

Day 1 Assignment Lab

IAAS:

1- Creating EC2

Success
Successfully initiated launch of instance (i-07ca4bf3663d65ec6)

Launch log

Next Steps
What would you like to do next with this instance, for example "create alarm" or "create backup"

- Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.
[Create billing alerts](#)
- Connect to your instance
Once your instance is running, log into it from your local computer.
[Connect to instance](#)
[Learn more](#)
- Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
[Connect an RDS database](#)
[Create a new RDS database](#) [Learn more](#)
- Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots
[Create EBS snapshot policy](#)
- Manage detailed monitoring
Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period.
[Manage detailed monitoring](#)
- Create Load Balancer
Create an application, network gateway or classic Elastic Load Balancer
[Create Load Balancer](#)
- Create AWS budget
AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location.
[Create AWS budget](#)
- Manage CloudWatch alarms
Create or update Amazon CloudWatch alarms for the instance.
[Manage CloudWatch alarms](#)

[View all instances](#)

2- Install web server as root user

Connect to instance | EC2 | us-east-1 console.aws.amazon.com/ec2-instance-connect/shh?contentType=standard&instanceId=i-07ca4bf3663d65ec6&user=ec2-user®ion=us-east-1&sshPort=22#

```
10/12: mailcap-2.1.49-3.amzn2023.0.3.noarch.rpm      1.8 MB/s | 33 kB  00:00
11/12: httpd-core-2.4.56-1.amzn2023.x86_64.rpm      17 MB/s | 1.4 MB  00:00
12/12: generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm  421 kB/s | 19 kB  00:00
total                                              9.1 MB/s | 2.3 MB  00:00

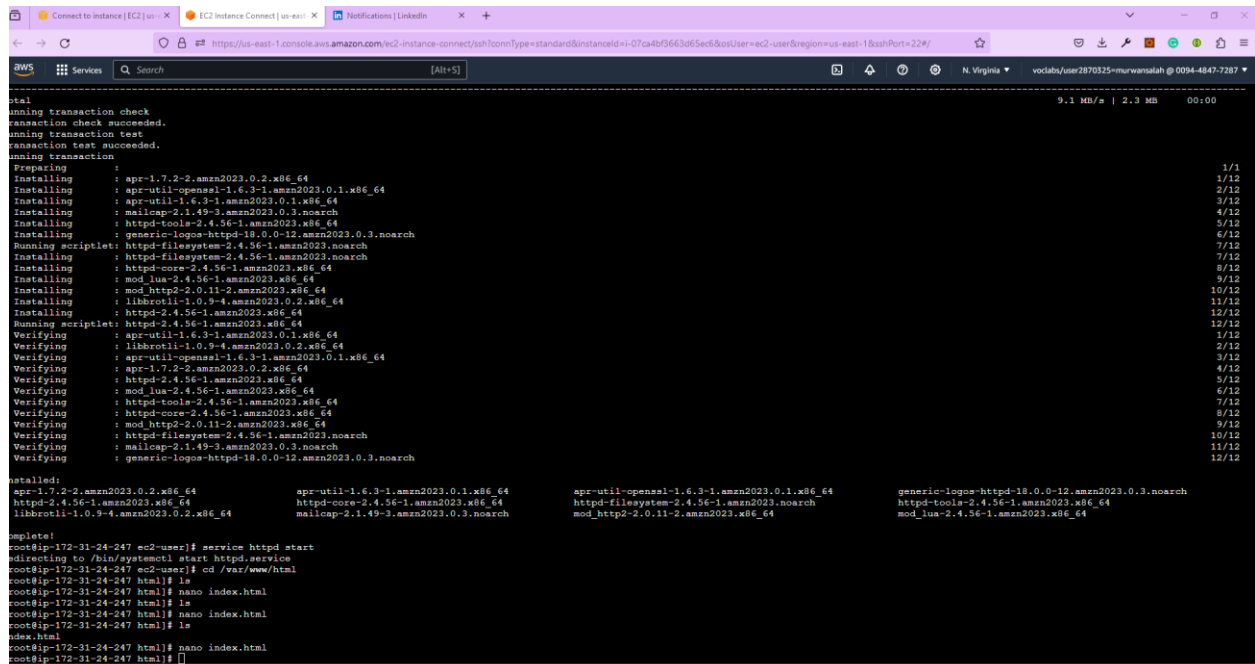
uninstall transaction check
transaction check succeeded.
uninstall transaction test
transaction test succeeded.
uninstall transaction
Preparing
Installing : apr-1.7.2-2.amzn2023.0.2.x86_64      1/1
Installing : apr-util-1.6.3-1.amzn2023.0.1.x86_64  1/12
Installing : apr-util-1.6.3-1.amzn2023.0.1.x86_64  2/12
Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch  3/12
Installing : httpd-tools-2.4.56-1.amzn2023.x86_64  4/12
Installing : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch  5/12
Running scriptlet: httpd-filesystem-2.4.56-1.amzn2023.noarch  6/12
Installing : httpd-filesystem-2.4.56-1.amzn2023.noarch  7/12
Installing : httpd-core-2.4.56-1.amzn2023.x86_64  8/12
Installing : mod_lua-2.4.56-1.amzn2023.x86_64      9/12
Installing : mod_http2-2.0.11-2.amzn2023.x86_64   10/12
Installing : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 11/12
Installing : httpd-2.4.56-1.amzn2023.x86_64      12/12
Running scriptlet: httpd-2.4.56-1.amzn2023.x86_64 12/12
Verifying : apr-util-1.6.3-1.amzn2023.0.1.x86_64  1/12
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64  2/12
Verifying : apr-1.7.2-2.amzn2023.0.2.x86_64      3/12
Verifying : httpd-core-2.4.56-1.amzn2023.x86_64  4/12
Verifying : mod_lua-2.4.56-1.amzn2023.x86_64      5/12
Verifying : httpd-tools-2.4.56-1.amzn2023.x86_64  6/12
Verifying : httpd-core-2.4.56-1.amzn2023.x86_64  7/12
Verifying : mod_http2-2.0.11-2.amzn2023.x86_64   8/12
Verifying : httpd-filesystem-2.4.56-1.amzn2023.noarch  9/12
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch 10/12
Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 11/12
Verifying : httpd-2.4.56-1.amzn2023.x86_64      12/12

Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64      apr-util-1.6.3-1.amzn2023.0.1.x86_64      apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64      generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-2.4.56-1.amzn2023.x86_64      httpd-core-2.4.56-1.amzn2023.x86_64      httpd-filesystem-2.4.56-1.amzn2023.noarch      httpd-tools-2.4.56-1.amzn2023.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64  mailcap-2.1.49-3.amzn2023.0.3.noarch      mod_http2-2.0.11-2.amzn2023.x86_64      mod_lua-2.4.56-1.amzn2023.x86_64

complete!
root@ip-172-31-24-247 ec2-user#
```

i-07ca4bf3663d65ec6 (My first server)
PublicIP: 54.225.34.200 PrivateIP: 172.31.24.247

3- Create an HTML file



```
ptal
unning transaction check
ransaction check succeeded.
unning transaction test
ransaction test succeeded.
unning transaction
Preparing
Installing : apr-1.7.2-2.amzn2023.0.2.x86_64 1/1
Installing : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 1/12
Installing : apr-util-1.6.3-1.amzn2023.0.1.x86_64 2/12
Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch 3/12
Installing : httpd-tools-2.4.56-1.amzn2023.x86_64 4/12
Installing : generic-logger-httpd-18.0.0-12.amzn2023.0.3.noarch 5/12
Running scriptlet: httpd-filesystem-2.4.56-1.amzn2023.noarch 6/12
Installing : httpd-filesystem-2.4.56-1.amzn2023.noarch 7/12
Installing : httpd-core-2.4.56-1.amzn2023.x86_64 7/12
Installing : mod_lua-2.4.56-1.amzn2023.x86_64 8/12
Installing : mod_http2-2.0.11-2.amzn2023.x86_64 9/12
Installing : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 10/12
Installing : httpd-2.4.56-1.amzn2023.x86_64 11/12
Running scriptlet: httpd-2.4.56-1.amzn2023.x86_64 12/12
Verifying : apr-util-1.6.3-1.amzn2023.0.1.x86_64 1/12
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 2/12
Verifying : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 3/12
Verifying : apr-1.7.2-2.amzn2023.0.2.x86_64 4/12
Verifying : httpd-2.4.56-1.amzn2023.x86_64 5/12
Verifying : mod_lua-2.4.56-1.amzn2023.x86_64 6/12
Verifying : httpd-tools-2.4.56-1.amzn2023.x86_64 7/12
Verifying : httpd-core-2.4.56-1.amzn2023.x86_64 8/12
Verifying : mod_http2-2.0.11-2.amzn2023.x86_64 9/12
Verifying : httpd-filesystem-2.4.56-1.amzn2023.noarch 10/12
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch 11/12
Verifying : generic-logger-httpd-18.0.0-12.amzn2023.0.3.noarch 12/12

Installed!
apr-1.7.2-2.amzn2023.0.2.x86_64      apr-util-1.6.3-1.amzn2023.0.1.x86_64      apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64      generic-logger-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-2.4.56-1.amzn2023.x86_64      httpd-core-2.4.56-1.amzn2023.x86_64      httpd-filesystem-2.4.56-1.amzn2023.noarch      httpd-tools-2.4.56-1.amzn2023.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64      mailcap-2.1.49-3.amzn2023.0.3.noarch      mod_http2-2.0.11-2.amzn2023.x86_64      mod_lua-2.4.56-1.amzn2023.x86_64

complete!
root@ip-172-31-24-247 ec2-user]# service httpd start
Redirecting to /bin/systemctl start httpd.service
root@ip-172-31-24-247 ec2-user]# cd /var/www/html
root@ip-172-31-24-247 html]# ls
root@ip-172-31-24-247 html]# nano index.html
root@ip-172-31-24-247 html]# ls
root@ip-172-31-24-247 html]# nano index.html
root@ip-172-31-24-247 html]# ls
root@ip-172-31-24-247 html]# nano index.html
root@ip-172-31-24-247 html]# ls
root@ip-172-31-24-247 html]#
```

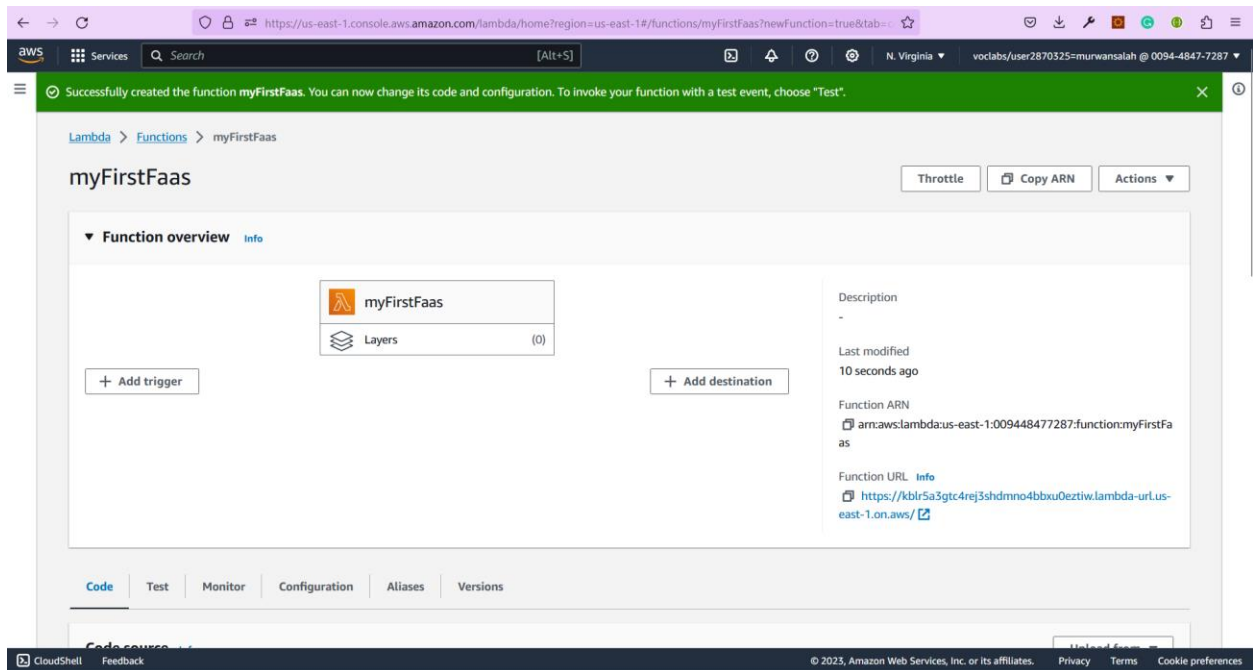
4- Web page in EC2 that shows my name.



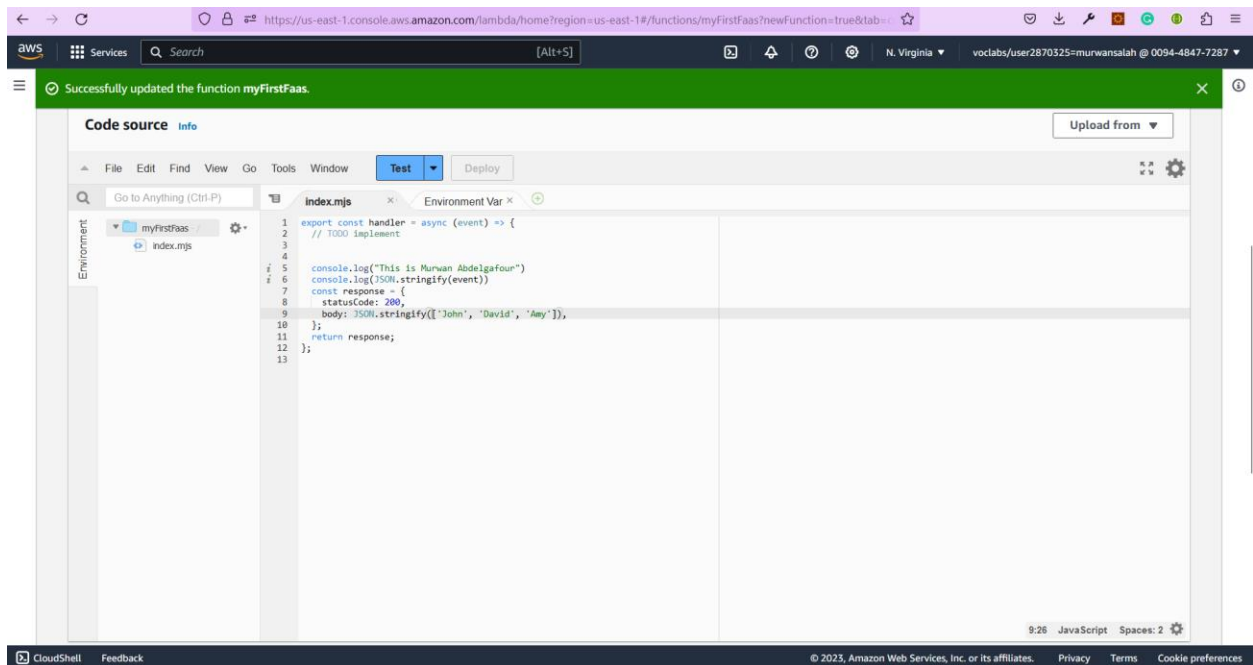
My name is Murwan Abdelfaour

Faas:

1- Creating Lambda function:



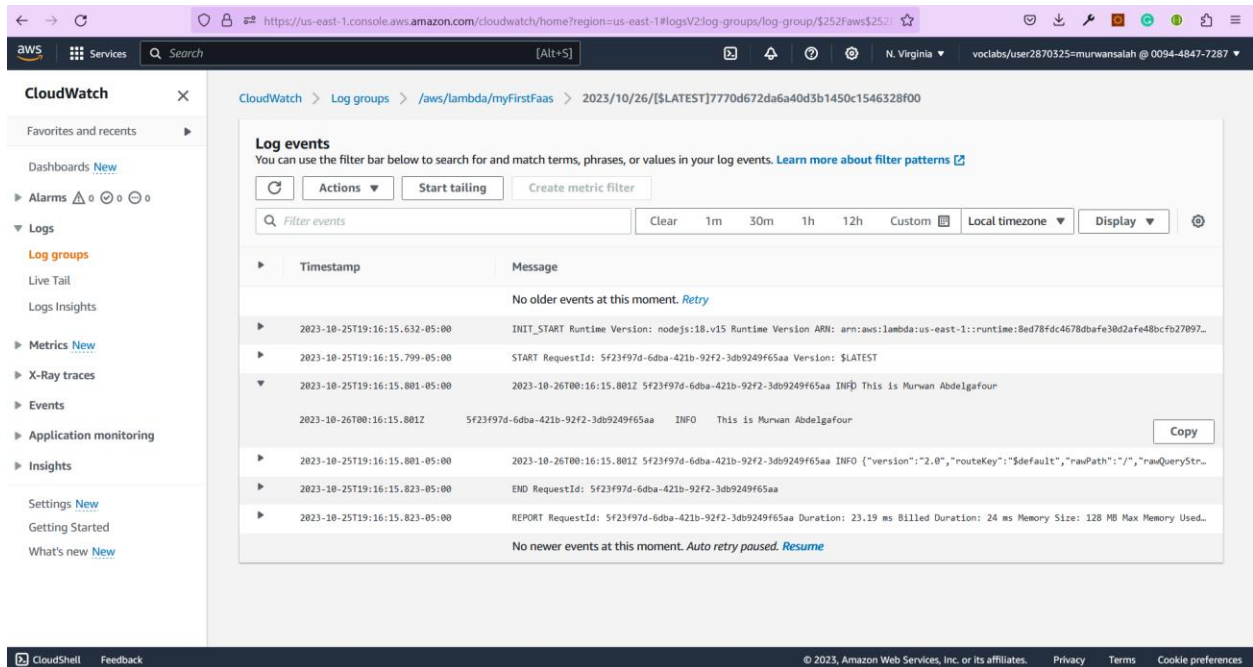
2- Write node app:



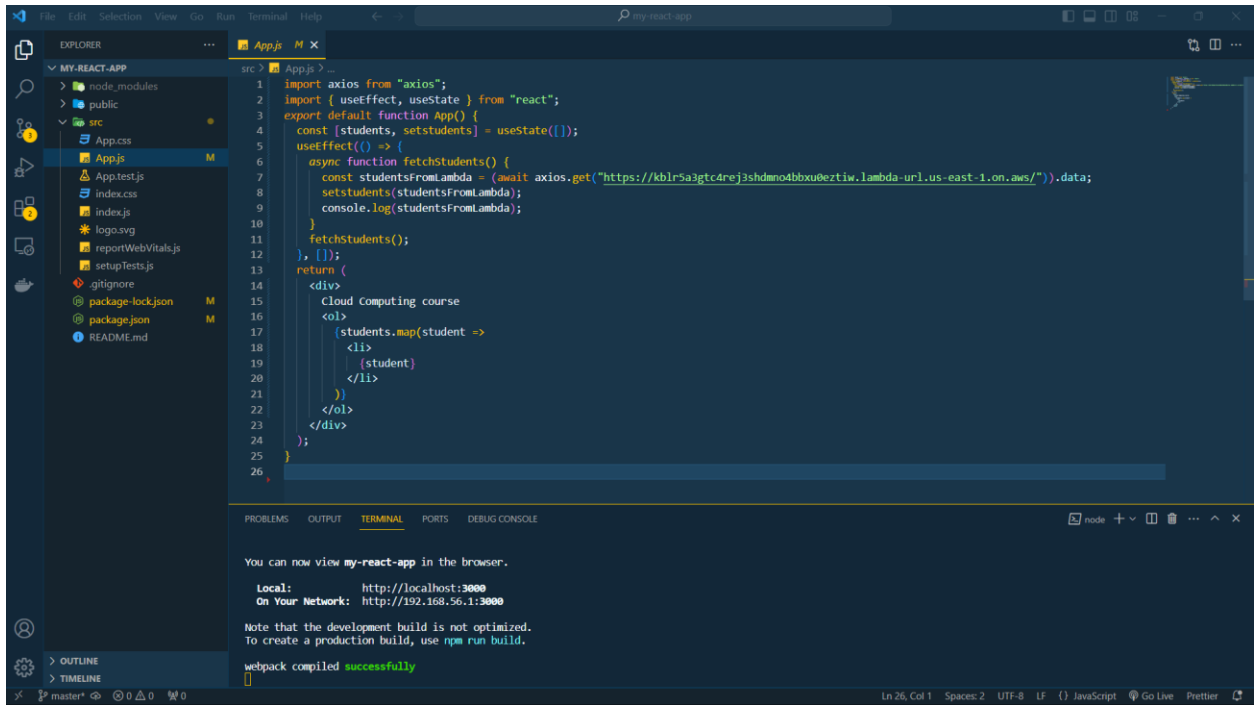
3- After clicking on the function URL, the method gets triggered.



4- Logs:



5- React calling AWS FAAS



The screenshot shows a Visual Studio Code editor window with a React application. The Explorer panel on the left shows the project structure: MY-REACT-APP, node_modules, public, src, App.css, App.js, App.test.js, index.css, index.js, logo.svg, reportWebVitals.js, setupTests.js, .gitignore, package-lock.json, package.json, and README.md. The App.js file is open in the editor, showing the following code:

```
1 import axios from "axios";
2 import { useEffect, useState } from "react";
3 export default function App() {
4   const [students, setstudents] = useState([]);
5   useEffect(() => {
6     async function fetchStudents() {
7       const studentsFromLambda = (await axios.get("https://kblr5a3gtc4rej3shdmno4bbxudeztw.lambda-url.us-east-1.on.aws/")).data;
8       setstudents(studentsFromLambda);
9       console.log(studentsFromLambda);
10    }
11    fetchStudents();
12  }, []);
13  return (
14    <div>
15      Cloud Computing course
16      <ol>
17        {students.map(student =>
18          <li>
19            {student}
20          </li>
21        )}
22      </ol>
23    </div>
24  );
25 }
```

The TERMINAL panel at the bottom shows the following output:

```
You can now view my-react-app in the browser.

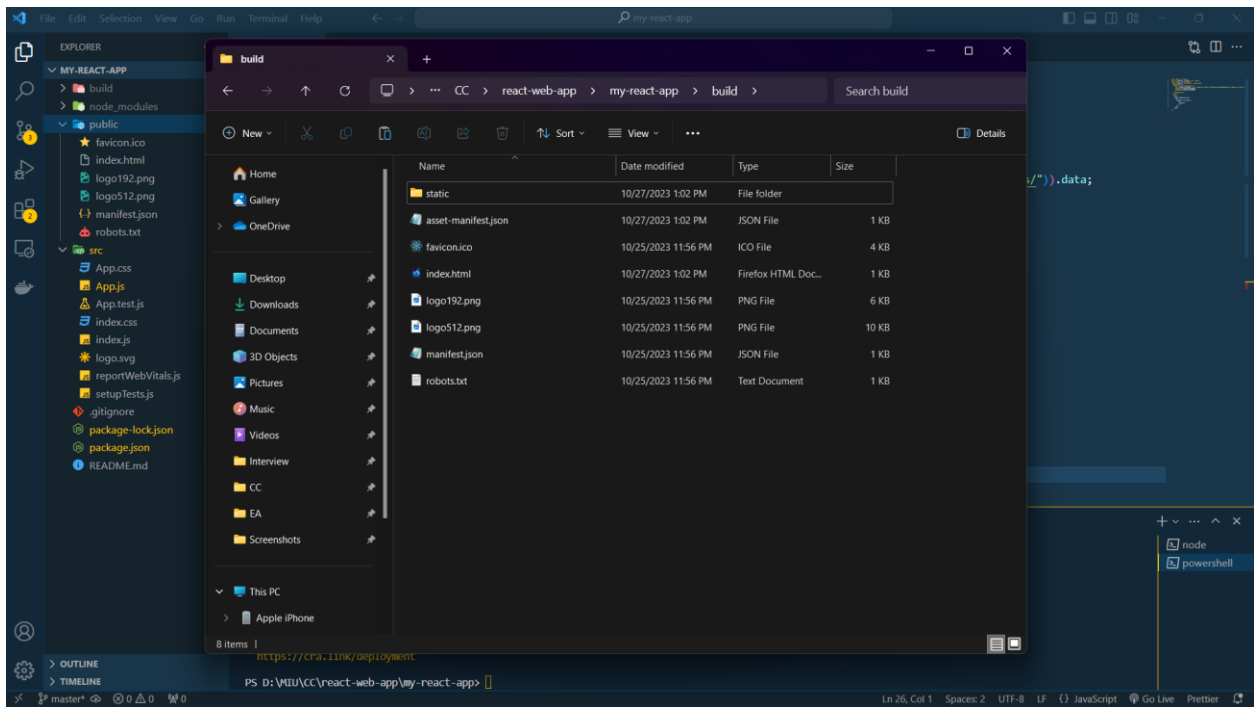
Local:      http://localhost:3000
On Your Network: http://192.168.56.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

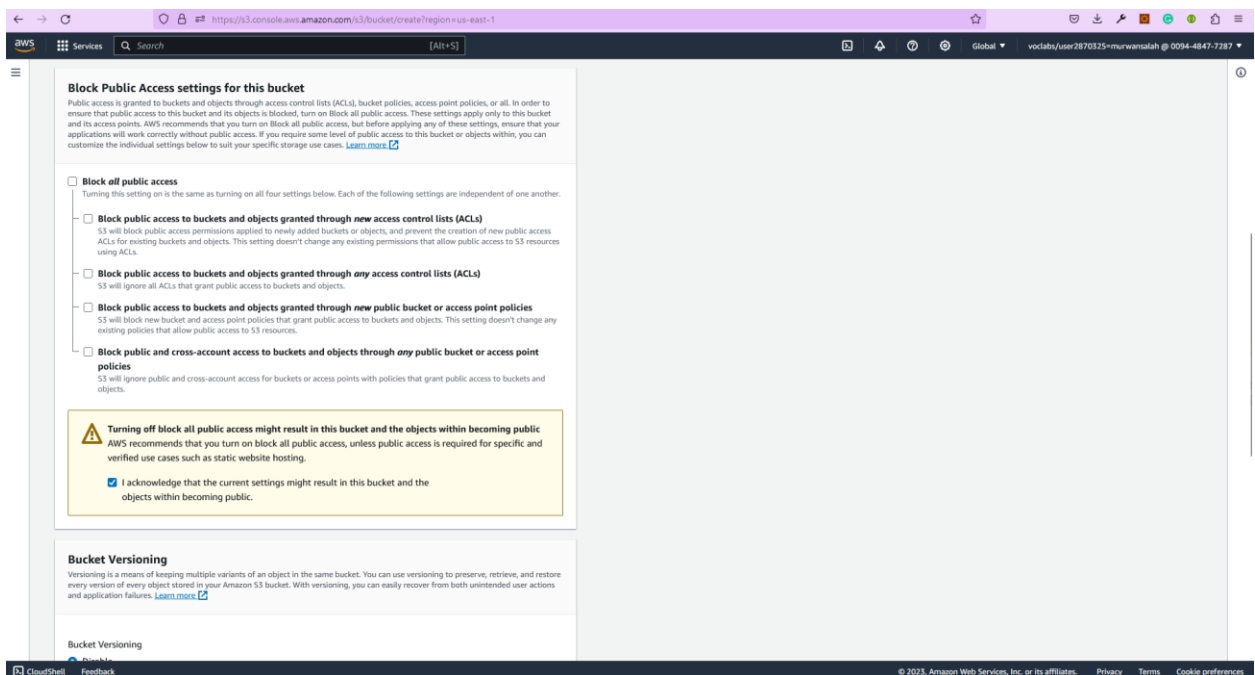
webpack compiled successfully
```



6- Building the react app (npm run build)



7- Creating S3 bucket



8- Bucket policy

The screenshot shows the AWS Management Console interface for editing a bucket policy. A green banner at the top indicates 'Successfully edited bucket policy.' The left sidebar shows the 'Amazon S3' navigation menu. The main content area is titled 'Bucket policy' and contains a JSON policy document. Below the policy, there is an 'Object Ownership' section.

Bucket policy

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. [Learn more](#)

```
{
  "Version": "2012-10-17",
  "Id": "Policy1650912821527",
  "Statement": [
    {
      "Sid": "Stmt1650912820312",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3::murwan-first-bucket/*"
    }
  ]
}
```

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.

Object Ownership
Bucket owner enforced

9- Upload react app files

The screenshot shows the AWS Management Console interface for uploading files to a bucket. The 'Upload' section includes a drag-and-drop area, a list of files and folders, and a destination selection.

Upload [Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 1600B, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose [Add files](#) or [Add folder](#).

Files and folders (14 Total, 745.4 KB) [Remove](#) [Add files](#) [Add folder](#)

All files and folders in this table will be uploaded.

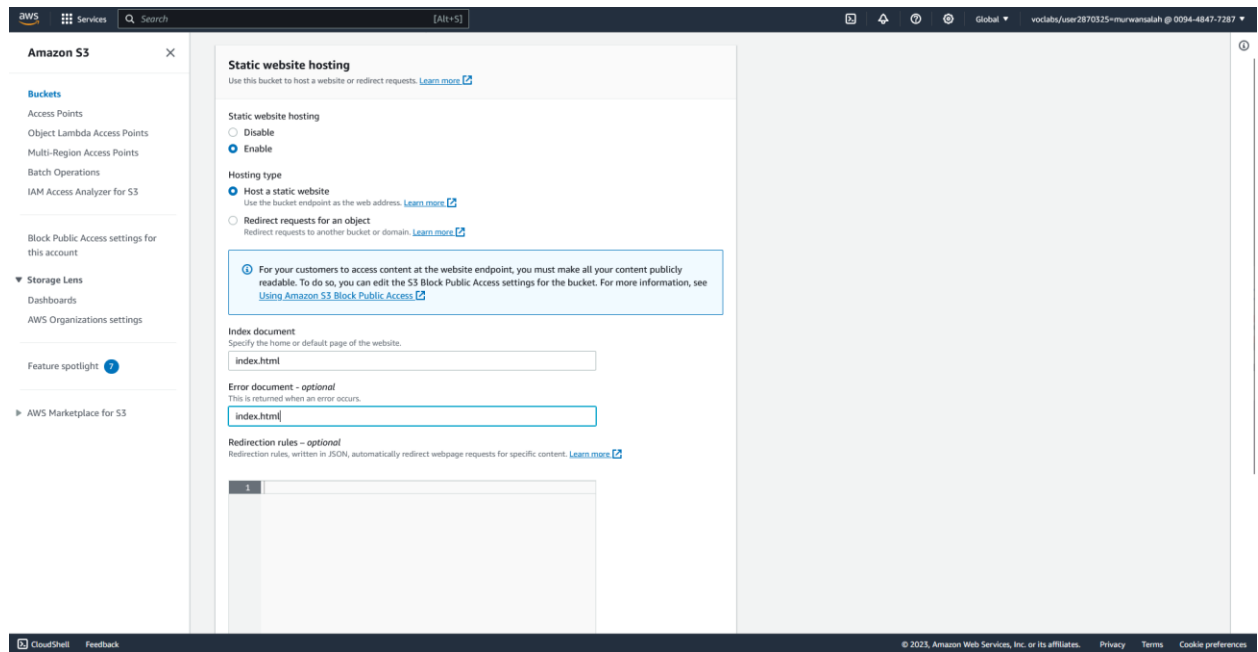
<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	asset-manifest.json	-	application/json	517.0 B
<input type="checkbox"/>	favicon.ico	-	image/x-icon	3.8 KB
<input type="checkbox"/>	index.html	-	text/html	644.0 B
<input type="checkbox"/>	logo192.png	-	image/png	5.2 KB
<input type="checkbox"/>	logo512.png	-	image/png	9.4 KB
<input type="checkbox"/>	manifest.json	-	application/json	492.0 B
<input type="checkbox"/>	robots.txt	-	text/plain	67.0 B
<input type="checkbox"/>	main.e6c13ad2.css	static/css/	-	337.0 B
<input type="checkbox"/>	main.e6c13ad2.css....	static/css/	-	584.0 B
<input type="checkbox"/>	787.bc7d429a.chunk.js	static/js/	-	4.5 KB

Destination

Destination
[s3://murwan-first-bucket](#)

[Destination details](#)

10- Change static website hosting properties



11- Final result:

