UNIVERSITY PARTNER
UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

Distributed And Cloud Computing

(5CS022)

Topic: Gadget Inventory Hosting Report

Student Id: 2330752

Student Name: Niroj Thapa

Group: L5CG12

Lecturer: Mr. Dipson Shrestha

Tutor: Ms. Jenny Rajak

Submitted on: 05/20/2024.

Acknowledgement

To begin with, I would like first and foremost thank our module leader, **Mr. Dipson Shrestha** for his ultimate help with our project positively and advising insightful feedback on multiple occasions that involved laughing, listening, and bonding. His care and unselfishness to the subject as well as primer guidance to me affected my comprehension meaningfully. Besides, they added the spur to my work.

In addition, the gratitude I feel for my teacher, **Ms. Jenny Rajak**, will be held paramount for her unfathomably patience and guidance. She is the one who gave me a useful feedback, patient help, and timely suggestions, which all played a vital role in the report´s design. The passion and zeal she has in her work to uplift and motivate the students brings me look up to her with admiration.

Contents

1. AWS Elastic Beanstalk, AWS DynamoDB, AWS S3 and AWS CloudFront Services

1.1 AWS Elastic Beanstalk

An application hosting platform called Elastic Beanstalk focuses on the fully managed web app deployment process and makes it convenient for scaling operations. We can save ourselves with curl the line of program - instead of running, managing it by implementing Elastic Beanstalk. Virtual instances are managed by cloud platforms with auto-scaling, load balancing, capacity provisioning and application health monitoring. Elastic Beanstalk stands as the best platform for hosting applications designed to have a wide number of services supported by other AWS services since it blends seamlessly (Anon., n.d.).

1.2 AWS DynamoDB

One can apply Elastic Beanstalk too service such as DynamoDB that is a NoSQL database which is used to store and retrieve application data. As it is designed for providing a high performing, scalable and low latency option, it is easy to scale it down or up and for applications with erratic or fluctuating traffic patterns. Elastic Beanstalk-deployed applications are also supported by DynamoDB, which contains security features that are integrated, availability features that are always in memory and fast, and the backup and restore, ensuring the safety of the data.

1.3 AWS S3

For our web applications running on Elastic Beanstalk, we can use S3 (Simple Storage Service) as a cloud object storage service to store and serve static content like photos, CSS and JavaScript files. S3 maintains reliable content delivery to users by employing the most reliable and scalable object storage. Access control, versioning, and encryption are just a few of the feature that guarantee that it is a flexible option that is also safe for serving and storing static assets.

1.4 AWS CloudFront Services

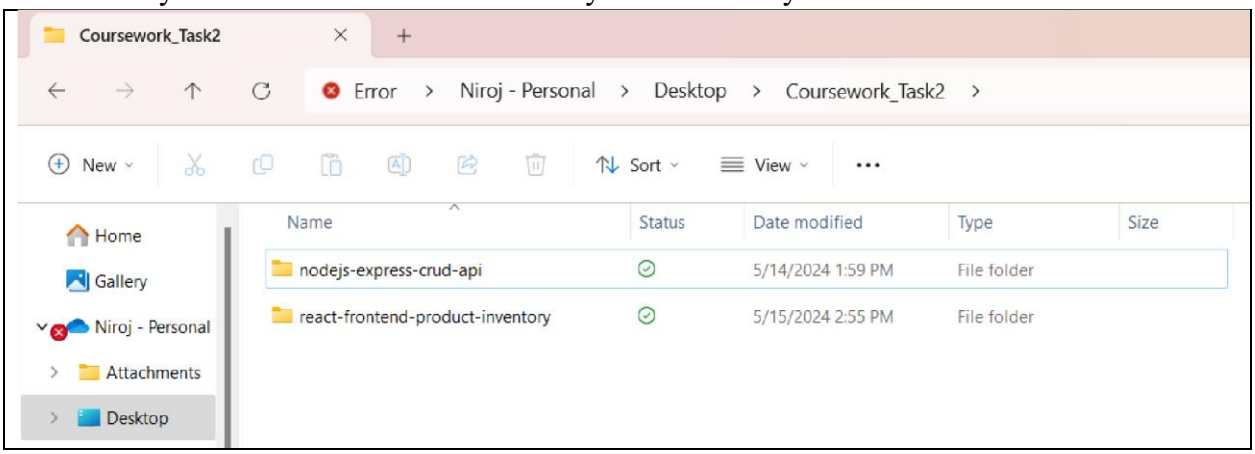
Worldwide content distribution and caching of our static content can be accomplished by leveraging size, which is offering content delivery network. By caching of our content at the edge locations of the world system our customers anywhere will benefit from shorter waiting time and improved load time. On top of that, it provides college and university students with the highly advanced features like SSL/TLS encryption and DDoS protection for content delivery that is safe and reliable.

The combinate of these services helps us to create and host high performance, scalable, and available to the web application while at the same time shifting infrastructure management tasks to AWS (Anon., n.d.).

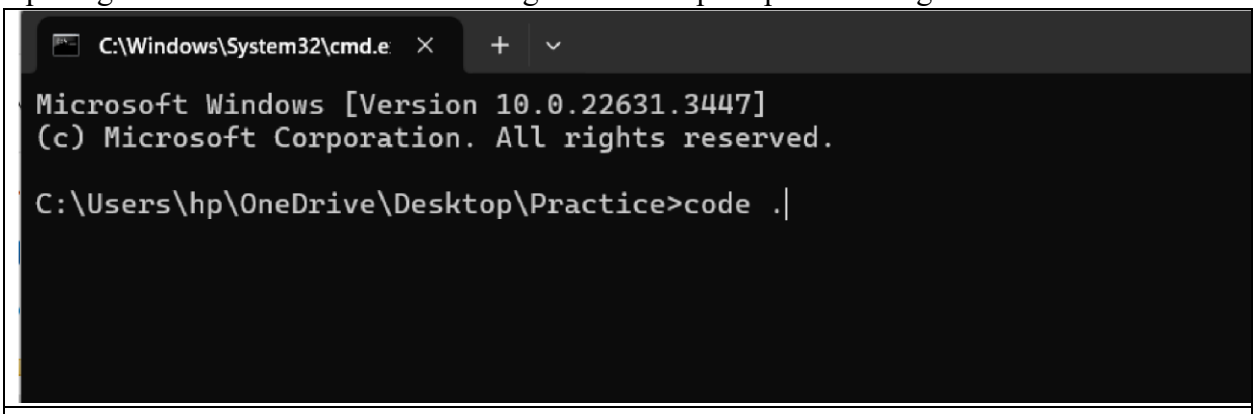
2. Hosting Process:

2.1 Part 1: Files and Code

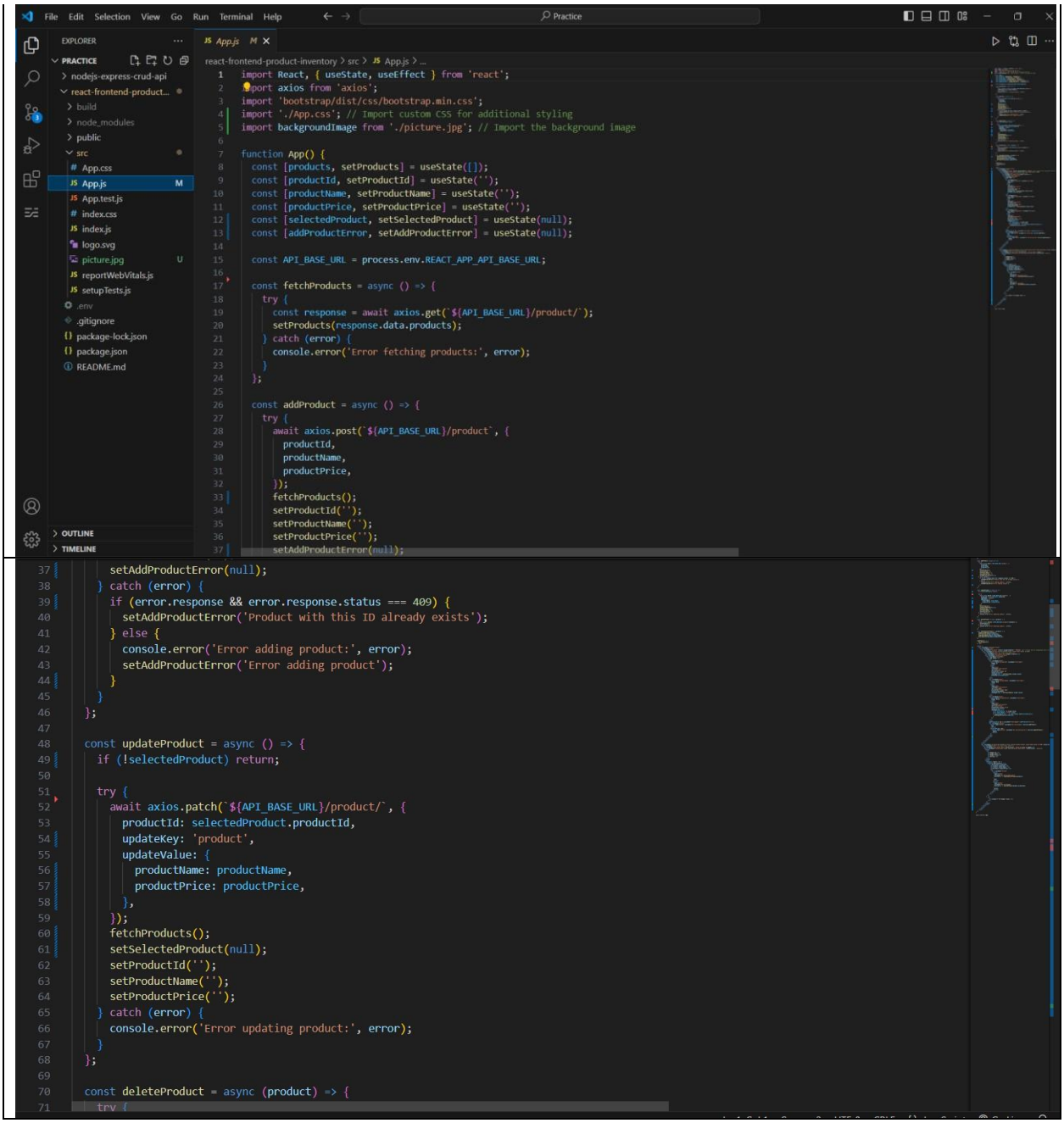
These are my two folders in the local directory with necessary files.

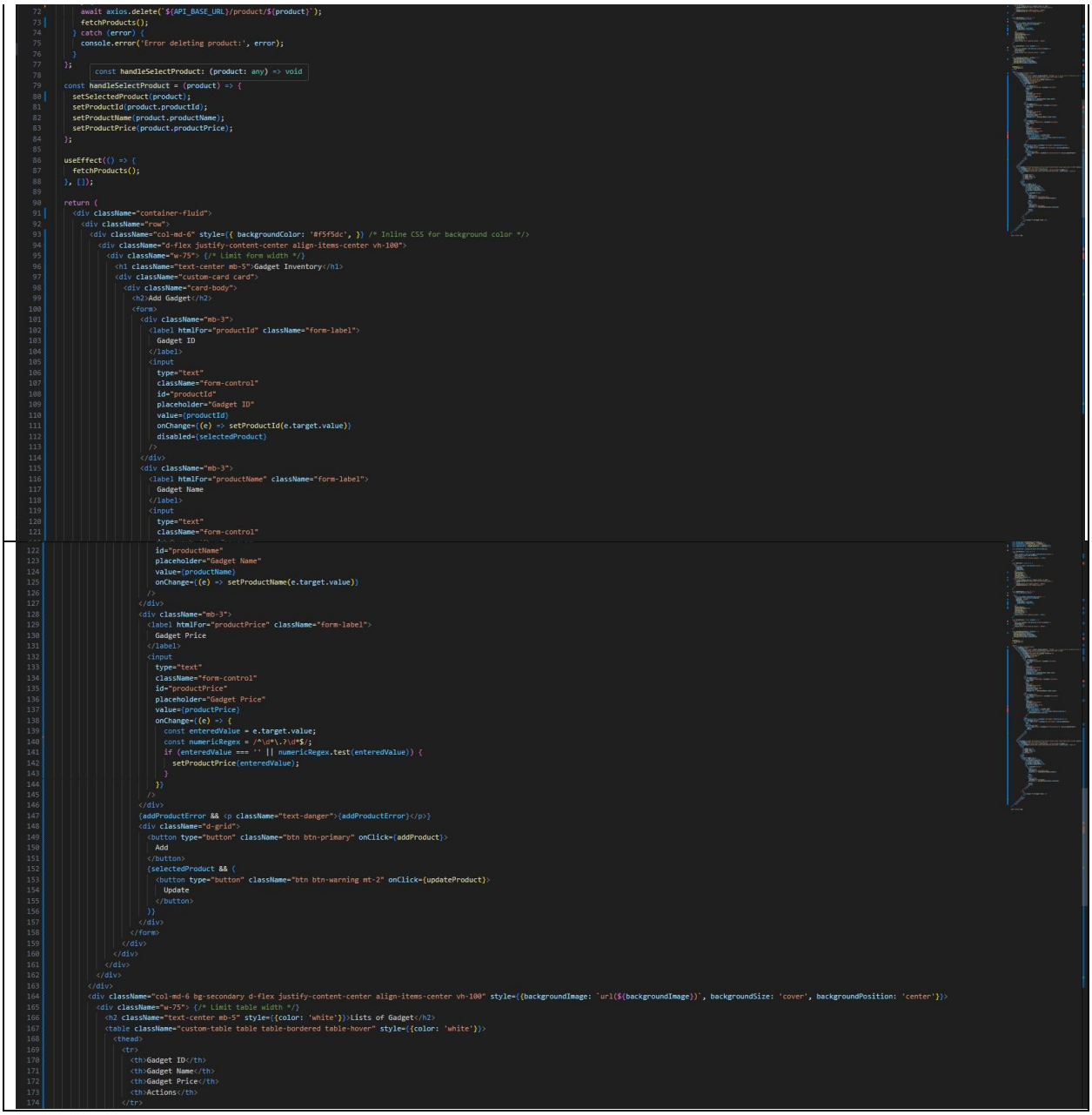


Opening It in a visual Studio Code through command prompt containing all code.



Code:


File Edit Selection View Go Run Terminal Help
Practice
EXPLORER
PRACTICE
nodejs-express-crud-api
react-frontend-product...
build
node_modules
public
src
App.css
App.js
App.test.js
index.css
index.js
logo.svg
picture.jpg
reportWebVitals.js
setupTests.js
.env
.gitignore
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
App.js M
react-frontend-product-inventory > src > App.js > ...
import React, { useState, useEffect } from 'react';
import axios from 'axios';
import 'bootstrap/dist/css/bootstrap.min.css';
import './App.css'; // Import custom CSS for additional styling
import backgroundImage from './picture.jpg'; // Import the background image
function App() {
const [products, setProducts] = useState([]);
const [productId, setProductId] = useState('');
const [productName, setProductName] = useState('');
const [productPrice, setProductPrice] = useState('');
const [selectedProduct, setSelectedProduct] = useState(null);
const [addProductError, setAddProductError] = useState(null);
const API_BASE_URL = process.env.REACT_APP_API_BASE_URL;
const fetchProducts = async () => {
try {
const response = await axios.get(`${API_BASE_URL}/product/`);
setProducts(response.data.products);
} catch (error) {
console.error('Error fetching products:', error);
}
};
const addProduct = async () => {
try {
await axios.post(`${API_BASE_URL}/product`, {
productId,
productName,
productPrice,
});
fetchProducts();
setProductId('');
setProductName('');
setProductPrice('');
setAddProductError(null);
} catch (error) {
if (error.response && error.response.status === 409) {
setAddProductError('Product with this ID already exists');
} else {
console.error('Error adding product:', error);
setAddProductError('Error adding product');
}
}
};
const updateProduct = async () => {
if (!selectedProduct) return;
try {
await axios.patch(`${API_BASE_URL}/product/`, {
productId: selectedProduct.productId,
updateKey: 'product',
updateValue: {
productName: productName,
productPrice: productPrice,
},
});
fetchProducts();
setSelectedProduct(null);
setProductId('');
setProductName('');
setProductPrice('');
} catch (error) {
console.error('Error updating product:', error);
}
};
const deleteProduct = async (product) => {
try {

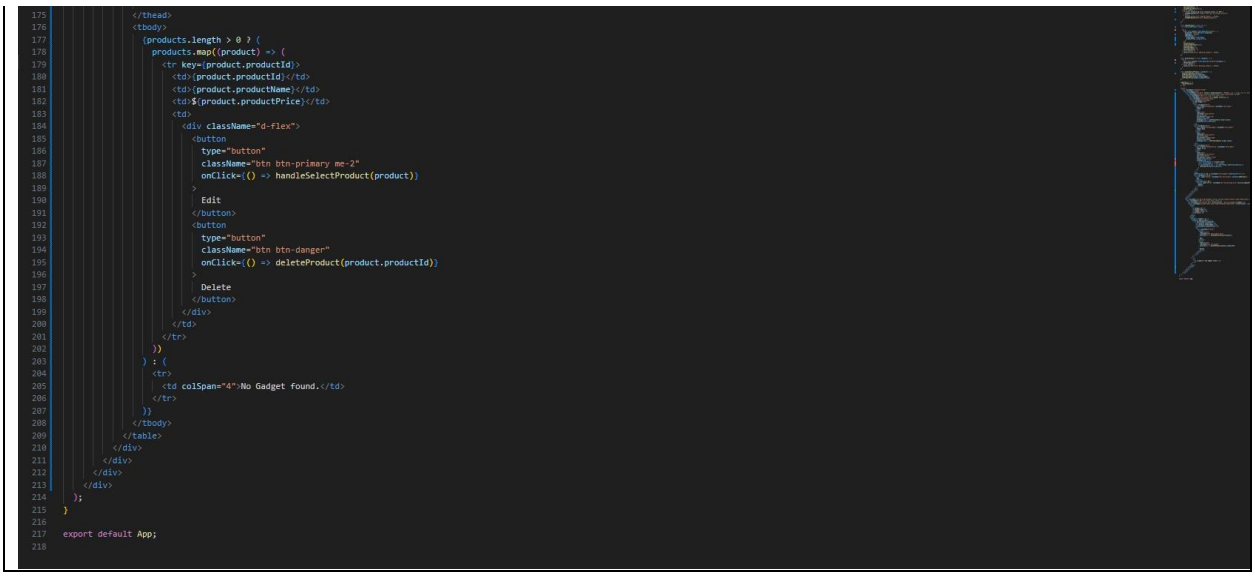


```
    await axios.delete(`${API_BASE_URL}/product/${product}`);
    fetchProducts();
  } catch (error) {
    console.error('Error deleting product:', error);
  }
};
      const handleSelectProduct: (product: any) => void
const handleSelectProduct = (product) => {
  setSelectedProduct(product);
  setProductId(product.productId);
  setProductName(product.productName);
  setProductPrice(product.productPrice);
};

useEffect(() => {
  fetchProducts();
}, []);

return (
  <div className="container-fluid">
    <div className="row">
      <div className="col-md-6" style={{ backgroundColor: '#f5f5dc', }} /* Inline CSS for background color */>
        <div className="d-flex justify-content-center align-items-center vh-100">
          <div className="w-75"> {/* Limit form width */}
            <h1 className="text-center mb-5">Gadget Inventory</h1>
            <div className="custom-card card">
              <div className="card-body">
                <h2>Add Gadget</h2>
                <form>
                  <div className="mb-3">
                    <label htmlFor="productId" className="form-label">
                      Gadget ID
                    </label>
                    <input
                      type="text"
                      className="form-control"
                      id="productId"
                      placeholder="Gadget ID"
                      value={productId}
                      onChange={(e) => setProductId(e.target.value)}
                      disabled={selectedProduct}
                    />
                  </div>
                  <div className="mb-3">
                    <label htmlFor="productName" className="form-label">
                      Gadget Name
                    </label>
                    <input
                      type="text"
                      className="form-control"
                      id="productName"
                      placeholder="Gadget Name"
                      value={productName}
                      onChange={(e) => setProductName(e.target.value)}
                    />
                  </div>
                  <div className="mb-3">
                    <label htmlFor="productPrice" className="form-label">
                      Gadget Price
                    </label>
                    <input
                      type="text"
                      className="form-control"
                      id="productPrice"
                      placeholder="Gadget Price"
                      value={productPrice}
                      onChange={(e) => {
                        const enteredValue = e.target.value;
                        const numericRegex = /^\d*\.?\d*$/;
                        if (enteredValue === '' || numericRegex.test(enteredValue)) {
                          setProductPrice(enteredValue);
                        }
                      }}
                    />
                  </div>
                  {addProductError && <p className="text-danger">{addProductError}</p>}
                  <div className="d-grid">
                    <button type="button" className="btn btn-primary" onClick={addProduct}>
                      Add
                    </button>
                    {selectedProduct && (
                      <button type="button" className="btn btn-warning mt-2" onClick={updateProduct}>
                        Update
                      </button>
                    )}
                  </div>
                </form>
              </div>
            </div>
          </div>
        </div>
      </div>
      <div className="col-md-6 bg-secondary d-flex justify-content-center align-items-center vh-100" style={{backgroundImage: `url(${backgroundImage})`, backgroundSize: 'cover', backgroundPosition: 'center'}}>
        <div className="w-75"> {/* Limit table width */}
          <h2 className="text-center mb-5" style={{color: 'white'}}>Lists of Gadget</h2>
          <table className="custom-table table table-bordered table-hover" style={{color: 'white'}}>
            <thead>
              <tr>
                <th>Gadget ID</th>
                <th>Gadget Name</th>
                <th>Gadget Price</th>
                <th>Actions</th>
              </tr>
```

2.2 Part 2: DynamoDB Table

Before deploying the API, I needed to create a DynamoDB table so, I named it 'product-inventory' in the same AWS region (i.e., us-east-01) where I planned to deploy my application. Here's how I did it:

1. **Access DynamoDB**:
    - I navigated to the AWS Management Console's DynamoDB service.
2. **Initiate Table Creation**:



```
                </thead>
                <tbody>
                  {products.length > 0 ? (
                    products.map((product) => (
                      <tr key={product.productId}>
                        <td>{product.productId}</td>
                        <td>{product.productName}</td>
                        <td>${product.productPrice}</td>
                        <td>
                          <div className="d-flex">
                            <button
                              type="button"
                              className="btn btn-primary me-2"
                              onClick={() => handleSelectProduct(product)}
                            >
                              Edit
                            </button>
                            <button
                              type="button"
                              className="btn btn-danger"
                              onClick={() => deleteProduct(product.productId)}
                            >
                              Delete
                            </button>
                          </div>
                        </td>
                      </tr>
                    ))
                  ) : (
                    <tr>
                      <td colSpan="4">No Gadget found.</td>
                    </tr>
                  )}
                </tbody>
              </table>
            </div>
          </div>
        </div>
      </div>
    );
  }

  export default App;
```

- To begin the process, I selected the "Create table" button.

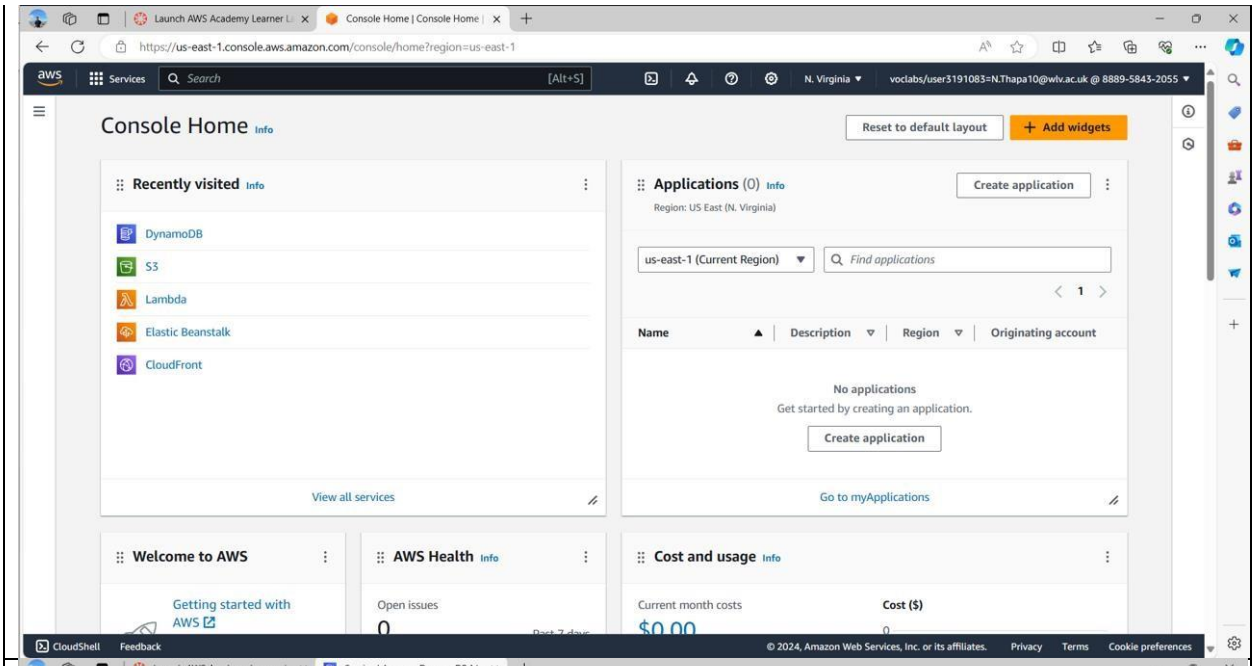
3. **Configure Table Details**:

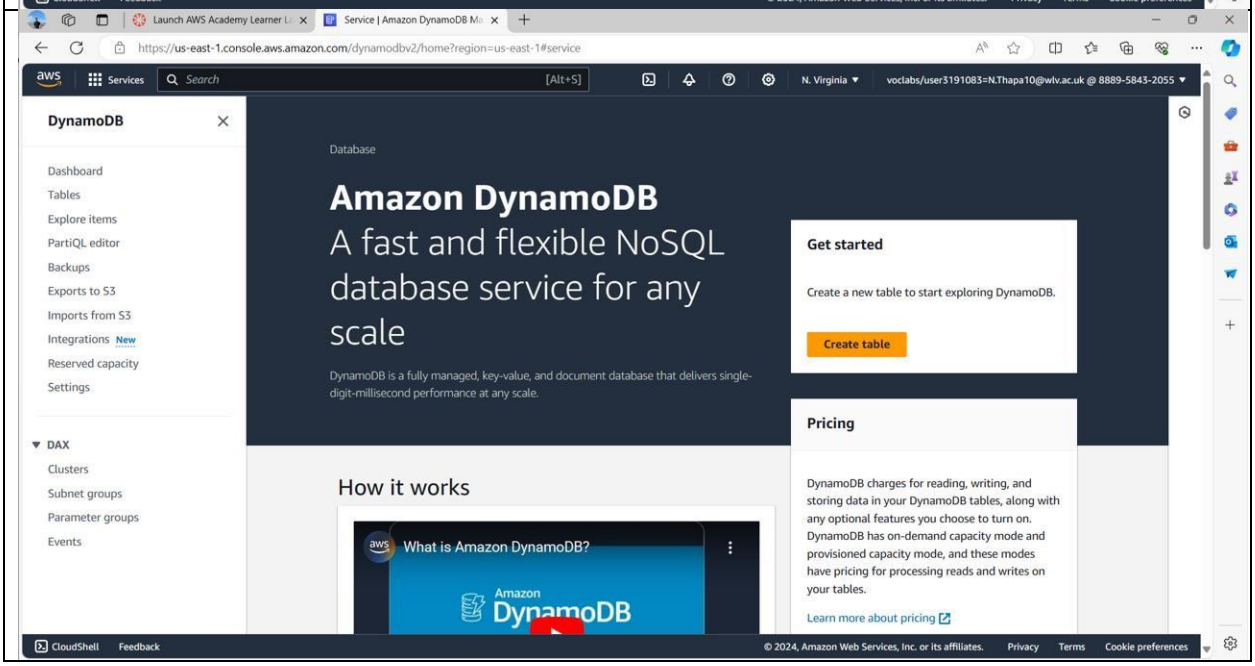
   - I typed product-inventory for the Table name.
   - I typed productId into the Partition key. Since it wasn't necessary, I left the Sort key field empty.

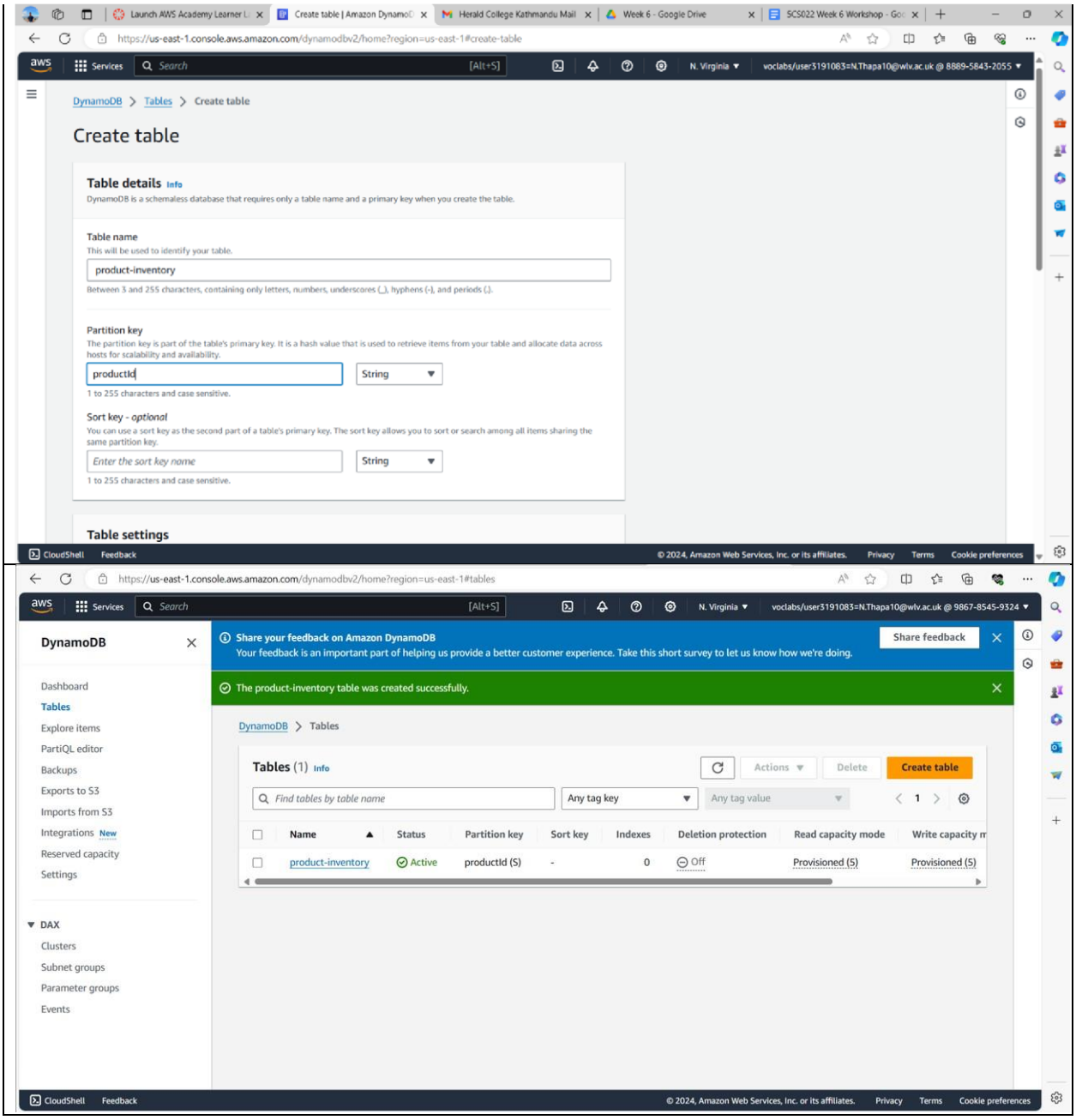
4. **Finalize Table Creation**:

   - I navigated to the bottom of the configuration page by scrolling down.
   - To create the table, I selected the "Create Table" button.

Screenshots By following all above Process:


Launch AWS Academy Learner L
Console Home | Console Home |
https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1
Services
Search
[Alt+S]
N. Virginia
voclabs/user3191083=N.Thapa10@wlv.ac.uk @ 8889-5843-2055
Console Home Info
Reset to default layout
+ Add widgets
Recently visited Info
DynamoDB
S3
Lambda
Elastic Beanstalk
CloudFront
View all services
Applications (0) Info
Region: US East (N. Virginia)
Create application
us-east-1 (Current Region)
Find applications
Name
Description
Region
Originating account
No applications
Get started by creating an application.
Create application
Go to myApplications
Welcome to AWS
Getting started with AWS
AWS Health Info
Open issues
0
Cost and usage Info
Current month costs
Cost ($)
$0.00
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Launch AWS Academy Learner L
Service | Amazon DynamoDB M
https://us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#service
Services
Search
[Alt+S]
N. Virginia
voclabs/user3191083=N.Thapa10@wlv.ac.uk @ 8889-5843-2055
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations New
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
Database
Amazon DynamoDB
A fast and flexible NoSQL database service for any scale
DynamoDB is a fully managed, key-value, and document database that delivers single-digit-millisecond performance at any scale.
Get started
Create a new table to start exploring DynamoDB.
Create table
Pricing
DynamoDB charges for reading, writing, and storing data in your DynamoDB tables, along with any optional features you choose to turn on. DynamoDB has on-demand capacity mode and provisioned capacity mode, and these modes have pricing for processing reads and writes on your tables.
Learn more about pricing
How it works
What is Amazon DynamoDB?
Amazon DynamoDB
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



2.3 Part 3: Web API Server Deployment to AWS Elastic Beanstalk

I took the following actions to deploy the Node.js CRUD API server to AWS Elastic Beanstalk:

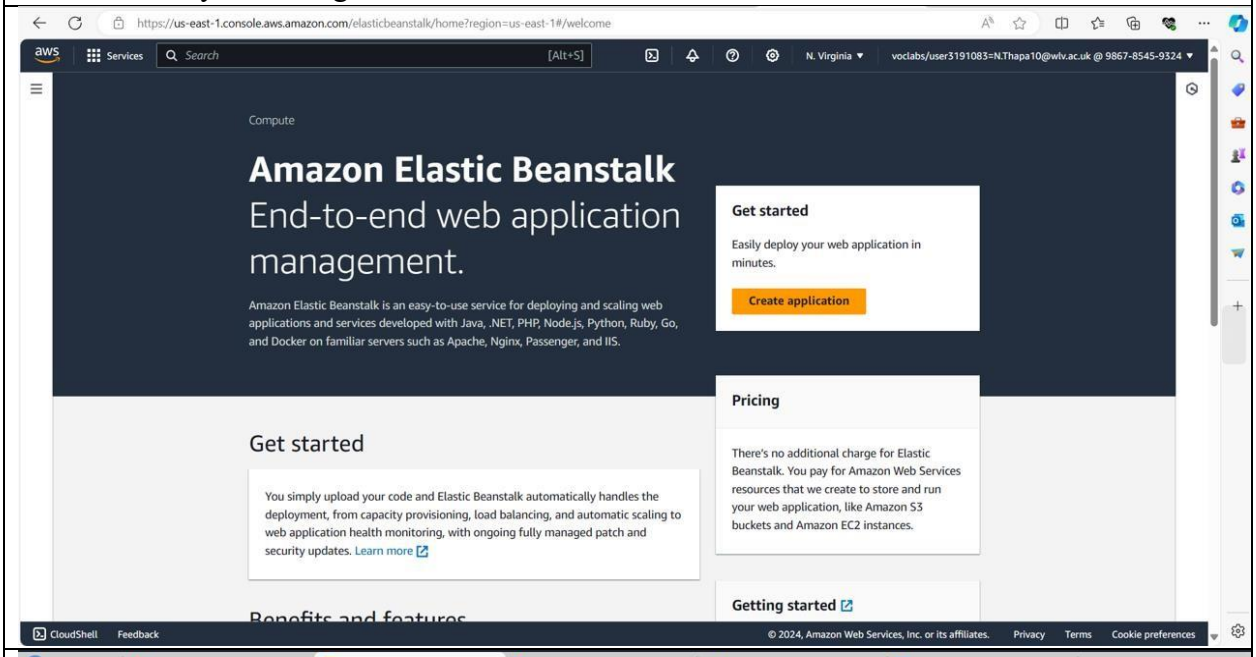
1. **Access Elastic Beanstalk**:
    - From the main AWS Console, I went to the **Services** menu. •

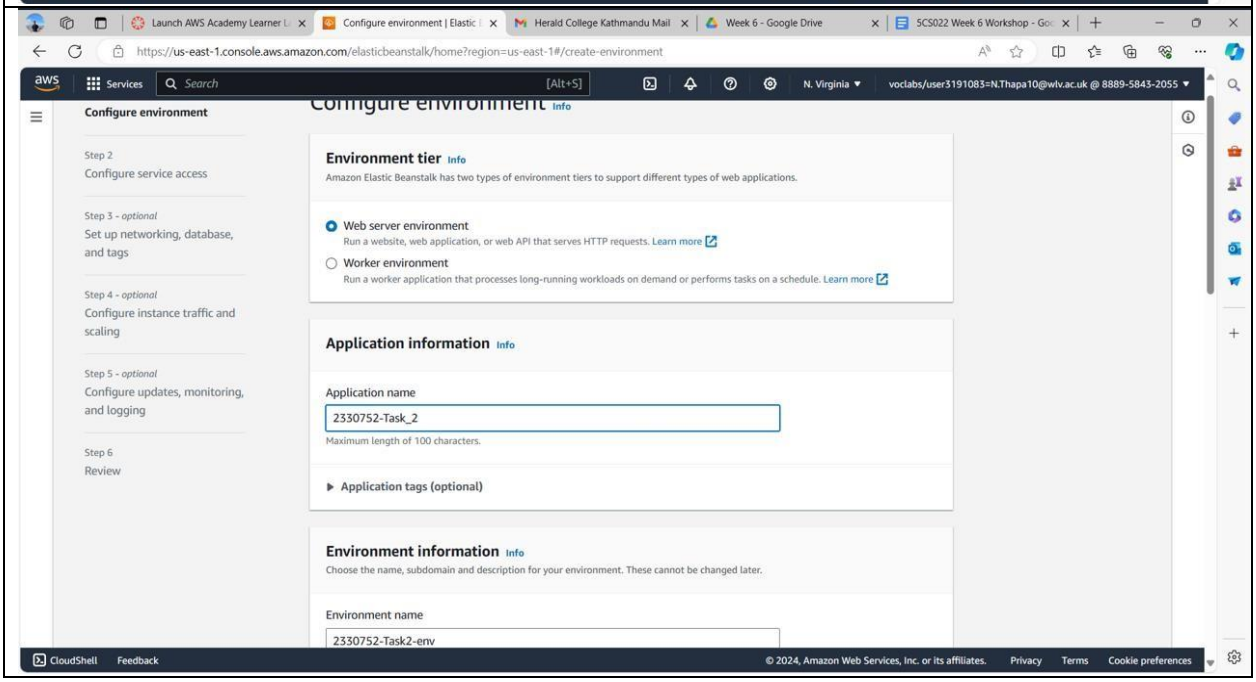
      Under the **Compute** section, I selected **Elastic Beanstalk**.
2. **Create an Application**:

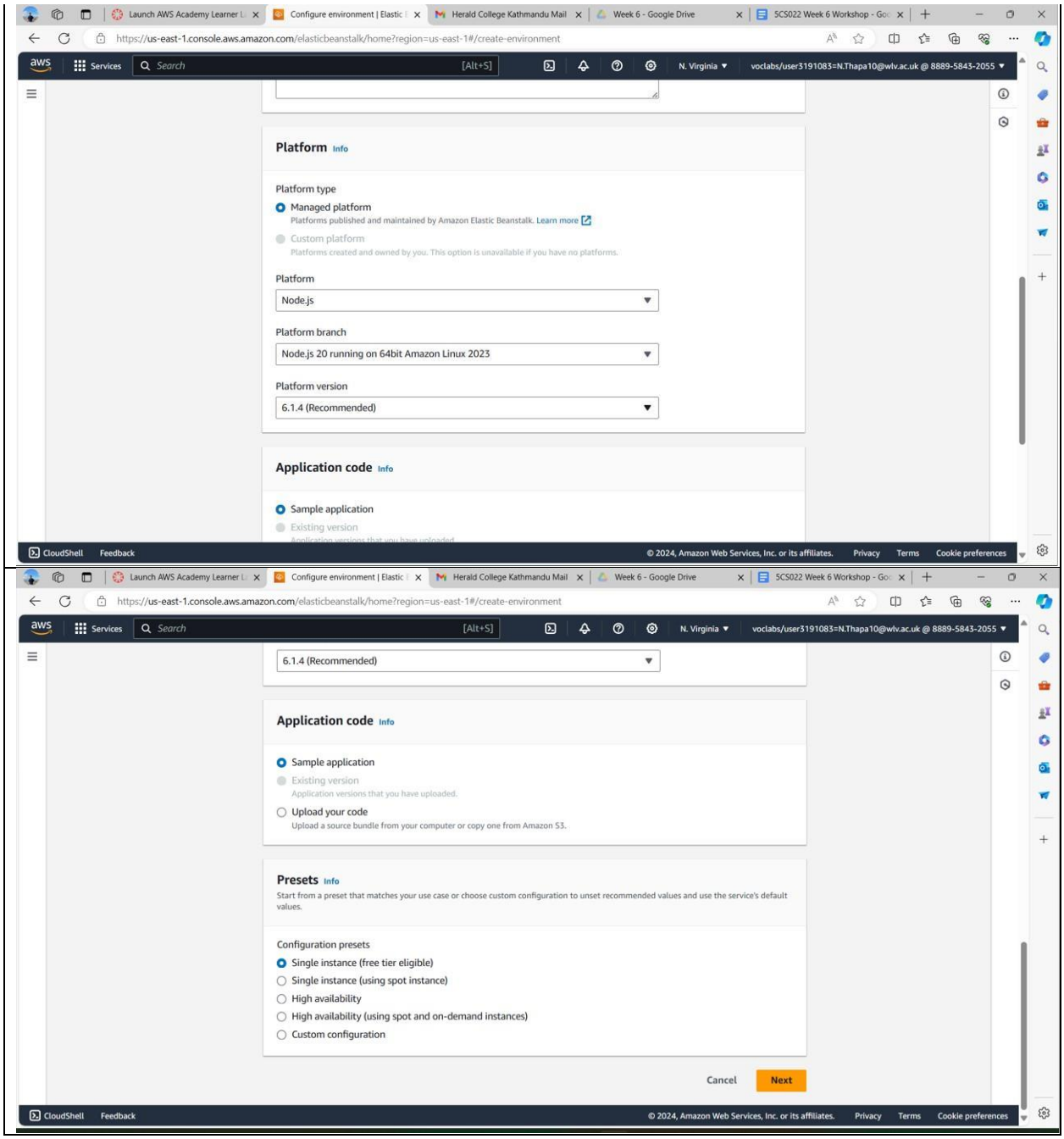
- I clicked on **Create Application**.
- I entered for the **Application name**.
- I chose **Node.js** for the **Platform**.
- I selected **Sample application** for the **Application code**.
- I clicked **Next** at the bottom of the page.

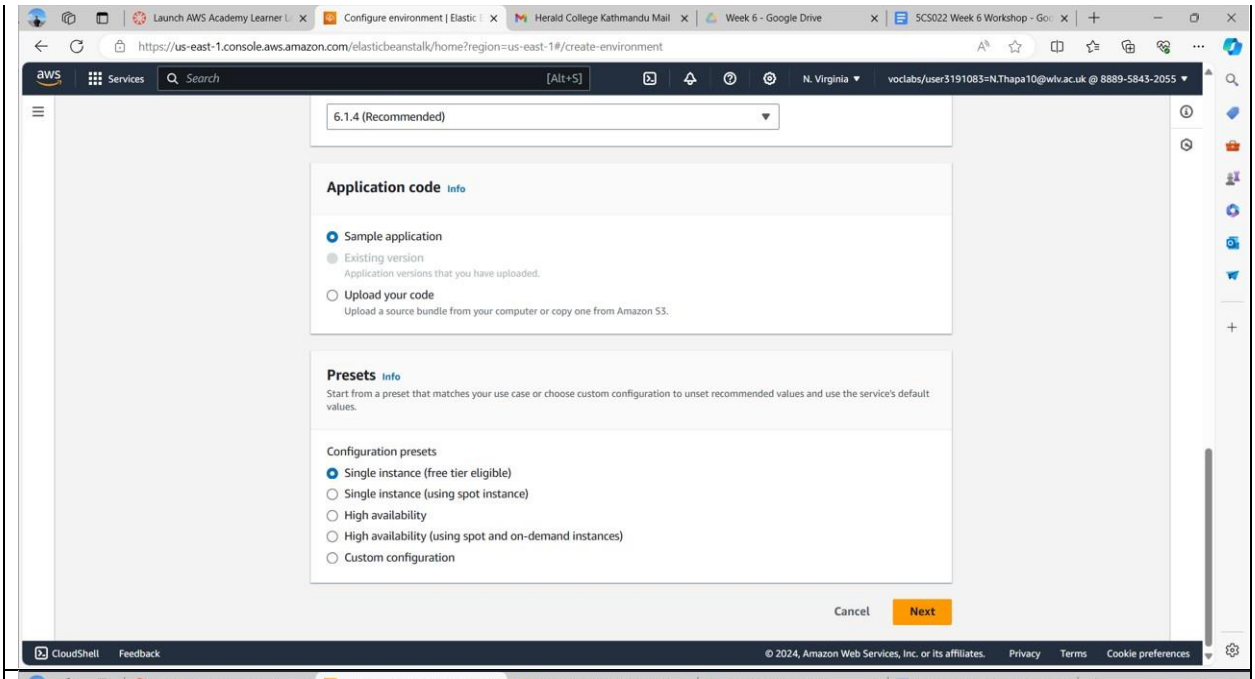
3. **Configure Service Access**:
    - On the **Configure service access** tab, I selected the required options.
    - I clicked **Skip to review**.
    - I scrolled down and clicked **Submit**.
4. **Wait for Environment Setup**:
    - I waited a few minutes for Elastic Beanstalk to set up the environment, which included creating an S3 bucket, a security group, launching an EC2 instance, and running the sample application.
5. **Upload and Deploy the Application**:
    - Once the environment was ready, the screen displayed the new environment.
    - I created a zip file of the backend code from Part 1.
    - I returned to the Elastic Beanstalk console.
    - I clicked **Upload and deploy**.
    - I chose **Choose file**, navigated to the zip file, and clicked **Open**.
    - I clicked **Deploy**.
6. Create a .env File:
    - In the root directory of the front-end project, I created a file named .env.
7. Add API Base URL
    - I changed the line in the.env file to my real Elastic Beanstalk domain URL, as follows:

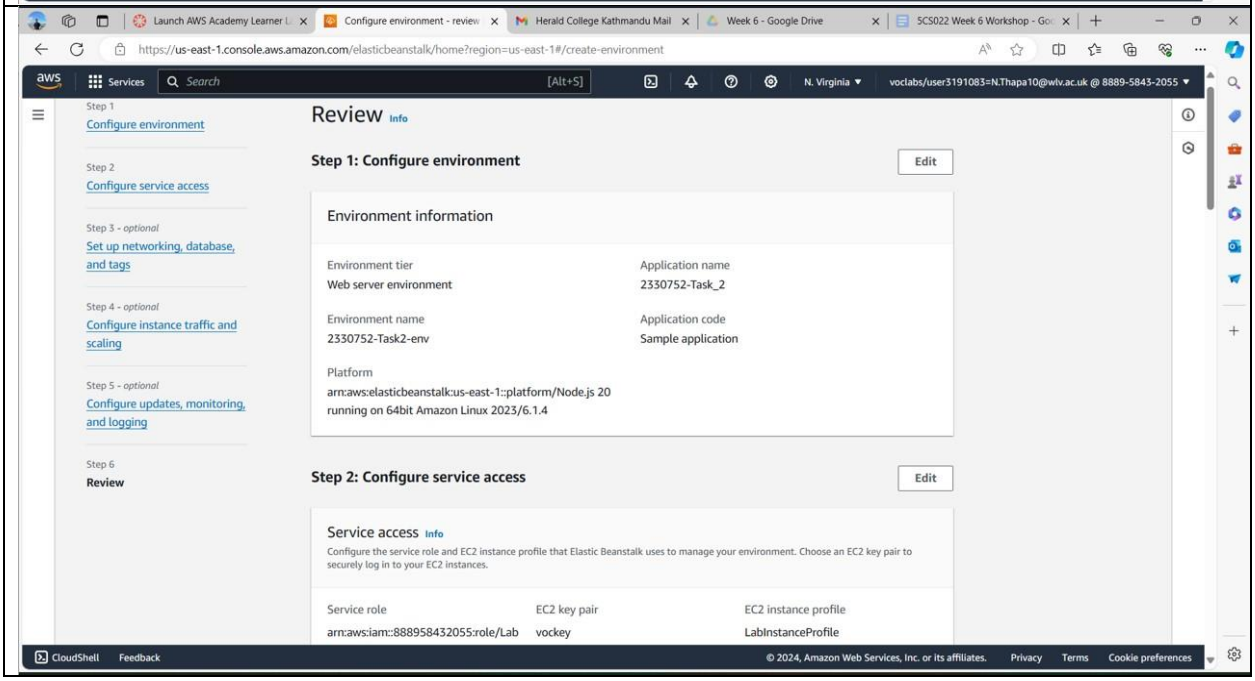
Screenshots By following all above Process:

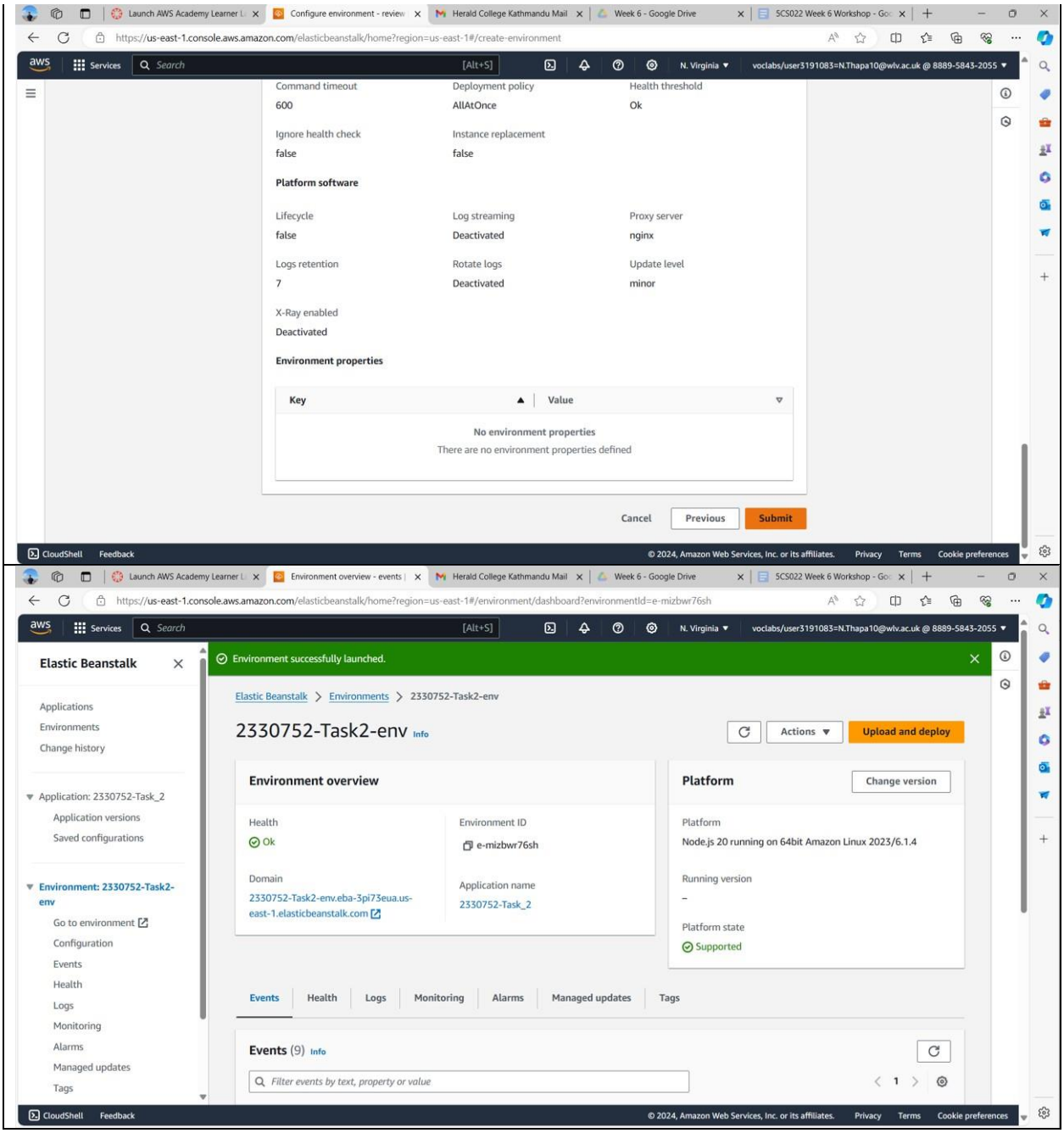


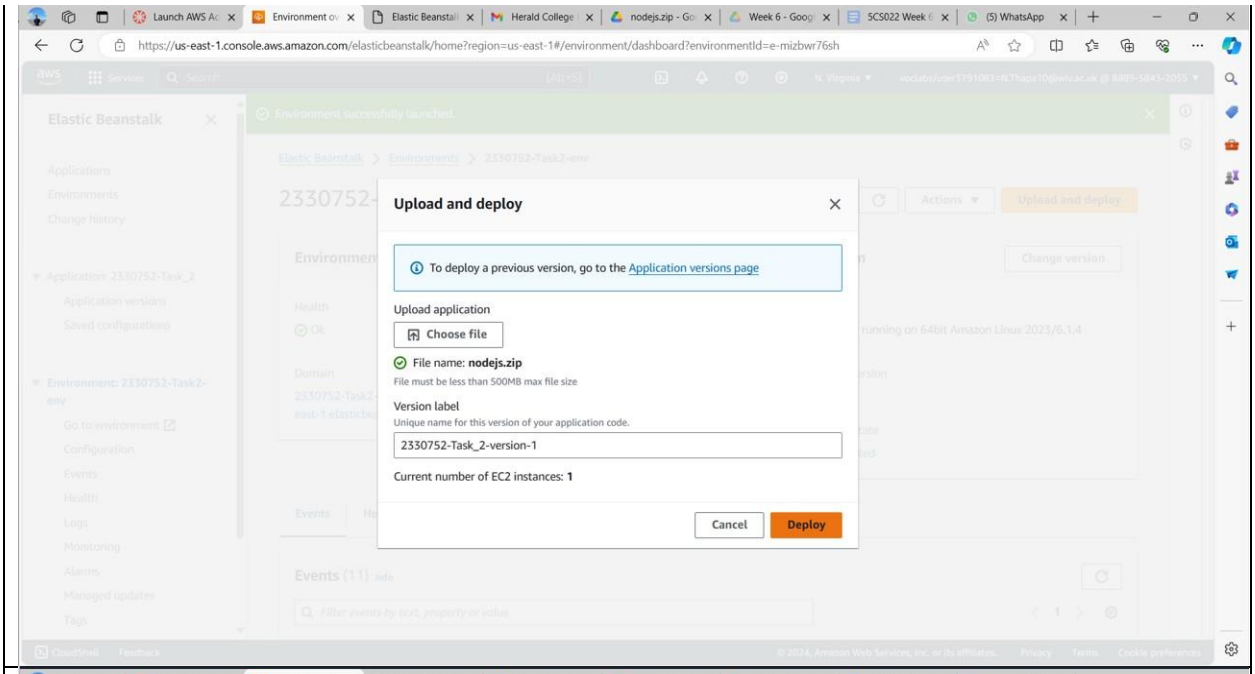


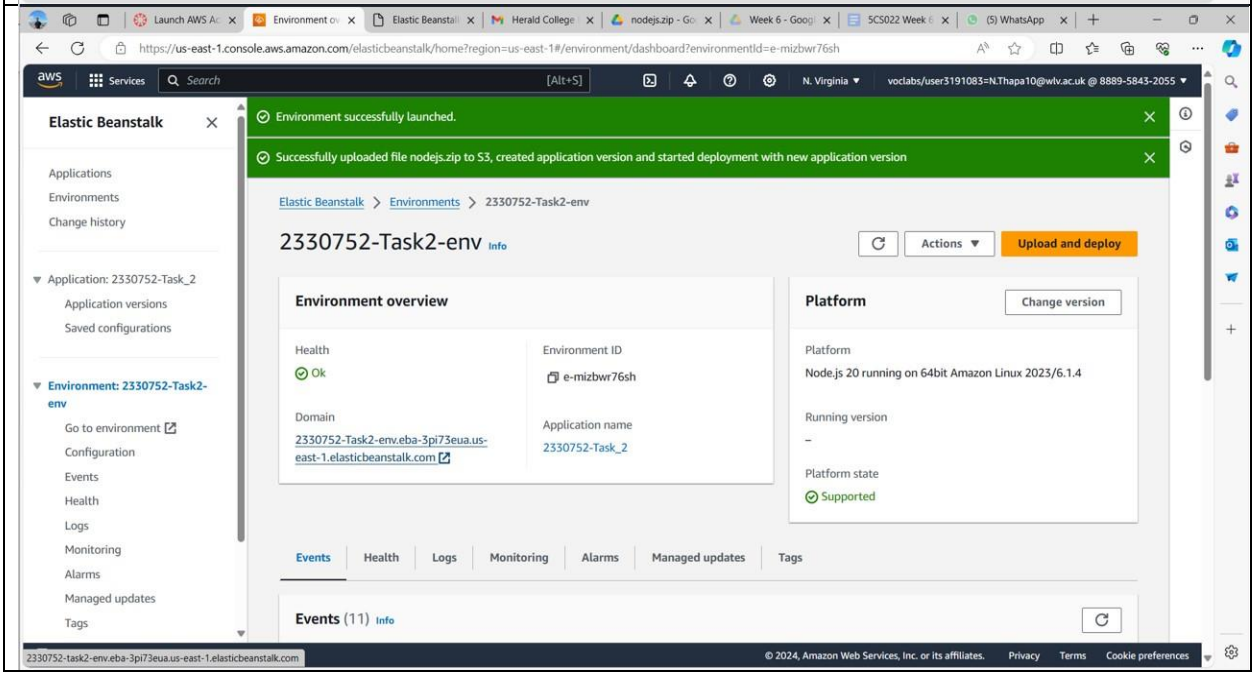

Launch AWS Academy Learner L
Configure environment | Elastic
Herald College Kathmandu Mail
Week 6 - Google Drive
5CS022 Week 6 Workshop - Goo
https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/create-environment
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3191083=N.Thapa10@wlv.ac.uk @ 8889-5843-2055
Platform Info
Platform type
Managed platform
Platforms published and maintained by Amazon Elastic Beanstalk. Learn more
Custom platform
Platforms created and owned by you. This option is unavailable if you have no platforms.
Platform
Node.js
Platform branch
Node.js 20 running on 64bit Amazon Linux 2023
Platform version
6.1.4 (Recommended)
Application code Info
Sample application
Existing version
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
6.1.4 (Recommended)
Application code Info
Sample application
Existing version
Application versions that you have uploaded.
Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.
Presets Info
Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.
Configuration presets
Single instance (free tier eligible)
Single instance (using spot instance)
High availability
High availability (using spot and on-demand instances)
Custom configuration
Cancel
Next

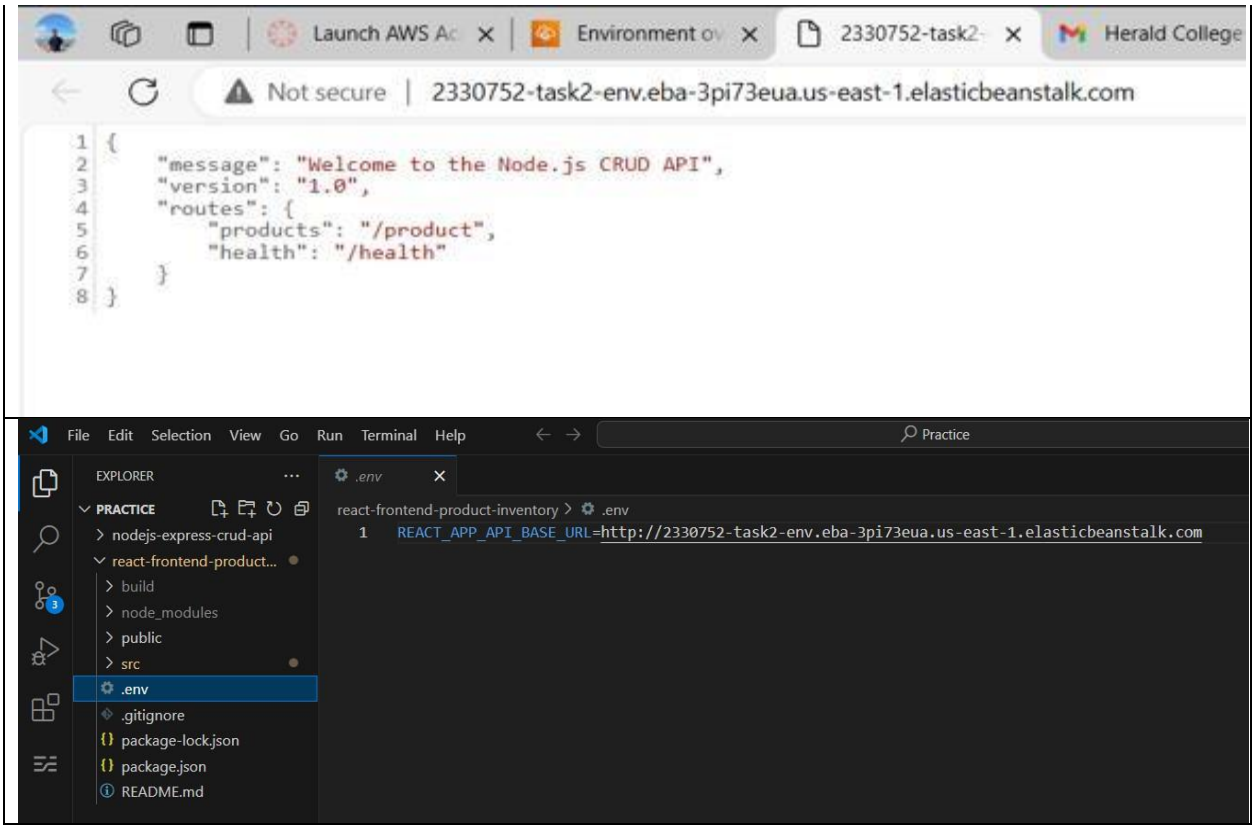

Launch AWS Academy Learner L
Configure environment | Elastic
Herald College Kathmandu Mail
Week 6 - Google Drive
5CS022 Week 6 Workshop - Goo
https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/create-environment
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3191083=N.Thapa10@wlv.ac.uk @ 8889-5843-2055
6.1.4 (Recommended)
Application code Info
Sample application
Existing version
Application versions that you have uploaded.
Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.
Presets Info
Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.
Configuration presets
Single instance (free tier eligible)
Single instance (using spot instance)
High availability
High availability (using spot and on-demand instances)
Custom configuration
Cancel
Next
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Launch AWS Academy Learner L
Configure environment - review
Herald College Kathmandu Mail
Week 6 - Google Drive
5CS022 Week 6 Workshop - Goo
https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/create-environment
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3191083=N.Thapa10@wlv.ac.uk @ 8889-5843-2055
Step 1
Configure environment
Step 2
Configure service access
Step 3 - optional
Set up networking, database, and tags
Step 4 - optional
Configure instance traffic and scaling
Step 5 - optional
Configure updates, monitoring, and logging
Step 6
Review
Review Info
Step 1: Configure environment
Edit
Environment information
Environment tier
Web server environment
Application name
2330752-Task_2
Environment name
2330752-Task2-env
Application code
Sample application
Platform
arn:aws:elasticbeanstalk:us-east-1::platform/Node.js 20 running on 64bit Amazon Linux 2023/6.1.4
Step 2: Configure service access
Edit
Service access Info
Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.
Service role
arn:aws:iam::888958432055:role/Lab
EC2 key pair
vockey
EC2 instance profile
LabInstanceProfile
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Configure environment - review
https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/create-environment
Command timeout
600
Deployment policy
AllAtOnce
Health threshold
Ok
Ignore health check
false
Instance replacement
false
Platform software
Lifecycle
false
Log streaming
Deactivated
Proxy server
nginx
Logs retention
7
Rotate logs
Deactivated
Update level
minor
X-Ray enabled
Deactivated
Environment properties
Key
Value
No environment properties
There are no environment properties defined
Cancel
Previous
Submit
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Environment overview - events
https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/environment/dashboard?environmentId=e-mizbwr76sh
Elastic Beanstalk
Environment successfully launched.
Elastic Beanstalk > Environments > 2330752-Task2-env
2330752-Task2-env
Info
Actions
Upload and deploy
Environment overview
Health
Ok
Environment ID
e-mizbwr76sh
Domain
2330752-Task2-env.eba-3pi73eua.us-east-1.elasticbeanstalk.com
Application name
2330752-Task_2
Platform
Change version
Platform
Node.js 20 running on 64bit Amazon Linux 2023/6.1.4
Running version
-
Platform state
Supported
Applications
Environments
Change history
Application: 2330752-Task_2
Application versions
Saved configurations
Environment: 2330752-Task2-env
Go to environment
Configuration
Events
Health
Logs
Monitoring
Alarms
Managed updates
Tags
Events
Health
Logs
Monitoring
Alarms
Managed updates
Tags
Events (9)
Info
Filter events by text, property or value


Upload and deploy
To deploy a previous version, go to the Application versions page
Upload application
Choose file
File name: nodejs.zip
File must be less than 500MB max file size
Version label
Unique name for this version of your application code.
2330752-Task_2-version-1
Current number of EC2 instances: 1
Cancel
Deploy


Elastic Beanstalk
Environment successfully launched.
Successfully uploaded file nodejs.zip to S3, created application version and started deployment with new application version
Elastic Beanstalk > Environments > 2330752-Task2-env
2330752-Task2-env Info
Actions
Upload and deploy
Environment overview
Health
Ok
Environment ID
e-mizbwr76sh
Domain
2330752-Task2-env.eba-3pi73eua.us-east-1.elasticbeanstalk.com
Application name
2330752-Task_2
Platform
Change version
Platform
Node.js 20 running on 64bit Amazon Linux 2023/6.1.4
Running version
-
Platform state
Supported
Events
Health
Logs
Monitoring
Alarms
Managed updates
Tags
Events (11) Info



2.4 Part 4: Uploading the Frontend

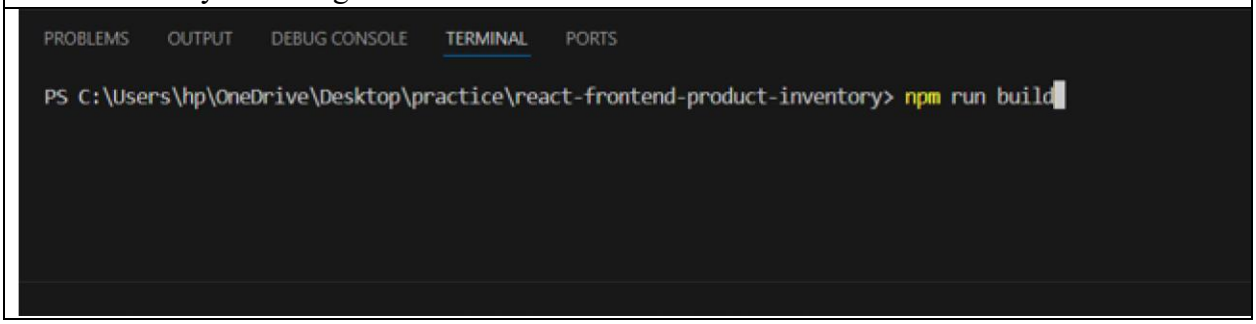
I took the following actions to upload the frontend React static files to AWS S3:

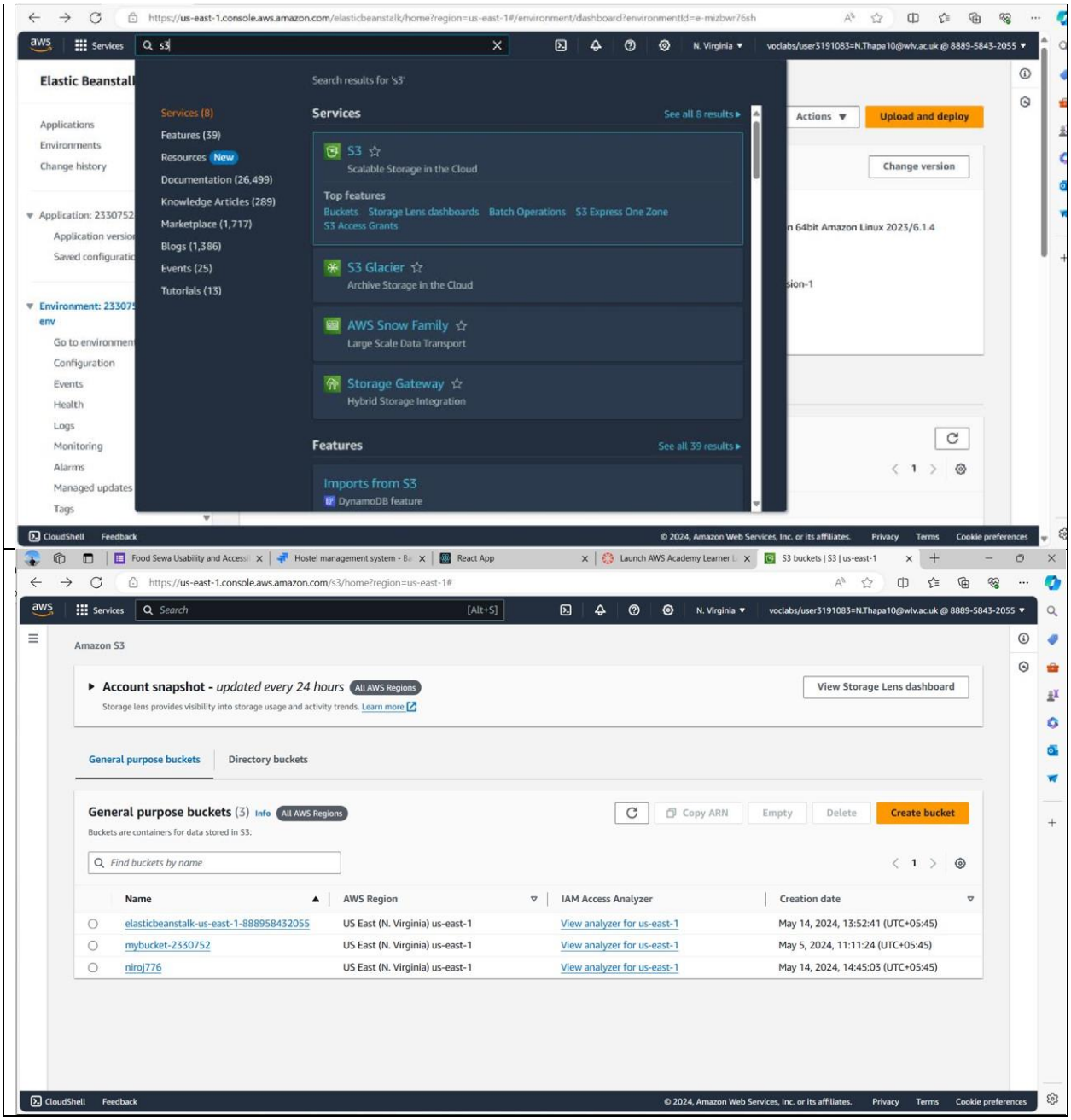
1. **Install Dependencies**:
    - I opened the frontend code directory that I had copied in Part 1.
    - To install all required dependencies, I used npm install.
2. **Build the React App**:
    - To build the React app, I used the command npm run build. The build directory containing the static files for deployment was created as a result.
3. **Access AWS S3**:
    - I looked for AWS S3 using the AWS Management Console.
4. **Create a New Bucket**:
    - I selected **Create bucket**.
    - I left the bucket type as **General purpose**.

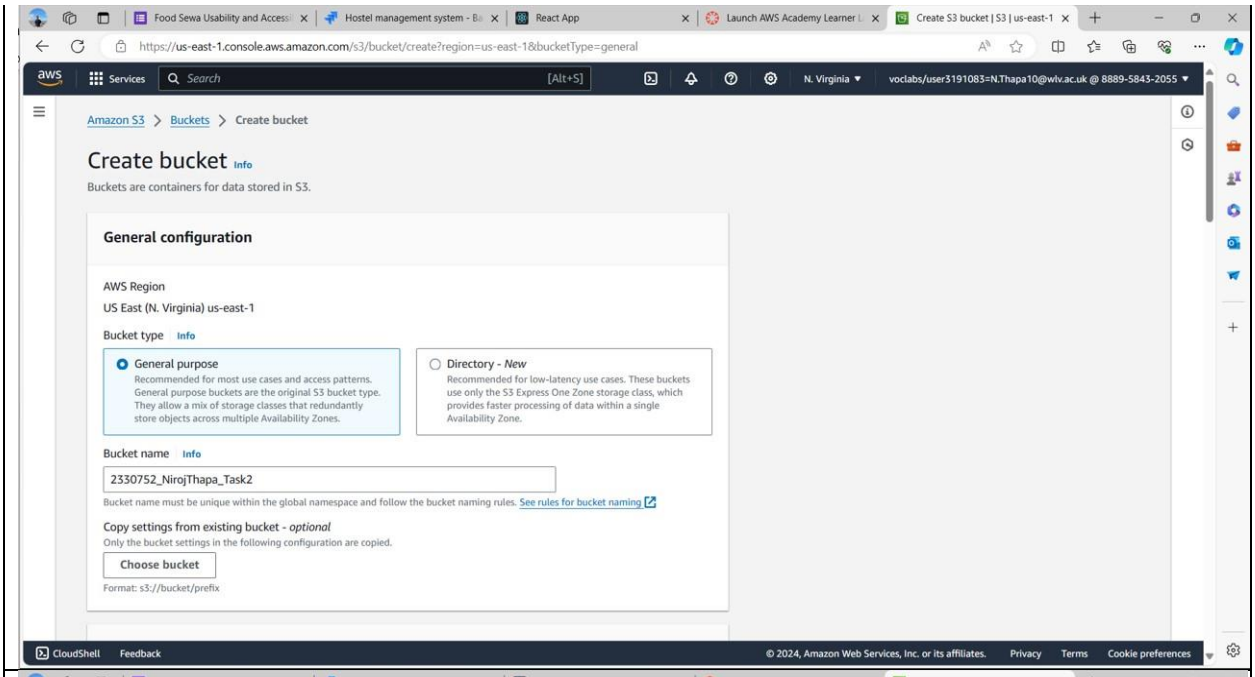
- I kept the AWS Region as **US East**.
- For the bucket name, I entered a unique DNS-compliant name, named as **myawsbuckettask2**. (The name contained only lowercase characters.)

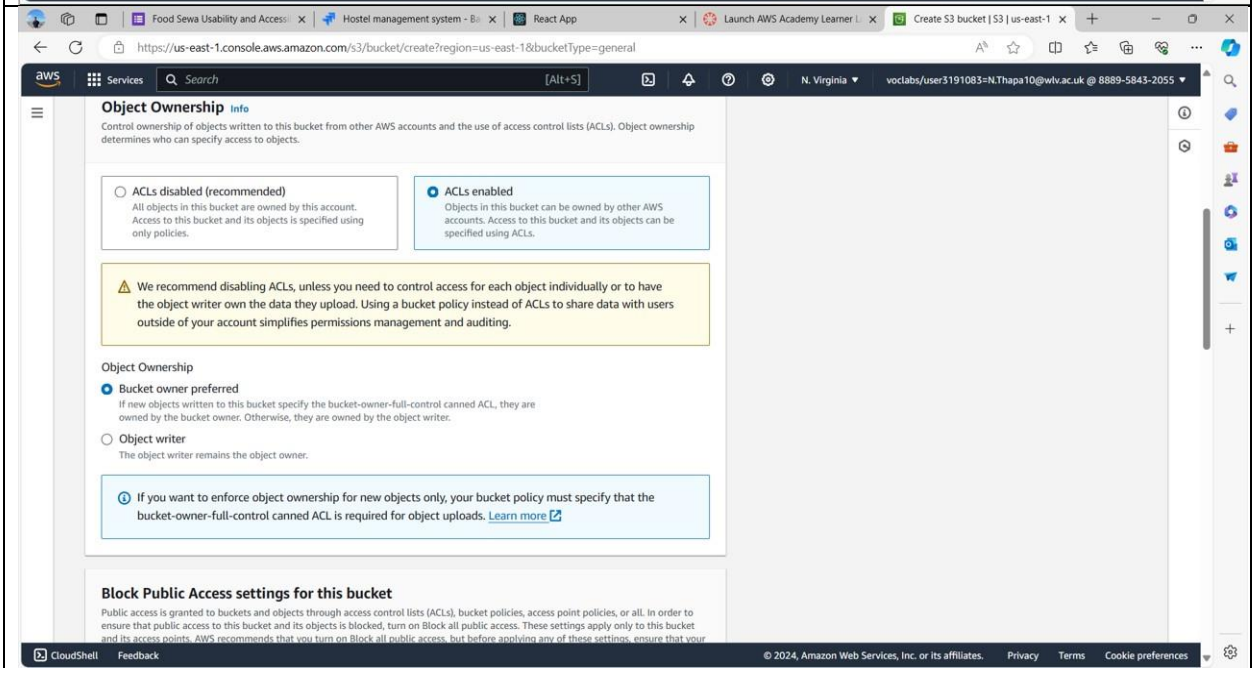
5. **Configure Bucket Settings**:
    - For **Object Ownership**, I selected **ACLs enabled**.
    - I left the **Block all public access** settings as they were.
6. **Finalize Bucket Creation**:
    - I selected Create bucket at the bottom of the page after scrolling down.
7. **Upload Build Files**:
    - After the bucket was created, I navigated to the newly created bucket.
    - I selected Upload.
    - After choosing Add files, I went to the previously made build directory.
    - I uploaded all files and folders within the **build** directory to the S3 bucket.

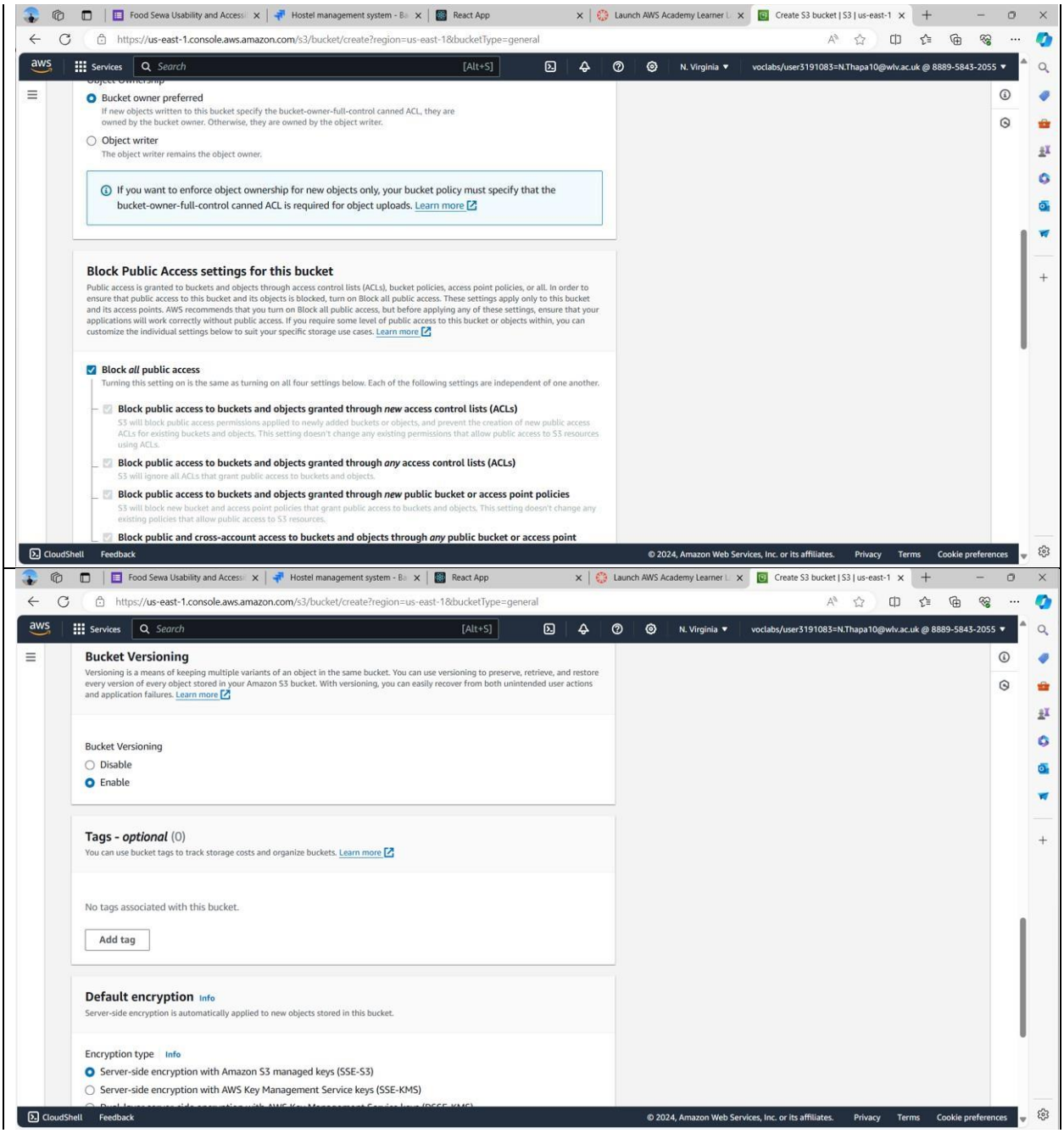
Screenshots By following all above Process:



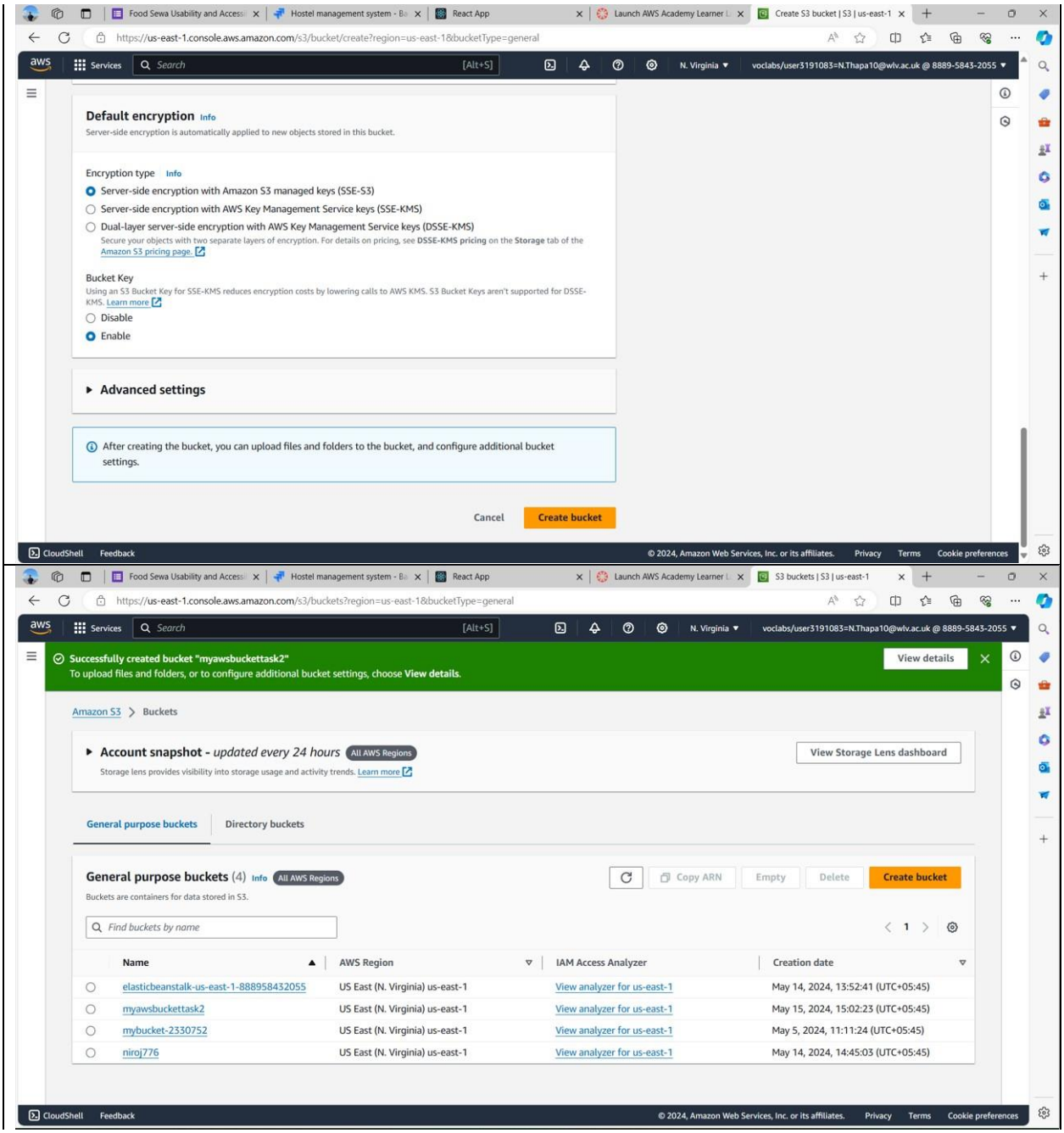

Elastic Beanstalk
Search results for 's3'
Services (8)
Features (39)
Resources New
Documentation (26,499)
Knowledge Articles (289)
Marketplace (1,717)
Blogs (1,386)
Events (25)
Tutorials (13)
Services
See all 8 results
S3
Scalable Storage in the Cloud
Top features
Buckets Storage Lens dashboards Batch Operations S3 Express One Zone
S3 Access Grants
S3 Glacier
Archive Storage in the Cloud
AWS Snow Family
Large Scale Data Transport
Storage Gateway
Hybrid Storage Integration
Features
See all 39 results
Imports from S3
DynamoDB feature
Actions
Upload and deploy
Change version
Amazon S3
Account snapshot - updated every 24 hours
All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
General purpose buckets
Directory buckets
General purpose buckets (3) Info
All AWS Regions
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
IAM Access Analyzer
Creation date
elasticbeanstalk-us-east-1-888958432055
US East (N. Virginia) us-east-1
View analyzer for us-east-1
May 14, 2024, 13:52:41 (UTC+05:45)
mybucket-2330752
US East (N. Virginia) us-east-1
View analyzer for us-east-1
May 5, 2024, 11:11:24 (UTC+05:45)
niroj776
US East (N. Virginia) us-east-1
View analyzer for us-east-1
May 14, 2024, 14:45:03 (UTC+05:45)
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Food Sewa Usability and Access
Hostel management system - Ba
React App
Launch AWS Academy Learner L
Create S3 bucket | S3 | us-east-1
https://us-east-1.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3191083=N.Thapa10@wlv.ac.uk @ 8889-5843-2055
Amazon S3 > Buckets > Create bucket
Create bucket Info
Buckets are containers for data stored in S3.
General configuration
AWS Region
US East (N. Virginia) us-east-1
Bucket type Info
General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
Directory - New
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.
Bucket name Info
2330752_NirojThapa_Task2
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket
Format: s3://bucket/prefix
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

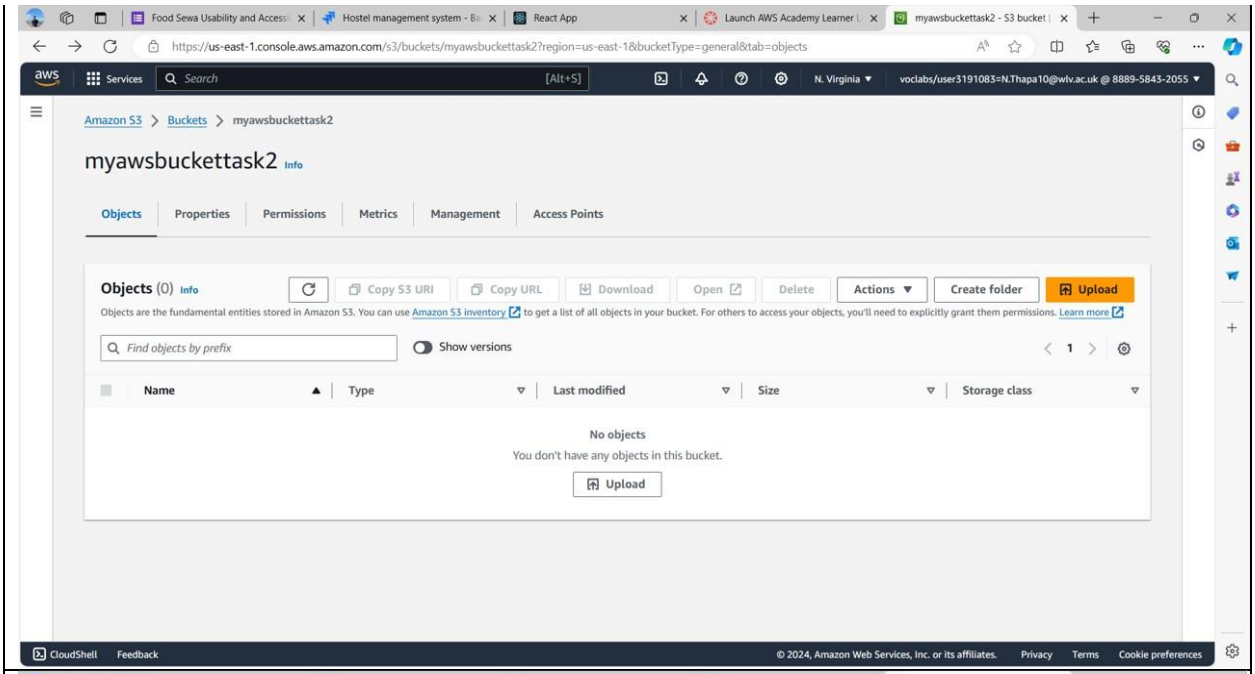

Food Sewa Usability and Access
Hostel management system - Ba
React App
Launch AWS Academy Learner L
Create S3 bucket | S3 | us-east-1
https://us-east-1.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3191083=N.Thapa10@wlv.ac.uk @ 8889-5843-2055
Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.
ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.
We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.
Object Ownership
Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.
Object writer
The object writer remains the object owner.
If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. Learn more
Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

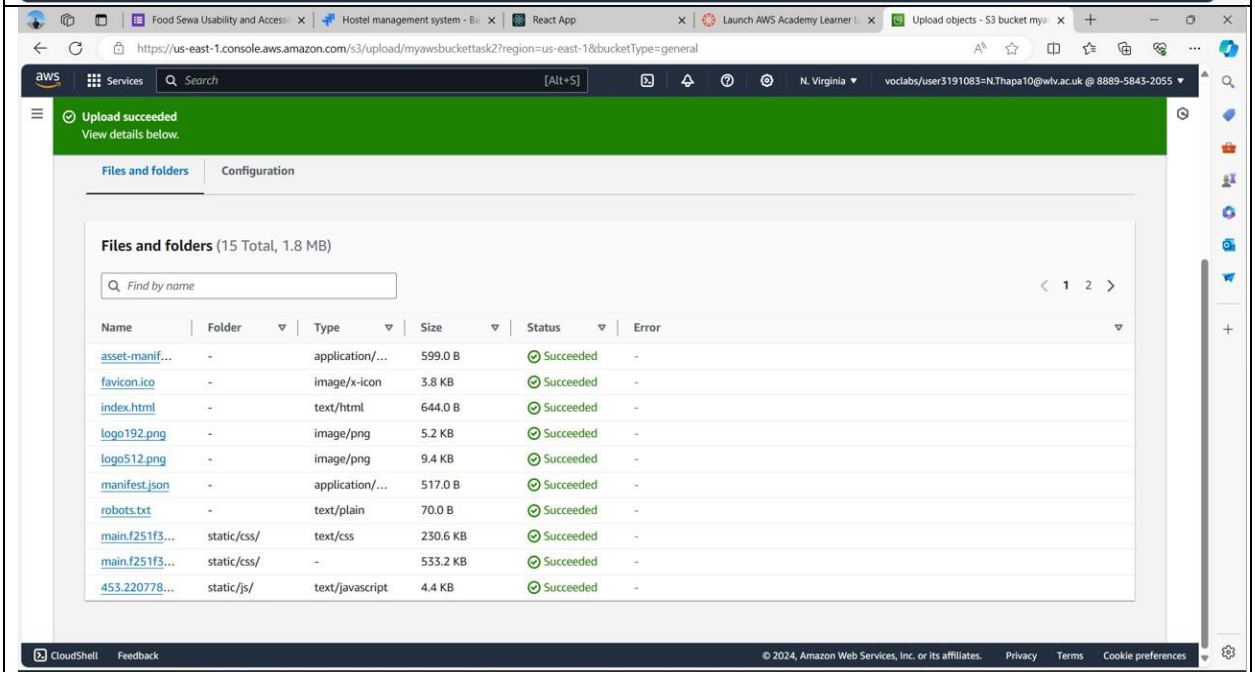

Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.
Object writer
The object writer remains the object owner.
If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. Learn more
Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
Block public and cross-account access to buckets and objects through any public bucket or access point
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

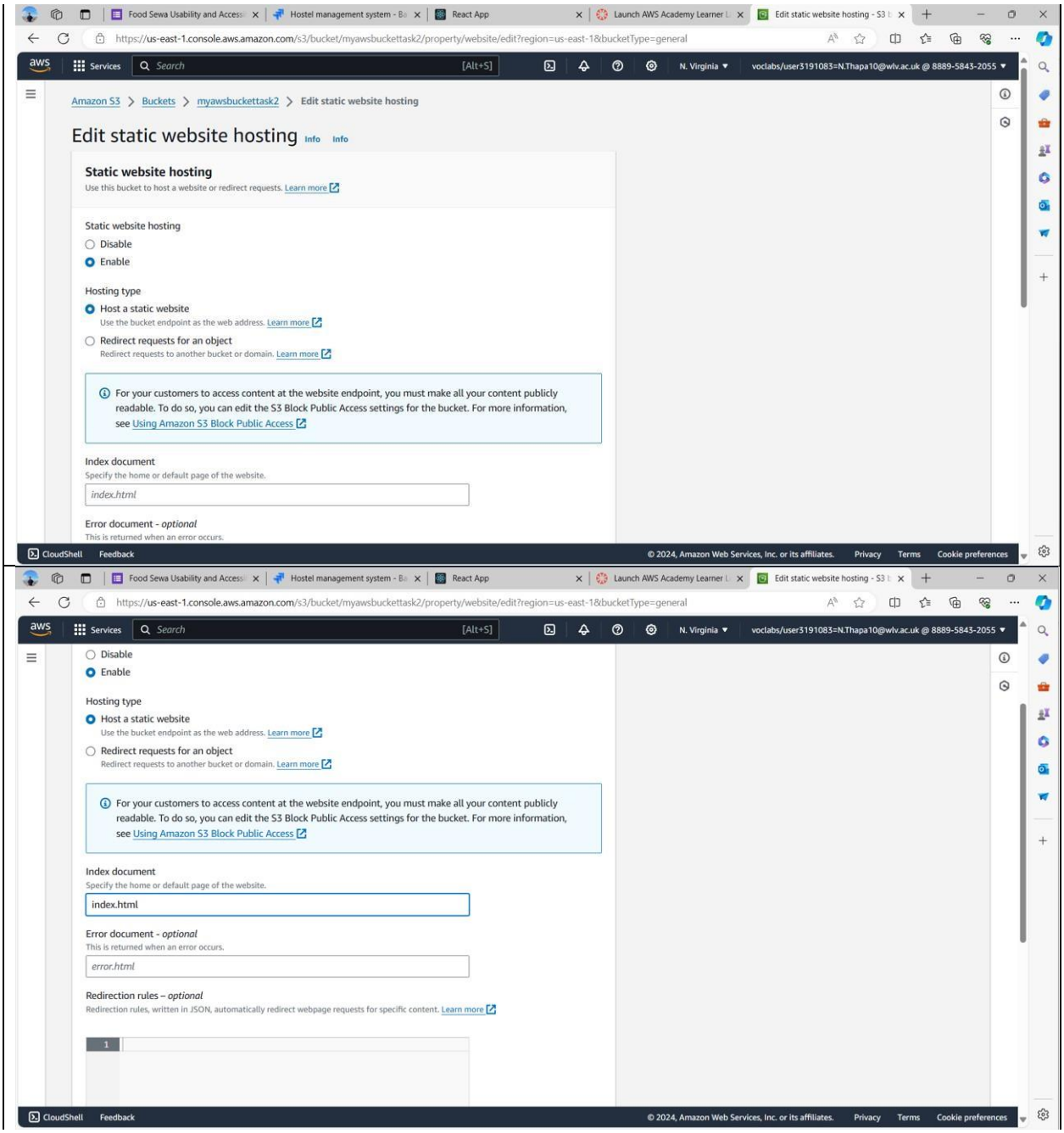
Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more
Bucket Versioning
Disable
Enable
Tags - optional (0)
You can use bucket tags to track storage costs and organize buckets. Learn more
No tags associated with this bucket.
Add tag
Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.
Encryption type Info
Server-side encryption with Amazon S3 managed keys (SSE-S3)
Server-side encryption with AWS Key Management Service keys (SSE-KMS)
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

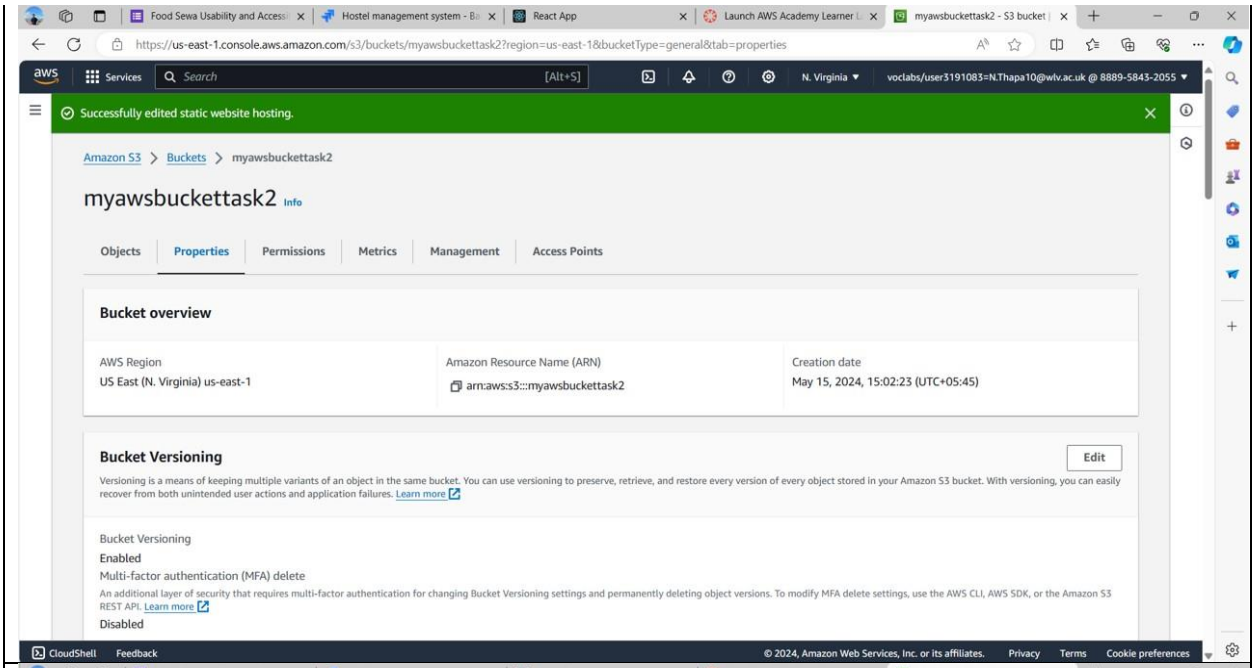

Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.
Encryption type Info
Server-side encryption with Amazon S3 managed keys (SSE-S3)
Server-side encryption with AWS Key Management Service keys (SSE-KMS)
Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the Amazon S3 pricing page.
Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more
Disable
Enable
Advanced settings
After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.
Cancel
Create bucket

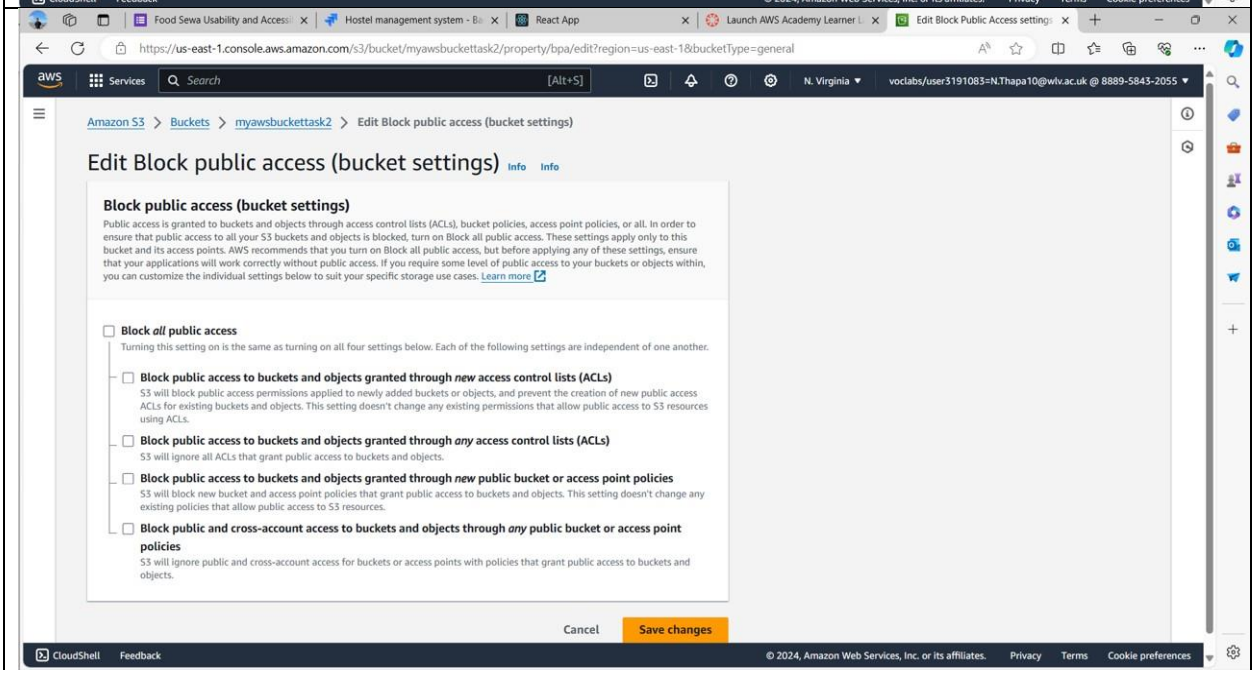
Successfully created bucket "myawsbuckettask2"
To upload files and folders, or to configure additional bucket settings, choose View details.
View details
Amazon S3 > Buckets
Account snapshot - updated every 24 hours
All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
General purpose buckets
Directory buckets
General purpose buckets (4) Info
All AWS Regions
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
IAM Access Analyzer
Creation date
elasticbeanstalk-us-east-1-888958432055
US East (N. Virginia) us-east-1
View analyzer for us-east-1
May 14, 2024, 13:52:41 (UTC+05:45)
myawsbuckettask2
US East (N. Virginia) us-east-1
View analyzer for us-east-1
May 15, 2024, 15:02:23 (UTC+05:45)
mybucket-2330752
US East (N. Virginia) us-east-1
View analyzer for us-east-1
May 5, 2024, 11:11:24 (UTC+05:45)
niroj776
US East (N. Virginia) us-east-1
View analyzer for us-east-1
May 14, 2024, 14:45:03 (UTC+05:45)

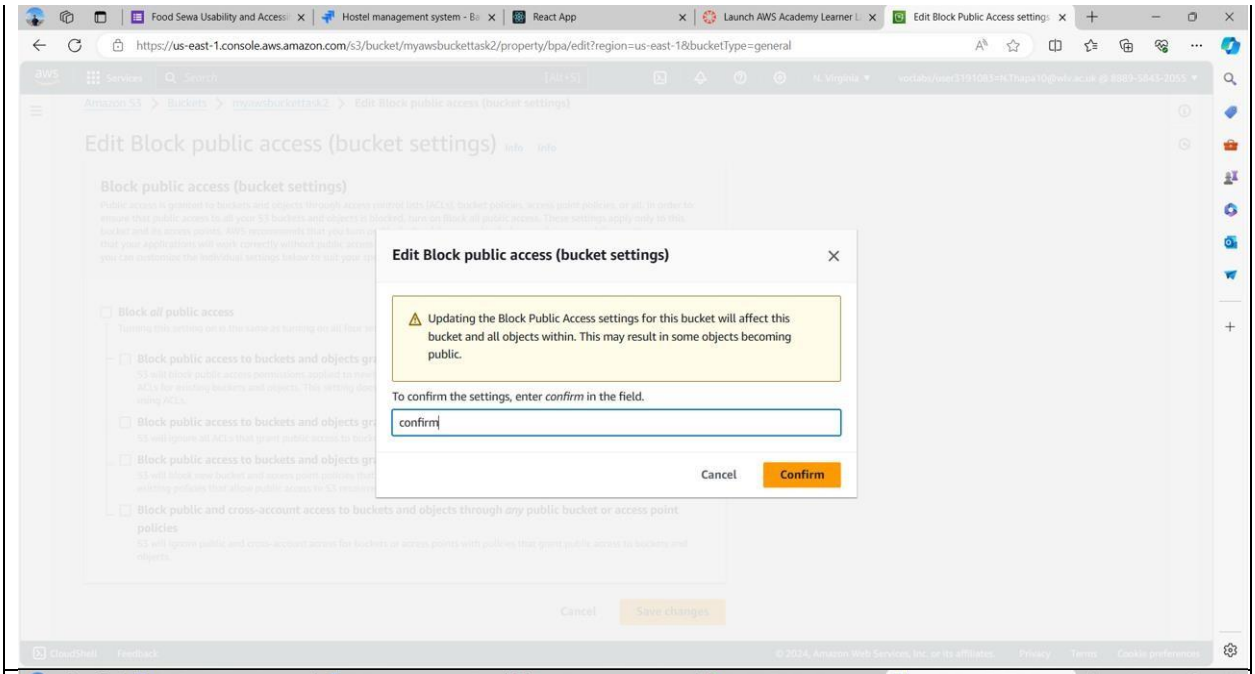

https://us-east-1.console.aws.amazon.com/s3/buckets/myawsbuckettask2?region=us-east-1&bucketType=general&tab=objects
Amazon S3 > Buckets > myawsbuckettask2
myawsbuckettask2 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (0) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Show versions
Name
Type
Last modified
Size
Storage class
No objects
You don't have any objects in this bucket.
Upload
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

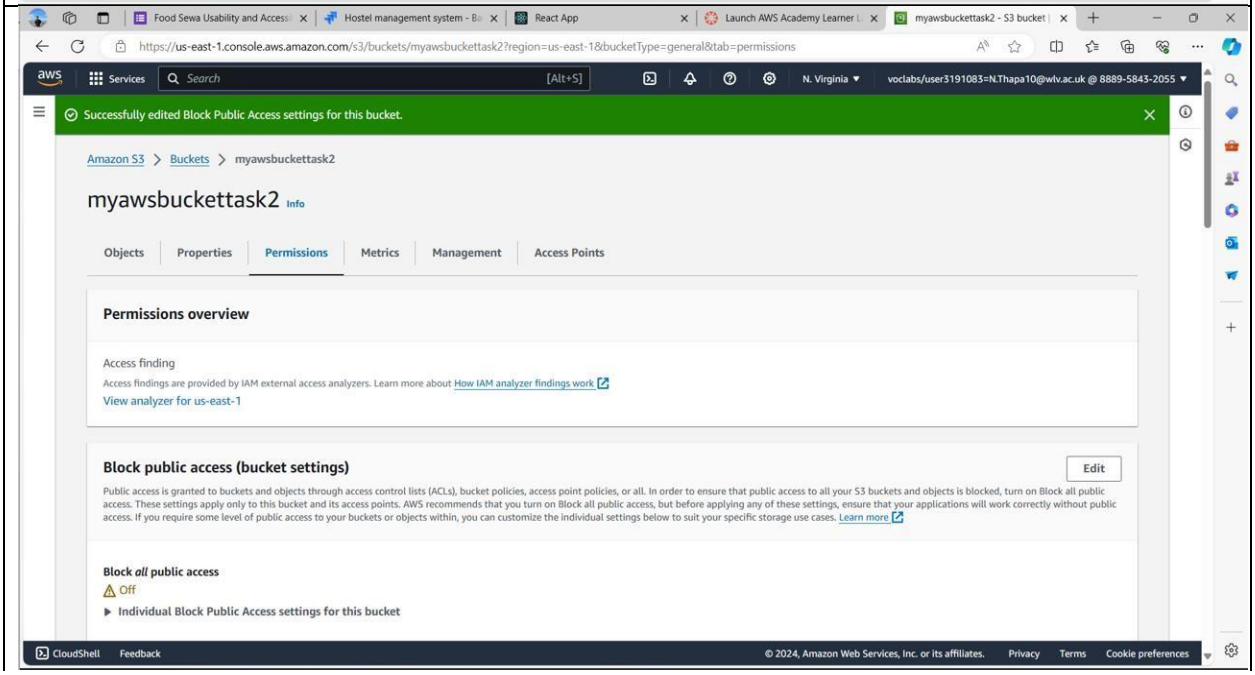

https://us-east-1.console.aws.amazon.com/s3/upload/myawsbuckettask2?region=us-east-1&bucketType=general
Upload succeeded
View details below.
Files and folders
Configuration
Files and folders (15 Total, 1.8 MB)
Find by name
Name
Folder
Type
Size
Status
Error
asset-manif... - application/... 599.0 B Succeeded -
favicon.ico - image/x-icon 3.8 KB Succeeded -
index.html - text/html 644.0 B Succeeded -
logo192.png - image/png 5.2 KB Succeeded -
logo512.png - image/png 9.4 KB Succeeded -
manifest.json - application/... 517.0 B Succeeded -
robots.txt - text/plain 70.0 B Succeeded -
main.f251f3... static/css/ text/css 230.6 KB Succeeded -
main.f251f3... static/css/ - 533.2 KB Succeeded -
453.220778... static/js/ text/javascript 4.4 KB Succeeded -
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

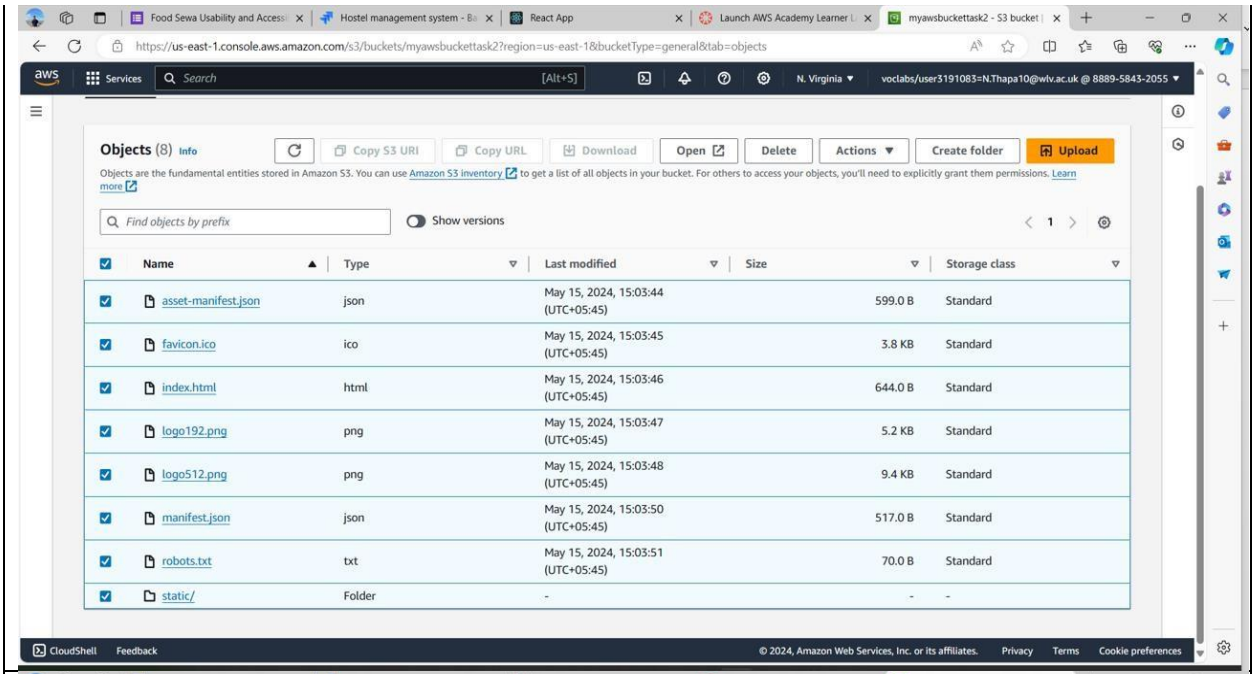

Food Sewa Usability and Access
Hostel management system - Ba
React App
Launch AWS Academy Learner L
Edit static website hosting - S3 b
https://us-east-1.console.aws.amazon.com/s3/bucket/myawsbuckettask2/property/website/edit?region=us-east-1&bucketType=general
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3191083=N.Thapa10@wlv.ac.uk @ 8889-5843-2055
Amazon S3 > Buckets > myawsbuckettask2 > Edit static website hosting
Edit static website hosting Info Info
Static website hosting
Use this bucket to host a website or redirect requests. Learn more
Static website hosting
Disable
Enable
Hosting type
Host a static website
Use the bucket endpoint as the web address. Learn more
Redirect requests for an object
Redirect requests to another bucket or domain. Learn more
For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access
Index document
Specify the home or default page of the website.
index.html
Error document - optional
This is returned when an error occurs.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Disable
Enable
Hosting type
Host a static website
Use the bucket endpoint as the web address. Learn more
Redirect requests for an object
Redirect requests to another bucket or domain. Learn more
For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access
Index document
Specify the home or default page of the website.
index.html
Error document - optional
This is returned when an error occurs.
error.html
Redirection rules – optional
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. Learn more
1
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

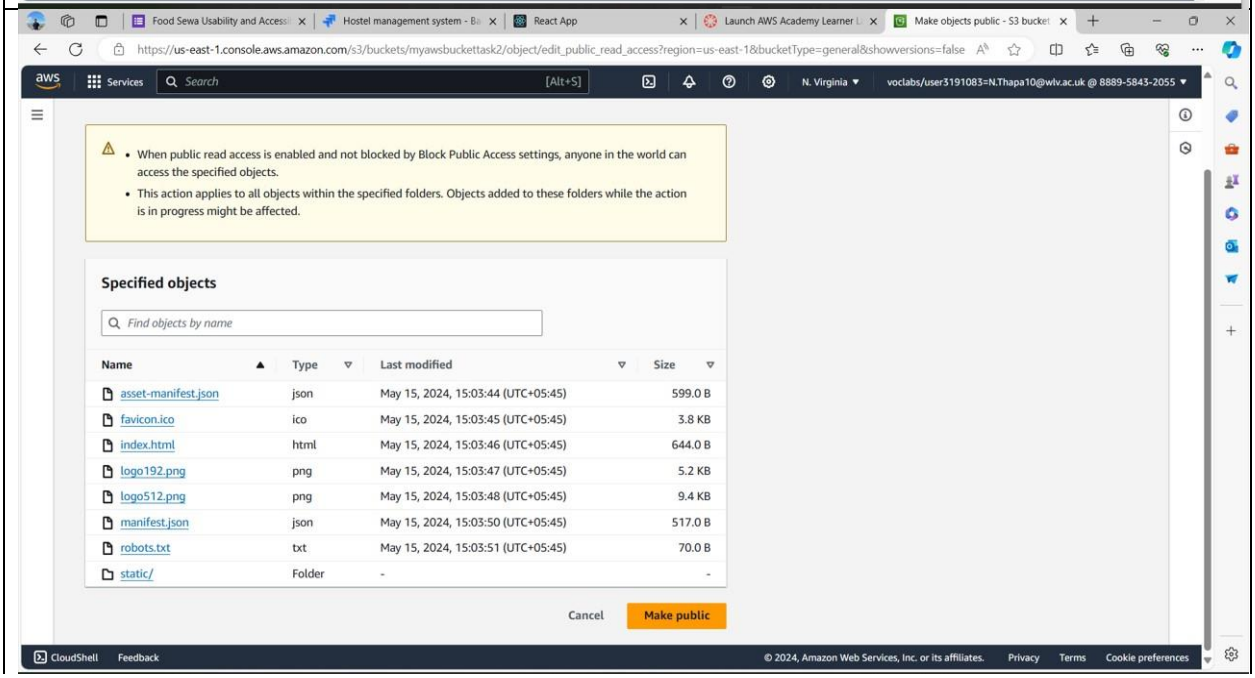

Food Sewa Usability and Access
Hostel management system - Ba
React App
Launch AWS Academy Learner L
myawsbuckettask2 - S3 bucket
https://us-east-1.console.aws.amazon.com/s3/buckets/myawsbuckettask2?region=us-east-1&bucketType=general&tab=properties
Services
Search
[Alt+S]
N. Virginia
voclabs/user3191083=N.Thapa10@wlv.ac.uk @ 8889-5843-2055
Successfully edited static website hosting.
Amazon S3 > Buckets > myawsbuckettask2
myawsbuckettask2 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Bucket overview
AWS Region
US East (N. Virginia) us-east-1
Amazon Resource Name (ARN)
arn:aws:s3:::myawsbuckettask2
Creation date
May 15, 2024, 15:02:23 (UTC+05:45)
Bucket Versioning
Edit
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more
Bucket Versioning
Enabled
Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. Learn more
Disabled
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Food Sewa Usability and Access
Hostel management system - Ba
React App
Launch AWS Academy Learner L
Edit Block Public Access settings
https://us-east-1.console.aws.amazon.com/s3/bucket/myawsbuckettask2/property/bpa/edit?region=us-east-1&bucketType=general
Services
Search
[Alt+S]
N. Virginia
voclabs/user3191083=N.Thapa10@wlv.ac.uk @ 8889-5843-2055
Amazon S3 > Buckets > myawsbuckettask2 > Edit Block public access (bucket settings)
Edit Block public access (bucket settings) Info Info
Block public access (bucket settings)
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.
Cancel
Save changes
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Edit Block public access (bucket settings)
Edit Block public access (bucket settings)
Updating the Block Public Access settings for this bucket will affect this bucket and all objects within. This may result in some objects becoming public.
To confirm the settings, enter confirm in the field.
confirm
Cancel
Confirm


Successfully edited Block Public Access settings for this bucket.
Amazon S3 > Buckets > myawsbuckettask2
myawsbuckettask2 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Permissions overview
Access finding
Access findings are provided by IAM external access analyzers. Learn more about How IAM analyzer findings work
View analyzer for us-east-1
Block public access (bucket settings)
Edit
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Off
Individual Block Public Access settings for this bucket
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

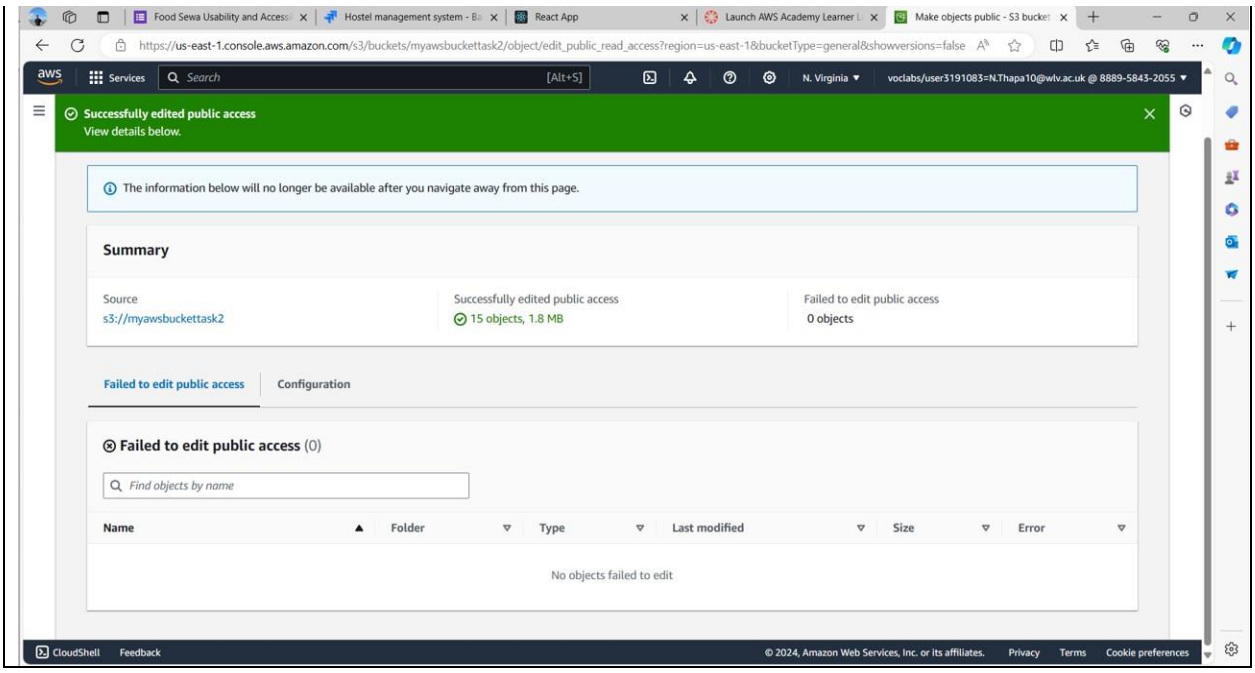




2.5 Part 5: Serving the Contents via AWS CloudFront

To configure a CloudFront distribution to serve the files from the S3 bucket, I followed these steps:

1. **Create a CloudFront Distribution**: • I made a new distribution by navigating to CloudFront in the AWS Management Console.
    - I specified my S3 bucket (**myawsbuckettask2**) as the origin.
2. **Copy the S3 Bucket Policy from CloudFront**:



- During the CloudFront setup, I copied the suggested S3 bucket policy.

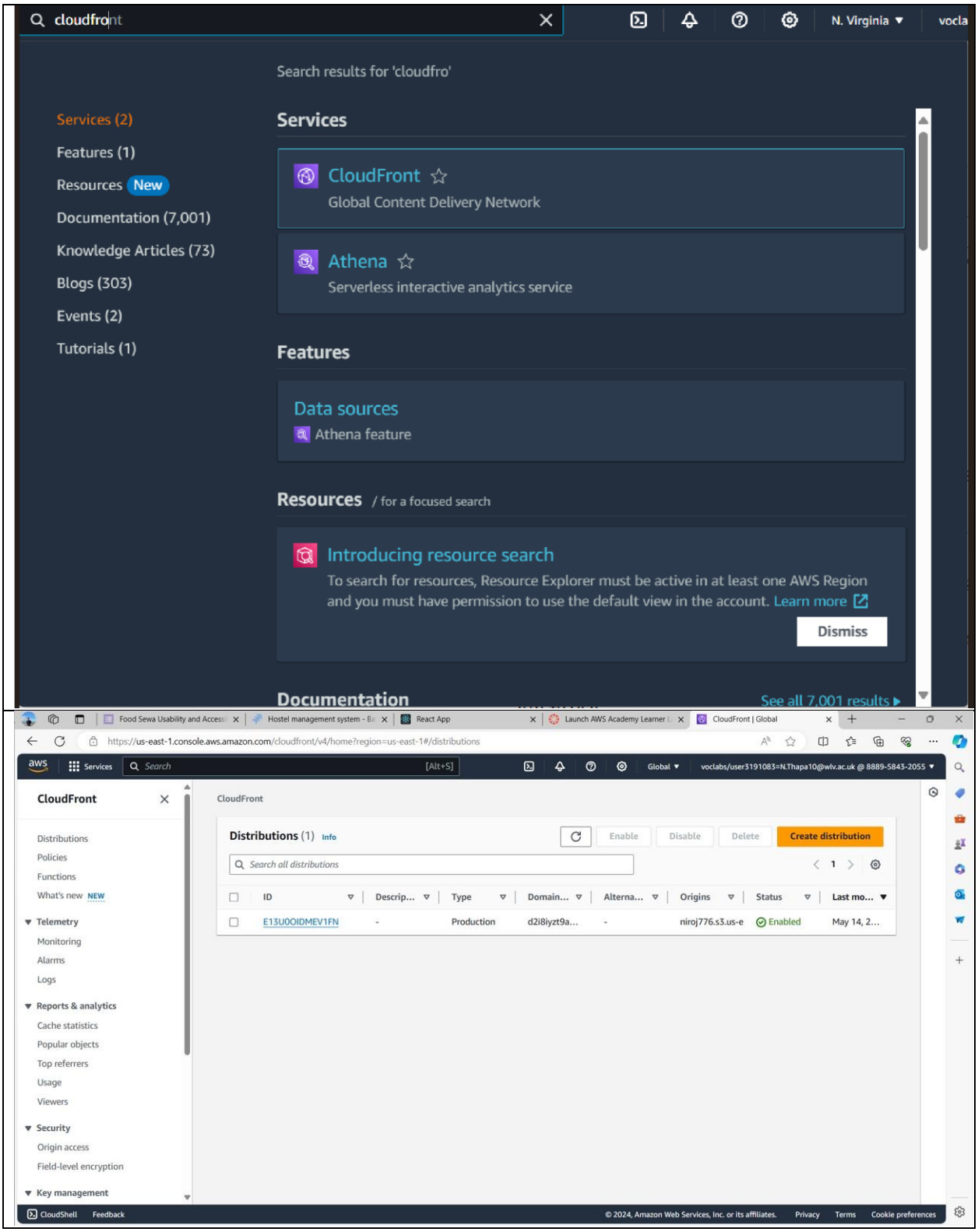
3. Revise the S3 Bucket Guidelines:

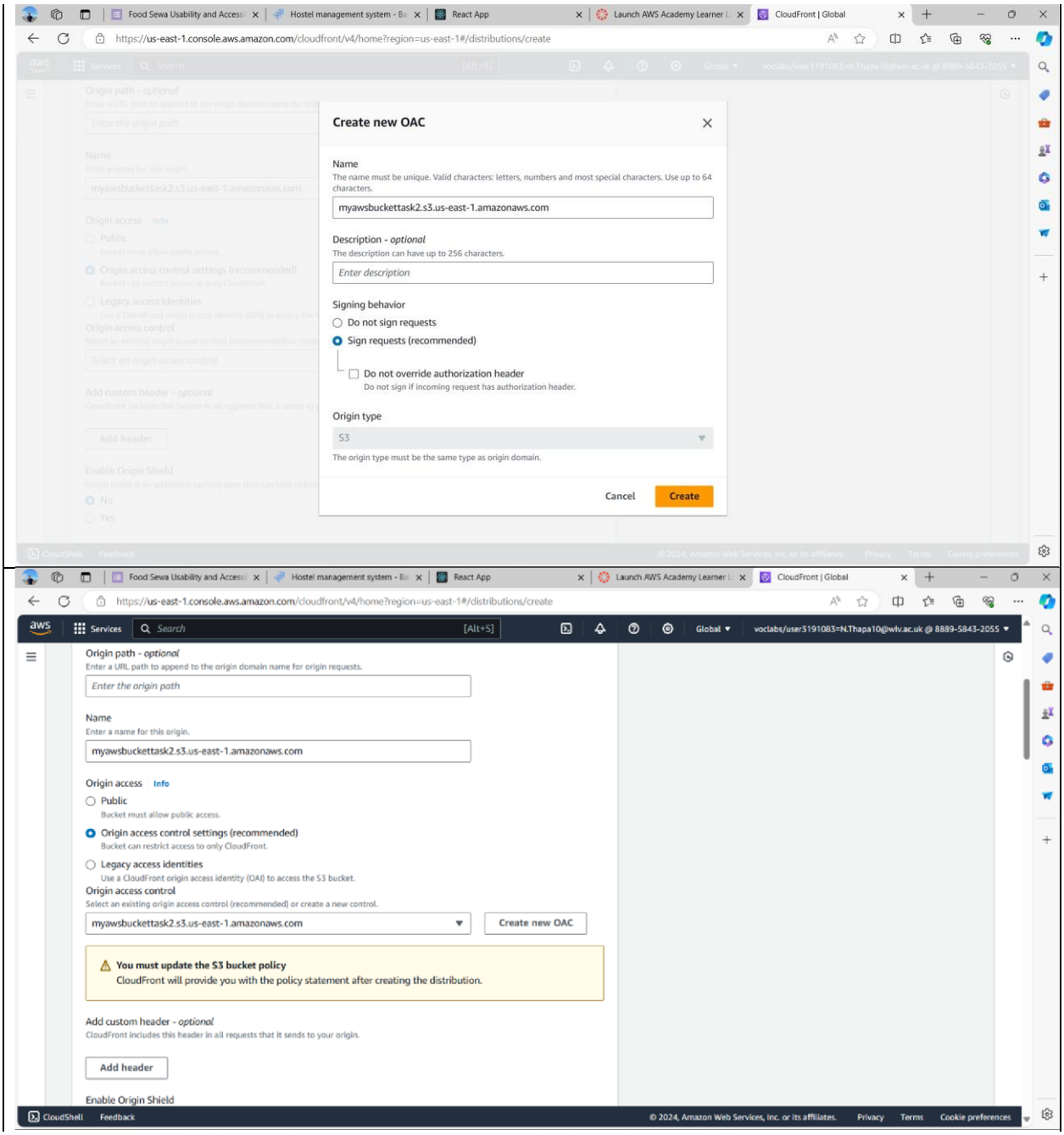
- I returned to the Amazon Management Console's S3 bucket.
- I went to the bucket's Permissions tab.
- I located the section on bucket policies and selected the Edit option.

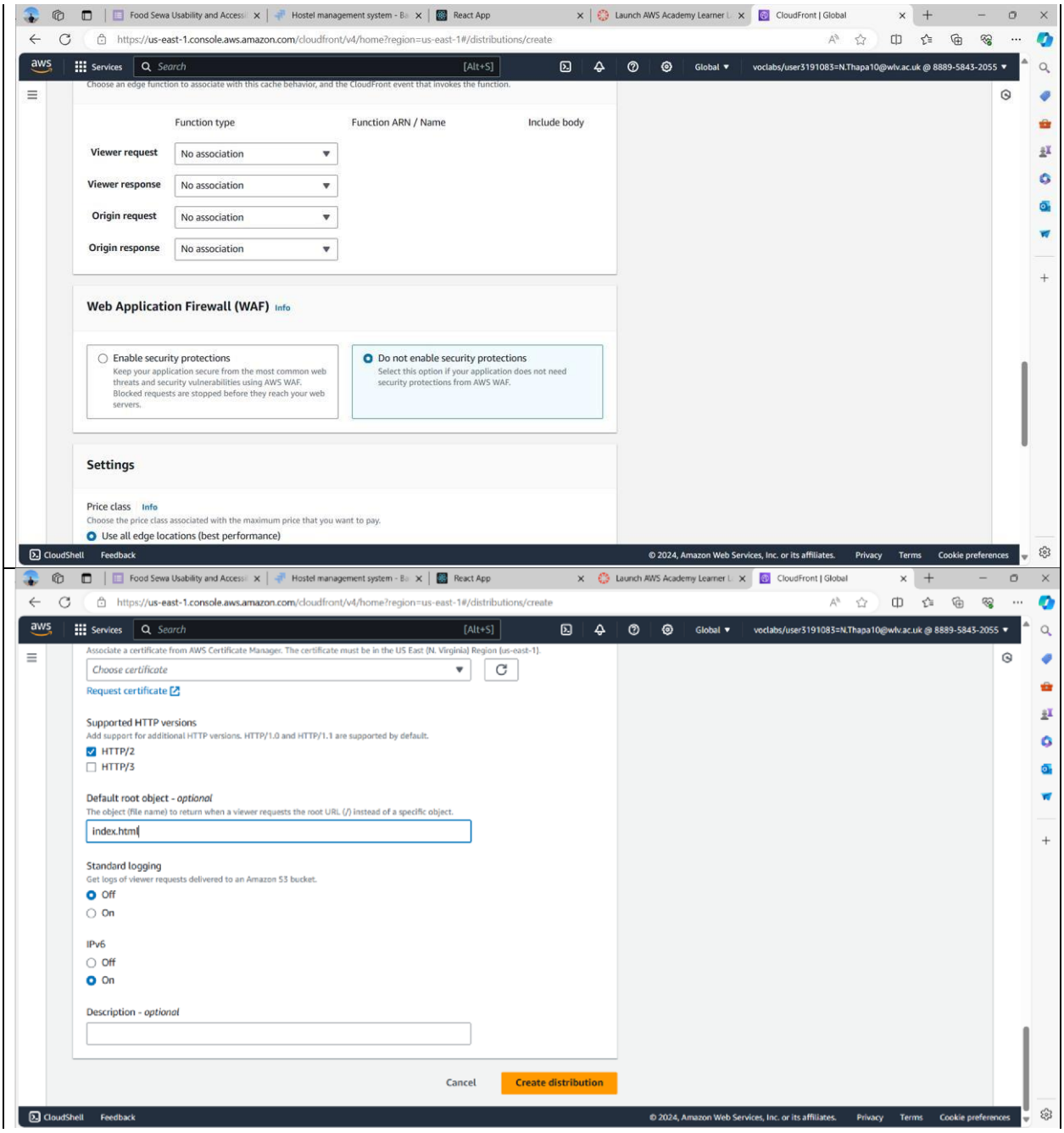
4. Use the Updated Procedure:

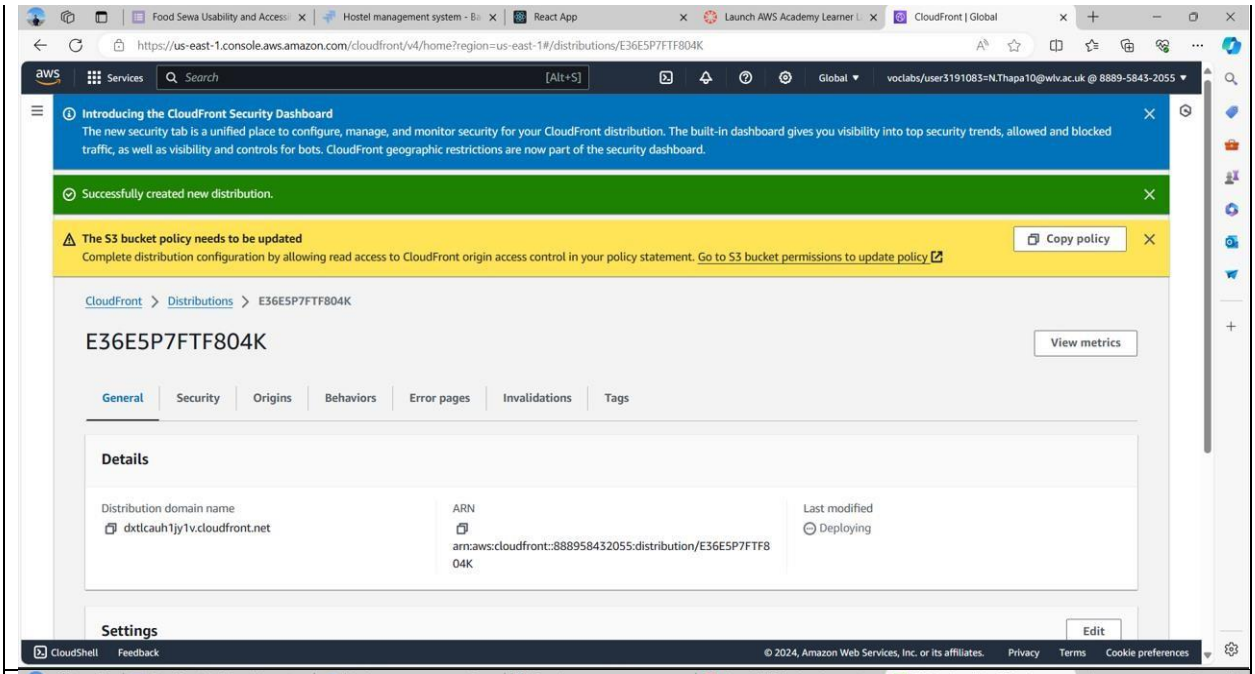
- I entered the CloudFront policy that I had copied into the policy editor.
- In order to update the policy, I selected Save changes.

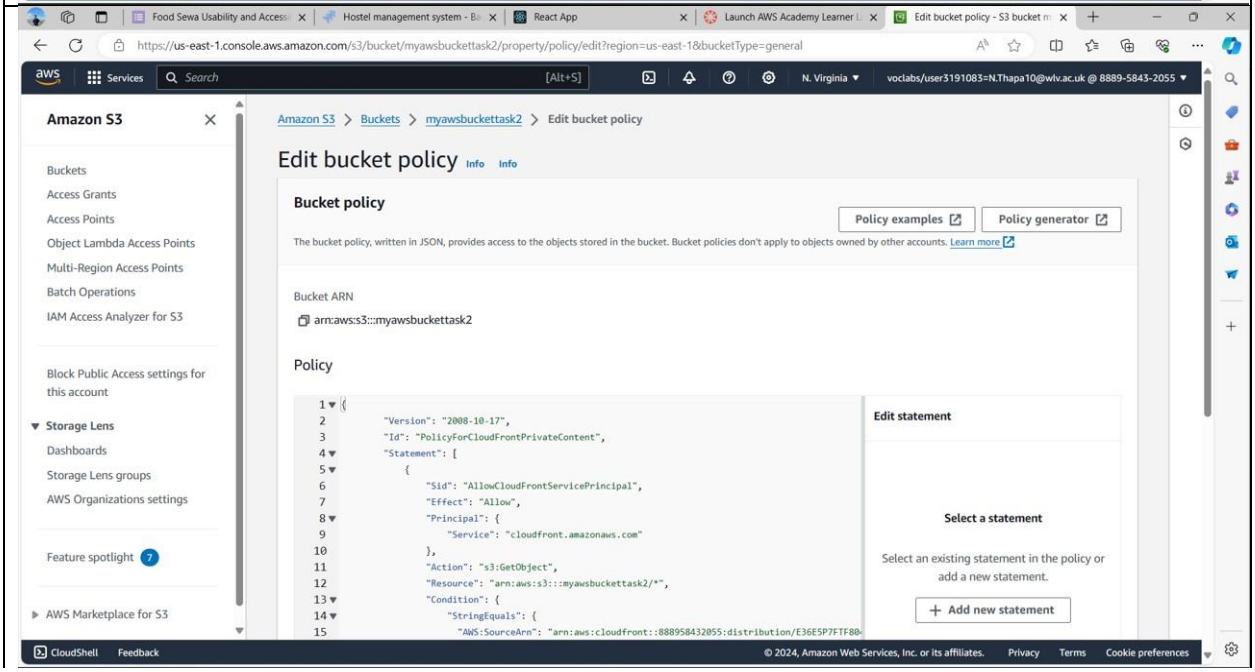
Screenshots By following all above Process:

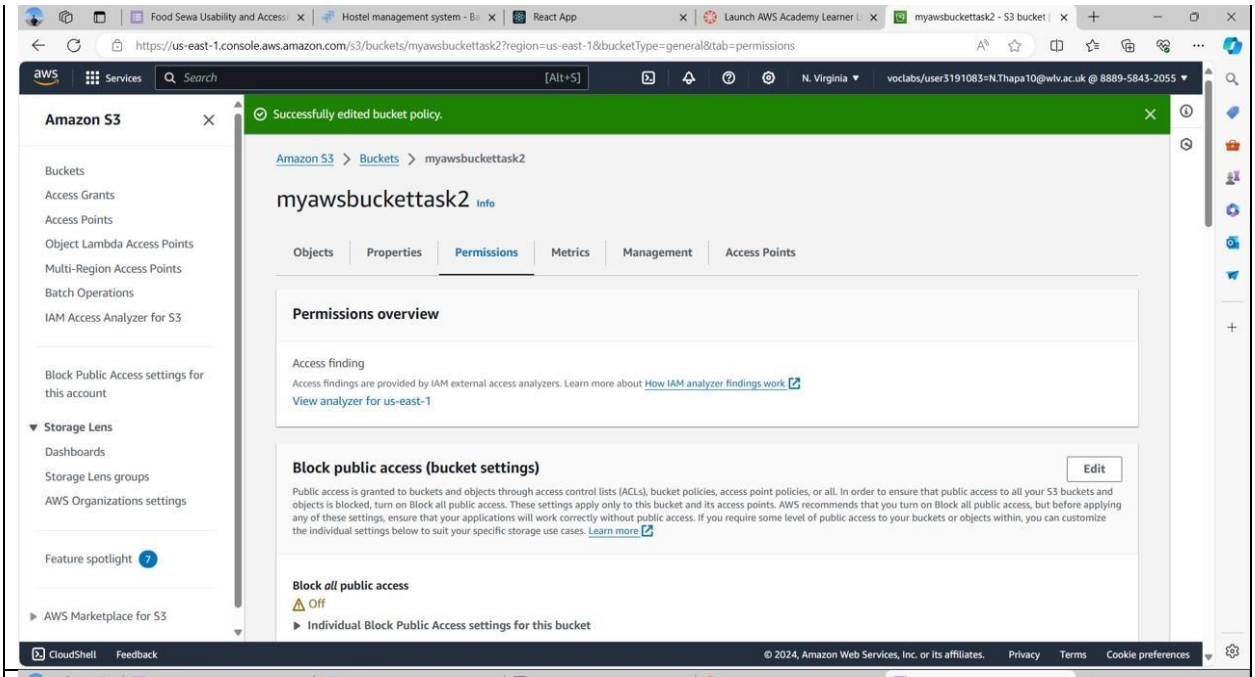

cloudfront
N. Virginia
vocla
Search results for 'cloudfro'
Services (2)
Features (1)
Resources New
Documentation (7,001)
Knowledge Articles (73)
Blogs (303)
Events (2)
Tutorials (1)
Services
CloudFront
Global Content Delivery Network
Athena
Serverless interactive analytics service
Features
Data sources
Athena feature
Resources / for a focused search
Introducing resource search
To search for resources, Resource Explorer must be active in at least one AWS Region and you must have permission to use the default view in the account. Learn more
Dismiss
Documentation
See all 7,001 results
https://us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=us-east-1#/distributions
Services
Search
[Alt+S]
Global
CloudFront
Distributions
Policies
Functions
What's new NEW
Telemetry
Monitoring
Alarms
Logs
Reports & analytics
Cache statistics
Popular objects
Top referrers
Usage
Viewers
Security
Origin access
Field-level encryption
Key management
Distributions (1) Info
Enable
Disable
Delete
Create distribution
Search all distributions
ID
Descrip...
Type
Domain...
Alterna...
Origins
Status
Last mo...
E13U0OIDMEV1FN
Production
d2i8iyzt9a...
niroj776.s3.us-e
Enabled
May 14, 2...
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

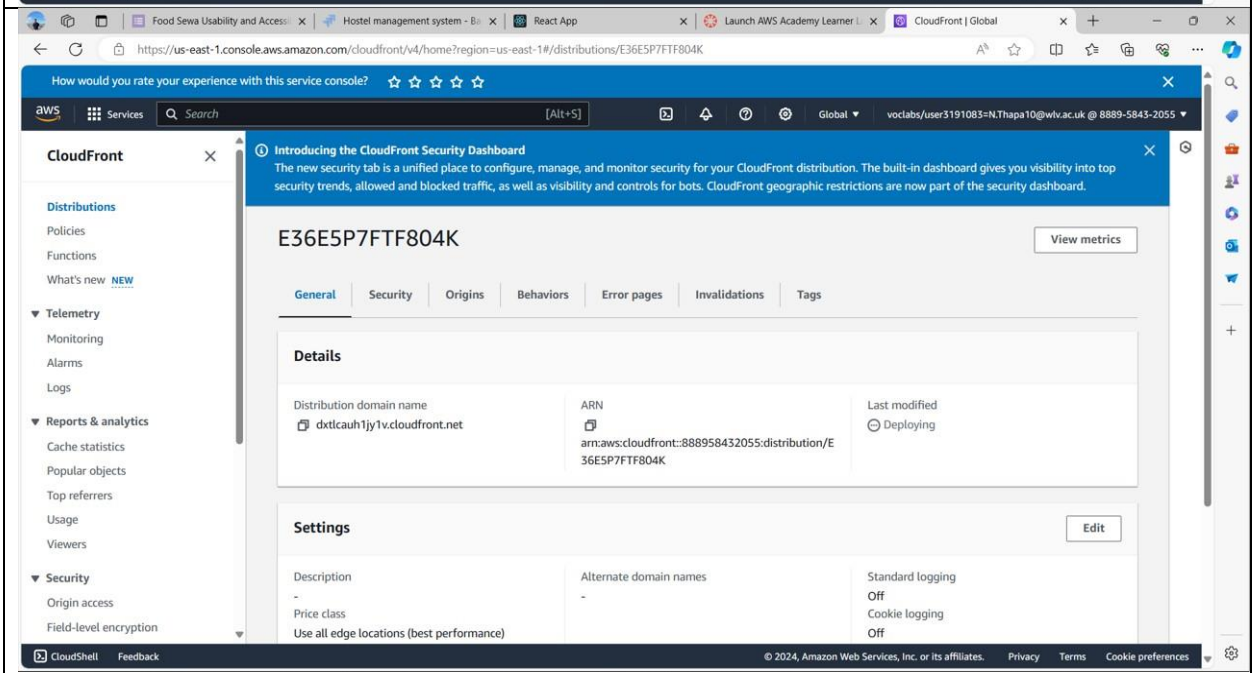

Create new OAC
Name
The name must be unique. Valid characters: letters, numbers and most special characters. Use up to 64 characters.
myawsbuckettask2.s3.us-east-1.amazonaws.com
Description - optional
The description can have up to 256 characters.
Enter description
Signing behavior
Do not sign requests
Sign requests (recommended)
Do not override authorization header
Do not sign if incoming request has authorization header.
Origin type
S3
The origin type must be the same type as origin domain.
Cancel
Create
https://us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=us-east-1#/distributions/create
Origin path - optional
Enter a URL path to append to the origin domain name for origin requests.
Enter the origin path
Name
Enter a name for this origin.
myawsbuckettask2.s3.us-east-1.amazonaws.com
Origin access Info
Public
Bucket must allow public access.
Origin access control settings (recommended)
Bucket can restrict access to only CloudFront.
Legacy access identities
Use a CloudFront origin access identity (OAI) to access the S3 bucket.
Origin access control
Select an existing origin access control (recommended) or create a new control.
myawsbuckettask2.s3.us-east-1.amazonaws.com
Create new OAC
You must update the S3 bucket policy
CloudFront will provide you with the policy statement after creating the distribution.
Add custom header - optional
CloudFront includes this header in all requests that it sends to your origin.
Add header
Enable Origin Shield
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

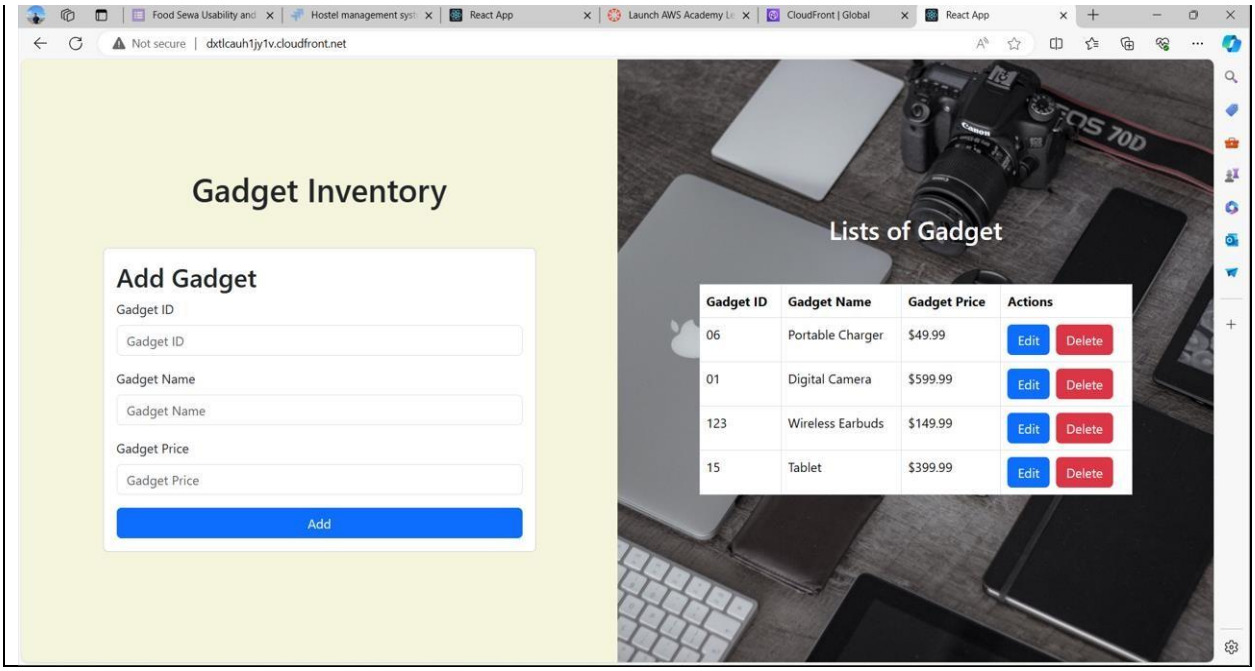

Function type
Function ARN / Name
Include body
Viewer request
No association
Viewer response
No association
Origin request
No association
Origin response
No association
Web Application Firewall (WAF) Info
Enable security protections
Keep your application secure from the most common web threats and security vulnerabilities using AWS WAF. Blocked requests are stopped before they reach your web servers.
Do not enable security protections
Select this option if your application does not need security protections from AWS WAF.
Settings
Price class Info
Choose the price class associated with the maximum price that you want to pay.
Use all edge locations (best performance)
Choose certificate
Request certificate
Supported HTTP versions
Add support for additional HTTP versions. HTTP/1.0 and HTTP/1.1 are supported by default.
HTTP/2
HTTP/3
Default root object - optional
The object (file name) to return when a viewer requests the root URL (/) instead of a specific object.
index.html
Standard logging
Get logs of viewer requests delivered to an Amazon S3 bucket.
Off
On
IPv6
Off
On
Description - optional
Cancel
Create distribution


Introducing the CloudFront Security Dashboard
The new security tab is a unified place to configure, manage, and monitor security for your CloudFront distribution. The built-in dashboard gives you visibility into top security trends, allowed and blocked traffic, as well as visibility and controls for bots. CloudFront geographic restrictions are now part of the security dashboard.
Successfully created new distribution.
The S3 bucket policy needs to be updated
Complete distribution configuration by allowing read access to CloudFront origin access control in your policy statement. Go to S3 bucket permissions to update policy
Copy policy
CloudFront > Distributions > E36E5P7FTF804K
E36E5P7FTF804K
View metrics
General
Security
Origins
Behaviors
Error pages
Invalidations
Tags
Details
Distribution domain name
dxtlcauh1jy1v.cloudfront.net
ARN
arn:aws:cloudfront::888958432055:distribution/E36E5P7FTF804K
Last modified
Deploying
Settings
Edit


Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight
AWS Marketplace for S3
Amazon S3 > Buckets > myawsbuckettask2 > Edit bucket policy
Edit bucket policy
Bucket policy
Policy examples
Policy generator
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts.
Bucket ARN
arn:aws:s3:::myawsbuckettask2
Policy
Edit statement
Select a statement
Select an existing statement in the policy or add a new statement.
Add new statement

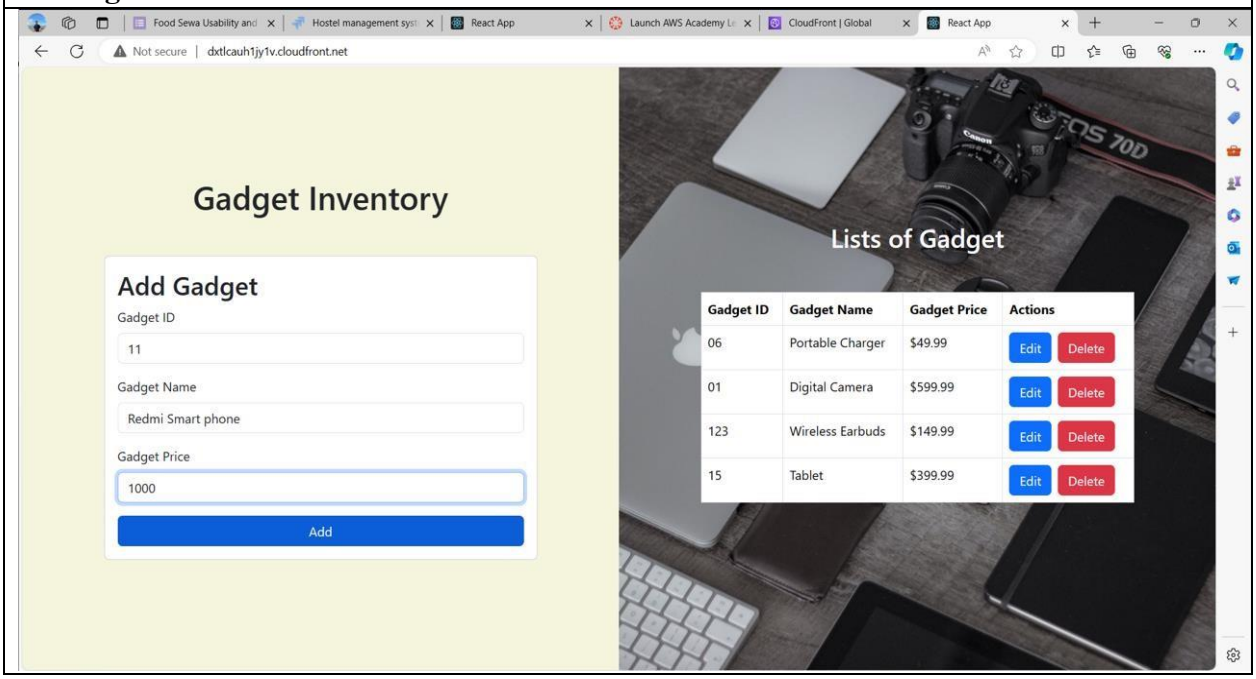

Amazon S3
Successfully edited bucket policy.
Amazon S3 > Buckets > myawsbuckettask2
myawsbuckettask2 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Permissions overview
Access finding
Block public access (bucket settings)
Edit
Block all public access
Off
Individual Block Public Access settings for this bucket

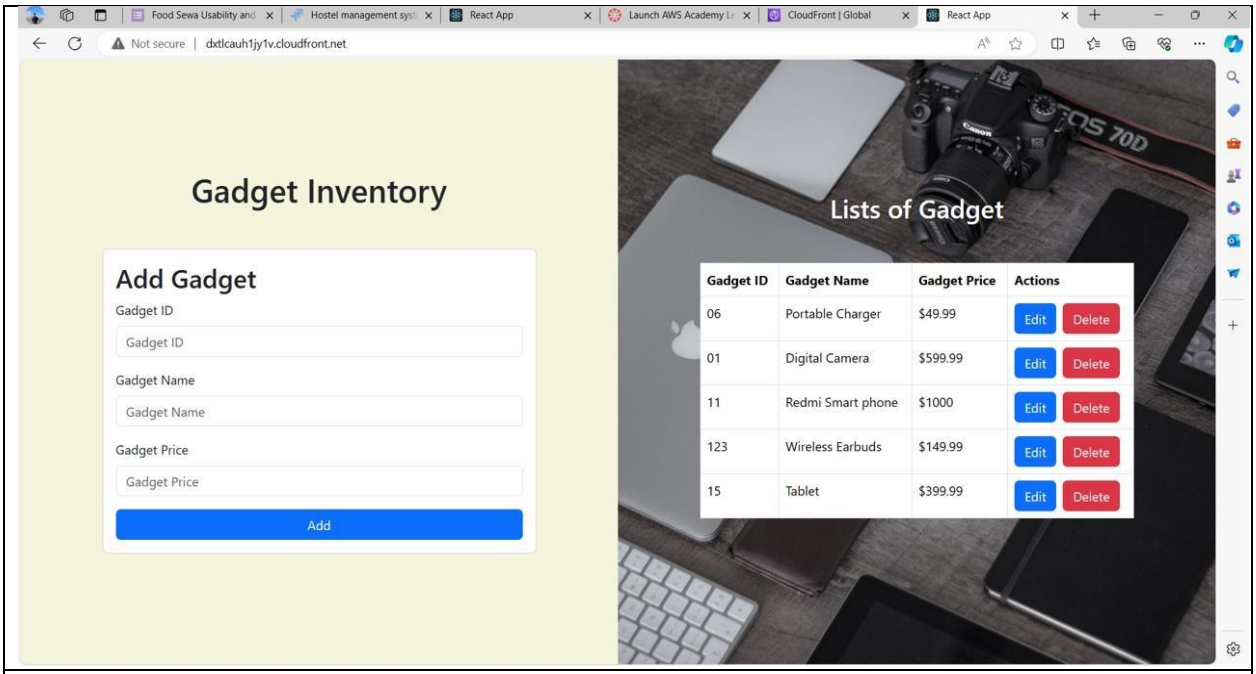

CloudFront
Introducing the CloudFront Security Dashboard
E36E5P7FTF804K
View metrics
General
Security
Origins
Behaviors
Error pages
Invalidations
Tags
Details
Distribution domain name
dxtlcauh1jy1v.cloudfront.net
ARN
arn:aws:cloudfront::888958432055:distribution/E36E5P7FTF804K
Last modified
Deploying
Settings
Edit
Description
Price class
Use all edge locations (best performance)
Alternate domain names
Standard logging
Off
Cookie logging
Off



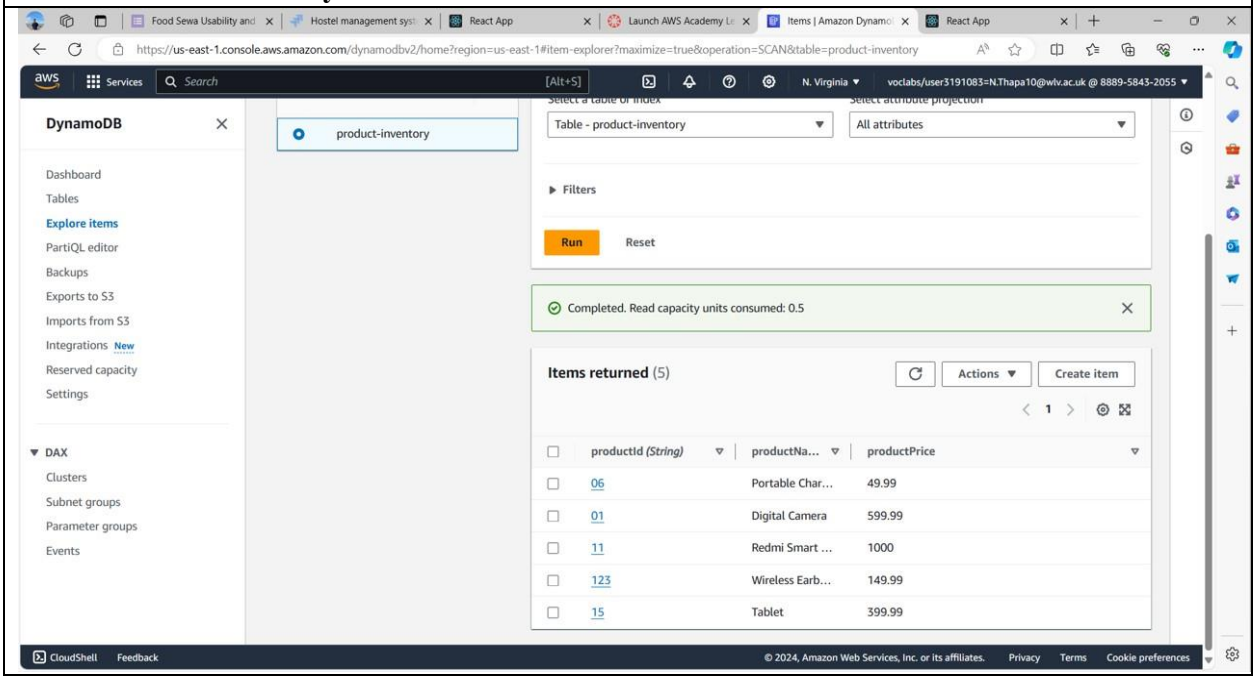
2.6 Part 6: Testing (add, Edit and delete) and checking in AWS DynamoDB

Adding:

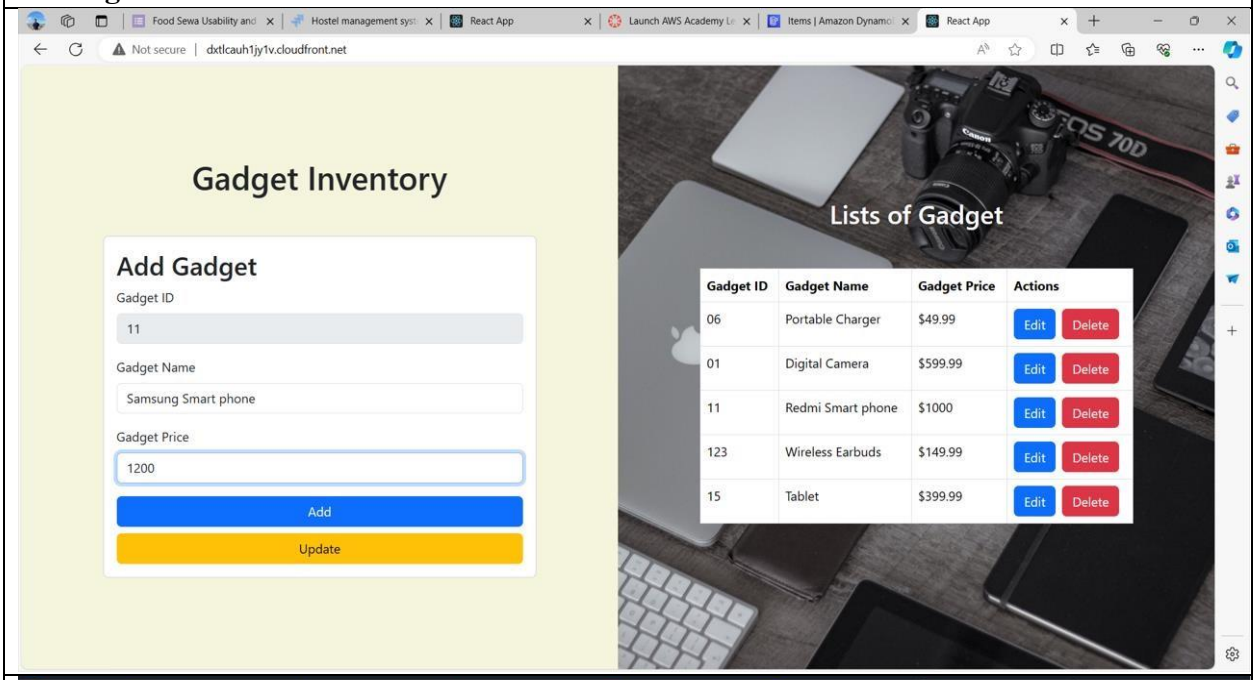


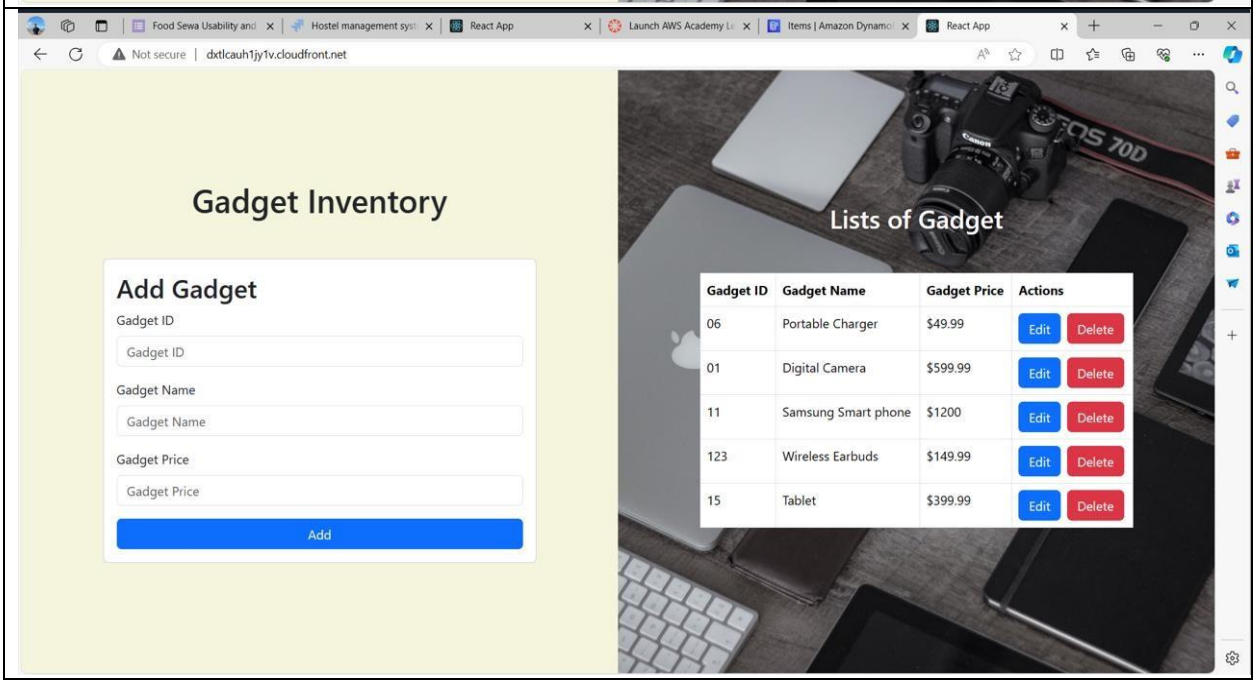


Added Item in AWS DynamoDB:

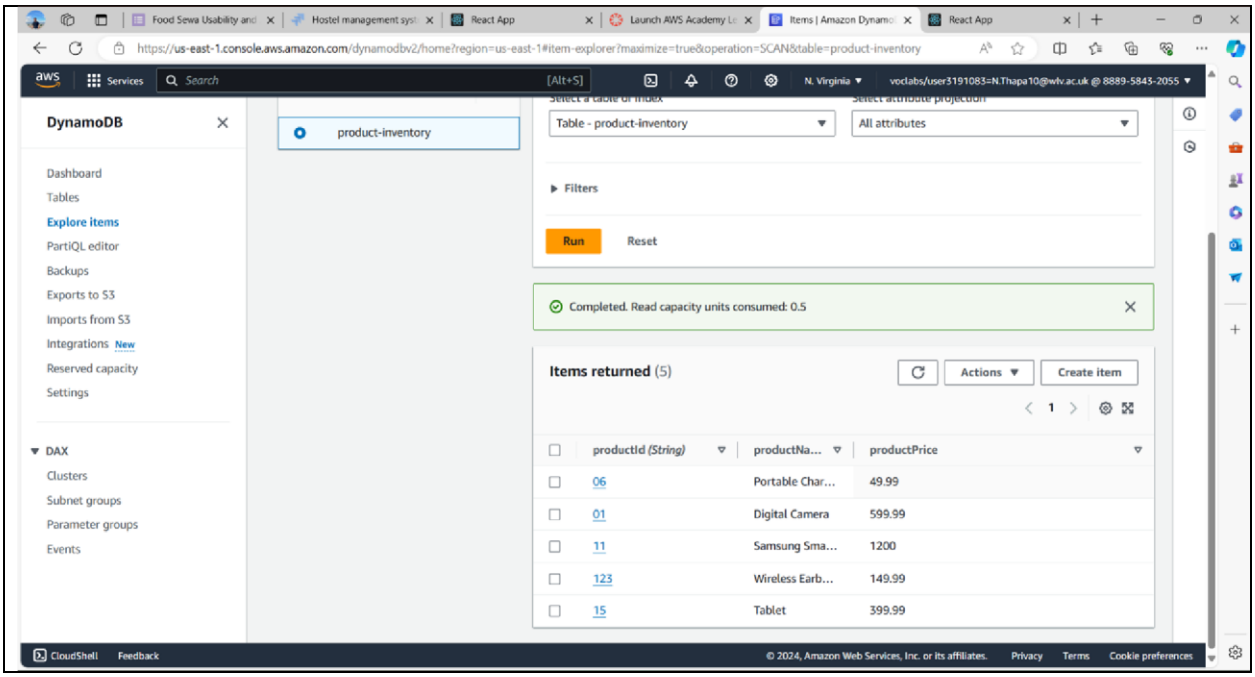


Editing:

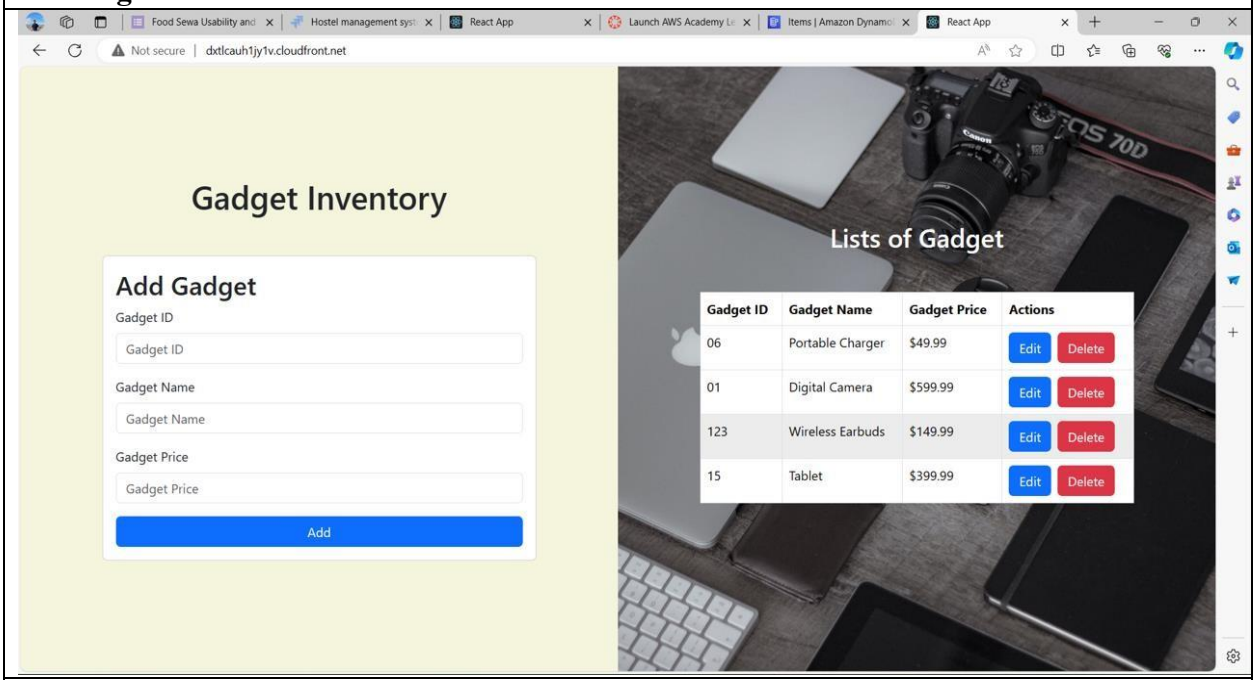




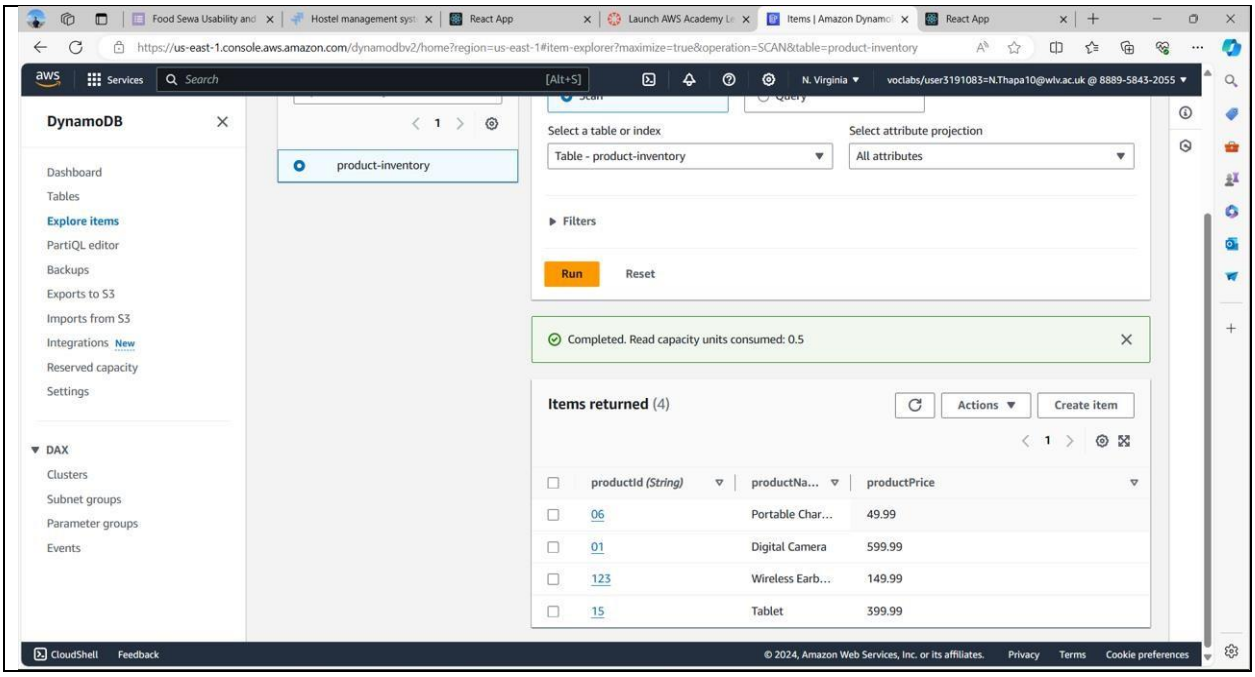
Edited Item in AWS DynamoDB:



Deleting:



Deleted Item in AWS DynamoDB:


DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations New
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
product-inventory
Select a table or index
Table - product-inventory
Select attribute projection
All attributes
Filters
Run
Reset
Completed. Read capacity units consumed: 0.5
Items returned (4)
Actions
Create item
productId (String)
productNa...
productPrice
06
Portable Char...
49.99
01
Digital Camera
599.99
123
Wireless Earb...
149.99
15
Tablet
399.99
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

3 link of AWS CloudFront Services after Hosting:

Link: React App (dxtlcauh1jy1v.cloudfront.net)

4 References

Anon., n.d. *AWS Developer Center.* [Online] Available at: https://aws.amazon.com/developer/ [Accessed 14 05 2024].

Anon., n.d. *Geeks for Geeks.* [Online] Available at: https://www.geeksforgeeks.org/introduction-to-aws-elastic-beanstalk/ [Accessed 14 05 2024].