

# Capstone Project

**Project Name:** Scalable Python and React App on AWS

**Description:** This project involves designing and deploying a Python-based API backend and a React frontend on AWS. The solution ensures high availability, security, and global performance by leveraging AWS services like Elastic Beanstalk, RDS for database management, S3 for secure image storage, CloudFront for global caching, and EC2 for scalability. Automated backups and disaster recovery solutions are also implemented to ensure reliability. The infrastructure is designed to handle variable traffic loads and maintain seamless user experience worldwide.

## Step 1: Set Up a Virtual Private Cloud (VPC)

1. **Create a VPC:**
  - Name: MyAppVPC
  - CIDR Block: 10.0.0.0/16
2. **Create Subnets:**
  - **Public Subnet:** 10.0.1.0/24, Auto-assign Public IP
  - **Private Subnet:** 10.0.2.0/24, No Public IP
3. **Create an Internet Gateway (IGW):**
  - Name: MyAppIGW
  - Attach to MyAppVPC.
4. **Create Route Tables:**
  - **Public Route Table:** Route 0.0.0.0/0 to IGW, associate with PublicSubnet.
  - **Private Route Table:** No additional routes, associate with PrivateSubnet.
5. **Create a NAT Gateway:**
  - Subnet: PublicSubnet
  - Allocate a new Elastic IP.

**Create VPC** Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

### VPC settings

**Resources to create** Info  
Create only the VPC resource or the VPC and other networking resources.

VPC only  VPC and more

**Name tag auto-generation** Info  
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

Auto-generate   
 MyAppVPC

**IPv4 CIDR block** Info  
Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs  
CIDR block size must be between /16 and /28.

**IPv6 CIDR block** Info  
 No IPv6 CIDR block  Amazon-provided IPv6 CIDR block

### Preview

**Create VPC workflow**

Wait for NAT Gateways to activate 70%

Details

- Create VPC: vpc-0b51ce74a7f6f84ed
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation: vpc-0b51ce74a7f6f84ed
- Create S3 endpoint: vpcce-03b229ea2bc1694d8
- Create subnet: subnet-0717a190a547858a5
- Create subnet: subnet-0d1657da29e2b1bc
- Create internet gateway: igw-0f528785410987d5c
- Attach internet gateway to the VPC
- Create route table: rtb-0edffccc732b2f461e
- Create route
- Associate route table
- Allocate elastic IP: eipalloc-01af02afa7df53382
- Create NAT gateway: nat-0bd38327e468a0618
- Wait for NAT Gateways to activate
- Create route table
- Create route

**VPC dashboard**

**vpc-0b51ce74a7f6f84ed / MyAppVPC-vpc**

Details			
VPC ID <a href="#">vpc-0b51ce74a7f6f84ed</a>	State <span>Available</span>	DNS hostnames Enabled	DNS resolution Enabled
Tenancy Default	DHCP option set <a href="#">dopt-084f8c0929a640f12</a>	Main route table <a href="#">rtb-0fa0940b6be4d56ad</a>	Main network ACL <a href="#">acl-00b5877413b204010</a>
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID <a href="#">891612578515</a>	

**Resource map**

## Step 2: Create Security Groups

### 2.1 React Frontend Security Group:

- Name: **ReactFrontendSG**.
- Inbound Rule: Allow HTTP (port 80) and HTTPS (port 443) from **0.0.0.0/0**.

### 2.2 Python API Security Group:

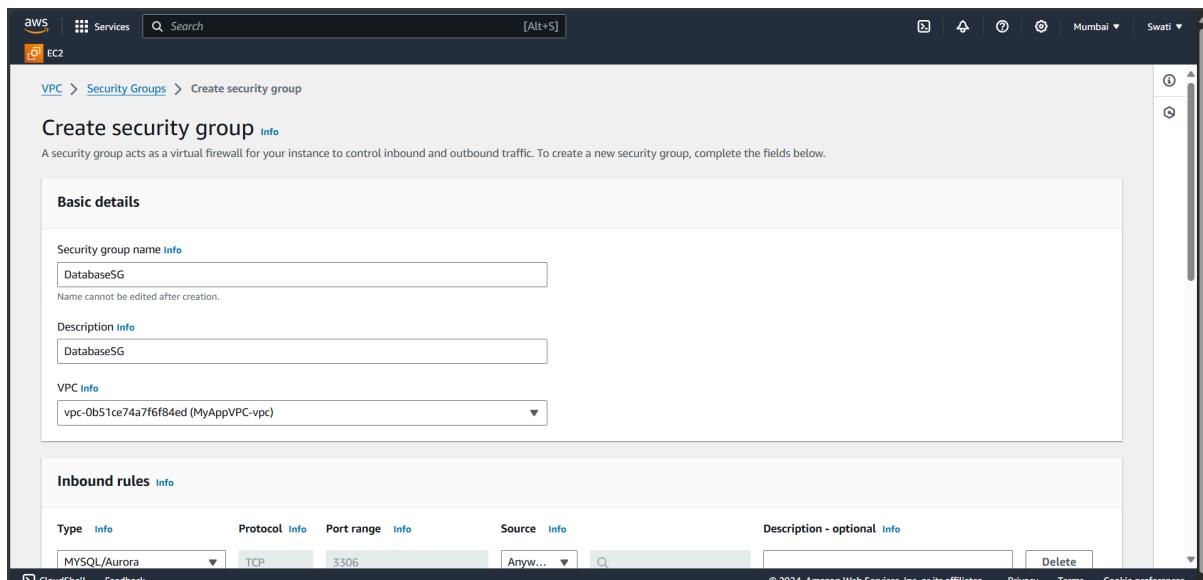
- Name: **PythonBackendSG**.
- Inbound Rule: Allow HTTP (port 80) only from **PublicSubnet**.

### 2.3 Database Security Group:

- Name: **DatabaseSG**.
- Inbound Rule: Allow MySQL/Aurora (port 3306) from **PrivateSubnet** (Python API).

The screenshot shows the 'Create security group' interface in the AWS VPC console. Under the 'Basic details' section, the security group name is set to 'ReactFrontendSG'. The 'Description' field contains 'ReactFrontendSG'. The 'VPC' dropdown is set to 'vpc-0b51ce74a7f6f84ed (MyAppVPC-vpc)'. The 'Inbound rules' section is expanded, showing a single rule: Type: HTTP, Protocol: TCP, Port range: 80, Source: Anywhere. A 'Delete' button is visible next to the rule.

The screenshot shows the 'Create security group' interface in the AWS VPC console. Under the 'Basic details' section, the security group name is set to 'PythonBackendSG'. The 'Description' field contains 'PythonBackendSG'. The 'VPC' dropdown is set to 'vpc-0b51ce74a7f6f84ed (MyAppVPC-vpc)'. The 'Inbound rules' section is expanded, showing a single rule: Type: HTTP, Protocol: TCP, Port range: 80, Source: Anywhere. A 'Delete' button is visible next to the rule.



## Step 3: Deploy React Frontend on Elastic Beanstalk

### 3.1 Create an S3 Bucket:

- Go to **S3 Console**.
- Click **Create Bucket**.
- Name: **react-frontend-bucket**.
- Upload your React app.

### 3.2 Set Up Elastic Beanstalk:

- Go to **Elastic Beanstalk Console**.
- Click **Create Application**.
- Name: **Simple-ReactJs-App**.
- Platform: **Node.js**.
- Upload the React source code from the S3 bucket.
- Instance type: **t2.micro**.
- Environment Type: Load-balanced, auto-scaling.
- Click **Create** to deploy.

### 3.3 Configure Auto-Scaling:

- Set **min instances** to 2 and **max instances** to 4.
- Add scaling policies based on CPU utilization (>70%).

AWS Services Search [Alt+S]

EC2

Amazon S3 > Buckets > Create bucket

### Create bucket Info

Buckets are containers for data stored in S3.

**General configuration**

AWS Region: Asia Pacific (Mumbai) ap-south-1

Bucket name: react-frontend-bucket

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

**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)  ACLs enabled

View details X

View Storage Lens dashboard

Successfully created bucket "react-frontend-bucket". To upload files and folders, or to configure additional bucket settings, choose [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 (1) Info All AWS Regions**

Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date
react-frontend-bucket	Asia Pacific (Mumbai) ap-south-1	<a href="#">View analyzer for ap-south-1</a>	September 6, 2024, 15:36:08 (UTC+05:30)

https://ap-south-1.console.aws.amazon.com/s3/buckets/react-frontend-bucket?region=ap-south-1&bucketType=general&tab=objects

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

**Objects (9) Info**

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
asset-manifest.json	json	September 6, 2024, 15:48:49 (UTC+05:30)	257.0 B	Standard
assets/	Folder	-	-	-
favicon.ico	ico	September 6, 2024, 15:48:49 (UTC+05:30)	3.8 KB	Standard
index.html	html	September 6, 2024, 15:48:49 (UTC+05:30)	873.0 B	Standard
manifest.json	json	September 6, 2024, 15:48:49 (UTC+05:30)	317.0 B	Standard
react-source-code.zip	zip	September 6, 2024, 16:02:01 (UTC+05:30)	42.7 MB	Standard
service-worker.js	js	September 6, 2024, 15:48:49 (UTC+05:30)	3.3 KB	Standard

aws Services Search [Alt+S]

EC2

Cancel Previous Submit

CloudShell

ap-south-1 +

```
[cloudshell-user@ip-10-134-38-162 simple-reactjs-app]$ eb init
Select a default region
1) us-east-1 : US East (N. Virginia)
2) us-west-1 : US West (N. California)
3) us-west-2 : US West (Oregon)
4) eu-west-1 : EU (Ireland)
5) eu-central-1 : EU (Frankfurt)
6) ap-south-1 : Asia Pacific (Mumbai)
7) ap-southeast-1 : Asia Pacific (Singapore)
8) ap-southeast-2 : Asia Pacific (Sydney)
9) ap-northeast-1 : Asia Pacific (Tokyo)
10) ap-northeast-2 : Asia Pacific (Seoul)
11) sa-east-1 : South America (Sao Paulo)
12) cn-north-1 : China (Beijing)
13) cn-northwest-1 : China (Ningxia)
14) us-east-2 : US East (Ohio)
15) ca-central-1 : Canada (Central)
16) eu-west-2 : EU (London)
17) eu-west-3 : EU (Paris)
18) eu-north-1 : EU (Stockholm)
19) eu-south-1 : EU (Milano)
20) ap-east-1 : Asia Pacific (Hong Kong)
21) me-south-1 : Middle East (Bahrain)
22) il-central-1 : Middle East (Israel)
```

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

EC2

Cancel Previous Submit

CloudShell

ap-south-1 +

```
25) ap-northeast-3 : Asia Pacific (Osaka)
(default is 3): 6

Select an application to use
1) MyReactApp
2) [ Create new Application ]
(default is 2): 2

Enter Application Name
(default is "simple-reactjs-app"):
Application simple-reactjs-app has been created.

It appears you are using Node.js. Is this correct?
(Y/n): Y
Select a platform branch.
1) Node.js 20 running on 64bit Amazon Linux 2023
2) Node.js 18 running on 64bit Amazon Linux 2023
3) Node.js 18 running on 64bit Amazon Linux 2
4) Node.js 16 running on 64bit Amazon Linux 2 (Deprecated)
```

CloudShell

ap-south-1 +

```
5) Node.js 14 running on 64bit Amazon Linux 2 (Deprecated)
(default is 1): 2

Do you wish to continue with CodeCommit? (Y/n): Y

Enter Repository Name
(default is "origin"): my-react-app
Successfully created repository: my-react-app

Enter Branch Name
***** Must have at least one commit to create a new branch with CodeCommit *****
(default is "master").
Successfully created branch: master
Do you want to set up SSH for your instances?
(Y/n): Y

Select a keypair.
1) key-pair
2) Key
3) [ Create new KeyPair ]
(default is 2): 1
```

AWS Services Search [Alt+S] Mumbai Swati

EC2 Elastic Beanstalk Applications Environments Change history Recent environments react-env

Elastic Beanstalk is launching your environment. This will take a few minutes.

Environment successfully launched.

Applications (1) Info Create application

Application name	Environments	Date created	Last modified
simple-reactjs-app	react-env	September 6, 2024 16:35:23 (U...)	September 6, 2024 16:35:23 (U...)

AWS Services Search [Alt+S] Mumbai Swati

EC2 Elastic Beanstalk Applications Environments Change history Application: simple-reactjs-app Application versions Saved configurations Environment: react-env Go to environment Configuration Events Health Logs Monitoring Alarms Managed updates

Elastic Beanstalk is launching your environment. This will take a few minutes.

Environment successfully launched.

Elastic Beanstalk > Environments > react-env react-env Info Upload and deploy

Environment overview

Health	Environment ID
⚠ Warning - View causes	e-mjhkdecu2
Domain	Application name
react-env.eba-hg6e2ddd.ap-south-1.elasticbeanstalk.com	simple-reactjs-app

Platform

Platform	Change version
Node.js 18 running on 64bit Amazon Linux 2023/6.2.0	
Running version	app-1b4a-240906_110902674948
Platform state	Supported

Events Health Logs Monitoring Alarms Managed updates Tags

AWS Services Search [Alt+S] Mumbai Swati

EC2 CloudShell ap-south-1 + Actions

```
[cloudshell-user@ip-10-132-83-167 ~]$ aws elasticbeanstalk create-application-version --application-name simple-reactjs-app --version-label 1 --source-bundle S3Bucket="react-frontend-bucket" S3Key="react-source-code.zip"
{
  "ApplicationVersion": {
    "ApplicationVersionArn": "arn:aws:elasticbeanstalk:ap-south-1:891612578515:applicationversion/simple-reactjs-app/1",
    "ApplicationName": "simple-reactjs-app",
    "VersionLabel": "1",
    "SourceBundle": {
      "S3Bucket": "react-frontend-bucket",
      "S3Key": "react-source-code.zip"
    },
    "DateCreated": "2024-09-06T13:01:20.025000+00:00",
    "DateUpdated": "2024-09-06T13:01:20.025000+00:00",
    "Status": "UNPROCESSED"
  }
}
[cloudshell-user@ip-10-132-83-167 ~]$ aws elasticbeanstalk update-environment --environment-name react-env --version-label 1
{
  "EnvironmentName": "react-env",
  "EnvironmentId": "e-mjhkdecu2",
  "ApplicationName": "simple-reactjs-app",
  "VersionLabel": "1",
  "SolutionStackName": "64bit Amazon Linux 2023 v6.2.0 running Node.js 18",
  "PlatformArn": "arn:aws:elasticbeanstalk:ap-south-1::platform/Node.js 18 running on 64bit Amazon Linux 2023/6.2.0",
  "Description": "Environment created from the EB CLI using \"eb create\"",
  "EndpointURL": "aws://ebs-AWSEB-GHNC7CJ52UXO-750787839.ap-south-1.elb.amazonaws.com",
  "CNAME": "react-env.eba-hg6e2ddd.ap-south-1.elasticbeanstalk.com",
  "DateCreated": "2024-09-06T13:01:15.776000+00:00",
  "DateUpdated": "2024-09-06T13:01:33.470000+00:00",
  "Status": "Updating",
  "AbortableOperationInProgress": true,
  "Health": "Grey",
  "Tier": {
    "Name": "WebServer"
  }
}
```

Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```
[cloudshell-user@ip-10-132-83-167 ~]$ aws elasticbeanstalk describe-environments --environment-names react-env
{
  "Environments": [
    {
      "EnvironmentName": "react-env",
      "EnvironmentId": "e-mjhkdecbu2",
      "ApplicationName": "simple-reactjs-app",
      "VersionLabel": "1",
      "SolutionStackName": "64bit Amazon Linux 2023 v6.2.0 running Node.js 18",
      "PlatformArn": "arn:aws:elasticbeanstalk:ap-south-1::platform/Node.js_18_running_on_64bit_Amazon_Linux_2023/6.2.0",
      "Description": "Environment created from the EB CLI using \"eb create\"",
      "EndpointURL": "awseb-AWSEB-GHN:7CJ52UxO-750787839.ap-south-1.elb.amazonaws.com",
      "CNAME": "react-env.eba-hg6e2ddd.ap-south-1.elasticbeanstalk.com",
      "DateCreated": "2024-09-06T11:09:15.776000+00:00",
      "DateUpdated": "2024-09-06T13:01:59.834000+00:00",
      "Status": "Ready",
      "AbortableOperationInProgress": false,
      "Health": "Red",
      "HealthStatus": "Degraded",
      "Tier": {
        "Name": "WebServer",
        "Type": "Standard",
        "Version": "1.0"
      },
      "EnvironmentLinks": [],
      "EnvironmentArn": "arn:aws:elasticbeanstalk:ap-south-1:891612578515:environment/simple-reactjs-app/react-env"
    }
  ]
}
```

**Elastic Beanstalk**

Environment successfully launched.

Environment update successfully completed.

react-env.eba-hg6e2ddd.ap-south-1.elasticbeanstalk.com	simple-reactjs-app
1	Platform state Supported

Events | Health | Logs | Monitoring | Alarms | Managed updates | Tags

**Events (30)**

Time	Type	Details
September 6, 2024 18:31:59 (UTC+5:30)	INFO	Environment update completed successfully.
September 6, 2024 18:31:59 (UTC+5:30)	INFO	New application version was deployed to running EC2 instances.
September 6, 2024 18:31:52 (UTC+5:30)	INFO	Instance deployment completed successfully.
September 6, 2024 18:31:38 (UTC+5:30)	INFO	Deploying new version to instance(s).
September 6, 2024 18:31:33 (UTC+5:30)	INFO	Environment update is starting.

CloudShell | Feedback

**Simple React App**

John Smith

jsmith@test.com  
123456789

Click to View Details

ABCD

abcd@test.com  
987654321

Click to View Details

Tyrion

tyrion@test.com  
123412345

Click to View Details

Name : John Smith  
Email : jsmith@test.com  
Phone : 123456789  
City : bangalore  
State : karnataka  
Country : India  
Organization : Company 1  
Job Profile : Software Developer  
Additional Info : Has Bought a lot of products before and a high Value Customer

## Step 4: Secure UI Images Using S3 and CloudFront

## 4.1 Upload UI Images to S3:

- Name the bucket: **myapp-ui-images**.
- Upload your static UI content here.

## 4.2 Set Up CloudFront:

- Go to **CloudFront Console**.
- Create a distribution with **S3 bucket myapp-ui-images** as the origin.
- Use the CloudFront domain in your React app to access images.

The image contains two screenshots of the AWS CloudFront console.

**Screenshot 1: Create bucket - General configuration**

This screenshot shows the "Create bucket" wizard. In the "General configuration" step, the "Bucket name" field is filled with "myapp-ui-images". The "AWS Region" is set to "Asia Pacific (Mumbai) ap-south-1". The "Object Ownership" section indicates that objects will be owned by the bucket owner.

**Screenshot 2: Create distribution - Origin**

This screenshot shows the "Create distribution" wizard. In the "Origin" step, the "Origin domain" field contains "myapp-ui-images.s3.ap-south-1.amazonaws.com". The "Name" field is also filled with the same URL. The "Origin access" section has "Public" selected, which is described as allowing public access to the bucket.

The screenshot shows the AWS CloudFront console under the 'Distributions' section. A single distribution is listed:

ID	Description	Type	Domain name	Alternate do...	Origins	Status	Last modified
ED4G5ISAW6G8Z	-	Production	d3gzujez8xgt4...	-	myapp-ui-images.s3.a	Enabled	September 6, 20...

The screenshot shows a code editor with the file 'App.js' open. The code is as follows:

```

1 import React, { Component } from 'react';
2 import logo from './Logo.svg';
3 import './App.css';
4 import Customers from './Customers';
5 import { BrowserRouter as Router, Switch, Route, Redirect } from 'react-router-dom';
6
7
8 class App extends Component {
9   render() {
10     console.log("Host URL:" + process.env.PUBLIC_URL);
11     return (
12       <Router basename={process.env.PUBLIC_URL}>
13         <div className="App">
14           <header className="App-header">
15             <img src={logo} className="App-logo" alt="Logo" />
16             <h1>Simple React App</h1>
17           </header>
18           <Switch>
19             <Route exact path="/" render={() => (
20               <Redirect to="/customerlist"/>
21             )}/>
22             <Route exact path="/customerlist" component={Customers} />
23           </Switch>
24         </div>
25       </Router>
26     );
27   }
28 }
29 
```

The URL for the logo image is set to a CloudFront endpoint: `https://d3gzujez8xgt4.cloudfront.net/logo.svg`.

The screenshot shows a code editor with the file 'App.js' open. The code is identical to the previous screenshot, with the logo URL pointing to a CloudFront endpoint.

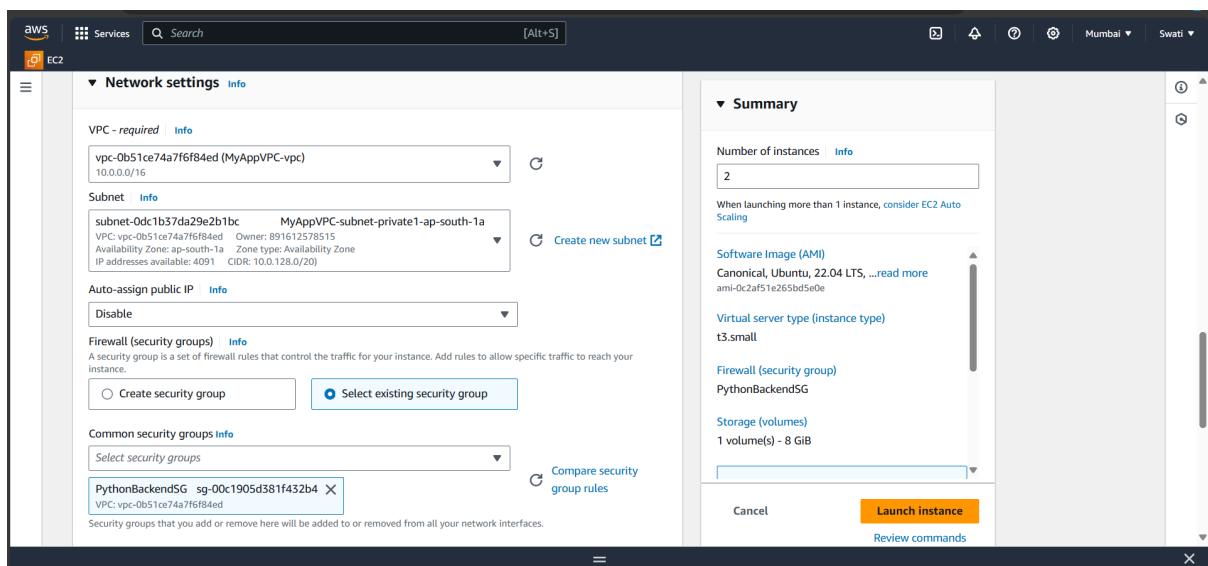
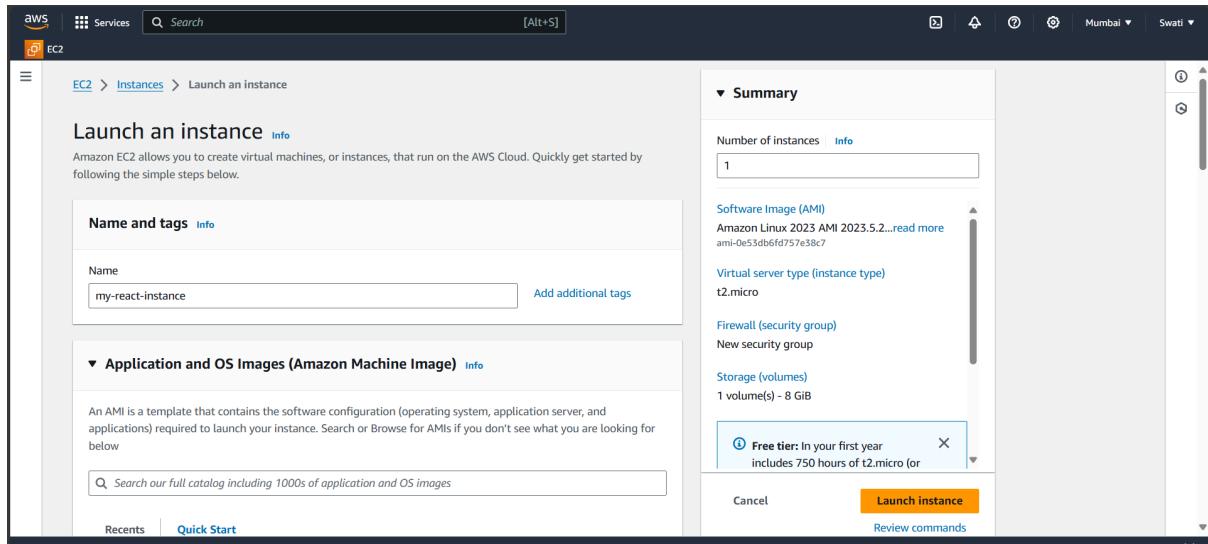
## Step 5: Deploy Python API Backend Using EC2

## 5.1 Launch EC2 Instances:

- Go to **EC2 Console**.
- Launch 2 EC2 instances in **PrivateSubnet**.
- Instance type: **t3.small**.
- Security Group: **PythonBackendSG**.
- Set up your Python API by cloning your GitHub repo or uploading the Python API code.

## 5.2 Auto-Scaling for Python API:

- Create an **Auto Scaling Group** for Python API EC2 instances.
- Set **min instances** to 2 and **max instances** to 4.
- Add scaling based on CPU utilization (>70%).



aws Services Search [Alt+S] Mumbai Swati

ubuntu@ip-10-0-14-138:~\$ sudo apt update -y  
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
113 packages can be upgraded. Run 'apt list --upgradable' to see them.  
ubuntu@ip-10-0-14-138:~\$ sudo apt install python3 -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
python3 is already the newest version (3.12.3-0ubuntu2).  
python3 set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 113 not upgraded.  
ubuntu@ip-10-0-14-138:~\$ sudo apt install python3-pip  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
python3-pip is already the newest version (24.0+dfsg-lubuntul).  
0 upgraded, 0 newly installed, 0 to remove and 113 not upgraded.  
ubuntu@ip-10-0-14-138:~\$

i-0acb961d06c876937 (ec2)  
PublicIPs: 13.233.216.63 PrivateIPs: 10.0.14.138

aws Services Search [Alt+S] Mumbai Swati

ubuntu@ip-10-0-14-138:~\$ git clone https://github.com/Swati-Adhe/Cloud\_Native\_Monitoring\_Application.git  
Cloning into 'Cloud\_Native\_Monitoring\_Application'...  
remote: Enumerating objects: 20, done.  
remote: Counting objects: 100% (20/20), done.  
remote: Compressing objects: 100% (18/18), done.  
remote: Total 20 (delta 7), reused 0 (delta 0), pack-reused 0 (from 0)  
Receiving objects: 100% (20/20), 6.90 KiB | 1.72 MiB/s, done.  
Resolving deltas: 100% (7/7), done.  
ubuntu@ip-10-0-14-138:~\$ ls  
Cloud\_Native\_Monitoring\_Application app.py  
ubuntu@ip-10-0-14-138:~\$ cd Cloud\_Native\_Monitoring\_Application/  
ubuntu@ip-10-0-14-138:~/Cloud\_Native\_Monitoring\_Application\$

i-0acb961d06c876937 (ec2)  
PublicIPs: 13.233.216.63 PrivateIPs: 10.0.14.138

aws Services Search [Alt+S] Mumbai Swati

(venv) ubuntu@ip-10-0-14-138:~\$ cd Cloud\_Native\_Monitoring\_Application/  
(venv) ubuntu@ip-10-0-14-138:~/Cloud\_Native\_Monitoring\_Application\$ pip install -r requirements.txt  
Collecting Flask==2.2.3 (from -r requirements.txt (line 1))  
 Downloading Flask-2.2.3-py3-none-any.whl.metadata (3.9 kB)  
Collecting MarkupSafe==2.1.2 (from -r requirements.txt (line 2))  
 Downloading MarkupSafe-2.1.2.tar.gz (19 kB)  
 Installing build dependencies ... done  
 Getting requirements to build wheel ... done  
 Preparing metadata (pyproject.toml) ... done  
Collecting Werkzeug==2.2.3 (from -r requirements.txt (line 3))  
 Downloading Werkzeug-2.2.3-py3-none-any.whl.metadata (4.4 kB)  
Collecting itsdangerous==2.1.2 (from -r requirements.txt (line 4))  
 Downloading itsdangerous-2.1.2-py3-none-any.whl.metadata (2.9 kB)  
Collecting psutil==5.8.0 (from -r requirements.txt (line 5))  
 Downloading psutil-5.8.0.tar.gz (470 kB)

470.9/470.9 kB 32.3 MB/s eta 0:00:00

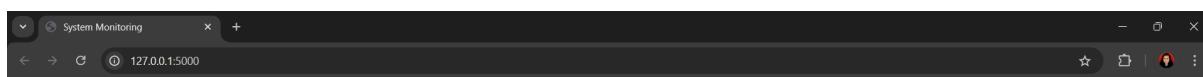
Installing build dependencies ... done  
 Getting requirements to build wheel ... done  
 Preparing metadata (pyproject.toml) ... done  
Collecting plotly==5.5.0 (from -r requirements.txt (line 6))  
 Downloading plotly-5.5.0-py2.py3-none-any.whl.metadata (7.1 kB)  
Collecting tenacity==8.0.1 (from -r requirements.txt (line 7))  
 Downloading tenacity-8.0.1-py3-none-any.whl.metadata (1.1 kB)  
Collecting boto3==1.9.148 (from -r requirements.txt (line 8))  
 Downloading boto3-1.9.148-py2.py3-none-any.whl.metadata (4.8 kB)  
Collecting kubernetes==10.0.1 (from -r requirements.txt (line 9))  
 Downloading kubernetes-10.0.1-py2.py3-none-any.whl.metadata (1.5 kB)  
Collecting Jinja2>=3.0 (from Flask==2.2.3->-r requirements.txt (line 1))  
 Downloading Jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)

i-0acb961d06c876937 (ec2)  
PublicIPs: 13.233.216.63 PrivateIPs: 10.0.14.138

AWS CloudWatch Terminal window showing application logs for a Flask app running on EC2. The logs indicate the app is serving on all addresses (0.0.0.0) and port 5000, with a debugger active. A warning message states: "WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead." The terminal also shows the EC2 instance ID (i-0acb961d06c876937) and its public and private IP addresses.

```
(venv) ubuntu@ip-10-0-14-138:~/Cloud_Native_Monitoring_Application$ python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.0.14.138:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 136-701-781
```

i-0acb961d06c876937 (ec2)  
PublicIPs: 13.233.216.63 PrivateIPs: 10.0.14.138



## System Monitoring

High CPU or Memory Detected, scale up!!!

AWS Auto Scaling Groups creation wizard, Step 1: Choose launch template. The "Name" field is populated with "PythonAPIAutoScalingGroup". A note indicates that for accounts created after May 31, 2023, creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1  
Choose launch template

Step 2  
Choose instance launch options

Step 3 - optional  
Configure advanced options

Step 4 - optional  
Configure group size and scaling

Step 5 - optional  
Add notifications

Step 6 - optional  
Add tags

Step 7  
Review

**Choose launch template** Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

**Name**

Auto Scaling group name  
Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.

**Launch template** Info

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

**Review**

**Launch template**  
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

**template**:

**Create a launch template**

**Version**: Default (1)

**Create a launch template version**

Description	Launch template <b>template</b> <input type="text"/> lt-04c2563eeeb72244d	Instance type <b>t2.micro</b>
AMI ID <b>ami-0522ab6e1ddcc7055</b>	Security groups	Request Spot Instances <b>No</b>
Key pair name <b>key-pair</b>	Security group IDs <b>sg-00c1905d381f1432b4</b> <input type="text"/>	

**Additional details**

Storage (volumes)	Date created <b>Fri Sep 06 2024 18:03:05 GMT+0530</b> (India Standard Time)
-------------------	---

**Step 6 - optional** [Add tags](#)

**Step 7** [Review](#)

**Network Info**

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

**VPC**  
Choose the VPC that defines the virtual network for your Auto Scaling group.  
**vpc-0b51ce74a7f6f84ed (MyAppVPC-vpc)**

**Create a VPC**

**Availability Zones and subnets**  
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

**Select Availability Zones and subnets**

ap-south-1a   subnet-0717a190a547858a5 (MyAppVPC-subnet-public1-ap-south-1a) 10.0.0.0/20
ap-south-1a   subnet-0dc1b37da29e2b1bc (MyAppVPC-subnet-private1-ap-south-1a) 10.0.128.0/20

**Create a subnet**

**Step 4 - optional** [Configure group size and scaling](#)

**Step 5 - optional** [Add notifications](#)

**Step 6 - optional** [Add tags](#)

**Step 7** [Review](#)

**Desired capacity type**  
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

**Units (number of instances)**

**Desired capacity**  
Specify your group size.  
**2**

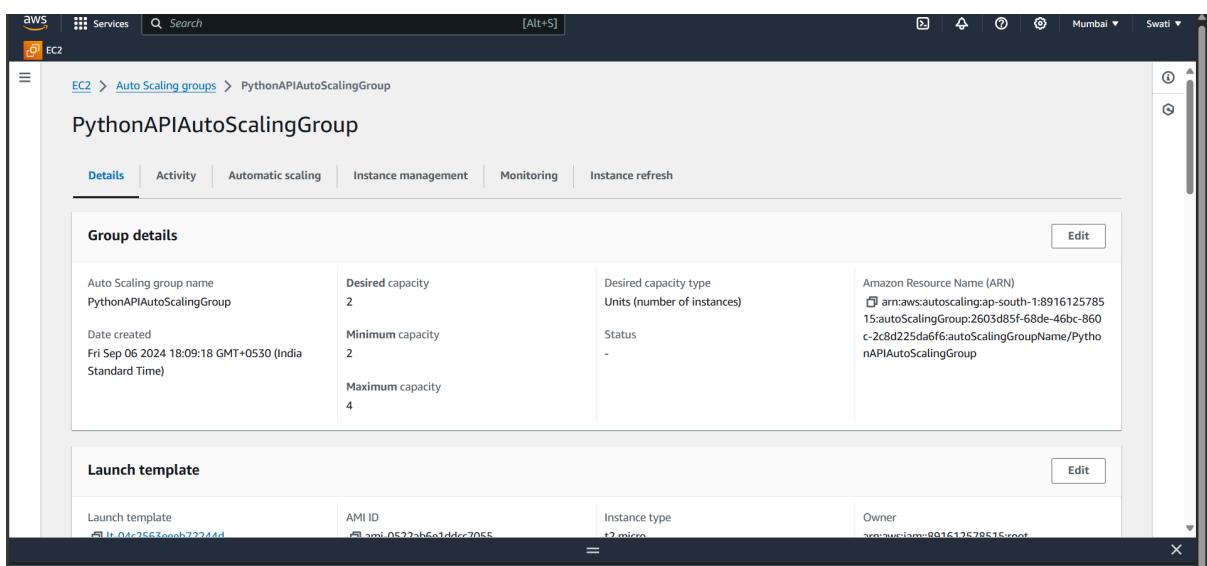
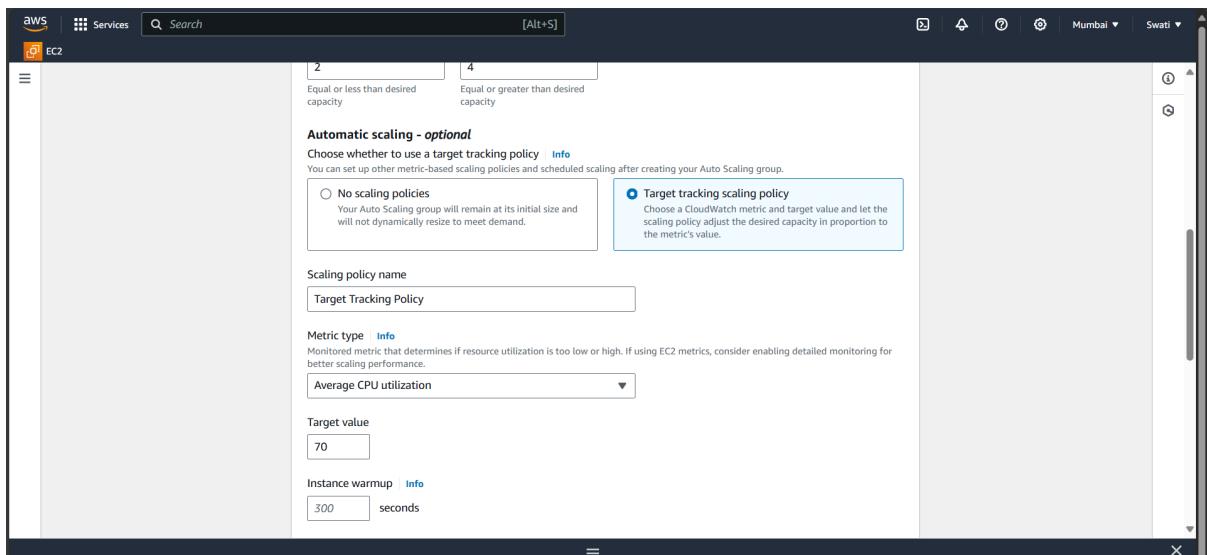
**Scaling Info**  
You can resize your Auto Scaling group manually or automatically to meet changes in demand.

**Scaling limits**  
Set limits on how much your desired capacity can be increased or decreased.

<b>Min desired capacity</b> <b>2</b> <input type="text"/>	<b>Max desired capacity</b> <b>4</b> <input type="text"/>
Equal or less than desired capacity	Equal or greater than desired capacity

**Automatic scaling - optional**  
Choose whether to use a target tracking policy [Info](#)  
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

<input type="radio"/> <b>No scaling policies</b> Your Auto Scaling group will remain at its initial size and	<input checked="" type="radio"/> <b>Target tracking scaling policy</b> Choose a CloudWatch metric and target value and let the
---	---



## Step 6: Set Up RDS for Database

### 6.1 Create an RDS Instance (MySQL or PostgreSQL):

- Go to **RDS Console**.
- Click **Create Database**.
  - Select MySQL or PostgreSQL.
  - Instance type: **db.t3.micro**.
  - Multi-AZ Deployment: Enable for high availability.
  - Public Access: No (private only).
  - Security Group: Attach **DatabaseSG**.

### 6.2 Set Up Automatic Backups for RDS:

- In the **Backup** section, enable **Automated Backups** with a retention period of days.

- Take manual RDS snapshots regularly.

**Create database**

**Choose a database creation method** [Info](#)

**Standard create**  
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

**Easy create**  
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

**Engine options**

**Engine type** [Info](#)

**Aurora (MySQL Compatible)** 

**Aurora (PostgreSQL Compatible)** 

**MySQL** 

**MariaDB** 

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

**DB cluster identifier** [Info](#)  
Enter a name for your DB cluster. The name must be unique across all DB clusters owned by your AWS account in the current AWS Region.

The DB cluster identifier is case-insensitive, but is stored as all lowercase (as in "mydbcluster"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

**Credentials Settings**

**Master username** [Info](#)  
Type a login ID for the master user of your DB instance.

1 to 32 alphanumeric characters. The first character must be a letter.

**Credentials management**  
You can use AWS Secrets Manager or manage your master user credentials.

**Managed in AWS Secrets Manager - most secure**  
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

**Self managed**  
Create your own password or have RDS create a password that you manage.

**Auto generate password**  
Amazon RDS can generate a password for you, or you can specify your own password.

**Master password** [Info](#)

**Password strength** Weak  
Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / " @

**Confirm master password** [Info](#)

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

AWS Services Search [Alt+S] Mumbai Swati EC2

**Virtual private cloud (VPC) Info**  
Choose the VPC. The VPC defines the virtual networking environment for this DB cluster.  
**MyAppVPC-vpc (vpc-0b51ce74a7f6f84d)**  
2 Subnets, 1 Availability zones

Only VPCs with a corresponding DB subnet group are listed.

**After a database is created, you can't change its VPC.**

**DB subnet group Info**  
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB cluster can use in the VPC that you selected.  
**Create new DB Subnet Group**

**Public access Info**  
RDS assigns a public IP address to the cluster. Amazon EC2 instances and other resources outside of the VPC can connect to your cluster. Resources inside the VPC can also connect to the cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.  
**No**  
RDS doesn't assign a public IP address to the cluster. Only Amazon EC2 instances and other resources inside the VPC can connect to your cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

**VPC security group (firewall) Info**  
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.  
**Choose existing** Choose existing VPC security groups  
**Create new** Create new VPC security group

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

AWS Services Search [Alt+S] Mumbai Swati EC2

**Amazon RDS** Databases

**Consider creating a Blue/Green Deployment to minimize downtime during upgrades**  
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. [RDS User Guide](#) [Aurora User Guide](#)

**Databases (3)**

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommend
mydatabase	Available	Regional cluster	Aurora MySQL	ap-south-1	2 instances	
mydatabase-instance-1	Available	Writer instance	Aurora MySQL	ap-south-1c	db.t3.medium	
mydatabase-instance-1-ap-south-1a	Available	Reader instance	Aurora MySQL	ap-south-1a	db.t3.medium	

AWS Services Search [Alt+S] Mumbai Swati EC2

**Amazon RDS**

**Pending maintenance (0)**

No pending maintenance available

**Backup**

Automated backups	Earliest restorable time	Latest restore time
Enabled (1 Day)	September 06, 2024, 18:48 (UTC+05:30)	September 06, 2024, 18:48 (UTC+05:30)
Copy tags to snapshots		Backup window
Enabled		19:53-20:23 UTC (GMT)

**Backtrack**

Backtrack window  
Disabled

AWS Services Search [Alt+S] Mumbai Swati

RDS > Snapshots > Take snapshot

## Take DB Snapshot

**Preferences**

To take a DB Snapshot, choose a database and name your DB Snapshot.

**Snapshot type**

DB instance

DB cluster

**DB cluster**

mydatabase

**Snapshot name**

Identifier for the DB Snapshot.

snap-1

Snapshot identifier is case insensitive, but stored as all lower-case, as in "mysnapshot". Cannot be null, empty, or blank. Must contain from 1 to 255 alphanumeric characters or hyphens. First character must be a letter. Cannot end with a hyphen or contain two consecutive hyphens.

Cancel Take snapshot

AWS Services Search [Alt+S] Mumbai Swati

Amazon RDS X

Successfully created snapshot snap-1. View details

RDS > Snapshots

## Snapshots

Manual System Shared with me Public Backup service Exports in Amazon S3

**Manual snapshots (1)**

Snapshot name	Engine version	DB instance or cluster	Snapshot creation time
snap-1	8.0.mysql_aurora.3.05.2	mydatabase	September 06, 2024, 18:51 (UTC+05:30)

AWS Services Search [Alt+S] Mumbai Swati

Amazon RDS X

Backup

Automated backups Enabled (1 Day) Earliest restorable time September 06, 2024, 18:48 (UTC+05:30) Latest restore time September 06, 2024, 19:28 (UTC+05:30)

Copy tags to snapshots Enabled Backup window 19:53-20:23 UTC (GMT)

Backtrack

Backtrack window Disabled

Snapshots (1)

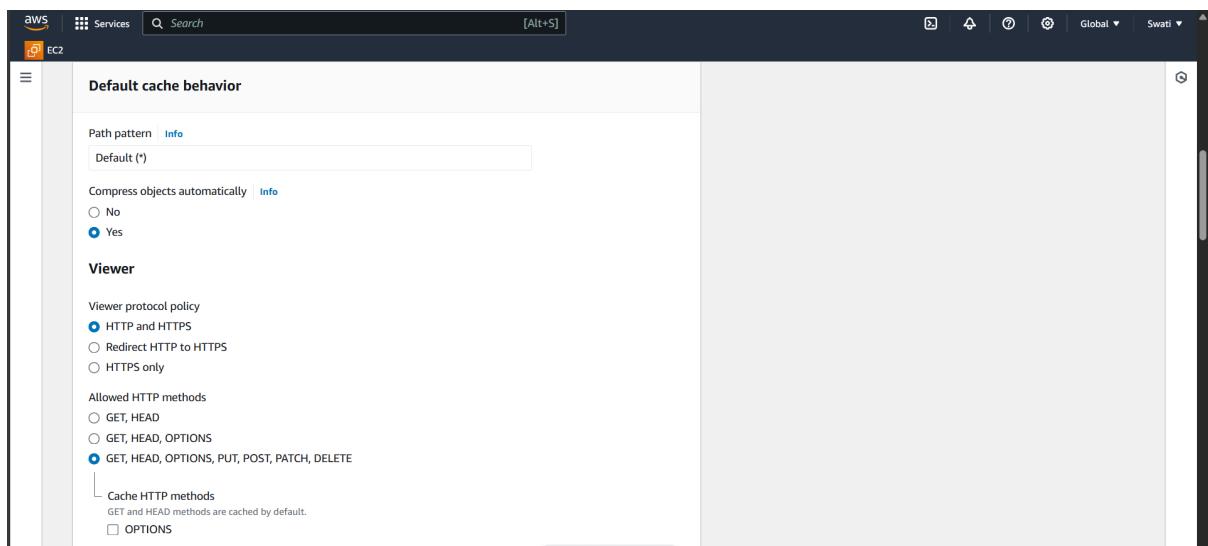
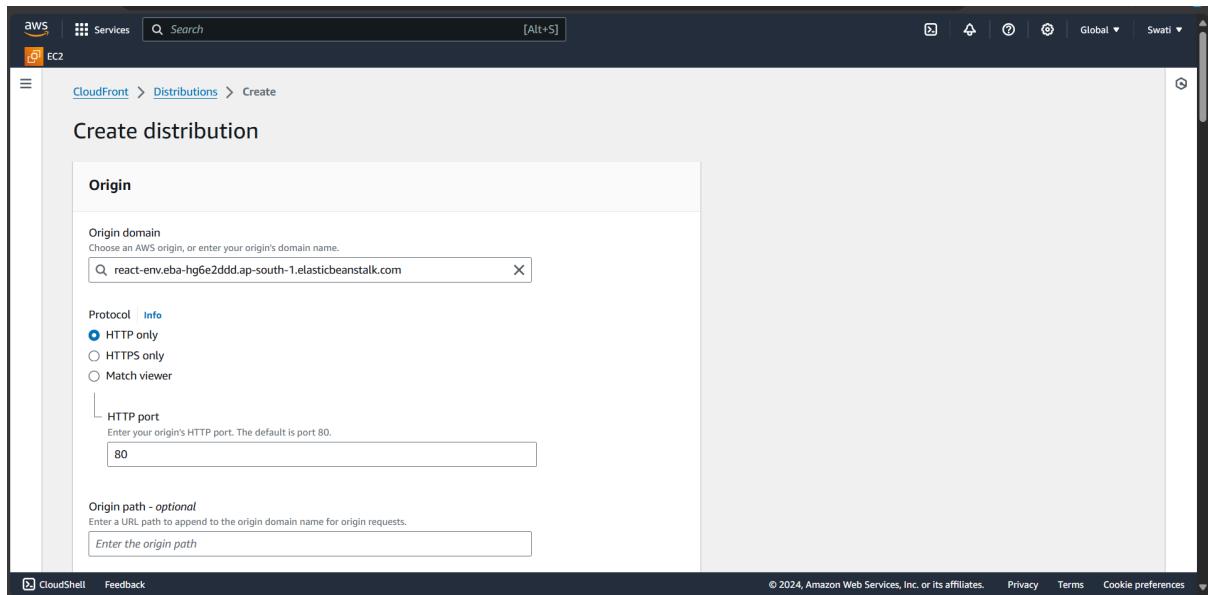
Snapshot name	Snapshot creation time	Status	Snapshot type
snap-1	September 06, 2024, 18:57 (UTC+05:30)	Available	Manual

Exports In Amazon S3 (0)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Step 7: Enable Global Caching with CloudFront

- You have already set up **CloudFront** for your S3 images.
- Similarly, for the entire web application (React frontend), create another CloudFront distribution:
  - Origin Domain: Elastic Beanstalk endpoint.
  - Enable caching for the entire application for faster access globally.



AWS Services Search [Alt+S] Global Swati

EC2

Cache settings

Cache HTTP methods: GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Cache key and origin requests

Restrict viewer access: If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content.

Cache policy and origin request policy (recommended)

Legacy cache settings

Cache policy: Choose an existing cache policy or create a new one.

UseOriginCacheControlHeaders Recommended for Elastic Beanstalk Policy for origins that return Cache-Control headers. Query strings are not included in the ...

Create cache policy View policy

Origin request policy - optional: Choose an existing origin request policy or create a new one.

AllViewer Recommended for Elastic Beanstalk Policy to forward all parameters in viewer requests

Create origin request policy View policy

AWS Services Search [Alt+S] Global Swati

CloudFront > Distributions > E2XULPBUP43HGX

### E2XULPBUP43HGX

View metrics

General Security Origins Behaviors Error pages Invalidations Tags

**Details**

Distribution domain name d3j7oveOnj3775.cloudfront.net	ARN arn:aws:cloudfront::891612578515:distribution/E2XULPBUP43HGX	Last modified Deploying
---	---	----------------------------

**Settings**

Description -	Alternate domain names -	Standard logging Off Cookie logging Off Default root object -
------------------	-----------------------------	--

AWS Services Search [Alt+S] Global Swati

CloudFront > Distributions > E2XULPBUP43HGX

### E2XULPBUP43HGX

View metrics

General Security Origins Behaviors Error pages Invalidations Tags

**Behaviors**

Filter behaviors by property or value

Preced...	Path pattern	Origin or origin group	Viewer protocol policy	Cache policy name	Origin request policy ...	Response headers pol...
0	/*.js	react-env.eba-hg6e2d...	HTTP and HTTPS	Managed-CachingOptimize...	Managed-AllViewer	-
1	/*.jpg	react-env.eba-hg6e2d...	HTTP and HTTPS	Managed-CachingOptimize...	Managed-AllViewer	-
2	/*.jpeg	react-env.eba-hg6e2d...	HTTP and HTTPS	Managed-CachingOptimize...	Managed-AllViewer	-
3	/*.gif	react-env.eba-hg6e2d...	HTTP and HTTPS	Managed-CachingOptimize...	Managed-AllViewer	-
4	/*.css	react-env.eba-hg6e2d...	HTTP and HTTPS	Managed-CachingOptimize...	Managed-AllViewer	-
5	/*.png	react-env.eba-hg6e2d...	HTTP and HTTPS	Managed-CachingOptimize...	Managed-AllViewer	-
6	Default (*)	react-env.eba-hg6e2d...	HTTP and HTTPS	UseOriginCacheControlHea...	Managed-AllViewer	-

## Step 8: Monitor Performance

- Use **AWS CloudWatch** to monitor the performance and health of your CloudFront distribution and Elastic Beanstalk application.

The screenshot shows the AWS CloudWatch Alarms page with four alarms listed:

- Name:** TargetTracking-PythonAPIAutoScalingGroupLow  
**State:** In alarm  
**Last state update (UTC):** 2024-09-06 12:52:11  
**Condition:** CPUUtilization < 49 for 15 datapoints within 15 minutes  
**Action:** Actions enabled
- Name:** TargetTracking-PythonAPIAutoScalingGroupHigh  
**State:** OK  
**Last state update (UTC):** 2024-09-06 12:42:00  
**Condition:** CPUUtilization > 70 for 3 datapoints within 3 minutes  
**Action:** Actions enabled
- Name:** awseb-e-mjhkdecbu2-stack-AWSEBCloudwatchAlarmHigh  
**State:** OK  
**Last state update (UTC):** 2024-09-06 11:11:58  
**Condition:** NetworkOut > 6000000 for 1 datapoints within 5 minutes  
**Action:** Actions enabled
- Name:** awseb-e-mjhkdecbu2-stack-AWSEBCloudwatchAlarmLow  
**State:** In alarm  
**Last state update (UTC):** 2024-09-06 11:11:41  
**Condition:** NetworkOut < 2000000 for 1 datapoints within 5 minutes  
**Action:** Actions enabled

