Building a Hybrid Web App on AWS (EC2 + CloudFront + S3)

For this project, my goal was to design and deploy a simple hybrid web architecture in which an EC2 instance serves HTML, while S3 and CloudFront deliver static assets securely and efficiently. This hands-on build reinforced core AWS architectural principles in networking, security, and performance optimization.

Project Goal

I set out to host a lightweight web application using three key AWS services:

- **EC2** → Serves the main HTML page
- **S3** → Stores static content (CSS + images)
- **CloudFront** → Distributes the static assets globally with low latency

This architecture allowed me to separate dynamic content from static content, reduce computing workload on EC2, and enforce more secure access patterns.

Architecture Overview

Here's how the components interacted:

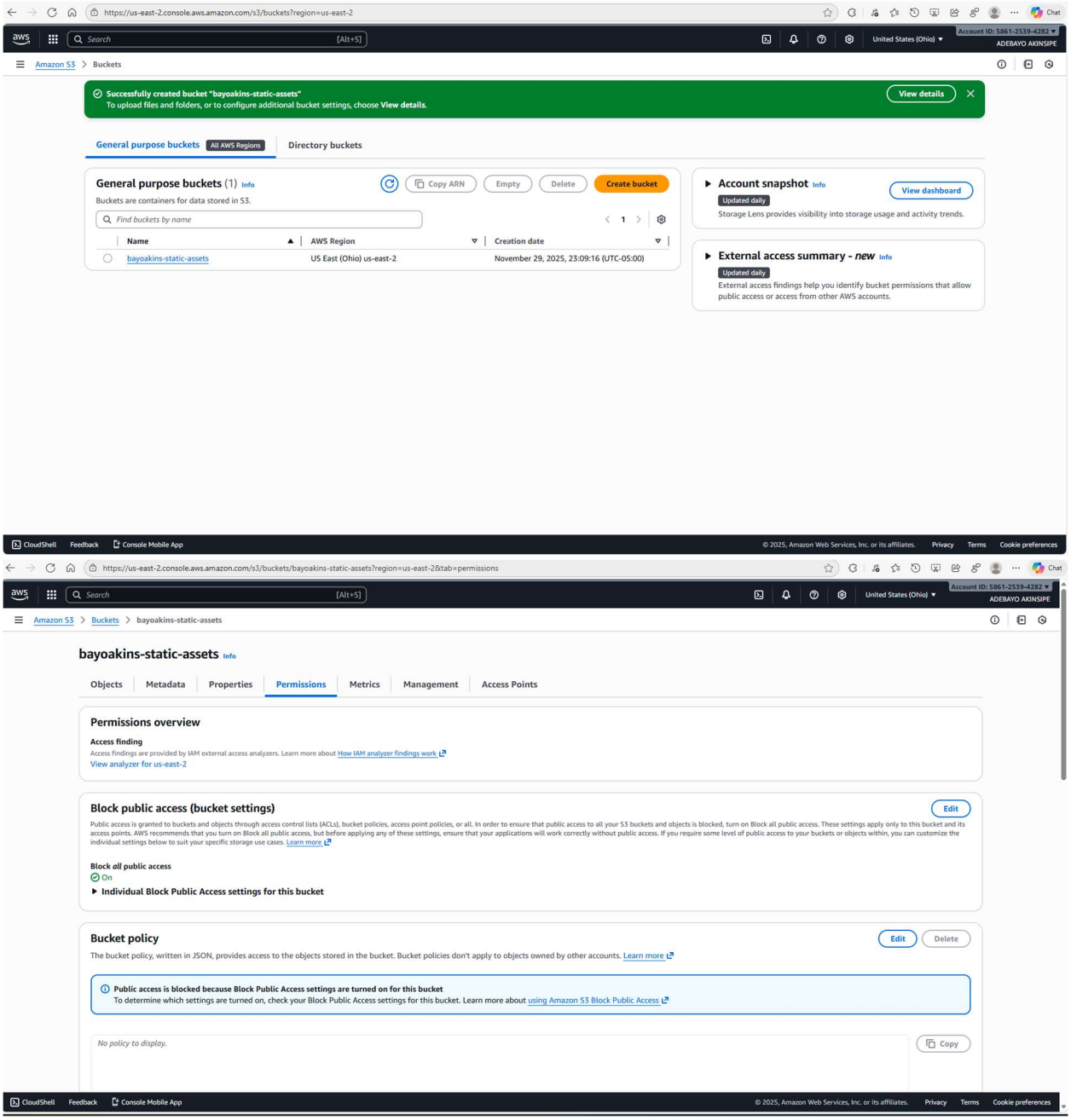
1. Users access the public web page hosted on **EC2**
2. The HTML references the **CloudFront domain**
3. CloudFront retrieves the assets from **S3**
4. S3 remains private — assets aren't publicly accessible except through CloudFront

By forcing access through CloudFront, I improved performance and security posture.

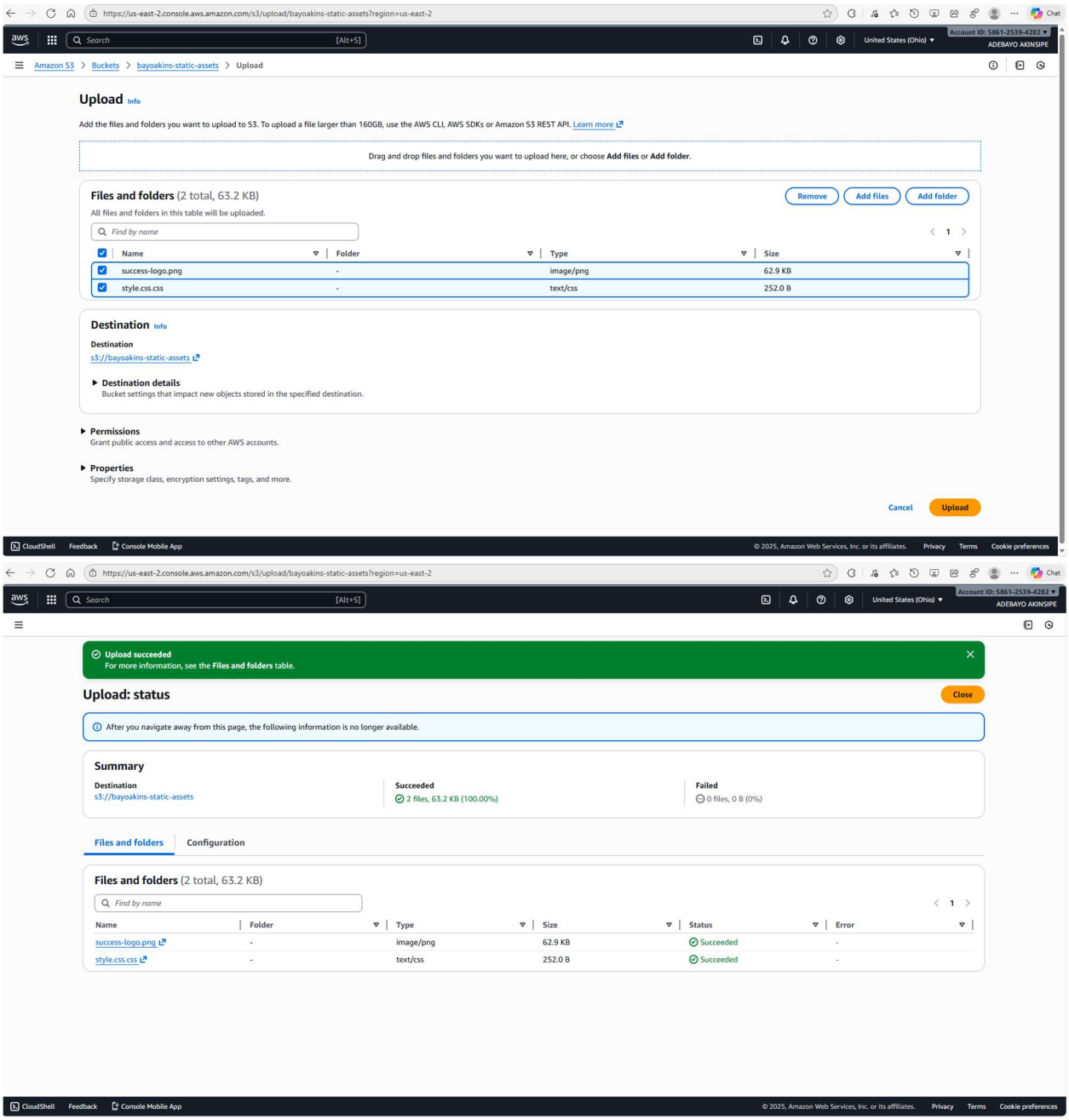
What I Built Step-by-Step

Creating the S3 Bucket

I created a private S3 bucket to hold style.css and a logo image. I intentionally kept public access blocked to verify that CloudFront would be the only service able to retrieve these objects.

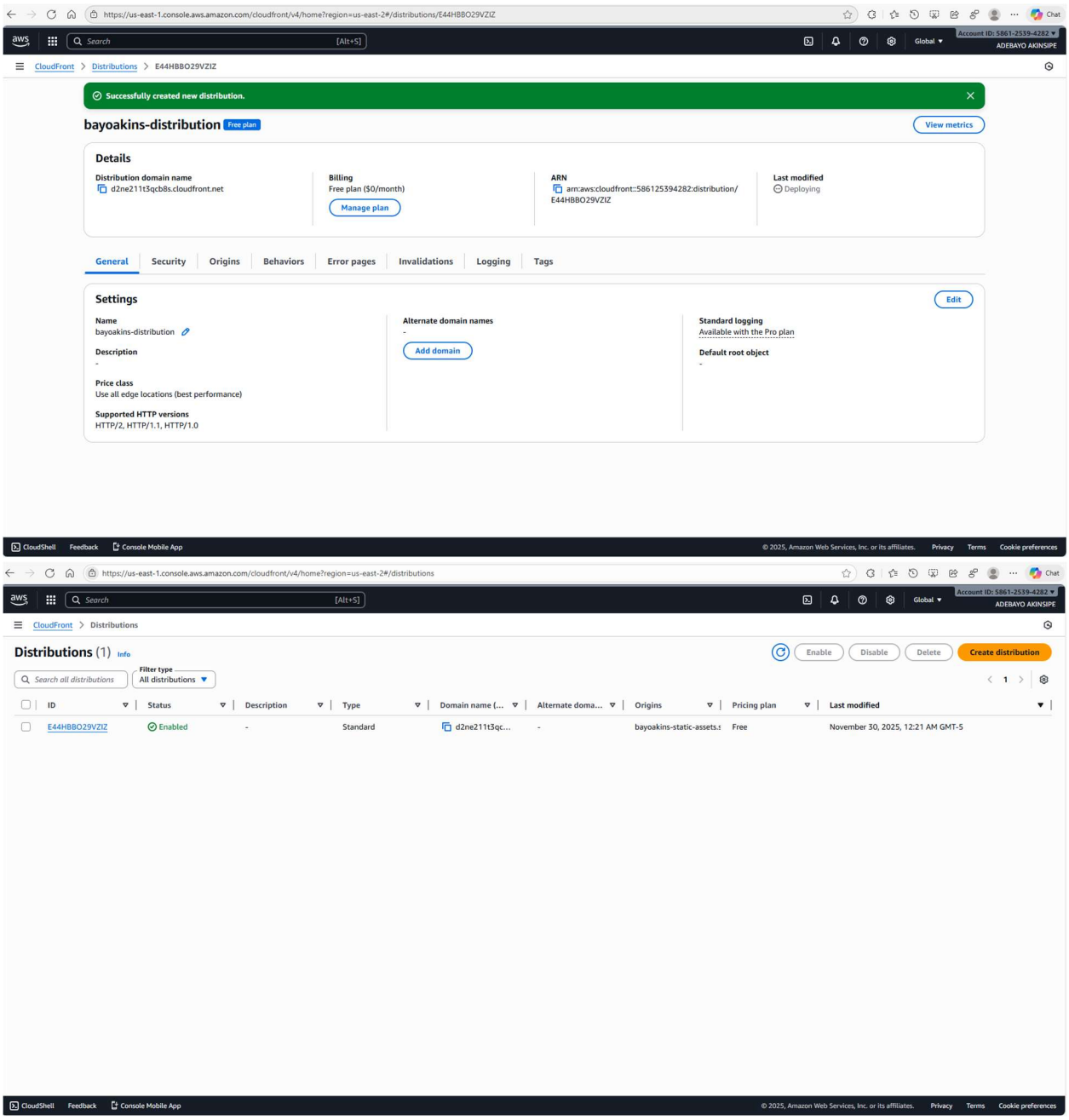

https://us-east-2.console.aws.amazon.com/s3/buckets?region=us-east-2
Search
[Alt+S]
United States (Ohio)
ADEBAYO AKINSIPE
Amazon S3 > Buckets
Successfully created bucket "bayoakins-static-assets"
To upload files and folders, or to configure additional bucket settings, choose View details.
View details
General purpose buckets
All AWS Regions
Directory buckets
General purpose buckets (1) Info
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
Creation date
bayoakins-static-assets
US East (Ohio) us-east-2
November 29, 2025, 23:09:16 (UTC-05:00)
Account snapshot Info
Updated daily
View dashboard
Storage Lens provides visibility into storage usage and activity trends.
External access summary - new Info
Updated daily
External access findings help you identify bucket permissions that allow public access or access from other AWS accounts.
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

https://us-east-2.console.aws.amazon.com/s3/buckets/bayoakins-static-assets?region=us-east-2&tab=permissions
Amazon S3 > Buckets > bayoakins-static-assets
bayoakins-static-assets Info
Objects
Metadata
Properties
Permissions
Metrics
Management
Access Points
Permissions overview
Access finding
Access findings are provided by IAM external access analyzers. Learn more about How IAM analyzer findings work
View analyzer for us-east-2
Block public access (bucket settings)
Edit
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
On
Individual Block Public Access settings for this bucket
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
Public access is blocked because Block Public Access settings are turned on for this bucket
To determine which settings are turned on, check your Block Public Access settings for this bucket. Learn more about using Amazon S3 Block Public Access
No policy to display.
Copy
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



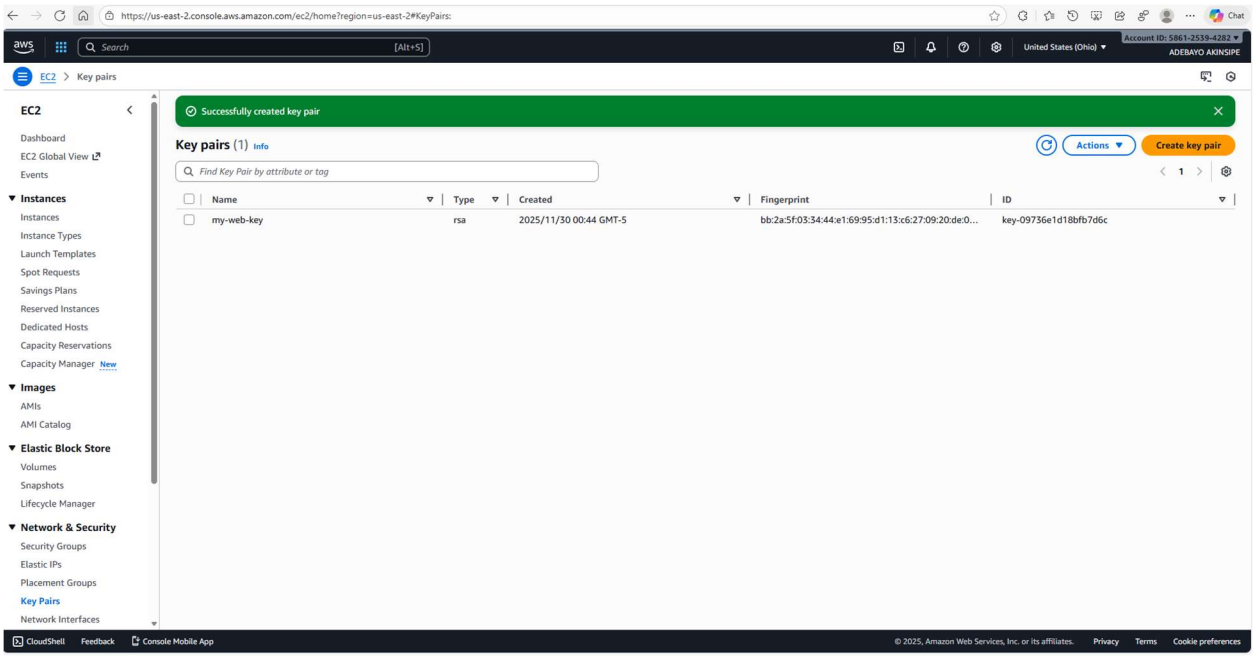
Setting Up CloudFront

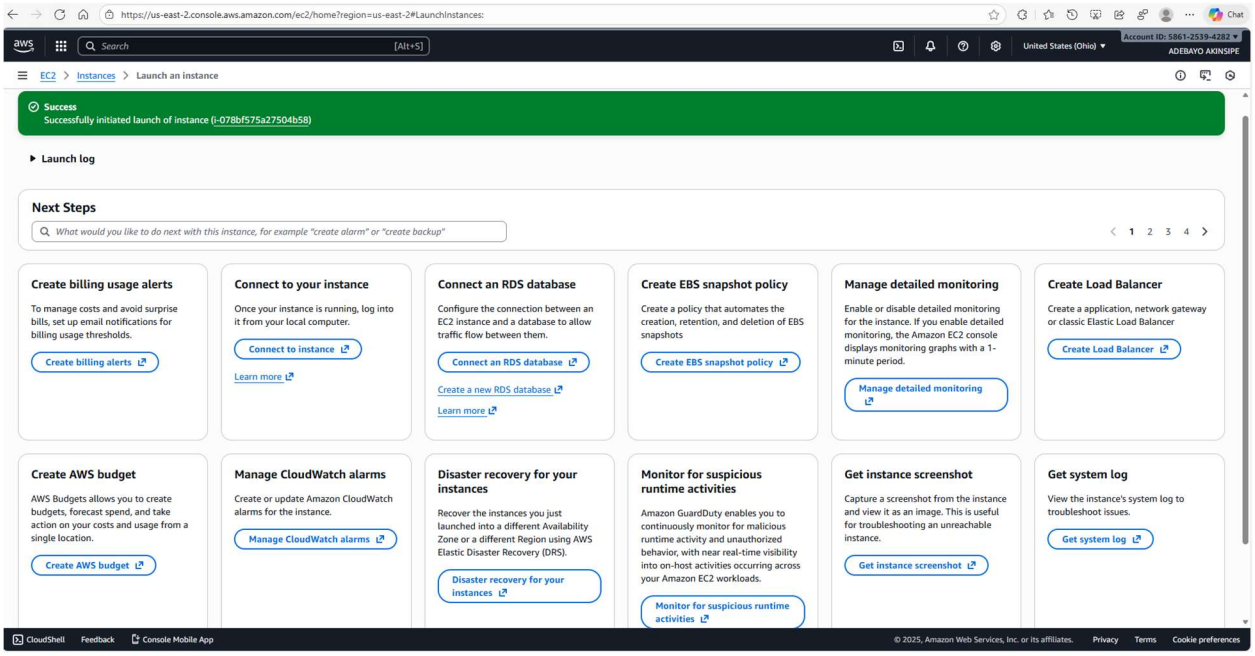
Next, I configured a CloudFront Distribution that pointed to the S3 bucket. Once active, I copied the CloudFront domain name — this became the reference inside my HTML and CSS calls.

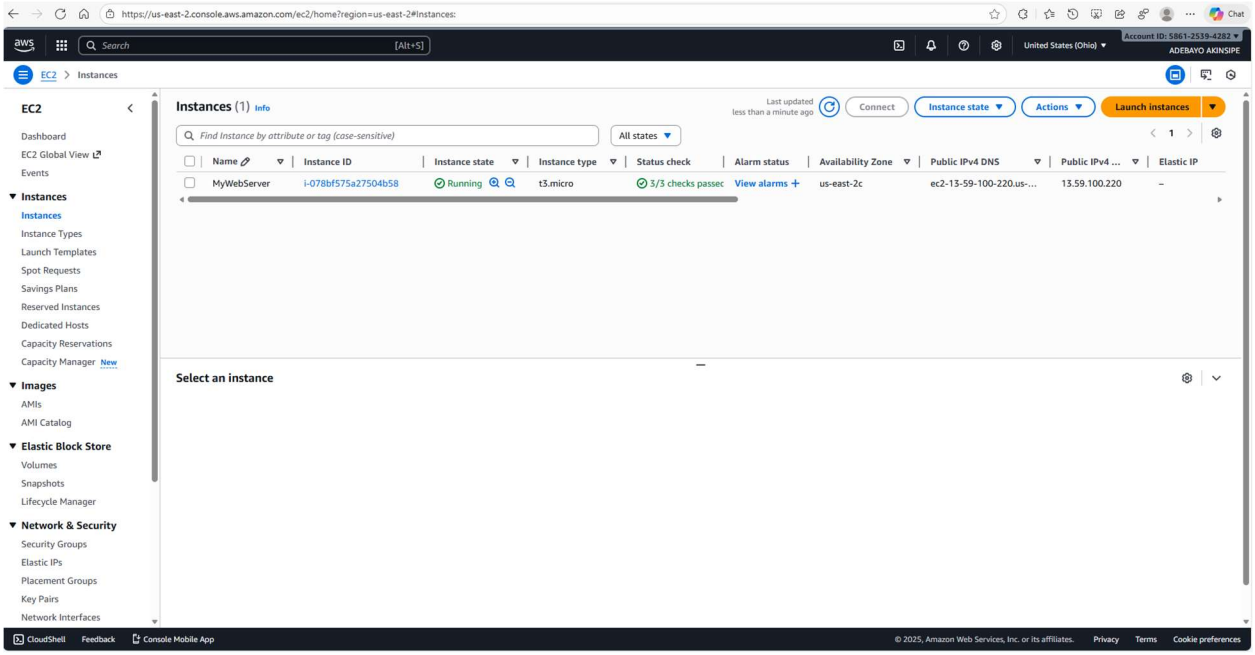


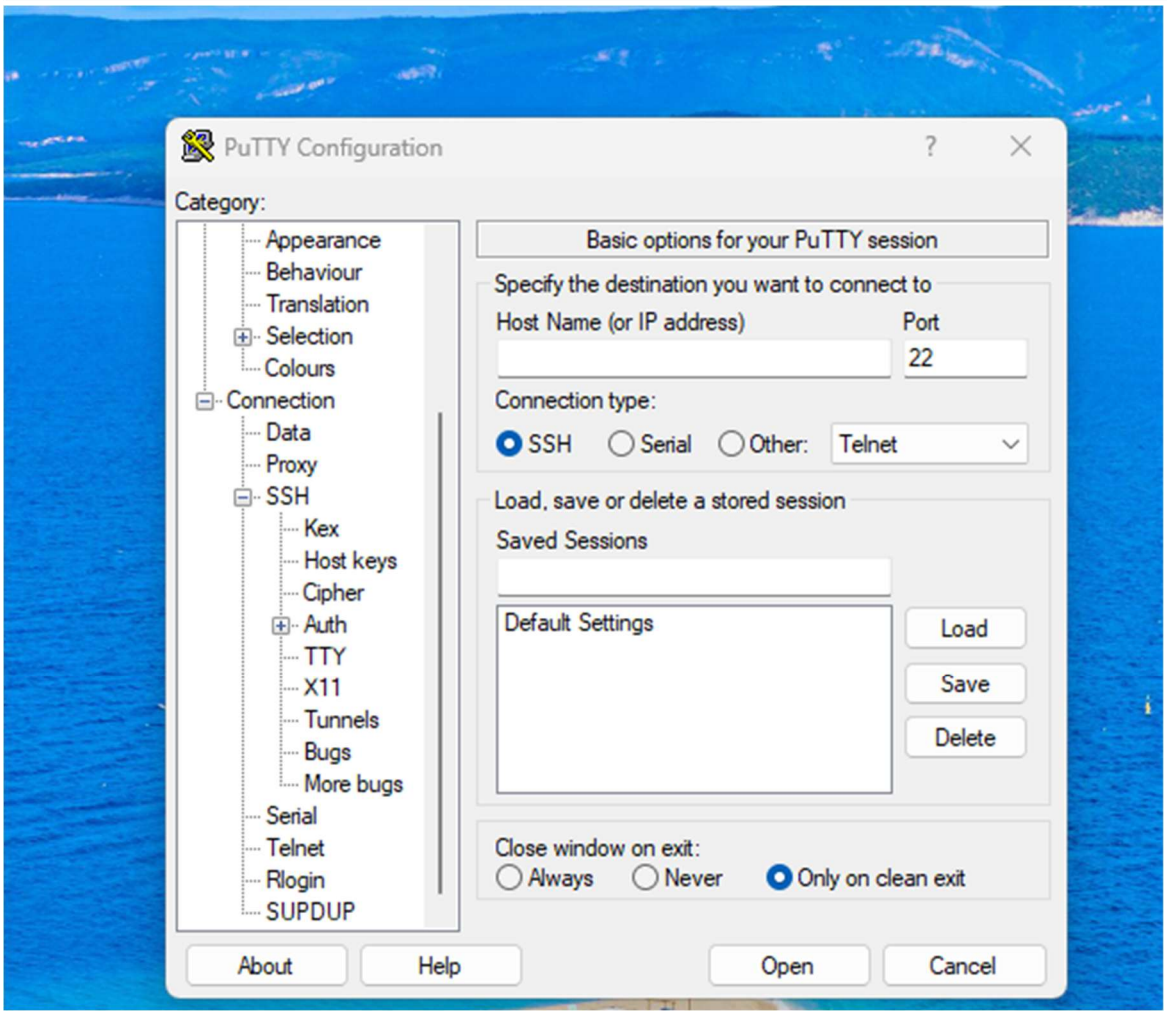
Launching the EC2 Instance, *SSH into EC2, and Installing Apache*

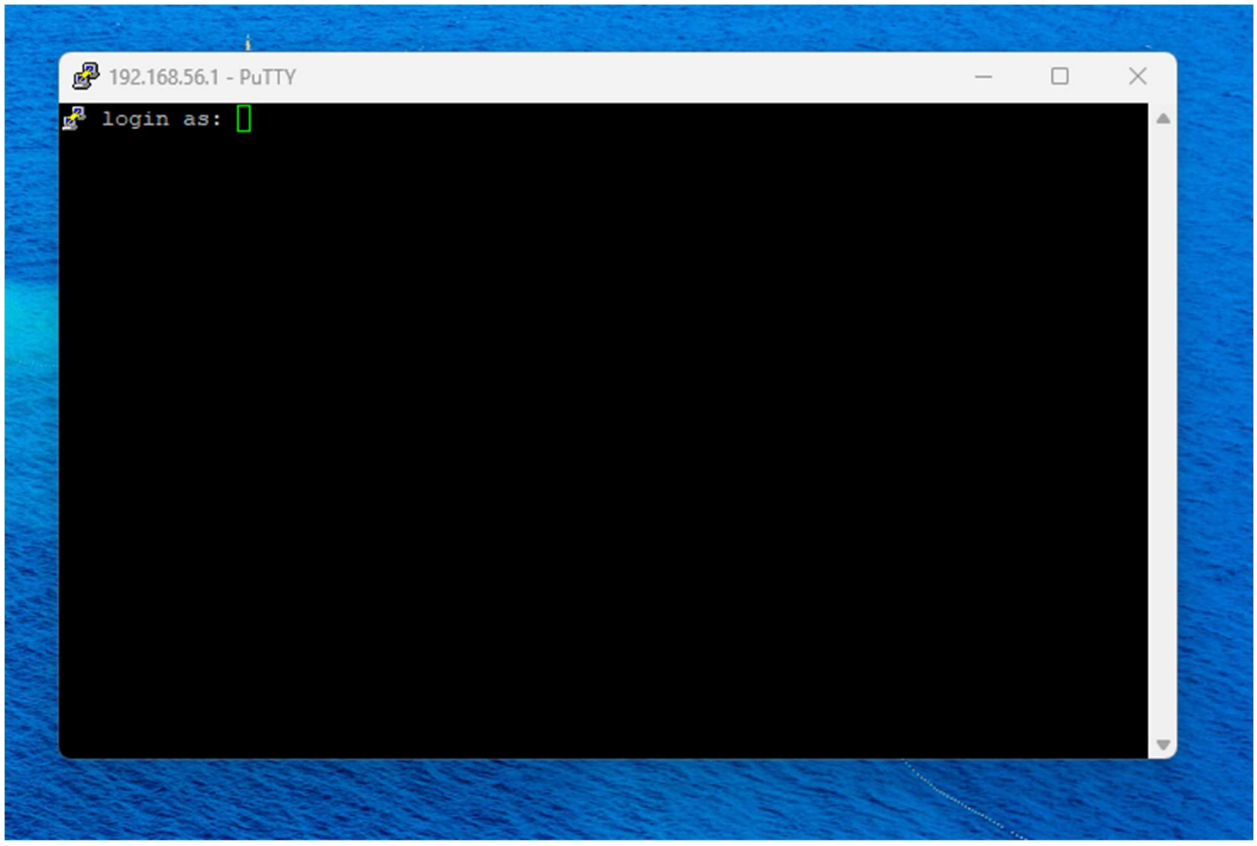
I deployed an Amazon Linux 2023 EC2 instance using a t2.micro instance type and enabled inbound HTTP traffic while restricting SSH access to my IP address. After connecting to the instance via PuTTY, I installed Apache and prepared it to serve the HTML file.

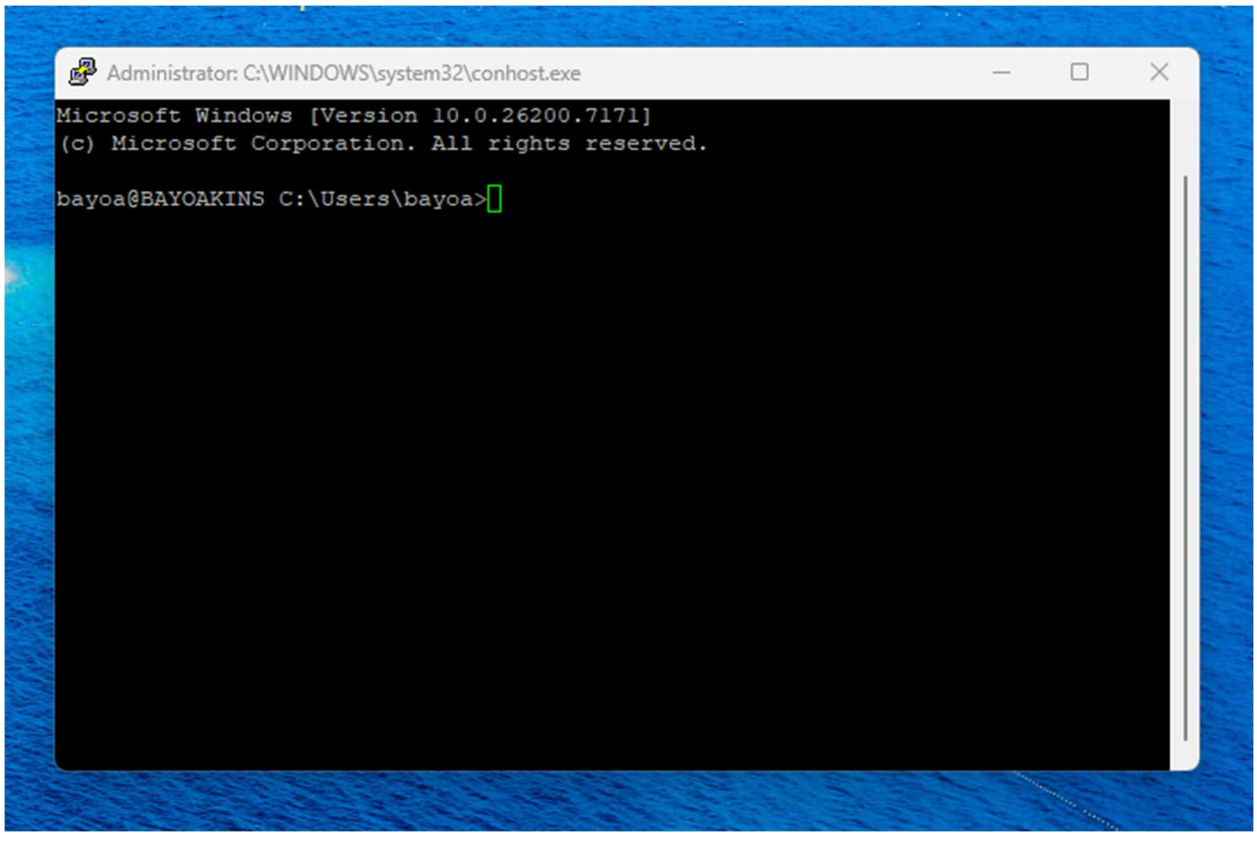

https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#KeyPairs:
Search [Alt+S]
United States (Ohio)
ADEBAYO AKINSIPE
EC2 > Key pairs
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Successfully created key pair
Key pairs (1) Info
Actions
Create key pair
Find Key Pair by attribute or tag
Name
Type
Created
Fingerprint
ID
my-web-key
rsa
2025/11/30 00:44 GMT-5
bb:2a:5f:03:34:44:e1:69:95:d1:13:c6:27:09:20:de:0...
key-09736e1d18bfb7d6c
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

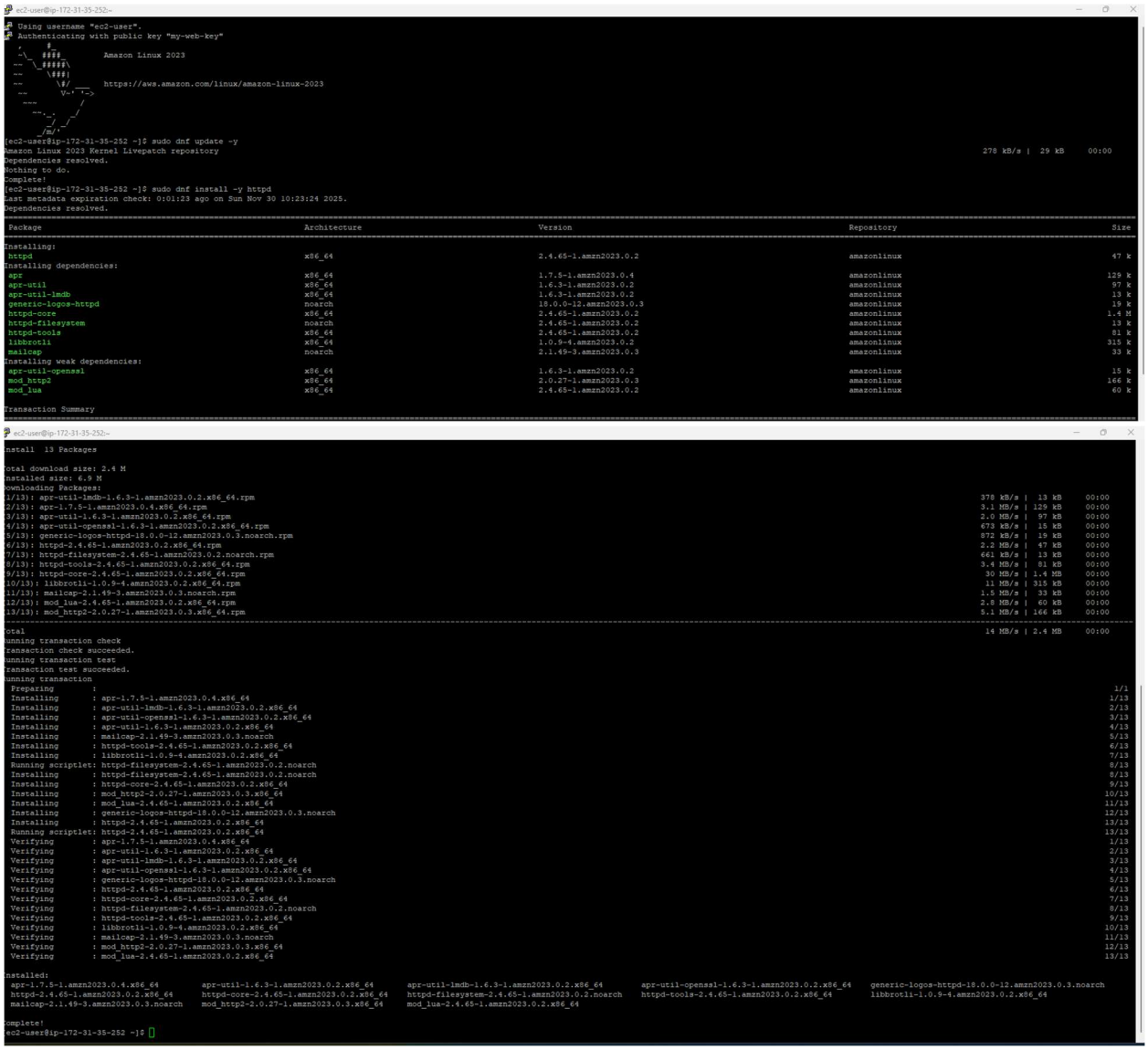

https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstances:
Search [Alt+S]
United States (Ohio)
ADEBAYO AKINSIPE
EC2 > Instances > Launch an instance
Success
Successfully initiated launch of instance (i-078bf575a27504b58)
Launch log
Next Steps
What would you like to do next with this instance, for example "create alarm" or "create backup"
Create billing usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing usage thresholds.
Create billing alerts
Connect to your instance
Once your instance is running, log into it from your local computer.
Connect to instance
Learn more
Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
Connect an RDS database
Create a new RDS database
Learn more
Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots
Create EBS snapshot policy
Manage detailed monitoring
Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period.
Manage detailed monitoring
Create Load Balancer
Create a application, network gateway or classic Elastic Load Balancer
Create Load Balancer
Create AWS budget
AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location.
Create AWS budget
Manage CloudWatch alarms
Create or update Amazon CloudWatch alarms for the instance.
Manage CloudWatch alarms
Disaster recovery for your instances
Recover the instances you just launched into a different Availability Zone or a different Region using AWS Elastic Disaster Recovery (DRS).
Disaster recovery for your instances
Monitor for suspicious runtime activities
Amazon GuardDuty enables you to continuously monitor for malicious runtime activity and unauthorized behavior, with near real-time visibility into on-host activities occurring across your Amazon EC2 workloads.
Monitor for suspicious runtime activities
Get instance screenshot
Capture a screenshot from the instance and view it as an image. This is useful for troubleshooting an unreachable instance.
Get instance screenshot
Get system log
View the instance's system log to troubleshoot issues.
Get system log
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Instances:
Search
[Alt+S]
United States (Ohio)
ADEBAYO AKINSIPE
EC2 > Instances
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Instances (1) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
MyWebServer
i-078bf575a27504b58
Running
t3.micro
3/3 checks passed
View alarms
us-east-2c
ec2-13-59-100-220.us-...
13.59.100.220
Select an instance
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


PuTTY Configuration
Category:
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
SSH
Kex
Host keys
Cipher
Auth
TTY
X11
Tunnels
Bugs
More bugs
Serial
Telnet
Rlogin
SUPDUP
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
22
Connection type:
SSH
Serial
Other:
Telnet
Load, save or delete a stored session
Saved Sessions
Default Settings
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Help
Open
Cancel


192.168.56.1 - PuTTY
login as:

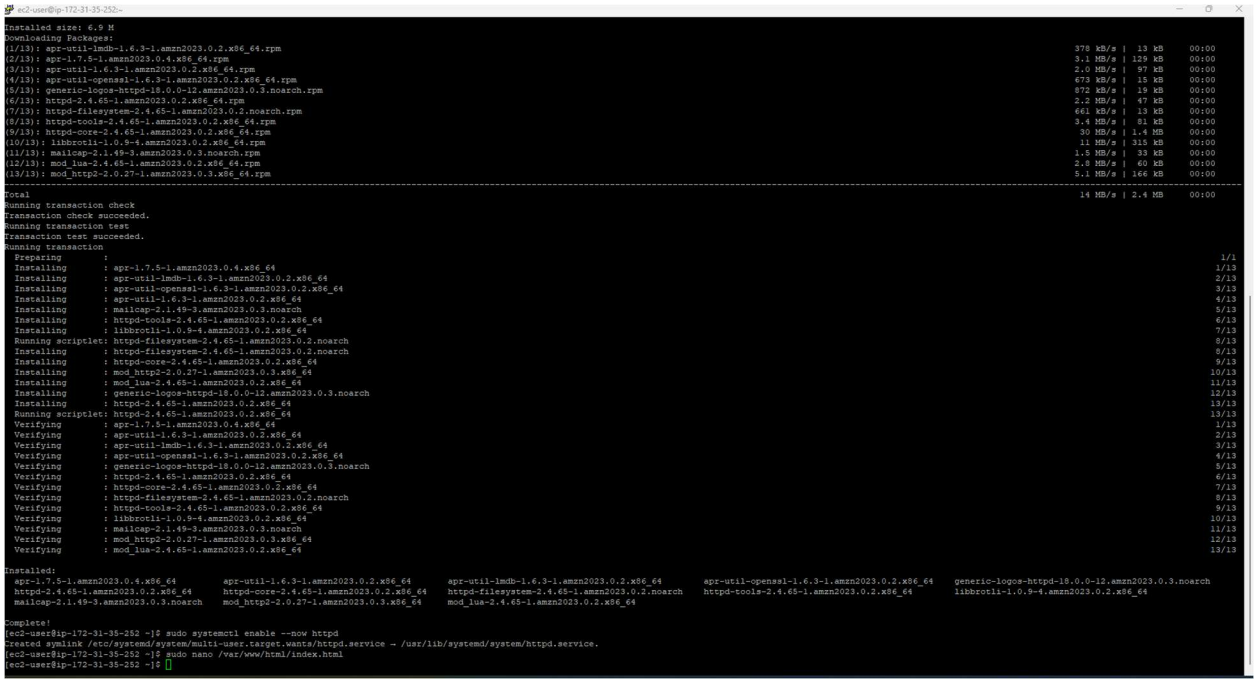

Administrator: C:\WINDOWS\system32\conhost.exe
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.
bayoa@BAYOAKINS C:\Users\bayoa>



Linking Everything in HTML

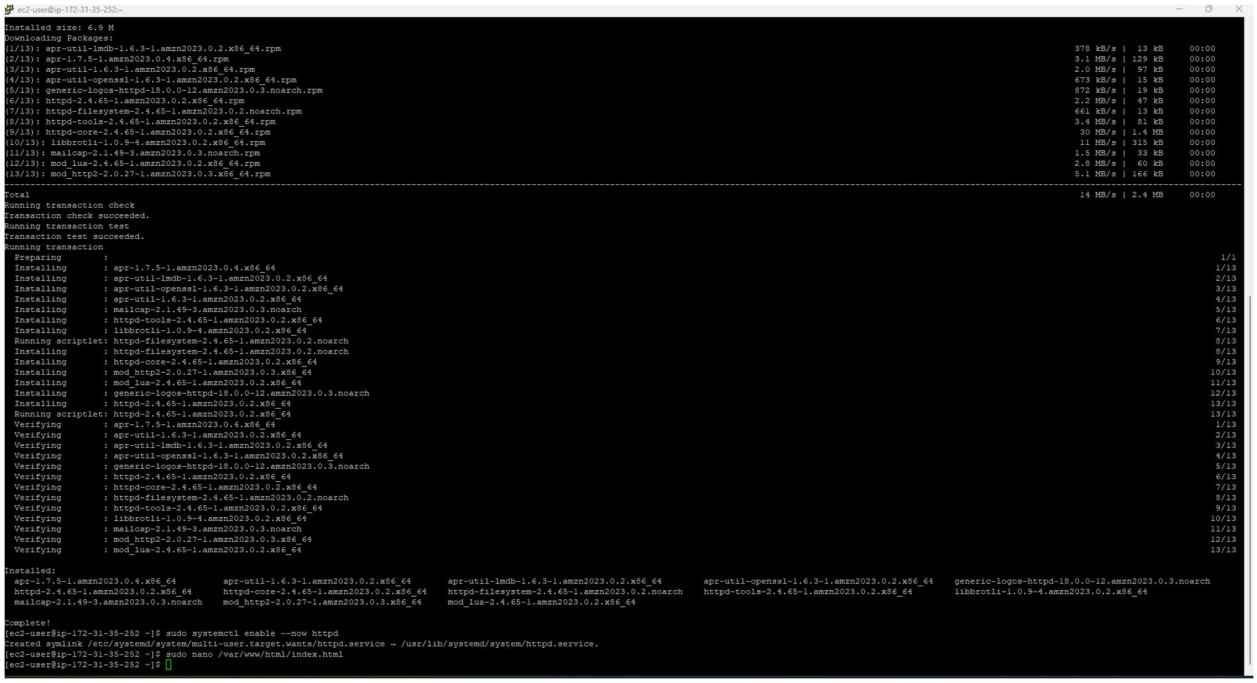
Inside /var/www/html/index.html, I created a basic web page that referenced the CloudFront-hosted CSS and the success image.

Once saved, loading the EC2 public IP displayed the final, formatted output — a blue background, a message, and the centered logo — confirming that both content paths were working.


ec2-user@ip-172-31-35-252:~
Installed size: 6.9 M
Downloading Packages:
(1/13): apr-util-lmdb-1.6.3-1.amzn2023.0.2.x86_64.rpm 378 kB/s | 13 kB 00:00
(2/13): apr-1.7.5-1.amzn2023.0.4.x86_64.rpm 3.1 MB/s | 129 kB 00:00
(3/13): apr-util-1.6.3-1.amzn2023.0.2.x86_64.rpm 2.0 MB/s | 97 kB 00:00
(4/13): apr-util-openssl-1.6.3-1.amzn2023.0.2.x86_64.rpm 673 kB/s | 15 kB 00:00
(5/13): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm 872 kB/s | 19 kB 00:00
(6/13): httpd-2.4.65-1.amzn2023.0.2.x86_64.rpm 2.2 MB/s | 47 kB 00:00
(7/13): httpd-filesystem-2.4.65-1.amzn2023.0.2.noarch.rpm 661 kB/s | 13 kB 00:00
(8/13): httpd-tools-2.4.65-1.amzn2023.0.2.x86_64.rpm 3.6 MB/s | 81 kB 00:00
(9/13): httpd-core-2.4.65-1.amzn2023.0.2.x86_64.rpm 30 MB/s | 1.4 MB 00:00
(10/13): libbrotli-1.0.9-4.amzn2023.0.2.x86_64.rpm 11 MB/s | 315 kB 00:00
(11/13): mailcap-2.1.49-3.amzn2023.0.3.noarch.rpm 1.5 MB/s | 33 kB 00:00
(12/13): mod_lua-2.4.65-1.amzn2023.0.2.x86_64.rpm 2.8 MB/s | 60 kB 00:00
(13/13): mod_http2-2.0.27-1.amzn2023.0.3.x86_64.rpm 5.1 MB/s | 166 kB 00:00
Total 14 MB/s | 2.4 MB 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : apr-1.7.5-1.amzn2023.0.4.x86_64 1/13
Installing : apr-util-lmdb-1.6.3-1.amzn2023.0.2.x86_64 2/13
Installing : apr-util-openssl-1.6.3-1.amzn2023.0.2.x86_64 3/13
Installing : apr-util-1.6.3-1.amzn2023.0.2.x86_64 4/13
Installing : mailcap-2.1.49-3.amzn2023.0.3.noarch 5/13
Installing : httpd-tools-2.4.65-1.amzn2023.0.2.x86_64 6/13
Installing : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 7/13
Running scriptlet: httpd-filesystem-2.4.65-1.amzn2023.0.2.noarch 8/13
Installing : httpd-filesystem-2.4.65-1.amzn2023.0.2.noarch 8/13
Installing : httpd-core-2.4.65-1.amzn2023.0.2.x86_64 9/13
Installing : mod_http2-2.0.27-1.amzn2023.0.3.x86_64 10/13
Installing : mod_lua-2.4.65-1.amzn2023.0.2.x86_64 11/13
Installing : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 12/13
Installing : httpd-2.4.65-1.amzn2023.0.2.x86_64 13/13
Running scriptlet: httpd-2.4.65-1.amzn2023.0.2.x86_64 13/13
Verifying : apr-1.7.5-1.amzn2023.0.4.x86_64 1/13
Verifying : apr-util-1.6.3-1.amzn2023.0.2.x86_64 2/13
Verifying : apr-util-lmdb-1.6.3-1.amzn2023.0.2.x86_64 3/13
Verifying : apr-util-openssl-1.6.3-1.amzn2023.0.2.x86_64 4/13
Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 5/13
Verifying : httpd-2.4.65-1.amzn2023.0.2.x86_64 6/13
Verifying : httpd-core-2.4.65-1.amzn2023.0.2.x86_64 7/13
Verifying : httpd-filesystem-2.4.65-1.amzn2023.0.2.noarch 8/13
Verifying : httpd-tools-2.4.65-1.amzn2023.0.2.x86_64 9/13
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 10/13
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch 11/13
Verifying : mod_http2-2.0.27-1.amzn2023.0.3.x86_64 12/13
Verifying : mod_lua-2.4.65-1.amzn2023.0.2.x86_64 13/13
Installed:
apr-1.7.5-1.amzn2023.0.4.x86_64 apr-util-1.6.3-1.amzn2023.0.2.x86_64 apr-util-lmdb-1.6.3-1.amzn2023.0.2.x86_64 apr-util-openssl-1.6.3-1.amzn2023.0.2.x86_64 generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-2.4.65-1.amzn2023.0.2.x86_64 httpd-core-2.4.65-1.amzn2023.0.2.x86_64 httpd-filesystem-2.4.65-1.amzn2023.0.2.noarch httpd-tools-2.4.65-1.amzn2023.0.2.x86_64 libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch mod_http2-2.0.27-1.amzn2023.0.3.x86_64 mod_lua-2.4.65-1.amzn2023.0.2.x86_64
Complete!
[ec2-user@ip-172-31-35-252 ~]$ sudo systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-35-252 ~]$ sudo nano /var/www/html/index.html
[ec2-user@ip-172-31-35-252 ~]$


ec2-user@ip-172-31-35-252:~
GNU nano 8.3 /var/www/html/index.html Modified
<!DOCTYPE html>
<html>
<head>
<title>Hello from Bayoakins!</title>
<link rel="stylesheet" href="https://d2ne2llt3qcb8s.cloudfront.net/style.css">
</head>
<body>
<h1>Hello from Bayoakins!</h1>

</body>
</html>
File Name to Write: /var/www/html/index.html
^G Help M-D DOS Format M-A Append M-B Backup File
^C Cancel M-M Mac Format M-P Prepend ^T Browse

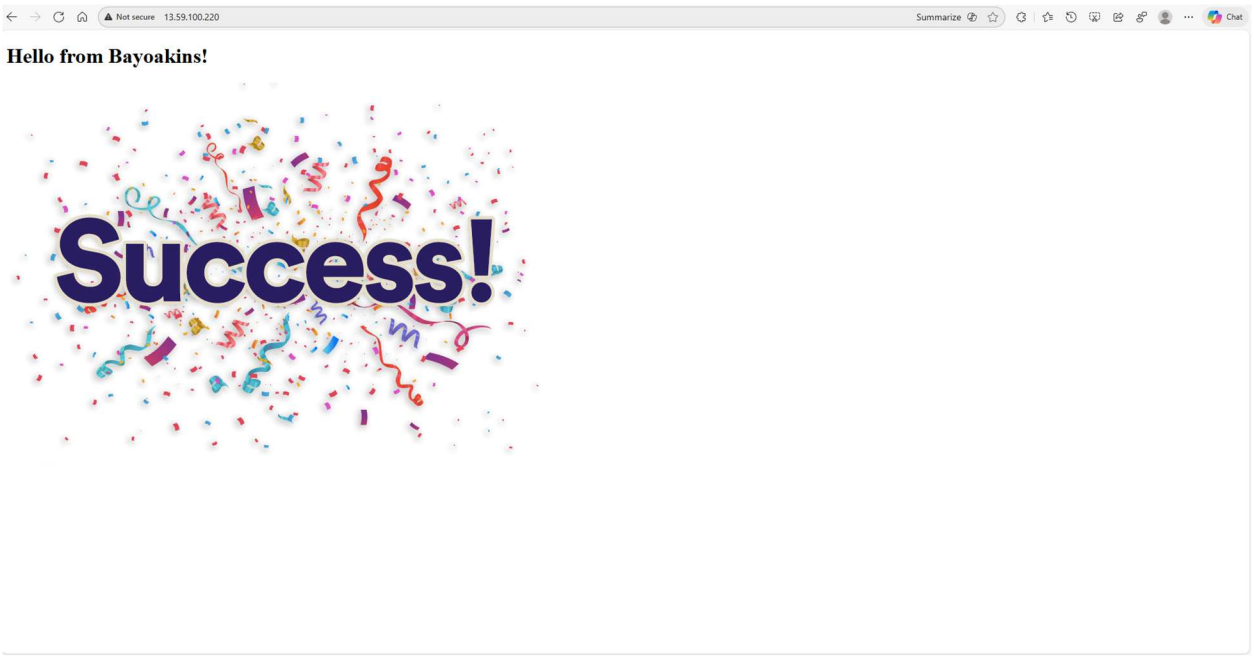


Testing Results

Once deployment was complete, I verified that:

- EC2 successfully served HTML
- S3 remained private (no direct access allowed)
- CSS and image loaded only through CloudFront
- The browser displayed all elements correctly

This validated that the hybrid architecture behaved as expected.

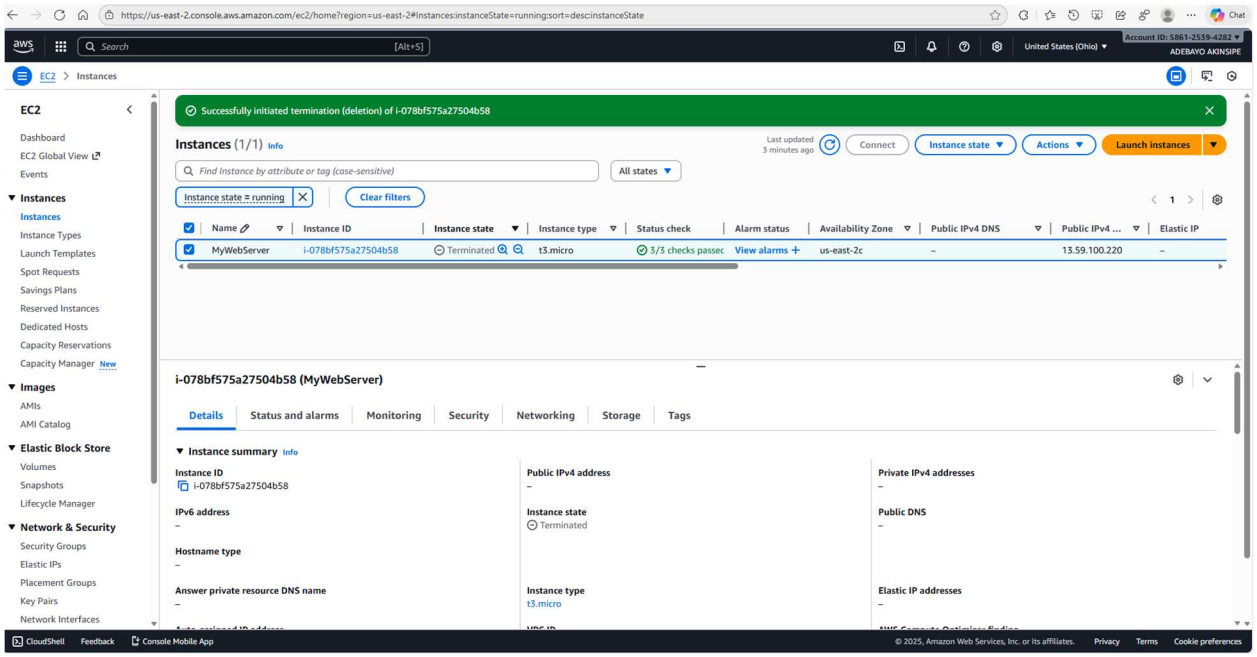


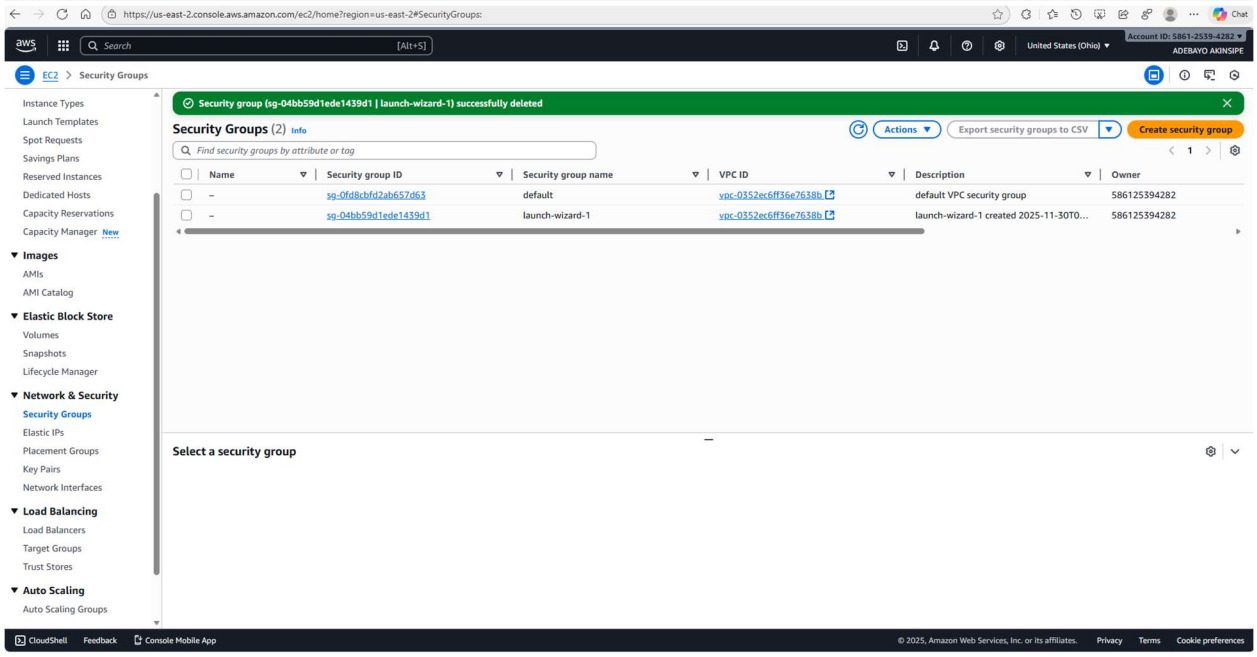
Cleanup

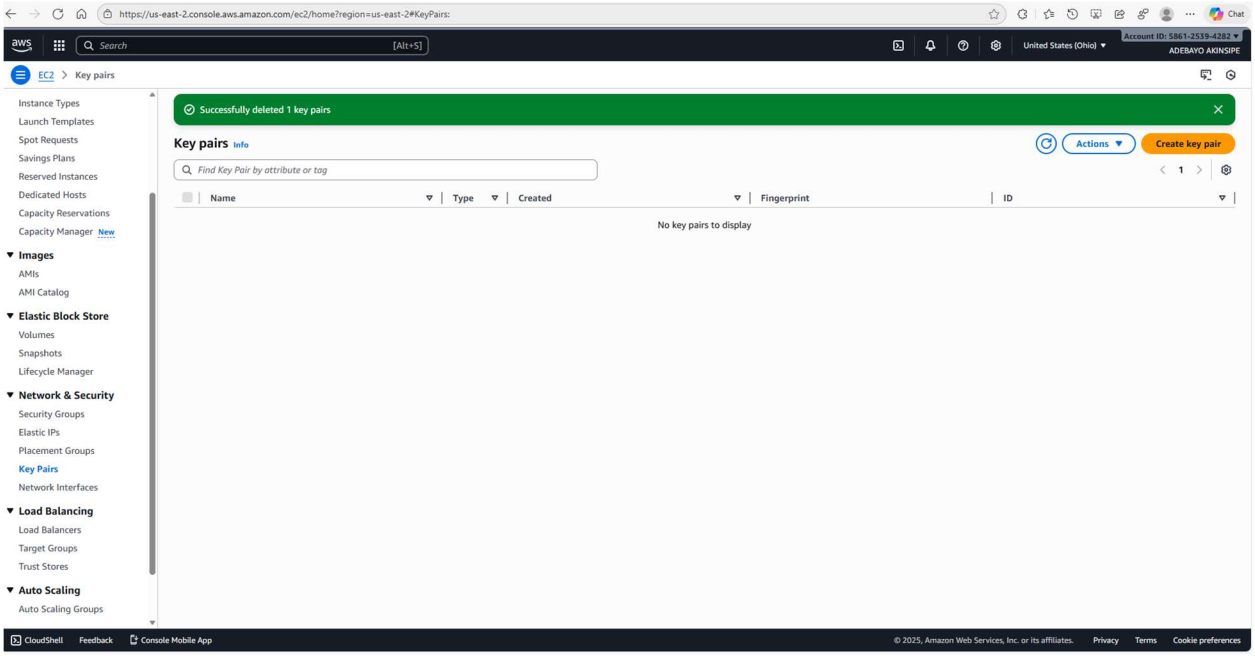
To avoid unnecessary billing, I deleted:

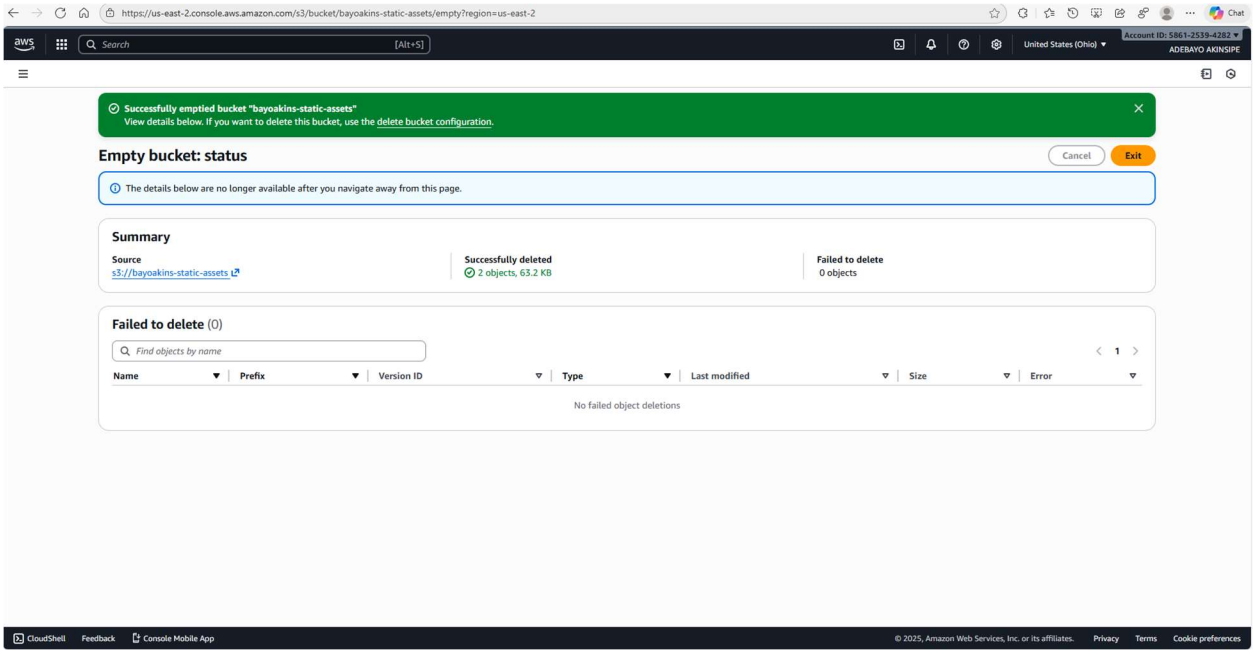
- The EC2 instance
- The security group
- The key pair
- The CloudFront distribution
- The S3 bucket

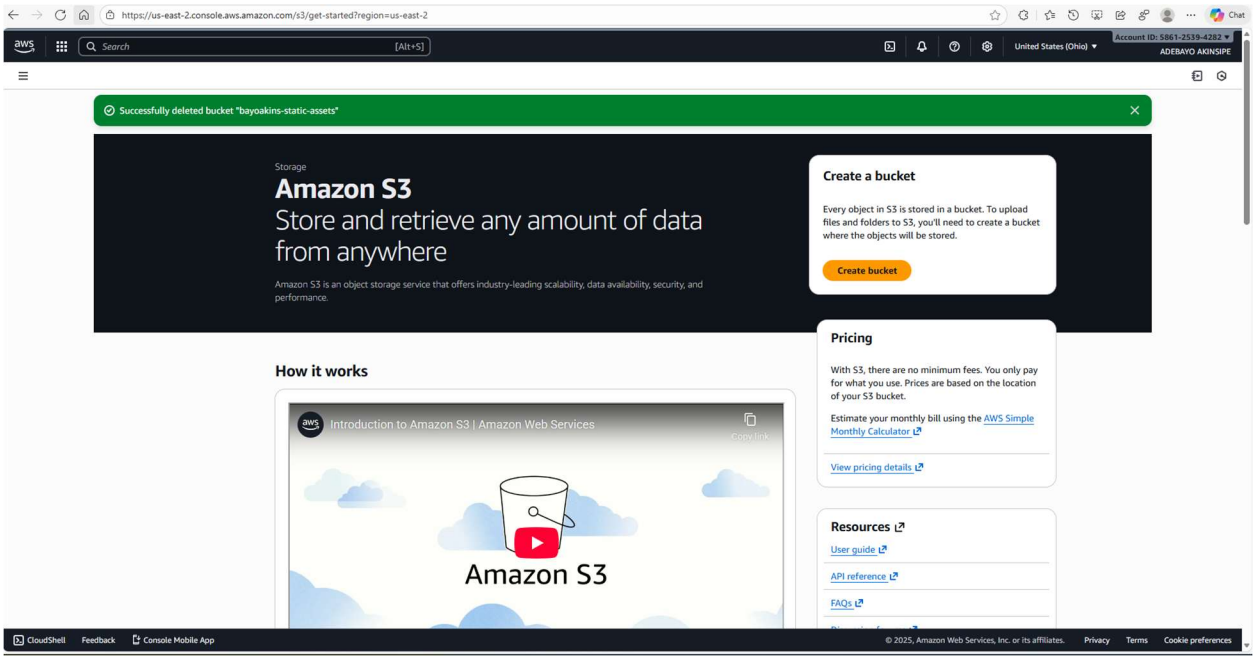
Cleanup is a crucial final step when working in a free-tier environment.


https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Instances:instanceState=running;sort=desc:instanceState
Search [Alt+S]
United States (Ohio)
ADEBAYO AKINSIPE
EC2 > Instances
Successfully initiated termination (deletion) of i-078bf575a27504b58
Instances (1/1) Info
Last updated 3 minutes ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Instance state = running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
MyWebServer
i-078bf575a27504b58
Terminated
t3.micro
3/3 checks passed
View alarms
us-east-2c
13.59.100.220
i-078bf575a27504b58 (MyWebServer)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-078bf575a27504b58
Public IPv4 address
Private IPv4 addresses
IPv6 address
Instance state
Terminated
Public DNS
Hostname type
Answer private resource DNS name
Instance type
t3.micro
Elastic IP addresses
EC2
Dashboard
EC2 Global View
Events
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#SecurityGroups:
Search [Alt+S]
United States (Ohio)
ADEBAYO AKINSIPE
EC2 > Security Groups
Security group (sg-04bb59d1ede1439d1 | launch-wizard-1) successfully deleted
Security Groups (2) Info
Actions
Export security groups to CSV
Create security group
Find security groups by attribute or tag
Name
Security group ID
Security group name
VPC ID
Description
Owner
sg-0fd8cbfd2ab657d63
default
vpc-0352ec6ff36e7638b
default VPC security group
586125394282
sg-04bb59d1ede1439d1
launch-wizard-1
launch-wizard-1 created 2025-11-30T0...
Select a security group
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores
Auto Scaling
Auto Scaling Groups
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#KeyPairs:
Search [Alt+S]
United States (Ohio)
ADEBAYO AKINSIPE
EC2 > Key pairs
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores
Auto Scaling
Auto Scaling Groups
Successfully deleted 1 key pairs
Key pairs Info
Actions
Create key pair
Find Key Pair by attribute or tag
Name
Type
Created
Fingerprint
ID
No key pairs to display
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


https://us-east-2.console.aws.amazon.com/s3/bucket/bayoakins-static-assets/empty?region=us-east-2
Search [Alt+S]
United States (Ohio)
ADEBAYO AKINSIPE
Successfully emptied bucket "bayoakins-static-assets"
View details below. If you want to delete this bucket, use the delete bucket configuration.
Empty bucket: status
Cancel
Exit
The details below are no longer available after you navigate away from this page.
Summary
Source
s3://bayoakins-static-assets
Successfully deleted
2 objects, 63.2 KB
Failed to delete
0 objects
Failed to delete (0)
Find objects by name
Name
Prefix
Version ID
Type
Last modified
Size
Error
No failed object deletions
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



Troubleshooting Moment

My first test returned missing images — a classic **403 "Access Denied"** error. The issue was that the HTML still referenced the S3 object URL instead of the CloudFront domain. After replacing the link, the assets loaded successfully.

Security + Performance Reflection

This project reinforced some important AWS design lessons:

- **Decoupling Compute and Content Delivery** improves scalability and cost efficiency
- **CloudFront caching** reduces latency and offloads traffic from the web server
- **Private S3 buckets with controlled access paths** reduce the attack surface
- **Security groups and minimal access controls** enforce the principle of least privilege

My Final Takeaways

This hands-on build helped me apply architectural reasoning rather than just service configuration. Working through each step deepened my understanding of how AWS services integrate — especially when designing secure, cost-optimized, and scalable web applications.

Next, I plan to extend this architecture with:

- HTTPS configuration
- Route 53 DNS mapping
- Lambda@Edge caching logic

Each improvement brings the build closer to production readiness — and one step further in my AWS Solutions Architect journey.