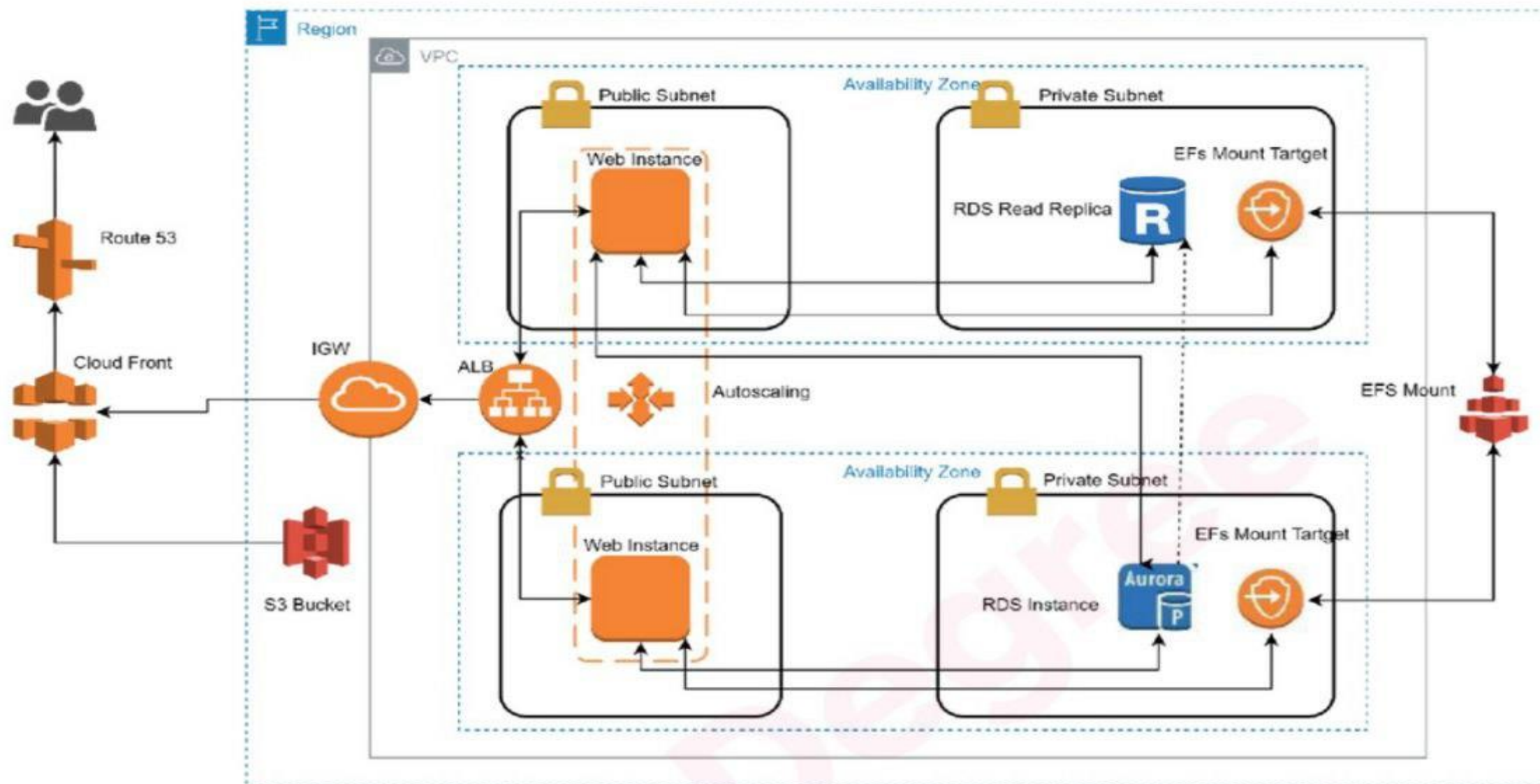


Abstract geometric lines in the top left corner, consisting of several overlapping, tilted rectangles and polygons in a light beige color.

AWS

Assignment Week 4

AWS Cloud architecture diagram by - MicroDegree



Clue * - Custom AMI should contain nginx and a custom index.html stating " This webpage is launched using custom AMI, in a Autoscaling group".

The screenshot shows the 'Edit user data' page in the AWS Management Console. The breadcrumb navigation at the top reads: EC2 > Instances > i-047187bf2d4817490 > Edit user data. The page title is 'Edit user data | EC2 | us-east-1'. The URL in the address bar is '374709127127-7u5sqm4p.us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#EditUserData:instanceId=i-047187bf2d4817490'. The AWS account ID is 3747-0912-7127 and the user is GangadharK@12. The page contains a 'Copy user data' button, a 'New user data' section with two radio buttons, and a text area for user data. The text area contains the following commands:

```
#!/bin/bash
yum update -y
yum install -y nginx
echo "This webpage is launched using custom AMI, in a Autoscaling group" > /var/www/html/index.html
systemctl start nginx
systemctl enable nginx
```

 There is an unchecked checkbox for 'Input is already base64-encoded'. At the bottom right are 'Cancel' and 'Save' buttons. The Windows taskbar at the bottom shows the date and time as 3:48 AM on 5/19/2025.

systemctl enable nginx

[Copy user data](#)

New user data
This user data will replace the current user data

☒ **Modify user data as text**
Add your user data below

☐ **Modify user data by importing a file**
Description of importing a file and what will happen to it

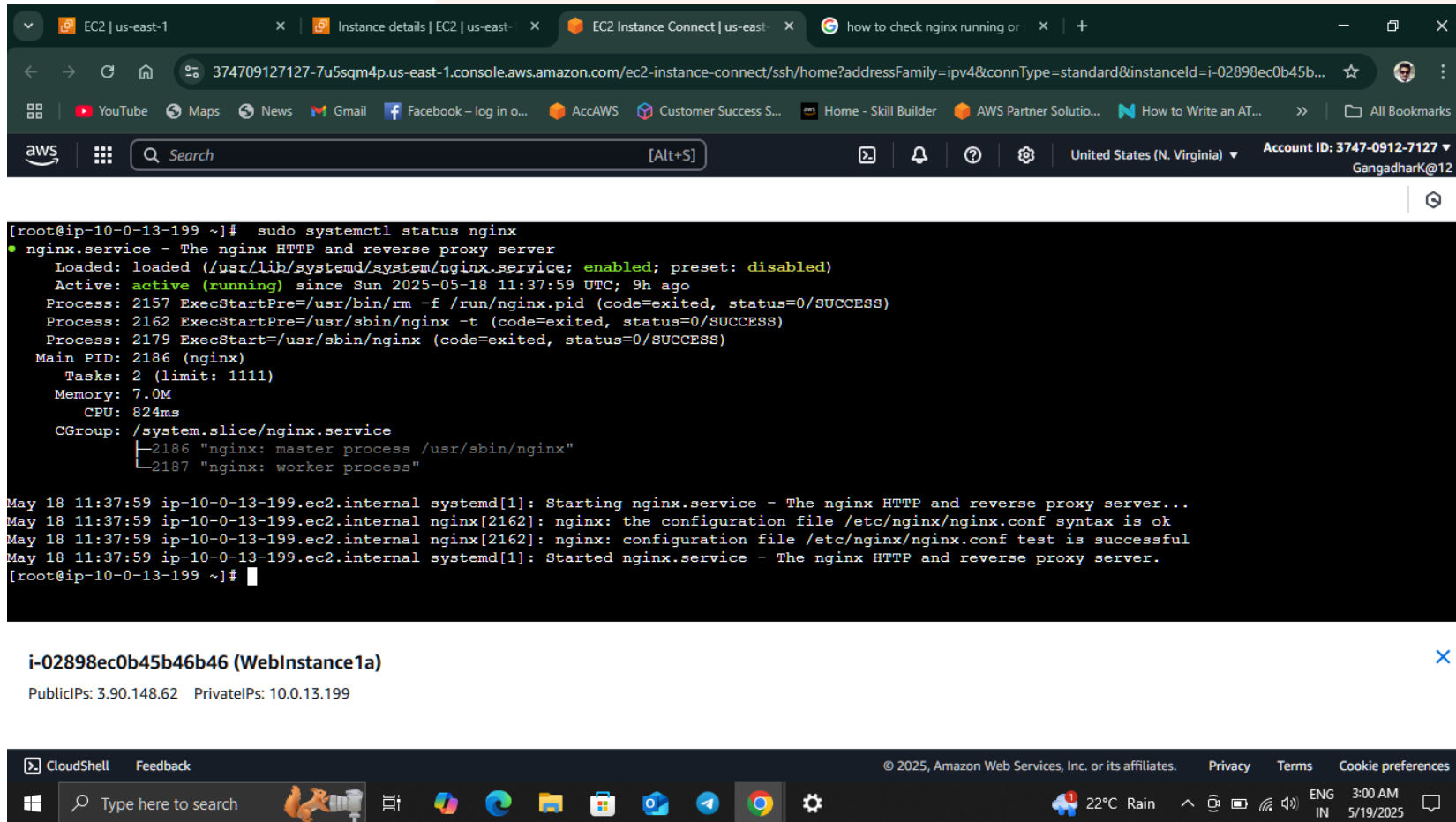
```
#!/bin/bash
yum update -y
yum install -y nginx
echo "This webpage is launched using custom AMI, in a Autoscaling group" > /var/www/html/index.html
systemctl start nginx
systemctl enable nginx
```

☐ Input is already base64-encoded

[Cancel](#) [Save](#)

EC2 User data

Clue * - Custom AMI should contain nginx and a custom index.html stating " This webpage is launched using custom AMI, in a Autoscaling group".



The screenshot shows the AWS Management Console with a terminal window open on an EC2 instance. The terminal output shows the command `sudo systemctl status nginx` and its output, which indicates that nginx is active and running. Below the terminal, the instance details for `i-02898ec0b45b46b46 (WebInstance1a)` are visible, showing public and private IP addresses. At the bottom, there is a Windows taskbar with various application icons and system information.

```
[root@ip-10-0-13-199 ~]# sudo systemctl status nginx
• nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Sun 2025-05-18 11:37:59 UTC; 9h ago
     Process: 2157 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 2162 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 2179 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
  Main PID: 2186 (nginx)
    Tasks: 2 (limit: 1111)
   Memory: 7.0M
      CPU: 824ms
  CGroup: /system.slice/nginx.service
          └─2186 "nginx: master process /usr/sbin/nginx"
            └─2187 "nginx: worker process"

May 18 11:37:59 ip-10-0-13-199.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
May 18 11:37:59 ip-10-0-13-199.ec2.internal nginx[2162]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
May 18 11:37:59 ip-10-0-13-199.ec2.internal nginx[2162]: nginx: configuration file /etc/nginx/nginx.conf test is successful
May 18 11:37:59 ip-10-0-13-199.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[root@ip-10-0-13-199 ~]#
```

i-02898ec0b45b46b46 (WebInstance1a)

PublicIPs: 3.90.148.62 PrivateIPs: 10.0.13.199

CloudShell Feedback

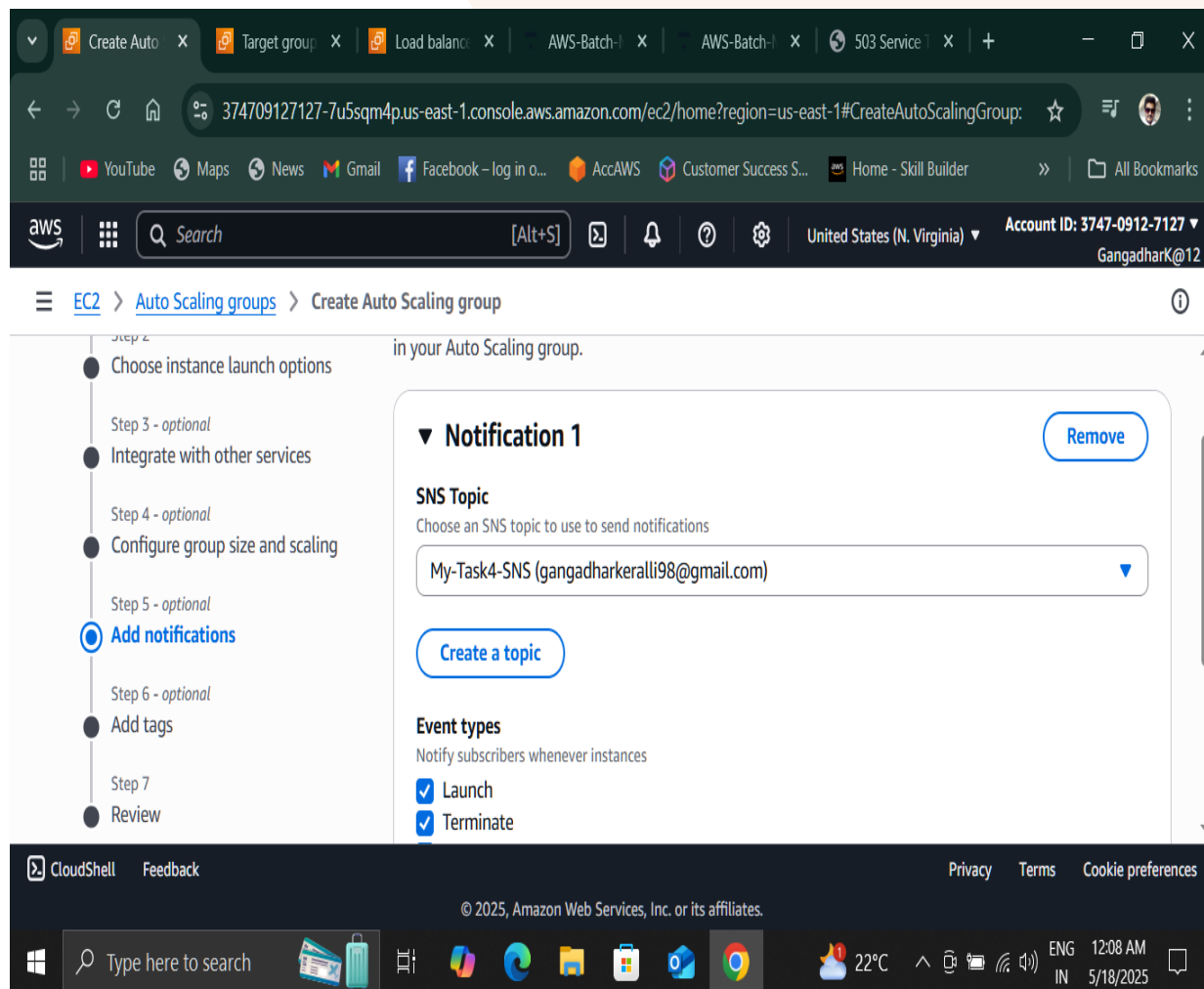
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search

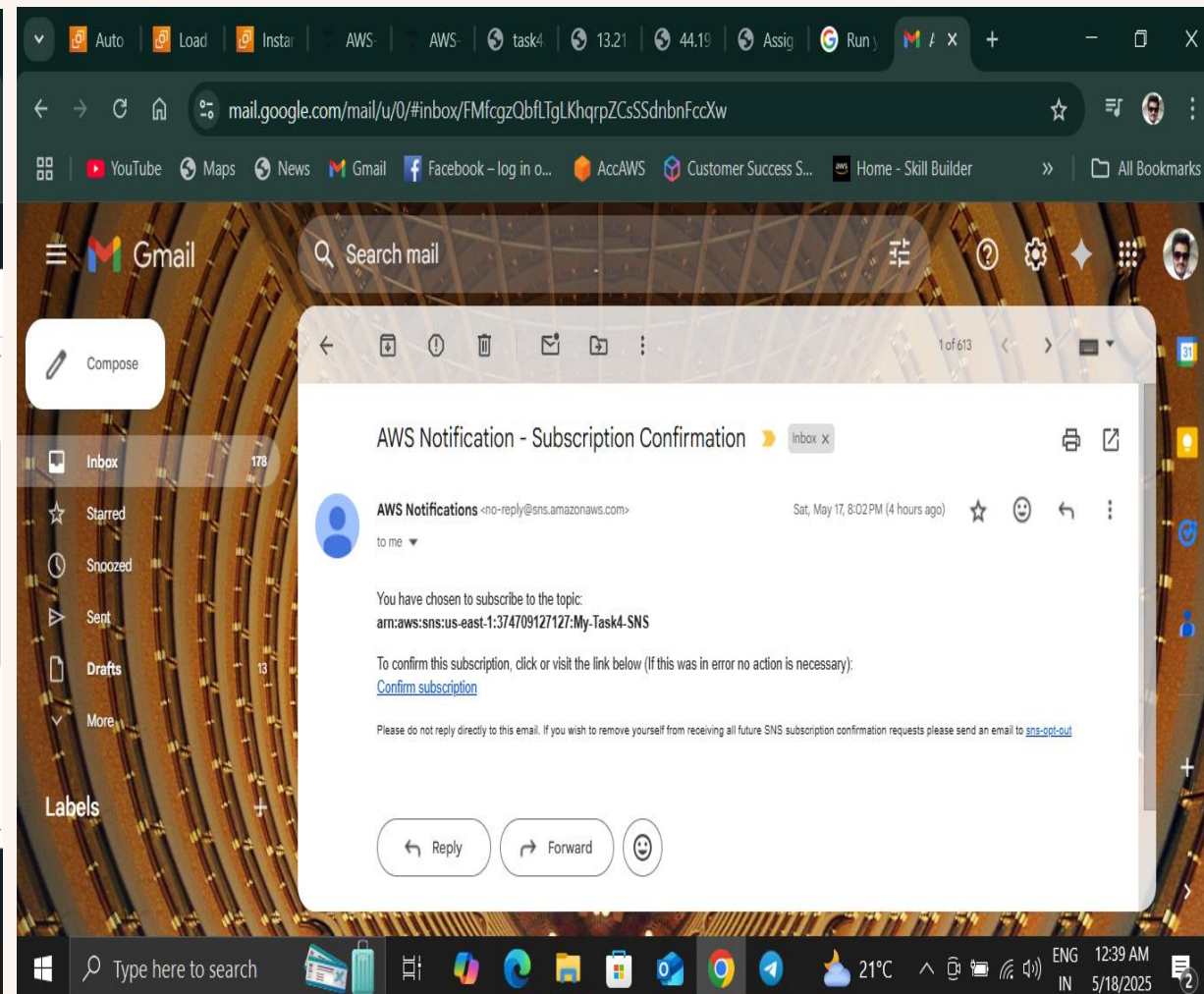
22°C Rain ENG IN 3:00 AM 5/19/2025

Nginx is running

Clue ** - SNS notifications.

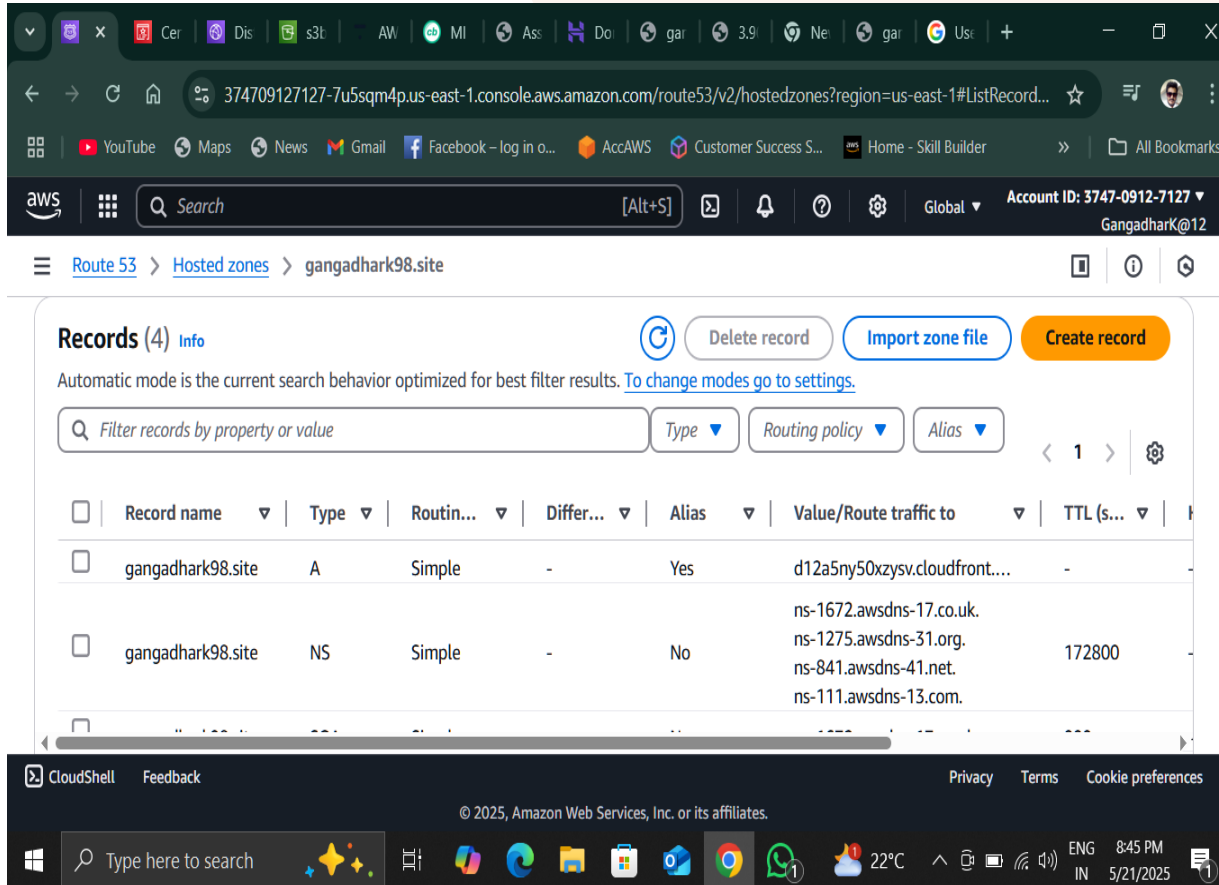


SNS topic created



Received notification

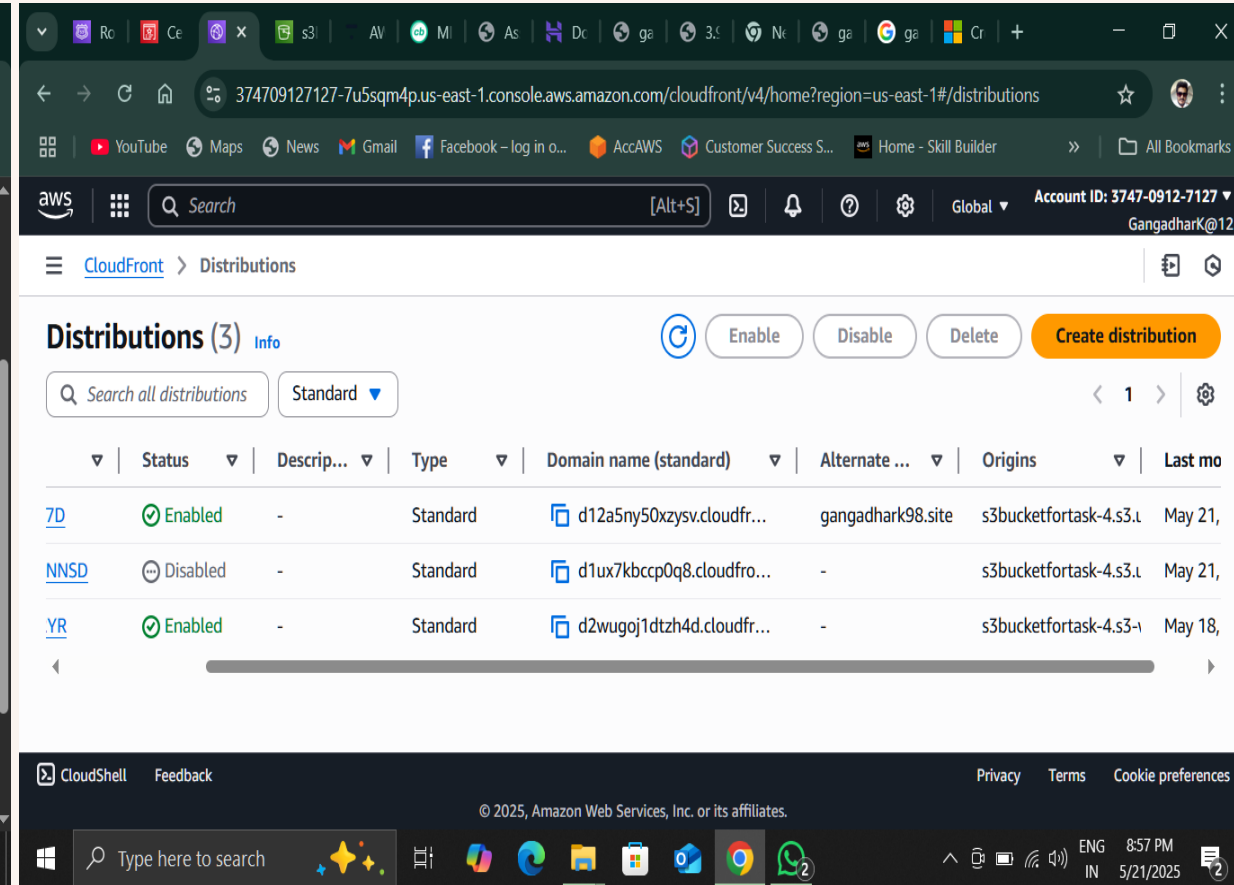
Step 1 - Use Amazon Route53 to connect Domain Name and use Amazon CloudFront to deliver static and dynamic content.



The screenshot shows the Amazon Route 53 console. The breadcrumb navigation indicates the path: Route 53 > Hosted zones > gangadhark98.site. The 'Records (4)' section is active, displaying a table of DNS records for the domain gangadhark98.site. The table includes columns for Record name, Type, Routing policy, Alias, Value/Route traffic to, and TTL. Two records are visible: an A record pointing to d12a5ny50xyzsv.cloudfront... and an NS record pointing to ns-1672.awsdns-17.co.uk, ns-1275.awsdns-31.org, ns-841.awsdns-41.net, and ns-111.awsdns-13.com.

Record name	Type	Routing policy	Alias	Value/Route traffic to	TTL (s...)
gangadhark98.site	A	Simple	Yes	d12a5ny50xyzsv.cloudfront...	-
gangadhark98.site	NS	Simple	No	ns-1672.awsdns-17.co.uk. ns-1275.awsdns-31.org. ns-841.awsdns-41.net. ns-111.awsdns-13.com.	172800

Hosted zone & Record created

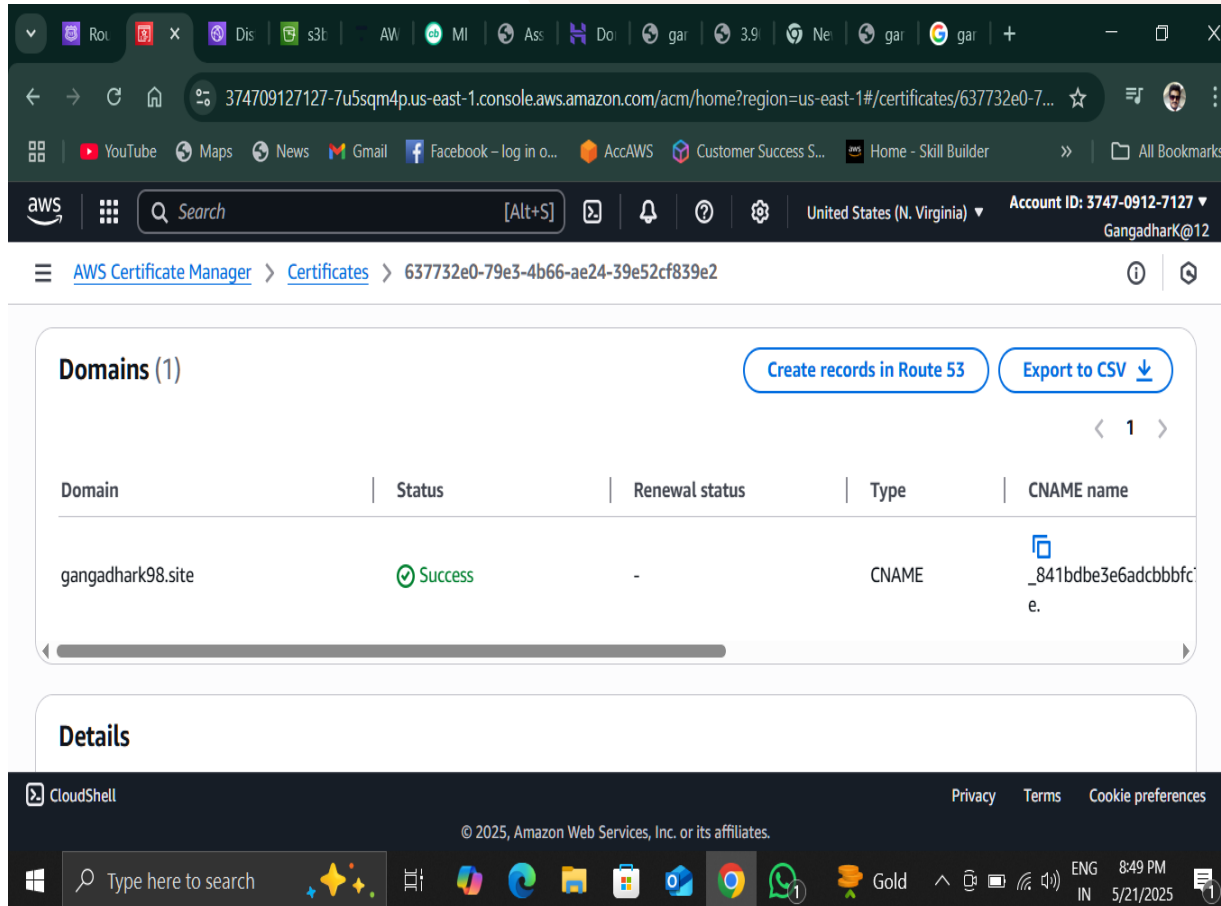


The screenshot shows the Amazon CloudFront console. The breadcrumb navigation indicates the path: CloudFront > Distributions. The 'Distributions (3)' section is active, displaying a table of distributions. The table includes columns for Status, Description, Type, Domain name (standard), Alternate domain names, Origins, and Last modified. Three distributions are visible: 7D (Enabled), NNSD (Disabled), and YR (Enabled). The origins for each distribution are listed as s3bucketfortask-4.s3.amazonaws.com.

Status	Descrip...	Type	Domain name (standard)	Alternate ...	Origins	Last mo
Enabled	-	Standard	d12a5ny50xyzsv.cloudfr...	gangadhark98.site	s3bucketfortask-4.s3.u	May 21,
Disabled	-	Standard	d1ux7kbccp0q8.cloudfro...	-	s3bucketfortask-4.s3.u	May 21,
Enabled	-	Standard	d2wugoj1dtzh4d.cloudfr...	-	s3bucketfortask-4.s3-1	May 18,

CloudFront distribution created & Origins mapped from S3

Step 1 - Use Amazon Route53 to connect Domain Name and use Amazon CloudFront to deliver static and dynamic content.

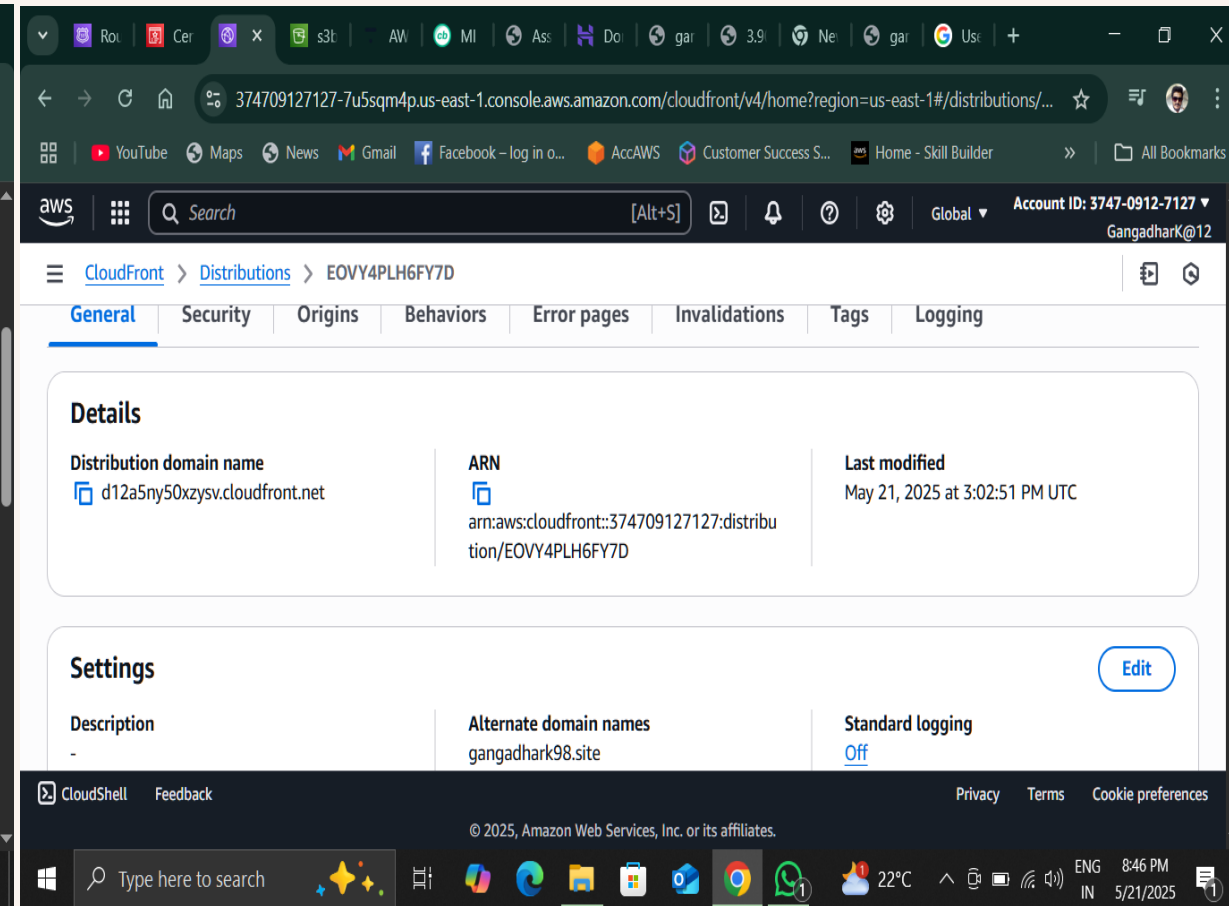


The screenshot shows the AWS Certificate Manager console. The breadcrumb navigation is 'AWS Certificate Manager > Certificates > 637732e0-79e3-4b66-ae24-39e52cf839e2'. The 'Domains (1)' section contains a table with one entry:

Domain	Status	Renewal status	Type	CNAME name
gangadhark98.site	Success	-	CNAME	_841bdbe3e6adcbbfce.

Buttons for 'Create records in Route 53' and 'Export to CSV' are visible. The 'Details' section is partially visible at the bottom.

SSL certificate



The screenshot shows the AWS CloudFront console. The breadcrumb navigation is 'CloudFront > Distributions > EOY4PLH6FY7D'. The 'General' tab is selected. The 'Details' section shows:

- Distribution domain name: d12a5ny50xyzsv.cloudfront.net
- ARN: arn:aws:cloudfront::374709127127:distribution/EOVY4PLH6FY7D
- Last modified: May 21, 2025 at 3:02:51 PM UTC

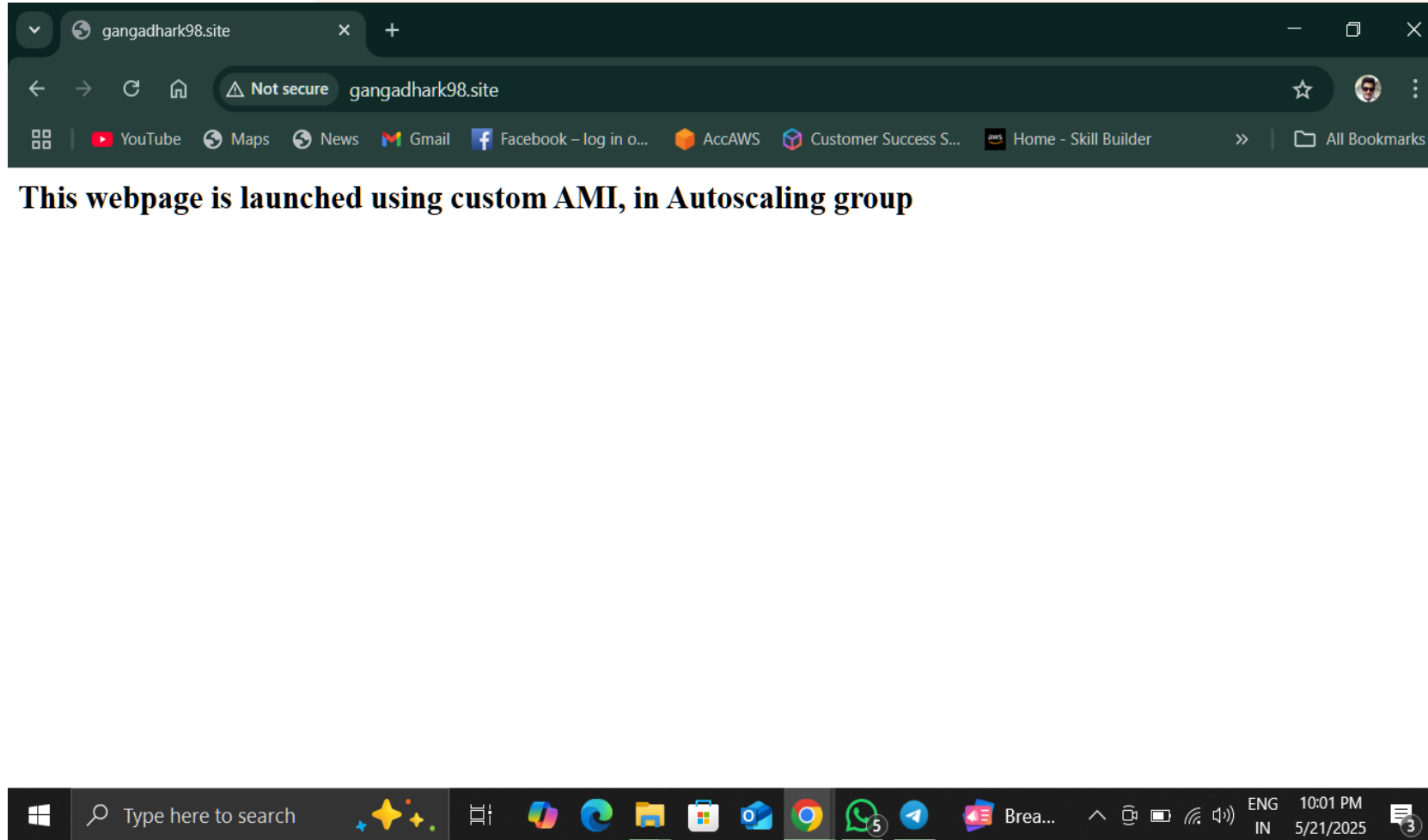
The 'Settings' section shows:

- Description: -
- Alternate domain names: gangadhark98.site
- Standard logging: Off

An 'Edit' button is visible in the top right of the Settings section.

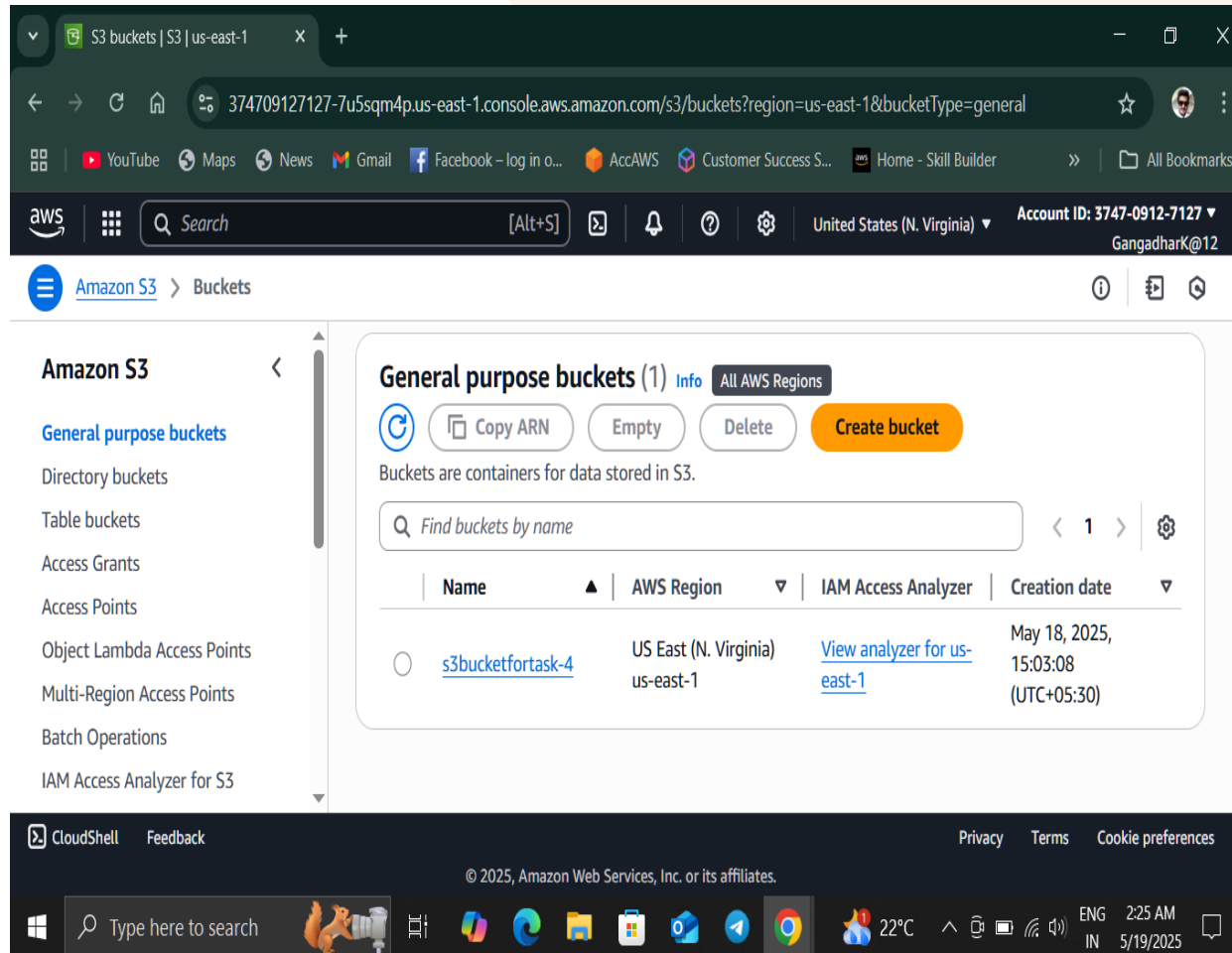
Deployment done in the CloudFront distribution

Step 1 - Use Amazon Route53 to connect Domain Name and use Amazon CloudFront to deliver static and dynamic content.

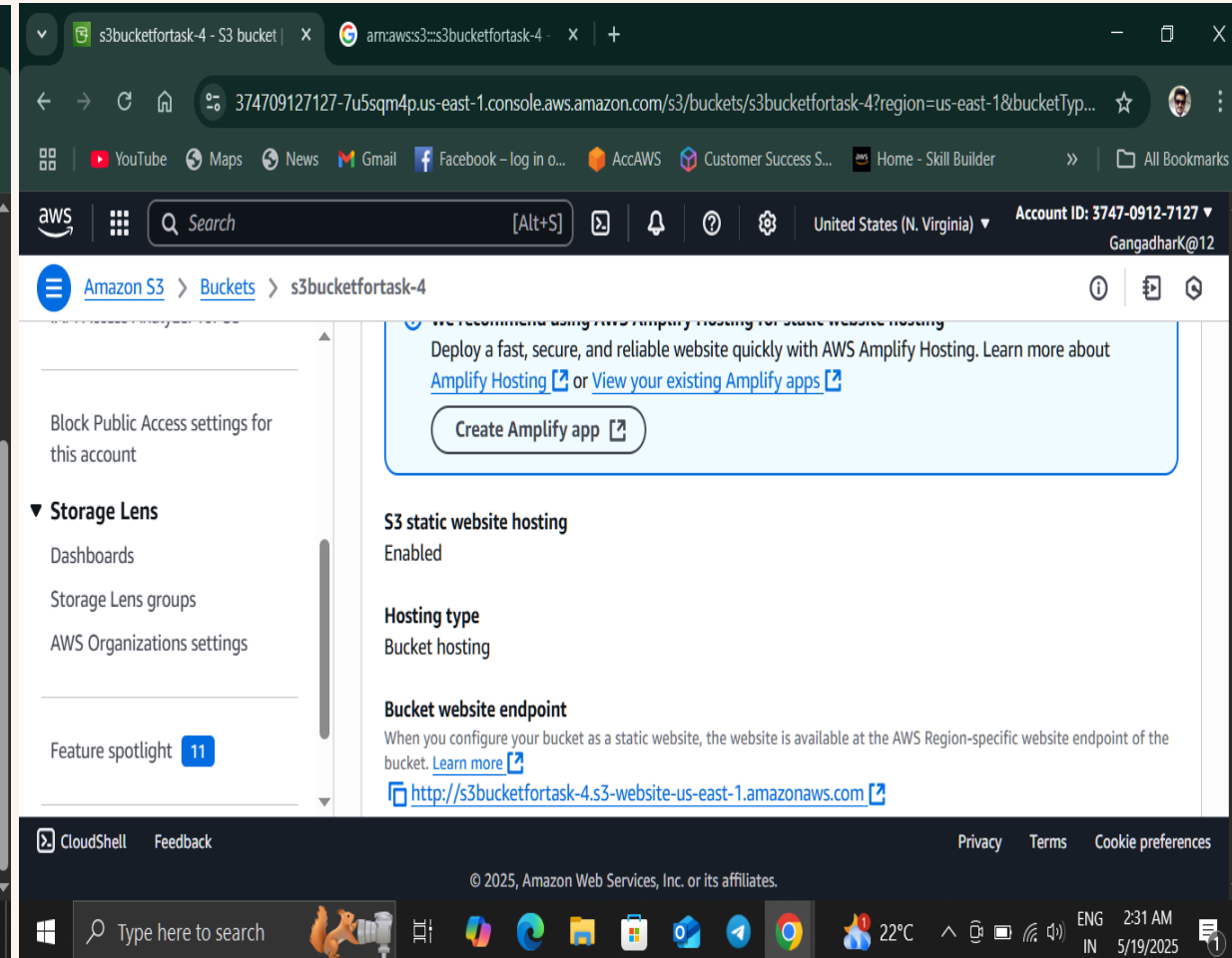


Finally Accessed
with my Domain
Name

Step 2 - Used Amazon S3 to store the static web connect.



S3 Bucket created



S3 static website hosting enabled

Step 3 - Internet Gateway is attached to VPC.

The screenshot displays the AWS Management Console interface for a VPC in the us-east-1 region. The breadcrumb navigation shows the path: VPC > Internet gateways > igw-07ff398ebde5fbb48. The main content area is titled "igw-07ff398ebde5fbb48 / Task4VPC-igw".

Details

Internet gateway ID	State	VPC ID	Owner
igw-07ff398ebde5fbb48	Attached	vpc-0c8b90278639d2c20 Task4VPC	374709127127

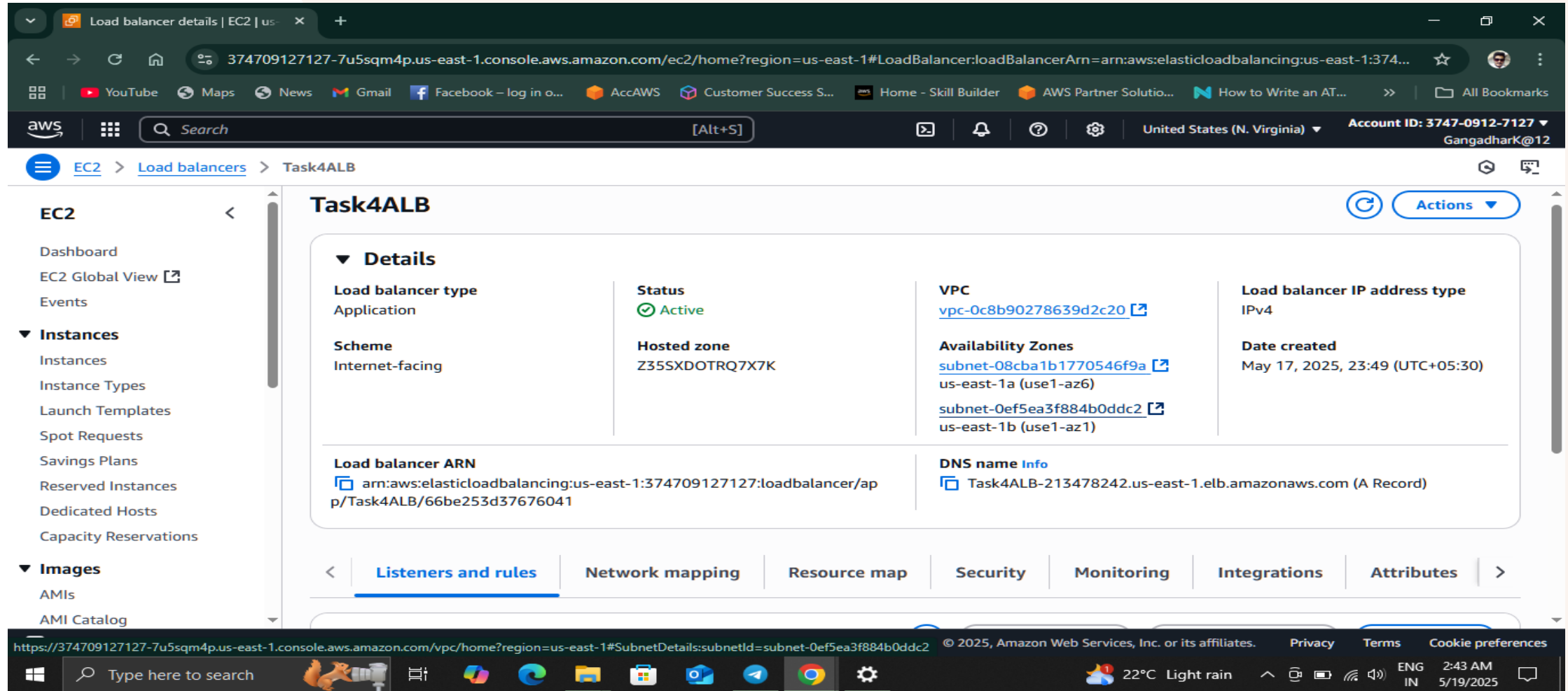
Tags

Search tags: []

Key	Value
-----	-------

The console footer includes the text "© 2025, Amazon Web Services, Inc. or its affiliates." and a Windows taskbar at the bottom with a search bar and various application icons.

Step 4 - ALB is used to distribute the web traffic across the ASG of EC2 instances in multiple AZ's.



The screenshot displays the AWS Management Console interface for an Application Load Balancer (ALB) named 'Task4ALB'. The browser address bar shows the URL: `374709127127-7u5sqm4p.us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancer:loadBalancerArn=arn:aws:elasticloadbalancing:us-east-1:374...`. The console header includes the AWS logo, a search bar, and account information for 'United States (N. Virginia)' with Account ID '3747-0912-7127'.

The left-hand navigation pane shows the 'EC2' section expanded, with 'Load balancers' selected. The main content area displays the 'Task4ALB' details page. The 'Details' tab is active, showing the following information:

- Load balancer type:** Application
- Status:** Active (indicated by a green checkmark)
- VPC:** [vpc-0c8b90278639d2c20](#)
- Load balancer IP address type:** IPv4
- Scheme:** Internet-facing
- Hosted zone:** Z35SXDOTRQ7X7K
- Availability Zones:** [subnet-08cba1b1770546f9a](#) (us-east-1a (use1-az6)) and [subnet-0ef5ea3f884b0ddc2](#) (us-east-1b (use1-az1))
- Date created:** May 17, 2025, 23:49 (UTC+05:30)
- Load balancer ARN:** [arn:aws:elasticloadbalancing:us-east-1:374709127127:loadbalancer/ap/Task4ALB/66be253d37676041](#)
- DNS name:** [Task4ALB-213478242.us-east-1.elb.amazonaws.com](#) (A Record)

Below the details, there are tabs for 'Listeners and rules', 'Network mapping', 'Resource map', 'Security', 'Monitoring', 'Integrations', and 'Attributes'. The 'Listeners and rules' tab is currently selected.

The bottom of the image shows the Windows taskbar with the search bar, taskbar icons, and system tray information including the date '5/19/2025' and time '2:43 AM'.

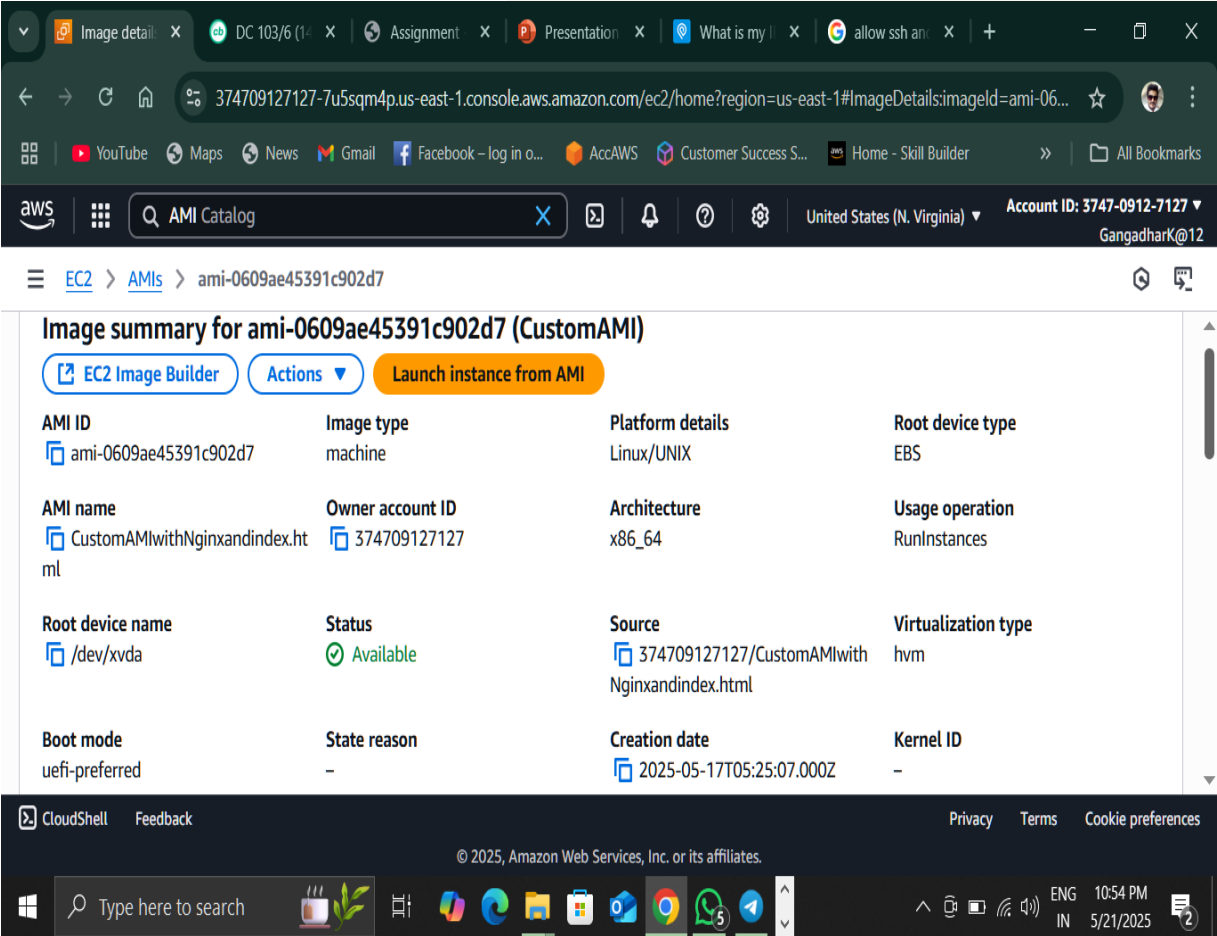
Step 5 - Allow SSH and db connection to your own laptop ip with security group for web instances.(This will act as Bastion server for itself. Allow http port 80 for all.)

The screenshot shows the AWS Management Console interface for a security group named 'sg-07ccd4d9bd8f64306 - SGLaunchTemplate'. The 'Inbound rules (5)' section is active, displaying a table of five rules. The rules are as follows:

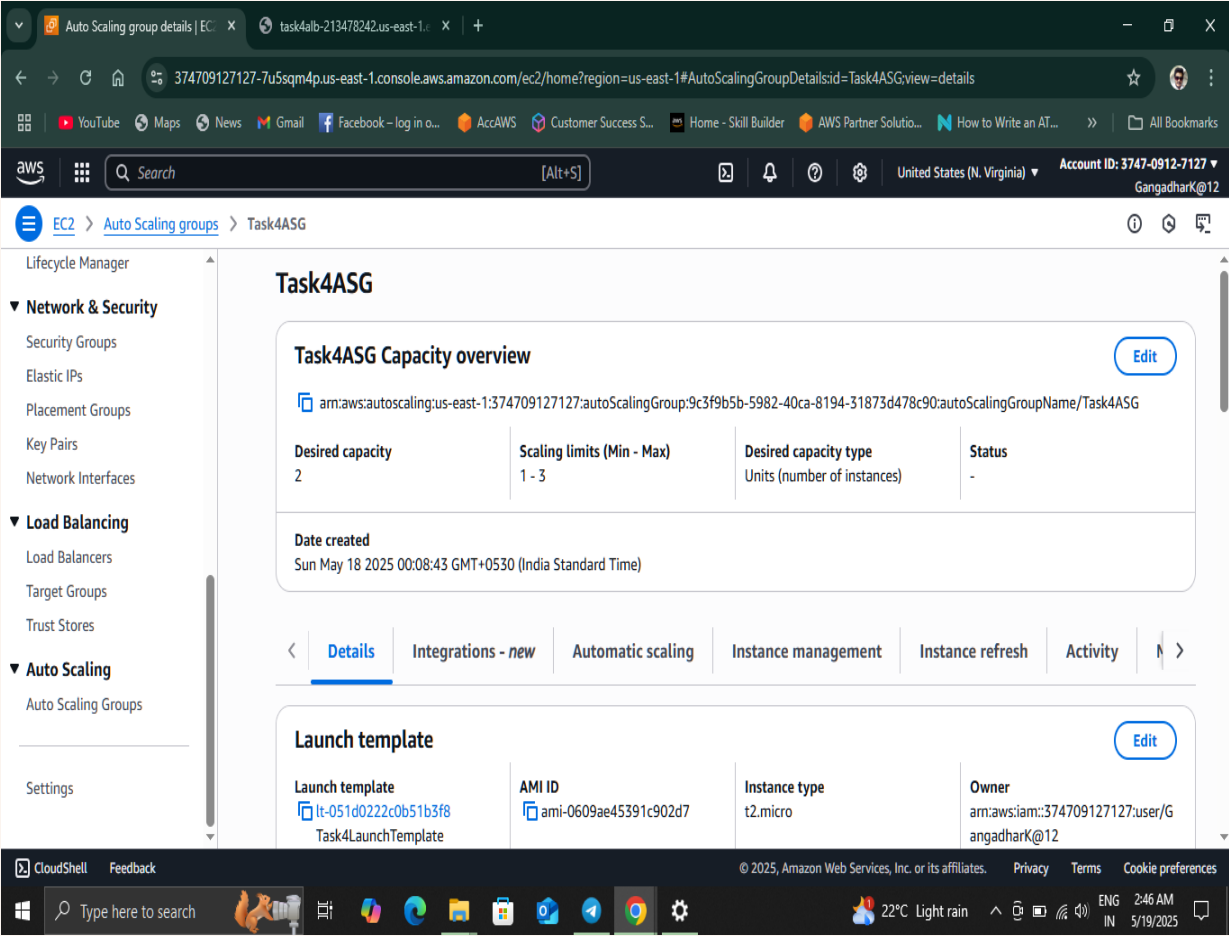
Security group rule ID	IP version	Type	Protocol	Port range
sgr-04c6f8467049a9ec2	IPv4	NFS	TCP	2049
sgr-05b1d476546b5d53b	IPv4	SSH	TCP	22
sgr-0f10f3106de50cc26	IPv4	All traffic	All	All
sgr-07ac5d76caddce39	IPv4	MYSQL/Aurora	TCP	3306
sgr-0972101e113c025c9	IPv4	HTTP	TCP	80

The interface also includes a search bar, 'Manage tags' and 'Edit inbound rules' buttons, and a pagination indicator showing '1' of 1 rules. The bottom of the screen shows the Windows taskbar with various application icons and system information like the date and time (10:47 PM, 5/21/2025).

Step 6 - Run your website using an ASG of EC2 instances using custom AMI (nginx & index.html).

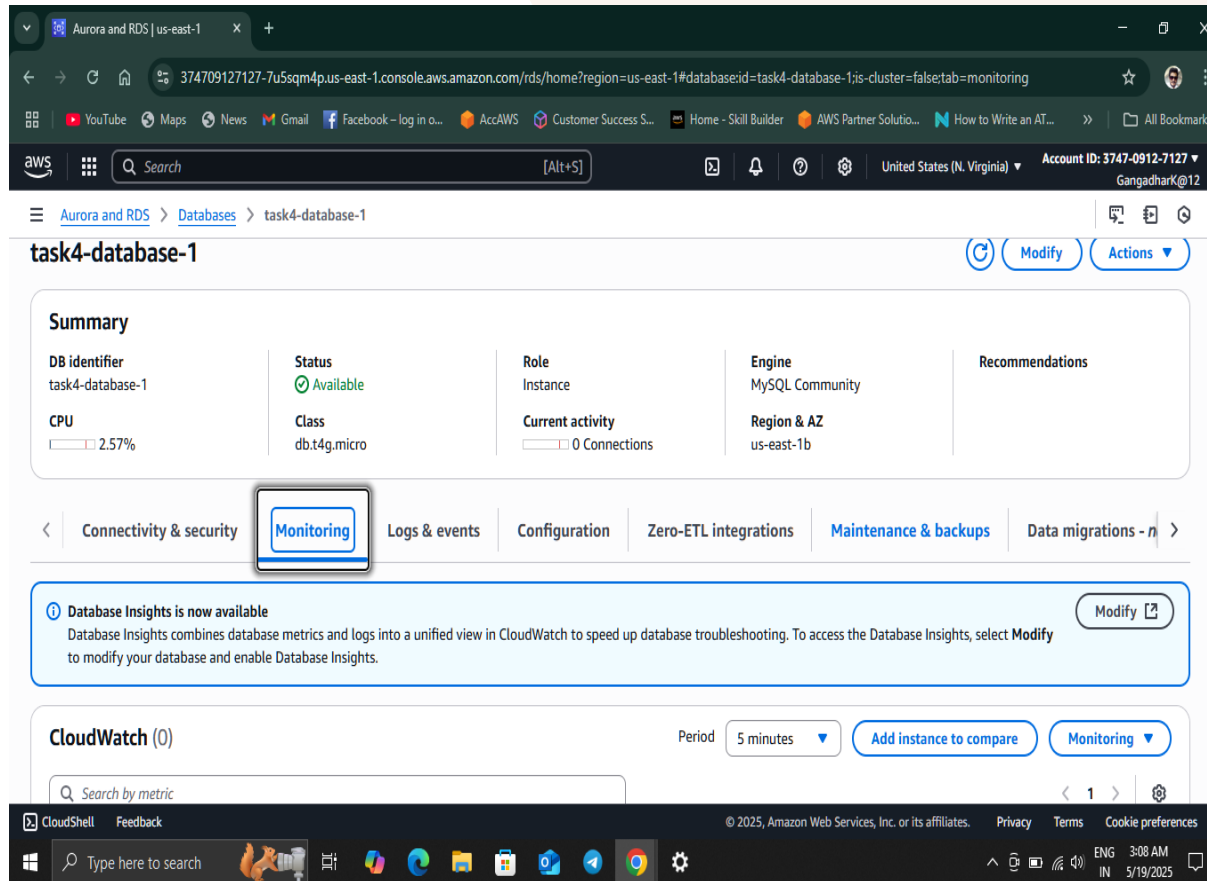


Custom AMI



Launch template & ASG created

Step 8 - Run your database layer in amazon RDS with aurora or MySQL to simplify your database administration.

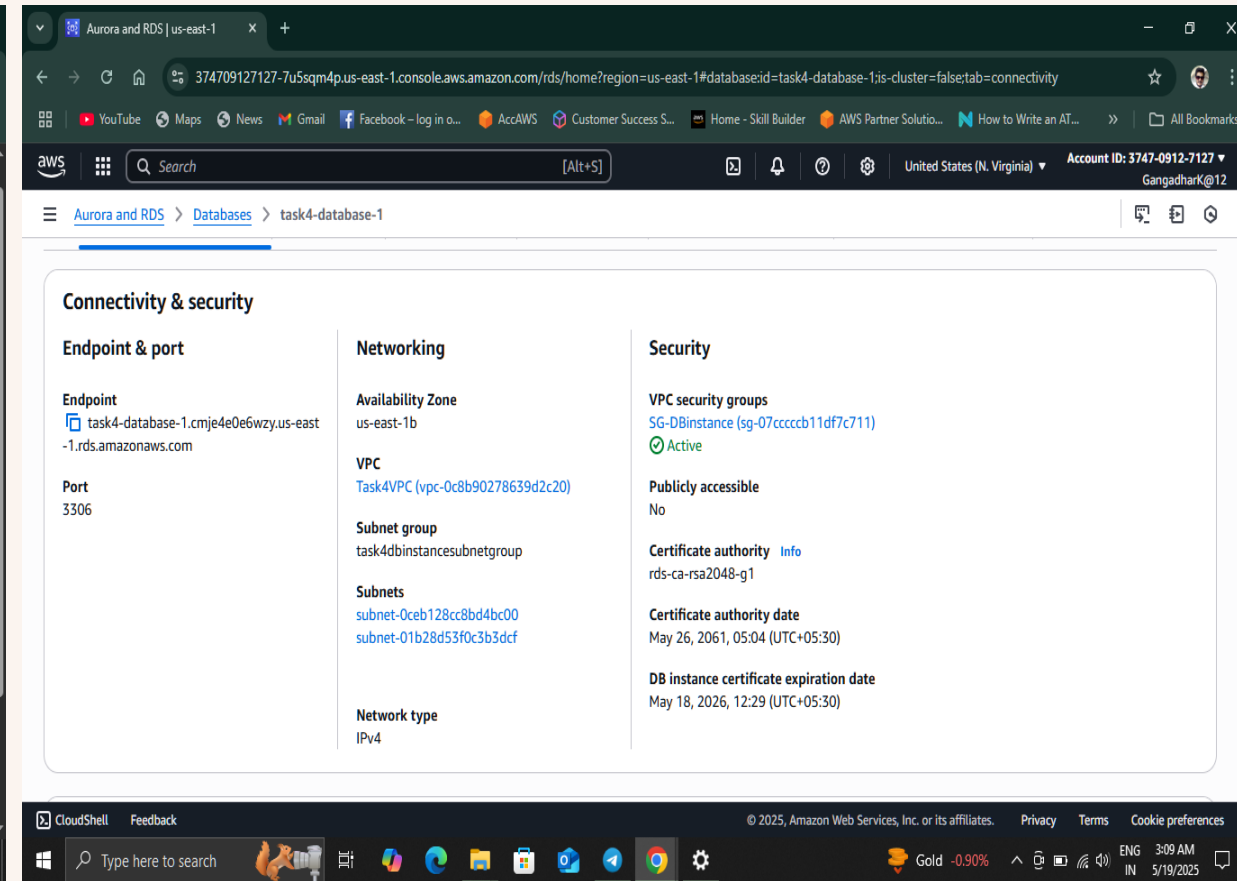


The screenshot shows the AWS RDS console for a database instance named 'task4-database-1'. The 'Monitoring' tab is selected and highlighted with a red box. The 'Summary' section displays the following details:

DB identifier	Status	Role	Engine	Recommendations
task4-database-1	Available	Instance	MySQL Community	

Additional details shown include CPU usage at 2.57%, Class 'db.t4g.micro', and 0 connections. The 'Connectivity & security' tab is also visible in the navigation bar. A notification banner for 'Database Insights' is present, and the 'CloudWatch' section shows a search bar and a 'Monitoring' dropdown.

RDS database using MySQL

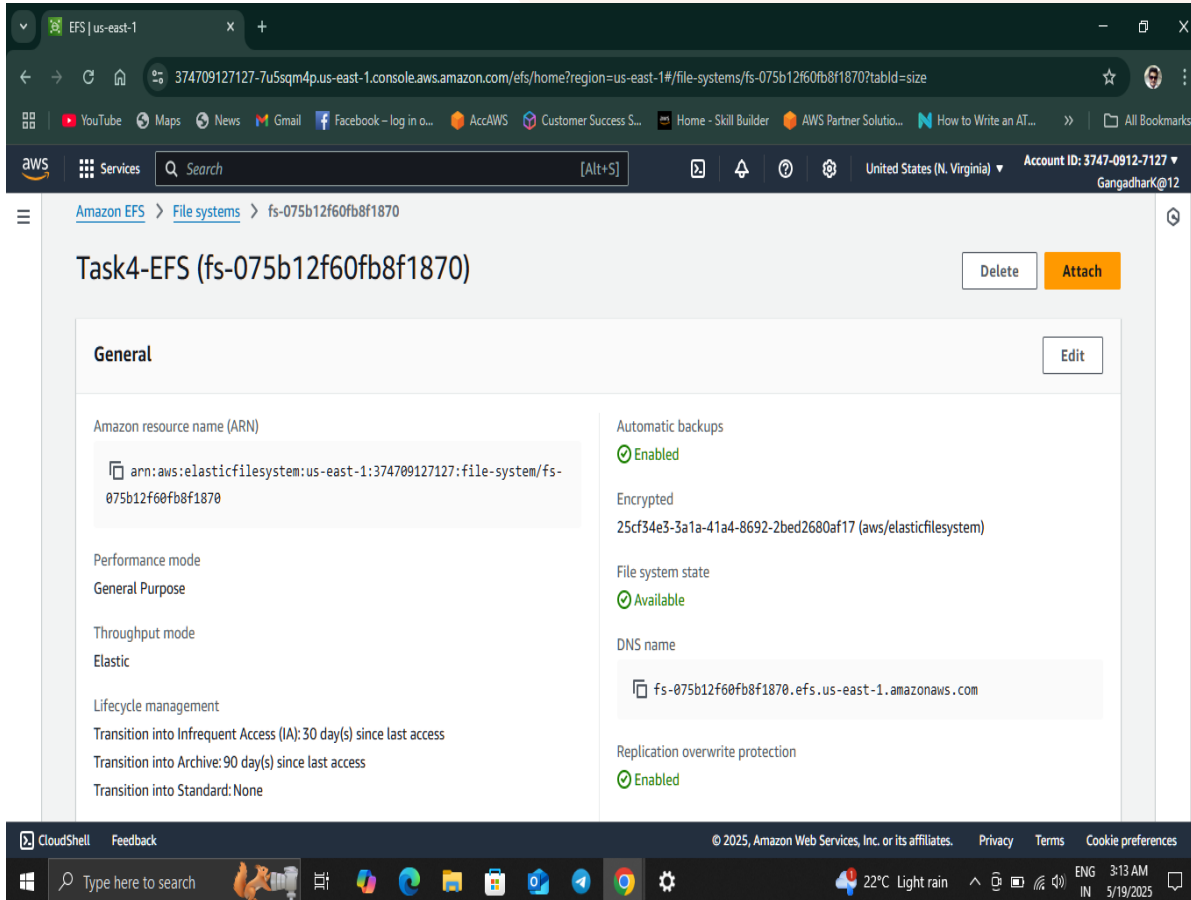


The screenshot shows the 'Connectivity & security' tab for the same database instance. It is divided into three columns:

- Endpoint & port:** Endpoint is 'task4-database-1.cmje4e0e6wzy.us-east-1.rds.amazonaws.com' and Port is '3306'.
- Networking:** Availability Zone is 'us-east-1b', VPC is 'Task4VPC (vpc-0c8b90278639d2c20)', Subnet group is 'task4dbinstancesubnetgroup', and Subnets are 'subnet-0ceb128cc8bd4bc00' and 'subnet-01b28d53f0c3b3dcf'. Network type is 'IPv4'.
- Security:** VPC security groups include 'SG-DBInstance (sg-07cccccb11df7c711)' which is 'Active'. It is 'Publicly accessible: No'. Certificate authority is 'rds-ca-rsa2048-g1'. Certificate authority date is 'May 26, 2061, 05:04 (UTC+05:30)'. DB instance certificate expiration date is 'May 18, 2026, 12:29 (UTC+05:30)'.

RDS connected VPC

Step 9 - After creating Amazon EFS system, create mount targets, mount the file system on your web amazon EC2 instances in each AZ in your VPC.

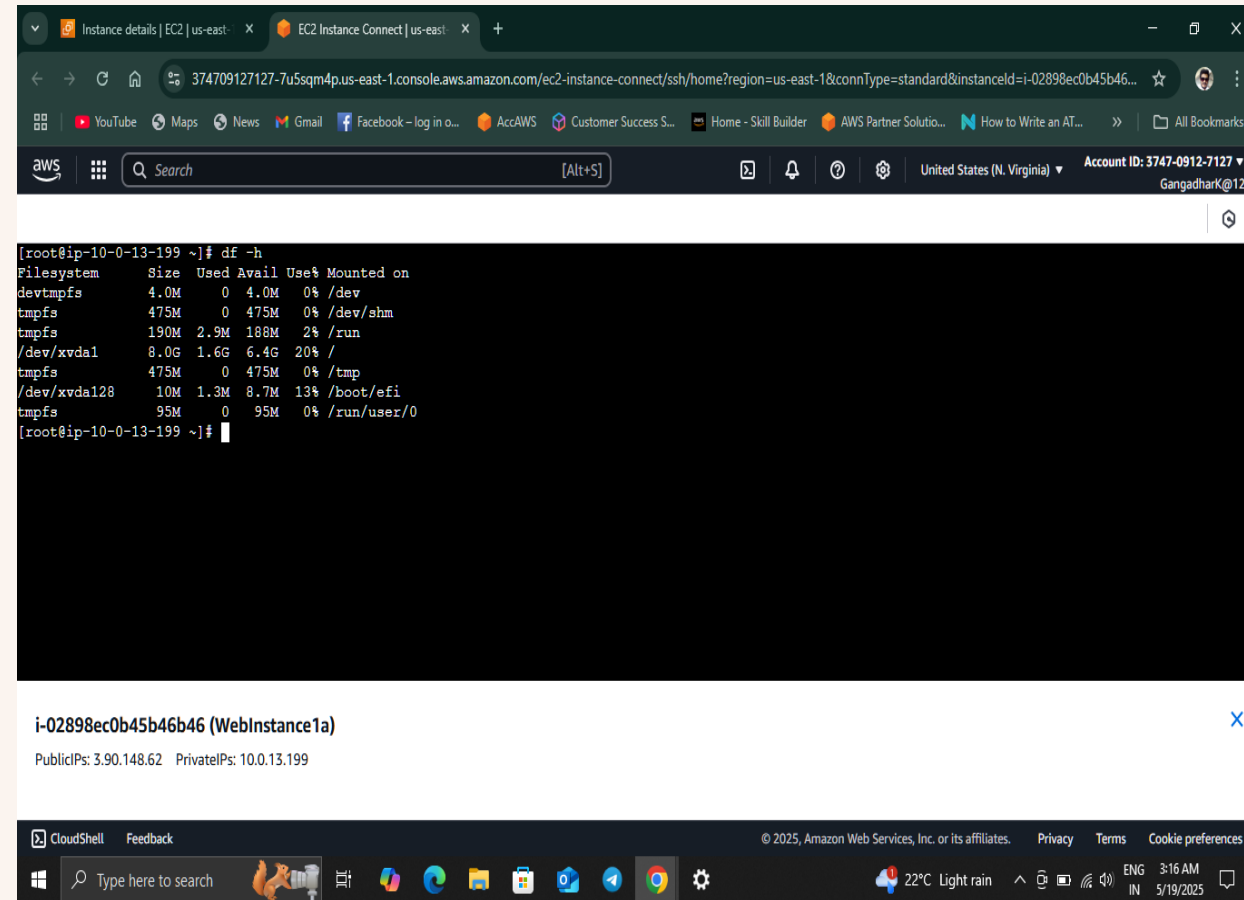


The screenshot shows the Amazon EFS console interface. The breadcrumb navigation is 'Amazon EFS > File systems > fs-075b12f60fb8f1870'. The main heading is 'Task4-EFS (fs-075b12f60fb8f1870)' with 'Delete' and 'Attach' buttons. The 'General' tab is selected, showing the following details:

- Amazon resource name (ARN): `arn:aws:elasticfilesystem:us-east-1:374709127127:file-system/fs-075b12f60fb8f1870`
- Performance mode: General Purpose
- Throughput mode: Elastic
- Lifecycle management: Transition into Infrequent Access (IA): 30 day(s) since last access; Transition into Archive: 90 day(s) since last access; Transition into Standard: None
- Automatic backups: Enabled
- Encrypted: 25cf34e3-3a1a-41a4-8692-2bed2680af17 (aws/elasticfilesystem)
- File system state: Available
- DNS name: `fs-075b12f60fb8f1870.efs.us-east-1.amazonaws.com`
- Replication overwrite protection: Enabled

The bottom of the screen shows the AWS CloudShell interface with a Windows taskbar at the bottom.

EFS created



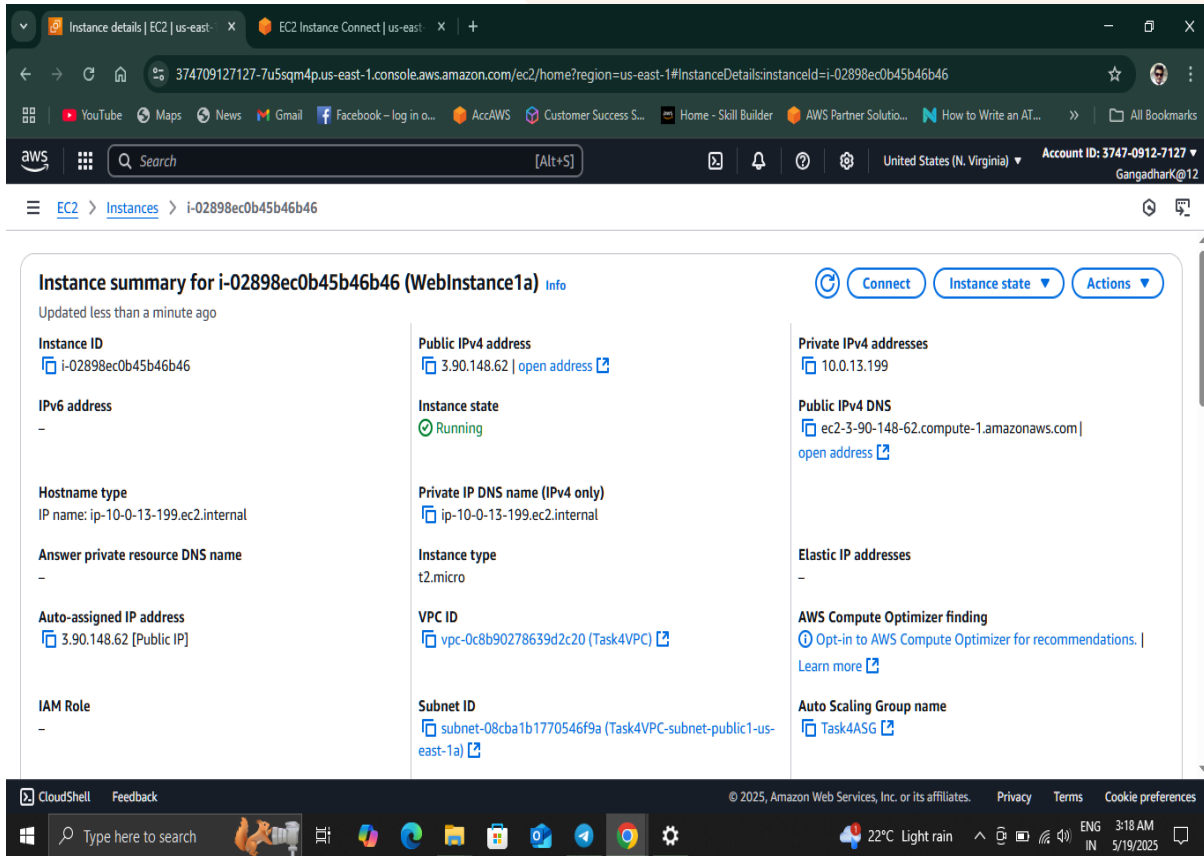
The screenshot shows the Amazon EC2 console interface. The breadcrumb navigation is 'Instance details | EC2 | us-east-1 > EC2 Instance Connect | us-east-1'. The main heading is 'i-02898ec0b45b46b46 (WebInstance1a)' with 'PublicIPs: 3.90.148.62' and 'PrivateIPs: 10.0.13.199'. A terminal window is open, displaying the output of the `df -h` command:

```
[root@ip-10-0-13-199 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M  0   4.0M  0% /dev
tmpfs           475M  0   475M  0% /dev/shm
tmpfs           190M  2.9M  188M  2% /run
/dev/xvda1       8.0G  1.6G   6.4G 20% /
tmpfs           475M  0   475M  0% /tmp
/dev/xvda128     10M  1.3M   8.7M 13% /boot/efi
tmpfs           95M  0    95M  0% /run/user/0
```

The bottom of the screen shows the AWS CloudShell interface with a Windows taskbar at the bottom.

EFS mounted (df -h)

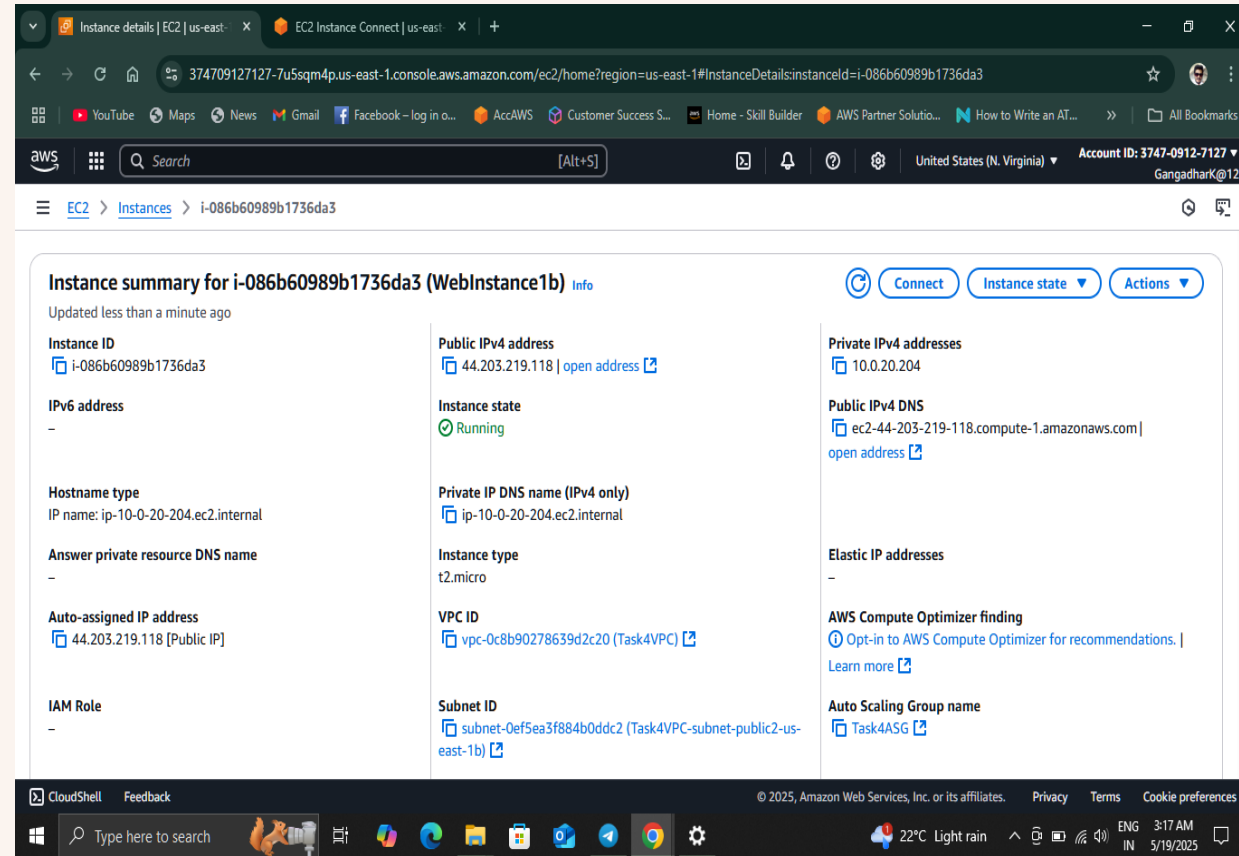
Step 10 - Use Amazon EFS Drupal instances to access your shared, unstructured Nginx data such as XML, Config, CSS & Plugins etc.



The screenshot displays the AWS Management Console for an EC2 instance named 'WebInstance1a' with ID 'i-02898ec0b45b46b46'. The instance is in a 'Running' state. Key details include:

- Instance ID:** i-02898ec0b45b46b46
- Public IPv4 address:** 3.90.148.62
- Private IPv4 addresses:** 10.0.13.199
- Public IPv4 DNS:** ec2-3-90-148-62.compute-1.amazonaws.com
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-10-0-13-199.ec2.internal
- Instance type:** t2.micro
- VPC ID:** vpc-0c8b90278639d2c20 (Task4VPC)
- Subnet ID:** subnet-08cba1b1770546f9a (Task4VPC-subnet-public1-us-east-1a)
- IAM Role:** -
- Auto Scaling Group name:** Task4ASG

WebInstance 1a

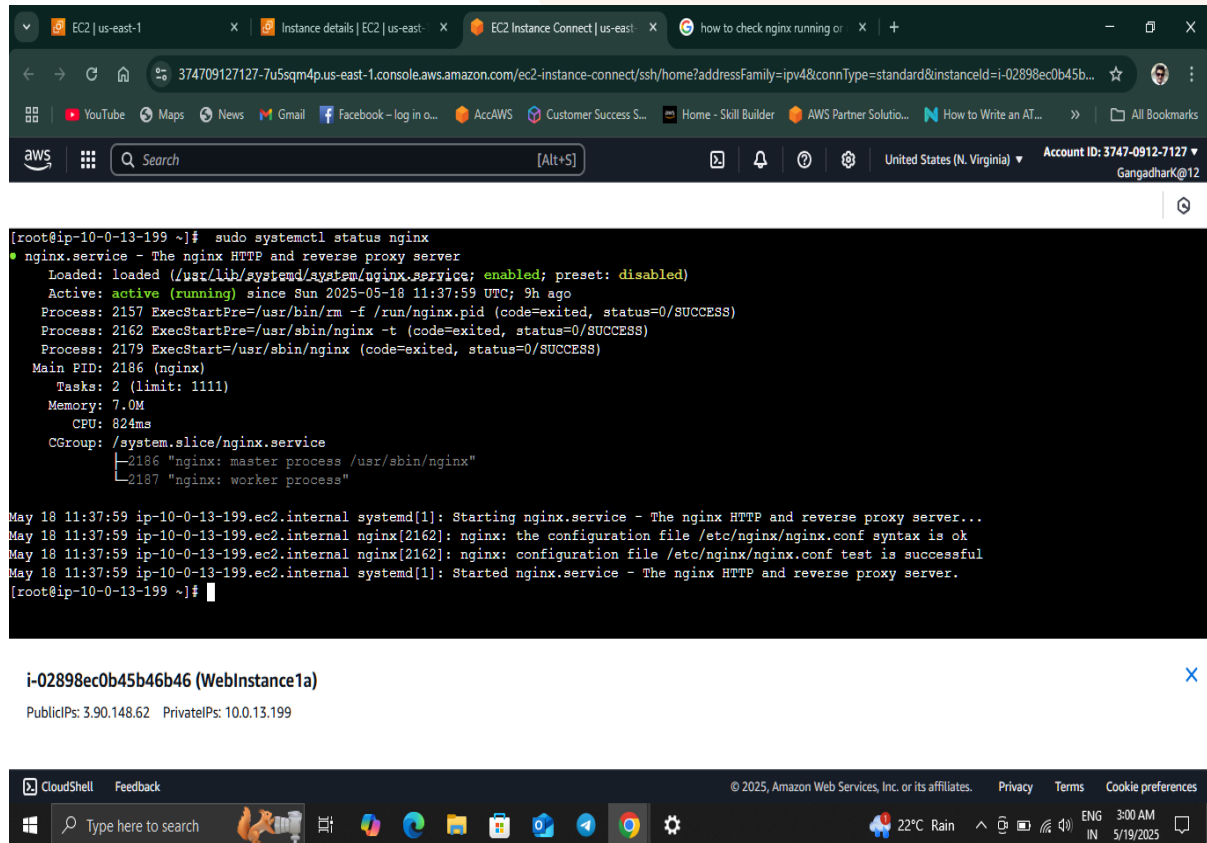


The screenshot displays the AWS Management Console for an EC2 instance named 'WebInstance1b' with ID 'i-086b60989b1736da3'. The instance is in a 'Running' state. Key details include:

- Instance ID:** i-086b60989b1736da3
- Public IPv4 address:** 44.203.219.118
- Private IPv4 addresses:** 10.0.20.204
- Public IPv4 DNS:** ec2-44-203-219-118.compute-1.amazonaws.com
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-10-0-20-204.ec2.internal
- Instance type:** t2.micro
- VPC ID:** vpc-0c8b90278639d2c20 (Task4VPC)
- Subnet ID:** subnet-0ef5ea3f884b0ddc2 (Task4VPC-subnet-public2-us-east-1b)
- IAM Role:** -
- Auto Scaling Group name:** Task4ASG

WebInstance 1b

Step 10 - Use Amazon EFS Drupal instances to access your shared, unstructured Nginx data such as XML, Config, CSS & Plugins etc.



The screenshot shows the AWS Management Console with a terminal window open for an EC2 instance. The terminal output displays the status of the nginx service, which is active and running. Below the terminal, the instance details for 'i-02898ec0b45b46b46 (WebInstance1a)' are visible, including public and private IP addresses. The bottom of the image shows a Windows taskbar with various application icons and system information.

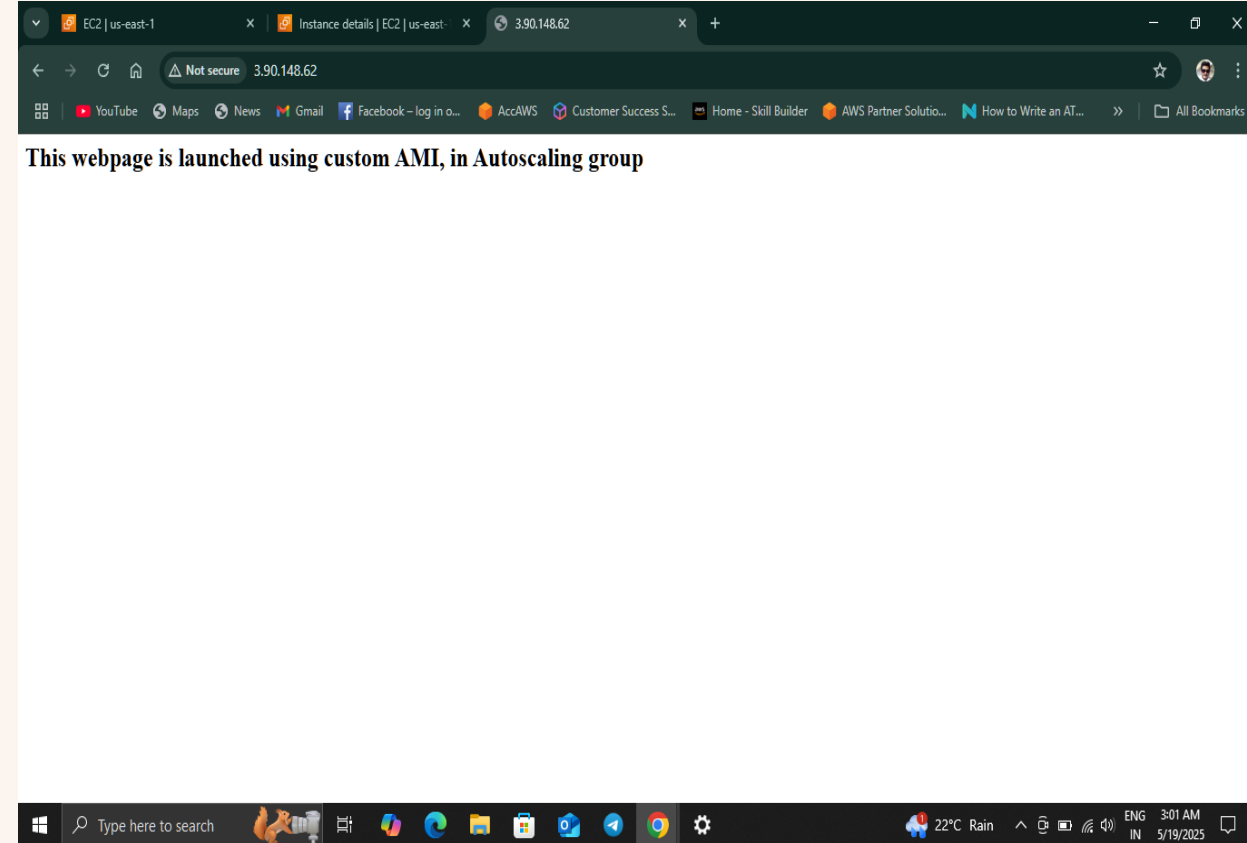
```
[root@ip-10-0-13-199 ~]# sudo systemctl status nginx
nginx.service - The nginx HTTP and reverse proxy server
Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
Active: active (running) since Sun 2025-05-18 11:37:59 UTC; 9h ago
Process: 2157 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
Process: 2162 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
Process: 2179 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
Main PID: 2186 (nginx)
Tasks: 2 (limit: 1111)
Memory: 7.0M
CPU: 824ms
CGroup: /system.slice/nginx.service
└─2186 "nginx: master process /usr/sbin/nginx"
    └─2187 "nginx: worker process"

May 18 11:37:59 ip-10-0-13-199.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
May 18 11:37:59 ip-10-0-13-199.ec2.internal nginx[2162]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
May 18 11:37:59 ip-10-0-13-199.ec2.internal nginx[2162]: nginx: configuration file /etc/nginx/nginx.conf test is successful
May 18 11:37:59 ip-10-0-13-199.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[root@ip-10-0-13-199 ~]#
```

i-02898ec0b45b46b46 (WebInstance1a)

PublicIPs: 3.90.148.62 PrivateIPs: 10.0.13.199

Nginx is Running



Public IPA Accessed

Abstract geometric lines in the top left corner, consisting of several overlapping, irregular polygons and lines in a light beige color.

Regards

Gangadhar K