalk

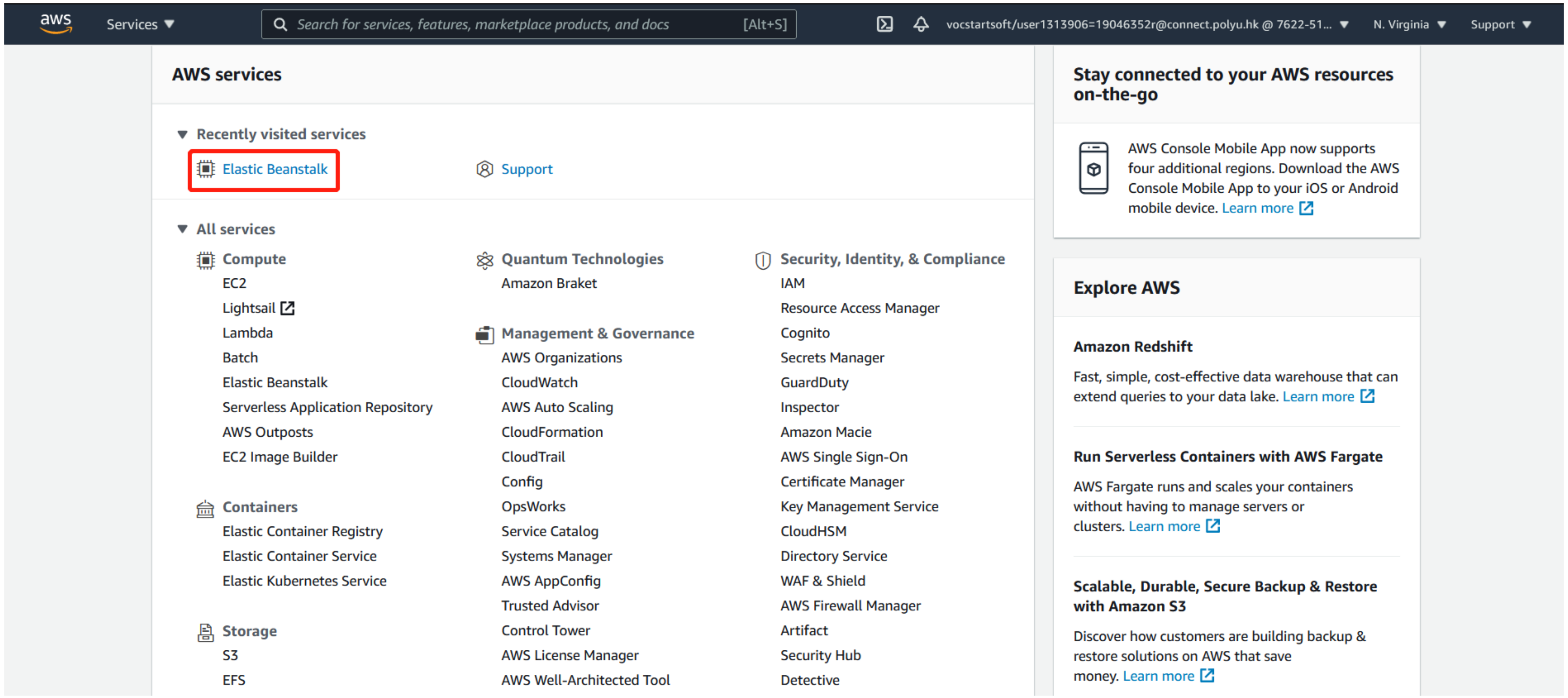
- Restore Beanstalk environment and upload the ZIP file as we do in lab-01.



Application name	Environments	Date created	Last modified	ARN
COMP4442-lab4	Comp4442lab4-env-1	2021-04-02 22:16:51 UTC+0800	2021-04-02 22:16:51 UTC+0800	arn:aws:elasticbeanstalk:us-east-1:762251576369:applica

Stop Elastic Beanstalk

- Access the beanstalk



aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] vocstartsoft/user1313906=19046352r@connect.polyu.hk @ 7622-51... ▼ N. Virginia ▼ Support ▼

AWS services

▼ Recently visited services

Elastic Beanstalk   Support

▼ All services

Compute
EC2
Lightsail
Lambda
Batch
Elastic Beanstalk
Serverless Application Repository
AWS Outposts
EC2 Image Builder

Containers
Elastic Container Registry
Elastic Container Service
Elastic Kubernetes Service

Storage
S3
EFS

Quantum Technologies
Amazon Braket

Management & Governance
AWS Organizations
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
AWS AppConfig
Trusted Advisor
Control Tower
AWS License Manager
AWS Well-Architected Tool

Security, Identity, & Compliance
IAM
Resource Access Manager
Cognito
Secrets Manager
GuardDuty
Inspector
Amazon Macie
AWS Single Sign-On
Certificate Manager
Key Management Service
CloudHSM
Directory Service
WAF & Shield
AWS Firewall Manager
Artifact
Security Hub
Detective

Stay connected to your AWS resources on-the-go

AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

Explore AWS

Amazon Redshift

Fast, simple, cost-effective data warehouse that can extend queries to your data lake. Learn more

Run Serverless Containers with AWS Fargate

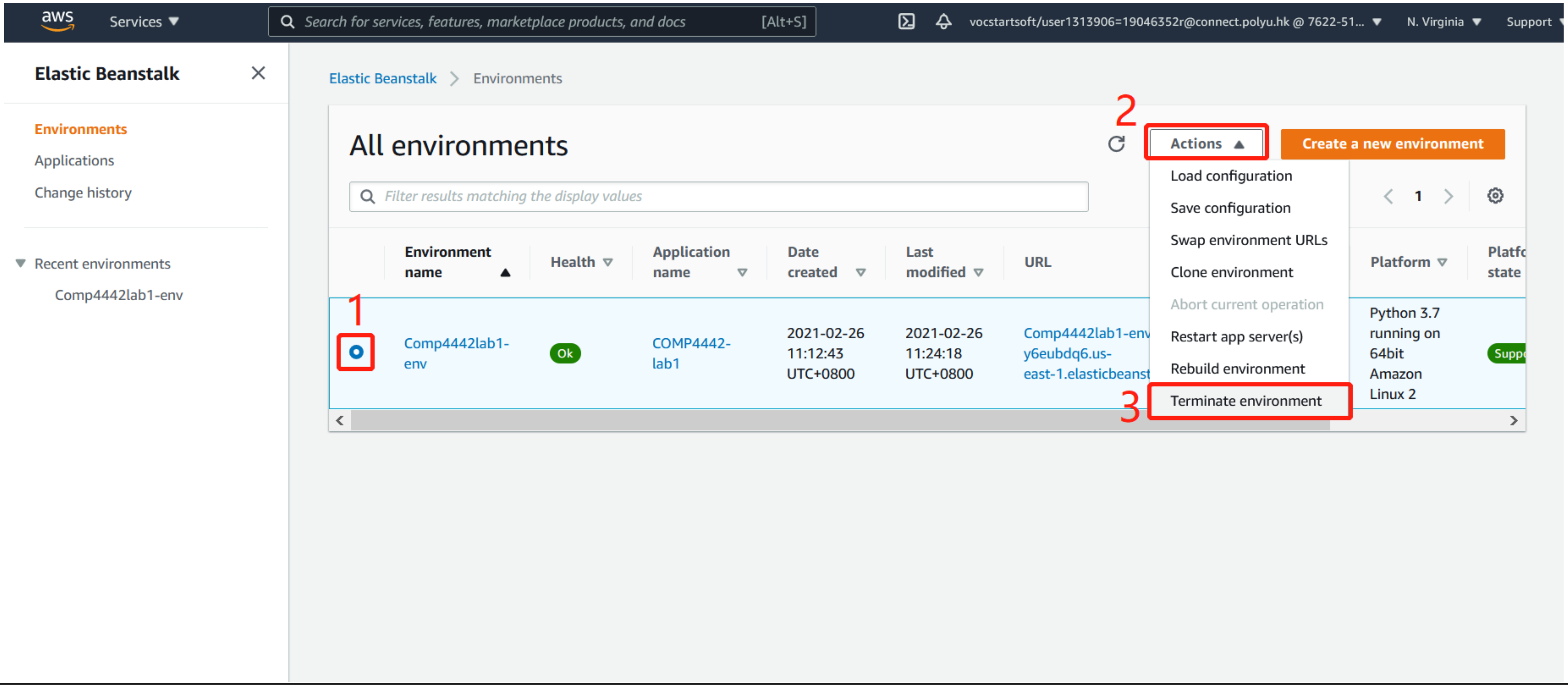
AWS Fargate runs and scales your containers without having to manage servers or clusters. Learn more

Scalable, Durable, Secure Backup & Restore with Amazon S3

Discover how customers are building backup & restore solutions on AWS that save money. Learn more

Stop Elastic Beanstalk

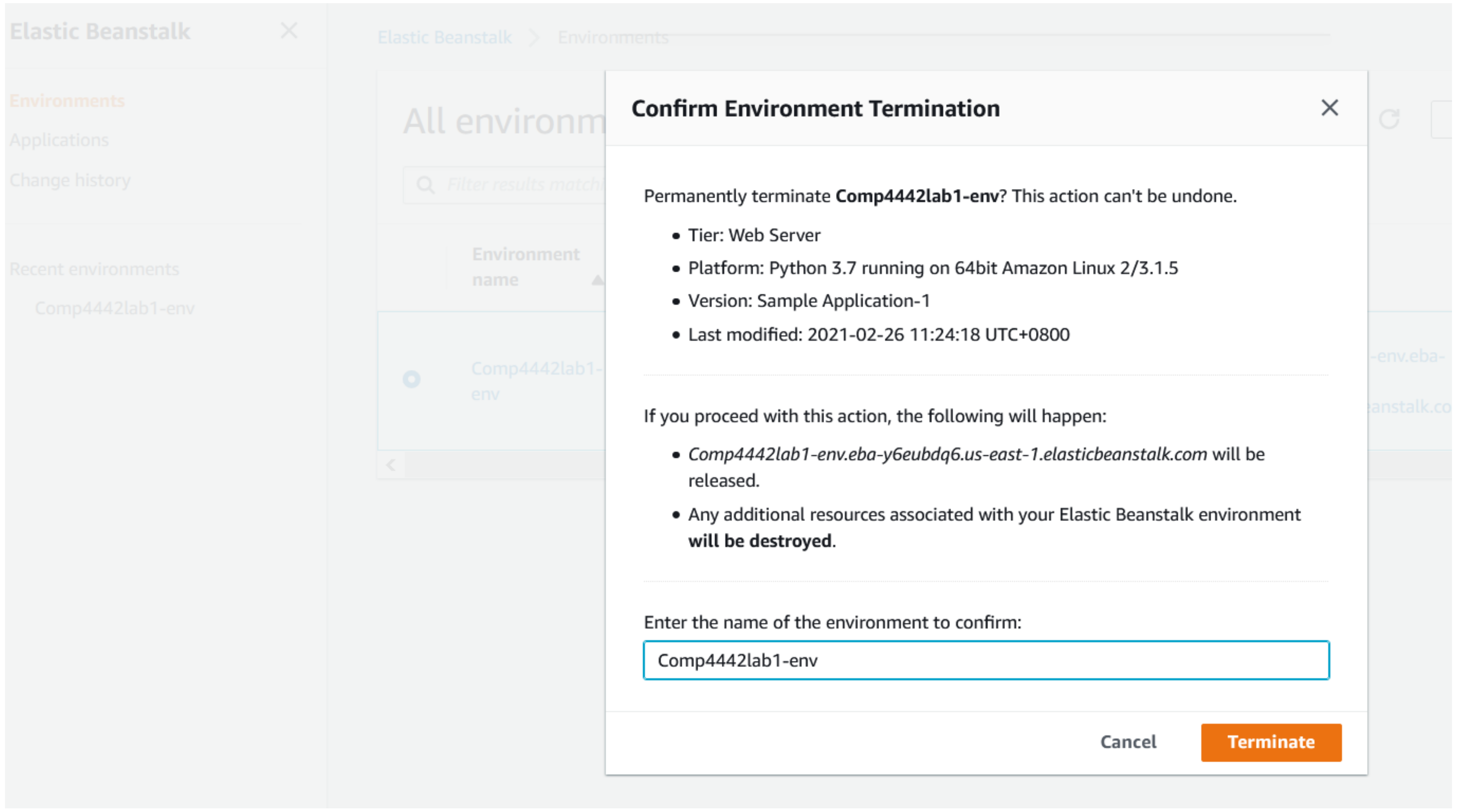
- Terminate the environment



	Environment name	Health	Application name	Date created	Last modified	URL	Platform	Platfo state
●	Comp4442lab1-env	Ok	COMP4442-lab1	2021-02-26 11:12:43 UTC+0800	2021-02-26 11:24:18 UTC+0800	Comp4442lab1-env y6eubdq6.us-east-1.elasticbeanst	Python 3.7 running on 64bit Amazon Linux 2	Supp

Stop Elastic Beanstalk

- Confirm your operation



Confirm Environment Termination

Permanently terminate **Comp4442lab1-env**? This action can't be undone.

- Tier: Web Server
- Platform: Python 3.7 running on 64bit Amazon Linux 2/3.1.5
- Version: Sample Application-1
- Last modified: 2021-02-26 11:24:18 UTC+0800

If you proceed with this action, the following will happen:

- *Comp4442lab1-env.eba-y6eubdq6.us-east-1.elasticbeanstalk.com* will be released.
- Any additional resources associated with your Elastic Beanstalk environment **will be destroyed**.

Enter the name of the environment to confirm:

Comp4442lab1-env

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Q&A