

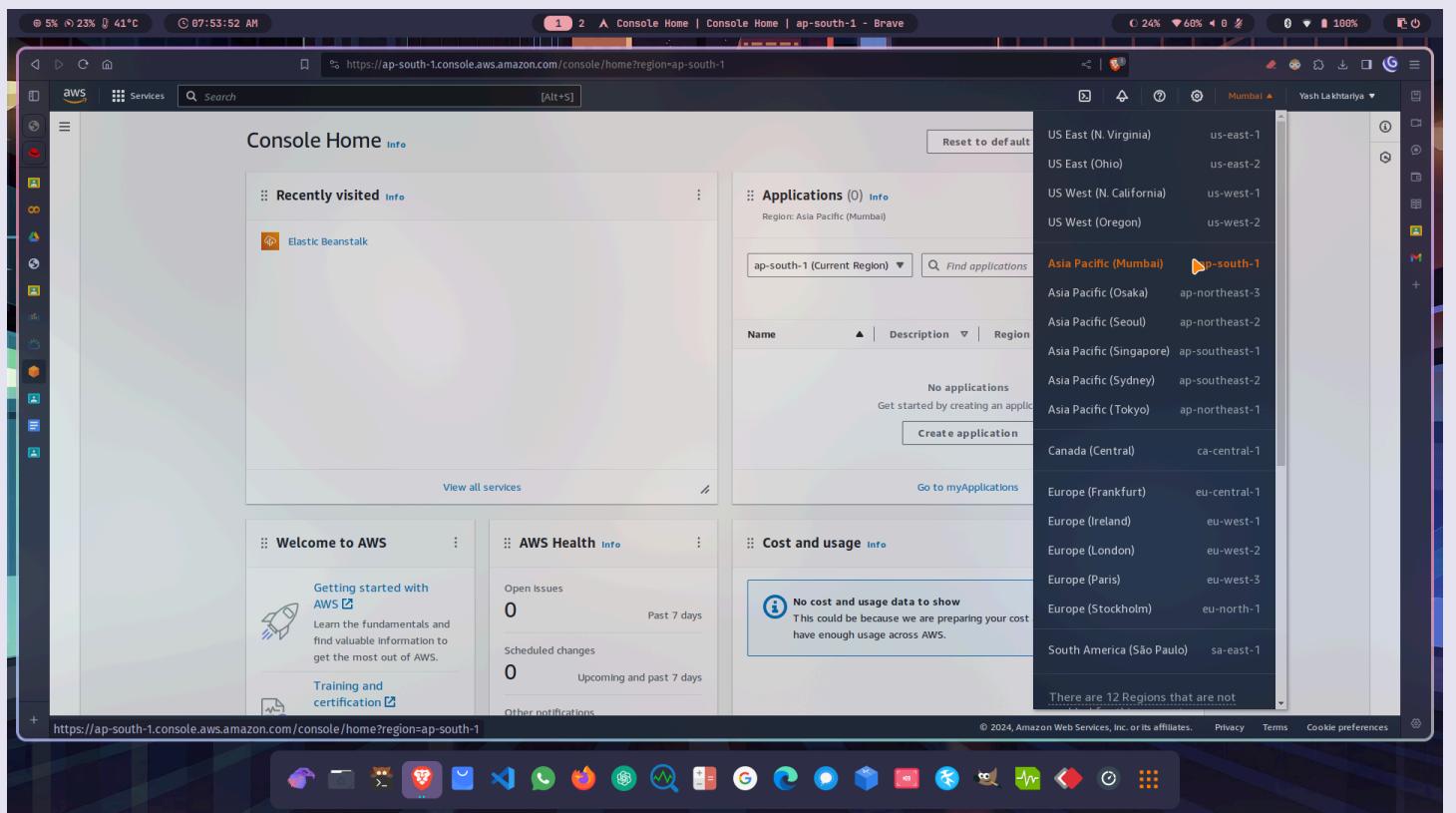
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1

**Aim :** You have a requirement for deploying an existing NodeJS-based application to AWS Cloud. There is a need for automatic scaling for the underlying environment. Implement the AWS Cloud service and resources used to deploy this environment in the quickest way possible.

### Exercises :

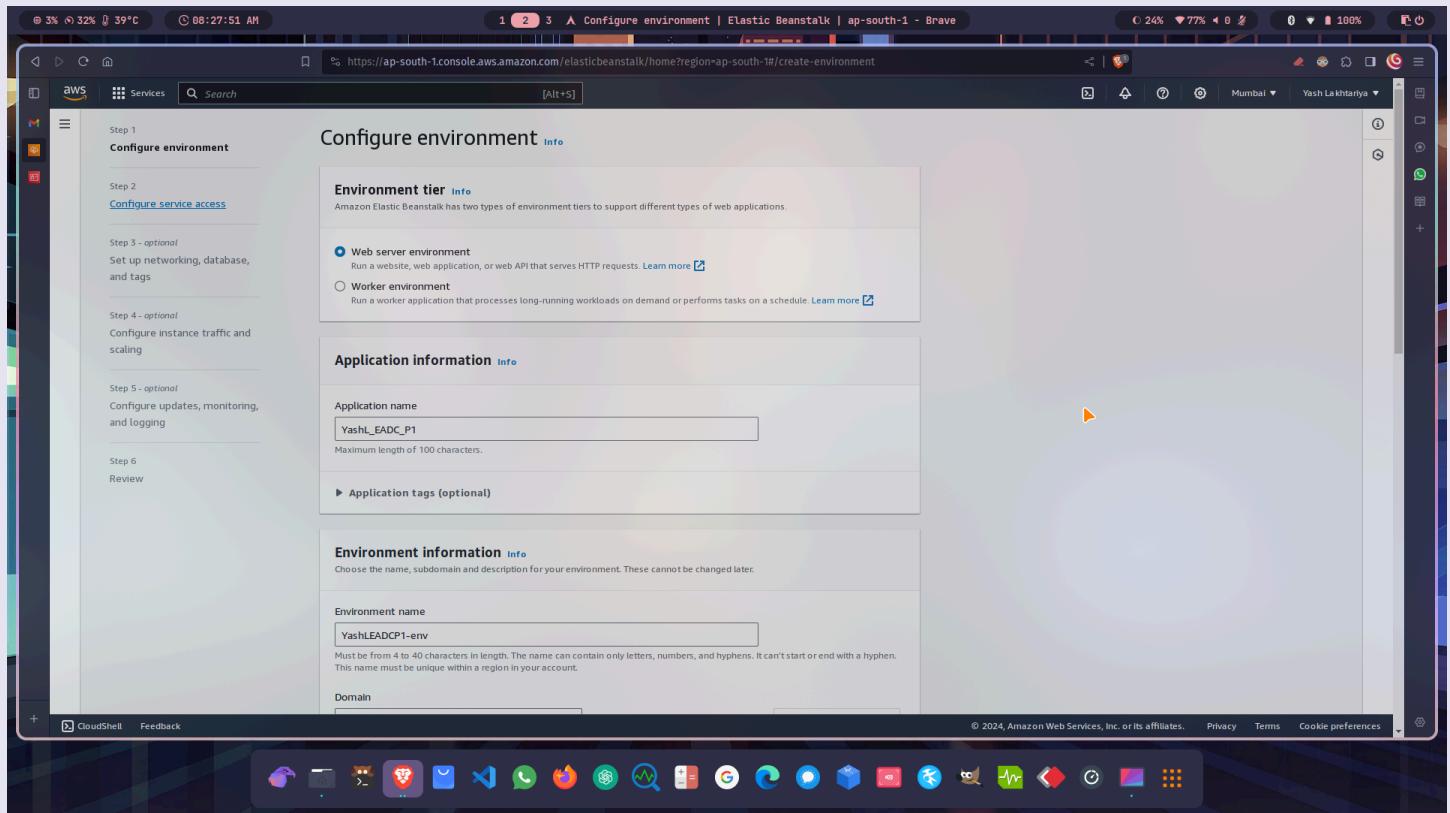
**Practical 1.1 : Developing NodeJS application to create any website GUI (use HTML, CSS, JavaScript) using AWS Cloud and AWS Elastic beanstalk Sample GUI should contain two buttons, with one button to turn ON the bulb provided in the user interface and a second button to turn OFF the bulb.**

1. Login to AWS Console and select near preferred region for reliability (Mumbai here)



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1

## 2. Select Elastic Beanstalk from services, and configure it



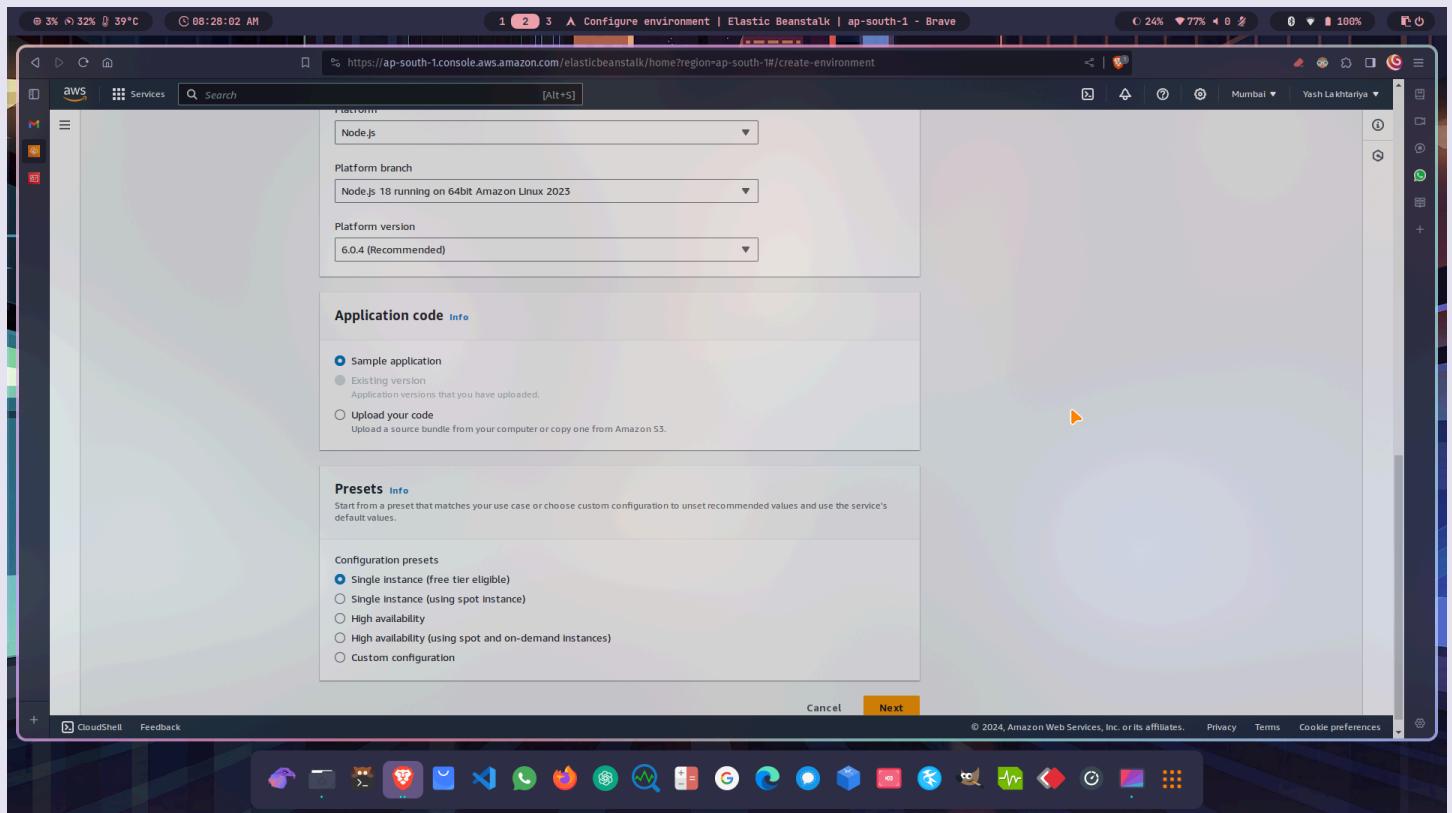
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 1

3. Select Node.js as platform with default options and recommended version, then select Sample Application on a Single Instance (free tier eligible)



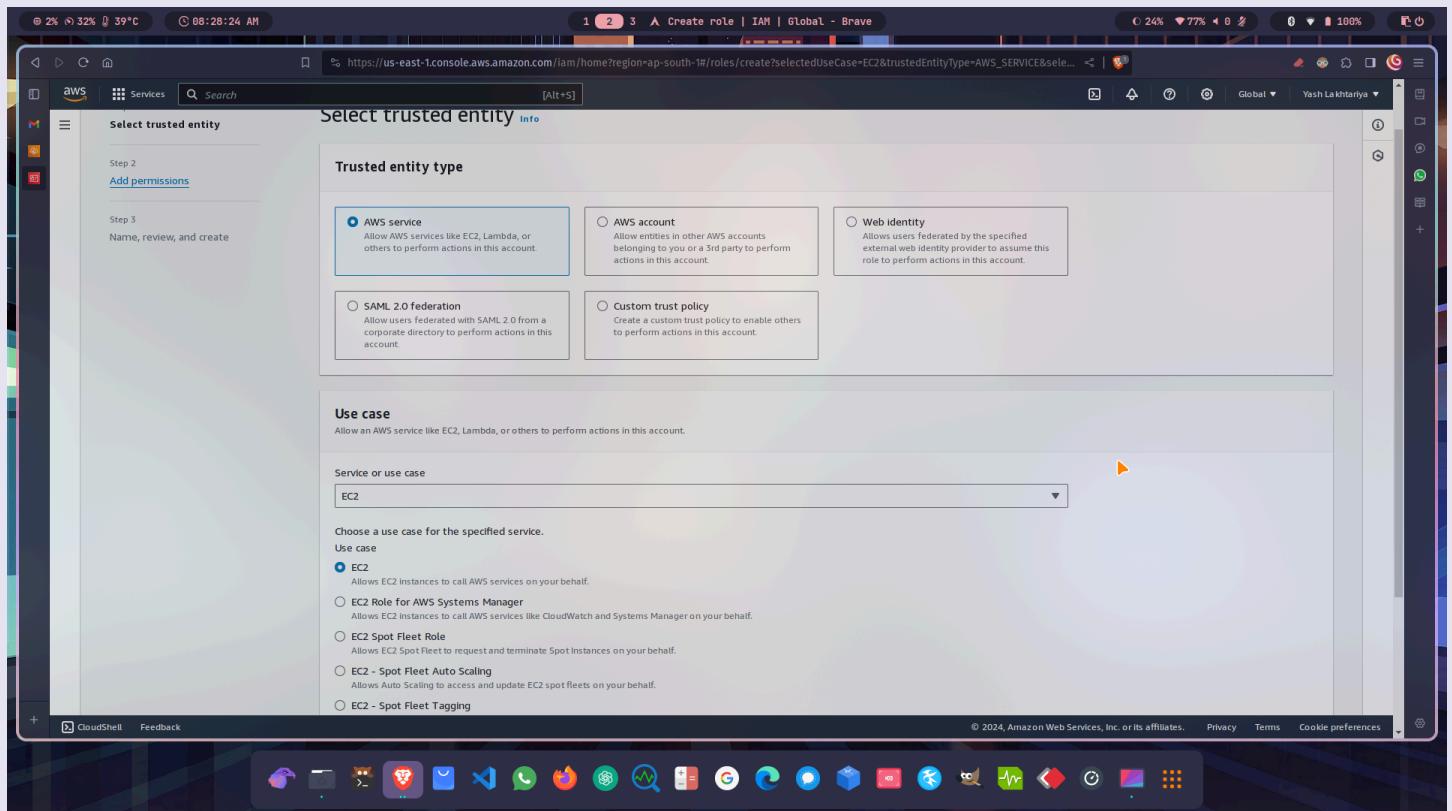
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

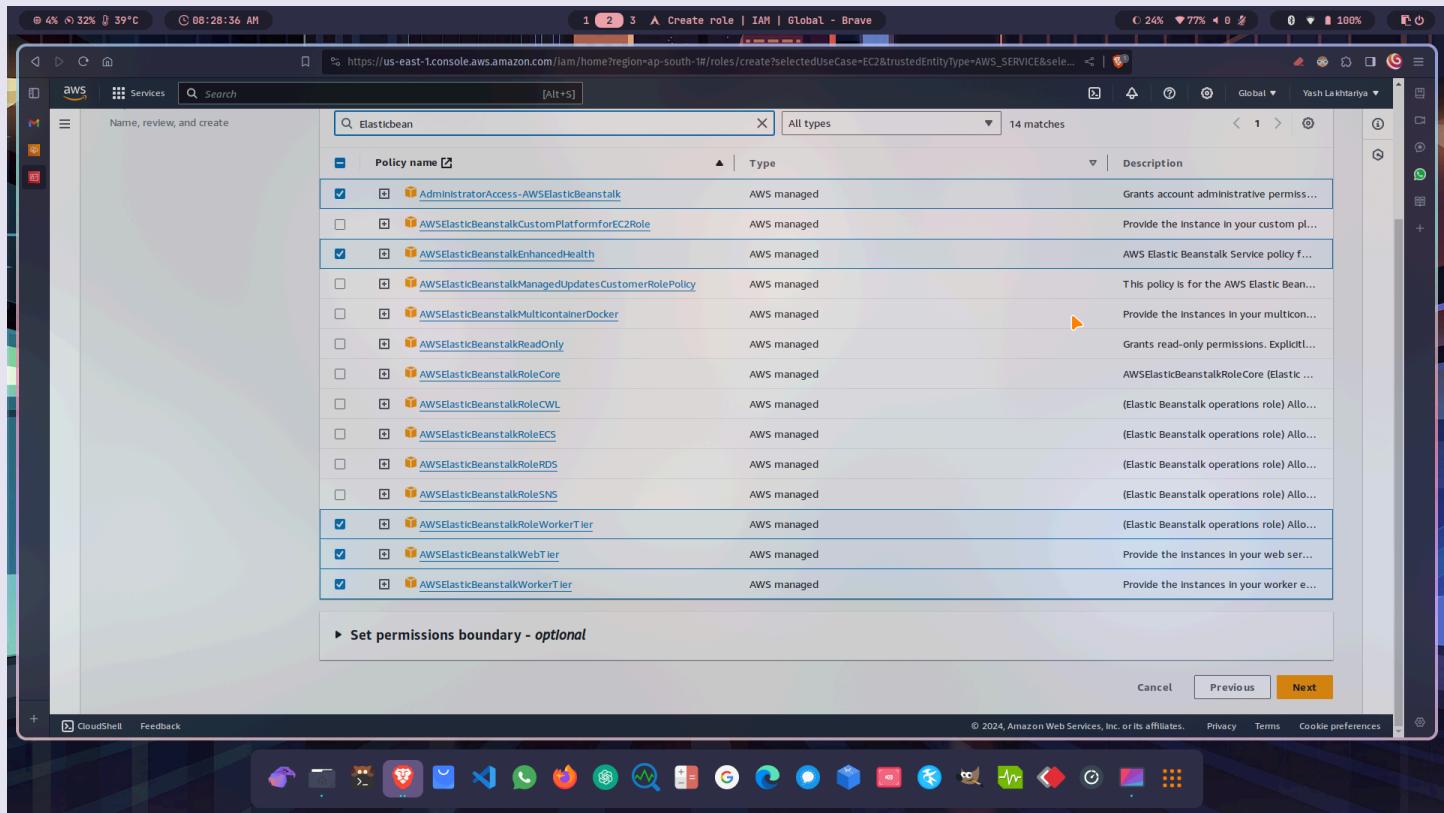
EADC Practical 1

4. Use a new tab to create a role via IAM. To create a role, use EC2 service role option and AWS service trusted entity type.



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1

## 5. Give the following permissions to the role for accessing ElasticBean services.



The screenshot shows the AWS IAM console during the creation of a new role. The search bar at the top has 'Elasticbean' typed into it. Below the search bar, a table lists 14 AWS managed policies. Several policies are selected with checkboxes:

Policy name	Type	Description
<input checked="" type="checkbox"/> AdministratorAccess-AWSelasticbeanstalk	AWS managed	Grants account administrative permis...
<input type="checkbox"/> AWSElasticBeanstalkCustomPlatformforEC2Role	AWS managed	Provide the instance in your custom pl...
<input checked="" type="checkbox"/> AWSElasticBeanstalkEnhancedHealth	AWS managed	AWS Elastic Beanstalk Service policy f...
<input type="checkbox"/> AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy	AWS managed	This policy is for the AWS Elastic Bean...
<input type="checkbox"/> AWSElasticBeanstalkMulticontainerDocker	AWS managed	Provide the instances in your multicon...
<input type="checkbox"/> AWSElasticBeanstalkReadOnly	AWS managed	Grants read-only permissions. Explicitl...
<input type="checkbox"/> AWSElasticBeanstalkRoleCore	AWS managed	AWSElasticBeanstalkRoleCore (Elastic ...
<input type="checkbox"/> AWSElasticBeanstalkRoleCWL	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/> AWSElasticBeanstalkRoleECS	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/> AWSElasticBeanstalkRoleRDS	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/> AWSElasticBeanstalkRoleSNS	AWS managed	(Elastic Beanstalk operations role) Allo...
<input checked="" type="checkbox"/> AWSElasticBeanstalkRoleWorkerTier	AWS managed	(Elastic Beanstalk operations role) Allo...
<input checked="" type="checkbox"/> AWSElasticBeanstalkWebTier	AWS managed	Provide the instances in your web ser...
<input checked="" type="checkbox"/> AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instances in your worker e...

Below the table, there is a section titled 'Set permissions boundary - optional' with a 'Next Step' button.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 1

## 6. Enter a preferable name and description of the role and create it.

The screenshot shows the AWS IAM 'Create role' wizard, Step 1: Select trusted entity. The URL is https://us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/roles/create?selectedUseCase=EC2&trustedEntityType=AWS\_SERVICE&selectedServiceName=ec2&selectedActionName=AssumeRole. The page title is 'Name, review, and create'. The 'Role details' section contains a 'Role name' field with the value 'AWS\_EstCbn\_EC2\_Role1' and a 'Description' field with the value 'Allows EC2 instances to call AWS services on your behalf.'. The 'Step 1: Select trusted entities' section shows a trust policy JSON code:

```
1 ~ [{}  
2 ~     "Version": "2012-10-17",  
3 ~     "Statement": [  
4 ~         {  
5 ~             "Effect": "Allow",  
6 ~             "Action": [  
7 ~                 "sts:AssumeRole"  
8 ~             ],  
9 ~             "Principal": {  
10 ~                 "Service": [  
11 ~                     "ec2.amazonaws.com"  
12 ~                 ]  
13 ~             }  
14 ~         }  
15 ~     ]  
16 ~ ]
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1

7. In Access management, check Roles section to ensure the role is successfully created.

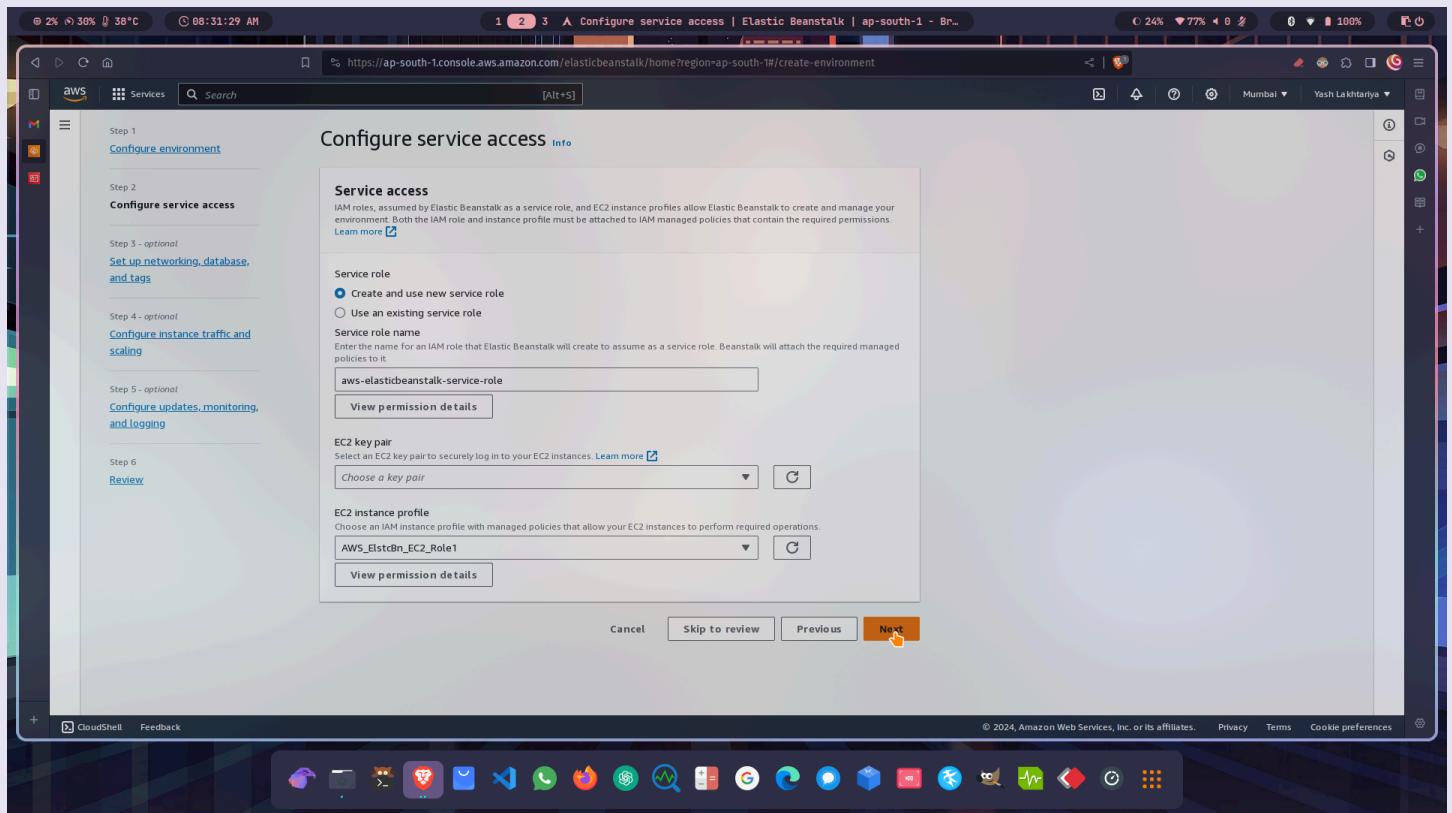
The screenshot shows the AWS Identity and Access Management (IAM) console. The left sidebar is collapsed, and the main area displays the 'Roles' section under the 'Access management' heading. There are three roles listed:

Role name	Trusted entities	Last activity
<a href="#">AWS_ElastBn_EC2_Role1</a>	AWS Service: ec2	-
<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linked)	-
<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service)	-

Below the roles, there is a section titled 'Roles Anywhere' with three options: 'Access AWS from your non AWS workloads', 'X.509 Standard', and 'Temporary credentials'.

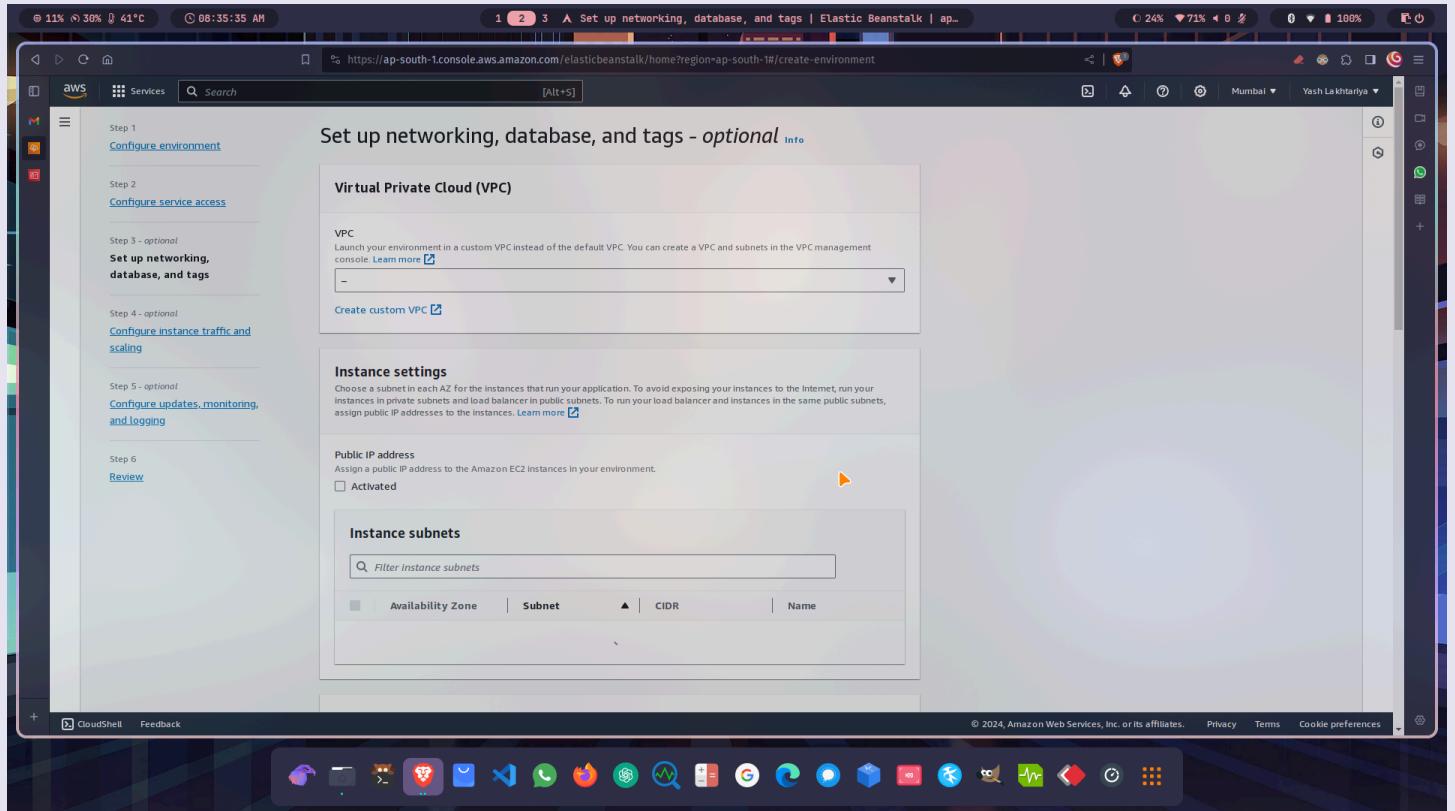
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1

- Now, open the project configuration tab again and after refreshing the roles available, enter the role just created.

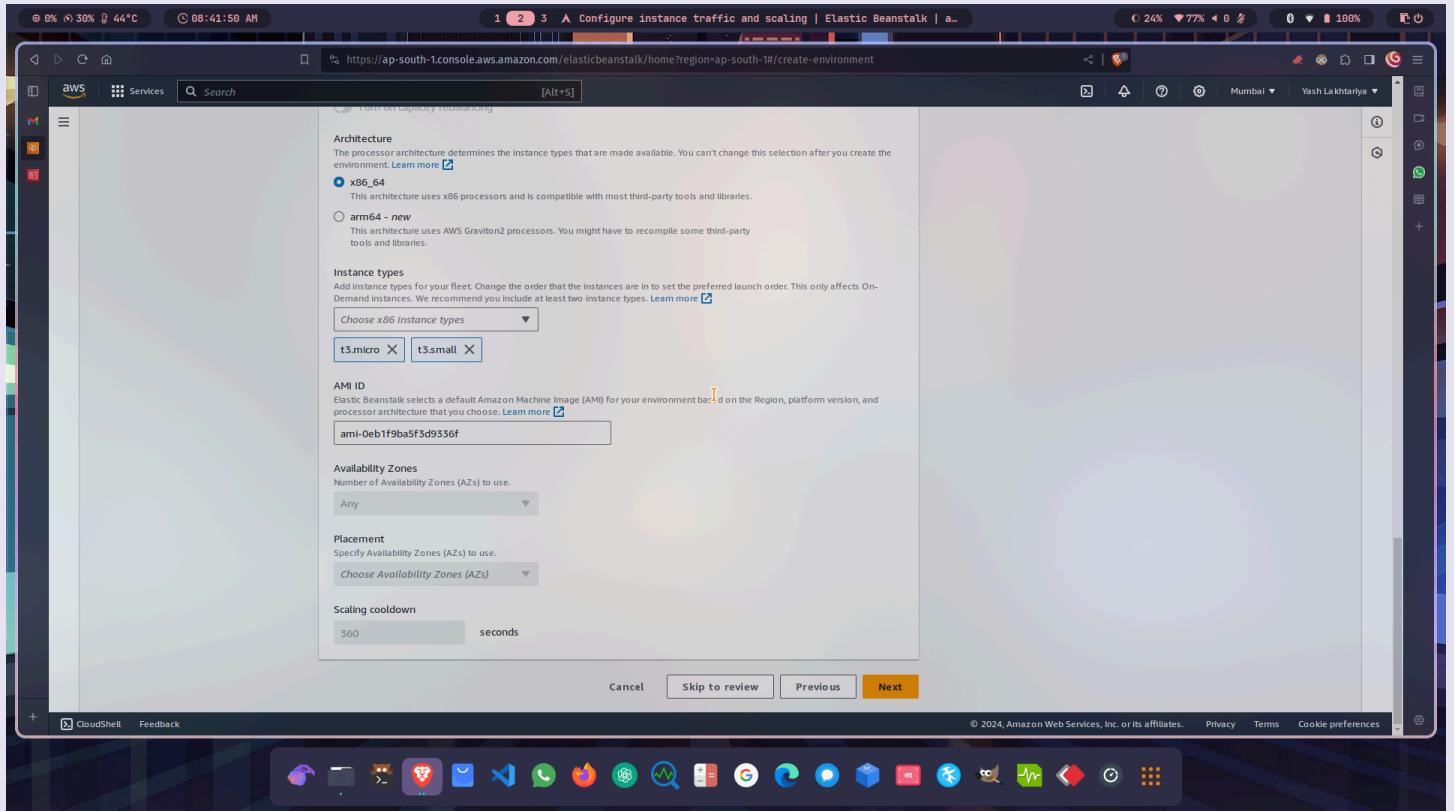


Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1

## 9. Leave the other setting default and submit the changes.



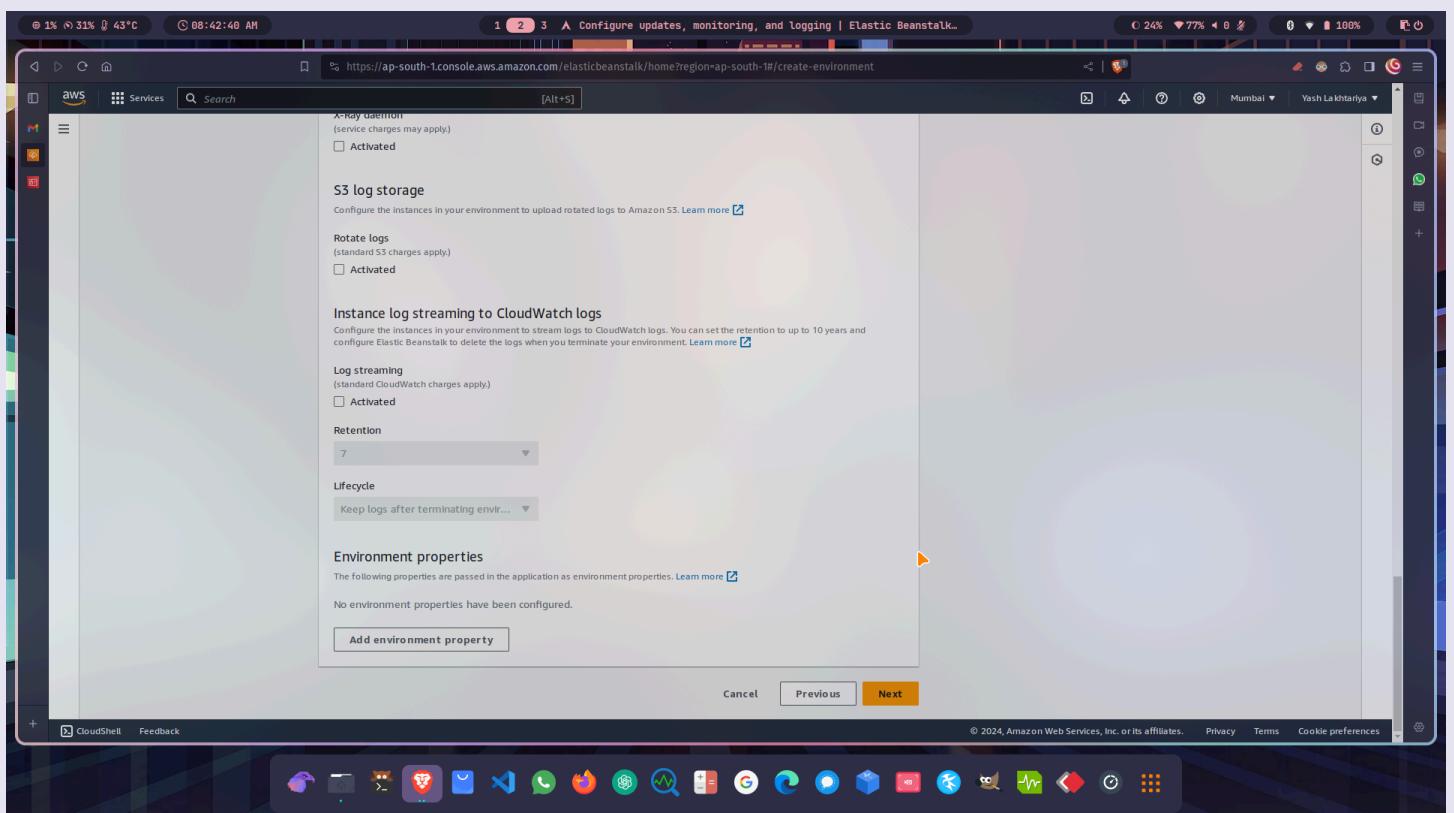
**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 1**



The screenshot shows the 'Configure instance traffic and scaling' step of the Elastic Beanstalk environment creation wizard. It includes sections for:

- Architecture:** Options for x86\_64 (selected) and arm64-new.
- Instance types:** A dropdown menu showing 'Choose x86 instance types' with 't3.micro' and 't3.small' selected.
- AMI ID:** Shows 'ami-0eb1f9ba5f3d9336f'.
- Availability Zones:** Set to 'Any'.
- Placement:** Set to 'Choose Availability Zones (AZs)'.
- Scaling cooldown:** Set to 360 seconds.

At the bottom are 'Cancel', 'Skip to review', 'Previous', and a yellow 'Next' button.



The screenshot shows the 'Configure updates, monitoring, and logging' step of the Elastic Beanstalk environment creation wizard. It includes sections for:

- X-Ray daemon:** An unchecked checkbox labeled '(service charges may apply)'.
- S3 log storage:** An unchecked checkbox labeled 'Activated'.
- Rotate logs:** An unchecked checkbox labeled '(standard S3 charges apply)'.
- Instance log streaming to CloudWatch logs:** An unchecked checkbox labeled 'Activated'.
- Log streaming:** An unchecked checkbox labeled '(standard CloudWatch charges apply)'.
- Retention:** Set to 7 days.
- Lifecycle:** Set to 'Keep logs after terminating envr...'.
- Environment properties:** A note stating 'The following properties are passed in the application as environment properties.' followed by 'No environment properties have been configured.' and a 'Add environment property' button.

At the bottom are 'Cancel', 'Previous', and a yellow 'Next' button.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 1

The screenshot shows the 'Configure environment - review' step in the AWS Elastic Beanstalk console. The URL in the address bar is <https://ap-south-1.console.aws.amazon.com/elasticbeanstalk/home?region=ap-south-1#/create-environment>. The page displays configuration settings for a new environment:

Updates		
Managed updates	Deployment batch size	Deployment batch size type
Activated	100	Percentage
Command timeout	Deployment policy	Health threshold
600	AllAtOnce	Ok
Ignore health check	Instance replacement	
false	false	
Platform software		
Lifecycle	Log streaming	Proxy server
false	Deactivated	nginx
Logs retention	Rotate logs	Update level
7	Deactivated	minor
X-Ray enabled		
Deactivated		
Environment properties		
Key	Value	
No environment properties		
There are no environment properties defined		

At the bottom right of the configuration area, there are 'Cancel', 'Previous', and 'Submit' buttons. The 'Submit' button is highlighted in yellow. The status bar at the bottom of the browser window shows 'CloudShell' and 'Feedback'.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1

10. Wait till the environment is created. Afterwards, click on the domain link to get the sample project.

The screenshot shows the AWS Elastic Beanstalk Environment overview page for the environment 'YashLEADCP1-env'. The left sidebar shows the navigation path: Elastic Beanstalk > Environments > YashLEADCP1-env. The main content area displays the 'Environment overview' section with the status 'Health: Ok' and 'Domain: YashLEADCP1-env.eba-hkpppe5g.ap-south-1.elasticbeanstalk.com'. The 'Platform' section shows 'Node.js 18 running on 64bit Amazon Linux 2023/6.0.4'. Below this, the 'Events' tab is selected, showing a list of four INFO-level events from January 17, 2024, detailing the successful launch and deployment of the environment. The bottom right corner of the browser window shows the AWS logo and the text 'Yash Lakhtariya'.

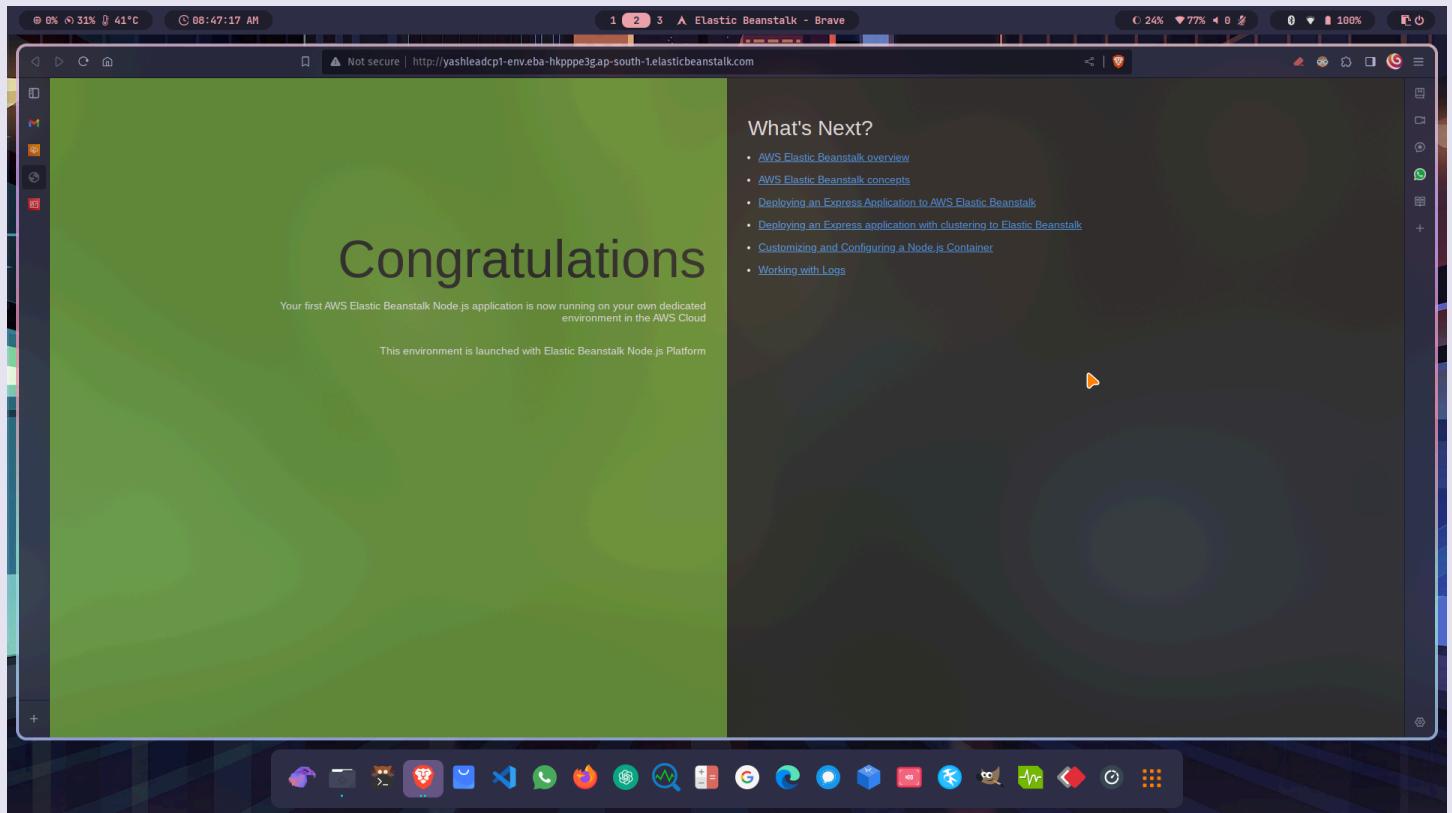
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

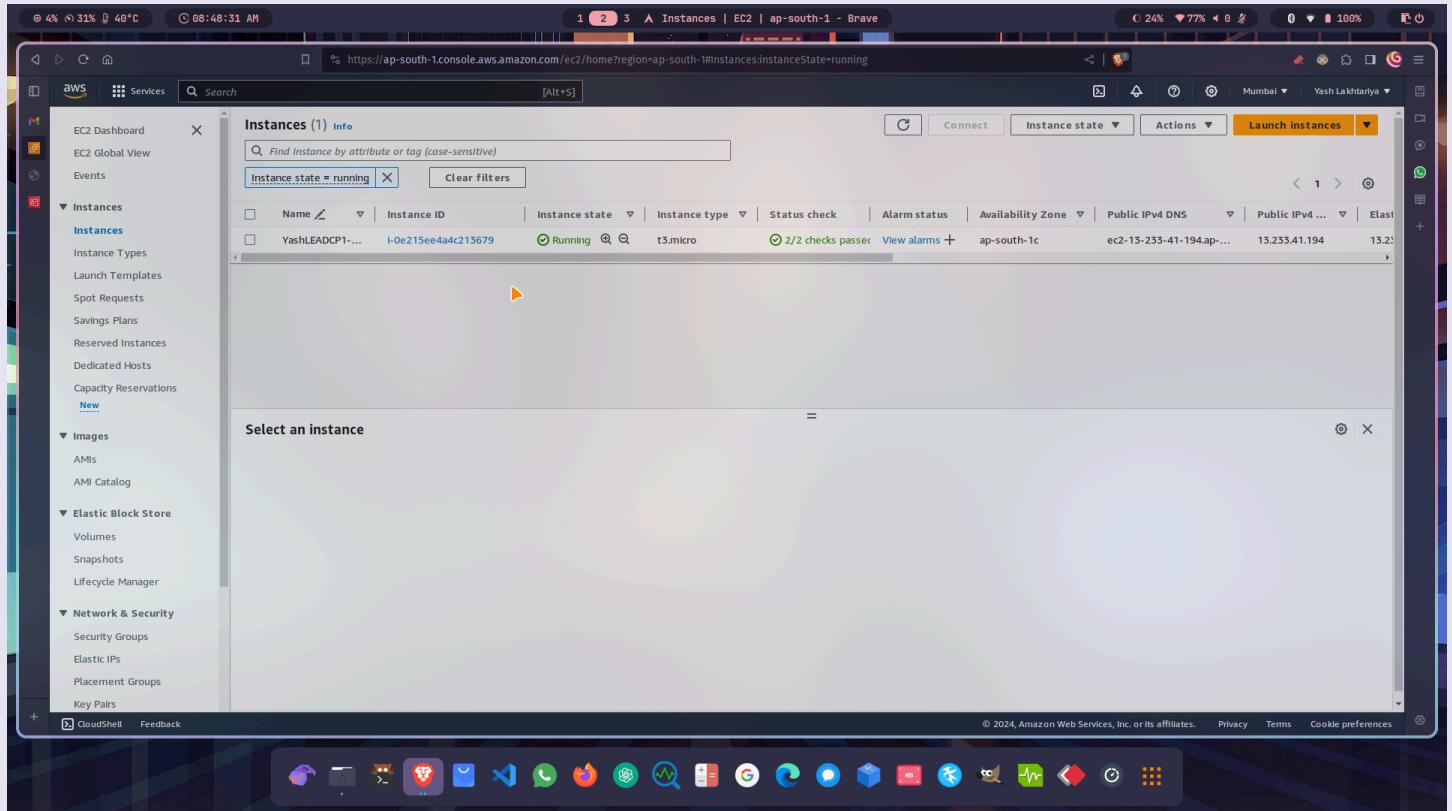
EADC Practical 1

11. The demo project of nodejs on AWS looks like this.



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 1**

**12. In Instances, the EC2 instance can be found and its IP can be found**



The screenshot shows the AWS EC2 Instances page. The left sidebar navigation includes: AWS Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations (New), Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs). The main content area displays the 'Instances (1) Info' section with a table. The table has columns: Name, Instance ID, Instance state, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv4 IP. One row is present: Name: YashLEADCPr..., Instance ID: i-0e215ee4a4c213679, Instance state: Running, Status check: 2/2 checks passed, Alarm status: View alarms, Availability Zone: ap-south-1c, Public IPv4 DNS: ec2-13-233-41-194.ap..., Public IPv4 IP: 15.233.41.194. The bottom of the page features a toolbar with various icons and links for CloudShell, Feedback, and legal information.

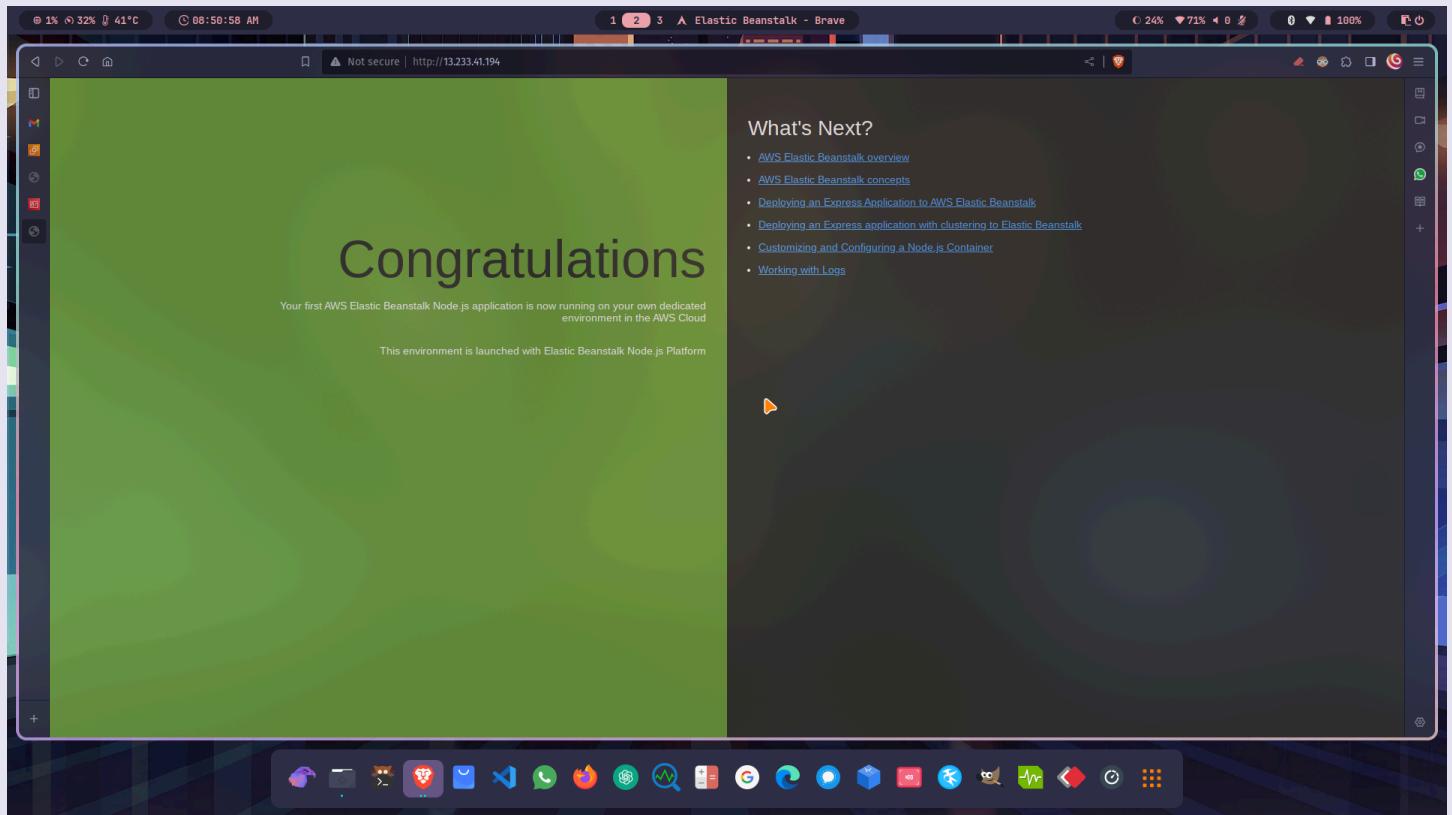
Name	Instance ID	Instance state	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP
YashLEADCPr...	i-0e215ee4a4c213679	Running	2/2 checks passed	View alarms	ap-south-1c	ec2-13-233-41-194.ap...	15.233.41.194

Name - Yash Lakhtariya

Enrollment number - 21162101012

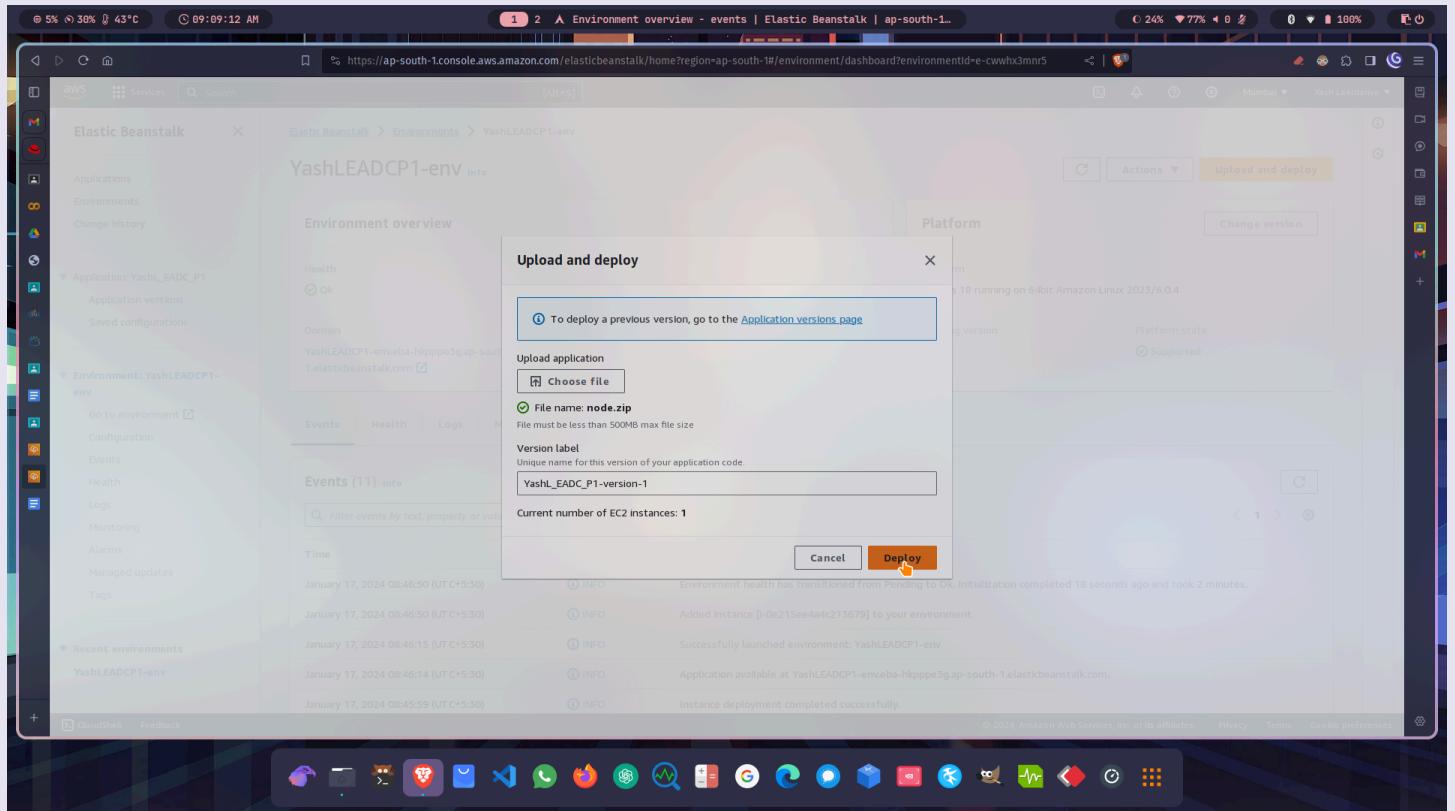
Branch - CBA      Batch - 61

EADC Practical 1



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 1**

13. Now, download the file and make changes to the project to check the compatibility. And upload the zip file of app.js, cron.yaml, index.html, package.json, public directory and other required files.

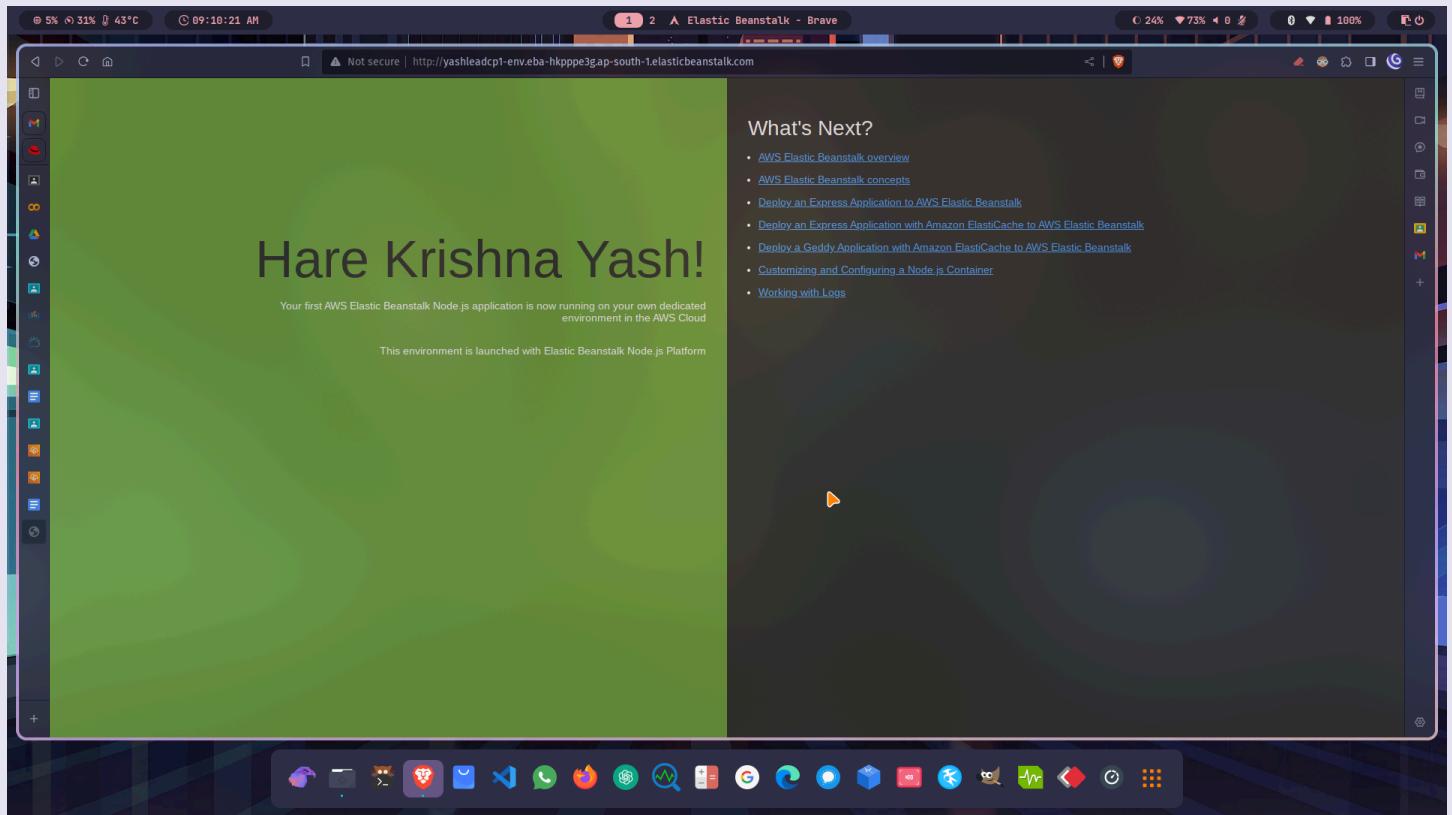


Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 1



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1

14. Similarly, update the code again of bulb buttons GUI as a new version and wait for finishing the update.

**HTML:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
  <title>YSL Light Bulb</title>
  <style>
    body {
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
      margin: 0;
      background-color: #ffffff;
    }

    #bulb {
      width: 400px;
      height: 400px;
      background-size: cover;
      background-repeat: no-repeat;
    }

    #controls {
      display: flex;
      justify-content: center;
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1

```
margin-top: 20px;
}

.btn {
    margin: 0 10px;
    padding: 10px 15px;
    font-size: 22px;
    background-color: #5e81cc;
    color: #ffffff;
    border: none;
    border-radius: 16px;
    cursor: pointer;
}

.btn:hover {
    background-color: #a347ba;
}
</style>
</head>
<body>
    <div id="bulb" style="background-image: url('bulb_off.png')"></div>
    &nbsp;&nbsp;
    <div id="controls">
        <button class="btn" onclick="turnOn()">ON</button>
        <button class="btn" onclick="turnOff()">OFF</button>
    </div>

    <script>
        function turnOn() {
            document.getElementById("bulb").style.backgroundImage =
                "url('bulb_on.png')";
        }
    </script>
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1

```
function turnOff() {
    document.getElementById("bulb").style.backgroundImage =
        "url('bulb_off.png')";
}

</script>
</body>
</html>
```

## NodeJS:

```
let express = require("express");
let app = express();
let port = process.env.PORT || 8000;

app.use(express.static(__dirname + "/public"));

app.listen(port, function () {
    console.log(`\nServer is being started on ${port}`);
});
```

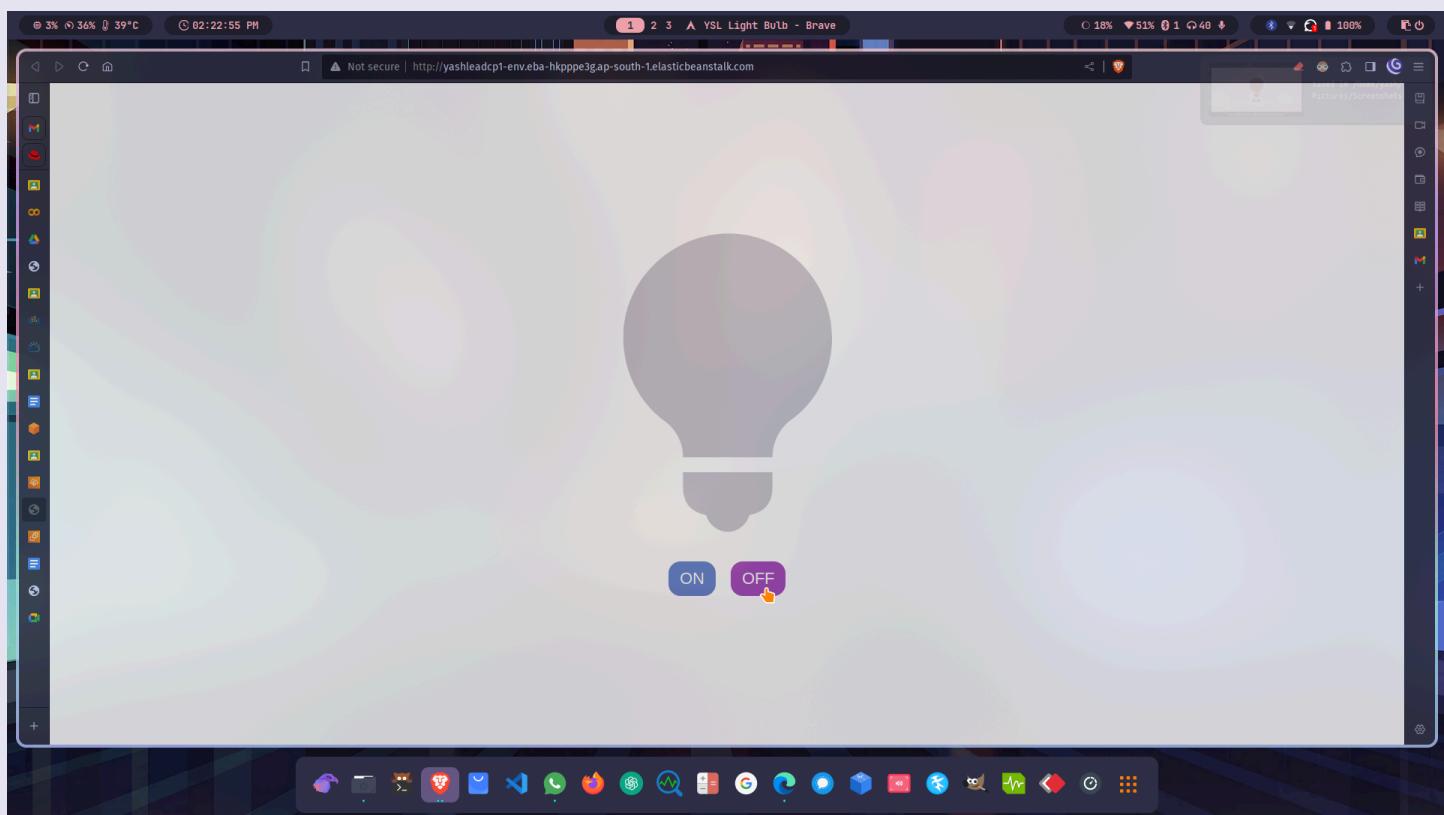
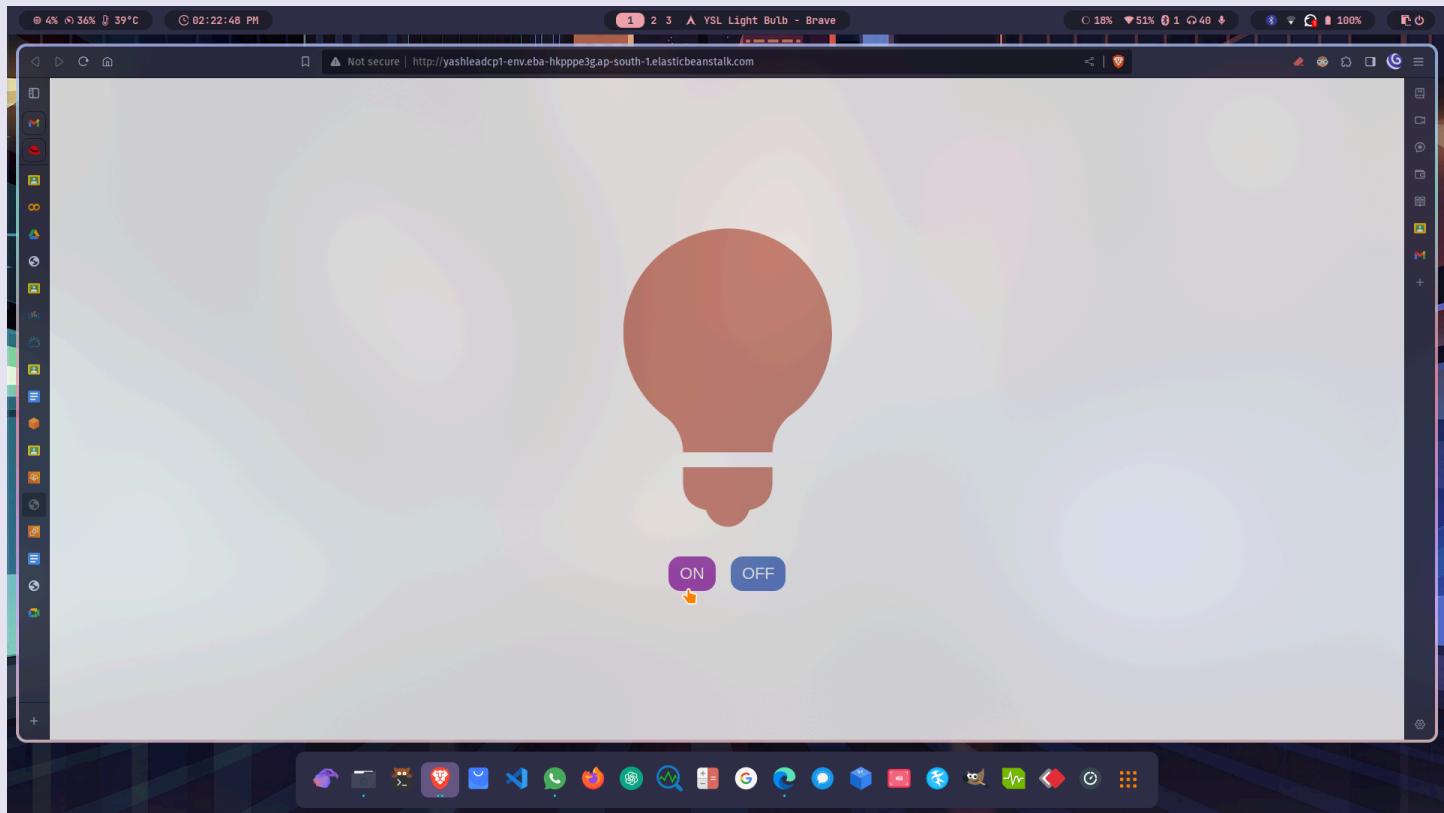
## Output:

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 1



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1

**Practical 1.2 : Developing NodeJS application to create website GUI (use HTML, CSS, JavaScript) using AWS Cloud and AWS Elastic Beanstalk to Click on the light bulb to turn on/off the light**

Similar to previous steps, upload zip file of new code for Bulb Click GUI

**HTML :**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
  <title>YSL Light Bulb</title>
  <style>
    body {
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
      margin: 0;
      background-color: #ffffff;
    }

    #bulb {
      width: 500px;
      height: 500px;
      background-size: cover;
      background-repeat: no-repeat;
      cursor: pointer;
    }
  </style>
</head>
<body>
  <div id="bulb" style="background-image: url('https://i.imgur.com/3QzJGZM.jpg');"></div>
</body>
</html>
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1

```
</style>
</head>
<body>
<div
  id="bulb"
  style="background-image: url('bulb_off.png')"
  onclick="toggleBulb()"
></div>

<script>
let isBulbOn = false;
function toggleBulb() {
  const bulbImage = document.getElementById("bulb");

  if (isBulbOn) {
    bulbImage.style.backgroundImage = "url('bulb_off.png')";
  } else {
    bulbImage.style.backgroundImage = "url('bulb_on.png')";
  }
  isBulbOn = !isBulbOn;
}
</script>
</body>
</html>
```

## NodeJS :

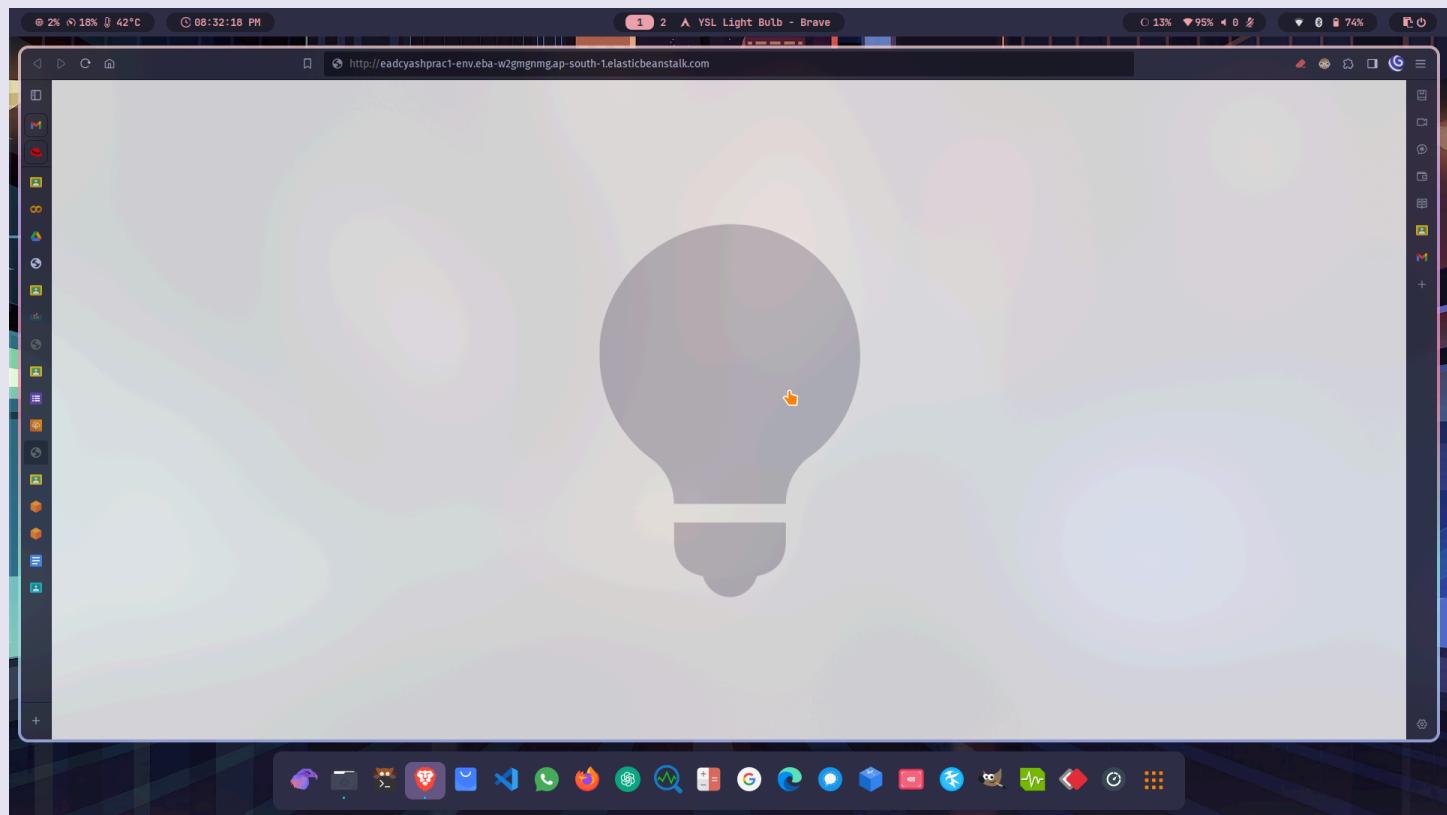
```
let express = require("express");
let app = express();
let port = process.env.PORT || 8000;

app.use(express.static(__dirname + "/public"));
```

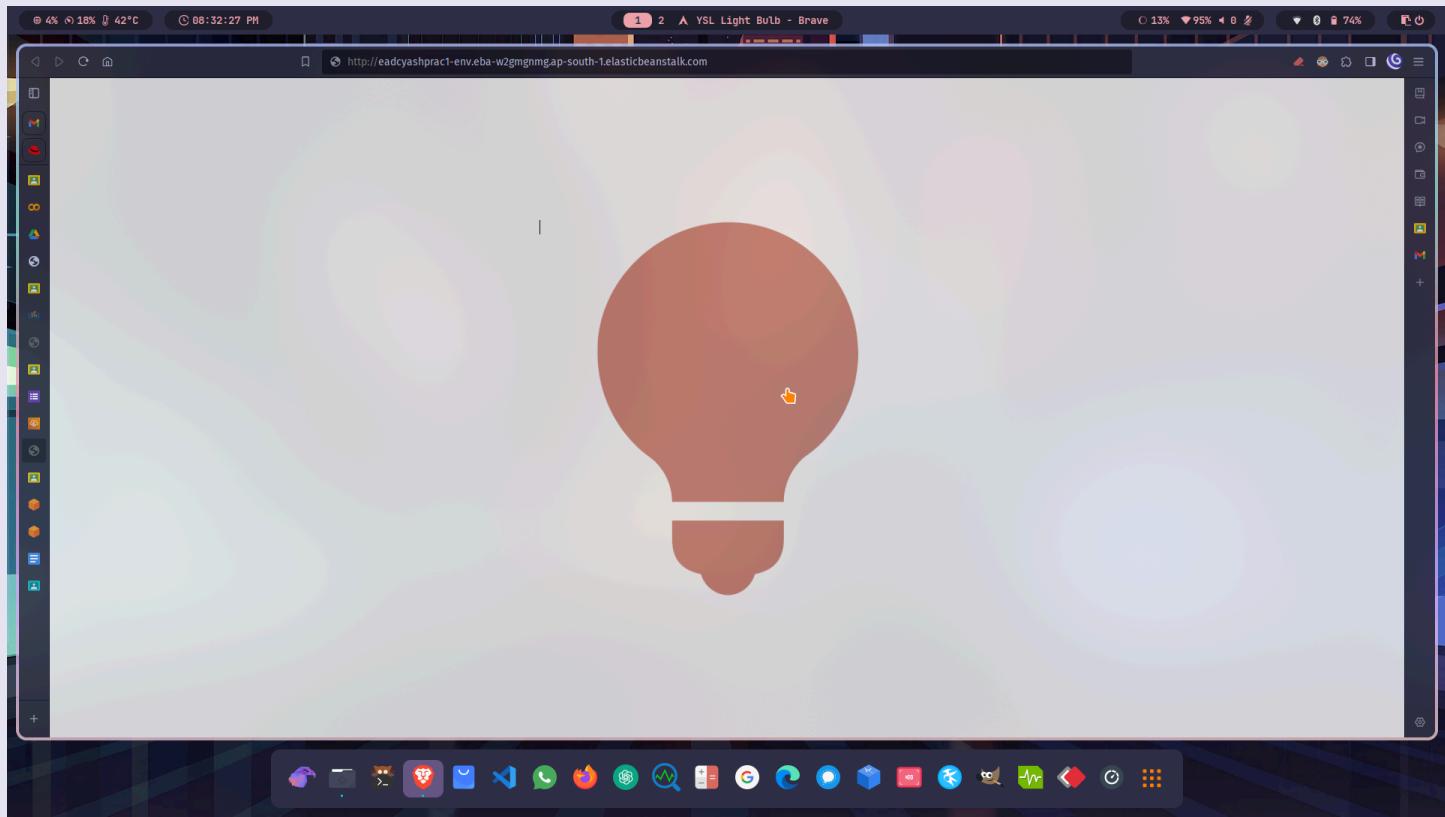
**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 1**

```
app.listen(port, function () {  
  console.log(`\nServer is being started on ${port}`);  
});
```

**Output:**



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1



**Practical 1.3 : Develop a Hello World NodeJS application on the AWS Cloud using the HTTP module create your module and include it in the NodeJS application to print the current date and time.**

Here also, upload the new code zip file to get date and time on the page

**NodeJS :**

```
let d = new Date();

function YSLdatetime() {
    let date = d.getDate();
    let month = d.getMonth() + 1;
    let year = d.getFullYear();
    let hour = d.getHours();
    let min = d.getMinutes();
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 1

```
let sec = d.getSeconds();

return {
  tarikh: `${date}-${month}-${year}`,
  samay: `${hour}:${min}:${sec}`
};

}

let dt = YSLdatetime();
let http = require("http");

http
.createServer(function (req, res) {

  res.write("\n\n\tWelcome to the YSL Date Time Module!\n\t");
  res.write(`\n\n\tDate : ${dt.tarikh}`);
  res.write(`\n\tTime : ${dt.samay}`);

  res.end();
})
.listen(8080, function () {
  console.log("\n\tServer is running on port 8080");
});
```

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 1

Output:

