

AWS Cloud Infrastructure Deployment and Automation

1. Create a VPC

What:

A Virtual Private Cloud (VPC) is a logically isolated section of AWS where you launch AWS resources.

How:

- In AWS Console, go to **VPC**, click **Create VPC**.
- Choose default or custom settings (CIDR block, subnets, etc).

Why:

This isolates your application's network for security and control.

The screenshot shows the AWS VPC console interface. On the left, there's a navigation sidebar with options like EC2 Global View, Filter by VPC, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), Security (Network ACLs, Security groups), and CloudShell/Feedback. The main area shows 'Your VPCs (1/1)' with a table header: Name, VPC ID, State, Block Public..., IPv4 CIDR, IPv6 CIDR. A single row is selected: 'my Project' (vpc-043f5889e32ef183f). The 'Actions' button is visible. Below the table, the details for 'vpc-043f5889e32ef183f / my Project' are shown in a card format with tabs for Details, Resource map, CIDRs, Flow logs, Tags, and Integrations. The 'Details' tab is active, showing fields like VPC ID, State (Available), Block Public Access (Off), DNS resolution (Enabled), Main network ACL (acl-0b5c48d01630fb0fa), Default VPC (No), IPv6 CIDR (Network border group), Network Address Usage metrics (Disabled), and Route 53 Resolver DNS Firewall rule groups. Other tabs include Resource map, CIDRs, Flow logs, Tags, and Integrations.

2. Create S3 and Upload File

What:

S3 is Amazon's object storage.

How:

- In AWS Console, go to **S3, Create Bucket**.
- Upload files (e.g., static assets or media for your app).

Why:

S3 is used for storing files, static data, backups, etc.

AWS Cloud Infrastructure Deployment and Automation

The screenshot shows the AWS S3 console interface. On the left, a sidebar menu includes options like General purpose buckets, Directory buckets, Table buckets, and Storage Lens. The main area displays a table for 'General purpose buckets (1)'. The table has columns for Name, AWS Region, and Creation date. One row is shown for 'projecrbucket' in 'US East (N. Virginia) us-east-1' created on 'July 25, 2025, 21:28:48 (UTC+05:30)'. To the right of the table are two boxes: 'Account snapshot' and 'External access summary - new'. The status bar at the bottom shows the date as 7/26/2025.

3. Create EFS

What:

EFS (Elastic File System) is scalable file storage for use with EC2.

How:

- Go to EFS, **Create file system**.
- Set up mount targets, security groups.

Why:

EFS can be mounted on multiple EC2s at once, useful for sharing files or app data.

The screenshot shows the AWS EFS console interface. The left sidebar lists options like File systems, Access points, AWS Backup, AWS DataSync, and AWS Transfer. The main area displays the details for a file system named 'My-Project-EFS (fs-0b7440b8f6a51c994)'. The 'General' tab is selected, showing fields such as ARN (arn:aws:elasticfilesystem:us-east-1:566801648969:file-system/fs-0b7440b8f6a51c994), Performance mode (General Purpose), 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), Availability zone (Regional), and various status indicators like Enabled, Available, and Enabled for automatic backups and replication. The status bar at the bottom shows the date as 7/24/2025.

AWS Cloud Infrastructure Deployment and Automation

4. Launch EC2, Install Nginx, Configure index.html

What:

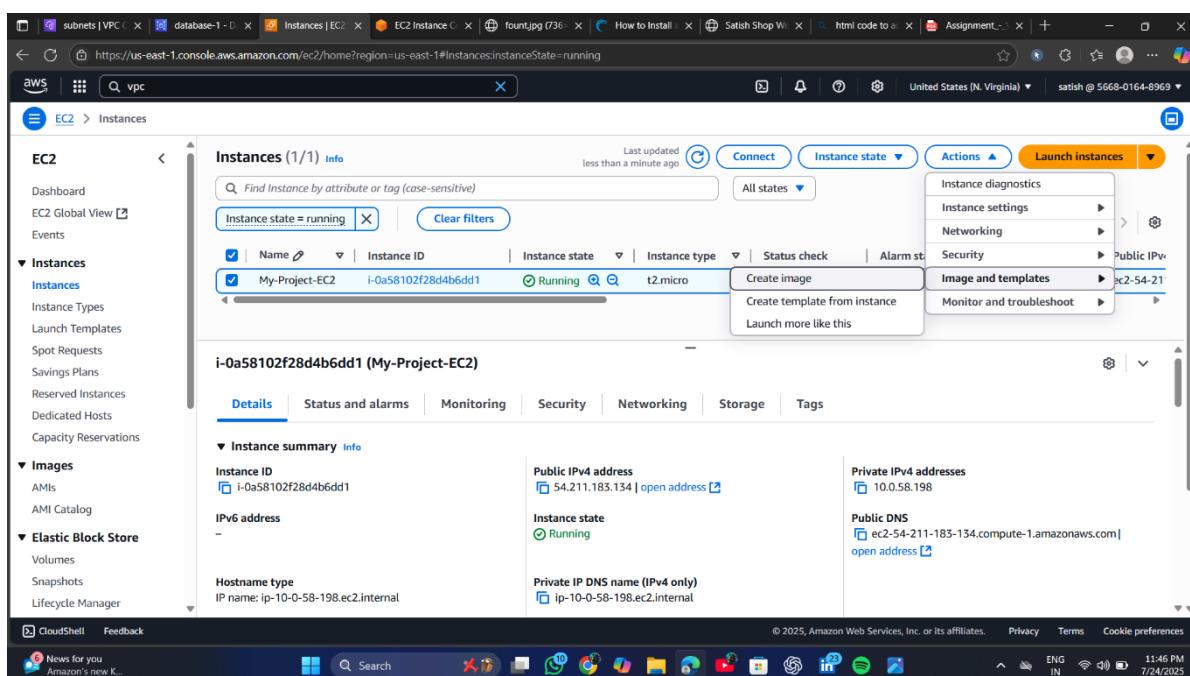
EC2 are the virtual servers. Nginx is a web server.

How:

- Launch an EC2 instance (Amazon Linux/Ubuntu).
- SSH into instance.
- Install nginx: `#bash sudo apt update sudo apt upgrade sudo apt install nginx sudo systemctl status nginx`
- Create and edit `/var/www/html/index.html`.

Why:

This is your web server that will host your app site.



5. Login to Your Ubuntu Instance

What:

SSH into your EC2 machine to perform administrative tasks.

How: `bash ssh -i your-key.pem`

`ubuntu@<public-ip>`

Use your PEM key and the public IP from the AWS console.

6. Update & Upgrade OS (on EC2)

What:

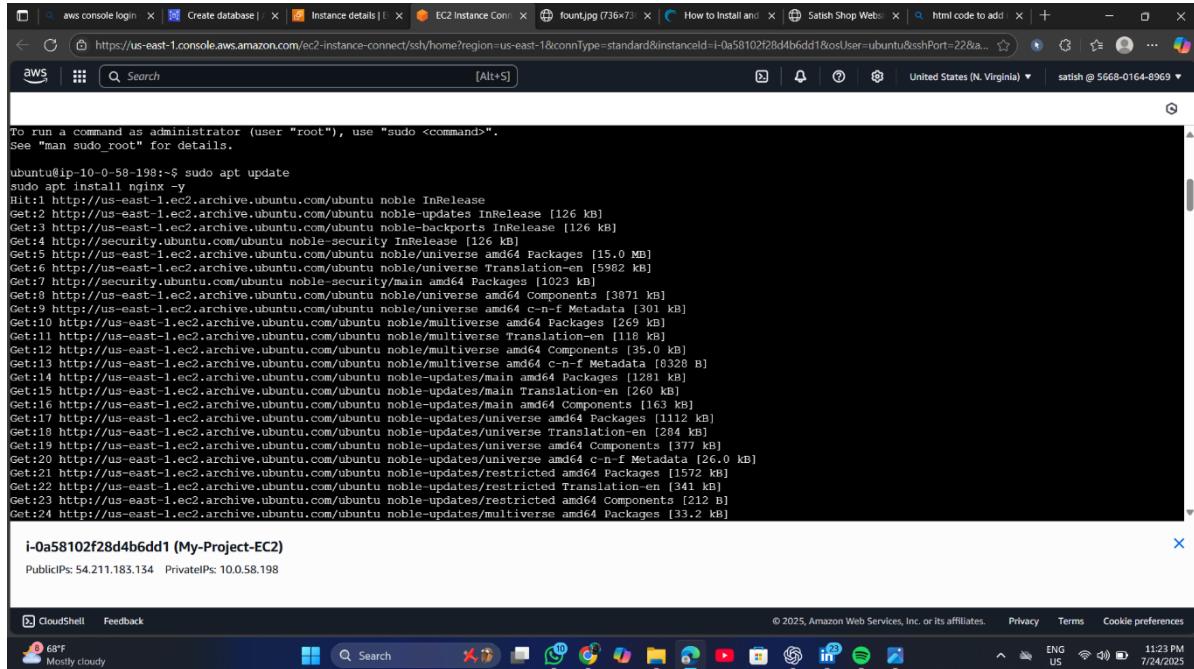
AWS Cloud Infrastructure Deployment and Automation

Keeps your system secure and up to date.

How: bash sudo

apt update sudo

apt upgrade



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-58-198:~$ sudo apt update
sudo apt install nginx -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1023 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1281 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [260 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [163 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1112 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [384 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [26.0 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [1572 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [341 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [33.2 kB]

i-0a58102f28d4b6dd1 (My-Project-EC2)
PublicIPs: 54.211.183.134 PrivateIPs: 10.0.58.198
```

7. Install and Check Nginx

How: #bash

sudo apt install nginx sudo systemctl status

nginx This installs and checks if nginx is running. sudo unlink /etc/nginx/sites-enabled/default sudo apt update

sudo systemctl status nginx

sudo mkdir -p /var/www/satish.shop/html sudo ln -s /etc/nginx/sites-available/satish.shop /etc/nginx/sites-enabled/

AWS Cloud Infrastructure Deployment and Automation

```
No services need to be restarted.  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-10-0-58-198:~$ sudo systemctl status nginx  
● nginx.service - A high performance web server and a reverse proxy server  
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)  
  Active: active (running) since Thu 2025-07-24 17:26:44 UTC; 17s ago  
    Docs: man:nginx(8)  
    Process: 2047 ExecStartPre=/usr/sbin/nginx -t -g daemon on; master_process on; (code=exited, status=0/SUCCESS)  
   Process: 2049 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)  
 Main PID: 2078 (nginx)  
    Tasks: 2 (limit: 1124)  
   Memory: 1.7M (peak: 3.7M)  
     CPU: 16ms  
    CGroup: /system.slice/nginx.service  
           └─2078 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"  
           ├─2080 "nginx: worker process"  
  
Jul 24 17:26:44 ip-10-0-58-198 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...  
Jul 24 17:26:44 ip-10-0-58-198 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.  
ubuntu@ip-10-0-58-198:~$ sudo systemctl start nginx  
ubuntu@ip-10-0-58-198:~$ sudo systemctl enable nginx  
Synchronizing state of nginx.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx  
  
i-0a58102f28d4b6dd1 (My-Project-EC2)  
PublicIPs: 54.211.183.134 PrivateIPs: 10.0.58.198
```

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 11:23 PM 7/24/2025

```
ubuntu@ip-10-0-58-198:~$ sudo ufw allow http  
sudo ufw reload  
Rules updated (v6)  
Firewall not enabled (skipping reload)  
ubuntu@ip-10-0-58-198:~$ sudo unlink /etc/nginx/sites-enabled/default  
ubuntu@ip-10-0-58-198:~$ sudo nano /etc/nginx/sites-available/satish.shop  
ubuntu@ip-10-0-58-198:~$ sudo In -s /etc/nginx/sites-available/satish.shop /etc/nginx/sites-enabled/  
nginx: configuration file /etc/nginx/nginx.conf test is successful  
ubuntu@ip-10-0-58-198:~$ sudo systemctl restart nginx  
ubuntu@ip-10-0-58-198:~$ sudo mkdir -p /var/www/satish.shop  
sudo nano /var/www/satish.shop/index.html  
ubuntu@ip-10-0-58-198:~$ sudo apt-get -y install nfs-common  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  keyutils libnfsidmap1 rpcbind  
Suggested packages:  
  watchdog  
The following NEW packages will be installed:  
  keyutils libnfsidmap1 nfs-common rpcbind  
0 upgraded, 4 newly installed, 0 to remove and 89 not upgraded.  
Need to get 401 kB of archives.  
After this operation, 1416 kB of additional disk space will be used.  
Gett http://us-east-1.ec2.archive.ubuntu.com/ubuntu nobile-updates/main amd64 libnfsidmap1 amd64 1:2.6.4-3ubuntu5.1 [48.3 kB]  
  
i-0a58102f28d4b6dd1 (My-Project-EC2)  
PublicIPs: 54.211.183.134 PrivateIPs: 10.0.58.198
```

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 11:24 PM 7/24/2025

8. Continue Nginx Setup (via Blog)

Follow the instructions from the linked Linode blog (from the "Use NGINX #" step onwards) for further nginx configuration.

https://www.linode.com/docs/guides/how-to-install-and-use-nginx-on-ubuntu-20-04/

Docs Blog Support Sales Log in English

Akamai Cloud Explore Products Solutions Pricing Resources Partners Sign up

EXPLORE DOCS Search Linode

Installing And Using NGINX On AlmaLinux 8
Installing And Using NGINX On Ubuntu 20.04
Installing NGINX On CentOS 8
Installing NGINX On Debian 10
Installing NGINX On Ubuntu 18.04
Pros And Cons Of Configuring A Load Balancer For Sticky Sessions
Serve PHP With PHP-FPM And NGINX
The Ultimate Web Servers List: 11 Popular Web Server To Use Today
Use NGINX As A Reverse Proxy
See All 68 NGINX Guides
Squid Proxy/Cache

Docs Home > Guides > Web Server Guides > NGINX

Installing and Using NGINX on Ubuntu 20.04

Published July 9, 2021 by Nathaniel Stickman

This guide was written for Ubuntu 20.04. Other distributions are available: Ubuntu 20.04

Create a Linode account to try this guide with a \$100 credit. This credit will be applied to any valid services used during your first 60 days.

ON THIS PAGE

Before You Begin
Install NGINX
Manage NGINX
Use NGINX

AWS Cloud Infrastructure Deployment and Automation

bash

```
sudo apt-get -y install nfs-common
```

2. **Mount EFS volume** (replace fs-xxxx with your EFS file system id):

bash

3. Create or move your index.html as needed.

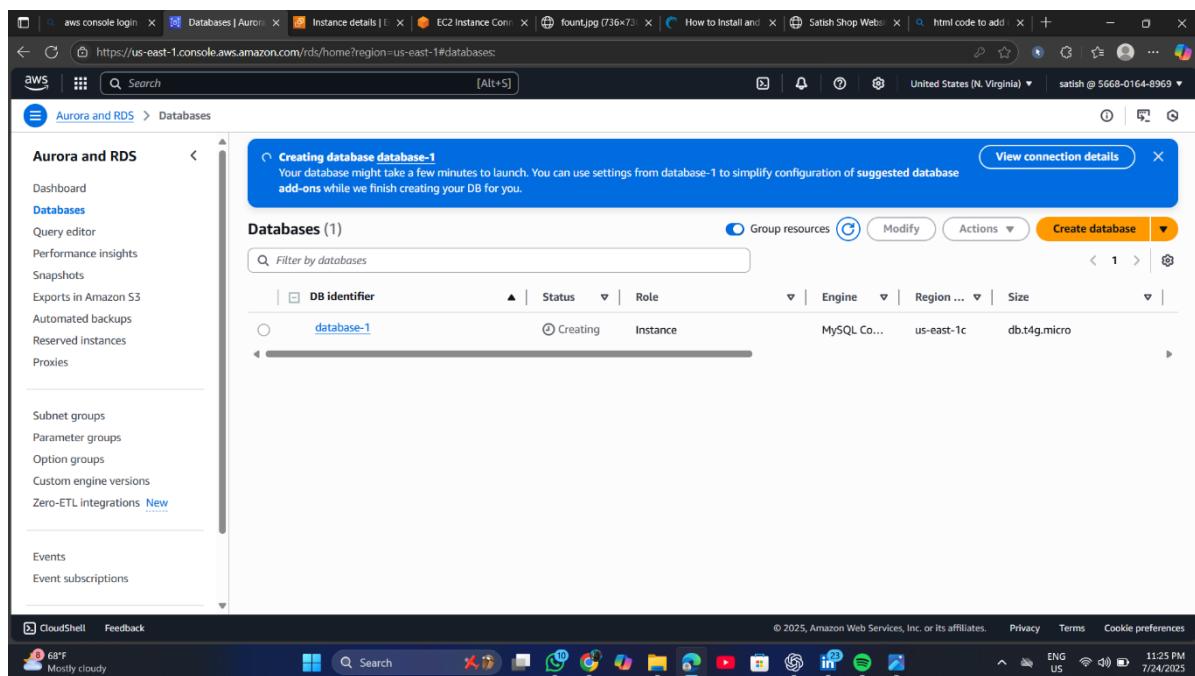
10. Create RDS Database

What:

RDS is AWS's managed relational database.

How:

- Go to **RDS > Create database**
- Choose DB engine (MySQL, Postgres, etc.)
- Set username, password, instance type.
- Create data base



AWS Cloud Infrastructure Deployment and Automation

11. Create a Read Replica (Optional)

What/Why:

Improves scalability by offloading reads from the primary database.

How:

- In your RDS instance, select **Create read replica**.

The screenshot shows the 'Create read replica' wizard in the AWS RDS console. In the 'Settings' step, the 'Replica source' is set to 'Source DB instance identifier' and 'database-1' is selected. The 'DB instance identifier' is set to 'myproject'. The 'Instance configuration' section shows options for 'DB instance class' (Info), including 'Standard classes', 'Memory optimized classes', and 'Burstable classes', with 'Burstable classes' selected. The bottom of the screen shows the Windows taskbar with various pinned icons.

The screenshot shows the 'Create read replica' wizard in the AWS RDS console. It has moved to the 'Additional storage configuration' step. The 'Availability' step is shown below it, featuring three deployment options:

- Multi-AZ DB cluster deployment (3 instances)**: Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:
 - 99.95% uptime
 - Redundancy across Availability Zones
 - Increased read capacity
 - Reduced write latencyA diagram shows a primary instance in AZ 1 connected to two readable standbys in AZ 2 and AZ 3, each with an SSD.
- Multi-AZ DB instance deployment (2 instances)**: Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides:
 - 99.95% uptime
 - Redundancy across Availability ZonesA diagram shows a primary instance in AZ 1 connected to a standby instance in AZ 2.
- Single-AZ DB instance deployment (1 instance)**: Creates a single writer DB instance with no reader instances. This setup provides:
 - 99.5% uptime
 - No data redundancyA diagram shows a single primary instance in AZ 1.

The bottom of the screen shows the Windows taskbar with various pinned icons.

AWS Cloud Infrastructure Deployment and Automation

12. Connect EC2 to RDS How:

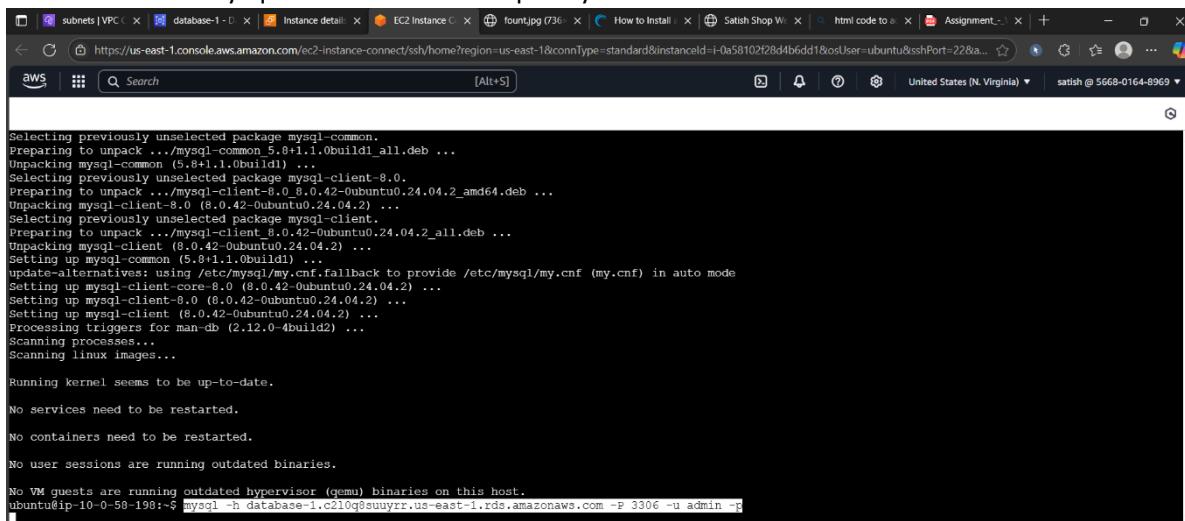
1. Install MySQL client:

```
bash sudo apt-get install mysql-client
```

2. Connect to the DB:

```
bash
```

```
mysql -h database-1.c210q8suuyrr.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
```



```
Selecting previously unselected package mysql-common.
Preparing to unpack .../mysql-common_5.8+1.1.0build1_all.deb ...
Unpacking mysql-common (5.8+1.1.0build1) ...
Selecting previously unselected package mysql-client-8.0.
Preparing to unpack .../mysql-client-8.0_8.0.42-0ubuntu0.24.04.2_amd64.deb ...
Unpacking mysql-client-8.0 (8.0.42-0ubuntu0.24.04.2) ...
Selecting previously unselected package mysql-client.
Preparing to unpack .../mysql-client_8.0.42-0ubuntu0.24.04.2_all.deb ...
Unpacking mysql-client (8.0.42-0ubuntu0.24.04.2) ...
Setting up mysql-common (5.8+1.1.0build1) ...
update-alternatives: using /etc/mysql/my.cnf.fallback to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Setting up mysql-client-core-8.0 (8.0.42-0ubuntu0.24.04.2) ...
Setting up mysql-client-8.0 (8.0.42-0ubuntu0.24.04.2) ...
Setting up mysql-client (8.0.42-0ubuntu0.24.04.2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

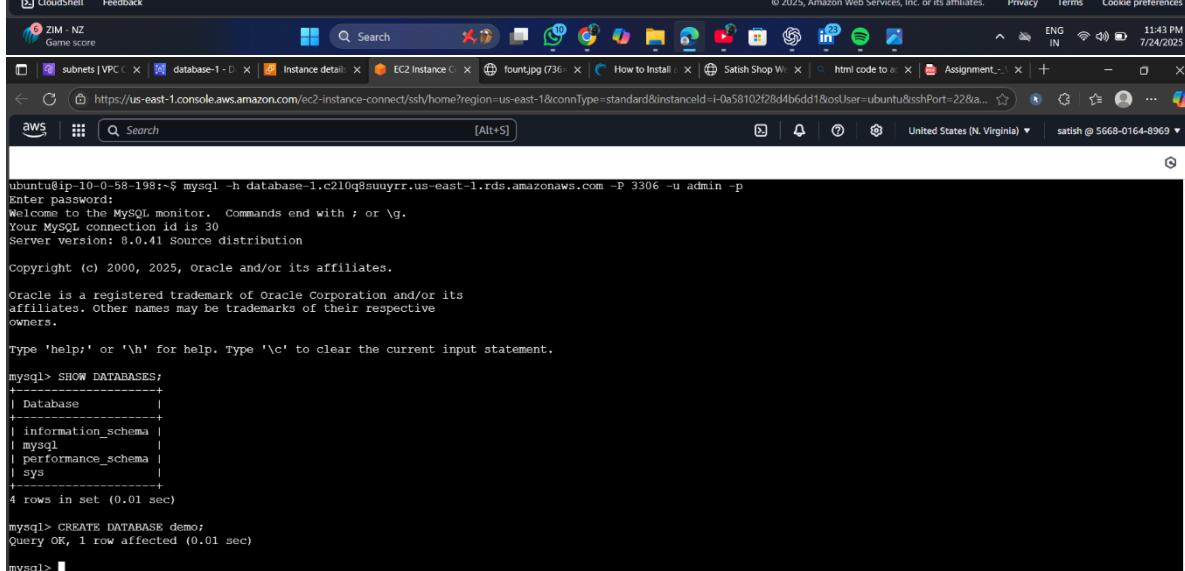
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-10-0-58-198:~$ mysql -h database-1.c210q8suuyrr.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
```

i-0a58102f28d4b6dd1 (My-Project-EC2)

PublicIPs: 54.211.183.134 PrivateIPs: 10.0.58.198

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



```
ubuntu@ip-10-0-58-198:~$ mysql -h database-1.c210q8suuyrr.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 8.0.41 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+--------------------+
| Database          |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.01 sec)

mysql> CREATE DATABASE demo;
Query OK, 1 row affected (0.01 sec)

mysql>
```

i-0a58102f28d4b6dd1 (My-Project-EC2)

PublicIPs: 54.211.183.134 PrivateIPs: 10.0.58.198

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

AWS Cloud Infrastructure Deployment and Automation

13. Create AMI

What:

AMI (Amazon Machine Image): A snapshot/template of your EC2 instance.

Why:

Allows quick scaling or disaster recovery.

How:

- Select EC2 instance, -> Actions -> Image and Templates -> Create Image.

The screenshot shows the AWS Cloud Console interface for managing Amazon Machine Images (AMIs). The left sidebar navigation includes 'EC2', 'Dashboard', 'EC2 Global View', 'Events', 'Instances' (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), 'Images' (with sub-options like AMIs, AMI Catalog), and 'Elastic Block Store' (with sub-options like Volumes, Snapshots, Lifecycle Manager). The main content area is titled 'Amazon Machine Images (AMIs) (1/1)' and shows a table with one row for 'Projectimage'. The table columns include 'Name', 'AMI ID', 'Source', 'Owner', and 'Visibility'. Below the table, a detailed view for the selected AMI ('AMI ID: ami-07a88fc270d6227c7') is displayed, showing fields such as AMI ID, AMI name, Owner account ID, Root device name, Status, Platform details, Architecture, Source, and Virtualization type. The status is listed as 'Available'.

14. Configure ALB (Application Load Balancer) and ASG (Auto Scaling Group)

What:

- ALB:** Distributes incoming traffic to multiple EC2 instances.
- ASG:** Automatically adjusts the number of EC2 instances based on demand.

How:

1. Create ALB:

- Go to **EC2 > Load Balancers**.
- Click **Create Load Balancer > Application Load Balancer**.
- Set up listeners (usually HTTP & HTTPS), choose subnets, and security groups.
- Register your EC2 instances (or the target group that will auto-register them via ASG).

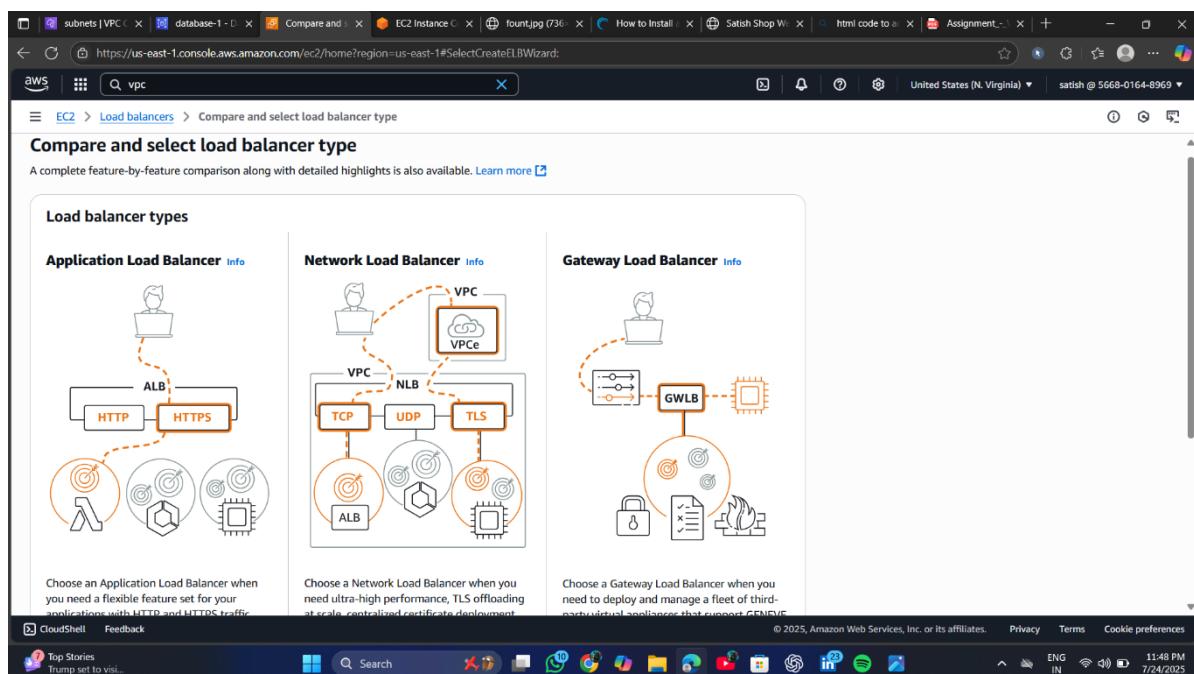
2. Create Launch Template/Configuration:

AWS Cloud Infrastructure Deployment and Automation

- Go to EC2 > Launch Templates and click **Create launch template**.
- Reference your previously created AMI.

3. Create Auto Scaling Group:

- In EC2 > Auto Scaling Groups, click **Create Auto Scaling group**.
- Pick your launch template (from the previous step).
- Select VPC, subnets, and attach your Target Group (from the ALB).
- Define min, max, and desired number of EC2 instances.
- Set scaling policies (optional, for automatic scaling based on CPU/memory, etc.).



AWS Cloud Infrastructure Deployment and Automation

The screenshot displays two consecutive steps in the AWS Cloud Infrastructure Deployment and Automation process:

Step 1: EC2 Launch Template Details

The user is viewing the "my-project-template (lt-0210cedb034578a40)" launch template details. Key information shown includes:

- Launch template ID:** lt-0210cedb034578a40
- Launch template name:** my-project-template
- Default version:** 1
- Owner:** arn:aws:iam::566801648969:user/satish

Step 2: Create Auto Scaling Group

The user is in the "Create Auto Scaling group" wizard, currently on Step 4 - optional: **Configure group size and scaling**. The configuration includes:

- Group size:** Set to 2 instances.
- Desired capacity:** Set to 2.
- Scaling limits:** Min desired capacity is 2, and Max desired capacity is 4.
- Automatic scaling - optional:** The "No scaling policies" option is selected, indicating the Auto Scaling group will remain at its initial size.

Both screenshots show the AWS navigation bar, search bar, and status bar at the bottom.

AWS Cloud Infrastructure Deployment and Automation

The screenshot shows the AWS CloudFront console with the URL <https://us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=us-east-1#/distributions>. The page displays a success message: "Successfully created new distribution." Below this, the "myproject-Cloud" distribution is listed under the "General" tab. The distribution details include:

- Name:** myproject-Cloud
- Distribution domain name:** d31ag68u2ujyn.cloudfront.net
- ARN:** arn:aws:cloudfront::566801648969:distribution/EXJ8OOEZ4FX6K
- Last modified:** Deploying

The "Settings" tab is also visible, showing options for Description, Price class (Use all edge locations (best performance)), Supported HTTP versions (HTTP/2, HTTP/1.1, HTTP/1.0), Alternate domain names, Standard logging (Off), Cookie logging (Off), and Default root object.

15. Create and Configure CloudFront with ALB

What:

CloudFront is AWS's global Content Delivery Network (CDN); it caches and distributes content closer to users worldwide.

How:

1. Go to **CloudFront > Distributions > Create Distribution.**
2. For origin, enter your ALB's DNS name (found in EC2 > Load Balancers).
3. Configure whether to use HTTP or HTTPS (set up certificates as needed).
4. Customize cache behavior, price class, and any desired settings.
5. Click **Create Distribution**. CloudFront assigns a Distribution Domain Name (e.g., d1xyz.cloudfront.net).

AWS Cloud Infrastructure Deployment and Automation

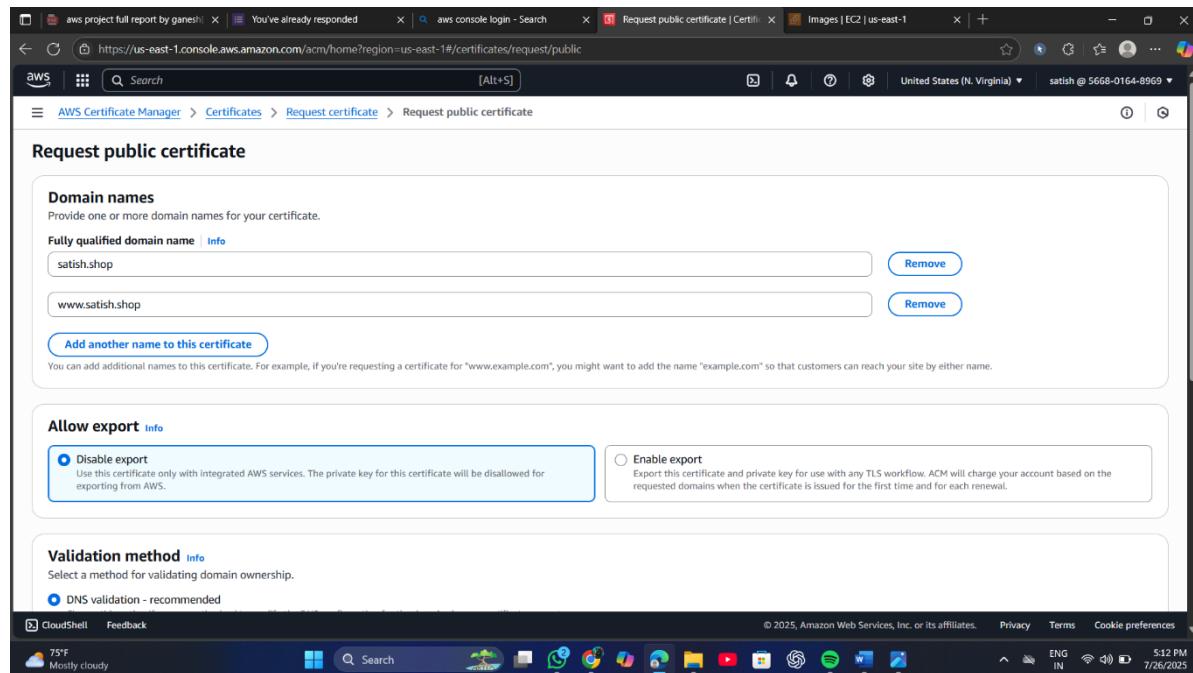
16. Create CNAME, Update Route 53 Records from ACM

What:

- **ACM (AWS Certificate Manager)**: Issues SSL/TLS certificates for HTTPS.
- **Route 53**: AWS DNS service.

How:

1. In **ACM** (Certificate Manager), request a new certificate for your domain.
2. ACM will provide a CNAME record for DNS validation.
3. Go to **Route 53 > Hosted zones**, and for your domain, **create a new CNAME record** with the name and value provided by ACM (for validation).
4. Wait for certificate validation (ACM status will update to "Issued").
5. Later, in **Route 53**, add/modify DNS records to point your domain/subdomain to your ALB or CloudFront:
 - **A/AAAA Alias** record for root/apex domains.
 - **CNAME** for subdomains.



AWS Cloud Infrastructure Deployment and Automation

The screenshot shows the AWS Certificate Manager (ACM) interface. On the left, there's a sidebar with options like 'List certificates', 'Request certificate', 'Import certificate', and 'AWS Private CA'. The main area is titled 'Domains (2)' and lists two entries:

Renewal status	Type	CNAME name	CNAME value
In use	CNAME	_b8837b0b3994004d4d2712c1947e1312.satish.shop.	_8276e08c0e67ce03ba965008851fd0fd.xlfgrmvvlj.acm-validations.aws.
	CNAME	_406deb1938eeaa34e46885bd87d8c9c9.www.satish.shop.	_506d99bfddc4aa0b15899b7889ecfb0c.xlfgrmvvlj.acm-validations.aws.

Below this, there's a 'Details' section with the following information:

In use	Serial number	Requested at	Renewal eligibility
No	N/A	July 26, 2025, 17:12:35 (UTC+05:30)	Ineligible
Domain name	Public key info	Issued at	Export option
satish.shop	RSA 2048	N/A	Disabled
Number of additional names	Signature algorithm	Not before	Not after
1	SHA-256 with RSA	N/A	

17. Buy Domain, Use Route 53

How:

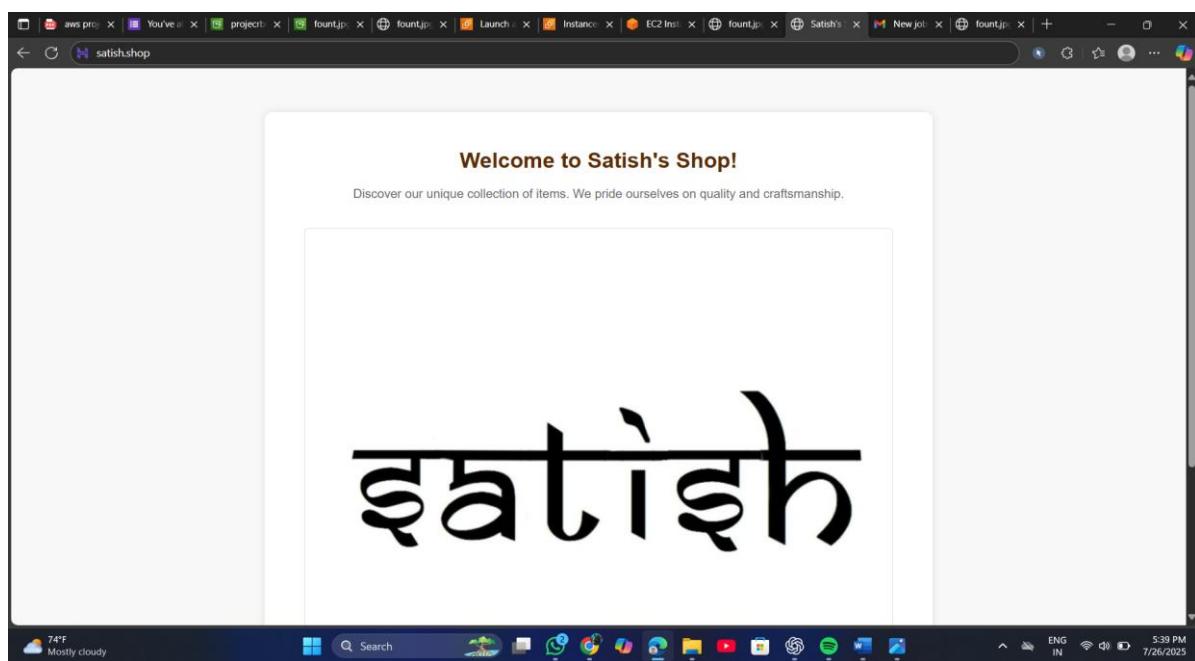
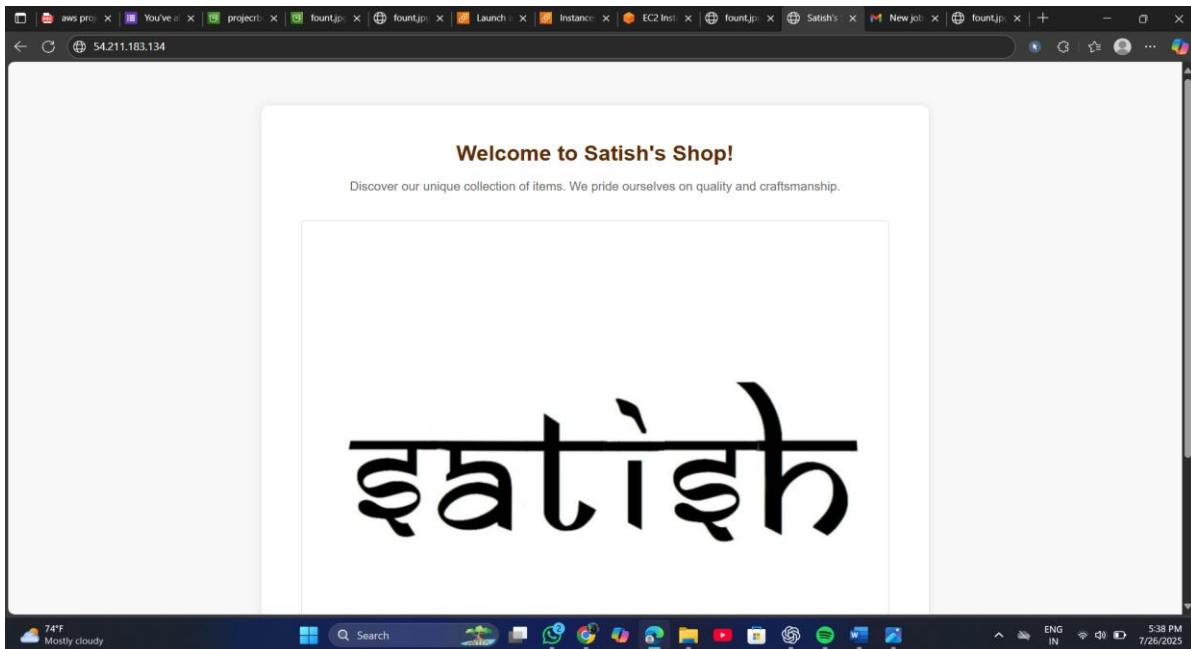
1. In Route 53 > Registered Domains, click Register Domain and search for your desired name.
2. Complete the registration and payment steps.
3. After registration, AWS sets up a Hosted Zone for your domain.
4. If using another registrar, update your domain's nameservers to those provided by AWS Route 53.

18. Configure Route 53 with CloudFront

How:

1. In Route 53 > Hosted zones, select your domain.
2. Create a new record:
 - **Type:** A or AAAA (Alias).
 - **Name:** (e.g., www or leave blank for the root domain).
 - **Alias:** Yes.
 - **Alias Target:** Select your CloudFront distribution's domain name (shows in dropdown if in the same account).
3. Save the record.

AWS Cloud Infrastructure Deployment and Automation



Now, internet users who visit your domain will:

- Be routed via Route 53 (DNS).
- Go through CloudFront (CDN/caching).
- Reach your Application Load Balancer (ALB).
- Get served by available EC2 instances (managed by Auto Scaling Group—ASG).
- Benefit from secure HTTPS (provided by ACM certificate).

Satish Kumar Singilese

Mail: satishkumarcse2003@gmail.com

AWS Cloud Infrastructure Deployment and Automation