Intro. to Network Programming

HW3

TA 廷翰

Due date

Sunday, May 24, 2020 by 11:55pm

Outline

- Introduction
- Amazon S3
- Details of this homework

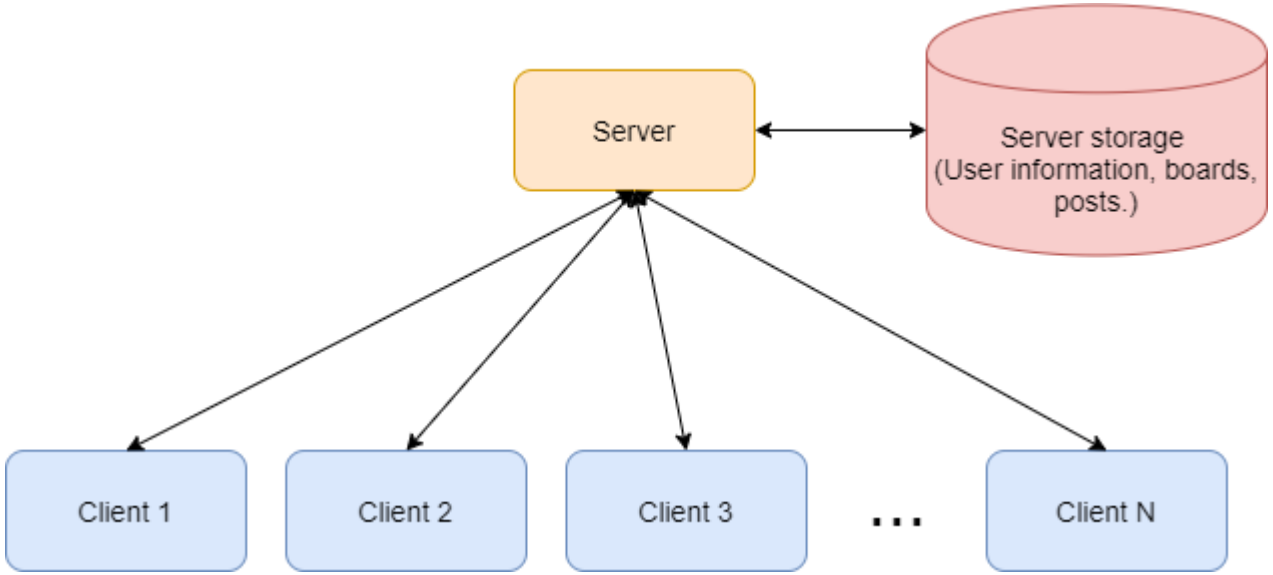
Introduction

- What you have to do in this homework.

1. Write a **client program** to connect to your server.
2. Store **the content of the post on the client-side (Amazon S3)** instead of the server-side.
3. Implement a simple **mail service**.

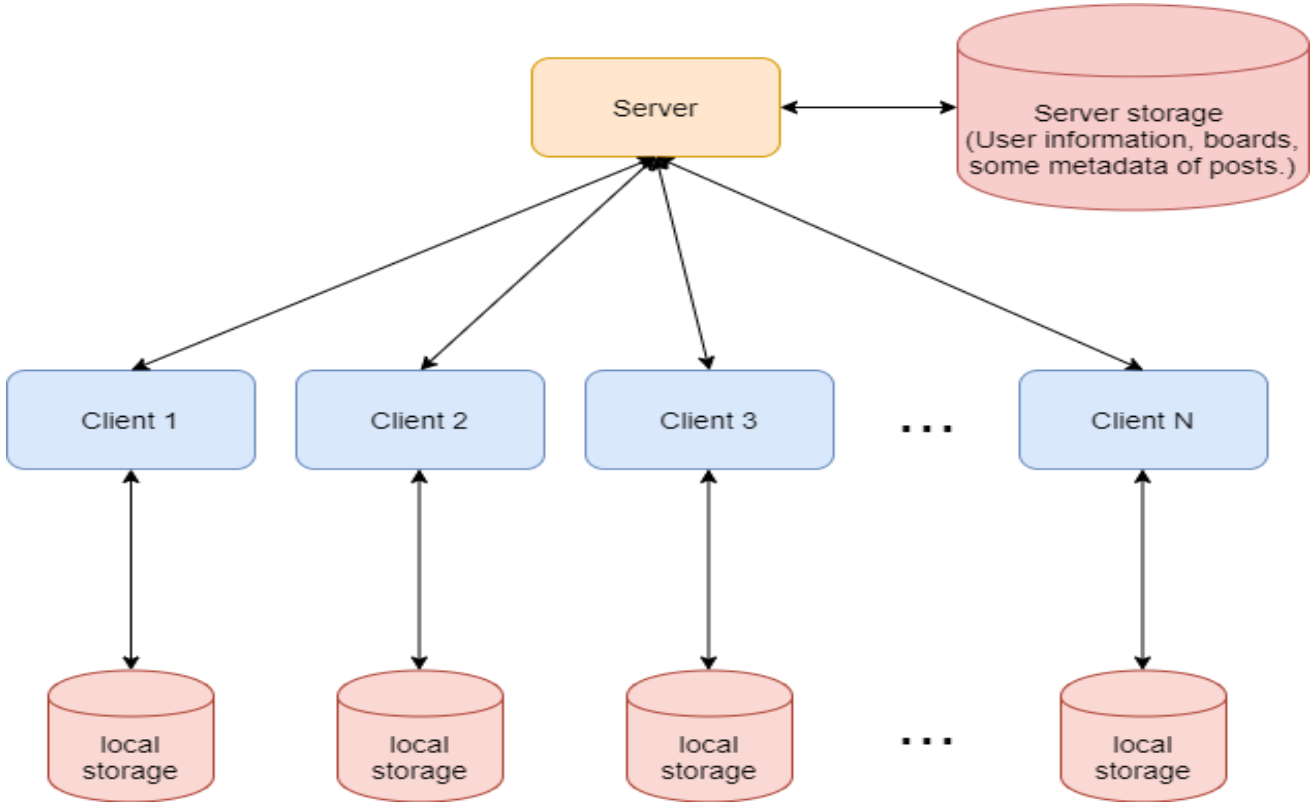
Introduction (cont.)

- System architecture in the previous part. Client program only sends command to the server and gets the corresponding result from the server.



Introduction (cont.)

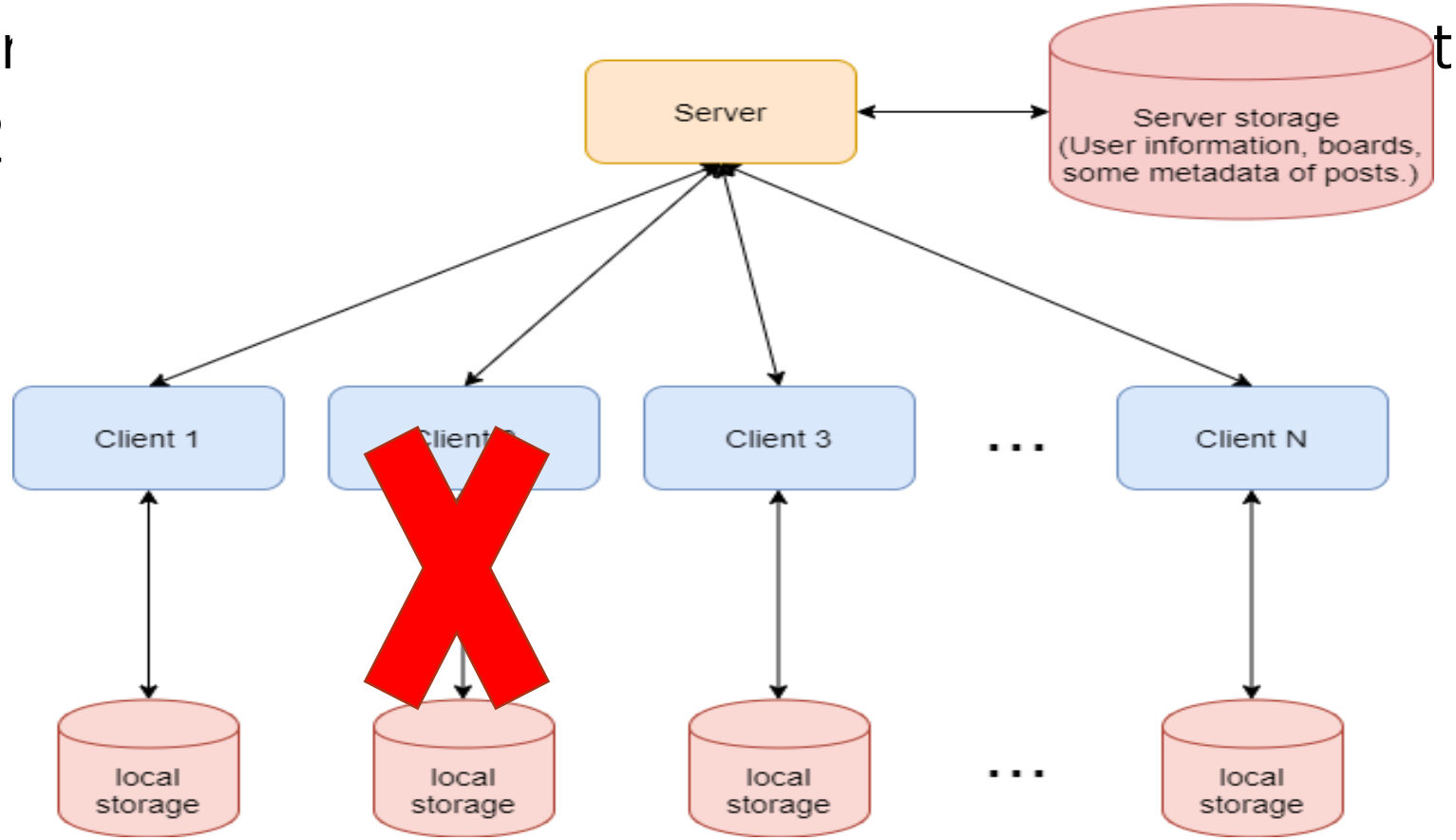
- Change to the new system architecture.



Introduction (cont.)

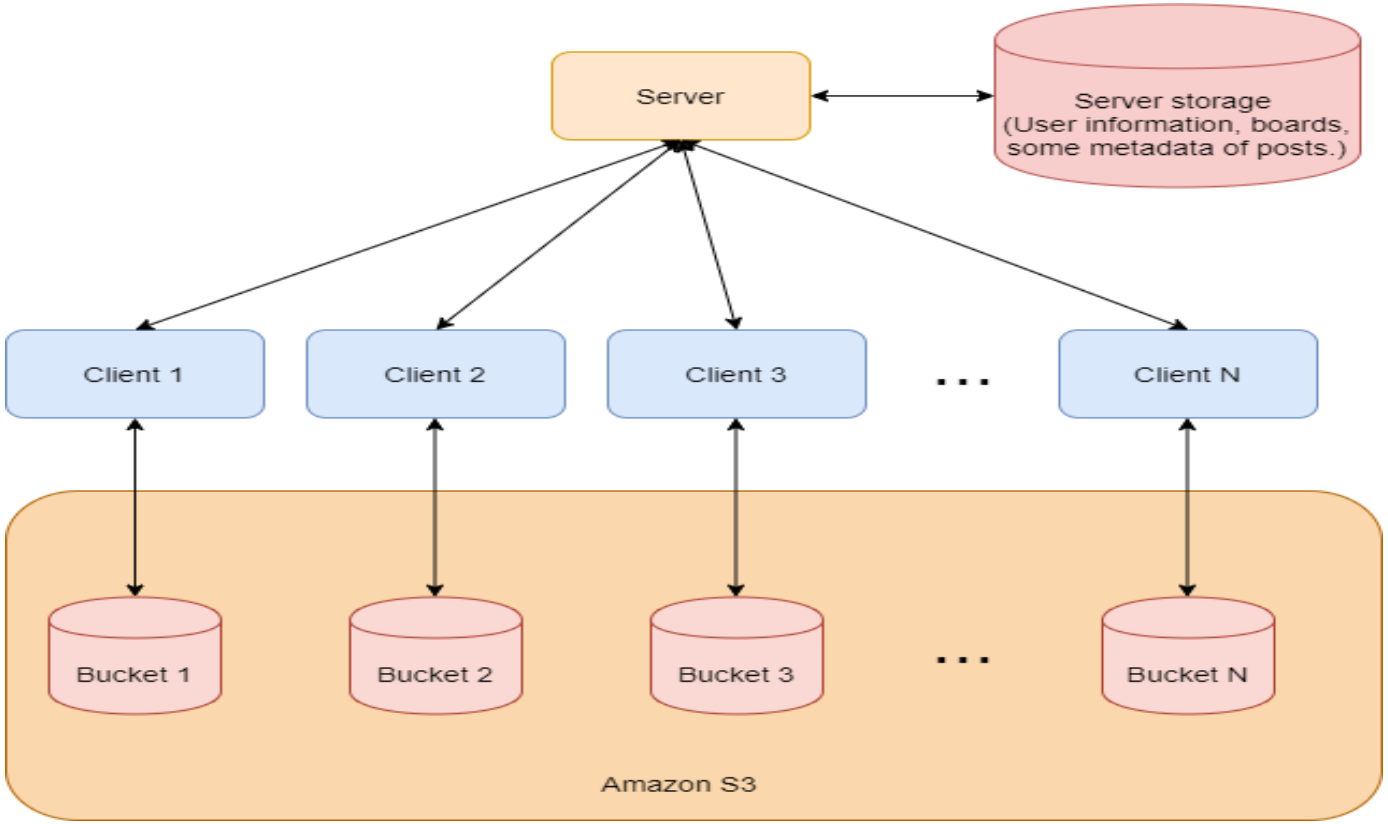
- Proble
  client2

t but



Introduction (cont.)

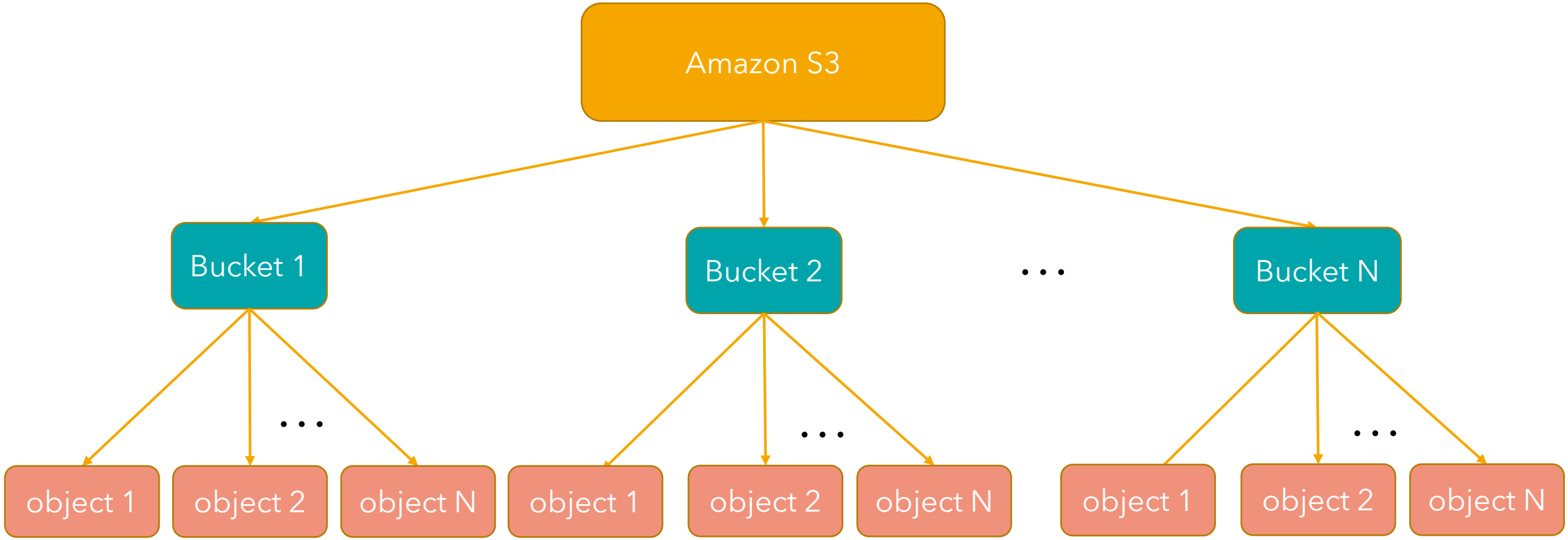
- System architecture we will use in this homework



Amazon S3

- Amazon Simple Storage Service (S3) is storage for the Internet.
- You can use Amazon S3 to store and retrieve any amount of data, at any time, from anywhere on the web.
- The Amazon S3 data model is a **flat** structure. There is no hierarchy of subbuckets or subfolders.

Amazon S3 - Overview



Amazon S3 - Bucket

- A bucket is a container for objects stored in Amazon S3.
- By default, you can create up to 100 buckets in each of your AWS accounts
- If you want to store data in Amazon S3, you have to create a bucket with a **unique** name first.
- Bucket names must be **unique** across all existing bucket names in Amazon S3.
- Bucket names must be **at least 3** and **no more than 63 characters long.**
- Bucket names must **not** contain **uppercase** characters or **underscores.**
- Bucket names must **start with a lowercase letter or number**.

Amazon S3 - Object

- Objects are the fundamental entities stored in Amazon S3.
- Objects consist of object data and metadata.
- Each object within a bucket has its unique key (name of the object).
- Size of each object: 0 Byte – 5 TB

Amazon S3 – Object Key

- Unique identifier within a bucket. If you upload the same key name object without versioning-enabled, it will overwrite the original one.
- The following character sets are generally safe for use in key names.

Alphanumeric characters	• 0-9 • a-z • A-Z
Special characters	• ! • - • _ • . • * • ' • (•)

Amazon S3 – Bucket Policy

- Bucket policy is used to manage access permissions of all resources within a bucket to other AWS accounts or AWS Identity and Access Management (IAM) users.

Amazon S3 – Bucket Policy

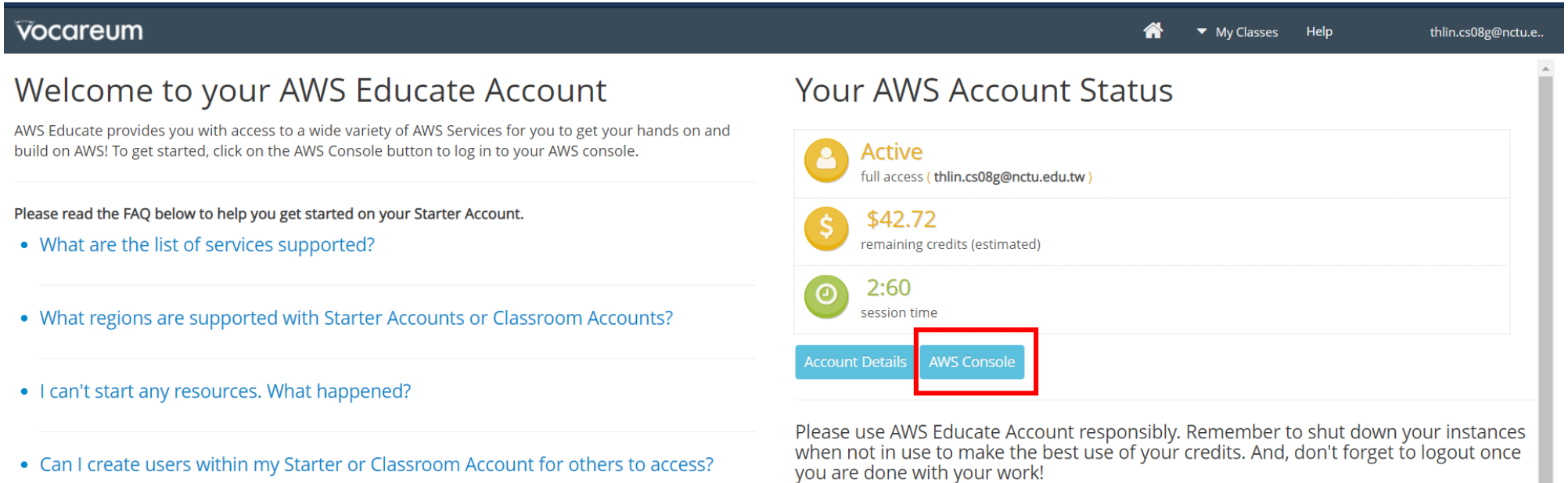
- Bucket Policy example – json format

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"PublicRead",
      “Effect”:“Allow",
      "Principal": "*",
      "Action":["s3:GetObject"],
      "Resource":["arn:aws:s3:::examplebucket/*"]
    }
  ]
}
```

Amazon S3 – Amazon S3 Console

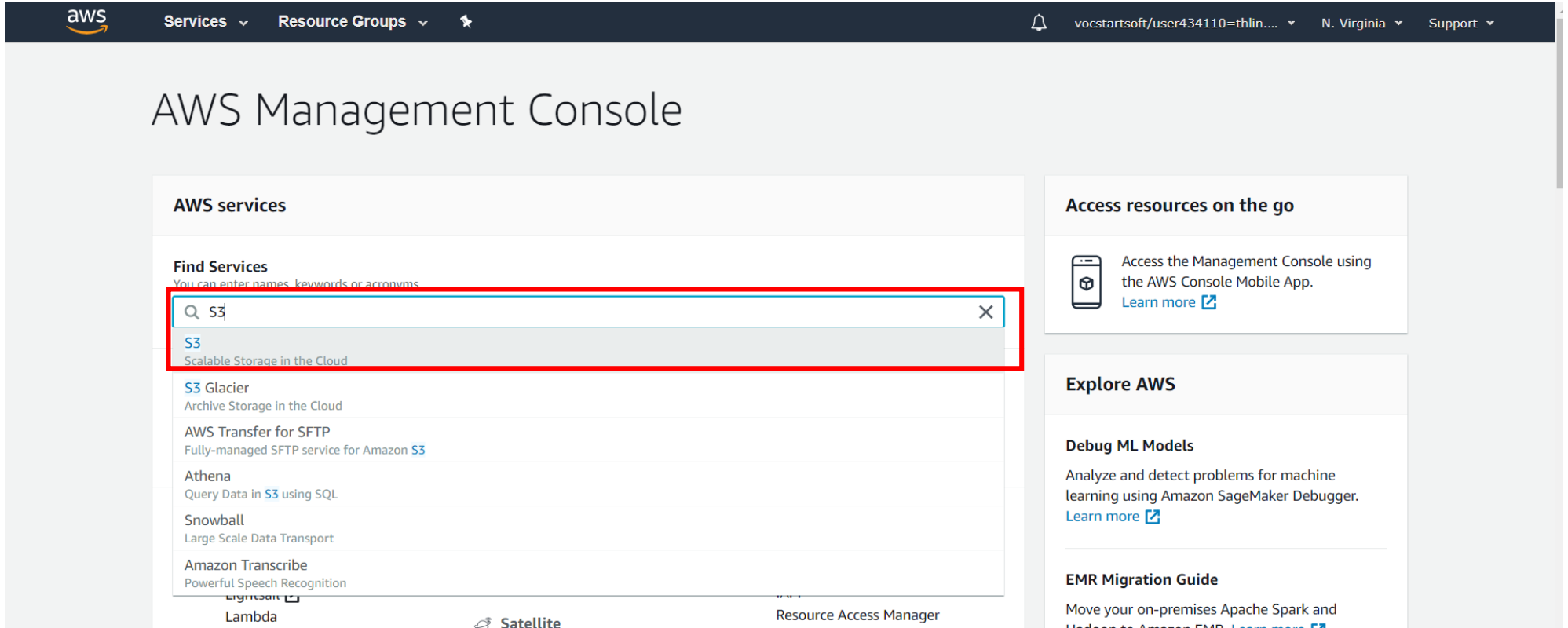
- Login your AWS Educate -> Go to Intro. to Network Programming classroom, then you will see the page below.

Click AWS Console



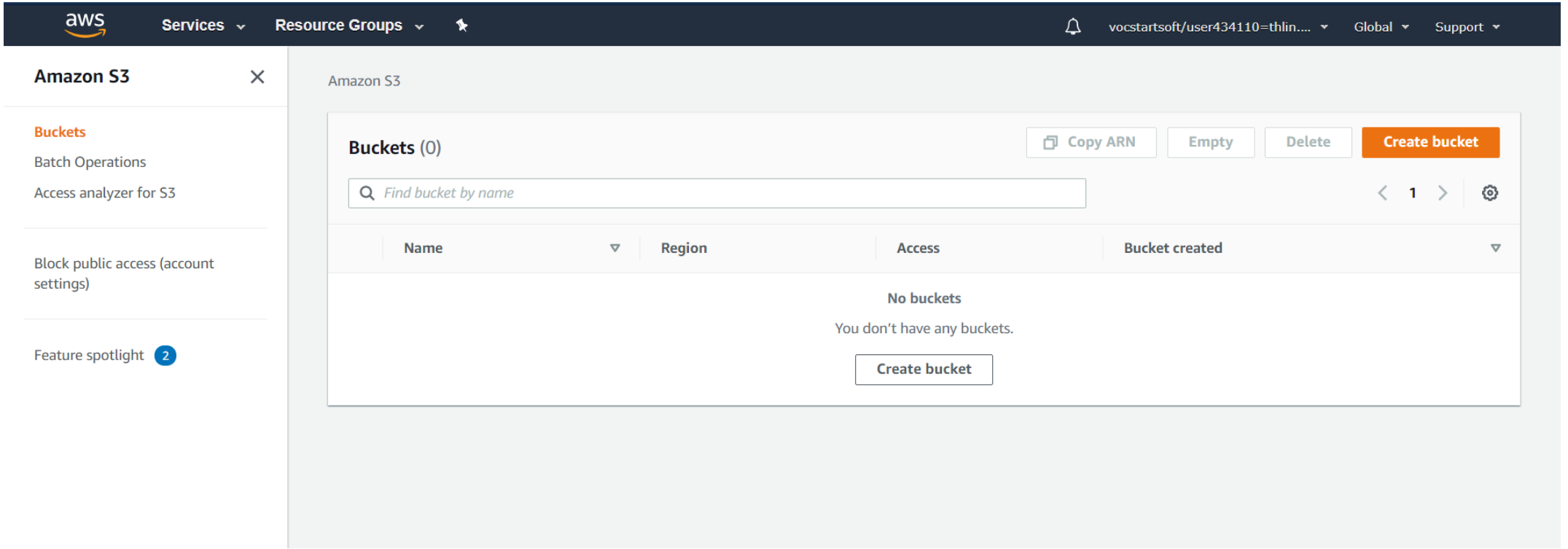
Amazon S3 – Amazon S3 Console

Select S3 Service



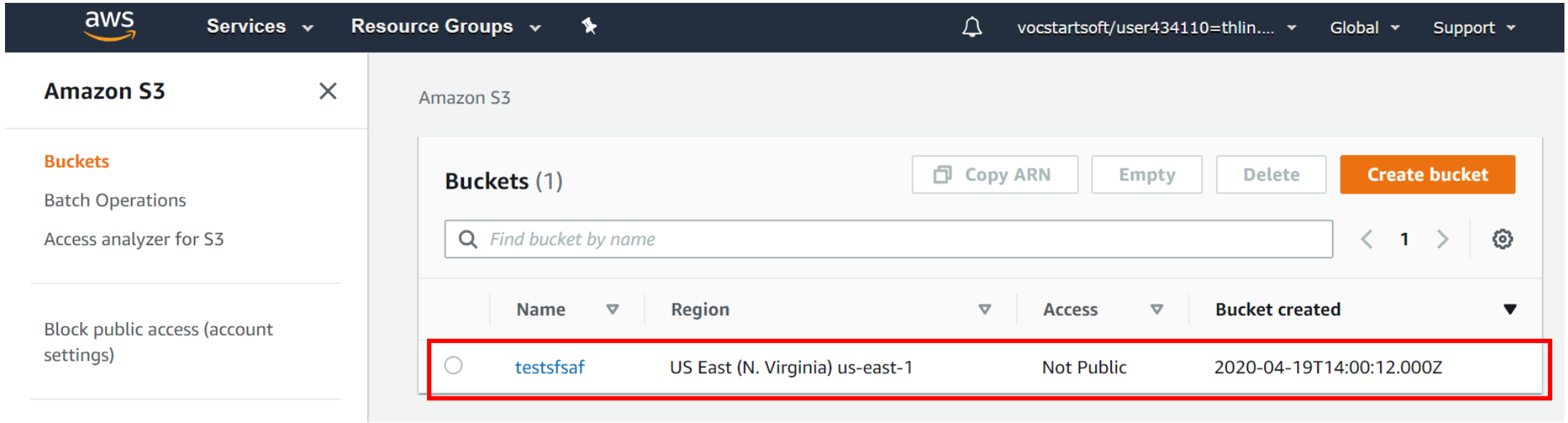
Amazon S3 – Amazon S3 Console

You can manipulate your Amazon S3 and see its state here.



Amazon S3 – Amazon S3 Console

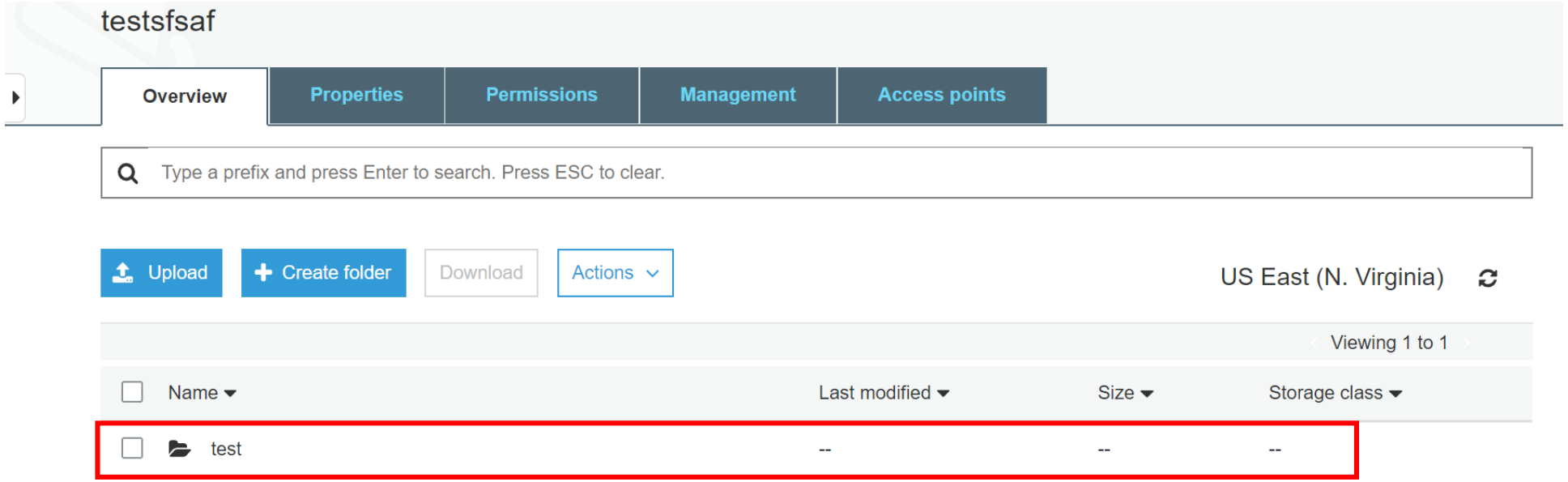
- Example: create a bucket (testsfsaf)



Amazon S3 – Amazon S3 Console

Example: create folder

Note: This is not a real folder. It's just an object with key "test/". Amazon S3 console infer logical hierarchy using key name prefixes and delimiters.



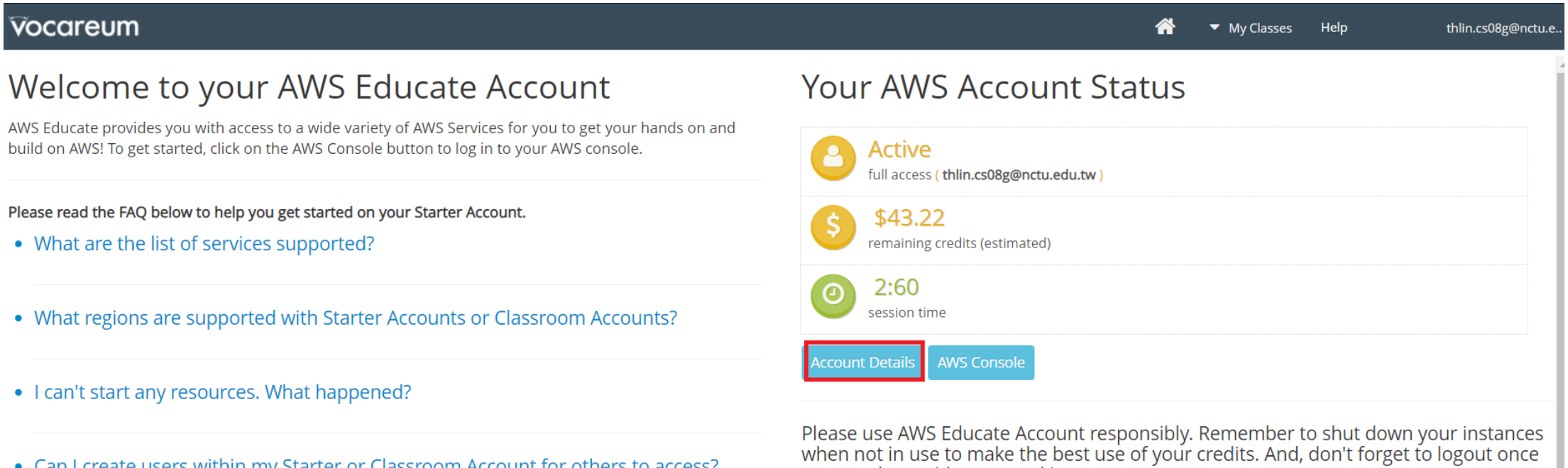
Amazon S3 – Amazon S3 API

- However, in this homework, we will manipulate Amazon S3 using AWS CLI or AWS SDK in the **client program**.
- If you want to use AWS SDK to make Amazon S3 API calls, you have to provide your AWS credential first.
- **How to set up authentication credential**:
  - Create a credential file at **~/.aws/credentials**. The content of this file is described as follows:

```
[default]
aws_access_key_id=<your access key>
aws_secret_access_key=<your secret access key>
aws_session_token=<your session token>
```

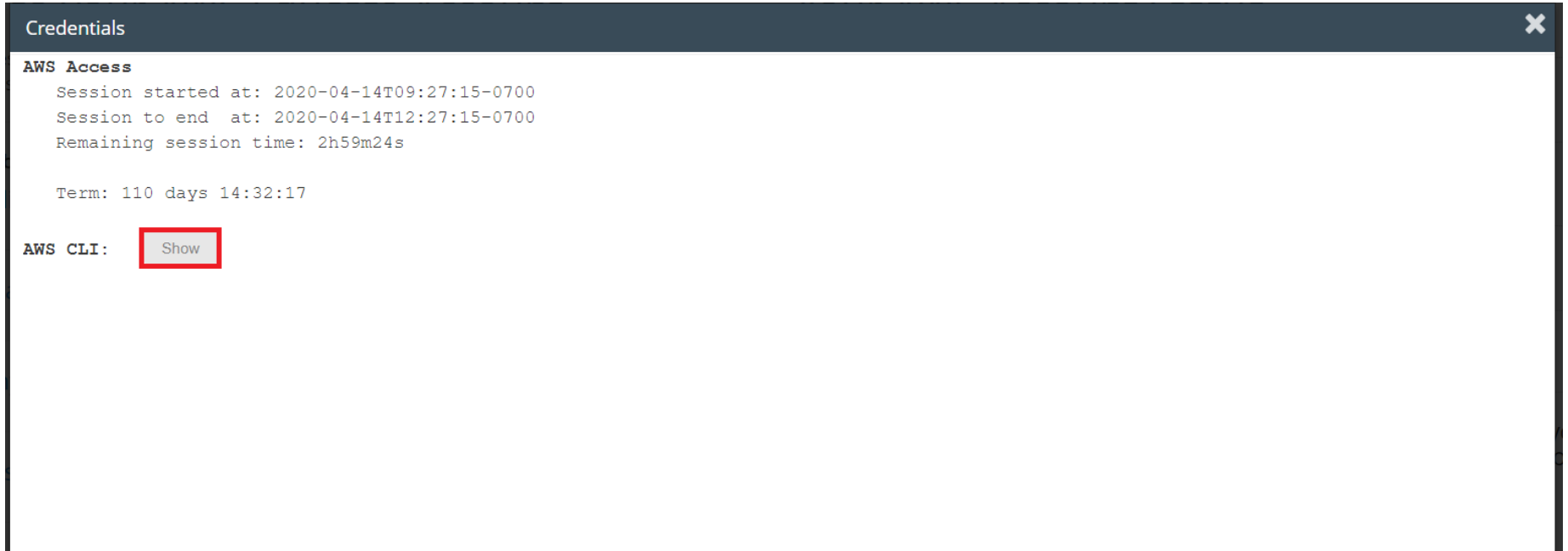
Amazon S3 – Amazon S3 API

- You can get these key from your AWS Educate account. Log in your account and go to Intro. to Network Programming classroom. Then, you will see the following page.



Amazon S3 – Amazon S3 API

- Click Show and copy those keys into ~/.aws/credentials
- **The credential we use here is temporary, so you have to copy and paste again when the credential expiration.**



Amazon S3 – Amazon S3 API

- Actions you might use:
  - Create bucket
  - Delete bucket
  - Upload object
  - Get object
  - Delete object

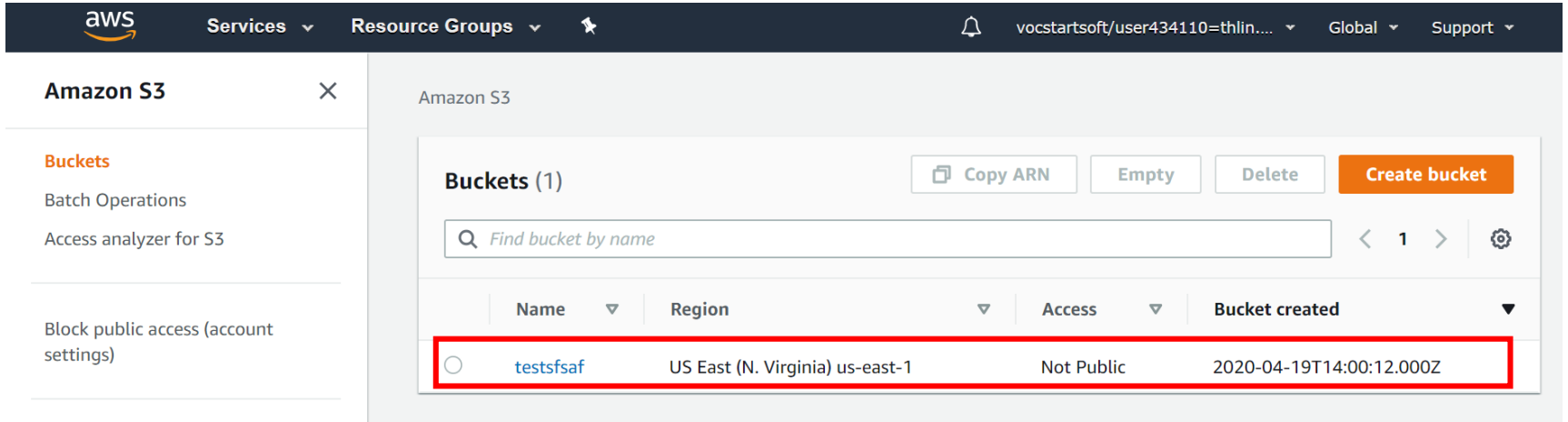
Amazon S3 – Amazon S3 API

- Create bucket

```
import boto3

s3 = boto3.resource('s3')
s3.create_bucket(Bucket='testsfsaf')
```

Amazon S3 Console



Amazon S3 – Amazon S3 API

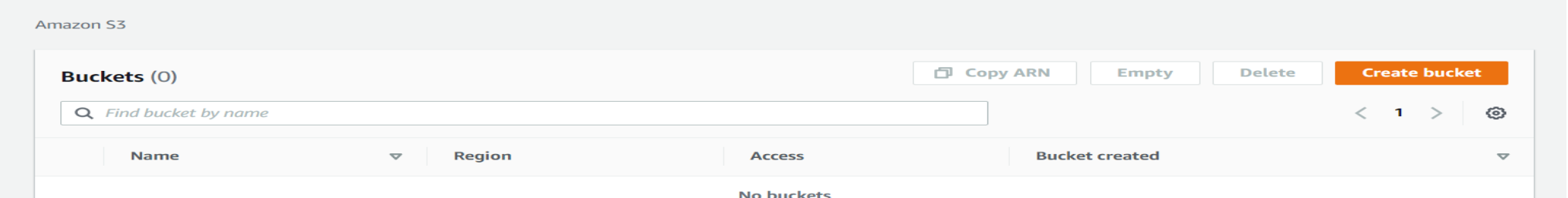
- Delete bucket

```
import boto3

s3 = boto3.resource('s3')
target_bucket = s3.Bucket('testsfsaf')
target_bucket.delete()
```

- Note: The bucket must be **empty** when deleting.

Amazon S3 Console



Amazon S3 – Amazon S3 API

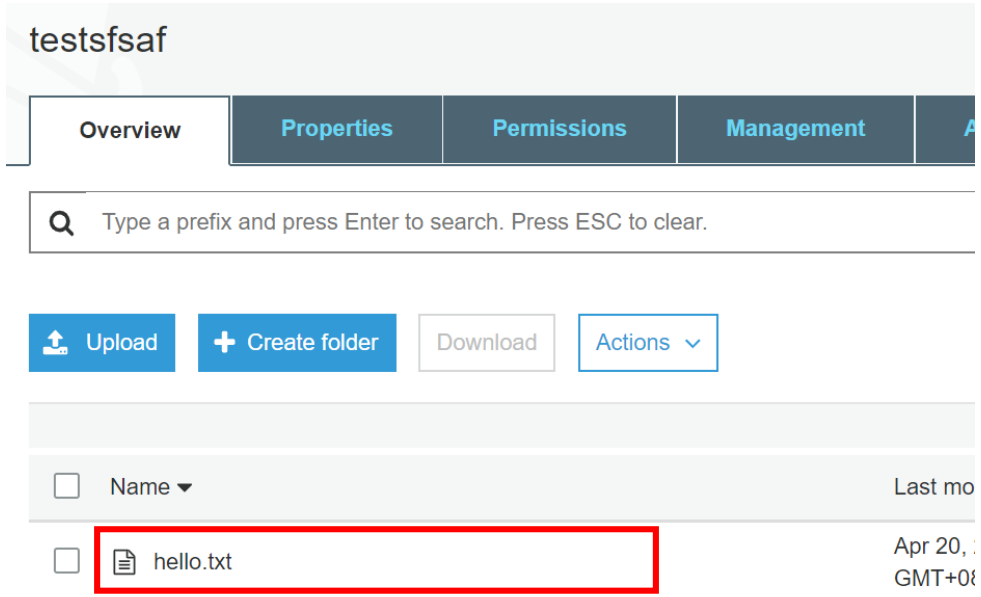
- Upload object

```
import boto3

s3 = boto3.resource('s3')
target_bucket = s3.Bucket('testsfsaf')
target_bucket.upload_file('./tmp/hello.txt', 'hello.txt')
```

- First argument of upload_file is the file you want to upload.
- Second argument is object key.

Amazon S3 Console



Amazon S3 – Amazon S3 API

- Get object

```
import boto3

s3 = boto3.resource('s3')
target_bucket = s3.Bucket('testsfsaf')
target_object = target_bucket.Object('hello.txt')
object_content = target_object.get()['Body'].read().decode()    #get the content of hello.txt
```

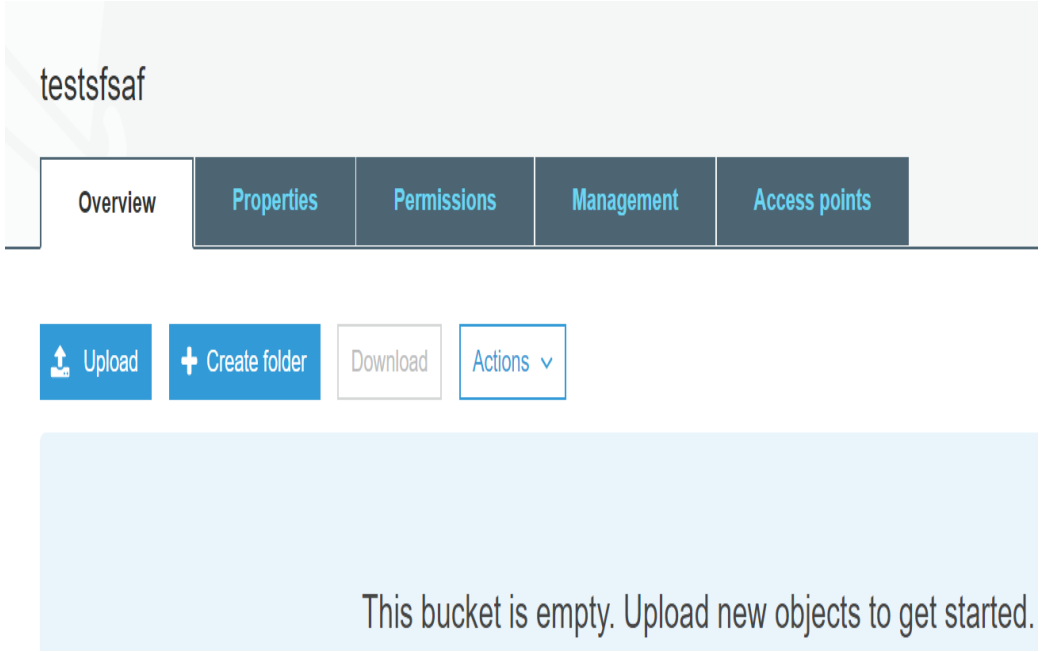
Amazon S3 – Amazon S3 API

- Delete object

```
import boto3

s3 = boto3.resource('s3')
target_bucket = s3.Bucket('testsfsaf')
target_object = target_bucket.Object('hello.txt')
target_object.delete()
```

Amazon S3 Console



Details of this homework

1. Your server and client program must be able to handle all commands in the previous part.
2. For some commands such as **whoami**, **exit**, **logout**, **create-board**, **list-board ##<key>** and **list-post <board-name> ##<key>**, your client program only sends the command to the server and gets the corresponding result from the server.
3. However, there are some commands that your **client program** will interact with **Amazon S3**.
4. Also, there are some **new commands** you have to implement for **simple mail service**.

We will discuss the details of 3. and 4. by reading the requirements and scenario parts of the spec.