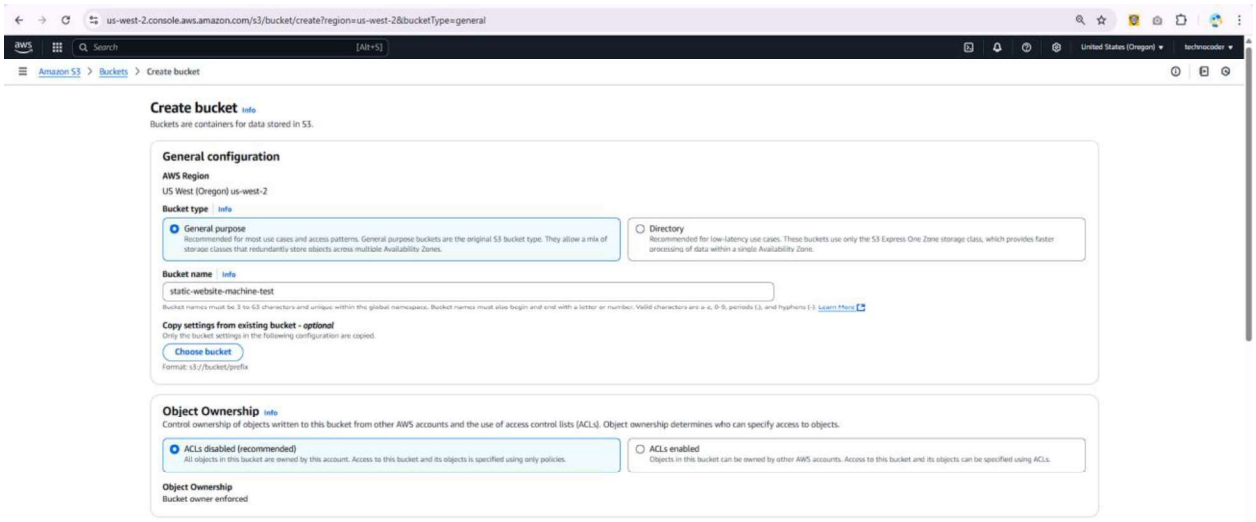
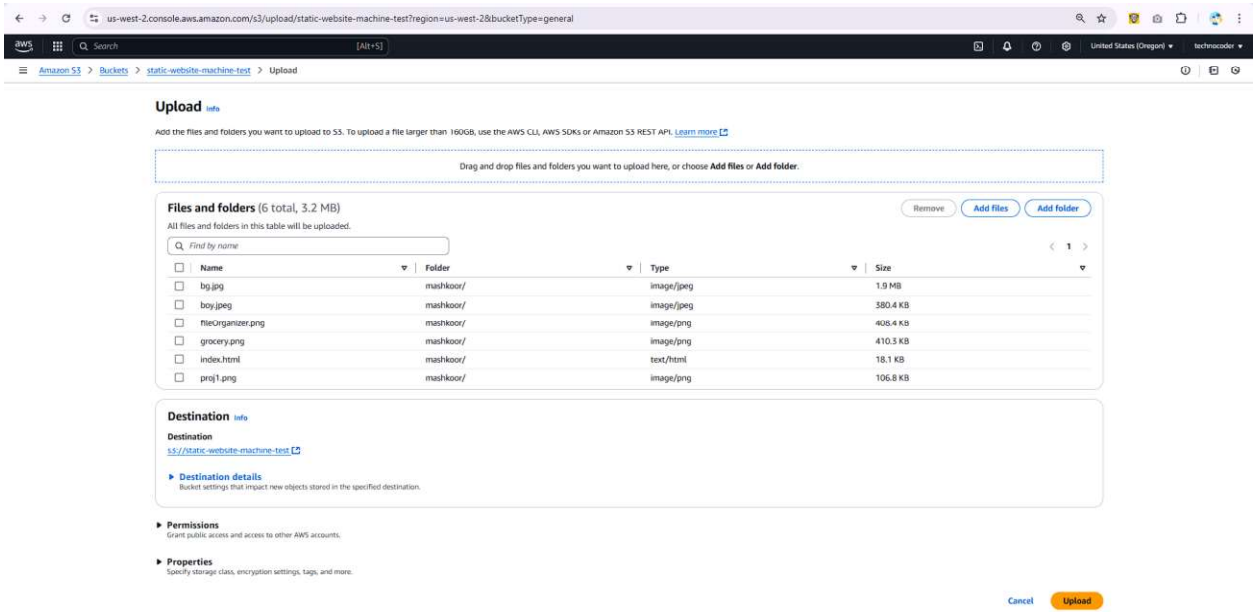
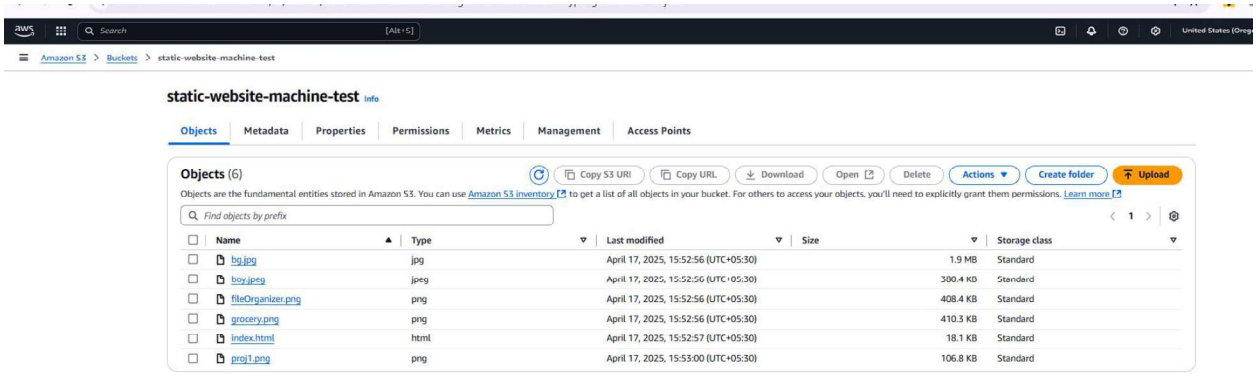
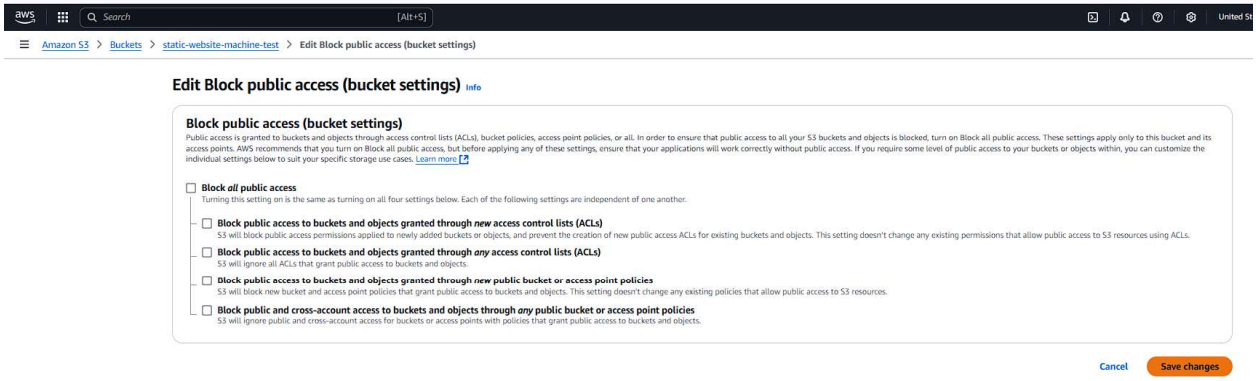
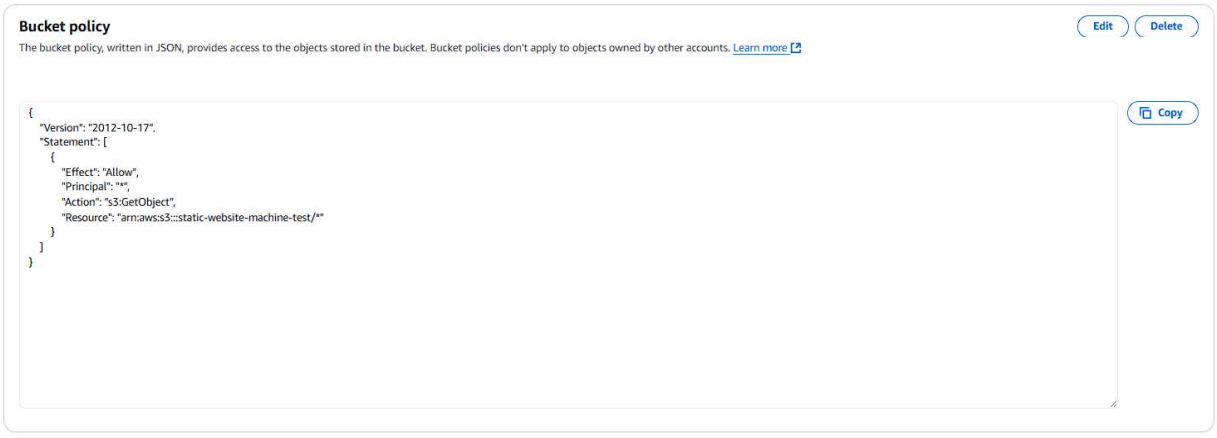
# Task 1: Host a Static Website using Amazon S3

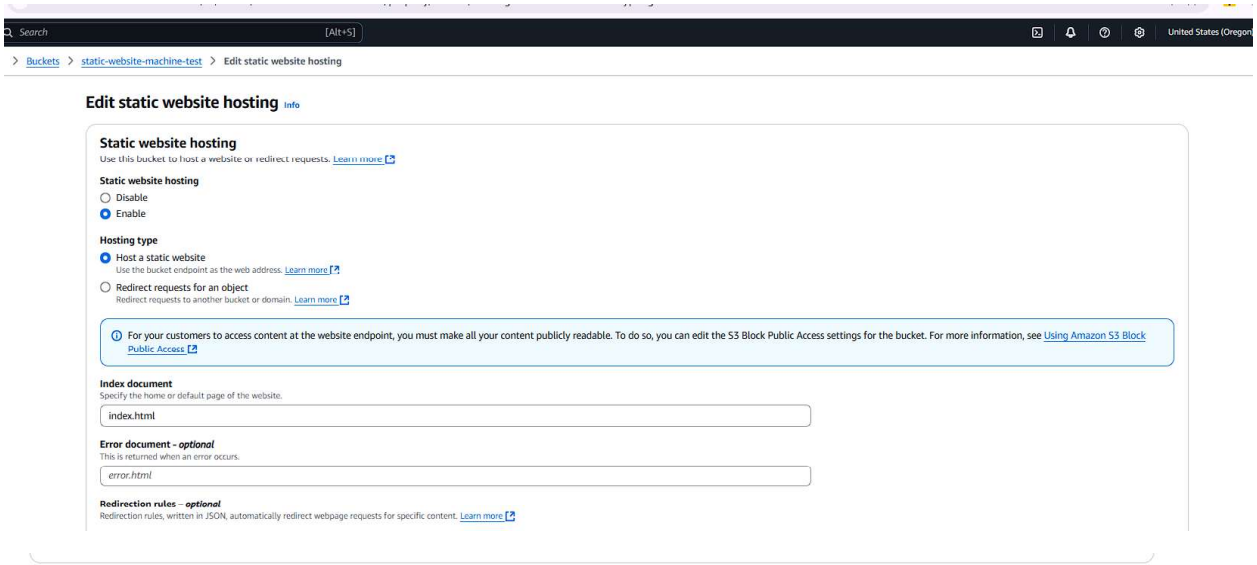


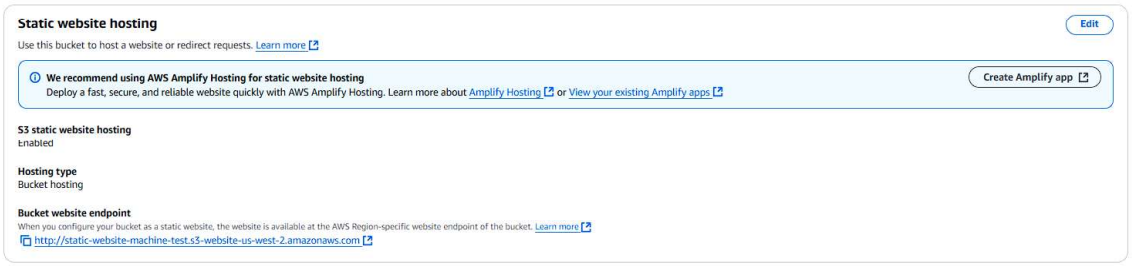



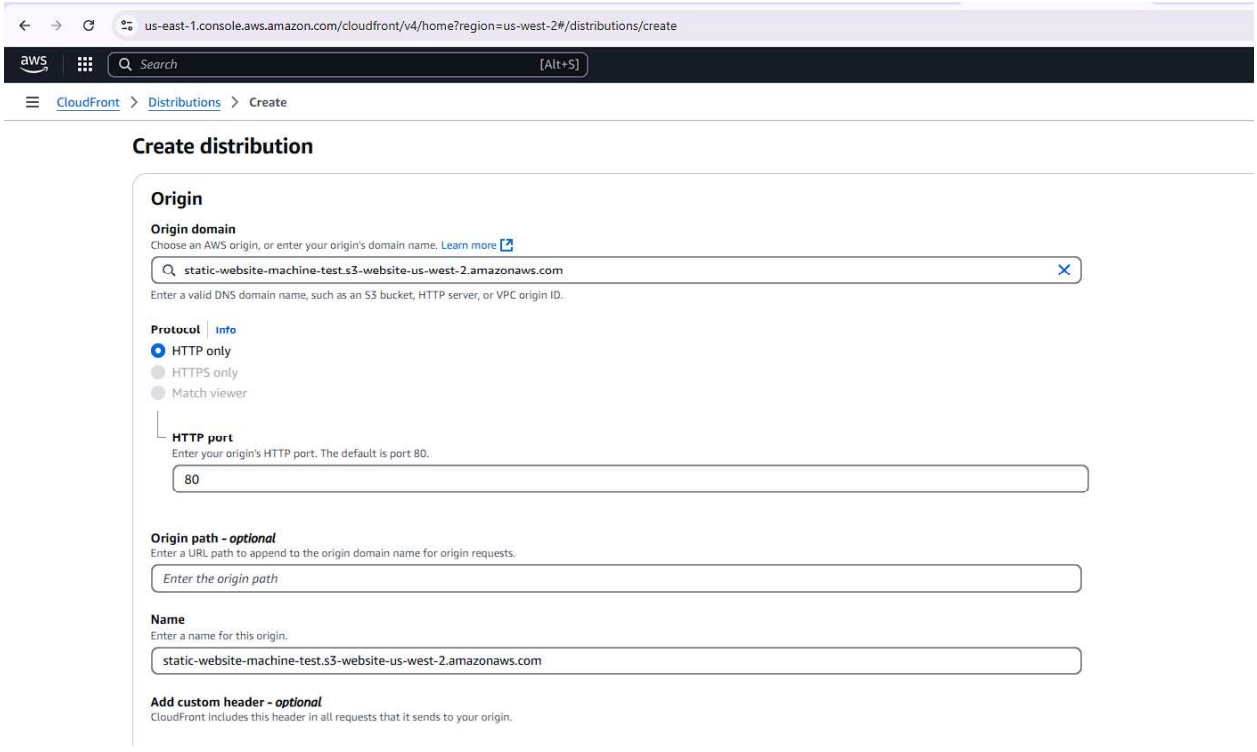

Search
[Alt+S]
Amazon S3 > Buckets > static-website-machine-test
static-website-machine-test Info
Objects
Metadata
Properties
Permissions
Metrics
Management
Access Points
Objects (6)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in 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
Name
Type
Last modified
Size
Storage class
bg.jpg jpg April 17, 2025, 15:52:56 (UTC+05:30) 1.9 MB Standard
boy.jpeg jpeg April 17, 2025, 15:52:56 (UTC+05:30) 380.4 KB Standard
fileOrganizer.png png April 17, 2025, 15:52:56 (UTC+05:30) 408.4 KB Standard
grocery.png png April 17, 2025, 15:52:56 (UTC+05:30) 410.3 KB Standard
index.html html April 17, 2025, 15:52:57 (UTC+05:30) 18.1 KB Standard
proj1.png png April 17, 2025, 15:53:00 (UTC+05:30) 106.8 KB Standard

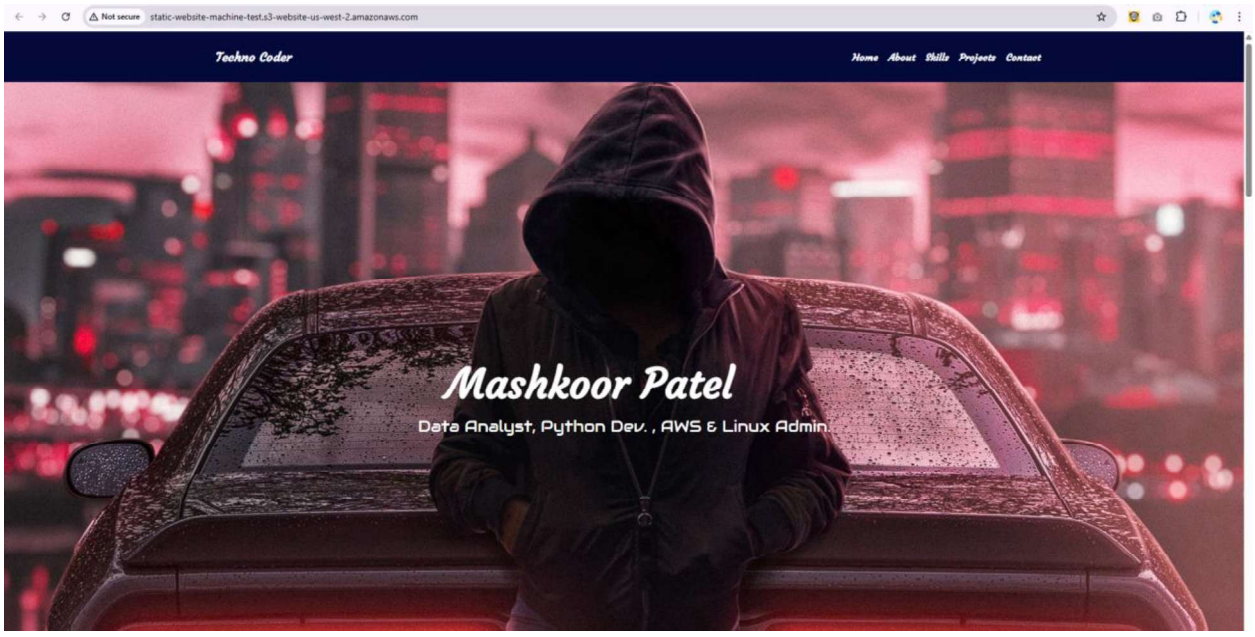

Search
[Alt+S]
Amazon S3 > Buckets > static-website-machine-test > Edit Block public access (bucket settings)
Edit Block public access (bucket settings) Info
Block public access (bucket settings)
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access 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

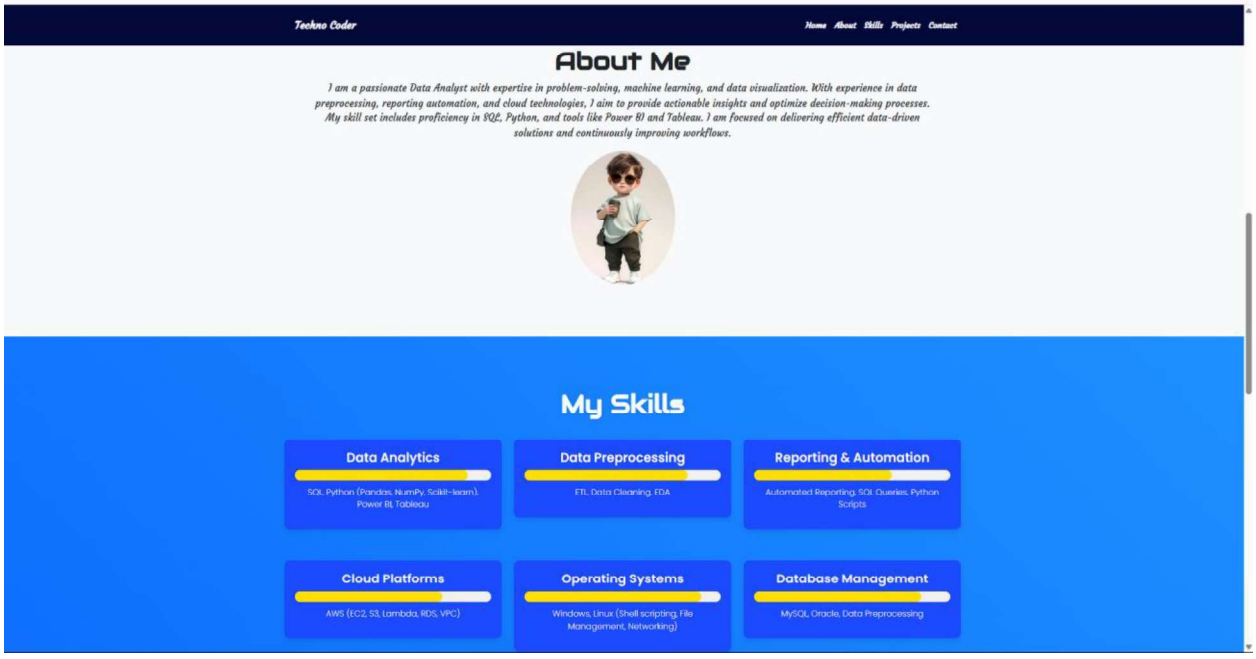

Bucket policy
Edit
Delete
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more
Copy
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Principal": "*",
"Action": "s3:GetObject",
"Resource": "arn:aws:s3:::static-website-machine-test/*"
}
]
}


Search [Alt+S]
United States (Oregon)
Buckets > static-website-machine-test > Edit static website hosting
Edit static website hosting 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.
error.html
Redirection rules – optional
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. Learn more


Static website hosting
Edit
Use this bucket to host a website or redirect requests. Learn more
We recommend using AWS Amplify Hosting for static website hosting
Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more about Amplify Hosting or View your existing Amplify apps
Create Amplify app
S3 static website hosting
Enabled
Hosting type
Bucket hosting
Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more
http://static-website-machine-test.s3-website-us-west-2.amazonaws.com


us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=us-west-2#/distributions/create
aws
Search [Alt+S]
CloudFront > Distributions > Create
Create distribution
Origin
Origin domain
Choose an AWS origin, or enter your origin's domain name. Learn more
static-website-machine-test.s3-website-us-west-2.amazonaws.com
Enter a valid DNS domain name, such as an S3 bucket, HTTP server, or VPC origin ID.
Protocol Info
HTTP only
HTTPS only
Match viewer
HTTP port
Enter your origin's HTTP port. The default is port 80.
80
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.
static-website-machine-test.s3-website-us-west-2.amazonaws.com
Add custom header - optional
CloudFront includes this header in all requests that it sends to your origin.


Not secure
static-website-machine-test.s3-website-us-west-2.amazonaws.com
Techno Coder
Home About Skills Projects Contact
Mashkoor Patel
Data Analyst, Python Dev. , AWS & Linux Admin.


Techno Coder
Home About Skills Projects Contact
About Me
I am a passionate Data Analyst with expertise in problem-solving, machine learning, and data visualization. With experience in data preprocessing, reporting automation, and cloud technologies, I aim to provide actionable insights and optimize decision-making processes. My skill set includes proficiency in SQL, Python, and tools like Power BI and Tableau. I am focused on delivering efficient data-driven solutions and continuously improving workflows.
My Skills
Data Analytics
SQL, Python (Pandas, NumPy, Scikit-learn), Power BI, Tableau
Data Preprocessing
ETL, Data Cleaning, EDA
Reporting & Automation
Automated Reporting, SQL Queries, Python Scripts
Cloud Platforms
AWS (EC2, S3, Lambda, RDS, VPC)
Operating Systems
Windows, Linux (Shell scripting, File Management, Networking)
Database Management
MySQL, Oracle, Data Preprocessing

# My Projects


## Patient Insights & Service Optimization Dashboard

***Problem:*** *Hospitals face challenges analyzing patient data, leading to inefficiencies.*
***Action:*** *Used Power BI to analyze data from 1,000 patients across 5 hospitals.*
***Result:*** *Improved care, reduced costs, and optimized operations.*

## File Organizer Software

***Problem:*** *Managing and organizing a large number of files manually can be time-consuming and lead to cluttered directories.*
***Action:*** *Built a file organizer software using Python with Tkinter for GUI and OS library for file management, providing an intuitive way to sort and organize files.*
***Result:*** *Simplified file organization, saved time, and boosted productivity with an easy-to-use interface.*

## Grocery-Billing-Software

***Problem:*** *Manual grocery billing leads to errors and inefficiencies.*
***Action:*** *Developed a Python-based grocery billing software to automate calculations and inventory management.*
***Result:*** *Streamlined the billing process, reduced errors, and improved overall efficiency.*

Techno Coder

Home About Skills Projects Contact

# Contact Me

Your Name

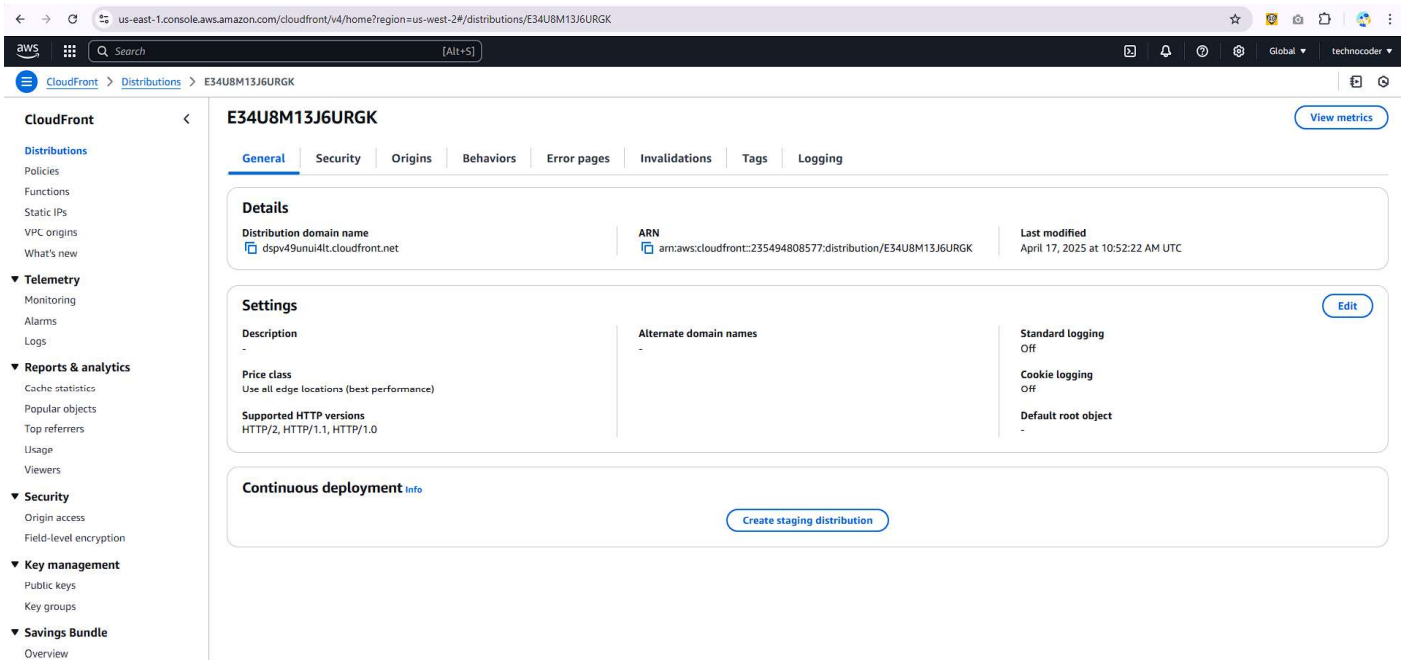
Your Email

Your Message

Send Message



LinkedIn | GitHub




Website link.

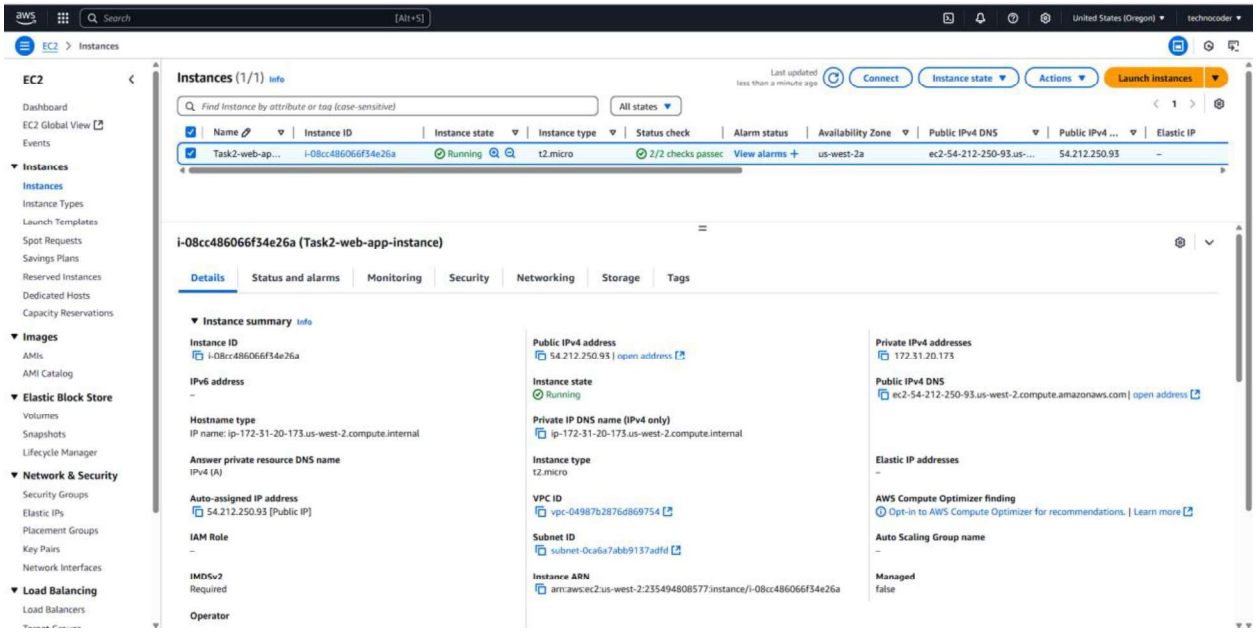
Before:

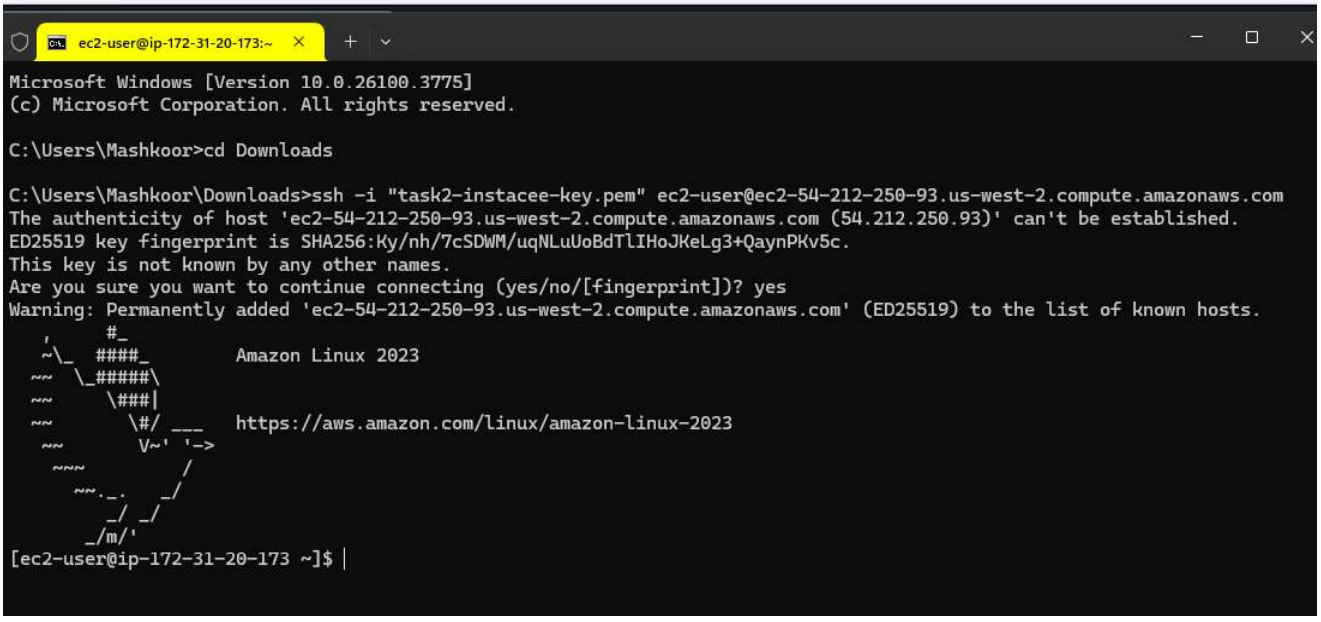
http://static-website-machine-test.s3-website-us-west-2.amazonaws.com/

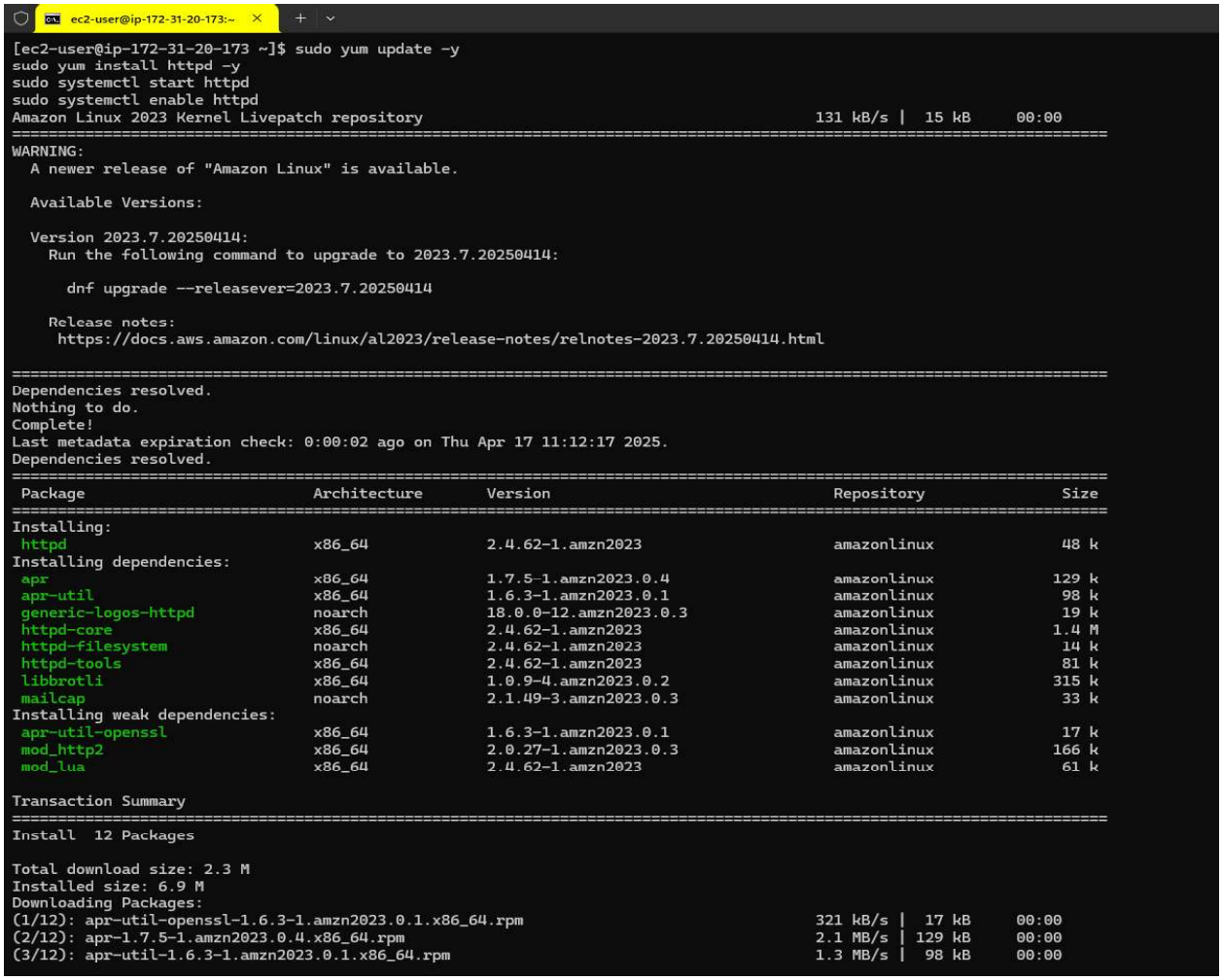
After:

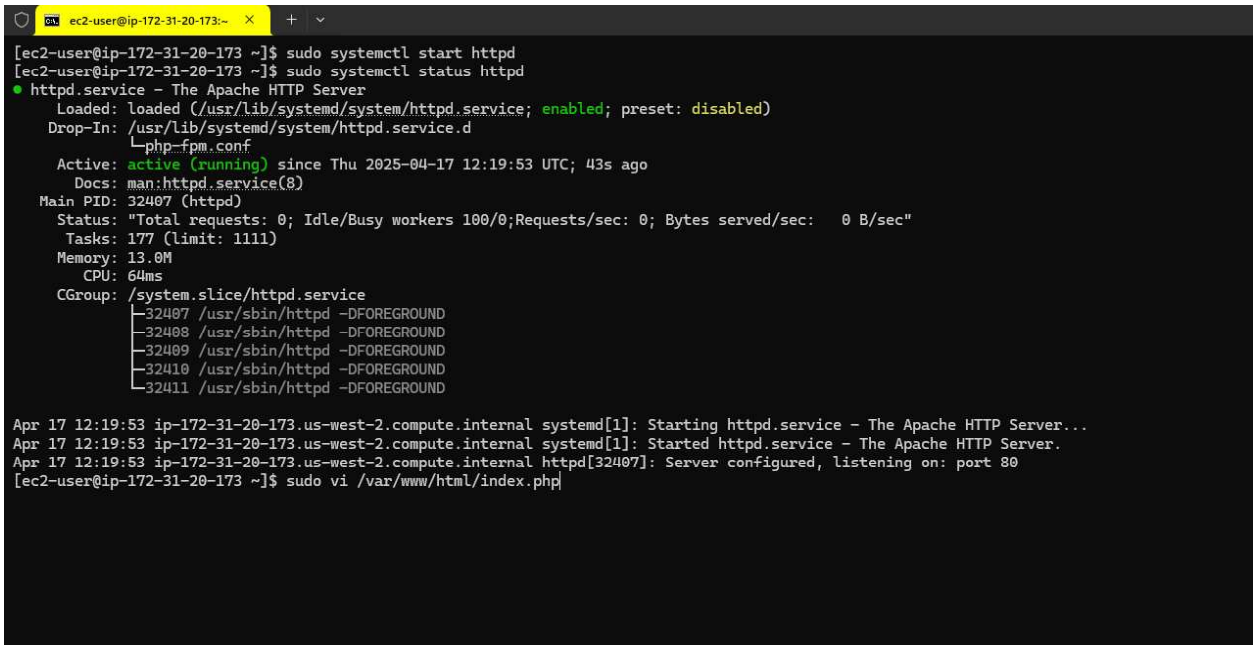
dspv49unui4lt.cloudfront.net

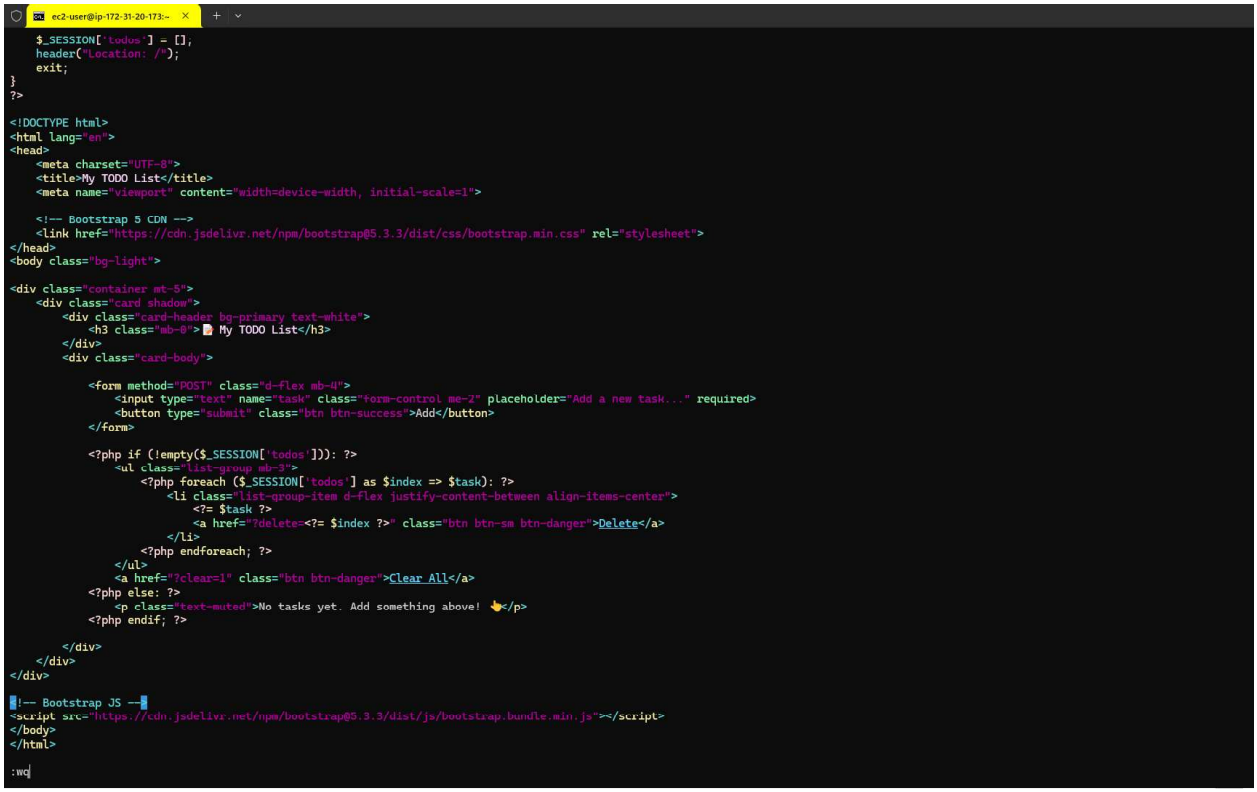
# Task 2: Launch an EC2 Instance and Host a Web Application




ec2-user@ip-172-31-20-173:~
[ec2-user@ip-172-31-20-173 ~]$ sudo yum update -y
sudo yum install httpd -y
sudo systemctl start httpd
sudo systemctl enable httpd
Amazon Linux 2023 Kernel Livepatch repository 131 kB/s | 15 kB 00:00
WARNING:
A newer release of "Amazon Linux" is available.
Available Versions:
Version 2023.7.20250414:
Run the following command to upgrade to 2023.7.20250414:
dnf upgrade --releasever=2023.7.20250414
Release notes:
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.7.20250414.html
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:00:02 ago on Thu Apr 17 11:12:17 2025.
Dependencies resolved.
Package Architecture Version Repository Size
Installing:
httpd x86_64 2.4.62-1.amzn2023 amazonlinux 48 k
Installing dependencies:
apr x86_64 1.7.5-1.amzn2023.0.4 amazonlinux 129 k
apr-util x86_64 1.6.3-1.amzn2023.0.1 amazonlinux 98 k
generic-logos-httpd noarch 18.0.0-12.amzn2023.0.3 amazonlinux 19 k
httpd-core x86_64 2.4.62-1.amzn2023 amazonlinux 1.4 M
httpd-filesystem noarch 2.4.62-1.amzn2023 amazonlinux 14 k
httpd-tools x86_64 2.4.62-1.amzn2023 amazonlinux 81 k
libbrotli x86_64 1.0.9-4.amzn2023.0.2 amazonlinux 315 k
mailcap noarch 2.1.49-3.amzn2023.0.3 amazonlinux 33 k
Installing weak dependencies:
apr-util-openssl x86_64 1.6.3-1.amzn2023.0.1 amazonlinux 17 k
mod_http2 x86_64 2.0.27-1.amzn2023.0.3 amazonlinux 166 k
mod_lua x86_64 2.4.62-1.amzn2023 amazonlinux 61 k
Transaction Summary
Install 12 Packages
Total download size: 2.3 M
Installed size: 6.9 M
Downloading Packages:
(1/12): apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64.rpm 321 kB/s | 17 kB 00:00
(2/12): apr-1.7.5-1.amzn2023.0.4.x86_64.rpm 2.1 MB/s | 129 kB 00:00
(3/12): apr-util-1.6.3-1.amzn2023.0.1.x86_64.rpm 1.3 MB/s | 98 kB 00:00

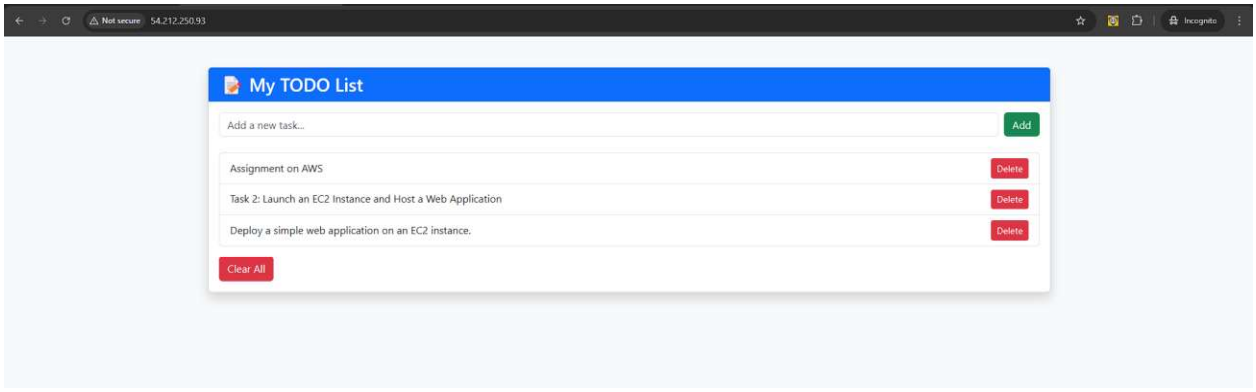

ec2-user@ip-172-31-20-173:~
[ec2-user@ip-172-31-20-173 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-20-173 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
Drop-In: /usr/lib/systemd/system/httpd.service.d
└─php-fpm.conf
Active: active (running) since Thu 2025-04-17 12:19:53 UTC; 43s ago
Docs: man:httpd.service(8)
Main PID: 32407 (httpd)
Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
Tasks: 177 (limit: 1111)
Memory: 13.0M
CPU: 64ms
CGroup: /system.slice/httpd.service
├─32407 /usr/sbin/httpd -DFOREGROUND
├─32408 /usr/sbin/httpd -DFOREGROUND
├─32409 /usr/sbin/httpd -DFOREGROUND
├─32410 /usr/sbin/httpd -DFOREGROUND
└─32411 /usr/sbin/httpd -DFOREGROUND
Apr 17 12:19:53 ip-172-31-20-173.us-west-2.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Apr 17 12:19:53 ip-172-31-20-173.us-west-2.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Apr 17 12:19:53 ip-172-31-20-173.us-west-2.compute.internal httpd[32407]: Server configured, listening on: port 80
[ec2-user@ip-172-31-20-173 ~]$ sudo vi /var/www/html/index.php


ec2-user@ip-172-31-20-173:~
$_SESSION['todos'] = [];
header("Location: /");
exit;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>My TODO List</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Bootstrap 5 CDN -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body class="bg-light">

<h3 class="mb-0">My TODO List</h3>

<form method="POST" class="d-flex mb-4">
<input type="text" name="task" class="form-control me-2" placeholder="Add a new task..." required>
<button type="submit" class="btn btn-success">Add</button>
</form>
<?php if (!empty($_SESSION['todos'])): ?>
<ul class="list-group mb-3">
<?php foreach ($_SESSION['todos'] as $index => $task): ?>
<li class="list-group-item d-flex justify-content-between align-items-center">
<?= $task ?>
<a href="?delete=<?= $index ?>" class="btn btn-sm btn-danger">Delete</a>
</li>
<?php endforeach; ?>
</ul>
<a href="?clear=1" class="btn btn-danger">Clear All</a>
<?php else: ?>
No tasks yet. Add something above! 👆
<?php endif; ?>

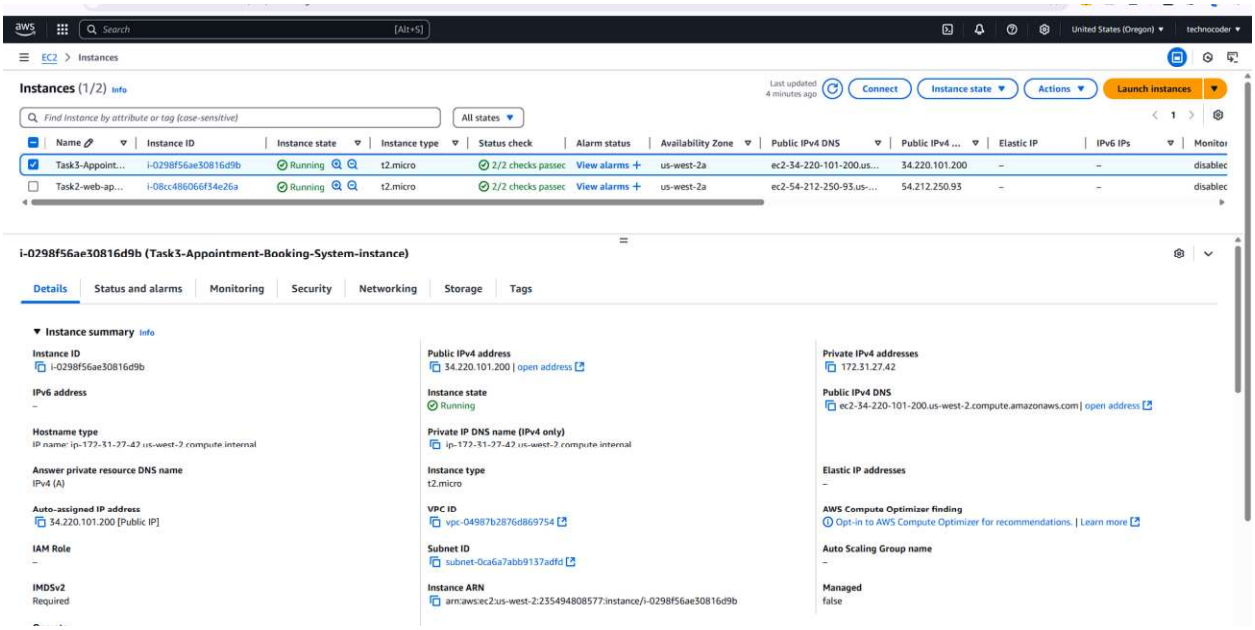
<!-- Bootstrap JS -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
:wq

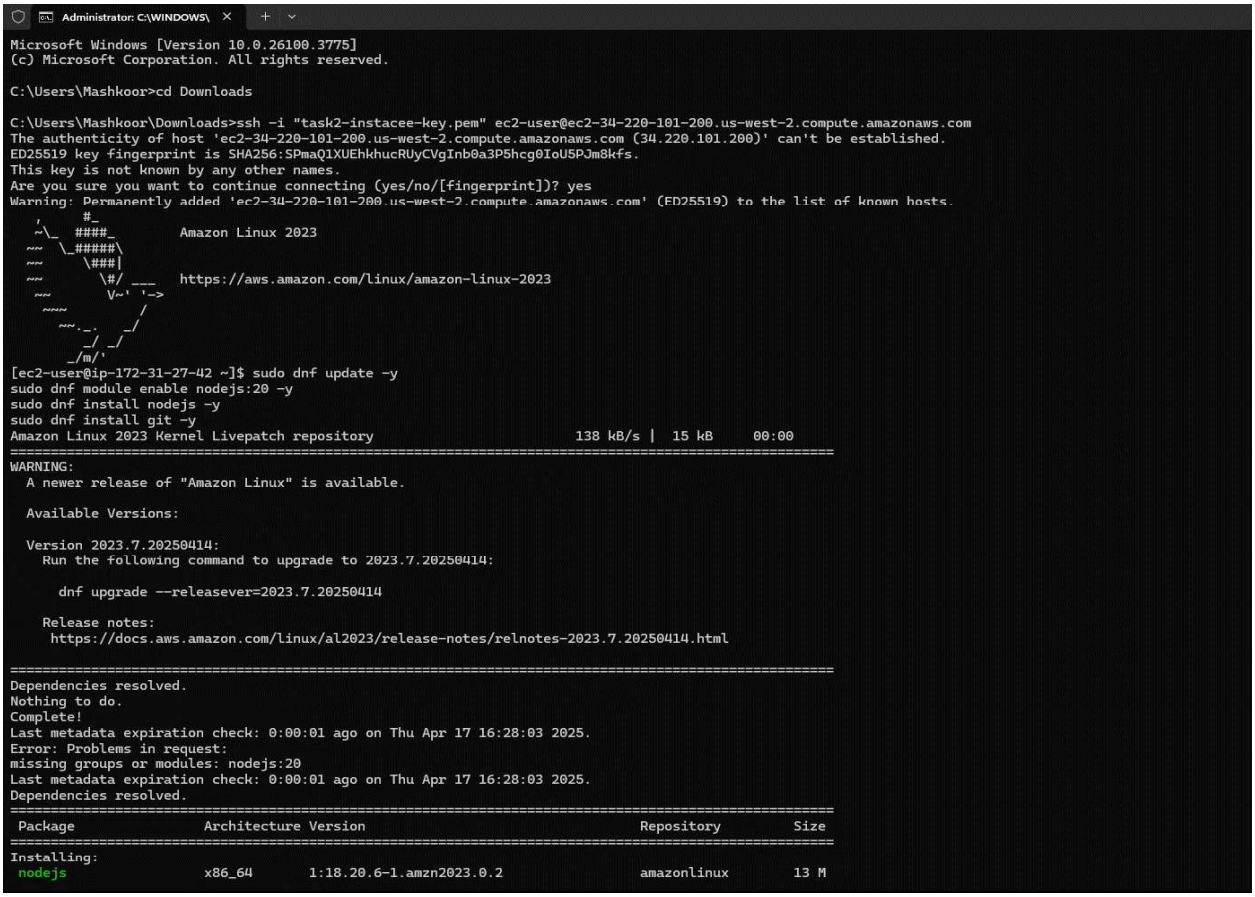


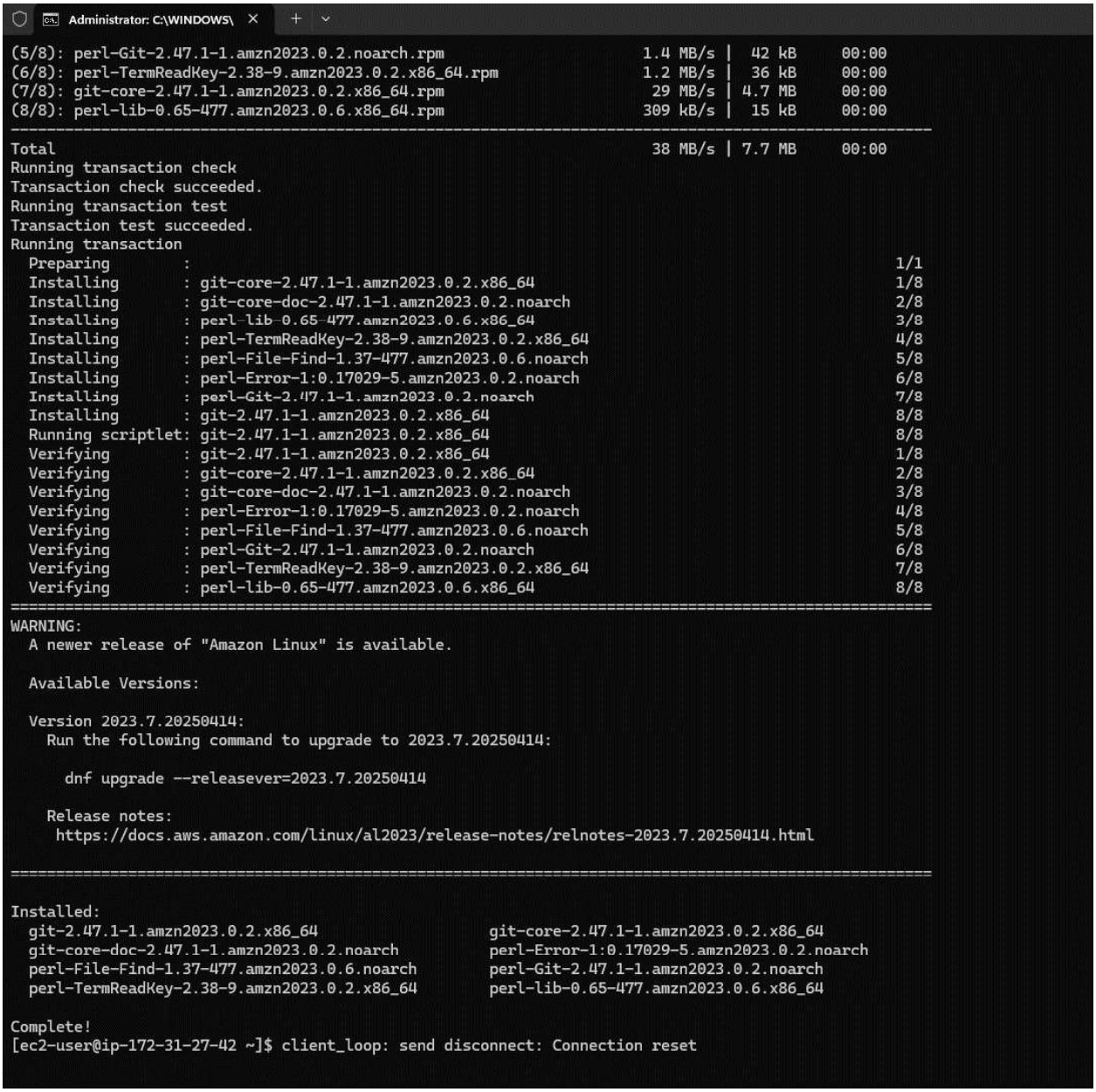

Link:

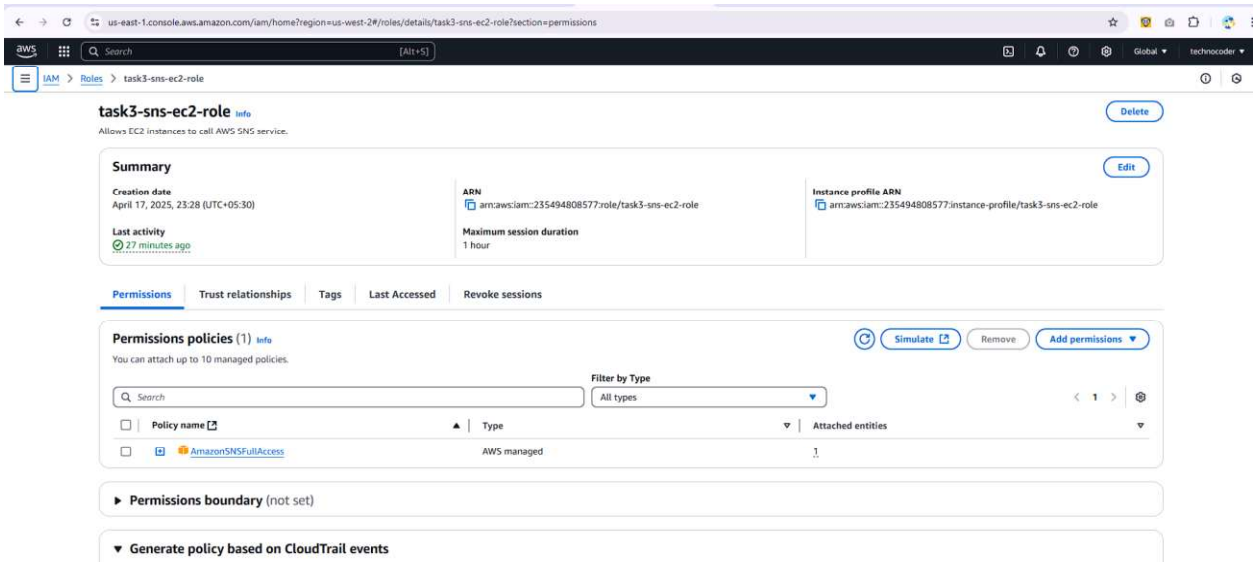
http://54.212.250.93/

# Task 3: Build an Appointment Booking System using AWS SES

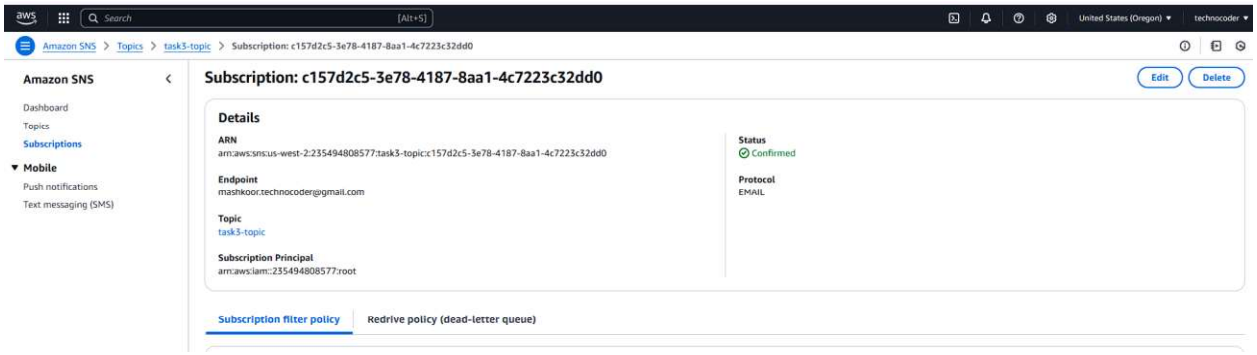


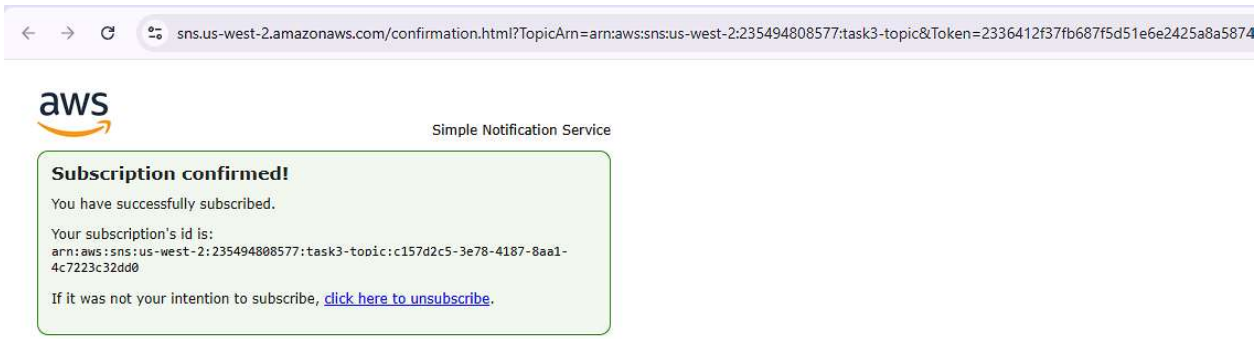




Administrator: C:\WINDOWS\
(5/8): perl-Git-2.47.1-1.amzn2023.0.2.noarch.rpm 1.4 MB/s | 42 kB 00:00
(6/8): perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64.rpm 1.2 MB/s | 36 kB 00:00
(7/8): git-core-2.47.1-1.amzn2023.0.2.x86_64.rpm 29 MB/s | 4.7 MB 00:00
(8/8): perl-lib-0.65-477.amzn2023.0.6.x86_64.rpm 309 kB/s | 15 kB 00:00
Total 38 MB/s | 7.7 MB 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : git-core-2.47.1-1.amzn2023.0.2.x86_64 1/8
Installing : git-core-doc-2.47.1-1.amzn2023.0.2.noarch 2/8
Installing : perl-lib-0.65-477.amzn2023.0.6.x86_64 3/8
Installing : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 4/8
Installing : perl-File-Find-1.37-477.amzn2023.0.6.noarch 5/8
Installing : perl-Error-1:0.17029-5.amzn2023.0.2.noarch 6/8
Installing : perl-Git-2.47.1-1.amzn2023.0.2.noarch 7/8
Installing : git-2.47.1-1.amzn2023.0.2.x86_64 8/8
Running scriptlet: git-2.47.1-1.amzn2023.0.2.x86_64 8/8
Verifying : git-2.47.1-1.amzn2023.0.2.x86_64 1/8
Verifying : git-core-2.47.1-1.amzn2023.0.2.x86_64 2/8
Verifying : git-core-doc-2.47.1-1.amzn2023.0.2.noarch 3/8
Verifying : perl-Error-1:0.17029-5.amzn2023.0.2.noarch 4/8
Verifying : perl-File-Find-1.37-477.amzn2023.0.6.noarch 5/8
Verifying : perl-Git-2.47.1-1.amzn2023.0.2.noarch 6/8
Verifying : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 7/8
Verifying : perl-lib-0.65-477.amzn2023.0.6.x86_64 8/8
WARNING:
A newer release of "Amazon Linux" is available.
Available Versions:
Version 2023.7.20250414:
Run the following command to upgrade to 2023.7.20250414:
dnf upgrade --releasever=2023.7.20250414
Release notes:
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.7.20250414.html
Installed:
git-2.47.1-1.amzn2023.0.2.x86_64 git-core-2.47.1-1.amzn2023.0.2.x86_64
git-core-doc-2.47.1-1.amzn2023.0.2.noarch perl-Error-1:0.17029-5.amzn2023.0.2.noarch
perl-File-Find-1.37-477.amzn2023.0.6.noarch perl-Git-2.47.1-1.amzn2023.0.2.noarch
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 perl-lib-0.65-477.amzn2023.0.6.x86_64
Complete!
[ec2-user@ip-172-31-27-42 ~]$ client_loop: send disconnect: Connection reset


us-east-1.console.aws.amazon.com/iam/home?region=us-west-2#/roles/details/task3-sns-ec2-role?section=permissions
IAM > Roles > task3-sns-ec2-role
task3-sns-ec2-role Info
Allows EC2 instances to call AWS SNS service.
Delete
Summary
Edit
Creation date
April 17, 2025, 23:28 (UTC+05:30)
ARN
arn:aws:iam::235494808577:role/task3-sns-ec2-role
Instance profile ARN
arn:aws:iam::235494808577:instance-profile/task3-sns-ec2-role
Last activity
27 minutes ago
Maximum session duration
1 hour
Permissions
Trust relationships
Tags
Last Accessed
Revoke sessions
Permissions policies (1) Info
You can attach up to 10 managed policies.
Simulate
Remove
Add permissions
Filter by Type
All types
Policy name
Type
Attached entities
AmazonSNSFullAccess
AWS managed
1
Permissions boundary (not set)
Generate policy based on CloudTrail events


us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#ModifyIAMRole:instanceId=i-0298f56ae30816d9b
EC2 > Instances > i-0298f56ae30816d9b > Modify IAM role
Modify IAM role Info
Attach an IAM role to your instance.
Instance ID
i-0298f56ae30816d9b (Task3-Appointment-Booking-System-instance)
IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.
task3-sns-ec2-role
Create new IAM role
Cancel
Update IAM role


Amazon SNS > Topics > task3-topic > Subscription: c157d2c5-3e78-4187-8aa1-4c7223c32dd0
Amazon SNS
Dashboard
Topics
Subscriptions
Mobile
Push notifications
Text messaging (SMS)
Subscription: c157d2c5-3e78-4187-8aa1-4c7223c32dd0
Edit
Delete
Details
ARN
arn:aws:sns:us-west-2:235494808577:task3-topic:c157d2c5-3e78-4187-8aa1-4c7223c32dd0
Endpoint
mashkoor.technocoder@gmail.com
Topic
task3-topic
Subscription Principal
arn:aws:iam::235494808577:root
Status
Confirmed
Protocol
EMAIL
Subscription filter policy
Redrive policy (dead-letter queue)


sns.us-west-2.amazonaws.com/confirmation.html?TopicArn=arn:aws:sns:us-west-2:235494808577:task3-topic&Token=2336412f37fb687f5d51e6e2425a8a5874
Simple Notification Service
Subscription confirmed!
You have successfully subscribed.
Your subscription's id is:
arn:aws:sns:us-west-2:235494808577:task3-topic:c157d2c5-3e78-4187-8aa1-4c7223c32dd0
If it was not your intention to subscribe, click here to unsubscribe.


```
[ec2-user@ip-172-31-27-42 ~]$ cd booking-app/
[ec2-user@ip-172-31-27-42 booking-app]$ cat server.js
const express = require('express');
const bodyParser = require('body-parser');
const sqlite3 = require('sqlite3').verbose();
const AWS = require('aws-sdk');
require('dotenv').config();

const app = express();
const port = 80;

// Middleware
app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.static('public'));

// SQLite DB setup
const db = new sqlite3.Database('appointments.db');
db.run("CREATE TABLE IF NOT EXISTS bookings (name TEXT, email TEXT, service TEXT, datetime TEXT)");

// AWS SNS setup
AWS.config.update({ region: process.env.AWS_REGION });
const sns = new AWS.SNS();

app.post('/book', (req, res) => {
  const { name, email, service, datetime } = req.body;

  // Save booking data to SQLite
  db.run("INSERT INTO bookings VALUES (?, ?, ?, ?)", [name, email, service, datetime], (err) => {
    if (err) {
      console.error("DB error:", err);
      return res.send("Failed to save booking.");
    }

    // Send SNS email
    const message = `Appointment Confirmation:
Name: ${name}
Email: ${email}
Service: ${service}
Date/Time: ${datetime}`;

    const params = {
      Message: message,
      TopicArn: process.env.SNS_TOPIC_ARN
    };

    sns.publish(params, (err, data) => {
      if (err) {
        console.error("SNS error:", err);
        res.send("Saved, but email failed.");
      } else {
        console.log("SNS sent:", data);
        res.send("Booking successful! Check your email.");
      }
    });
  });
});

app.listen(port, () => {
  console.log(`Server running at http://localhost:${port}`);
});

[ec2-user@ip-172-31-27-42 booking-app]$
```

```
[ec2-user@ip-172-31-27-42 booking-app]$ cat .env
SNS_TOPIC_ARN=arn:aws:sns:us-west-2:235494808577:task3-topic
AWS_REGION=us-west-2

[ec2-user@ip-172-31-27-42 booking-app]$ ls
appointments.db  node_modules  package-lock.json  package.json  public  server.js
[ec2-user@ip-172-31-27-42 booking-app]$
```

```
[ec2-user@ip-172-31-27-42 booking-app]$ cat public/index.html
<!DOCTYPE html>
<html>
  <head>
    <title>Book Appointment</title>
  </head>
  <body>
    <h2>Book Appointment</h2>
    <form action="/book" method="POST">
      <input type="text" name="name" placeholder="Name" required><br>
      <input type="email" name="email" placeholder="Email" required><br>
      <input type="text" name="service" placeholder="Service" required><br>
      <input type="datetime-local" name="datetime" required><br>
      <button type="submit">Book Now</button>
    </form>
  </body>
</html>

[ec2-user@ip-172-31-27-42 booking-app]$
```

```
[ec2-user@ip-172-31-27-42 booking-app]$ sqlite3 appointments.db
SQLite version 3.40.0 2023-06-02 12:56:32
Enter ".help" for usage hints.
sqlite> SELECT * FROM bookings;
rocky|admin@gmail.com|hey testing|2025-04-17T23:01
rocky|kaifpatel200@gmail.com|aaaaaa|2025-04-18T23:29
rocky|mashkoor.technocoder@gmail.com|aaaaaa|2025-04-18T23:29
rocky|mashkoor.technocoder@gmail.com|aaaaaa|2025-04-18T23:29
rocky|kaifpatel200@gmail.com|success bro|2025-04-17T23:35
aa|test@gmail.com|aaaaaaaaaaaa|2025-05-10T00:05
sqlite>
```

```
[ec2-user@ip-172-31-27-42 booking-app]$ sudo node server.js
Server running at http://localhost:80
(node:33625) NOTE: The AWS SDK for JavaScript (v2) is in maintenance mode.
 SDK releases are limited to address critical bug fixes and security issues only.

Please migrate your code to use AWS SDK for JavaScript (v3).
For more information, check the blog post at https://a.co/cUPnyil
(Use `node --trace-warnings ...` to show where the warning was created)
SNS sent: {
  ResponseMetadata: { RequestId: 'cc2a16c9-ad31-50f5-94bf-76b2f9cb42a8' },
  MessageId: '4d6524e2-5620-579a-adb7-449902dac5c7'
}
SNS sent: {
  ResponseMetadata: { RequestId: '4e36b4da-841e-52e3-a9ed-d4a1d17e0ae0' },
  MessageId: '171f687c-da1f-5f74-9cd3-08078f7f5777'
}
```

```
[ec2-user@ip-172-31-27-42 ~]$ sudo yum install nginx -y
Last metadata expiration check: 12:11:21 ago on Thu Apr 17 16:28:03 2025.
Package nginx-1:1.26.3-1.amzn2023.0.1.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-27-42 ~]$ sudo vi /etc/nginx/nginx.conf
[ec2-user@ip-172-31-27-42 ~]$ cat  /etc/nginx/nginx.conf
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    keepalive_timeout   65;
    types_hash_max_size 4096;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    include /etc/nginx/conf.d/*.conf;
}

[ec2-user@ip-172-31-27-42 ~]$
```

```
[ec2-user@ip-172-31-27-42 ~]$ cat  /etc/nginx/nginx.conf
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    keepalive_timeout   65;
    types_hash_max_size 4096;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    include /etc/nginx/conf.d/*.conf;
}

[ec2-user@ip-172-31-27-42 ~]$ cat /etc/nginx/conf.d/booking-app.conf
server {
    listen 80;
    server_name _;

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}

[ec2-user@ip-172-31-27-42 ~]$
```

```
[ec2-user@ip-172-31-27-42 ~]$ sudo systemctl daemon-reload
sudo systemctl start booking-app
sudo systemctl enable booking-app
[ec2-user@ip-172-31-27-42 ~]$ sudo systemctl status booking-app
● booking-app.service - Booking App - Node.js
     Loaded: loaded (/etc/systemd/system/booking-app.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-04-18 04:07:02 UTC; 37min ago
   Main PID: 9684 (node)
      Tasks: 15 (limit: 1111)
     Memory: 26.2M
        CPU: 1.021s
     CGroup: /system.slice/booking-app.service
             └─9684 /usr/bin/node /home/ec2-user/booking-app/server.js

Apr 18 04:14:17 ip-172-31-27-42.us-west-2.compute.internal systemd[1]: /etc/systemd/system/booking-app.service:11: Standard output type syslog is obsolete, automatically updating to journal. Please update you>
Apr 18 04:14:17 ip-172-31-27-42.us-west-2.compute.internal systemd[1]: /etc/systemd/system/booking-app.service:12: Standard output type syslog is obsolete, automatically updating to journal. Please update you>
Apr 18 04:14:44 ip-172-31-27-42.us-west-2.compute.internal node[9684]: SNS sent: {
Apr 18 04:14:44 ip-172-31-27-42.us-west-2.compute.internal node[9684]:   ResponseMetadata: { RequestId: 'b3bfbf93-1eed-5806-9d36-dfe8cd1935d1' },
Apr 18 04:14:44 ip-172-31-27-42.us-west-2.compute.internal node[9684]:   MessageId: 'c7e4db02-891d-59ec-a439-bb795ae9f43b'
Apr 18 04:14:44 ip-172-31-27-42.us-west-2.compute.internal node[9684]: }
Apr 18 04:44:38 ip-172-31-27-42.us-west-2.compute.internal systemd[1]: /etc/systemd/system/booking-app.service:11: Standard output type syslog is obsolete, automatically updating to journal. Please update you>
Apr 18 04:44:38 ip-172-31-27-42.us-west-2.compute.internal systemd[1]: /etc/systemd/system/booking-app.service:12: Standard output type syslog is obsolete, automatically updating to journal. Please update you>
Apr 18 04:44:38 ip-172-31-27-42.us-west-2.compute.internal systemd[1]: /etc/systemd/system/booking-app.service:11: Standard output type syslog is obsolete, automatically updating to journal. Please update you>
Apr 18 04:44:38 ip-172-31-27-42.us-west-2.compute.internal systemd[1]: /etc/systemd/system/booking-app.service:12: Standard output type syslog is obsolete, automatically updating to journal. Please update you>
lines 1-20/20 (END)
[ec2-user@ip-172-31-27-42 ~]$
[ec2-user@ip-172-31-27-42 ~]$
[ec2-user@ip-172-31-27-42 ~]$
```

got to -> http://34.220.101.200/